

Minnal C++ Servlet Container

Version 1.7.0
by Sans Pareil Technologies, Inc.

Generated by Doxygen 1.8.7

Mon Oct 6 2014 16:58:03

Contents

1	Minnal C++ Servlet Container	1
1.1	Contents	1
1.2	Overview	1
1.3	Implementation	1
1.4	Libraries	1
2	Namespace Index	3
2.1	Namespace List	3
3	Hierarchical Index	5
3.1	Class Hierarchy	5
4	Class Index	7
4.1	Class List	7
5	File Index	11
5.1	File List	11
6	Namespace Documentation	13
6.1	ctemplate Namespace Reference	13
6.2	Poco Namespace Reference	13
6.3	Poco::Util Namespace Reference	13
6.4	spt Namespace Reference	13
6.5	spt::servlet Namespace Reference	14
6.5.1	Detailed Description	16
6.5.2	Typedef Documentation	16
6.5.2.1	ServletPtr	16
6.5.2.2	VirtualHostList	16
6.5.2.3	VirtualHostMap	16
6.5.2.4	VirtualHostPtr	16
6.5.3	Function Documentation	16
6.5.3.1	getCacheCompressedFiles	16
6.5.3.2	getConnectionKeepAlive	17

6.5.3.3	getDenyQueryStrings	17
6.5.3.4	getDocrootHandler	17
6.5.3.5	getDocumentRoot	17
6.5.3.6	getDocumentRootCache	17
6.5.3.7	getMaxQueued	17
6.5.3.8	getMaxThreads	18
6.5.3.9	getMimeTypeMapping	18
6.5.3.10	getServerGroup	18
6.5.3.11	getServerIdentifier	18
6.5.3.12	getServerPort	18
6.5.3.13	getServerUser	18
6.5.3.14	getServletLoggerArchive	19
6.5.3.15	getServletLoggerCompress	19
6.5.3.16	getServletLoggerFile	19
6.5.3.17	getServletLoggerPattern	19
6.5.3.18	getServletLoggerPurgeCount	19
6.5.3.19	getServletLoggerTimes	19
6.5.3.20	getServletLogRotation	19
6.5.3.21	getVersionNumber	20
6.6	spt::servlet::form Namespace Reference	20
6.6.1	Detailed Description	20
6.7	spt::servlet::model Namespace Reference	20
6.7.1	Detailed Description	22
6.7.2	Typedef Documentation	22
6.7.2.1	ContainerConfigurationPtr	22
6.7.2.2	ContextParams	22
6.7.2.3	ErrorPage	23
6.7.2.4	ErrorPagePtr	23
6.7.2.5	HostConfigurationPtr	23
6.7.2.6	HostContextPtr	23
6.7.2.7	InitParams	23
6.7.2.8	ListEntry	23
6.7.2.9	ListPtr	23
6.7.2.10	LoginConfigPtr	23
6.7.2.11	MimeTypeypes	23
6.7.2.12	PathMapping	23
6.7.2.13	Roles	23
6.7.2.14	SecurityConstraintPtr	23
6.7.2.15	SecurityConstraints	24
6.7.2.16	ServerConfigurationPtr	24

6.7.2.17	ServletConfigPtr	24
6.7.2.18	Servlets	24
6.7.2.19	ServletSecurityMap	24
6.7.2.20	ServletSecurityPtr	24
6.7.2.21	UserMap	24
6.7.2.22	UserPtr	24
6.8	spt::servlet::search Namespace Reference	24
6.8.1	Detailed Description	25
6.8.2	Typedef Documentation	25
6.8.2.1	DocRootIndexerPtr	25
6.8.2.2	SearchIndexManagerPtr	25
6.8.3	Function Documentation	26
6.8.3.1	convertToUTF8	26
6.8.3.2	indexable	26
6.8.3.3	operator<	26
6.8.3.4	operator==	26
6.9	spt::servlet::ServletRegistry Namespace Reference	26
6.9.1	Detailed Description	27
6.9.2	Typedef Documentation	27
6.9.2.1	ServletMap	27
6.9.3	Function Documentation	27
6.9.3.1	deRegisterServlet	27
6.9.3.2	getRegisteredServlets	27
6.9.3.3	getServlet	28
6.9.3.4	registerServlet	28
6.9.3.5	writeTimestamp	28
6.10	spt::servlet::templating Namespace Reference	28
6.10.1	Detailed Description	28
7	Class Documentation	29
7.1	spt::servlet::model::AccessLogRecord Class Reference	29
7.1.1	Detailed Description	29
7.1.2	Member Function Documentation	29
7.1.2.1	log	29
7.2	spt::servlet::form::CaptchaServlet Class Reference	30
7.2.1	Detailed Description	32
7.2.2	Member Typedef Documentation	33
7.2.2.1	CaptchaResponse	33
7.2.3	Constructor & Destructor Documentation	33
7.2.3.1	~CaptchaServlet	33

7.2.4	Member Function Documentation	33
7.2.4.1	getPrivateKey	33
7.2.4.2	getPublicKey	33
7.2.4.3	getRemoteHost	33
7.2.4.4	getVerifyUrl	34
7.2.4.5	validateCaptcha	35
7.3	spt::servlet::templating::CMSServlet Class Reference	35
7.3.1	Detailed Description	38
7.3.2	Constructor & Destructor Documentation	38
7.3.2.1	CMSServlet	38
7.3.2.2	~CMSServlet	38
7.3.3	Member Function Documentation	38
7.3.3.1	getLogger	38
7.3.3.2	getServletName	39
7.3.3.3	init	39
7.4	spt::servlet::form::ContactService Class Reference	39
7.4.1	Detailed Description	41
7.4.2	Member Function Documentation	41
7.4.2.1	getServletName	41
7.4.2.2	handleForm	42
7.5	spt::servlet::model::ContainerConfiguration Class Reference	43
7.5.1	Detailed Description	43
7.5.2	Constructor & Destructor Documentation	44
7.5.2.1	ContainerConfiguration	44
7.5.3	Member Function Documentation	45
7.5.3.1	getContentType	45
7.5.3.2	getContentType	45
7.5.3.3	getMimeTypes	45
7.5.3.4	getServerConfiguration	45
7.6	spt::servlet::templating::Datastore Class Reference	46
7.6.1	Detailed Description	46
7.6.2	Member Typedef Documentation	47
7.6.2.1	Configuration	47
7.6.2.2	Dictionary	47
7.6.2.3	Ptr	47
7.6.3	Constructor & Destructor Documentation	47
7.6.3.1	Datastore	47
7.6.4	Member Function Documentation	47
7.6.4.1	getDatastore	47
7.6.4.2	getLastModified	47

7.6.4.3	load	48
7.7	spt::servlet::search::DocRootIndexer Class Reference	48
7.7.1	Detailed Description	49
7.7.2	Constructor & Destructor Documentation	49
7.7.2.1	DocRootIndexer	49
7.7.2.2	~DocRootIndexer	49
7.7.3	Member Function Documentation	50
7.7.3.1	isRunning	50
7.7.3.2	start	50
7.7.3.3	stop	50
7.7.4	Member Data Documentation	50
7.7.4.1	IGNORE_DIRECTORY	50
7.8	spt::servlet::DocRootServlet Class Reference	50
7.8.1	Detailed Description	52
7.8.2	Constructor & Destructor Documentation	52
7.8.2.1	DocRootServlet	52
7.8.2.2	~DocRootServlet	52
7.8.3	Member Function Documentation	52
7.8.3.1	getLogger	52
7.8.3.2	getServletName	52
7.8.3.3	init	53
7.9	spt::servlet::search::DocumentData Class Reference	53
7.9.1	Detailed Description	53
7.9.2	Constructor & Destructor Documentation	54
7.9.2.1	DocumentData	54
7.9.2.2	DocumentData	54
7.9.2.3	DocumentData	54
7.9.3	Member Function Documentation	54
7.9.3.1	deserialise	54
7.9.3.2	getContent	54
7.9.3.3	getContentType	54
7.9.3.4	getFile	54
7.9.3.5	getLastModified	55
7.9.3.6	getTitle	55
7.9.3.7	getUriPath	55
7.9.3.8	serialise	55
7.10	Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr > Class Template Reference	55
7.10.1	Constructor & Destructor Documentation	56
7.10.1.1	DynamicAnyHolderImpl	56
7.10.1.2	~DynamicAnyHolderImpl	56

7.10.2	Member Function Documentation	57
7.10.2.1	clone	57
7.10.2.2	convert	57
7.10.2.3	convert	57
7.10.2.4	convert	57
7.10.2.5	convert	57
7.10.2.6	convert	57
7.10.2.7	convert	57
7.10.2.8	convert	57
7.10.2.9	convert	57
7.10.2.10	convert	57
7.10.2.11	convert	57
7.10.2.12	convert	57
7.10.2.13	convert	57
7.10.2.14	convert	57
7.10.2.15	convert	57
7.10.2.16	convert	57
7.10.2.17	convert	57
7.10.2.18	isArray	57
7.10.2.19	isInteger	57
7.10.2.20	isNumeric	57
7.10.2.21	isSigned	57
7.10.2.22	isString	57
7.10.2.23	type	57
7.10.2.24	value	57
7.11	Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr > Class Template Reference	58
7.11.1	Constructor & Destructor Documentation	59
7.11.1.1	DynamicAnyHolderImpl	59
7.11.1.2	~DynamicAnyHolderImpl	59
7.11.2	Member Function Documentation	59
7.11.2.1	clone	59
7.11.2.2	convert	59
7.11.2.3	convert	59
7.11.2.4	convert	59
7.11.2.5	convert	59
7.11.2.6	convert	59
7.11.2.7	convert	59
7.11.2.8	convert	59
7.11.2.9	convert	59
7.11.2.10	convert	59

7.11.2.11 convert	59
7.11.2.12 convert	59
7.11.2.13 convert	59
7.11.2.14 convert	59
7.11.2.15 convert	59
7.11.2.16 convert	59
7.11.2.17 convert	60
7.11.2.18 isArray	60
7.11.2.19 isInteger	60
7.11.2.20 isNumeric	60
7.11.2.21 isSigned	60
7.11.2.22 isString	60
7.11.2.23 type	60
7.11.2.24 value	60
7.12 spt::servlet::FileServlet Class Reference	60
7.12.1 Detailed Description	62
7.12.2 Member Function Documentation	62
7.12.2.1 checksum	62
7.12.2.2 checkURI	62
7.12.2.3 doGet	63
7.12.2.4 doHead	63
7.12.2.5 doOptions	63
7.13 spt::servlet::form::FormHandler Class Reference	63
7.13.1 Detailed Description	66
7.13.2 Constructor & Destructor Documentation	66
7.13.2.1 ~FormHandler	66
7.13.3 Member Function Documentation	66
7.13.3.1 doOptions	66
7.13.3.2 doPost	66
7.13.3.3 handleForm	67
7.13.3.4 toJson	67
7.13.3.5 writeHeaders	67
7.14 spt::servlet::model::HostConfiguration Class Reference	67
7.14.1 Detailed Description	68
7.14.2 Constructor & Destructor Documentation	68
7.14.2.1 HostConfiguration	68
7.14.3 Member Function Documentation	69
7.14.3.1 getErrorPages	69
7.14.3.2 getLoginConfig	69
7.14.3.3 getSecurityConstraints	69

7.14.3.4	getSecurityMap	69
7.14.3.5	getSecurityMap	69
7.14.3.6	getUserMap	69
7.14.3.7	getWelcomeFiles	69
7.14.3.8	setErrorPages	70
7.14.3.9	setLoginConfig	71
7.14.3.10	setSecurityConstraints	71
7.14.3.11	setUserMap	71
7.14.3.12	setWelcomeFiles	71
7.15	spt::servlet::model::HostContext Class Reference	71
7.15.1	Detailed Description	72
7.15.2	Constructor & Destructor Documentation	72
7.15.2.1	HostContext	72
7.15.2.2	~HostContext	72
7.15.3	Member Function Documentation	73
7.15.3.1	getAccessLogger	73
7.15.3.2	getContainerConfiguration	73
7.15.3.3	getDocumentRoot	73
7.15.3.4	getHostConfiguration	73
7.15.3.5	getHostConfiguration	73
7.15.3.6	getSearchIndexManager	73
7.15.3.7	setDocumentRoot	73
7.15.3.8	setSearchIndexManager	74
7.16	spt::servlet::search::HtmlParser Class Reference	74
7.16.1	Detailed Description	76
7.16.2	Constructor & Destructor Documentation	76
7.16.2.1	HtmlParser	76
7.16.2.2	~HtmlParser	76
7.16.3	Member Function Documentation	76
7.16.3.1	closingTag	76
7.16.3.2	compact	77
7.16.3.3	decodeEntities	77
7.16.3.4	getParameter	77
7.16.3.5	openingTag	77
7.16.3.6	parseHtml	78
7.16.3.7	processText	79
7.16.3.8	toPlainText	79
7.16.3.9	toPlainText	79
7.16.4	Member Data Documentation	79
7.16.4.1	charset	79

7.16.4.2	inScript	79
7.16.4.3	namedEntities	79
7.17	spt::servlet::search::HtmlParserImpl Class Reference	80
7.17.1	Detailed Description	81
7.17.2	Constructor & Destructor Documentation	82
7.17.2.1	HtmlParserImpl	82
7.17.3	Member Function Documentation	82
7.17.3.1	closingTag	82
7.17.3.2	getDescription	82
7.17.3.3	getIgnoreMetaRobots	82
7.17.3.4	getIndexingAllowed	82
7.17.3.5	getKeywords	82
7.17.3.6	getPlainText	83
7.17.3.7	getTitle	83
7.17.3.8	openingTag	83
7.17.3.9	parseHtml	83
7.17.3.10	parseHtml	83
7.17.3.11	processText	84
7.17.3.12	reset	84
7.17.3.13	setIgnoreMetaRobots	84
7.18	Poco::Util::JSONArray Class Reference	84
7.18.1	Member Typedef Documentation	85
7.18.1.1	Ptr	85
7.18.2	Constructor & Destructor Documentation	85
7.18.2.1	JSONArray	85
7.18.2.2	JSONArray	85
7.18.2.3	~JSONArray	85
7.18.3	Member Function Documentation	85
7.18.3.1	add	85
7.18.3.2	get	85
7.18.3.3	getArray	86
7.18.3.4	getElement	86
7.18.3.5	getObject	86
7.18.3.6	isArray	86
7.18.3.7	isNull	86
7.18.3.8	isObject	86
7.18.3.9	optElement	86
7.18.3.10	remove	86
7.18.3.11	size	86
7.18.3.12	stringify	86

7.19 Poco::Util::JSONConfiguration Class Reference	87
7.19.1 Detailed Description	88
7.19.2 Constructor & Destructor Documentation	88
7.19.2.1 JSONConfiguration	88
7.19.2.2 JSONConfiguration	88
7.19.2.3 JSONConfiguration	88
7.19.2.4 JSONConfiguration	88
7.19.2.5 ~JSONConfiguration	88
7.19.3 Member Function Documentation	88
7.19.3.1 enumerate	88
7.19.3.2 getRaw	88
7.19.3.3 load	89
7.19.3.4 load	89
7.19.3.5 loadEmpty	89
7.19.3.6 save	89
7.19.3.7 setBool	89
7.19.3.8 setDouble	89
7.19.3.9 setInt	89
7.19.3.10 setRaw	89
7.20 Poco::Util::JSONObject Class Reference	89
7.20.1 Detailed Description	90
7.20.2 Member Typedef Documentation	90
7.20.2.1 Ptr	90
7.20.3 Constructor & Destructor Documentation	90
7.20.3.1 JSONObject	90
7.20.3.2 JSONObject	90
7.20.3.3 ~JSONObject	90
7.20.4 Member Function Documentation	91
7.20.4.1 get	91
7.20.4.2 getArray	91
7.20.4.3 getNames	91
7.20.4.4 getObject	91
7.20.4.5 getValue	91
7.20.4.6 has	91
7.20.4.7 isArray	91
7.20.4.8 isNull	91
7.20.4.9 isObject	91
7.20.4.10 optValue	91
7.20.4.11 remove	91
7.20.4.12 set	92

7.20.4.13	size	92
7.20.4.14	stringify	92
7.21	Poco::Util::JSONParser Class Reference	92
7.21.1	Detailed Description	92
7.21.2	Constructor & Destructor Documentation	92
7.21.2.1	JSONParser	92
7.21.2.2	~JSONParser	92
7.21.3	Member Function Documentation	93
7.21.3.1	parse	93
7.21.3.2	parse	93
7.22	Poco::Util::JSONQuery Class Reference	93
7.22.1	Detailed Description	93
7.22.2	Constructor & Destructor Documentation	93
7.22.2.1	JSONQuery	93
7.22.2.2	~JSONQuery	94
7.22.3	Member Function Documentation	94
7.22.3.1	find	94
7.22.3.2	findArray	94
7.22.3.3	findObject	94
7.22.3.4	findValue	94
7.22.3.5	findValue	94
7.23	Poco::Util::JSONStringifier Class Reference	94
7.23.1	Detailed Description	94
7.23.2	Member Function Documentation	95
7.23.2.1	stringify	95
7.24	spt::servlet::model::LoginConfig Class Reference	95
7.24.1	Detailed Description	95
7.24.2	Constructor & Destructor Documentation	95
7.24.2.1	LoginConfig	95
7.24.3	Member Function Documentation	96
7.24.3.1	getMethod	96
7.24.3.2	getRealm	96
7.25	spt::servlet::LogManager Class Reference	96
7.25.1	Detailed Description	96
7.25.2	Member Function Documentation	97
7.25.2.1	getLogger	97
7.25.2.2	getLogManager	97
7.26	spt::servlet::OwnershipManager Class Reference	97
7.26.1	Detailed Description	98
7.26.2	Constructor & Destructor Documentation	98

7.26.2.1	OwnershipManager	98
7.26.3	Member Function Documentation	98
7.26.3.1	switchUser	98
7.27	spt::servlet::search::PdfExtractor Class Reference	98
7.27.1	Detailed Description	99
7.27.2	Member Function Documentation	99
7.27.2.1	extractText	99
7.27.2.2	hasCommand	100
7.28	spt::servlet::search::PdfParser Class Reference	100
7.28.1	Detailed Description	100
7.28.2	Constructor & Destructor Documentation	100
7.28.2.1	PdfParser	100
7.28.2.2	~PdfParser	100
7.28.3	Member Function Documentation	100
7.28.3.1	extract	100
7.29	spt::servlet::RequestDispatcher Class Reference	101
7.29.1	Detailed Description	102
7.29.2	Constructor & Destructor Documentation	102
7.29.2.1	RequestDispatcher	102
7.29.3	Member Function Documentation	102
7.29.3.1	createRequestHandler	102
7.30	spt::servlet::search::SearchIndexManager Class Reference	103
7.30.1	Detailed Description	103
7.30.2	Constructor & Destructor Documentation	104
7.30.2.1	SearchIndexManager	104
7.30.2.2	~SearchIndexManager	104
7.30.3	Member Function Documentation	104
7.30.3.1	getDocument	104
7.30.3.2	getLastDocumentId	104
7.30.3.3	getLastModified	104
7.30.3.4	index	105
7.30.3.5	indexIfUpdated	105
7.30.3.6	remove	105
7.30.3.7	remove	105
7.30.3.8	search	106
7.31	spt::servlet::search::SearchResult Class Reference	106
7.31.1	Detailed Description	106
7.31.2	Constructor & Destructor Documentation	107
7.31.2.1	SearchResult	107
7.31.3	Member Function Documentation	107

7.31.3.1	getDocument	107
7.31.3.2	getPercent	107
7.31.3.3	getRank	107
7.31.3.4	getSnippet	107
7.31.3.5	getWeight	108
7.32	spt::servlet::search::SearchResults Class Reference	108
7.32.1	Detailed Description	108
7.32.2	Constructor & Destructor Documentation	109
7.32.2.1	SearchResults	109
7.32.3	Member Function Documentation	110
7.32.3.1	getOffset	110
7.32.3.2	getPageSize	110
7.32.3.3	getQuery	110
7.32.3.4	getResults	110
7.32.3.5	getSearchResult	110
7.32.3.6	getTotal	110
7.33	spt::servlet::search::SearchServlet Class Reference	111
7.33.1	Detailed Description	113
7.33.2	Member Function Documentation	113
7.33.2.1	getServletName	113
7.33.2.2	handleForm	113
7.33.2.3	init	114
7.34	spt::servlet::model::SecurityConstraint Class Reference	114
7.34.1	Detailed Description	114
7.34.2	Constructor & Destructor Documentation	114
7.34.2.1	SecurityConstraint	114
7.34.3	Member Function Documentation	115
7.34.3.1	getAuthConstraint	115
7.34.3.2	getMethod	115
7.34.3.3	getUriPattern	115
7.35	spt::servlet::Servlet Class Reference	115
7.35.1	Detailed Description	118
7.35.2	Member Typedef Documentation	119
7.35.2.1	Range	119
7.35.3	Constructor & Destructor Documentation	119
7.35.3.1	Servlet	119
7.35.3.2	~Servlet	119
7.35.4	Member Function Documentation	119
7.35.4.1	authenticate	119
7.35.4.2	booleanValue	119

7.35.4.3	checkURI	119
7.35.4.4	deflateResponse	120
7.35.4.5	doDelete	120
7.35.4.6	doGet	120
7.35.4.7	doHead	120
7.35.4.8	doOptions	121
7.35.4.9	doPost	121
7.35.4.10	doPut	121
7.35.4.11	doTrace	121
7.35.4.12	getClassName	121
7.35.4.13	getContentType	121
7.35.4.14	getContextParams	122
7.35.4.15	getDocRoot	122
7.35.4.16	getHostConfiguration	122
7.35.4.17	getHostContext	122
7.35.4.18	getInitParams	122
7.35.4.19	getLogger	122
7.35.4.20	getName	122
7.35.4.21	getRealm	123
7.35.4.22	getRequestedRange	123
7.35.4.23	getServerIdentifier	123
7.35.4.24	getServletConfig	123
7.35.4.25	getServletName	123
7.35.4.26	gzipResponse	123
7.35.4.27	init	124
7.35.4.28	intValue	124
7.35.4.29	log	124
7.35.4.30	log	124
7.35.4.31	rangeRequest	124
7.35.4.32	requiresAuthentication	124
7.35.4.33	securityForPath	125
7.35.4.34	sendErrorResponse	125
7.35.4.35	service	125
7.35.4.36	servletSecurity	125
7.35.4.37	setHostContext	126
7.35.4.38	setServletConfig	127
7.35.4.39	stringValue	127
7.35.5	Member Data Documentation	127
7.35.5.1	ACCEPT_RANGES	127
7.35.5.2	ALLOW	127

7.35.5.3	CACHE_CONTROL	127
7.35.5.4	CONTENT_RANGE	127
7.35.5.5	ETAG_HEADER	127
7.35.5.6	IF_MODIFIED_SINCE	127
7.35.5.7	IF_RANGE	128
7.35.5.8	LAST_MODIFIED_HEADER	128
7.35.5.9	MUST_REVALIDATE	128
7.35.5.10	RANGE	128
7.36	spt::servlet::model::ServletConfig Class Reference	128
7.36.1	Detailed Description	128
7.36.2	Constructor & Destructor Documentation	129
7.36.2.1	ServletConfig	129
7.36.3	Member Function Documentation	129
7.36.3.1	getContextParams	129
7.36.3.2	getInitParams	129
7.36.3.3	getPathMapping	129
7.36.3.4	getServletClass	129
7.36.3.5	getServletName	130
7.37	spt::servlet::ServletContainer Class Reference	130
7.37.1	Detailed Description	131
7.37.2	Constructor & Destructor Documentation	131
7.37.2.1	ServletContainer	131
7.37.3	Member Function Documentation	131
7.37.3.1	defineOptions	131
7.37.3.2	handleOption	132
7.37.3.3	initialize	132
7.37.3.4	main	132
7.38	spt::servlet::ServletHandler Class Reference	132
7.38.1	Detailed Description	133
7.38.2	Constructor & Destructor Documentation	134
7.38.2.1	ServletHandler	134
7.38.2.2	~ServletHandler	134
7.38.3	Member Function Documentation	134
7.38.3.1	getForm	134
7.38.3.2	getRequest	134
7.38.3.3	getResponse	134
7.38.3.4	handleRequest	135
7.39	spt::servlet::ServletHolder Struct Reference	136
7.39.1	Detailed Description	136
7.39.2	Constructor & Destructor Documentation	136

7.39.2.1	~ServletHolder	136
7.39.3	Member Function Documentation	136
7.39.3.1	create	136
7.40	spt::servlet::ServletHolderImpl< S > Class Template Reference	137
7.40.1	Detailed Description	138
7.40.2	Constructor & Destructor Documentation	138
7.40.2.1	ServletHolderImpl	138
7.40.2.2	~ServletHolderImpl	138
7.40.3	Member Function Documentation	139
7.40.3.1	create	139
7.40.3.2	getName	139
7.41	spt::servlet::model::ServletSecurity Class Reference	139
7.41.1	Detailed Description	139
7.41.2	Constructor & Destructor Documentation	140
7.41.2.1	ServletSecurity	140
7.41.3	Member Function Documentation	140
7.41.3.1	getAuthConstraint	140
7.41.3.2	getMethod	140
7.41.3.3	getPath	140
7.42	spt::servlet::templating::TemplateManager Class Reference	140
7.42.1	Detailed Description	141
7.42.2	Member Function Documentation	141
7.42.2.1	addDirectory	141
7.42.2.2	expand	142
7.42.2.3	getCache	142
7.42.2.4	getFilePath	142
7.42.2.5	getTemplateManager	142
7.42.2.6	init	142
7.42.2.7	reload	142
7.42.2.8	setInterval	143
7.43	spt::servlet::model::User Class Reference	143
7.43.1	Detailed Description	143
7.43.2	Constructor & Destructor Documentation	144
7.43.2.1	User	144
7.43.3	Member Function Documentation	145
7.43.3.1	getPassword	145
7.43.3.2	getRoles	145
7.43.3.3	getUsername	145
7.44	spt::servlet::model::UserXml Class Reference	145
7.44.1	Detailed Description	145

7.44.2	Member Function Documentation	146
7.44.2.1	getUserMap	146
7.44.2.2	load	146
7.45	spt::servlet::VirtualHostServer Class Reference	146
7.45.1	Detailed Description	146
7.45.2	Constructor & Destructor Documentation	147
7.45.2.1	VirtualHostServer	147
7.45.3	Member Function Documentation	147
7.45.3.1	createServlet	147
7.46	spt::servlet::model::WebXml Class Reference	147
7.46.1	Detailed Description	148
7.46.2	Member Function Documentation	148
7.46.2.1	getContextParams	148
7.46.2.2	getErrorPages	148
7.46.2.3	getLoginConfig	148
7.46.2.4	getSecurityConstraints	149
7.46.2.5	getServlets	149
7.46.2.6	getWelcomeFiles	149
7.46.2.7	load	149
8	File Documentation	151
8.1	/Users/rakesh/svn/cpp/web/servlet/src/api/mainpage.dox File Reference	151
8.2	/Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSON.h File Reference	151
8.2.1	Macro Definition Documentation	151
8.2.1.1	JSON_API	151
8.3	/Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONArray.h File Reference	151
8.4	/Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONConfiguration.h File Reference	152
8.5	/Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONException.h File Reference	153
8.6	/Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONObject.h File Reference	154
8.7	/Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONParser.h File Reference	155
8.8	/Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONQuery.h File Reference	155
8.9	/Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONStringifier.h File Reference	156
8.10	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/DocRootServlet.h File Reference	157
8.11	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/FileServlet.h File Reference	157
8.12	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/form/CaptchaServlet.h File Reference	158
8.13	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/form/ContactService.h File Reference	159
8.14	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/form/FormHandler.h File Reference	160
8.15	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/LogManager.h File Reference	161
8.16	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/AccessLogRecord.h File Reference	161
8.17	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/Configuration.h File Reference	162

8.18	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/ContainerConfiguration.h File Reference	164
8.19	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/HostConfiguration.h File Reference	165
8.20	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/HostContext.h File Reference	166
8.21	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/LoginConfig.h File Reference	168
8.22	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/SecurityConstraint.h File Reference	169
8.23	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/ServletConfig.h File Reference	171
8.24	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/ServletSecurity.h File Reference	172
8.25	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/User.h File Reference	173
8.26	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/UserXml.h File Reference	175
8.27	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/WebXml.h File Reference	176
8.28	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/OwnershipManager.h File Reference	177
8.29	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/RequestDispatcher.h File Reference	178
8.30	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/DocRootIndexer.h File Reference	178
8.31	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/DocumentData.h File Reference	180
8.32	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/HtmlParser.h File Reference	182
8.33	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/HtmlParserImpl.h File Reference	183
8.34	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/NamedEntities.h File Reference	184
8.35	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/PdfExtractor.h File Reference	184
8.36	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/PdfParser.h File Reference	184
8.37	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/SearchIndexManager.h File Reference	185
8.38	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/SearchResult.h File Reference	187
8.39	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/SearchResults.h File Reference	188
8.40	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/SearchServlet.h File Reference	190
8.41	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/Servlet.h File Reference	190
8.42	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/ServletContainer.h File Reference	191
8.43	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/ServletHandler.h File Reference	192
8.44	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/ServletRegistry.h File Reference	193
8.44.1	Macro Definition Documentation	194
8.44.1.1	REGISTER_SERVLET	194
8.45	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/templating/CMSServlet.h File Reference	194
8.46	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/templating/Datastore.h File Reference	195
8.47	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/templating/TemplateManager.h File Reference	196
8.48	/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/VirtualHostServer.h File Reference	196

Chapter 1

Minnal C++ Servlet Container

1.1 Contents

- [Overview](#) Overview
- [Implementation](#) Implementation
- [Libraries](#) Libraries

1.2 Overview

Minnal C++ Servlet Container is a medium performance C++ servlet container built using [Poco](#) HTTPServer implementation. Individual servlets for handling various paths are configured in a JEE standard web.xml file. The container (much like JEE web containers) handle the responsibility of routing the requests to the appropriate servlets based on the request URI and the servlet-mapping configuration in web.xml

1.3 Implementation

The core classes that constitute the container are the following:

- [spt::servlet::ServletContainer](#) is a standard Poco::Util::ServerApplication sub-class. It initialises a [spt::servlet::RequestDispatcher](#) instance, which takes over request handling.
- [spt::servlet::RequestDispatcher](#) is an implementation of Poco::Net::HTTPRequestHandlerFactory. RequestDispatcher parses all virtualHost elements defined in the minnal.xml container configuration file and creates instances of [for each virtual host configured](#). [spt::servlet::VirtualHostServer](#) is our implementation of Poco::Net::HTTPRequestHandlerFactory. The web.xml file is parsed and the servlet name to servlet configuration (path, initialisation parameters, context parameters etc) mapping is stored and used to route client requests to the appropriate [spt::servlet::Servlet](#) instance.
- [spt::servlet::Servlet](#) class implements Poco::Net::AbstractHTTPRequestHandler. The base Servlet class implements the run() method and delegates to the appropriate doXXX method as JEE HTTPServlet does. The default implementation of the doXXX methods returns a HTTP_METHOD_NOT_IMPLEMENTED error. Sub-classes implement the appropriate doXXX methods.

1.4 Libraries

Minnal uses a few very popular libraries in its implementation. The major libraries used include:

- [Poco C++ Libraries](#) ([Boost](#) Licence).
- [Xapian](#) search engine library for site search ([GPL](#)).
- [PoDoFo](#) for extracting text from PDF documents for search indexing ([LGPL](#)).
- [CTemplate](#) templating engine for templated pages and content management. ([New BSD licence](#))
- [C++ Template Unit Test Framework](#) (TUT - [BSD](#) licence).
- [Qt](#) - required for qmake only. The Qt API is not used in Minnal, although it is an excellent API. This may change in future, as we have succeeded in building a usable version of Qt using Oracle SolarisStudio compiler suite for Solaris 11, which is our deployment platform.

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

ctemplate	13
Poco	13
Poco::Util	13
spt	13
spt::servlet	
Minnal C++ Servlet Container implementation	14
spt::servlet::form	
Servlets and supporting classes for form handling	20
spt::servlet::model	
Namespace for value objects used to represent various configuration information for the server. Minnal is configured through a server wide configuration file, and additional <code>web.xml</code> style configuration files for each virtual host configured in the server	20
spt::servlet::search	
Namespace for classes that add support for full-text search indexing of content served by the server	24
spt::servlet::ServletRegistry	
Namespace that provides functions for registering servlets. Servlets registered may be fetched from the registry by the fully qualified class name	26
spt::servlet::templating	
Templating system for Minnal servlet container	28

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

AbstractConfiguration	
Poco::Util::JSONConfiguration	87
spt::servlet::model::AccessLogRecord	29
spt::servlet::model::ContainerConfiguration	43
spt::servlet::templating::Datastore	46
spt::servlet::search::DocRootIndexer	48
spt::servlet::search::DocumentData	53
DynamicAnyHolder	
Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >	55
Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >	58
spt::servlet::model::HostConfiguration	67
spt::servlet::model::HostContext	71
spt::servlet::search::HtmlParser	74
spt::servlet::search::HtmlParserImpl	80
HTTPRequestHandler	
spt::servlet::ServletHandler	132
HTTPRequestHandlerFactory	
spt::servlet::RequestDispatcher	101
Poco::Util::JSONArray	84
Poco::Util::JSONObject	89
Poco::Util::JSONParser	92
Poco::Util::JSONQuery	93
Poco::Util::JSONStringifier	94
spt::servlet::model::LoginConfig	95
spt::servlet::LogManager	96
spt::servlet::OwnershipManager	97
spt::servlet::search::PdfExtractor	98
spt::servlet::search::PdfParser	100
spt::servlet::search::SearchIndexManager	103
spt::servlet::search::SearchResult	106
spt::servlet::search::SearchResults	108
spt::servlet::model::SecurityConstraint	114
ServerApplication	
spt::servlet::ServletContainer	130
spt::servlet::Servlet	115
spt::servlet::FileServlet	60
spt::servlet::DocRootServlet	50

spt::servlet::templating::CMSServlet	35
spt::servlet::form::FormHandler	63
spt::servlet::form::CaptchaServlet	30
spt::servlet::form::ContactService	39
spt::servlet::search::SearchServlet	111
spt::servlet::model::ServletConfig	128
spt::servlet::ServletHolder	136
spt::servlet::ServletHolderImpl< S >	137
spt::servlet::model::ServletSecurity	139
spt::servlet::templating::TemplateManager	140
spt::servlet::model::User	143
spt::servlet::model::UserXml	145
spt::servlet::VirtualHostServer	146
spt::servlet::model::WebXml	147

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

spt::servlet::model::AccessLogRecord	Utility class that provides support for generating virtual host access logs in common log format	29
spt::servlet::form::CaptchaServlet	A base servlet class that adds support for captcha validation. Sub-classes should use the validateCaptcha method to validate the client captcha response in addition to any other business rules	30
spt::servlet::templating::CMSServlet	The Servlet that builds dynamic output based on files with extension <code>cms</code> stored under the virtual host document root	35
spt::servlet::form::ContactService	A handler for contact forms	39
spt::servlet::model::ContainerConfiguration	Value object that stores the various configuration objects used to manage the servlet container	43
spt::servlet::templating::Datastore	Factory class for interacting with backend systems used to store/retrieve data used to fill templates	46
spt::servlet::search::DocRootIndexer	A background indexer used to index files under the specified document root directory	48
spt::servlet::DocRootServlet	Default servlet for serving static files	50
spt::servlet::search::DocumentData	A simple POD used to store data relevant to a document indexed in the Xapian search engine	53
Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >		55
Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >		58
spt::servlet::FileServlet	Abstract base servlet for serving files	60
spt::servlet::form::FormHandler	A base form handler servlet	63
spt::servlet::model::HostConfiguration	A value object that represents the configured values for a virtual host in the container. The configuration is parsed from the <code>web.xml</code> and <code>user.xml</code> files configured for a virtual host	67
spt::servlet::model::HostContext	A value object that stores the information pertinent to a virtual host	71
spt::servlet::search::HtmlParser	A parser for extracting the plain text contents and some meta-data from a HTML string	74
spt::servlet::search::HtmlParserImpl	A more robust HTML parser implementation based on the Xapian Omega search engine project	80
Poco::Util::JSONArray		84

Poco::Util::JSONConfiguration	
This configuration class extracts configuration properties from a JSON object. An XPath-like syntax for property names is supported to allow full access to the JSON object	87
Poco::Util::JSONObject	
Represents a JSON object	89
Poco::Util::JSONParser	
A class for passing JSON strings or streams	92
Poco::Util::JSONQuery	
Class that can be used to search for a value in a JSON object or array	93
Poco::Util::JSONStringifier	
Helper class for creating a String from a JSON object or array	94
spt::servlet::model::LoginConfig	
A simple value object that represents a login-config element in a web.xml file	95
spt::servlet::LogManager	
A singleton class that is used to set up the loggers for the container	96
spt::servlet::OwnershipManager	
A utility class used to handle switching of user from root to unprivileged user after the server has started up and bound to the desired port	97
spt::servlet::search::PdfExtractor	
Extract text content from PDF files using pdftotext (part of xpdf or the newer poppler system). Return the extracted plain text content	98
spt::servlet::search::PdfParser	
A parser for extracting the text contents of a PDF file using PoDoFo	100
spt::servlet::RequestDispatcher	
The standard request handler factory that will parse the Host HTTP header and delegate to the appropriate VirtualHost to process the request. The first configured VirtualHost will be used if no Host header has been specified	101
spt::servlet::search::SearchIndexManager	
A manager class for maintaining the Xapian search engine index. Usually used to manage indices for one site/virtual host	103
spt::servlet::search::SearchResult	
A wrapper class that encapsulates a document that is part of the search results returned by Xapian for a query	106
spt::servlet::search::SearchResults	
A wrapper class that encapsulates the search results returned by Xapian for a query	108
spt::servlet::search::SearchServlet	
The servlet that handles site-search for each virtual host. Initialises a DocRootIndexer at start to index all static files. This leads to the issue that pages that are not reachable through the site also gets indexed and served through site search	111
spt::servlet::model::SecurityConstraint	
A simple value object used to store security constraint declarations in a web.xml file	114
spt::servlet::Servlet	
The base servlet class from which all concrete servlet instances are to be derived	115
spt::servlet::model::ServletConfig	
A simple value object that represents a servlet configuration in a web.xml file. This object stores the information from the servlet and servlet-mapping elements	128
spt::servlet::ServletContainer	
The main servlet container implementation. Loads the properties file and web.xml file from directory specified in constructor. Hands off to RequestDispatcher to perform the actual request handling	130
spt::servlet::ServletHandler	
The temporary per request handler instance that is required by the Poco handler factory	132
spt::servlet::ServletHolder	
A base holder for creating servlet instances	136
spt::servlet::ServletHolderImpl< S >	
A template class that is used to register a servlet class. Used by the REGISTER_SERVLET macro to create a new instance and register by class name	137

spt::servlet::model::ServletSecurity	
A simple value object that holds the security configuration for a path that has been mapped to a servlet	139
spt::servlet::templating::TemplateManager	
A manager class used to abstract interactions with the templating engine	140
spt::servlet::model::User	
A simple value object that represents a user configured in the users.xml file	143
spt::servlet::model::UserXml	
Parser for a users.xml file based on tomcat-users.xml format. This is the default user database supported by the container	145
spt::servlet::VirtualHostServer	
Handler for each virtual host handled by Minnal	146
spt::servlet::model::WebXml	
Parser for a JEE standard web.xml servlet configuration file. Note that not all JEE configuration is supported by the servlet container	147

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

/Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSON.h	151
/Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONArray.h	151
/Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONConfiguration.h	152
/Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONException.h	153
/Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONObject.h	154
/Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONParser.h	155
/Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONQuery.h	155
/Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONStringifier.h	156
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/DocRootServlet.h	157
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/FileServlet.h	157
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/LogManager.h	161
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/OwnershipManager.h	177
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/RequestDispatcher.h	178
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/Servlet.h	190
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/ServletContainer.h	191
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/ServletHandler.h	192
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/ServletRegistry.h	193
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/VirtualHostServer.h	196
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/form/CaptchaServlet.h	158
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/form/ContactService.h	159
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/form/FormHandler.h	160
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/AccessLogRecord.h	161
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/Configuration.h	162
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/ContainerConfiguration.h	164
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/HostConfiguration.h	165
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/HostContext.h	166
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/LoginConfig.h	168
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/SecurityConstraint.h	169
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/ServletConfig.h	171
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/ServletSecurity.h	172
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/User.h	173
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/UserXml.h	175
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/WebXml.h	176
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/DocRootIndexer.h	178
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/DocumentData.h	180
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/HtmlParser.h	182
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/HtmlParserImpl.h	183
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/NamedEntities.h	184

/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/PdfExtractor.h	184
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/PdfParser.h	184
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/SearchIndexManager.h	185
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/SearchResult.h	187
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/SearchResults.h	188
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/SearchServlet.h	190
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/templating/CMSServlet.h	194
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/templating/Datastore.h	195
/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/templating/TemplateManager.h	196

Chapter 6

Namespace Documentation

6.1 ctemplate Namespace Reference

6.2 Poco Namespace Reference

Namespaces

- [Util](#)

Classes

- class [DynamicAnyHolderImpl](#)< [Util::JSONArray::Ptr](#) >
- class [DynamicAnyHolderImpl](#)< [Util::JSONObject::Ptr](#) >

6.3 Poco::Util Namespace Reference

Classes

- class [JSONArray](#)
- class [JSONConfiguration](#)
This configuration class extracts configuration properties from a JSON object. An XPath-like syntax for property names is supported to allow full access to the JSON object.
- class [JSONObject](#)
Represents a JSON object.
- class [JSONParser](#)
A class for passing JSON strings or streams.
- class [JSONQuery](#)
Class that can be used to search for a value in a JSON object or array.
- class [JSONStringifier](#)
Helper class for creating a String from a JSON object or array.

6.4 spt Namespace Reference

Namespaces

- [servlet](#)

Minnal C++ [Servlet](#) Container implementation.

6.5 spt::servlet Namespace Reference

Minnal C++ [Servlet](#) Container implementation.

Namespaces

- [form](#)
Servlets and supporting classes for form handling.
- [model](#)
Namespace for value objects used to represent various configuration information for the server. Minnal is configured through a server wide configuration file, and additional `web.xml` style configuration files for each virtual host configured in the server.
- [search](#)
Namespace for classes that add support for full-text search indexing of content served by the server.
- [ServletRegistry](#)
Namespace that provides functions for registering servlets. Servlets registered may be fetched from the registry by the fully qualified class name.
- [templating](#)
Templating system for Minnal servlet container.

Classes

- class [DocRootServlet](#)
Default servlet for serving static files.
- class [FileServlet](#)
Abstract base servlet for serving files.
- class [LogManager](#)
A singleton class that is used to set up the loggers for the container.
- class [OwnershipManager](#)
A utility class used to handle switching of user from root to unprivileged user after the server has started up and bound to the desired port.
- class [RequestDispatcher](#)
The standard request handler factory that will parse the Host HTTP header and delegate to the appropriate VirtualHost to process the request. The first configured VirtualHost will be used if no Host header has been specified.
- class [Servlet](#)
The base servlet class from which all concrete servlet instances are to be derived.
- class [ServletContainer](#)
The main servlet container implementation. Loads the properties file and web.xml file from directory specified in constructor. Hands off to [RequestDispatcher](#) to perform the actual request handling.
- class [ServletHandler](#)
The temporary per request handler instance that is required by the [Poco](#) handler factory.
- struct [ServletHolder](#)
A base holder for creating servlet instances.
- class [ServletHolderImpl](#)
A template class that is used to register a servlet class. Used by the [REGISTER_SERVLET](#) macro to create a new instance and register by class name.
- class [VirtualHostServer](#)
Handler for each virtual host handled by Minnal.

Typedefs

- typedef Poco::SharedPtr
< [VirtualHostServer](#) > [VirtualHostPtr](#)
Shared pointer to a virtual host instance.
- typedef std::vector
< [VirtualHostPtr](#) > [VirtualHostList](#)
A collection of virtual hosts.
- typedef Poco::HashMap
< std::string, [VirtualHostPtr](#) > [VirtualHostMap](#)
Mapping of virtual hosts instances to their names.
- typedef Poco::SharedPtr< [Servlet](#) > [ServletPtr](#)
Shared pointer to a servlet instance.

Functions

- std::string & [getVersionNumber](#) ()
Return the current version number for Minnal.
- std::string & [getServerIdentifier](#) ()
Return the standard server identifier which will be sent back as part of standard HTTP response header.
- std::string & [getServerPort](#) ()
Return the key used to configure the port on which the server port listens to in the `minnal.xml` file.
- std::string & [getServerUser](#) ()
Return the key used to configure the user used to own the servlet container process. The process is meant to be started by the `root` user, and once the process has bound to the `port`, the effective group and user id of the process will be switched to the configured user and group values.
- std::string & [getServerGroup](#) ()
Return the key used to configure the group used to own the servlet container process. The process is meant to be started by the `root` user, and once the process has bound to the `port`, the effective group and user id of the process will be switched to the configured user and group values.
- std::string & [getConnectionKeepAlive](#) ()
Return the key used to configure whether `keep-alive` support is enabled for client connections.
- std::string & [getMaxQueued](#) ()
Return the maximum size of request queue to be maintained by the server before rejecting client connections.
- std::string & [getMaxThreads](#) ()
Return the key used to configure the maximum number of worker threads the server process will maintain to serve requests.
- std::string & [getMimeTypeMapping](#) ()
Return the key used to configure the mime type mapping file used to look up the mime-type for files.
- std::string & [getServletLogFile](#) ()
Return the key used to configure the name of the log file for the virtual host.
- std::string & [getServletLogRotation](#) ()
Return the key used to configure the log file rotation interval for the virtual host.
- std::string & [getServletLoggerArchive](#) ()
Return the key used to configure the log file archival policy for the virtual host.
- std::string & [getServletLoggerCompress](#) ()
Return the key used to configure whether the archived log files are to be compressed for the virtual host.
- std::string & [getServletLoggerPurgeCount](#) ()
Return the key used to configure the number of archived log files to be maintained for the virtual host.
- std::string & [getServletLoggerTimes](#) ()
Return the key used to configure the time zone to be used for log messages for the virtual host.
- std::string & [getServletLoggerPattern](#) ()

Return the key used to configure the log record format for log messages for the virtual host.

- `std::string & getCacheCompressedFiles ()`

Return the key used to configure whether the server should attempt to cache compressed versions of documents served to clients.

- `std::string & getDocumentRoot ()`

Return the key used to configure the document root directory for the virtual host.

- `std::string & getDocumentRootCache ()`

Return the key used to configure the cache directory for the virtual host.

- `std::string & getDenyQueryStrings ()`

Return the key used to configure whether the server should deny all requests that have query strings for the virtual host.

- `std::string & getDocrootHandler ()`

Return the key used to configure the servlet used to server static files for the virtual host.

6.5.1 Detailed Description

Minnal C++ [Servlet](#) Container implementation.

Namespace with the main classes used to implement the Minnal C++ [Servlet](#) Container. The container presents a JEE style base `spt::servlet::Servlet` class from which concrete implementations are derived to support application requirements. A JEE style `web.xml` file is used to configure the servlet to HTTP request URI mapping.

6.5.2 Typedef Documentation

6.5.2.1 `typedef Poco::SharedPtr<Servlet> spt::servlet::ServletPtr`

Shared pointer to a servlet instance.

6.5.2.2 `typedef std::vector<VirtualHostPtr> spt::servlet::VirtualHostList`

A collection of virtual hosts.

6.5.2.3 `typedef Poco::HashMap<std::string,VirtualHostPtr> spt::servlet::VirtualHostMap`

Mapping of virtual hosts instances to their names.

6.5.2.4 `typedef Poco::SharedPtr<VirtualHostServer> spt::servlet::VirtualHostPtr`

Shared pointer to a virtual host instance.

6.5.3 Function Documentation

6.5.3.1 `std::string& spt::servlet::getCacheCompressedFiles ( ) [inline]`

Return the key used to configure whether the server should attempt to cache compressed versions of documents served to clients.

Returns

`std::string` The configuration key used in `minnal.xml`

6.5.3.2 std::string& spt::servlet::getConnectionKeepAlive () [inline]

Return the key used to configure whether `keep-alive` support is enabled for client connections.

Returns

std::string The configuration key used in `minnal.xml`

6.5.3.3 std::string& spt::servlet::getDenyQueryStrings () [inline]

Return the key used to configure whether the server should deny all requests that have query strings for the virtual host.

Returns

std::string The configuration key used in `minnal.xml`

6.5.3.4 std::string& spt::servlet::getDocrootHandler () [inline]

Return the key used to configure the servlet used to server static files for the virtual host.

Returns

std::string The configuration key used in `minnal.xml`

6.5.3.5 std::string& spt::servlet::getDocumentRoot () [inline]

Return the key used to configure the document root directory for the virtual host.

Returns

std::string The configuration key used in `minnal.xml`

6.5.3.6 std::string& spt::servlet::getDocumentRootCache () [inline]

Return the key used to configure the cache directory for the virtual host.

Returns

std::string The configuration key used in `minnal.xml`

6.5.3.7 std::string& spt::servlet::getMaxQueued () [inline]

Return the maximum size of request queue to be maintained by the server before rejecting client connections.

Returns

std::string The configuration key used in `minnal.xml`

6.5.3.8 `std::string& spt::servlet::getMaxThreads ( ) [inline]`

Return the key used to configure the maximum number of worker threads the server process will maintain to serve requests.

Returns

`std::string` The configuration key used in `minnal.xml`

6.5.3.9 `std::string& spt::servlet::getMimeTypeMapping ( ) [inline]`

Return the key used to configure the mime type mapping file used to look up the mime-type for files.

Returns

`std::string` The configuration key used in `minnal.xml`

6.5.3.10 `std::string& spt::servlet::getServerGroup ( ) [inline]`

Return the key used to configure the group used to own the servlet container process. The process is meant to be started by the `root` user, and once the process has bound to the `port`, the effective group and user id of the process will be switched to the configured user and group values.

Returns

`std::string` The key for the group to switch process ownership to

6.5.3.11 `std::string& spt::servlet::getServerIdentifier ( ) [inline]`

Return the standard server identifier which will be sent back as part of standard HTTP response header.

Returns

`std::string` The server identifier.

6.5.3.12 `std::string& spt::servlet::getServerPort ( ) [inline]`

Return the key used to configure the port on which the server port listens to in the `minnal.xml` file.

Returns

`std::string` The key to use to look up the configured value.

6.5.3.13 `std::string& spt::servlet::getServerUser ( ) [inline]`

Return the key used to configure the user used to own the servlet container process. The process is meant to be started by the `root` user, and once the process has bound to the `port`, the effective group and user id of the process will be switched to the configured user and group values.

Returns

`std::string` The key for the user to switch the process ownership to

6.5.3.14 `std::string& spt::servlet::getServletLoggerArchive ( ) [inline]`

Return the key used to configure the log file archival policy for the virtual host.

Returns

`std::string` The configuration key used in `minnal.xml`

6.5.3.15 `std::string& spt::servlet::getServletLoggerCompress ( ) [inline]`

Return the key used to configure whether the archived log files are to be compressed for the virtual host.

Returns

`std::string` The configuration key used in `minnal.xml`

6.5.3.16 `std::string& spt::servlet::getServletLoggerFile ( ) [inline]`

Return the key used to configure the name of the log file for the virtual host.

Returns

`std::string` The configuration key used in `minnal.xml`

6.5.3.17 `std::string& spt::servlet::getServletLoggerPattern ( ) [inline]`

Return the key used to configure the log record format for log messages for the virtual host.

Returns

`std::string` The configuration key used in `minnal.xml`

6.5.3.18 `std::string& spt::servlet::getServletLoggerPurgeCount ( ) [inline]`

Return the key used to configure the number of archived log files to be maintained for the virtual host.

Returns

`std::string` The configuration key used in `minnal.xml`

6.5.3.19 `std::string& spt::servlet::getServletLoggerTimes ( ) [inline]`

Return the key used to configure the time zone to be used for log messages for the virtual host.

Returns

`std::string` The configuration key used in `minnal.xml`

6.5.3.20 `std::string& spt::servlet::getServletLogRotation ( ) [inline]`

Return the key used to configure the log file rotation interval for the virtual host.

Returns

`std::string` The configuration key used in `minnal.xml`

6.5.3.21 `std::string& spt::servlet::getVersionNumber ( )` `[inline]`

Return the current version number for Minnal.

Returns

`std::string` The current version number

6.6 `spt::servlet::form` Namespace Reference

Servlets and supporting classes for form handling.

Classes

- class [CaptchaServlet](#)
A base servlet class that adds support for captcha validation. Sub-classes should use the [validateCaptcha](#) method to validate the client captcha response in addition to any other business rules.
- class [ContactService](#)
A handler for contact forms.
- class [FormHandler](#)
A base form handler servlet.

6.6.1 Detailed Description

Servlets and supporting classes for form handling.

Namespace with the main classes used to support various types of form POST requests. Common features such as contact forms, polls etc. are supported.

Since

Version 1.6

6.7 `spt::servlet::model` Namespace Reference

Namespace for value objects used to represent various configuration information for the server. Minnal is configured through a server wide configuration file, and additional `web.xml` style configuration files for each virtual host configured in the server.

Classes

- class [AccessLogRecord](#)
Utility class that provides support for generating virtual host access logs in common log format.
- class [ContainerConfiguration](#)
Value object that stores the various configuration objects used to manage the servlet container.
- class [HostConfiguration](#)
A value object that represents the configured values for a virtual host in the container. The configuration is parsed from the `web.xml` and `user.xml` files configured for a virtual host.
- class [HostContext](#)
A value object that stores the information pertinent to a virtual host.
- class [LoginConfig](#)
A simple value object that represents a login-config element in a `web.xml` file.

- class [SecurityConstraint](#)
A simple value object used to store security constraint declarations in a web.xml file.
- class [ServletConfig](#)
A simple value object that represents a servlet configuration in a web.xml file. This object stores the information from the servlet and servlet-mapping elements.
- class [ServletSecurity](#)
A simple value object that holds the security configuration for a path that has been mapped to a servlet.
- class [User](#)
A simple value object that represents a user configured in the users.xml file.
- class [UserXml](#)
Parser for a users.xml file based on tomcat-users.xml format. This is the default user database supported by the container.
- class [WebXml](#)
Parser for a JEE standard web.xml servlet configuration file. Note that not all JEE configuration is supported by the servlet container.

Typedefs

- typedef Poco::AutoPtr
< Poco::Util::XMLConfiguration > [ServerConfigurationPtr](#)
Auto pointer to the xml configuration data for the server.
- typedef
Poco::Util::LayeredConfiguration [MimeTypes](#)
The mime-type configuration data.
- typedef Poco::SharedPtr
< [ContainerConfiguration](#) > [ContainerConfigurationPtr](#)
Shared pointer to a container configuration instance.
- typedef Poco::SharedPtr
< [HostConfiguration](#) > [HostConfigurationPtr](#)
Shared pointer to a host configuration instance.
- typedef Poco::SharedPtr
< [HostContext](#) > [HostContextPtr](#)
Shared pointer to a host context instance.
- typedef Poco::SharedPtr
< [LoginConfig](#) > [LoginConfigPtr](#)
Shared pointer to a login configuration instance.
- typedef Poco::HashSet
< std::string > [ListEntry](#)
Set of string values.
- typedef Poco::SharedPtr
< [ListEntry](#) > [ListPtr](#)
Shared pointer to a set of string values.
- typedef Poco::SharedPtr
< [SecurityConstraint](#) > [SecurityConstraintPtr](#)
Shared pointer to a security constraint instance.
- typedef Poco::HashMap
< std::string, std::string > [ContextParams](#)
Mapping for context-param name-value pairs for each virtual host.
- typedef Poco::HashMap
< std::string, std::string > [InitParams](#)
Mapping of init-param name-value pairs for each configured servlet.

- typedef Poco::HashSet
 < std::string, Poco::Hash
 < std::string > > [PathMapping](#)
Set of paths mapped for a servlet.
- typedef Poco::SharedPtr
 < [ServletConfig](#) > [ServletConfigPtr](#)
Shared pointer to a servlet configuration instance.
- typedef Poco::SharedPtr
 < [ServletSecurity](#) > [ServletSecurityPtr](#)
Shared pointer to a servlet security instance.
- typedef Poco::HashMap
 < std::string,
[ServletSecurityPtr](#) > [ServletSecurityMap](#)
Map for path to servlet security configuration look up.
- typedef Poco::HashSet
 < std::string > [Roles](#)
The roles used to categorise users. Usually used for authorisation.
- typedef Poco::SharedPtr< [User](#) > [UserPtr](#)
Shared pointer to a user instance.
- typedef Poco::HashMap
 < std::string, [UserPtr](#) > [UserMap](#)
Map for looking up a user instance by username.
- typedef Poco::HashMap
 < std::string,
[ServletConfigPtr](#) > [Servlets](#)
Map for servlet name to servlet configuration information.
- typedef Poco::HashMap< int,
 std::string > [ErrorPage](#)
Map for error code to error page lookup.
- typedef Poco::SharedPtr
 < [ErrorPage](#) > [ErrorPagePtr](#)
Shared pointer for error page configuration instance.
- typedef std::vector
 < [SecurityConstraintPtr](#) > [SecurityConstraints](#)
A collection of security constraints.

6.7.1 Detailed Description

Namespace for value objects used to represent various configuration information for the server. Minnal is configured through a server wide configuration file, and additional `web.xml` style configuration files for each virtual host configured in the server.

6.7.2 Typedef Documentation

6.7.2.1 typedef Poco::SharedPtr<ContainerConfiguration> spt::servlet::model::ContainerConfigurationPtr

Shared pointer to a container configuration instance.

6.7.2.2 typedef Poco::HashMap<std::string, std::string> spt::servlet::model::ContextParams

Mapping for context-param name-value pairs for each virtual host.

6.7.2.3 `typedef Poco::HashMap<int,std::string> spt::servlet::model::ErrorPage`

Map for error code to error page lookup.

6.7.2.4 `typedef Poco::SharedPtr<ErrorPage> spt::servlet::model::ErrorPagePtr`

Shared pointer for error page configuration instance.

6.7.2.5 `typedef Poco::SharedPtr<HostConfiguration> spt::servlet::model::HostConfigurationPtr`

Shared pointer to a host configuration instance.

6.7.2.6 `typedef Poco::SharedPtr<HostContext> spt::servlet::model::HostContextPtr`

Shared pointer to a host context instance.

6.7.2.7 `typedef Poco::HashMap<std::string, std::string> spt::servlet::model::InitParams`

Mapping of init-param name-value pairs for each configured servlet.

6.7.2.8 `typedef Poco::HashSet<std::string> spt::servlet::model::ListEntry`

Set of string values.

6.7.2.9 `typedef Poco::SharedPtr<ListEntry> spt::servlet::model::ListPtr`

Shared pointer to a set of string values.

6.7.2.10 `typedef Poco::SharedPtr<LoginConfig> spt::servlet::model::LoginConfigPtr`

Shared pointer to a login configuration instance.

6.7.2.11 `typedef Poco::Util::LayeredConfiguration spt::servlet::model::MimeTypes`

The mime-type configuration data.

6.7.2.12 `typedef Poco::HashSet<std::string, Poco::Hash<std::string> > spt::servlet::model::PathMapping`

Set of paths mapped for a servlet.

6.7.2.13 `typedef Poco::HashSet<std::string> spt::servlet::model::Roles`

The roles used to categorise users. Usually used for authorisation.

6.7.2.14 `typedef Poco::SharedPtr<SecurityConstraint> spt::servlet::model::SecurityConstraintPtr`

Shared pointer to a security constraint instance.

6.7.2.15 `typedef std::vector<SecurityConstraintPtr> spt::servlet::model::SecurityConstraints`

A collection of security constraints.

6.7.2.16 `typedef Poco::AutoPtr<Poco::Util::XMLConfiguration> spt::servlet::model::ServerConfigurationPtr`

Auto pointer to the xml configuration data for the server.

6.7.2.17 `typedef Poco::SharedPtr<ServletConfig> spt::servlet::model::ServletConfigPtr`

Shared pointer to a servlet configuration instance.

6.7.2.18 `typedef Poco::HashMap<std::string, ServletConfigPtr> spt::servlet::model::Servlets`

Map for servlet name to servlet configuration information.

6.7.2.19 `typedef Poco::HashMap<std::string, ServletSecurityPtr> spt::servlet::model::ServletSecurityMap`

Map for path to servlet security configuration look up.

6.7.2.20 `typedef Poco::SharedPtr<ServletSecurity> spt::servlet::model::ServletSecurityPtr`

Shared pointer to a servlet security instance.

6.7.2.21 `typedef Poco::HashMap<std::string, UserPtr> spt::servlet::model::UserMap`

Map for looking up a user instance by username.

6.7.2.22 `typedef Poco::SharedPtr<User> spt::servlet::model::UserPtr`

Shared pointer to a user instance.

6.8 `spt::servlet::search` Namespace Reference

Namespace for classes that add support for full-text search indexing of content served by the server.

Classes

- class [DocRootIndexer](#)
A background indexer used to index files under the specified document root directory.
- class [DocumentData](#)
A simple POD used to store data relevant to a document indexed in the Xapian search engine.
- class [HtmlParser](#)
A parser for extracting the plain text contents and some meta-data from a HTML string.
- class [HtmlParserImpl](#)
A more robust HTML parser implementation based on the Xapian Omega search engine project.
- class [PdfExtractor](#)
Extract text content from PDF files using pdftotext (part of xpdf or the newer poppler system). Return the extracted plain text content.

- class [PdfParser](#)
A parser for extracting the text contents of a PDF file using PoDoFo.
- class [SearchIndexManager](#)
A manager class for maintaining the Xapian search engine index. Usually used to manage indices for one site/virtual host.
- class [SearchResult](#)
A wrapper class that encapsulates a document that is part of the search results returned by Xapian for a query.
- class [SearchResults](#)
A wrapper class that encapsulates the search results returned by Xapian for a query.
- class [SearchServlet](#)
The servlet that handles site-search for each virtual host. Initialises a [DocRootIndexer](#) at start to index all static files. This leads to the issue that pages that are not reachable through the site also gets indexed and served through site search.

Typedefs

- typedef Poco::SharedPtr
 < [DocRootIndexer](#) > [DocRootIndexerPtr](#)
 Shared pointer to a document root indexer instance.
- typedef Poco::SharedPtr
 < [SearchIndexManager](#) > [SearchIndexManagerPtr](#)
 Shared pointer to a search index manager.

Functions

- bool [operator==](#) (const [DocumentData](#) &lhs, const [DocumentData](#) &rhs)
 Compare two document data instances for equality.
- bool [operator<](#) (const [DocumentData](#) &lhs, const [DocumentData](#) &rhs)
 Compare two document data instances for ordering. Compares by the file names, followed by modification timestamp if they have same name.
- bool [indexable](#) (const std::string &contentType)
 Common function used to determine if the specified content type is indexable by the search engine.
- void [convertToUTF8](#) (std::string &text, const std::string &charset)
 Convert the specified text in the specified charset to UTF-8.

6.8.1 Detailed Description

Namespace for classes that add support for full-text search indexing of content served by the server.

Since

1.5

6.8.2 Typedef Documentation

6.8.2.1 typedef Poco::SharedPtr<[DocRootIndexer](#)> spt::servlet::search::DocRootIndexerPtr

Shared pointer to a document root indexer instance.

6.8.2.2 typedef Poco::SharedPtr<[SearchIndexManager](#)> spt::servlet::search::SearchIndexManagerPtr

Shared pointer to a search index manager.

6.8.3 Function Documentation

6.8.3.1 void spt::servlet::search::convertToUTF8 (std::string & *text*, const std::string & *charset*)

Convert the specified text in the specified charset to UTF-8.

Parameters

<i>text</i>	The text to convert (in-place)
<i>charset</i>	The charset used for the text.

6.8.3.2 bool spt::servlet::search::indexable (const std::string & *contentType*)

Common function used to determine if the specified content type is indexable by the search engine.

Parameters

<i>contentType</i>	The content type of the file/resource to be checked for indexability
--------------------	--

Returns

bool Return `true` if the content type is supported.

6.8.3.3 bool spt::servlet::search::operator< (const DocumentData & *lhs*, const DocumentData & *rhs*)

Compare two document data instances for ordering. Compares by the file names, followed by modification timestamp if they have same name.

Parameters

<i>lhs</i>	The left hand document data instance.
<i>rhs</i>	The right hand document data instance.

Returns

bool operator Return `true` if *lhs* is lexically less than or has an earlier modification time.

6.8.3.4 bool spt::servlet::search::operator== (const DocumentData & *lhs*, const DocumentData & *rhs*)

Compare two document data instances for equality.

Parameters

<i>lhs</i>	The left hand document data instance
<i>rhs</i>	The right hand document data instance.

Returns

bool operator Return `if` all fields are equal.

6.9 spt::servlet::ServletRegistry Namespace Reference

Namespace that provides functions for registering servlets. Servlets registered may be fetched from the registry by the fully qualified class name.

Typedefs

- typedef Poco::HashMap
`< std::string, ServletHolder * > ServletMap`
Map of servlet class name to servlet holder instance.

Functions

- void `writeTimestamp ()`
Utility method to output a formatted current timestamp to `std::cout`.
- `ServletMap & getRegisteredServlets ()`
Return the map used to registered servlets.
- void `registerServlet (const std::string &className, ServletHolder *servlet)`
Register a servlet with the specified fully qualified class name, and specified holder class that can be used to create new instances of the servlet.
- Poco::SharedPtr< Servlet > `getServlet (const std::string &className)`
Return a pointer to the servlet that was registered with the specified fully qualified class name.
- void `deRegisterServlet (const std::string &className)`
Use to clean up any resources held by the servlet. Normally used only during server shutdown, or reload process.

6.9.1 Detailed Description

Namespace that provides functions for registering servlets. Servlets registered may be fetched from the registry by the fully qualified class name.

Copyright 2013, Sans Pareil Technologies, Inc.

Id

[ServletRegistry.h](#) 5567 2013-02-23 22:13:21Z rakesh

6.9.2 Typedef Documentation

6.9.2.1 typedef Poco::HashMap<std::string,ServletHolder*> spt::servlet::ServletRegistry::ServletMap

Map of servlet class name to servlet holder instance.

6.9.3 Function Documentation

6.9.3.1 void spt::servlet::ServletRegistry::deRegisterServlet (const std::string & className) [inline]

Use to clean up any resources held by the servlet. Normally used only during server shutdown, or reload process.

Parameters

<i>className</i>	The fully qualified class name of the servlet.
------------------	--

6.9.3.2 ServletMap& spt::servlet::ServletRegistry::getRegisteredServlets () [inline]

Return the map used to registered servlets.

Returns

ServletMap The map for registering servlets.

6.9.3.3 `Poco::SharedPtr<Servlet> spt::servlet::ServletRegistry::getServlet ( const std::string & className ) [inline]`

Return a pointer to the servlet that was registered with the specified fully qualified class name.

Note that similar to JEE web containers, only a single instance for a servlet is maintained for a virtual host. [Servlet](#) instances are `stateless` and all work is handled via the re-entrant [Servlet::service](#) method.

Parameters

<i>className</i>	The fully qualified class name used to register the servlet
------------------	---

Returns

`Poco::SharedPtr<Servlet>` A shared pointer to the common servlet instance.

6.9.3.4 `void spt::servlet::ServletRegistry::registerServlet ( const std::string & className, ServletHolder * servlet ) [inline]`

Register a servlet with the specified fully qualified class name, and specified holder class that can be used to create new instances of the servlet.

Parameters

<i>className</i>	The fully qualified class name of the servlet
<i>servlet</i>	The holder used to create new instances of the servlet.

6.9.3.5 `void spt::servlet::ServletRegistry::writeTimestamp ( ) [inline]`

Utility method to output a formatted current timestamp to `std::cout`.

6.10 spt::servlet::templating Namespace Reference

Templating system for Minnal servlet container.

Classes

- class [CMSServlet](#)
The [Servlet](#) that builds dynamic output based on files with extension `cms` stored under the virtual host document root.
- class [Datastore](#)
Factory class for interacting with backend systems used to store/retrieve data used to fill templates.
- class [TemplateManager](#)
A manager class used to abstract interactions with the templating engine.

6.10.1 Detailed Description

Templating system for Minnal servlet container.

Namespace with the main classes used to implement the templating system for Minnal C++ [Servlet](#) Container. Minnal uses Google ctemplate as the underlying template engine. Minnal provides means for sharing templates across pages, loading templates and associated data from a variety of sources including filesystem, database ...

Chapter 7

Class Documentation

7.1 spt::servlet::model::AccessLogRecord Class Reference

Utility class that provides support for generating virtual host access logs in common log format.

```
#include <AccessLogRecord.h>
```

Static Public Member Functions

- static void [log](#) (const Poco::Net::HTTPServerRequest &req, const Poco::Net::HTTPServerResponse &resp, Poco::Logger &channel)
Log to the specified channel a client request.

7.1.1 Detailed Description

Utility class that provides support for generating virtual host access logs in common log format.

Date

Created 2011/09/06 18:41

Copyright

Copyright © 2011, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[AccessLogRecord.h](#) 5567 2013-02-23 22:13:21Z rakesh

7.1.2 Member Function Documentation

- 7.1.2.1 static void spt::servlet::model::AccessLogRecord::log (const Poco::Net::HTTPServerRequest & *req*, const Poco::Net::HTTPServerResponse & *resp*, Poco::Logger & *channel*) [static]

Log to the specified channel a client request.

Parameters

<i>req</i>	The client request to log
<i>resp</i>	The response object for the client request.
<i>channel</i>	The log channel to log the message to.

The documentation for this class was generated from the following file:

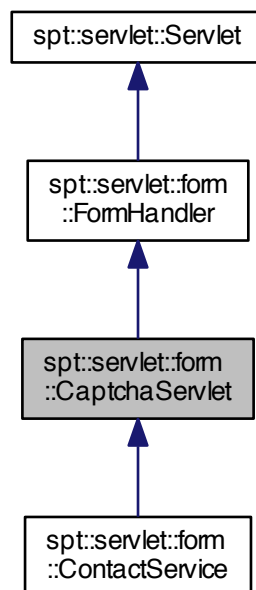
- /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/[AccessLogRecord.h](#)

7.2 spt::servlet::form::CaptchaServlet Class Reference

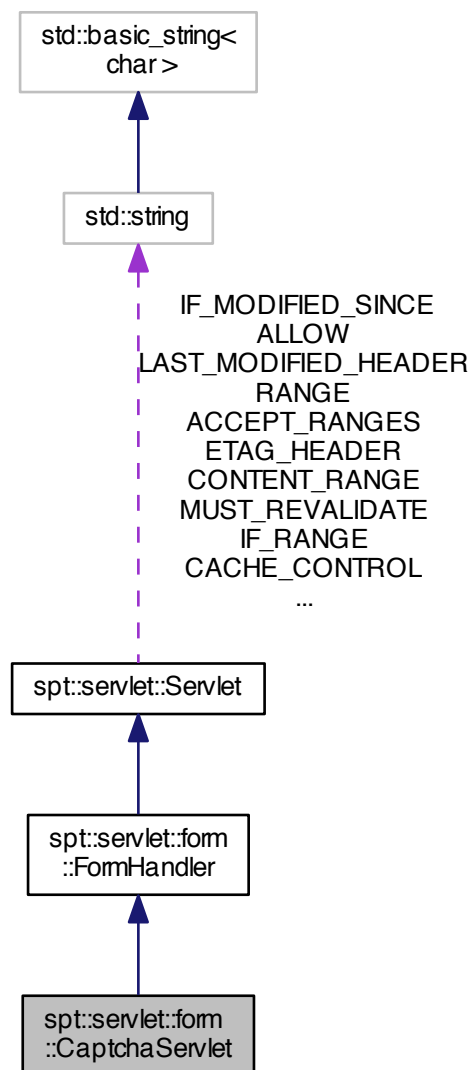
A base servlet class that adds support for captcha validation. Sub-classes should use the [validateCaptcha](#) method to validate the client captcha response in addition to any other business rules.

```
#include <CaptchaServlet.h>
```

Inheritance diagram for spt::servlet::form::CaptchaServlet:



Collaboration diagram for spt::servlet::form::CaptchaServlet:



Public Member Functions

- virtual `~CaptchaServlet()`
Virtual DTOR for sub-classes.

Protected Types

- typedef `std::pair< bool, std::string >` `CaptchaResponse`
A pair that represents the response from reCAPTCHA verification service.

Protected Member Functions

- [CaptchaResponse validateCaptcha](#) (const std::string &remoteHost, const Poco::Net::HTMLForm &form)
Validate the captcha information in the client request.
- const std::string [getRemoteHost](#) (const Poco::Net::HTTPServerRequest &req) const
Return the IP address for the remote host from which the request originated.
- const std::string & [getVerifyUrl](#) (const std::string &defaultValue=std::string()) const
Return the captcha verify URL configured for the host.
- const std::string & [getPublicKey](#) (const std::string &defaultValue=std::string()) const
Return the API public key for the domain (usually the primary domain) for the host.
- const std::string & [getPrivateKey](#) (const std::string &defaultValue=std::string()) const
Return the API private key for the domain (usually the primary domain) for the host.

Additional Inherited Members

7.2.1 Detailed Description

A base servlet class that adds support for captcha validation. Sub-classes should use the [validateCaptcha](#) method to validate the client captcha response in addition to any other business rules.

Captcha support is made available using reCAPTCHA. The API keys needed to set up captcha service support must be configured as context params for the virtual host. To keep all information configurable, the captcha verify URL etc. are also expected to be configured as context param values. The following context parameters must be configured per virtual host that needs to support captcha service

- captchaVerify - The reCAPTCHA verify URL.
- captchaPublicKey - The reCAPTCHA public key value.
- captchaPrivateKey - The reCAPTCHA private key value.

Since

Version 1.6

Date

Created 2013/02/23 19:05

Copyright

Copyright © 2013, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[CaptchaServlet.h](#) 5572 2013-02-25 17:00:43Z rakesh

7.2.2 Member Typedef Documentation

7.2.2.1 `typedef std::pair<bool,std::string> spt::servlet::form::CaptchaServlet::CaptchaResponse`
`[protected]`

A pair that represents the response from reCAPTCHA verification service.

7.2.3 Constructor & Destructor Documentation

7.2.3.1 `virtual spt::servlet::form::CaptchaServlet::~CaptchaServlet ( )` `[inline],[virtual]`

Virtual DTOR for sub-classes.

7.2.4 Member Function Documentation

7.2.4.1 `const std::string& spt::servlet::form::CaptchaServlet::getPrivateKey ( const std::string & defaultValue = std::string() ) const` `[inline],[protected]`

Return the API private key for the domain (usually the primary domain) for the host.

Parameters

<i>defaultValue</i>	The default value to return if not configured.
---------------------	--

Returns

`const std::string` The configured private key value

7.2.4.2 `const std::string& spt::servlet::form::CaptchaServlet::getPublicKey ( const std::string & defaultValue = std::string() ) const` `[inline],[protected]`

Return the API public key for the domain (usually the primary domain) for the host.

Parameters

<i>defaultValue</i>	The default value to return if not configured.
---------------------	--

Returns

`const std::string` The configured public key value

7.2.4.3 `const std::string spt::servlet::form::CaptchaServlet::getRemoteHost ( const Poco::Net::HTTPServerRequest & req ) const` `[inline],[protected]`

Return the IP address for the remote host from which the request originated.

Parameters

<i>req</i>	The client request object.
------------	----------------------------

Returns

`const std::string` The client IP address

7.2.4.4 `const std::string& spt::servlet::form::CaptchaServlet::getVerifyUrl ( const std::string & defaultValue = std::string() )const` [inline],[protected]

Return the captcha verify URL configured for the host.

Parameters

<i>defaultValue</i>	The default value to return if not configured.
---------------------	--

Returns

const std::string The configured verification URL

7.2.4.5 **CaptchaResponse** spt::servlet::form::CaptchaServlet::validateCaptcha (const std::string & *remoteHost*, const Poco::Net::HTMLForm & *form*) [protected]

Validate the captcha information in the client request.

Attempt to validate the captcha information in the request. If the optional message parameter is specified, set the response text as its value.

See also

[getRemoteHost](#)

Parameters

<i>remoteHost</i>	The IP address of the remote host making the request.
<i>form</i>	The parsed HTML form from the client request.
<i>message</i>	The optional string that if specified will be populated with the response text from the service.

Returns

CaptchaResponse Return the response from the verification service

The documentation for this class was generated from the following file:

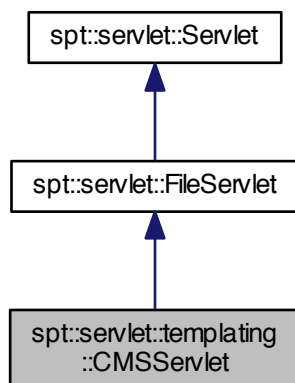
- /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/form/[CaptchaServlet.h](#)

7.3 spt::servlet::templating::CMSServlet Class Reference

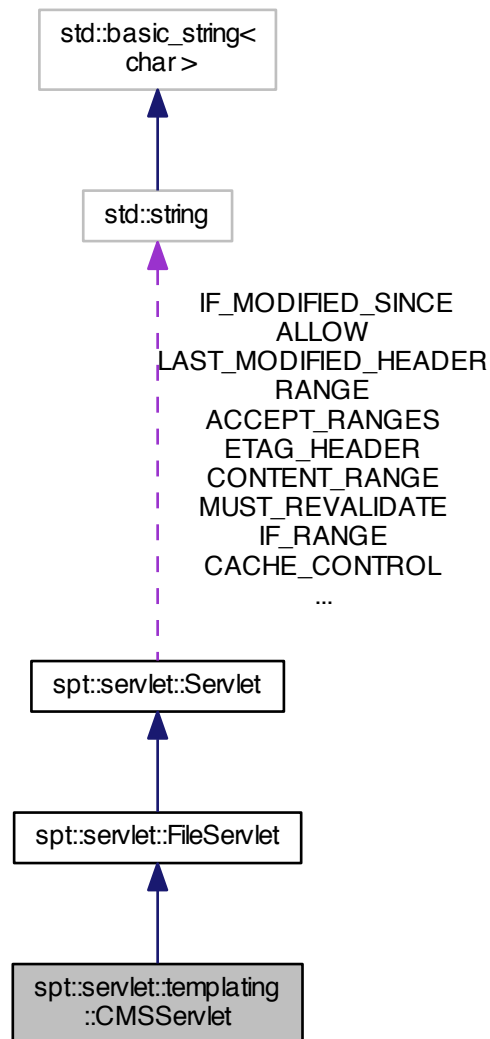
The [Servlet](#) that builds dynamic output based on files with extension `cms` stored under the virtual host document root.

```
#include <CMSServlet.h>
```

Inheritance diagram for spt::servlet::templating::CMSServlet:



Collaboration diagram for spt::servlet::templating::CMSServlet:



Public Member Functions

- [CMSServlet](#) ()
Default CTOR.
- virtual [~CMSServlet](#) ()
Virtual DTOR for sub-classes.
- const std::string [getServletName](#) () const
Return a descriptive name for this servlet (useful for logging).
- void [init](#) ()
Initialise the servlet instance. Invoked from [VirtualHostServer](#).

Protected Member Functions

- Poco::Logger & [getLogger](#) ()

Additional Inherited Members

7.3.1 Detailed Description

The [Servlet](#) that builds dynamic output based on files with extension `cms` stored under the virtual host document root.

These files are JSON files that specify the location of template to use to render the dynamic content, as well as the type of datastore to retrieve the associated data from.

Data (also represented as JSON) will be loaded from the appropriate datastore and used to build the ctemplate data dictionary. Data will need to be stored with a path that is the same as the path to the requested `cms` file.

Date

Created 2014/10/03 16:41

Copyright

Copyright © 2014, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[CMSServlet.h](#) 5632 2014-10-06 22:04:21Z rakesh

Since

1.7

7.3.2 Constructor & Destructor Documentation

7.3.2.1 `spt::servlet::templating::CMSServlet::CMSServlet ( )` `[inline]`

Default CTOR.

7.3.2.2 `virtual spt::servlet::templating::CMSServlet::~~CMSServlet ( )` `[inline]`, `[virtual]`

Virtual DTOR for sub-classes.

7.3.3 Member Function Documentation

7.3.3.1 `Poco::Logger& spt::servlet::templating::CMSServlet::getLogger ( )` `[inline]`, `[protected]`, `[virtual]`

Returns

Poco::Logger Return the logger used by this servlet.

Reimplemented from [spt::servlet::Servlet](#).

7.3.3.2 `const std::string spt::servlet::templating::CMSServlet::getServletName ( ) const` `[inline], [virtual]`

Return a descriptive name for this servlet (useful for logging).

Returns

`const std::string` The name for this servlet.

Reimplemented from [spt::servlet::Servlet](#).

7.3.3.3 `void spt::servlet::templating::CMSServlet::init ( )` `[virtual]`

Initialise the servlet instance. Invoked from [VirtualHostServer](#).

Reimplemented from [spt::servlet::Servlet](#).

The documentation for this class was generated from the following file:

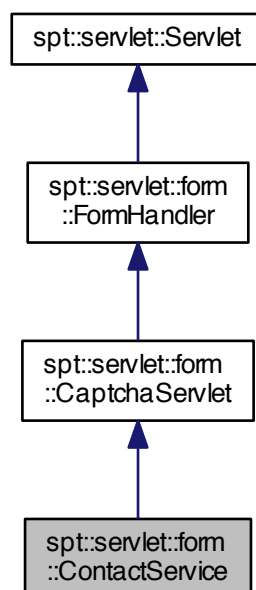
- `/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/templating/CMSServlet.h`

7.4 spt::servlet::form::ContactService Class Reference

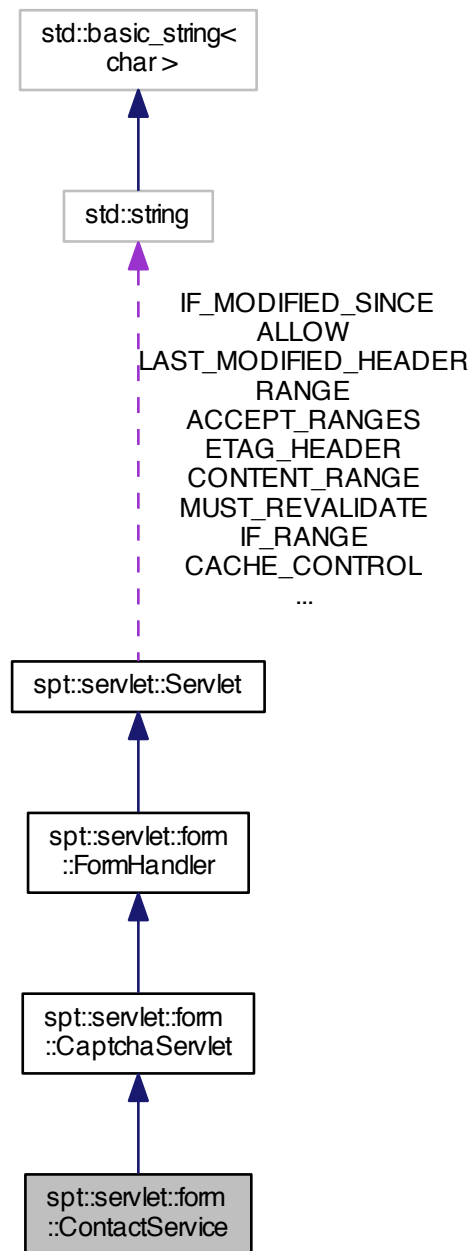
A handler for contact forms.

```
#include <ContactService.h>
```

Inheritance diagram for `spt::servlet::form::ContactService`:



Collaboration diagram for `spt::servlet::form::ContactService`:



Public Member Functions

- `const std::string getServletName () const`

Protected Member Functions

- void [handleForm](#) (const Poco::Net::HTMLForm &form, Poco::Net::HTTPServerRequest &req, Poco::Net::HTTPServerResponse &resp)
Handle the form post request from the client.

Additional Inherited Members

7.4.1 Detailed Description

A handler for contact forms.

Details of the SMTP server and port, recipient e-mail address, SMTP authentication credentials (if required) etc. must be configured as either context or init param values. The following params are looked up while attempting to send the email with the information in the contact form:

- `smtpServer` - The SMTP server address
- `smtpPort` - The port to use to connect (eg. 25, 587, ...)
- `contactRecipient` - The e-mail address of the recipient of the contact form requests.
- `smtpUser` - The user name to use to authenticate with SMTP server (optional).
- `smtpPassword` - The password to use to authenticate with SMTP server (optional). Must be specified if `smtpUser` is specified.

Since

Version 1.6

Date

Created 2013/02/24 11:20

Copyright

Copyright © 2013, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[ContactService.h](#) 5578 2013-02-26 02:31:52Z rakesh

7.4.2 Member Function Documentation

7.4.2.1 `const std::string spt::servlet::form::ContactService::getServletName ( ) const` `[inline], [virtual]`

Returns

`std::string` Return a descriptive name of servlet (mainly for logging)

Reimplemented from [spt::servlet::Servlet](#).

7.4.2.2 void spt::servlet::form::ContactService::handleForm (const Poco::Net::HTMLForm & *form*,
Poco::Net::HTTPServerRequest & *req*, Poco::Net::HTTPServerResponse & *resp*) [protected],[virtual]

Handle the form post request from the client.

Parameters

<i>form</i>	The form that encapsulates the data submitted by the client.
<i>req</i>	The client request object
<i>resp</i>	The server response object.

Implements [spt::servlet::form::FormHandler](#).

The documentation for this class was generated from the following file:

- /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/form/[ContactService.h](#)

7.5 spt::servlet::model::ContainerConfiguration Class Reference

Value object that stores the various configuration objects used to manage the servlet container.

```
#include <ContainerConfiguration.h>
```

Public Member Functions

- [ContainerConfiguration](#) ([ServerConfigurationPtr](#) sptr, const [MimeTypes](#) &mt)
Create a new instance using the specified server configuration object and mime types map.
- const [ServerConfigurationPtr](#) [getServerConfiguration](#) () const
Return the pointer to the server configuration used to initialise this container wide configuration object.
- const [MimeTypes](#) & [getMimeTypes](#) () const
Return the mime type mapping file contents.
- const std::string [getContentType](#) (const std::string &path) const
Return the mime-type for the file at the specified path based on the mime-mapping used to initialise this instance.
- const std::string [getContentType](#) (const Poco::File &file) const
Return the mime-type for the file at the specified path based on the mime-mapping used to initialise this instance.

7.5.1 Detailed Description

Value object that stores the various configuration objects used to manage the servlet container.

Date

Created 2011/09/06 18:21

Copyright

Copyright © 2011, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[ContainerConfiguration.h](#) 5625 2014-09-29 03:14:49Z rakesh

7.5.2 Constructor & Destructor Documentation

7.5.2.1 `spt::servlet::model::ContainerConfiguration::ContainerConfiguration ( ServerConfigurationPtr sptr, const MimeTypes & mt )` `[inline]`

Create a new instance using the specified server configuration object and mime types map.

Parameters

<i>sptr</i>	The server configuration instance.
<i>mt</i>	The mime type mapping file.

7.5.3 Member Function Documentation

7.5.3.1 `const std::string spt::servlet::model::ContainerConfiguration::getContentType ( const std::string & path ) const`
`[inline]`

Return the mime-type for the file at the specified path based on the mime-mapping used to initialise this instance.

Parameters

<i>path</i>	The path to the file.
-------------	-----------------------

Returns

`const std::string` The mime type for the file. If no mapping if found return `application/octet-stream`.

7.5.3.2 `const std::string spt::servlet::model::ContainerConfiguration::getContentType ( const Poco::File & file ) const`
`[inline]`

Return the mime-type for the file at the specified path based on the mime-mapping used to initialise this instance.

Parameters

<i>file</i>	The file for which mime-type is to be retrieved.
-------------	--

Returns

`const std::string` The mime type for the file. If no mapping if found return `application/octet-stream`.

7.5.3.3 `const MimeTypes& spt::servlet::model::ContainerConfiguration::getMimeTypes ( ) const` `[inline]`

Return the mime type mapping file contents.

Returns

`const MimeTypes` The mime mapping

7.5.3.4 `const ServerConfigurationPtr spt::servlet::model::ContainerConfiguration::getServerConfiguration ( ) const`
`[inline]`

Return the pointer to the server configuration used to initialise this container wide configuration object.

Returns

`const ServerConfigurationPtr` The server configuration

The documentation for this class was generated from the following file:

- `/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/ContainerConfiguration.h`

7.6 spt::servlet::templating::Datastore Class Reference

Factory class for interacting with backend systems used to store/retrieve data used to fill templates.

```
#include <Datastore.h>
```

Public Types

- typedef [Poco::Util::JSONObject::Ptr](#) [Configuration](#)
Configuration used to initialise a datastore implementation.
- typedef Poco::SharedPtr
 < [Datastore](#) > [Ptr](#)
Shared pointer to a datastore implementation.
- typedef Poco::SharedPtr
 < ctemplate::TemplateDictionary > [Dictionary](#)
Shared pointer to a template dictionary.

Public Member Functions

- virtual [Dictionary](#) [load](#) (const std::string &path)=0
Load a dictionary from data stored at the specified path.
- virtual Poco::Timestamp [getLastModified](#) (const std::string &path)=0
Return the last modification timestamp value for the data at the specified virtual path in the datastore implementation.

Static Public Member Functions

- static [Ptr](#) [getDatastore](#) (const std::string &type, [Configuration](#) configuration)
Initialise a datastore of the specified type using the specified configuration.

Protected Member Functions

- [Datastore](#) ()

7.6.1 Detailed Description

Factory class for interacting with backend systems used to store/retrieve data used to fill templates.

Date

Created 2014/10/04 21:38

Copyright

Copyright © 2014, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[Datastore.h](#) 5632 2014-10-06 22:04:21Z rakesh

Since

1.7

7.6.2 Member Typedef Documentation

7.6.2.1 typedef Poco::Util::JSONObject::Ptr spt::servlet::templating::Datastore::Configuration

Configuration used to initialise a datastore implementation.

7.6.2.2 typedef Poco::SharedPtr<template::TemplateDictionary> spt::servlet::templating::Datastore::Dictionary

Shared pointer to a template dictionary.

7.6.2.3 typedef Poco::SharedPtr<Datastore> spt::servlet::templating::Datastore::Ptr

Shared pointer to a datastore implementation.

7.6.3 Constructor & Destructor Documentation

7.6.3.1 spt::servlet::templating::Datastore::Datastore () [inline], [protected]

7.6.4 Member Function Documentation

7.6.4.1 static Ptr spt::servlet::templating::Datastore::getDatastore (const std::string & type, Configuration configuration) [static]

Initialise a datastore of the specified type using the specified configuration.

Configuration may include location on filesystem, datastore connection etc.

Parameters

<i>type</i>	The type of datastore to initialise (file, xapian, mongodb, leveldb, hamsterdb ...)
<i>configuration</i>	Configuration necessary to properly initialise the datastore

7.6.4.2 virtual Poco::Timestamp spt::servlet::templating::Datastore::getLastModified (const std::string & path) [pure virtual]

Return the last modification timestamp value for the data at the specified virtual path in the datastore implementation.

Parameters

<i>path</i>	The virtual path representing the data in the datastore.
-------------	--

Returns

The last modification timestamp for the data at virtual path.

7.6.4.3 virtual Dictionary spt::servlet::templating::Datastore::load (const std::string & path) [pure virtual]

Load a dictionary from data stored at the specified path.

Note: A path is a virtual construct which may indicate the location on filesystem, a key/property used to retrieve from a datastore such as leveldb, mongodb etc.

Parameters

<i>path</i>	The abstract path representing the data to be loaded.
-------------	---

Returns

The dictionary instance with contents of the data at the specified path

The documentation for this class was generated from the following file:

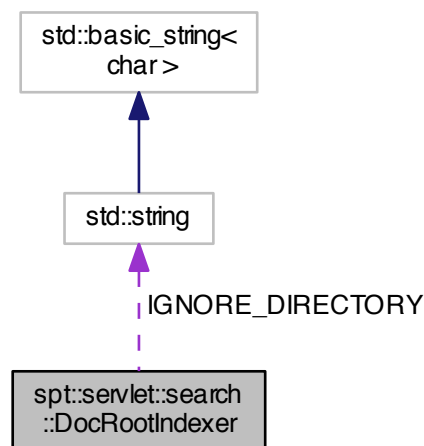
- /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/templating/Datastore.h

7.7 spt::servlet::search::DocRootIndexer Class Reference

A background indexer used to index files under the specified document root directory.

```
#include <DocRootIndexer.h>
```

Collaboration diagram for spt::servlet::search::DocRootIndexer:



Public Member Functions

- **DocRootIndexer** (const std::string &path, const [SearchIndexManagerPtr](#) searchManager, const [spt::servlet::model::ContainerConfigurationPtr](#) containerConfiguration)
Create a new indexer for the specified path.
- **~DocRootIndexer** ()
Stops the background indexer if running.
- void **start** ()

Start indexing files. Uses a Poco::Activity to run process in background.

- void [stop](#) ()

Stop the indexing process.

- bool [isRunning](#) ()

Static Public Attributes

- static std::string [IGNORE_DIRECTORY](#)

The name of the (empty) file to be used to make directory tree ignored by indexer.

7.7.1 Detailed Description

A background indexer used to index files under the specified document root directory.

Starts at the document root directory for the virtual host and indexes files that are supported by the search engine. There is no support for `robots.txt`, but directory trees may be marked to be ignored by the indexer by placing a (empty) file named `_DO_NOT_INDEX` ([IGNORE_DIRECTORY](#)) under the parent of the tree to ignore.

Date

Created 2013/01/10 18:41

Copyright

Copyright © 2013, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[DocRootIndexer.h](#) 5588 2013-03-02 02:50:42Z rakesh

7.7.2 Constructor & Destructor Documentation

- 7.7.2.1** `spt::servlet::search::DocRootIndexer::DocRootIndexer ( const std::string & path, const SearchIndexManagerPtr searchManager, const spt::servlet::model::ContainerConfigurationPtr containerConfiguration )`

Create a new indexer for the specified path.

Parameters

<i>path</i>	The base path from which files are to be indexed.
<i>searchManager</i>	The search index manager instance to use.
<i>containerConfiguration</i>	The servlet container configuration.

- 7.7.2.2** `spt::servlet::search::DocRootIndexer::~DocRootIndexer ( )`

Stops the background indexer if running.

7.7.3 Member Function Documentation

7.7.3.1 `bool spt::servlet::search::DocRootIndexer::isRunning ( )` `[inline]`

Returns

`bool` Return a flag indicating whether the indexing operation is running.

7.7.3.2 `void spt::servlet::search::DocRootIndexer::start ( )`

Start indexing files. Uses a `Poco::Activity` to run process in background.

7.7.3.3 `void spt::servlet::search::DocRootIndexer::stop ( )`

Stop the indexing process.

7.7.4 Member Data Documentation

7.7.4.1 `std::string spt::servlet::search::DocRootIndexer::IGNORE_DIRECTORY` `[static]`

The name of the (empty) file to be used to make directory tree ignored by indexer.

The documentation for this class was generated from the following file:

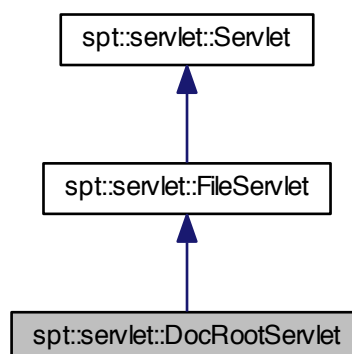
- `/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/DocRootIndexer.h`

7.8 spt::servlet::DocRootServlet Class Reference

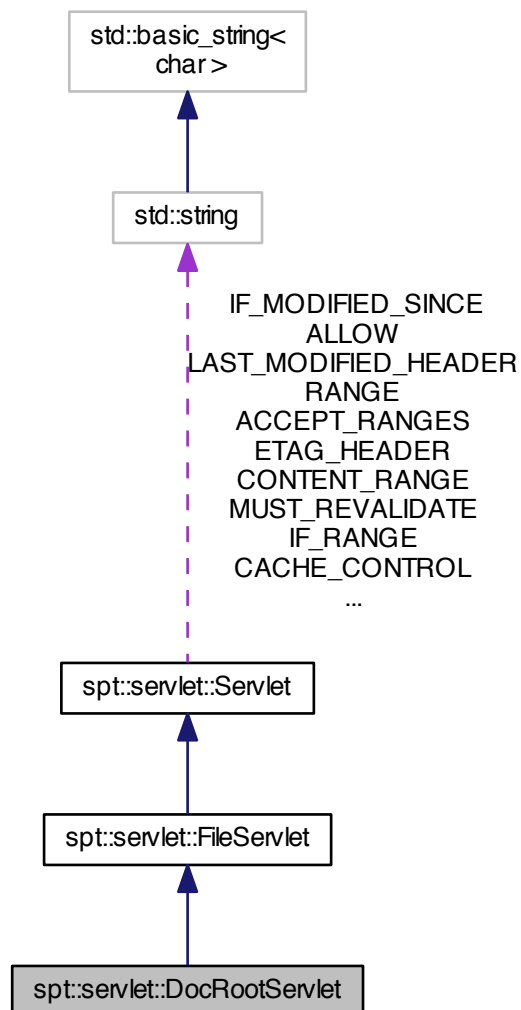
Default servlet for serving static files.

```
#include <DocRootServlet.h>
```

Inheritance diagram for `spt::servlet::DocRootServlet`:



Collaboration diagram for spt::servlet::DocRootServlet:



Public Member Functions

- [DocRootServlet](#) ()
Default CTOR.
- virtual [~DocRootServlet](#) ()
Virtual DTOR for sub-classes.
- const std::string [getServletName](#) () const
- void [init](#) ()
Initialise the servlet instance. Invoked from [VirtualHostServer](#).

Protected Member Functions

- Poco::Logger & [getLogger](#) ()

Additional Inherited Members

7.8.1 Detailed Description

Default servlet for serving static files.

The default servlet that serves static files from a configured document root directory. Note that for default handling logic to work, this servlet must be named docroot in web.xml.

Date

Created 2011/08/20 18:41

Copyright

Copyright © 2011, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[DocRootServlet.h](#) 5632 2014-10-06 22:04:21Z rakesh

7.8.2 Constructor & Destructor Documentation

7.8.2.1 `spt::servlet::DocRootServlet( )` `[inline]`

Default CTOR.

7.8.2.2 `virtual spt::servlet::DocRootServlet::~~DocRootServlet( )` `[inline]`, `[virtual]`

Virtual DTOR for sub-classes.

7.8.3 Member Function Documentation

7.8.3.1 `Poco::Logger& spt::servlet::DocRootServlet::getLogger( )` `[inline]`, `[protected]`, `[virtual]`

Returns

Poco::Logger Return the logger used by this servlet.

Reimplemented from [spt::servlet::Servlet](#).

7.8.3.2 `const std::string spt::servlet::DocRootServlet::getServletName( )` `const` `[inline]`, `[virtual]`

Returns

std::string Return a descriptive name of servlet (mainly for logging)

Reimplemented from [spt::servlet::Servlet](#).

7.8.3.3 void spt::servlet::DocRootServlet::init () [virtual]

Initialise the servlet instance. Invoked from [VirtualHostServer](#).

Reimplemented from [spt::servlet::Servlet](#).

The documentation for this class was generated from the following file:

- [/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/DocRootServlet.h](#)

7.9 spt::servlet::search::DocumentData Class Reference

A simple POD used to store data relevant to a document indexed in the Xapian search engine.

```
#include <DocumentData.h>
```

Public Member Functions

- [DocumentData](#) ()
Default CTOR.
- [DocumentData](#) (const std::string &buffer)
Create a new instance from the serialised data specified.
- [DocumentData](#) (const Poco::File &file, const std::string &urlPath, const std::string &contentType, const std::string &title, const std::string &content)
Create a new instance from the specified values.
- const std::string & [getFile](#) () const
- Poco::Timestamp::TimeVal [getLastModified](#) () const
- const std::string & [getUriPath](#) () const
- const std::string & [getContentType](#) () const
- const std::string & [getTitle](#) () const
- const std::string & [getContent](#) () const
const std::string Return the content associated with the file.
- const std::string [serialise](#) () const
- void [deserialise](#) (const std::string &buffer)
De-serialise the contents of the specified string into the fields of this instance.

7.9.1 Detailed Description

A simple POD used to store data relevant to a document indexed in the Xapian search engine.

Date

Created 2013/01/10 15:41

Copyright

Copyright © 2013, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[DocumentData.h](#) 5588 2013-03-02 02:50:42Z rakesh

7.9.2 Constructor & Destructor Documentation

7.9.2.1 `spt::servlet::search::DocumentData::DocumentData ( )` `[inline]`

Default CTOR.

7.9.2.2 `spt::servlet::search::DocumentData::DocumentData ( const std::string & buffer )` `[explicit]`

Create a new instance from the serialised data specified.

Parameters

<i>buffer</i>	The serialised version of the data to create the new instance from.
---------------	---

7.9.2.3 `spt::servlet::search::DocumentData::DocumentData ( const Poco::File & file, const std::string & urlPath, const std::string & contentType, const std::string & title, const std::string & content )`

Create a new instance from the specified values.

Parameters

<i>file</i>	The file (resource) that was indexed
<i>urlPath</i>	The relative URI for the file
<i>contentType</i>	The mime-type for the file
<i>title</i>	The title for the resource (optional)
<i>content</i>	The content that was indexed for the file.

7.9.3 Member Function Documentation

7.9.3.1 `void spt::servlet::search::DocumentData::deserialise ( const std::string & buffer )`

De-serialise the contents of the specified string into the fields of this instance.

Parameters

<i>buffer</i>	The serialised version to de-serialise.
---------------	---

7.9.3.2 `const std::string& spt::servlet::search::DocumentData::getContent ( ) const` `[inline]`

`const std::string` Return the content associated with the file.

7.9.3.3 `const std::string& spt::servlet::search::DocumentData::getContentType ( ) const` `[inline]`

Returns

`const std::string` Return the mime-type for the file

7.9.3.4 `const std::string& spt::servlet::search::DocumentData::getFile ( ) const` `[inline]`

Returns

`const std::string` Return the path to the file that was indexed.

7.9.3.5 Poco::Timestamp::TimeVal spt::servlet::search::DocumentData::getLastModified () const [inline]

Returns

Poco::Timestamp::TimeVal Return the last modified time for the file.

7.9.3.6 const std::string& spt::servlet::search::DocumentData::getTitle () const [inline]

Returns

std::string Return the title for the file.

7.9.3.7 const std::string& spt::servlet::search::DocumentData::getUriPath () const [inline]

Returns

const std::string Return the relative URI for the file

7.9.3.8 const std::string spt::servlet::search::DocumentData::serialise () const

Returns

const std::string Return the serialised version of this object.

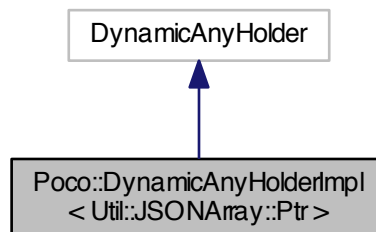
The documentation for this class was generated from the following file:

- /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/[DocumentData.h](#)

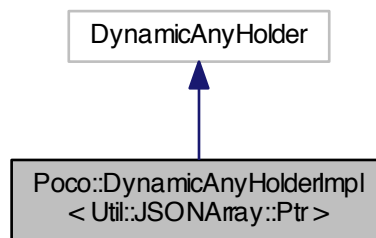
7.10 Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr > Class Template Reference

```
#include <JSONArray.h>
```

Inheritance diagram for Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >:



Collaboration diagram for Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >:



Public Member Functions

- [DynamicAnyHolderImpl](#) (const [Util::JSONArray::Ptr](#) &val)
- [~DynamicAnyHolderImpl](#) ()
- const std::type_info & [type](#) () const
- void [convert](#) (Int8 &) const
- void [convert](#) (Int16 &) const
- void [convert](#) (Int32 &) const
- void [convert](#) (Int64 &) const
- void [convert](#) (UInt8 &) const
- void [convert](#) (UInt16 &) const
- void [convert](#) (UInt32 &) const
- void [convert](#) (UInt64 &) const
- void [convert](#) (bool &value) const
- void [convert](#) (float &) const
- void [convert](#) (double &) const
- void [convert](#) (char &) const
- void [convert](#) (std::string &s) const
- void [convert](#) (DateTime &val) const
- void [convert](#) (LocalDateTime &ldt) const
- void [convert](#) (Timestamp &ts) const
- DynamicAnyHolder * [clone](#) () const
- const [Util::JSONArray::Ptr](#) & [value](#) () const
- bool [isArray](#) () const
- bool [isInteger](#) () const
- bool [isSigned](#) () const
- bool [isNumeric](#) () const
- bool [isString](#) () const

7.10.1 Constructor & Destructor Documentation

7.10.1.1 `Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >::DynamicAnyHolderImpl ( const Util::JSONArray::Ptr & val ) [inline]`

7.10.1.2 `Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >::~~DynamicAnyHolderImpl ( ) [inline]`

7.10.2 Member Function Documentation

- 7.10.2.1 `DynamicAnyHolder* Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >::clone ( ) const` [inline]
- 7.10.2.2 `void Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >::convert ( Int8 & ) const` [inline]
- 7.10.2.3 `void Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >::convert ( Int16 & ) const` [inline]
- 7.10.2.4 `void Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >::convert ( Int32 & ) const` [inline]
- 7.10.2.5 `void Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >::convert ( Int64 & ) const` [inline]
- 7.10.2.6 `void Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >::convert ( UInt8 & ) const` [inline]
- 7.10.2.7 `void Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >::convert ( UInt16 & ) const` [inline]
- 7.10.2.8 `void Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >::convert ( UInt32 & ) const` [inline]
- 7.10.2.9 `void Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >::convert ( UInt64 & ) const` [inline]
- 7.10.2.10 `void Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >::convert ( bool & value ) const` [inline]
- 7.10.2.11 `void Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >::convert ( float & ) const` [inline]
- 7.10.2.12 `void Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >::convert ( double & ) const` [inline]
- 7.10.2.13 `void Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >::convert ( char & ) const` [inline]
- 7.10.2.14 `void Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >::convert ( std::string & s ) const` [inline]
- 7.10.2.15 `void Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >::convert ( DateTime & val ) const` [inline]
- 7.10.2.16 `void Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >::convert ( LocalDateTime & ldt ) const`
[inline]
- 7.10.2.17 `void Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >::convert ( Timestamp & ts ) const` [inline]
- 7.10.2.18 `bool Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >::isArray ( ) const` [inline]
- 7.10.2.19 `bool Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >::isInteger ( ) const` [inline]
- 7.10.2.20 `bool Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >::isNumeric ( ) const` [inline]
- 7.10.2.21 `bool Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >::isSigned ( ) const` [inline]
- 7.10.2.22 `bool Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >::isString ( ) const` [inline]
- 7.10.2.23 `const std::type_info& Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >::type ( ) const` [inline]
- 7.10.2.24 `const Util::JSONArray::Ptr& Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >::value ( ) const`
[inline]

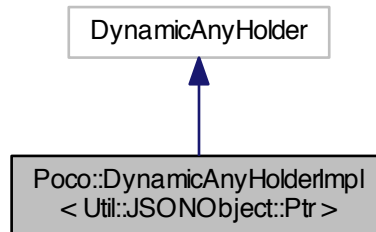
The documentation for this class was generated from the following file:

- [/Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONArray.h](#)

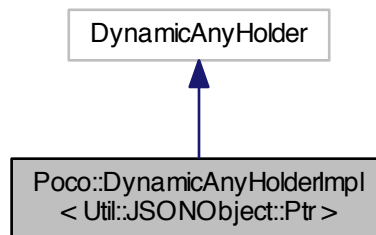
7.11 Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr > Class Template Reference

```
#include <JSONObject.h>
```

Inheritance diagram for Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >:



Collaboration diagram for Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >:



Public Member Functions

- [DynamicAnyHolderImpl](#) (const [Util::JSONObject::Ptr](#) &val)
- [~DynamicAnyHolderImpl](#) ()
- const std::type_info & [type](#) () const
- void [convert](#) (Int8 &) const
- void [convert](#) (Int16 &) const
- void [convert](#) (Int32 &) const
- void [convert](#) (Int64 &) const
- void [convert](#) (UInt8 &) const
- void [convert](#) (UInt16 &) const
- void [convert](#) (UInt32 &) const
- void [convert](#) (UInt64 &) const
- void [convert](#) (bool &[value](#)) const
- void [convert](#) (float &) const
- void [convert](#) (double &) const

- void [convert](#) (char &) const
- void [convert](#) (std::string &s) const
- void [convert](#) (DateTime &val) const
- void [convert](#) (LocalDateTime &ldt) const
- void [convert](#) (Timestamp &ts) const
- DynamicAnyHolder * [clone](#) () const
- const Util::JSONObject::Ptr & [value](#) () const
- bool [isArray](#) () const
- bool [isInteger](#) () const
- bool [isSigned](#) () const
- bool [isNumeric](#) () const
- bool [isString](#) () const

7.11.1 Constructor & Destructor Documentation

7.11.1.1 Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >::DynamicAnyHolderImpl (const Util::JSONObject::Ptr & val) [\[inline\]](#)

7.11.1.2 Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >::~~DynamicAnyHolderImpl () [\[inline\]](#)

7.11.2 Member Function Documentation

7.11.2.1 DynamicAnyHolder* Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >::clone () const [\[inline\]](#)

7.11.2.2 void Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >::convert (Int8 &) const [\[inline\]](#)

7.11.2.3 void Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >::convert (Int16 &) const [\[inline\]](#)

7.11.2.4 void Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >::convert (Int32 &) const [\[inline\]](#)

7.11.2.5 void Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >::convert (Int64 &) const [\[inline\]](#)

7.11.2.6 void Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >::convert (UInt8 &) const [\[inline\]](#)

7.11.2.7 void Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >::convert (UInt16 &) const [\[inline\]](#)

7.11.2.8 void Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >::convert (UInt32 &) const [\[inline\]](#)

7.11.2.9 void Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >::convert (UInt64 &) const [\[inline\]](#)

7.11.2.10 void Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >::convert (bool & value) const [\[inline\]](#)

7.11.2.11 void Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >::convert (float &) const [\[inline\]](#)

7.11.2.12 void Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >::convert (double &) const [\[inline\]](#)

7.11.2.13 void Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >::convert (char &) const [\[inline\]](#)

7.11.2.14 void Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >::convert (std::string & s) const [\[inline\]](#)

7.11.2.15 void Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >::convert (DateTime & val) const [\[inline\]](#)

7.11.2.16 void Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >::convert (LocalDateTime & ldt) const [\[inline\]](#)

```

7.11.2.17 void Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >::convert ( Timestamp & ts ) const [inline]

7.11.2.18 bool Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >::isArray ( ) const [inline]

7.11.2.19 bool Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >::isInteger ( ) const [inline]

7.11.2.20 bool Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >::isNumeric ( ) const [inline]

7.11.2.21 bool Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >::isSigned ( ) const [inline]

7.11.2.22 bool Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >::isString ( ) const [inline]

7.11.2.23 const std::type_info& Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >::type ( ) const [inline]

7.11.2.24 const Util::JSONObject::Ptr& Poco::DynamicAnyHolderImpl< Util::JSONObject::Ptr >::value ( ) const
[inline]

```

The documentation for this class was generated from the following file:

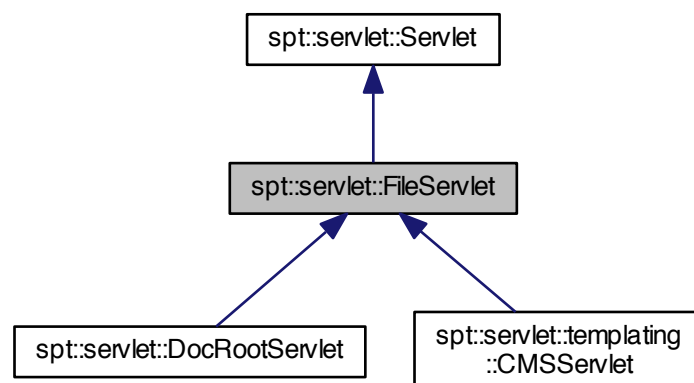
- </Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONObject.h>

7.12 spt::servlet::FileServlet Class Reference

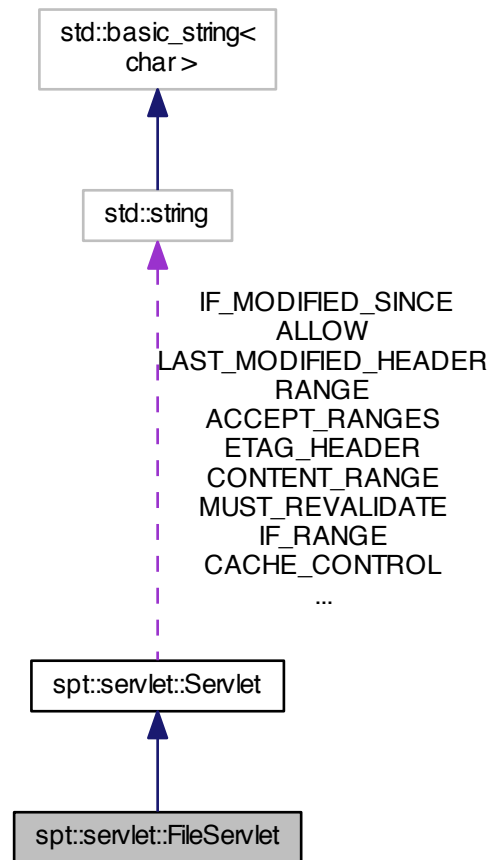
Abstract base servlet for serving files.

```
#include <FileServlet.h>
```

Inheritance diagram for spt::servlet::FileServlet:



Collaboration diagram for spt::servlet::FileServlet:



Protected Member Functions

- void `doGet` (Poco::Net::HTTPServerRequest &req, Poco::Net::HTTPServerResponse &resp)
Handle a GET HTTP request from a client.
- void `doHead` (Poco::Net::HTTPServerRequest &req, Poco::Net::HTTPServerResponse &resp)
Handle a HEAD HTTP request from a client.
- void `doOptions` (Poco::Net::HTTPServerRequest &req, Poco::Net::HTTPServerResponse &resp)
Handle a OPTIONS HTTP request from a client.
- bool `checkURI` (Poco::Net::HTTPServerRequest &req, Poco::Net::HTTPServerResponse &resp)
Check the URI specified in client request to ensure it is valid. Validity checks try to protect against common attacks attempted by malicious scripts and hackers.
- std::string `checksum` (const Poco::File &filePath)
Compute a checksum for the specified file. May be used for etags or similar purposes.

Additional Inherited Members

7.12.1 Detailed Description

Abstract base servlet for serving files.

Implementations may use common logic provided through this base class to serve static files or dynamically generated content that is written to a local file and then served statically.

Date

Created 2014/10/04 05:47

Copyright

Copyright © 2014, Sans Pareil Technologies, Inc.

Author

Rakesh

Since

1.7

Version

Id

[FileServlet.h](#) 5632 2014-10-06 22:04:21Z rakesh

7.12.2 Member Function Documentation

7.12.2.1 `std::string spt::servlet::FileServlet::checksum ( const Poco::File & filePath )` `[protected]`

Compute a checksum for the specified file. May be used for etags or similar purposes.

Parameters

<i>filePath</i>	The path to the file for which a checksum is to be computed
-----------------	---

Returns

The checksum value.

7.12.2.2 `bool spt::servlet::FileServlet::checkURI ( Poco::Net::HTTPServerRequest & req, Poco::Net::HTTPServerResponse & resp )` `[protected]`, `[virtual]`

Check the URI specified in client request to ensure it is valid. Validity checks try to protect against common attacks attempted by malicious scripts and hackers.

Parameters

<i>req</i>	The client request object to check
<i>resp</i>	The server response object

Returns

bool Return `true` if the request URI is clean.

Reimplemented from [spt::servlet::Servlet](#).

7.12.2.3 `void spt::servlet::FileServlet::doGet ( Poco::Net::HTTPServerRequest & req, Poco::Net::HTTPServerResponse & resp )` [protected],[virtual]

Handle a `GET` HTTP request from a client.

Parameters

<i>req</i>	The client request object
<i>resp</i>	The server response object

Reimplemented from [spt::servlet::Servlet](#).

7.12.2.4 `void spt::servlet::FileServlet::doHead ( Poco::Net::HTTPServerRequest & req, Poco::Net::HTTPServerResponse & resp )` [protected],[virtual]

Handle a `HEAD` HTTP request from a client.

Parameters

<i>req</i>	The client request object
<i>resp</i>	The server response object

Reimplemented from [spt::servlet::Servlet](#).

7.12.2.5 `void spt::servlet::FileServlet::doOptions ( Poco::Net::HTTPServerRequest & req, Poco::Net::HTTPServerResponse & resp )` [protected],[virtual]

Handle a `OPTIONS` HTTP request from a client.

Parameters

<i>req</i>	The client request object
<i>resp</i>	The server response object

Reimplemented from [spt::servlet::Servlet](#).

The documentation for this class was generated from the following file:

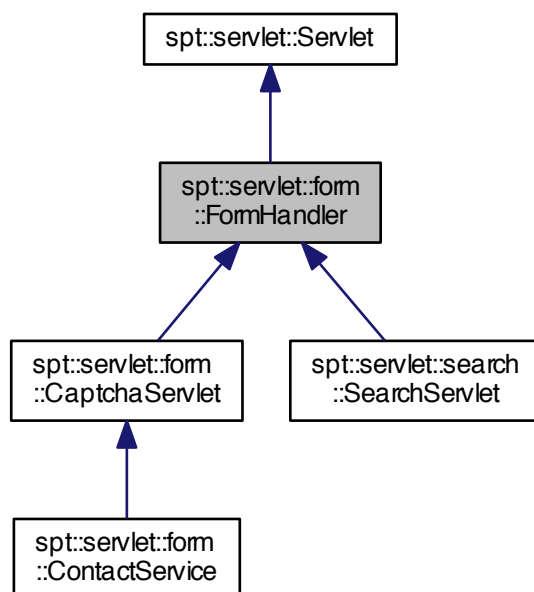
- /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/[FileServlet.h](#)

7.13 spt::servlet::form::FormHandler Class Reference

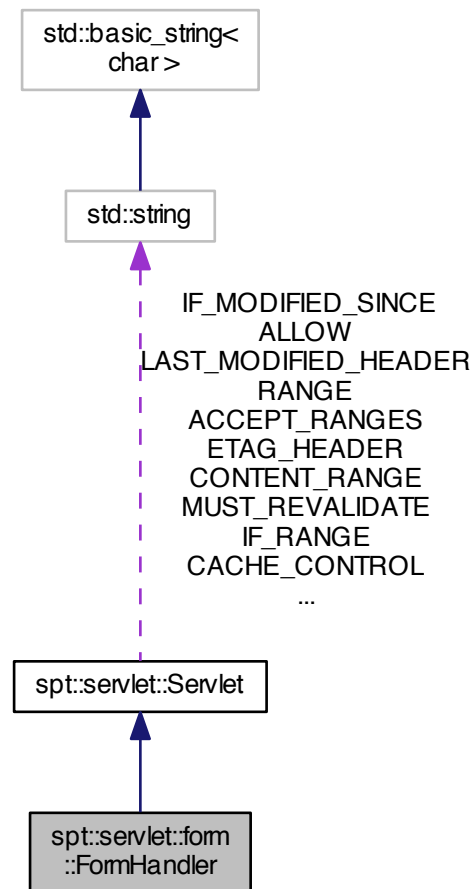
A base form handler servlet.

```
#include <FormHandler.h>
```

Inheritance diagram for spt::servlet::form::FormHandler:



Collaboration diagram for spt::servlet::form::FormHandler:



Public Member Functions

- virtual [~FormHandler](#) ()
Virtual DTOR for sub-classes.

Protected Member Functions

- void [doOptions](#) (Poco::Net::HTTPServerRequest &req, Poco::Net::HTTPServerResponse &resp)
Handle a `OPTIONS` HTTP request from a client.
- void [doPost](#) (Poco::Net::HTTPServerRequest &req, Poco::Net::HTTPServerResponse &resp)
Handle a `POST` HTTP request from a client.
- virtual void [handleForm](#) (const Poco::Net::HTMLForm &form, Poco::Net::HTTPServerRequest &req, Poco::Net::HTTPServerResponse &resp)=0
Handle the form post request from the client.
- void [writeHeaders](#) (Poco::Net::HTTPServerResponse &resp)
Write the standard response headers. Sets the Content-Type to application/json, current date for Last-Modified and specifies no-cache.

- void [toJson](#) (std::string &str)

Perform common transformations to ensure that the specified string be passed as a JSON string value.

Additional Inherited Members

7.13.1 Detailed Description

A base form handler servlet.

Provides default implementations for [doOptions](#) and [doPost](#). Implementations should implement only [handleForm](#).

Since

Version 1.6

Date

Created 2013/02/24 05:48

Copyright

Copyright © 2013, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[FormHandler.h](#) 5588 2013-03-02 02:50:42Z rakesh

7.13.2 Constructor & Destructor Documentation

7.13.2.1 virtual spt::servlet::form::FormHandler () [inline],[virtual]

Virtual DTOR for sub-classes.

7.13.3 Member Function Documentation

7.13.3.1 void spt::servlet::form::FormHandler::doOptions (Poco::Net::HTTPServerRequest & req, Poco::Net::HTTPServerResponse & resp) [protected],[virtual]

Handle a OPTIONS HTTP request from a client.

Parameters

<i>req</i>	The client request object
<i>resp</i>	The server response object

Reimplemented from [spt::servlet::Servlet](#).

7.13.3.2 void spt::servlet::form::FormHandler::doPost (Poco::Net::HTTPServerRequest & req, Poco::Net::HTTPServerResponse & resp) [protected],[virtual]

Handle a POST HTTP request from a client.

Parameters

<i>req</i>	The client request object
<i>resp</i>	The server response object

Reimplemented from [spt::servlet::Servlet](#).

7.13.3.3 `virtual void spt::servlet::form::FormHandler::handleForm ( const Poco::Net::HTMLForm & form, Poco::Net::HTTPServerRequest & req, Poco::Net::HTTPServerResponse & resp ) [protected], [pure virtual]`

Handle the form post request from the client.

Parameters

<i>form</i>	The form that encapsulates the data submitted by the client.
<i>req</i>	The client request object
<i>resp</i>	The server response object.

Implemented in [spt::servlet::search::SearchServlet](#), and [spt::servlet::form::ContactService](#).

7.13.3.4 `void spt::servlet::form::FormHandler::toJson ( std::string & str ) [protected]`

Perform common transformations to ensure that the specified string be passed as a JSON string value.

Parameters

<i>str</i>	The string that is to be transformed.
------------	---------------------------------------

7.13.3.5 `void spt::servlet::form::FormHandler::writeHeaders ( Poco::Net::HTTPServerResponse & resp ) [protected]`

Write the standard response headers. Sets the Content-Type to application/json, current date for Last-Modified and specifies no-cache.

Parameters

<i>resp</i>	The server response object.
-------------	-----------------------------

The documentation for this class was generated from the following file:

- /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/form/[FormHandler.h](#)

7.14 spt::servlet::model::HostConfiguration Class Reference

A value object that represents the configured values for a virtual host in the container. The configuration is parsed from the web.xml and user.xml files configured for a virtual host.

```
#include <HostConfiguration.h>
```

Public Member Functions

- [HostConfiguration](#) ()
Default CTOR.
- const [SecurityConstraints](#) & [getSecurityConstraints](#) () const
Return the security constraints configured in web.xml for the virtual host.
- void [setSecurityConstraints](#) (const [SecurityConstraints](#) &sc)

- Set the security constraints configured in `web.xml` for the virtual host.*

 - [ServletSecurityMap](#) & [getSecurityMap](#) ()

Return the map that stores the security configured for paths for the virtual host.

 - const [ServletSecurityMap](#) & [getSecurityMap](#) () const

const version

 - const [LoginConfigPtr](#) [getLoginConfig](#) () const

Return the login configuration for resources configured in `web.xml` for the virtual host.

 - void [setLoginConfig](#) (const [LoginConfigPtr](#) lp)

const version

 - const [UserMap](#) & [getUserMap](#) () const

Return the mapping of user names to user objects.

 - void [setUserMap](#) (const [UserMap](#) &um)

Set the mapping of user names to user objects for the virtual host.

 - const [ListPtr](#) [getWelcomeFiles](#) () const

Return the list of welcome files configured for the virtual host in `web.xml`.

 - void [setWelcomeFiles](#) (const [ListPtr](#) wf)

Set the list of welcome files configured for the virtual host in `web.xml`.

 - const [ErrorPagePtr](#) [getErrorPages](#) () const

Set the mapping of error pages for various HTTP error codes as configured in `web.xml` for the virtual host.

 - void [setErrorPages](#) (const [ErrorPagePtr](#) ep)

Set the error page mapping for the virtual host.

7.14.1 Detailed Description

A value object that represents the configured values for a virtual host in the container. The configuration is parsed from the `web.xml` and `user.xml` files configured for a virtual host.

Date

Created 2013/01/06 18:41

Copyright

Copyright © 2013, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[HostConfiguration.h](#) 5629 2014-10-01 10:44:20Z rakesh

7.14.2 Constructor & Destructor Documentation

7.14.2.1 `spt::servlet::model::HostConfiguration::HostConfiguration ( )` [inline]

Default CTOR.

7.14.3 Member Function Documentation

7.14.3.1 `const ErrorPagePtr spt::servlet::model::HostConfiguration::getErrorPages ( ) const` `[inline]`

Set the mapping of error pages for various HTTP error codes as configured in `web.xml` for the virtual host.

Returns

`const ErrorPagePtr` The mapping of error pages to codes.

7.14.3.2 `const LoginConfigPtr spt::servlet::model::HostConfiguration::getLoginConfig ( ) const` `[inline]`

Return the login configuration for resources configured in `web.xml` for the virtual host.

Returns

`const LoginConfigPtr` The login configuration

7.14.3.3 `const SecurityConstraints& spt::servlet::model::HostConfiguration::getSecurityConstraints ( ) const` `[inline]`

Return the security constraints configured in `web.xml` for the virtual host.

Returns

`const SecurityConstraints` The security constraints.

7.14.3.4 `ServletSecurityMap& spt::servlet::model::HostConfiguration::getSecurityMap ( )` `[inline]`

Return the map that stores the security configured for paths for the virtual host.

Returns

`ServletSecurityMap` The mapping

7.14.3.5 `const ServletSecurityMap& spt::servlet::model::HostConfiguration::getSecurityMap ( ) const` `[inline]`

`const version`

7.14.3.6 `const UserMap& spt::servlet::model::HostConfiguration::getUserMap ( ) const` `[inline]`

Return the mapping of user names to user objects.

Returns

`const UserMap` The mapping of user names.

7.14.3.7 `const ListPtr spt::servlet::model::HostConfiguration::getWelcomeFiles ( ) const` `[inline]`

Return the list of welcome files configured for the virtual host in `web.xml`.

Returns

`const ListPtr` The list of welcome files.

7.14.3.8 `void spt::servlet::model::HostConfiguration::setErrorPages ( const ErrorPagePtr ep )` `[inline]`

Set the error page mapping for the virtual host.

Parameters

<i>ep</i>	The error page mappings to store in this instance.
-----------	--

7.14.3.9 void spt::servlet::model::HostConfiguration::setLoginConfig (const LoginConfigPtr *lp*) [inline]

const version

7.14.3.10 void spt::servlet::model::HostConfiguration::setSecurityConstraints (const SecurityConstraints & *sc*) [inline]

Set the security constraints configured in `web.xml` for the virtual host.

Parameters

<i>sc</i>	The security constraints to store in this instance.
-----------	---

7.14.3.11 void spt::servlet::model::HostConfiguration::setUserMap (const UserMap & *um*) [inline]

Set the mapping of user names to user objects for the virtual host.

Parameters

<i>um</i>	The map to store in this instance.
-----------	------------------------------------

7.14.3.12 void spt::servlet::model::HostConfiguration::setWelcomeFiles (const ListPtr *wf*) [inline]

Set the list of welcome files configured for the virtual host in `web.xml`.

Parameters

<i>wf</i>	The list of welcome files to store in this instance.
-----------	--

The documentation for this class was generated from the following file:

- [/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/HostConfiguration.h](#)

7.15 spt::servlet::model::HostContext Class Reference

A value object that stores the information pertinent to a virtual host.

```
#include <HostContext.h>
```

Public Member Functions

- [HostContext](#) (const [ContainerConfigurationPtr](#) *ccp*, Poco::Logger **al*)
Create a new instance using the specified container configuration and access logger.
- [~HostContext](#) ()
DTOR to clean up resources. Stops the document root indexer if running.
- const [ContainerConfigurationPtr](#) [getContainerConfiguration](#) () const
Returns the container configuration for the server.
- const [HostConfigurationPtr](#) [getHostConfiguration](#) () const
Return the virtual host configuration information.

- [HostConfigurationPtr getHostConfiguration \(\)](#)
Non-const version.
- [spt::servlet::search::SearchIndexManagerPtr getSearchIndexManager \(\)](#)
Return the site search index manager instance for the virtual host. Only available for hosts with a search index configured. Callers must check for valid pointer.
- void [setSearchIndexManager](#) (const [spt::servlet::search::SearchIndexManagerPtr](#) sm)
Set the search index manager to use for the virtual host.
- const std::string & [getDocumentRoot](#) () const
Return the document root directory location configured for the virtual host.
- void [setDocumentRoot](#) (const std::string &dr)
Set the document root directory for the virtual host.
- Poco::Logger * [getAccessLogger](#) ()
Return the access logger configured for the virtual host. Note that this may be `NULL` if not configured.

7.15.1 Detailed Description

A value object that stores the information pertinent to a virtual host.

Convenience class that makes available configuration information, host scoped objects (search index managers, file monitors etc.) to objects down stream (servlets and other support classes).

Date

Created 2013/01/05 18:41

Copyright

Copyright © 2013, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[HostContext.h](#) 5629 2014-10-01 10:44:20Z rakesh

7.15.2 Constructor & Destructor Documentation

7.15.2.1 [spt::servlet::model::HostContext::HostContext \(const ContainerConfigurationPtr ccp, Poco::Logger * al \)](#)
[inline]

Create a new instance using the specified container configuration and access logger.

Parameters

<i>ccp</i>	The container configuration for the server.
<i>al</i>	The access logger to use for the virtual host.

7.15.2.2 [spt::servlet::model::HostContext::~~HostContext \(\)](#)

DTOR to clean up resources. Stops the document root indexer if running.

7.15.3 Member Function Documentation

7.15.3.1 Poco::Logger* spt::servlet::model::HostContext::getAccessLogger () [inline]

Return the access logger configured for the virtual host. Note that this may be `NULL` if not configured.

Returns

Poco::Logger The pointer to logger to use.

7.15.3.2 const ContainerConfigurationPtr spt::servlet::model::HostContext::getContainerConfiguration () const [inline]

Returns the container configuration for the server.

Returns

const ContainerConfigurationPtr The pointer to the container configuration.

7.15.3.3 const std::string& spt::servlet::model::HostContext::getDocumentRoot () const [inline]

Return the document root directory location configured for the virtual host.

Returns

const std::string

7.15.3.4 const HostConfigurationPtr spt::servlet::model::HostContext::getHostConfiguration () const [inline]

Return the virtual host configuration information.

Returns

const HostConfigurationPtr The pointer to the virtual host configuration data.

7.15.3.5 HostConfigurationPtr spt::servlet::model::HostContext::getHostConfiguration () [inline]

Non-const version.

7.15.3.6 spt::servlet::search::SearchIndexManagerPtr spt::servlet::model::HostContext::getSearchIndexManager () [inline]

Return the site search index manager instance for the virtual host. Only available for hosts with a search index configured. Callers must check for valid pointer.

Returns

[spt::servlet::search::SearchIndexManagerPtr](#) The search index manager for the virtual host if applicable.

7.15.3.7 void spt::servlet::model::HostContext::setDocumentRoot (const std::string & dr) [inline]

Set the document root directory for the virtual host.

Parameters

<i>dr</i>	The document root directory
-----------	-----------------------------

7.15.3.8 `void spt::servlet::model::HostContext::setSearchIndexManager ( const spt::servlet::search::SearchIndexManagerPtr sm ) [inline]`

Set the search index manager to use for the virtual host.

Parameters

<i>sm</i>	The search index manager to use.
-----------	----------------------------------

The documentation for this class was generated from the following file:

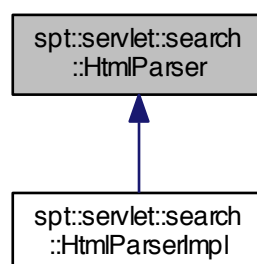
- [/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/HostContext.h](#)

7.16 spt::servlet::search::HtmlParser Class Reference

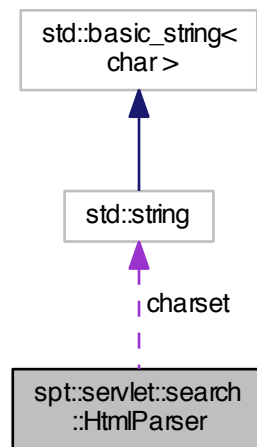
A parser for extracting the plain text contents and some meta-data from a HTML string.

```
#include <HtmlParser.h>
```

Inheritance diagram for spt::servlet::search::HtmlParser:



Collaboration diagram for spt::servlet::search::HtmlParser:



Public Member Functions

- [HtmlParser](#) ()
Default CTOR.
- virtual [~HtmlParser](#) ()
Virtual DTOR for sub-classes.
- std::string [toPlainText](#) (const std::string &html)
Convert the specified HTML text into plain text by stripping tags. Very rudimentary approach.
- std::string [toPlainText](#) (const Poco::File &file)
Extract the plain text contents from the HTML file specified. Very rudimentary approach.
- void [parseHtml](#) (const std::string &text)
Extract the plain text content and some meta-data from the HTML content specified.

Protected Member Functions

- void [decodeEntities](#) (std::string &s)
Replace encoded HTML entities with their proper character equivalents.
- void [compact](#) (std::string &text)
Attempt to remove unnecessary spaces and new lines from the string. This is usually run on the plain text extracted from HTML.
- virtual void [processText](#) (const std::string &)
SAX style callback method for processing the text extracted from a HTML tag.
- virtual bool [openingTag](#) (const std::string &)
SAX style callback method invoked when an opening tag is encountered while parsing the HTML.
- virtual bool [closingTag](#) (const std::string &)
SAX style callback method invoked when a closing tag is encountered while parsing the HTML.
- bool [getParameter](#) (const std::string ¶m, std::string &value) const
Returns the value of a parameter/attribute for a tag. During parsing attribute values are stored in a map, and later looked up as necessary.

Protected Attributes

- bool [inScript](#)
Flag to keep track of whether a script element is being processed.
- std::string [charset](#)
The charset determined by header information.

Static Protected Attributes

- static Poco::HashMap
< std::string, unsigned int > [namedEntities](#)
HTML entity to character equivalent mapping.

7.16.1 Detailed Description

A parser for extracting the plain text contents and some meta-data from a HTML string.

Date

Created 2013/01/11 15:41

Copyright

Copyright © 2013, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[HtmlParser.h](#) 5567 2013-02-23 22:13:21Z rakesh

7.16.2 Constructor & Destructor Documentation

7.16.2.1 `spt::servlet::search::HtmlParser::HtmlParser ( )`

Default CTOR.

7.16.2.2 `virtual spt::servlet::search::HtmlParser::~~HtmlParser ( ) [inline], [virtual]`

Virtual DTOR for sub-classes.

7.16.3 Member Function Documentation

7.16.3.1 `virtual bool spt::servlet::search::HtmlParser::closingTag ( const std::string & ) [inline], [protected], [virtual]`

SAX style callback method invoked when a closing tag is encountered while parsing the HTML.

Parameters

<i>tag</i>	The closing tag that was encountered.
------------	---------------------------------------

Returns

bool Not used at present. Always returns `true`

Reimplemented in [spt::servlet::search::HtmlParserImpl](#).

7.16.3.2 void spt::servlet::search::HtmlParser::compact (std::string & *text*) [protected]

Attempt to remove unnecessary spaces and new lines from the string. This is usually run on the plain text extracted from HTML.

Parameters

<i>text</i>	The text to clean
-------------	-------------------

7.16.3.3 void spt::servlet::search::HtmlParser::decodeEntities (std::string & *s*) [protected]

Replace encoded HTML entities with their proper character equivalents.

Parameters

<i>s</i>	The HTML string that is to be converted.
----------	--

7.16.3.4 bool spt::servlet::search::HtmlParser::getParameter (const std::string & *param*, std::string & *value*) const [protected]

Returns the value of a parameter/attribute for a tag. During parsing attribute values are stored in a map, and later looked up as necessary.

Parameters

<i>param</i>	The name of the attribute to look up.
<i>value</i>	The value that will be populated with the parsed value.

Returns

bool Return `true` if the value was populated.

7.16.3.5 virtual bool spt::servlet::search::HtmlParser::openingTag (const std::string &) [inline], [protected], [virtual]

SAX style callback method invoked when an opening tag is encountered while parsing the HTML.

Parameters

<i>tag</i>	The tag that was encountered
------------	------------------------------

Returns

bool A flag indicating that the contents of the tag should be processed or not.

Reimplemented in [spt::servlet::search::HtmlParserImpl](#).

7.16.3.6 `void spt::servlet::search::HtmlParser::parseHtml ( const std::string & text )`

Extract the plain text content and some meta-data from the HTML constant specified.

Parameters

<i>text</i>	The HTML content to parse.
-------------	----------------------------

7.16.3.7 `virtual void spt::servlet::search::HtmlParser::processText ( const std::string & )` `[inline]`, `[protected]`, `[virtual]`

SAX style callback method for processing the text extracted from a HTML tag.

Parameters

<i>text</i>	The text to process
-------------	---------------------

Reimplemented in [spt::servlet::search::HtmlParserImpl](#).

7.16.3.8 `std::string spt::servlet::search::HtmlParser::toPlainText ( const std::string & html )`

Convert the specified HTML text into plain text by stripping tags. Very rudimentary approach.

Parameters

<i>html</i>	The html text to parse
-------------	------------------------

Returns

`std::string` The plain text content

7.16.3.9 `std::string spt::servlet::search::HtmlParser::toPlainText ( const Poco::File & file )`

Extract the plain text contents from the HTML file specified. Very rudimentary approach.

Parameters

<i>file</i>	The file to parse.
-------------	--------------------

Returns

`std::string` The plain text contents.

7.16.4 Member Data Documentation

7.16.4.1 `std::string spt::servlet::search::HtmlParser::charset` `[protected]`

The charset determined by header information.

7.16.4.2 `bool spt::servlet::search::HtmlParser::inScript` `[protected]`

Flag to keep track of whether a script element is being processed.

7.16.4.3 `Poco::HashMap<std::string, unsigned int> spt::servlet::search::HtmlParser::namedEntities` `[static]`, `[protected]`

HTML entity to character equivalent mapping.

The documentation for this class was generated from the following file:

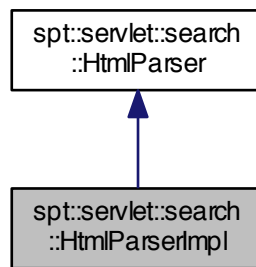
- `/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/HtmlParser.h`

7.17 spt::servlet::search::HtmlParserImpl Class Reference

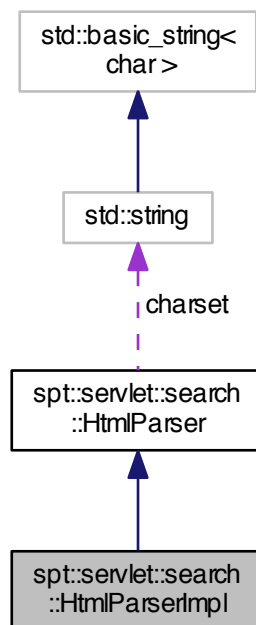
A more robust HTML parser implementation based on the Xapian Omega search engine project.

```
#include <HtmlParserImpl.h>
```

Inheritance diagram for spt::servlet::search::HtmlParserImpl:



Collaboration diagram for spt::servlet::search::HtmlParserImpl:



Public Member Functions

- [HtmlParserImpl\(\)](#)

Default CTOR.

- void [parseHtml](#) (const Poco::File &file, const std::string &charset="utf-8", const bool charsetFromMeta=false)
Parse the contents of the file using the specified encoding.
- void [parseHtml](#) (const std::string &text, const std::string &charset, const bool charsetFromMeta)
Parse the HTML text specified using the specified encoding.
- const std::string & [getPlainText](#) () const
Return the plain text parsed from the previously processed HTML file/text.
- const std::string & [getTitle](#) () const
Return the title for the HTML file parsed from the meta tag.
- const std::string & [getDescription](#) () const
Return the description for the HTML file parsed from the meta tag.
- const std::string & [getKeywords](#) () const
Return the keywords parsed from the meta tag.
- bool [getIndexingAllowed](#) () const
A flag that indicates whether header meta tag included a tag to prevent crawlers/robots indexing the file.
- bool [getIgnoreMetaRobots](#) () const
A flag used to control whether the parser should ignore meta tags that prohibit robots indexing the page.
- void [setIgnoreMetaRobots](#) (const bool value=true)
Set the flag used to indicate that the parser should ignore any meta tags that indicate the HTML should not be indexed.
- void [reset](#) ()
Reset the parser. Used to re-use the instance for parsing other files, or re-trying the parse.

Protected Member Functions

- void [processText](#) (const std::string &text)
SAX style callback method for processing the text extracted from a HTML tag.
- bool [openingTag](#) (const std::string &tag)
SAX style callback method invoked when an opening tag is encountered while parsing the HTML.
- bool [closingTag](#) (const std::string &tag)
SAX style callback method invoked when a closing tag is encountered while parsing the HTML.

Additional Inherited Members

7.17.1 Detailed Description

A more robust HTML parser implementation based on the Xapian Omega search engine project.

Date

Created 2013/01/13 15:41

Copyright

Copyright © 2013, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[HtmlParserImpl.h](#) 5567 2013-02-23 22:13:21Z rakesh

7.17.2 Constructor & Destructor Documentation

7.17.2.1 `spt::servlet::search::HtmlParserImpl::HtmlParserImpl ( )`

Default CTOR.

7.17.3 Member Function Documentation

7.17.3.1 `bool spt::servlet::search::HtmlParserImpl::closingTag ( const std::string & )` `[protected]`, `[virtual]`

SAX style callback method invoked when a closing tag is encountered while parsing the HTML.

Parameters

<i>tag</i>	The closing tag that was encountered.
------------	---------------------------------------

Returns

`bool` Not used at present. Always returns `true`

Reimplemented from [spt::servlet::search::HtmlParser](#).

7.17.3.2 `const std::string& spt::servlet::search::HtmlParserImpl::getDescription ( ) const` `[inline]`

Return the description for the HTML file parsed from the meta tag.

Returns

`const std::string` The description parsed from header meta tag.

7.17.3.3 `bool spt::servlet::search::HtmlParserImpl::getIgnoreMetaRobots ( ) const` `[inline]`

A flag used to control whether the parser should ignore meta tags that prohibit robots indexing the page.

Returns

`bool` Return the value set.

7.17.3.4 `bool spt::servlet::search::HtmlParserImpl::getIndexingAllowed ( ) const` `[inline]`

A flag that indicates whether header meta tag included a tag to prevent crawlers/robots indexing the file.

Returns

`bool` Return `true` if indexing is allowed

7.17.3.5 `const std::string& spt::servlet::search::HtmlParserImpl::getKeywords ( ) const` `[inline]`

Return the keywords parsed from the meta tag.

Returns

`const std::string` The keywords from header meta tag

7.17.3.6 `const std::string& spt::servlet::search::HtmlParserImpl::getPlainText ( ) const` `[inline]`

Return the plain text parsed from the previously processed HTML file/text.

Returns

`const std::string` The plain text content

7.17.3.7 `const std::string& spt::servlet::search::HtmlParserImpl::getTitle ( ) const` `[inline]`

Return the title for the HTML file parsed from the meta tag.

Returns

`const std::string` The title parsed from header meta tag.

7.17.3.8 `bool spt::servlet::search::HtmlParserImpl::openingTag ( const std::string & )` `[protected]`, `[virtual]`

SAX style callback method invoked when an opening tag is encountered while parsing the HTML.

Parameters

<i>tag</i>	The tag that was encountered
------------	------------------------------

Returns

`bool` A flag indicating that the contents of the tag should be processed or not.

Reimplemented from [spt::servlet::search::HtmlParser](#).

7.17.3.9 `void spt::servlet::search::HtmlParserImpl::parseHtml ( const Poco::File & file, const std::string & charset = "utf-8", const bool charsetFromMeta = false )`

Parse the contents of the file using the specified encoding.

Parameters

<i>file</i>	The HTML file to parse.
<i>charset</i>	The charset to use to read the content
<i>charsetFromMeta</i> ↔ <i>Meta</i>	Flag indicating whether the charset to use to read the content should be taken from the HTML meta tags.

7.17.3.10 `void spt::servlet::search::HtmlParserImpl::parseHtml ( const std::string & text, const std::string & charset, const bool charsetFromMeta )`

Parse the HTML text specified using the specified encoding.

Parameters

<i>text</i>	
<i>charset</i>	The charset to use to read the content

<code>charsetFromMeta</code>	Flag indicating whether the charset to use to read the content should be taken from the HTML meta tags.
------------------------------	---

7.17.3.11 `void spt::servlet::search::HtmlParserImpl::processText ( const std::string & )` [protected], [virtual]

SAX style callback method for processing the text extracted from a HTML tag.

Parameters

<code>text</code>	The text to process
-------------------	---------------------

Reimplemented from [spt::servlet::search::HtmlParser](#).

7.17.3.12 `void spt::servlet::search::HtmlParserImpl::reset ( )`

Reset the parser. Used to re-use the instance for parsing other files, or re-trying the parse.

7.17.3.13 `void spt::servlet::search::HtmlParserImpl::setIgnoreMetaRobots ( const bool value = true )` [inline]

Set the flag used to indicate that the parser should ignore any meta tags that indicate the HTML should not be indexed.

Parameters

<code>value</code>	
--------------------	--

The documentation for this class was generated from the following file:

- [/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/HtmlParserImpl.h](#)

7.18 Poco::Util::JSONArray Class Reference

```
#include <JSONArray.h>
```

Public Types

- typedef SharedPtr< [JSONArray](#) > Ptr

Public Member Functions

- [JSONArray](#) ()
Default constructor.
- [JSONArray](#) (const [JSONArray](#) ©)
Copy Constructor.
- virtual [~JSONArray](#) ()
Destructor.
- DynamicAny [get](#) (unsigned int index) const
Retrieves an element. Will return an empty value when the element doesn't exist.
- [JSONArray::Ptr](#) [getArray](#) (unsigned int index) const
Retrieves an array. When the element is not an array or doesn't exist, an empty SharedPtr is returned.
- template<typename T >
T [getElement](#) (unsigned int index) const

Retrieves an element and tries to convert it to the template type. The `convert<T>` method of `Dynamic` is called which can also throw exceptions for invalid values. Note: This will not work for an array or an object.

- `SharedPtr< JSONObject > getObject` (unsigned int index) const
Retrieves an object. When the element is not an object or doesn't exist, an empty `SharedPtr` is returned.
- unsigned int `size` () const
Returns the size of the array.
- bool `isArray` (unsigned int index) const
Returns true when the element is an array.
- bool `isNull` (unsigned int index) const
Returns true when the element is null or when the element doesn't exist.
- bool `isObject` (unsigned int index) const
Returns true when the element is an object.
- template<typename T >
T `optElement` (unsigned int index, const T &def) const
Returns the element at the given index. When the element is null, doesn't exist or can't be converted to the given type, the default value will be returned.
- void `add` (const `DynamicAny` &value)
Add the given value to the array.
- void `stringify` (std::ostream &out, unsigned int indent) const
Prints the array to out. When indent is 0, the array will be printed on one line without indentation.
- void `remove` (unsigned int index)
Removes the element on the given index.

7.18.1 Member Typedef Documentation

7.18.1.1 `typedef SharedPtr<JSONArray> Poco::Util::JSONArray::Ptr`

7.18.2 Constructor & Destructor Documentation

7.18.2.1 `Poco::Util::JSONArray::JSONArray ( )`

Default constructor.

7.18.2.2 `Poco::Util::JSONArray::JSONArray ( const JSONArray © )`

Copy Constructor.

7.18.2.3 `virtual Poco::Util::JSONArray::~JSONArray ( )` [virtual]

Destructor.

7.18.3 Member Function Documentation

7.18.3.1 `void Poco::Util::JSONArray::add ( const DynamicAny &value )` [inline]

Add the given value to the array.

7.18.3.2 `DynamicAny Poco::Util::JSONArray::get ( unsigned int index ) const`

Retrieves an element. Will return an empty value when the element doesn't exist.

7.18.3.3 `JSONArray::Ptr` `Poco::Util::JSONArray::getArray ( unsigned int index ) const`

Retrieves an array. When the element is not an array or doesn't exist, an empty `SharedPtr` is returned.

7.18.3.4 `template<typename T> T Poco::Util::JSONArray::getElement ( unsigned int index ) const` `[inline]`

Retrieves an element and tries to convert it to the template type. The `convert<T>` method of `Dynamic` is called which can also throw exceptions for invalid values. Note: This will not work for an array or an object.

7.18.3.5 `SharedPtr<JSONObject> Poco::Util::JSONArray::getObject ( unsigned int index ) const`

Retrieves an object. When the element is not an object or doesn't exist, an empty `SharedPtr` is returned.

7.18.3.6 `bool Poco::Util::JSONArray::isArray ( unsigned int index ) const` `[inline]`

Returns true when the element is an array.

7.18.3.7 `bool Poco::Util::JSONArray::isNull ( unsigned int index ) const` `[inline]`

Returns true when the element is null or when the element doesn't exist.

7.18.3.8 `bool Poco::Util::JSONArray::isObject ( unsigned int index ) const`

Returns true when the element is an object.

7.18.3.9 `template<typename T> T Poco::Util::JSONArray::optElement ( unsigned int index, const T & def ) const` `[inline]`

Returns the element at the given index. When the element is null, doesn't exist or can't be converted to the given type, the default value will be returned.

7.18.3.10 `void Poco::Util::JSONArray::remove ( unsigned int index )` `[inline]`

Removes the element on the given index.

7.18.3.11 `unsigned int Poco::Util::JSONArray::size ( ) const` `[inline]`

Returns the size of the array.

7.18.3.12 `void Poco::Util::JSONArray::stringify ( std::ostream & out, unsigned int indent ) const`

Prints the array to `out`. When `indent` is 0, the array will be printed on one line without indentation.

The documentation for this class was generated from the following file:

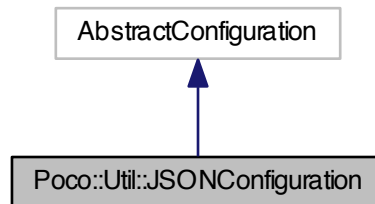
- `/Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONArray.h`

7.19 Poco::Util::JSONConfiguration Class Reference

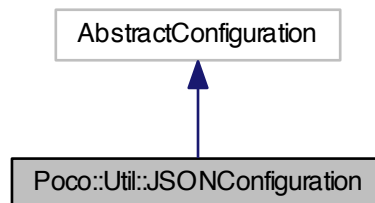
This configuration class extracts configuration properties from a JSON object. An XPath-like syntax for property names is supported to allow full access to the JSON object.

```
#include <JSONConfiguration.h>
```

Inheritance diagram for Poco::Util::JSONConfiguration:



Collaboration diagram for Poco::Util::JSONConfiguration:



Public Member Functions

- [JSONConfiguration](#) ()
Creates an empty configuration.
- [JSONConfiguration](#) (const std::string &path)
Creates a configuration and loads the JSON structure from the given stream.
- [JSONConfiguration](#) (std::istream &istr)
Creates a configuration from the given JSON object.
- [JSONConfiguration](#) (const [JSONObject::Ptr](#) &object)
Creates a configuration from the given JSON object.
- virtual [~JSONConfiguration](#) ()
Destructor.
- void [load](#) (const std::string &path)
Loads the configuration from the given file.
- void [load](#) (std::istream &istr)
Loads the configuration from the given stream.

- void `loadEmpty` (const std::string &root)
Loads an empty object containing only a root object with the given name.
- void `save` (std::ostream &ostr, unsigned int indent=2) const
Saves the configuration to the given stream.
- void `setInt` (const std::string &key, int value)
- void `setBool` (const std::string &key, bool value)
- void `setDouble` (const std::string &key, double value)

Protected Member Functions

- bool `getRaw` (const std::string &key, std::string &value) const
- void `setRaw` (const std::string &key, const std::string &value)
- void `enumerate` (const std::string &key, Keys &range) const

7.19.1 Detailed Description

This configuration class extracts configuration properties from a JSON object. An XPath-like syntax for property names is supported to allow full access to the JSON object.

Given the following JSON object as an example: { "config" : { "prop1" : "value1", "prop2" : 10, "prop3" : ["element1", "element2"], "prop4" : { "prop5" : false, "prop6" : null } } } The following property names would be valid and would yield the shown values:

config.prop1 -> "value1" config.prop3[1] -> "element2" config.prop4.prop5 -> false

7.19.2 Constructor & Destructor Documentation

7.19.2.1 Poco::Util::JSONConfiguration::JSONConfiguration ()

7.19.2.2 Poco::Util::JSONConfiguration::JSONConfiguration (const std::string & path)

Creates an empty configuration.

Creates a configuration and loads the JSON structure from the given file

7.19.2.3 Poco::Util::JSONConfiguration::JSONConfiguration (std::istream & istr)

Creates a configuration and loads the JSON structure from the given stream.

7.19.2.4 Poco::Util::JSONConfiguration::JSONConfiguration (const JSONObject::Ptr & object)

Creates a configuration from the given JSON object.

7.19.2.5 virtual Poco::Util::JSONConfiguration::~~JSONConfiguration () [virtual]

Destructor.

7.19.3 Member Function Documentation

7.19.3.1 void Poco::Util::JSONConfiguration::enumerate (const std::string & key, Keys & range) const [protected]

7.19.3.2 bool Poco::Util::JSONConfiguration::getRaw (const std::string & key, std::string & value) const [protected]

7.19.3.3 void Poco::Util::JSONConfiguration::load (const std::string & *path*)

Loads the configuration from the given file.

7.19.3.4 void Poco::Util::JSONConfiguration::load (std::istream & *istr*)

Loads the configuration from the given stream.

7.19.3.5 void Poco::Util::JSONConfiguration::loadEmpty (const std::string & *root*)

Loads an empty object containing only a root object with the given name.

7.19.3.6 void Poco::Util::JSONConfiguration::save (std::ostream & *ostr*, unsigned int *indent* = 2) const

Saves the configuration to the given stream.

7.19.3.7 void Poco::Util::JSONConfiguration::setBool (const std::string & *key*, bool *value*)

7.19.3.8 void Poco::Util::JSONConfiguration::setDouble (const std::string & *key*, double *value*)

7.19.3.9 void Poco::Util::JSONConfiguration::setInt (const std::string & *key*, int *value*)

7.19.3.10 void Poco::Util::JSONConfiguration::setRaw (const std::string & *key*, const std::string & *value*) [protected]

The documentation for this class was generated from the following file:

- /Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONConfiguration.h

7.20 Poco::Util::JSONObject Class Reference

Represents a JSON object.

```
#include <JSONObject.h>
```

Public Types

- typedef SharedPtr< [JSONObject](#) > [Ptr](#)

Public Member Functions

- [JSONObject](#) ()
Default constructor.
- [JSONObject](#) (const [JSONObject](#) ©)
- virtual [~JSONObject](#) ()
Destructor.
- DynamicAny [get](#) (const std::string &key) const
Retrieves a property. An empty value is returned when the property doesn't exist.
- [JSONArray::Ptr](#) [getArray](#) (const std::string &key) const
Returns a SharedPtr to an array when the property is an array. An empty SharedPtr is returned when the element doesn't exist or is not an array.
- [JSONObject::Ptr](#) [getObject](#) (const std::string &key) const

Returns a `SharedPtr` to an object when the property is an object. An empty `SharedPtr` is returned when the property doesn't exist or is not an object.

- `template<typename T >`
`T getValue (const std::string &key) const`
Retrieves the property with the given name and will try to convert the value to the given template type. The `convert<↔ T>` method of `Dynamic` is called which can also throw exceptions for invalid values. Note: This will not work for an array or an object.
- `void getNames (std::vector< std::string > &names) const`
Returns all property names.
- `bool has (const std::string &key) const`
Returns true when the given property exists.
- `bool isArray (const std::string &key) const`
Returns true when the given property contains an array.
- `bool isNull (const std::string &key) const`
Returns true when the given property contains a null value.
- `bool isObject (const std::string &key) const`
Returns true when the given property contains an object.
- `template<typename T >`
`T optValue (const std::string &key, const T &def) const`
Returns the value of a property when the property exists and can be converted to the given type. Otherwise `def` will be returned.
- `unsigned int size () const`
Returns the number of properties.
- `void set (const std::string &key, const DynamicAny &value)`
Sets a new value.
- `void stringify (std::ostream &out, unsigned int indent=0) const`
Prints the object to `out`. When `indent` is 0, the object will be printed on one line without indentation.
- `void remove (const std::string &key)`
Removes the property with the given key.

7.20.1 Detailed Description

Represents a JSON object.

7.20.2 Member Typedef Documentation

7.20.2.1 `typedef SharedPtr<JSONObject> Poco::Util::JSONObject::Ptr`

7.20.3 Constructor & Destructor Documentation

7.20.3.1 `Poco::Util::JSONObject::JSONObject ( )`

Default constructor.

7.20.3.2 `Poco::Util::JSONObject::JSONObject ( const JSONObject © )`

7.20.3.3 `virtual Poco::Util::JSONObject::~~JSONObject ( ) [virtual]`

Destructor.

7.20.4 Member Function Documentation

7.20.4.1 DynamicAny Poco::Util::JSONObject::get (const std::string & *key*) const

Retrieves a property. An empty value is returned when the property doesn't exist.

7.20.4.2 JSONArray::Ptr Poco::Util::JSONObject::getArray (const std::string & *key*) const

Returns a SharedPtr to an array when the property is an array. An empty SharedPtr is returned when the element doesn't exist or is not an array.

7.20.4.3 void Poco::Util::JSONObject::getNames (std::vector< std::string > & *names*) const

Returns all property names.

7.20.4.4 JSONObject::Ptr Poco::Util::JSONObject::getObject (const std::string & *key*) const

Returns a SharedPtr to an object when the property is an object. An empty SharedPtr is returned when the property doesn't exist or is not an object.

7.20.4.5 template<typename T > T Poco::Util::JSONObject::getValue (const std::string & *key*) const [inline]

Retrieves the property with the given name and will try to convert the value to the given template type. The convert<T> method of Dynamic is called which can also throw exceptions for invalid values. Note: This will not work for an array or an object.

7.20.4.6 bool Poco::Util::JSONObject::has (const std::string & *key*) const [inline]

Returns true when the given property exists.

7.20.4.7 bool Poco::Util::JSONObject::isArray (const std::string & *key*) const [inline]

Returns true when the given property contains an array.

7.20.4.8 bool Poco::Util::JSONObject::isNull (const std::string & *key*) const [inline]

Returns true when the given property contains a null value.

7.20.4.9 bool Poco::Util::JSONObject::isObject (const std::string & *key*) const [inline]

Returns true when the given property contains an object.

7.20.4.10 template<typename T > T Poco::Util::JSONObject::optValue (const std::string & *key*, const T & *def*) const [inline]

Returns the value of a property when the property exists and can be converted to the given type. Otherwise def will be returned.

7.20.4.11 void Poco::Util::JSONObject::remove (const std::string & *key*) [inline]

Removes the property with the given key.

7.20.4.12 `void Poco::Util::JSONObject::set ( const std::string & key, const DynamicAny & value ) [inline]`

Sets a new value.

7.20.4.13 `unsigned int Poco::Util::JSONObject::size ( ) const [inline]`

Returns the number of properties.

7.20.4.14 `void Poco::Util::JSONObject::stringify ( std::ostream & out, unsigned int indent = 0 ) const`

Prints the object to out. When indent is 0, the object will be printed on one line without indentation.

The documentation for this class was generated from the following file:

- [/Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONObject.h](#)

7.21 Poco::Util::JSONParser Class Reference

A class for passing JSON strings or streams.

```
#include <JSONParser.h>
```

Public Member Functions

- [JSONParser \(\)](#)
Constructor.
- virtual [~JSONParser \(\)](#)
Destructor.
- DynamicAny [parse](#) (const std::string &source)
Parses a string.
- DynamicAny [parse](#) (std::istream &in)
Parses a JSON from the input stream.

7.21.1 Detailed Description

A class for passing JSON strings or streams.

7.21.2 Constructor & Destructor Documentation

7.21.2.1 `Poco::Util::JSONParser::JSONParser ( )`

Constructor.

7.21.2.2 `virtual Poco::Util::JSONParser::~~JSONParser ( ) [virtual]`

Destructor.

7.21.3 Member Function Documentation

7.21.3.1 DynamicAny Poco::Util::JSONParser::parse (const std::string & source) [inline]

Parses a string.

7.21.3.2 DynamicAny Poco::Util::JSONParser::parse (std::istream & in)

Parses a JSON from the input stream.

The documentation for this class was generated from the following file:

- [/Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONParser.h](#)

7.22 Poco::Util::JSONQuery Class Reference

Class that can be used to search for a value in a JSON object or array.

```
#include <JSONQuery.h>
```

Public Member Functions

- [JSONQuery](#) (const DynamicAny &source)
Constructor. Pass the start object/array.
- virtual [~JSONQuery](#) ()
Destructor.
- [JSONObject::Ptr findObject](#) (const std::string &path) const
Search for an object. When the object can't be found, an empty SharedPtr is returned.
- [JSONArray::Ptr findArray](#) (const std::string &path) const
Search for an array. When the array can't be found, an empty SharedPtr is returned.
- DynamicAny [find](#) (const std::string &path) const
Searches a value For example: "person.children[0].name" will return the the name of the first child. When the value can't be found an empty value is returned.
- template<typename T >
T [findValue](#) (const std::string &path, const T &def) const
Searches for a value will convert it to the given type. When the value can't be found or has an invalid type the default value will be returned.
- std::string [findValue](#) (const char *path, const char *def) const
Searches for a value will convert it to the given type. When the value can't be found or has an invalid type the default value will be returned.

7.22.1 Detailed Description

Class that can be used to search for a value in a JSON object or array.

7.22.2 Constructor & Destructor Documentation

7.22.2.1 Poco::Util::JSONQuery::JSONQuery (const DynamicAny & source)

Constructor. Pass the start object/array.

7.22.2.2 `virtual Poco::Util::JSONQuery::~~JSONQuery ( ) [virtual]`

Destructor.

7.22.3 Member Function Documentation

7.22.3.1 `DynamicAny Poco::Util::JSONQuery::find ( const std::string & path ) const`

Searches a value For example: "person.children[0].name" will return the the name of the first child. When the value can't be found an empty value is returned.

7.22.3.2 `JSONArray::Ptr Poco::Util::JSONQuery::findArray ( const std::string & path ) const`

Search for an array. When the array can't be found, an empty SharedPtr is returned.

7.22.3.3 `JSONObject::Ptr Poco::Util::JSONQuery::findObject ( const std::string & path ) const`

Search for an object. When the object can't be found, an empty SharedPtr is returned.

7.22.3.4 `template<typename T > T Poco::Util::JSONQuery::findValue ( const std::string & path, const T & def ) const [inline]`

Searches for a value will convert it to the given type. When the value can't be found or has an invalid type the default value will be returned.

7.22.3.5 `std::string Poco::Util::JSONQuery::findValue ( const char * path, const char * def ) const [inline]`

Searches for a value will convert it to the given type. When the value can't be found or has an invalid type the default value will be returned.

The documentation for this class was generated from the following file:

- `/Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONQuery.h`

7.23 Poco::Util::JSONStringifier Class Reference

Helper class for creating a String from a JSON object or array.

```
#include <JSONStringifier.h>
```

Static Public Member Functions

- static void `stringify` (const DynamicAny &any, std::ostream &out, unsigned int indent=0)
Writes a String representation of the value to the output stream. When indent is 0, the String will be created as small as possible.

7.23.1 Detailed Description

Helper class for creating a String from a JSON object or array.

7.23.2 Member Function Documentation

7.23.2.1 `static void Poco::Util::JSONStringifier::stringify ( const DynamicAny & any, std::ostream & out, unsigned int indent = 0 ) [static]`

Writes a String representation of the value to the output stream. When indent is 0, the String will be created as small as possible.

The documentation for this class was generated from the following file:

- [/Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONStringifier.h](#)

7.24 spt::servlet::model::LoginConfig Class Reference

A simple value object that represents a login-config element in a web.xml file.

```
#include <LoginConfig.h>
```

Public Member Functions

- [LoginConfig](#) (const std::string &methodName, const std::string &realmText)
Create a new instance that represents the configuration for the specified HTTP method and realm.
- const std::string & [getMethod](#) () const
- const std::string & [getRealm](#) () const

7.24.1 Detailed Description

A simple value object that represents a login-config element in a web.xml file.

Date

Created 2011/09/15 18:41

Copyright

Copyright © 2011, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[LoginConfig.h](#) 5567 2013-02-23 22:13:21Z rakesh

7.24.2 Constructor & Destructor Documentation

7.24.2.1 `spt::servlet::model::LoginConfig::LoginConfig ( const std::string & methodName, const std::string & realmText ) [inline]`

Create a new instance that represents the configuration for the specified HTTP method and realm.

Parameters

<i>methodName</i>	The HTTP method
<i>realmText</i>	The realm to apply

7.24.3 Member Function Documentation

7.24.3.1 `const std::string& spt::servlet::model::LoginConfig::getMethod ( ) const` `[inline]`

Returns

`std::string` The HTTP method for this configuration

7.24.3.2 `const std::string& spt::servlet::model::LoginConfig::getRealm ( ) const` `[inline]`

Returns

`std::string` The realm for this configuration

The documentation for this class was generated from the following file:

- [/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/LoginConfig.h](#)

7.25 spt::servlet::LogManager Class Reference

A singleton class that is used to set up the loggers for the container.

```
#include <LogManager.h>
```

Public Member Functions

- `Poco::Logger & getLogger (const std::string &name, const int level=Poco::Message::PRIO_INFORMATION)`
Return the logger with the specified name.

Static Public Member Functions

- static `LogManager & getLogManager (const model::ContainerConfigurationPtr containerConfiguration)`
Return the singleton log manager instance initialised from the specified container configuration.

7.25.1 Detailed Description

A singleton class that is used to set up the loggers for the container.

Date

Created 2011/08/17 18:41

Copyright

Copyright © 2011, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[LogManager.h](#) 5567 2013-02-23 22:13:21Z rakesh

7.25.2 Member Function Documentation

7.25.2.1 Poco::Logger& spt::servlet::LogManager::getLogger (const std::string & *name*, const int *level* = Poco::Message::PRIO_INFORMATION)

Return the logger with the specified name.

Parameters

<i>name</i>	The name of the logger to retrieve.
<i>level</i>	The priority level for the logger.

Returns

Poco::Logger The logger instance.

7.25.2.2 static LogManager& spt::servlet::LogManager::getLogManager (const model::ContainerConfigurationPtr *containerConfiguration*) [inline],[static]

Return the singleton log manager instance initialised from the specified container configuration.

Parameters

<i>container← Configuration</i>	The container configuration information.
-------------------------------------	--

Returns

[LogManager](#) The singleton log manager instance.

The documentation for this class was generated from the following file:

- /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/[LogManager.h](#)

7.26 spt::servlet::OwnershipManager Class Reference

A utility class used to handle switching of user from root to unprivileged user after the server has started up and bound to the desired port.

```
#include <OwnershipManager.h>
```

Public Member Functions

- [OwnershipManager](#) (const [model::ContainerConfigurationPtr](#) config)
Create a new instance using the specified container configuration data.
- void [switchUser](#) ()
Switch user and group as configured.

7.26.1 Detailed Description

A utility class used to handle switching of user from root to unprivileged user after the server has started up and bound to the desired port.

Date

Created 2011/08/28 18:41

Copyright

Copyright © 2011, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[OwnershipManager.h](#) 5567 2013-02-23 22:13:21Z rakesh

7.26.2 Constructor & Destructor Documentation

7.26.2.1 `spt::servlet::OwnershipManager::OwnershipManager ( const model::ContainerConfigurationPtr config )`

Create a new instance using the specified container configuration data.

Parameters

<i>config</i>	The container configuration data which provides the information about user and group to switch the process ownership to.
---------------	--

7.26.3 Member Function Documentation

7.26.3.1 `void spt::servlet::OwnershipManager::switchUser ( )`

Switch user and group as configured.

The documentation for this class was generated from the following file:

- `/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/OwnershipManager.h`

7.27 `spt::servlet::search::PdfExtractor` Class Reference

Extract text content from PDF files using pdftotext (part of xpdf or the newer poppler system). Return the extracted plain text content.

```
#include <PdfExtractor.h>
```

Public Member Functions

- `bool hasCommand ()`
Indicate whether the `pdftotext` command is available on the system. Command invoked will be `PDFTOTEXT`.
- `std::string extractText (const Poco::File &file)`
Extract the plain text content from the specified PDF file. Returns an empty string if `pdftotext` is not found.

7.27.1 Detailed Description

Extract text content from PDF files using pdftotext (part of xpdf or the newer poppler system). Return the extracted plain text content.

The command line invocation used is as follows:

```
pdftotext -eol unix -layout -nopgbrk /path/to/pdf -
```

The `pdftotext` command is searched for in the following locations:

- /opt/local/bin
- /usr/bin
- /usr/local/bin
- /opt/csw/bin

Date

Created 2013/02/15 18:41

Copyright

Copyright © 2013, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[PdfExtractor.h](#) 5567 2013-02-23 22:13:21Z rakesh

7.27.2 Member Function Documentation

7.27.2.1 std::string spt::servlet::search::PdfExtractor::extractText (const Poco::File & file)

Extract the plain text content from the specified PDF file. Returns an empty string if `pdftotext` is not found.

Any exceptions encountered while extracting text are logged. The plain text returned may represent only a small part of the original PDF content if an exception was handled while extracting the text.

Parameters

<i>file</i>	The PDF file to extract text from.
-------------	------------------------------------

Returns

std::string The extracted plain text.

7.27.2.2 `bool spt::servlet::search::PdfExtractor::hasCommand ( )`

Indicate whether the `pdftotext` command is available on the system. Command invoked will be [PDFTOTEXT](#).

Returns

`bool` Returns `true` if the command exists on the filesystem.

The documentation for this class was generated from the following file:

- [/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/PdfExtractor.h](#)

7.28 `spt::servlet::search::PdfParser` Class Reference

A parser for extracting the text contents of a PDF file using PoDoFo.

```
#include <PdfParser.h>
```

Public Member Functions

- [PdfParser](#) (const Poco::File &file)
- [~PdfParser](#) ()
Clean up resources.
- `std::string` [extract](#) ()

7.28.1 Detailed Description

A parser for extracting the text contents of a PDF file using PoDoFo.

Copyright 2013, Sans Pareil Technologies, Inc.

Id

[PdfParser.h](#) 5567 2013-02-23 22:13:21Z rakesh

7.28.2 Constructor & Destructor Documentation

7.28.2.1 `spt::servlet::search::PdfParser::PdfParser ( const Poco::File & file )`

Create a new instance for extracting the content from the specified file.

Parameters

<i>file</i>	The PDF file from which plain text is to be extracted
-------------	---

7.28.2.2 `spt::servlet::search::PdfParser::~~PdfParser ( )`

Clean up resources.

7.28.3 Member Function Documentation

7.28.3.1 `std::string spt::servlet::search::PdfParser::extract ( )`

Returns

std::string Extract text from the file and return the plain text

The documentation for this class was generated from the following file:

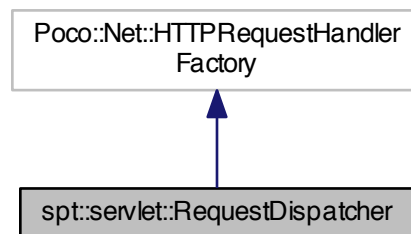
- /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/[PdfParser.h](#)

7.29 spt::servlet::RequestDispatcher Class Reference

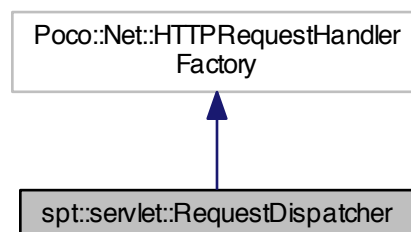
The standard request handler factory that will parse the Host HTTP header and delegate to the appropriate VirtualHost to process the request. The first configured VirtualHost will be used if no Host header has been specified.

```
#include <RequestDispatcher.h>
```

Inheritance diagram for spt::servlet::RequestDispatcher:



Collaboration diagram for spt::servlet::RequestDispatcher:

**Public Member Functions**

- [RequestDispatcher](#) (const std::string &configDirectory, const [model::ContainerConfigurationPtr](#) containerConfiguration)

Create a new request dispatcher using the specified configuration directory and container configuration information.
- Poco::Net::HTTPRequestHandler * [createRequestHandler](#) (const Poco::Net::HTTPServerRequest &request)

Return a new [spt::servlet::ServletHandler](#) instance initialised with the [spt::servlet::Servlet](#) instance mapped to the path, or the default file handler servlet. Delegates to the virtual host mapped to the specified host header.

7.29.1 Detailed Description

The standard request handler factory that will parse the Host HTTP header and delegate to the appropriate VirtualHost to process the request. The first configured VirtualHost will be used if no Host header has been specified.

Date

Created 2011/08/15 18:41

Copyright

Copyright © 2011, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[RequestDispatcher.h](#) 5629 2014-10-01 10:44:20Z rakesh

7.29.2 Constructor & Destructor Documentation

7.29.2.1 `spt::servlet::RequestDispatcher::RequestDispatcher ( const std::string & configDirectory, const model::ContainerConfigurationPtr containerConfiguration )`

Create a new request dispatcher using the specified configuration directory and container configuration information.

Parameters

<i>configDirectory</i>	The directory in which additional configuration files are stored.
<i>containerConfiguration</i>	The container configuration data.

7.29.3 Member Function Documentation

7.29.3.1 `Poco::Net::HTTPRequestHandler* spt::servlet::RequestDispatcher::createRequestHandler ( const Poco::Net::HTTPServerRequest & request )`

Return a new [spt::servlet::ServletHandler](#) instance initialised with the [spt::servlet::Servlet](#) instance mapped to the path, or the default file handler servlet. Delegates to the virtual host mapped to the specified host header.

Parameters

<i>request</i>	The client request object
----------------	---------------------------

Returns

Poco::Net::HTTPRequestHandler The new per request handler object which will be deleted by the Poco implementation.

The documentation for this class was generated from the following file:

- /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/[RequestDispatcher.h](#)

7.30 spt::servlet::search::SearchIndexManager Class Reference

A manager class for maintaining the Xapian search engine index. Usually used to manage indices for one site/virtual host.

```
#include <SearchIndexManager.h>
```

Public Member Functions

- [SearchIndexManager](#) (const std::string &databasePath)
Create a new search index manager for a database at specified path.
- [~SearchIndexManager](#) ()
DTOR. Cleans up resources.
- [SearchResults search](#) (const std::string &querystring, Xapian::doccount offset=0, Xapian::doccount page-size=10)
Search for the specified queryString and return results from the offset up to pagesize. The offset and pagesize parameters allow for paginated search results.
- bool [index](#) (const Poco::File &file, const std::string &urlPath, const std::string &contentType="text/html")
Index the file specified. Only HTML and PDF documents are supported at present. This method does not throw. All exceptions are handled and logged.
- void [remove](#) (const Poco::File &file)
Remove the document for the specified file from the search index.
- void [remove](#) (const Xapian::docid docid)
Remove the document with the specified identifier from the search index.
- Poco::Timestamp::TimeVal [getLastModified](#) (const Poco::File &file)
Return the last modified timestamp value stored for the document associated with the specified file. May be used to check for freshness of the indexed document version.
- bool [indexIfUpdated](#) (const Poco::File &file, const std::string &urlPath, const std::string &contentType="text/html")
Convenience method that checks the timestamp of the specified file against the timestamp for the document representation of the file and re-indexes if necessary.
- Xapian::docid [getLastDocumentId](#) ()
Return the document id of the last document indexed in the search index. May be used to iterate over all the documents indexed.
- [DocumentData getDocument](#) (const Xapian::docid docid)
Retrieve the document stored in the search index with the specified id.

7.30.1 Detailed Description

A manager class for maintaining the Xapian search engine index. Usually used to manage indices for one site/virtual host.

Date

Created 2013/01/12 18:41

Copyright

Copyright © 2013, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[SearchIndexManager.h](#) 5567 2013-02-23 22:13:21Z rakesh

7.30.2 Constructor & Destructor Documentation

7.30.2.1 `spt::servlet::search::SearchIndexManager::SearchIndexManager ( const std::string & databasePath )`

Create a new search index manager for a database at specified path.

Parameters

<i>databasePath</i>	The path to the Xapian search database.
---------------------	---

7.30.2.2 `spt::servlet::search::SearchIndexManager::~~SearchIndexManager ( )`

DTOR. Cleans up resources.

7.30.3 Member Function Documentation

7.30.3.1 `DocumentData spt::servlet::search::SearchIndexManager::getDocument ( const Xapian::docid docid )`

Retrieve the document stored in the search index with the specified id.

Parameters

<i>docid</i>	The document id of the document to retrieve.
--------------	--

Returns

[DocumentData](#) The de-serialised data from the search index.

7.30.3.2 `Xapian::docid spt::servlet::search::SearchIndexManager::getLastDocumentId ( )`

Return the document id of the last document indexed in the search index. May be used to iterate over all the documents indexed.

Returns

Xapian::docid The last document id.

7.30.3.3 `Poco::Timestamp::TimeVal spt::servlet::search::SearchIndexManager::getLastModified ( const Poco::File & file )`

Return the last modified timestamp value stored for the document associated with the specified file. May be used to check for freshness of the indexed document version.

Parameters

<i>file</i>	The file whose associated document timestamp is to be retrieved.
-------------	--

Returns

Poco::Timestamp::TimeVal The timestamp for the file in the index.

7.30.3.4 `bool spt::servlet::search::SearchIndexManager::index ( const Poco::File & file, const std::string & urlPath, const std::string & contentType = "text/html" )`

Index the file specified. Only HTML and PDF documents are supported at present. This method does not `throw`. All exceptions are handled and logged.

Parameters

<i>file</i>	The file to index.
<i>urlPath</i>	The relative URI for the file
<i>contentType</i>	The mime-type for the file.

Returns

`bool` Return `true` if the indexing succeeds. Returns `false` if any exceptions are handled while indexing.

7.30.3.5 `bool spt::servlet::search::SearchIndexManager::indexIfUpdated ( const Poco::File & file, const std::string & urlPath, const std::string & contentType = "text/html" )`

Convenience method that checks the timestamp of the specified file against the timestamp for the document representation of the file and re-indexes if necessary.

Parameters

<i>file</i>	The file to check and index if necessary
<i>urlPath</i>	The relative URI for the file.
<i>contentType</i>	The mime-type for the file.

Returns

`bool` Return `true` only if the file was successfully re-indexed. In particular this method returns `false` if re-indexing is unnecessary.

7.30.3.6 `void spt::servlet::search::SearchIndexManager::remove ( const Poco::File & file )`

Remove the document for the specified file from the search index.

Parameters

<i>file</i>	The file whose associated document is to be removed.
-------------	--

7.30.3.7 `void spt::servlet::search::SearchIndexManager::remove ( const Xapian::docid docid )`

Remove the document with the specified identifier from the search index.

Parameters

<i>docid</i>	The document id for the document.
--------------	-----------------------------------

7.30.3.8 SearchResults `spt::servlet::search::SearchIndexManager::search (const std::string & querystring,
Xapian::doccount offset = 0, Xapian::doccount pagesize = 10)`

Search for the specified `queryString` and return results from the `offset` up to `pagesize`. The `offset` and `pagesize` parameters allow for paginated search results.

Parameters

<i>querystring</i>	The search query to run against the Xapian database
<i>offset</i>	The offset to start returning results from.
<i>pagesize</i>	The number of results to include for this search

Returns

[SearchResults](#) The documents matching the `querystring`.

The documentation for this class was generated from the following file:

- `/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/SearchIndexManager.h`

7.31 spt::servlet::search::SearchResult Class Reference

A wrapper class that encapsulates a document that is part of the search results returned by Xapian for a query.

```
#include <SearchResult.h>
```

Public Member Functions

- [SearchResult](#) (const Xapian::MSetIterator &iter)
Create a new instance from the result set iterator.
- const [DocumentData](#) & [getDocument](#) () const
- double [getWeight](#) () const
- int [getPercent](#) () const
- unsigned [getRank](#) () const
- std::string [getSnippet](#) (const std::string &term, const std::string &colour) const
Return a formatted string that represents a small snippet of text that matched the search from the document.

7.31.1 Detailed Description

A wrapper class that encapsulates a document that is part of the search results returned by Xapian for a query.

Date

Created 2013/01/13 18:41

Copyright

Copyright © 2013, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[SearchResult.h](#) 5567 2013-02-23 22:13:21Z rakesh

7.31.2 Constructor & Destructor Documentation

7.31.2.1 spt::servlet::search::SearchResult::SearchResult (const Xapian::MSetIterator & *iter*)

Create a new instance from the result set iterator.

Parameters

<i>iter</i>	The result set iterator from which the result instance is to be initialised.
-------------	--

7.31.3 Member Function Documentation

7.31.3.1 const DocumentData& spt::servlet::search::SearchResult::getDocument () const [inline]

Returns

const [DocumentData](#) Return the document encapsulated in this result

7.31.3.2 int spt::servlet::search::SearchResult::getPercent () const [inline]

Returns

int Return the percent relevance for the document for the search

7.31.3.3 unsigned spt::servlet::search::SearchResult::getRank () const [inline]

Returns

unsigned Return the ranking for the document for the search

7.31.3.4 std::string spt::servlet::search::SearchResult::getSnippet (const std::string & *term*, const std::string & *colour*) const

Return a formatted string that represents a small snippet of text that matched the search from the document.

Parameters

<i>term</i>	The term to use to try and extract the snippet.
<i>colour</i>	The colour to use to high-light the matching text.

Returns

std::string The snippet of text from the indexed document.

7.31.3.5 `double spt::servlet::search::SearchResult::getWeight ( ) const` `[inline]`

Returns

double Return the weight factor for the document for the search

The documentation for this class was generated from the following file:

- </Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/SearchResult.h>

7.32 spt::servlet::search::SearchResults Class Reference

A wrapper class that encapsulates the search results returned by Xapian for a query.

```
#include <SearchResults.h>
```

Public Member Functions

- [SearchResults](#) (const std::string &query, const Xapian::MSet &results, const unsigned offset=0, const unsigned pageSize=10)
Create a new instance that represents the results for the search for `query`. The result set is limited by the `offset` and `pageSize` parameters.
- const [SearchResult](#) [getSearchResult](#) (const unsigned index) const
Return the search result at the index specified. The index is only for the results represented by this result set and is thus represented by $0 \leq index \leq pageSize$.
- const std::string & [getQuery](#) () const
- const Xapian::MSet & [getResults](#) () const
- unsigned [getOffset](#) () const
- unsigned [getPageSize](#) () const
- unsigned [getTotal](#) () const

7.32.1 Detailed Description

A wrapper class that encapsulates the search results returned by Xapian for a query.

Date

Created 2013/01/15 18:41

Copyright

Copyright © 2013, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[SearchResults.h](#) 5567 2013-02-23 22:13:21Z rakesh

7.32.2 Constructor & Destructor Documentation

7.32.2.1 spt::servlet::search::SearchResults::SearchResults (const std::string & *query*, const Xapian::MSet & *results*, const unsigned *offset* = 0, const unsigned *pageSize* = 10)

Create a new instance that represents the results for the search for *query*. The result set is limited by the *offset* and *pageSize* parameters.

Parameters

<i>query</i>	The search query
<i>results</i>	The iterator into the result set.
<i>offset</i>	The starting offset into the overall result set
<i>pageSize</i>	The total number of results to represent.

7.32.3 Member Function Documentation

7.32.3.1 `unsigned spt::servlet::search::SearchResults::getOffset ( ) const` `[inline]`

Returns

unsigned Return the offset into the entire result set

7.32.3.2 `unsigned spt::servlet::search::SearchResults::getPageSize ( ) const` `[inline]`

Returns

unsigned Return the page size used to restrict this collection

7.32.3.3 `const std::string& spt::servlet::search::SearchResults::getQuery ( ) const` `[inline]`

Returns

const std::string Return the query used for this search

7.32.3.4 `const Xapian::MSet& spt::servlet::search::SearchResults::getResults ( ) const` `[inline]`

Returns

Xapian::MSet Return the original result set.

7.32.3.5 `const SearchResult spt::servlet::search::SearchResults::getSearchResult ( const unsigned index ) const`

Return the search result at the index specified. The index is only for the results represented by this result set and is thus represented by $0 \leq \text{index} \leq \text{pageSize}$.

Parameters

<i>index</i>	The index for the result in this results collection
--------------	---

Returns

const [SearchResult](#) The result at specified index

7.32.3.6 `unsigned spt::servlet::search::SearchResults::getTotal ( ) const` `[inline]`

Returns

unsigned The estimated total number of matching results.

The documentation for this class was generated from the following file:

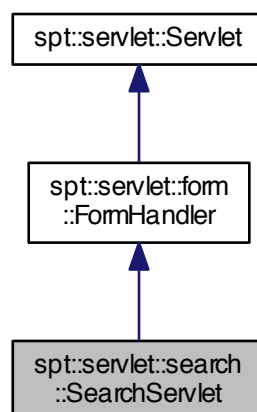
- `/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/SearchResults.h`

7.33 spt::servlet::search::SearchServlet Class Reference

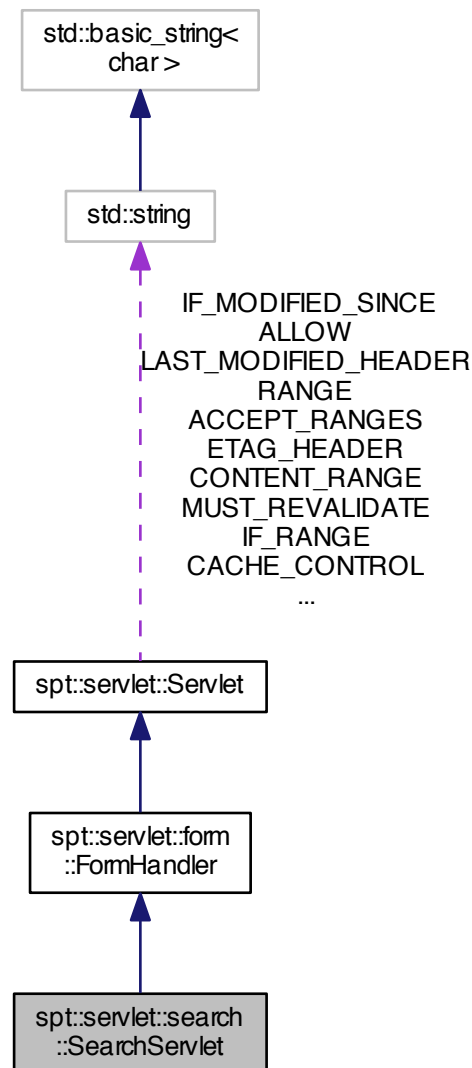
The servlet that handles site-search for each virtual host. Initialises a [DocRootIndexer](#) at start to index all static files. This leads to the issue that pages that are not reachable through the site also gets indexed and served through site search.

```
#include <SearchServlet.h>
```

Inheritance diagram for spt::servlet::search::SearchServlet:



Collaboration diagram for spt::servlet::search::SearchServlet:



Public Member Functions

- `const std::string getServletName () const`
Return a descriptive name for this servlet (useful for logging).
- `void init ()`
Not relevant for this servlet.

Protected Member Functions

- `void handleForm (const Poco::Net::HTMLForm &form, Poco::Net::HTTPServerRequest &req, Poco::Net::HTTPServerResponse &resp)`
Handle form POST with the search query.

Additional Inherited Members

7.33.1 Detailed Description

The servlet that handles site-search for each virtual host. Initialises a [DocRootIndexer](#) at start to index all static files. This leads to the issue that pages that are not reachable through the site also gets indexed and served through site search.

Date

Created 2013/01/25 18:41

Copyright

Copyright © 2013, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[SearchServlet.h](#) 5629 2014-10-01 10:44:20Z rakesh

7.33.2 Member Function Documentation

7.33.2.1 `const std::string spt::servlet::search::SearchServlet::getServletName ( ) const` `[inline], [virtual]`

Return a descriptive name for this servlet (useful for logging).

Returns

`const std::string` The name for this servlet.

Reimplemented from [spt::servlet::Servlet](#).

7.33.2.2 `void spt::servlet::search::SearchServlet::handleForm ( const Poco::Net::HTMLForm & form, Poco::Net::HTTPServerRequest & req, Poco::Net::HTTPServerResponse & resp )` `[protected], [virtual]`

Handle form POST with the search query.

Returns a JSON structure with information about the total number of matches, current offset and pageSize, and a nested array of [SearchResult](#) JSON representations.

Parameters

<i>form</i>	The form data submitted by the client
<i>req</i>	The client request object
<i>resp</i>	The server response object.

Implements [spt::servlet::form::FormHandler](#).

7.33.2.3 `void spt::servlet::search::SearchServlet::init ( ) [inline],[virtual]`

Not relevant for this servlet.

Reimplemented from [spt::servlet::Servlet](#).

The documentation for this class was generated from the following file:

- [/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/SearchServlet.h](#)

7.34 spt::servlet::model::SecurityConstraint Class Reference

A simple value object used to store security constraint declarations in a web.xml file.

```
#include <SecurityConstraint.h>
```

Public Member Functions

- [SecurityConstraint](#) (const [ListPtr](#) pattern, const [ListPtr](#) httpMethods, const [ListPtr](#) constraints)
Create a new instance that represents a security constraint configuration in web.xml file.
- const [ListPtr](#) [getUrlPattern](#) () const
- const [ListPtr](#) [getMethod](#) () const
- const [ListPtr](#) [getAuthConstraint](#) () const

7.34.1 Detailed Description

A simple value object used to store security constraint declarations in a web.xml file.

Date

Created 2011/09/10 18:41

Copyright

Copyright © 2011, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[SecurityConstraint.h](#) 5567 2013-02-23 22:13:21Z rakesh

7.34.2 Constructor & Destructor Documentation

7.34.2.1 `spt::servlet::model::SecurityConstraint::SecurityConstraint ( const ListPtr pattern, const ListPtr httpMethods, const ListPtr constraints ) [inline]`

Create a new instance that represents a security constraint configuration in web.xml file.

Parameters

<i>pattern</i>	The URL pattern to which this configuration applies.
<i>httpMethods</i>	The HTTP methods to which this configuration applies.
<i>constraints</i>	The constraints to be enforced for this configuration.

7.34.3 Member Function Documentation

7.34.3.1 `const ListPtr spt::servlet::model::SecurityConstraint::getAuthConstraint ( ) const` `[inline]`

Returns

ListPtr Return the constraints this configuration applies.

7.34.3.2 `const ListPtr spt::servlet::model::SecurityConstraint::getMethod ( ) const` `[inline]`

Returns

ListPtr Return the http method for which this configuration applies.

7.34.3.3 `const ListPtr spt::servlet::model::SecurityConstraint::getUriPattern ( ) const` `[inline]`

Returns

ListPtr Return the url pattern for which this configuration applies.

The documentation for this class was generated from the following file:

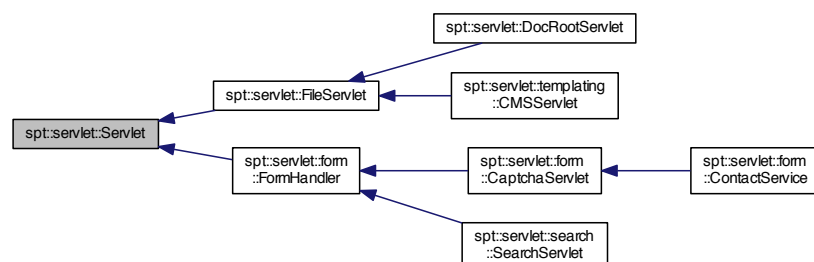
- `/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/SecurityConstraint.h`

7.35 spt::servlet::Servlet Class Reference

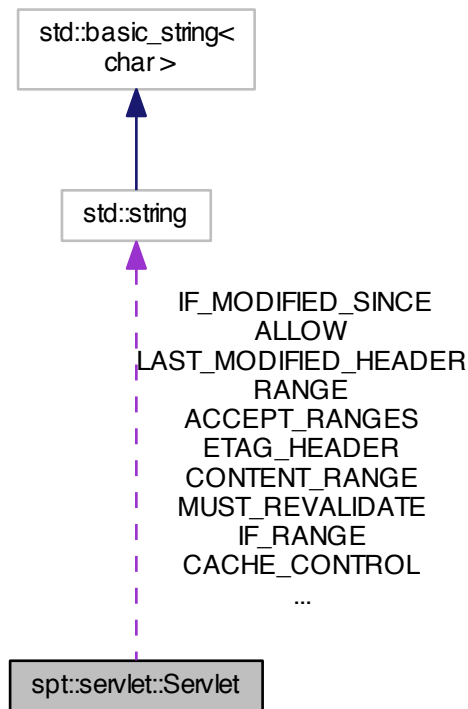
The base servlet class from which all concrete servlet instances are to be derived.

```
#include <Servlet.h>
```

Inheritance diagram for spt::servlet::Servlet:



Collaboration diagram for spt::servlet::Servlet:



Public Member Functions

- [Servlet](#) ()
Default CTOR.
- virtual [~Servlet](#) ()
Virtual DTOR for sub-classes.
- virtual const std::string [getServletName](#) () const
- virtual void [init](#) ()
Initialise the servlet instance. Invoked from [VirtualHostServer](#).
- void [service](#) (Poco::Net::HTTPServerRequest &req, Poco::Net::HTTPServerResponse &resp)
The primary request/response handling method.
- virtual const std::string & [getServerIdentifier](#) () const
- void [setServletConfig](#) (const [model::ServletConfigPtr](#) sc)
Set the servlet configuration information for this instance.
- void [setHostContext](#) (const [model::HostContextPtr](#) hptr)
Set the host context information for this instance.

Static Public Attributes

- static const std::string [ALLOW](#)
The HTTP Allow header.
- static const std::string [ACCEPT_RANGES](#)

- *The HTTP Accept-Ranges header.*
- static const std::string [CACHE_CONTROL](#)
The HTTP Cache-Control directive.
- static const std::string [CONTENT_RANGE](#)
The HTTP Content-Range header.
- static const std::string [ETAG_HEADER](#)
The HTTP ETag header.
- static const std::string [IF_MODIFIED_SINCE](#)
The HTTP If-Modified-Since header.
- static const std::string [IF_RANGE](#)
The HTTP If-Range header.
- static const std::string [LAST_MODIFIED_HEADER](#)
The HTTP Last-Modified header.
- static const std::string [MUST_REVALIDATE](#)
The HTTP must-revalidate header.
- static const std::string [RANGE](#)
The HTTP Range header.

Protected Types

- typedef std::pair
 < Poco::UInt32, Poco::UInt32 > [Range](#)
 Pair of the starting and ending byte range specified in request.

Protected Member Functions

- const [model::ServletConfigPtr](#) [getServletConfig](#) () const
- [model::HostContextPtr](#) [getHostContext](#) ()
- const [model::HostConfigurationPtr](#) [getHostConfiguration](#) ()
- const [model::ContextParams](#) & [getContextParams](#) () const
- const [model::InitParams](#) & [getInitParams](#) () const
- const std::string & [getName](#) () const
- const std::string & [getClassName](#) () const
- void [log](#) (const std::string &msg)
 Log the specified information message to the [getLogger](#).
- void [log](#) (const std::string &className, const std::string &method, const std::string &msg)
 Log the specified information message to the [getLogger](#) Prefix the message with the `className` and `method`.
- void [sendErrorResponse](#) (Poco::Net::HTTPResponse::HTTPStatus status, const std::string &message, Poco::Net::HTTPServerResponse &resp)
 Send an error response with the specified status and explanation message to the response.
- virtual bool [authenticate](#) (const Poco::Net::HTTPServerRequest &req)
- bool [requiresAuthentication](#) (const Poco::Net::HTTPServerRequest &req)
 Checks rules in web.xml to determine if URI is protected.
- virtual const std::string * [getRealm](#) () const
 Sub-classes that need to specify custom realms must over-ride. The default implementation uses the realm configured under the `login-config` element in web.xml.
- virtual void [doGet](#) (Poco::Net::HTTPServerRequest &req, Poco::Net::HTTPServerResponse &resp)
 Handle a GET HTTP request from a client.
- virtual void [doPost](#) (Poco::Net::HTTPServerRequest &req, Poco::Net::HTTPServerResponse &resp)
 Handle a POST HTTP request from a client.
- virtual void [doHead](#) (Poco::Net::HTTPServerRequest &req, Poco::Net::HTTPServerResponse &resp)

- Handle a HEAD HTTP request from a client.*
- virtual void [doOptions](#) (Poco::Net::HTTPServerRequest &req, Poco::Net::HTTPServerResponse &resp)
- Handle a OPTIONS HTTP request from a client.*
- virtual void [doTrace](#) (Poco::Net::HTTPServerRequest &req, Poco::Net::HTTPServerResponse &resp)
- Handle a TRACE HTTP request from a client.*
- virtual void [doPut](#) (Poco::Net::HTTPServerRequest &req, Poco::Net::HTTPServerResponse &resp)
- Handle a PUT HTTP request from a client.*
- virtual void [doDelete](#) (Poco::Net::HTTPServerRequest &req, Poco::Net::HTTPServerResponse &resp)
- Handle a DELETE HTTP request from a client.*
- bool [gzipResponse](#) (const Poco::Net::HTTPServerRequest &req) const
- Check to see if request specified gzip compressed response.*
- bool [deflateResponse](#) (const Poco::Net::HTTPServerRequest &req) const
- Check to see if request specified deflate compressed response.*
- bool [rangeRequest](#) (const Poco::Net::HTTPServerRequest &req) const
- Check to see if request specified a byte range from full data.*
- const [Range](#) [getRequestedRange](#) (const Poco::Net::HTTPServerRequest &req) const
- Return the byte range specified in the request.*
- virtual bool [checkURI](#) (Poco::Net::HTTPServerRequest &req, Poco::Net::HTTPServerResponse &resp)
- Check the URI specified in client request to ensure it is valid. Validity checks try to protect against common attacks attempted by malicious scripts and hackers.*
- const std::string [getContentType](#) (const std::string &path) const
- Return the mime-type for the resource at specified path.*
- virtual Poco::Logger & [getLogger](#) ()
- const std::string & [stringValue](#) (const std::string &key, const std::string &defaultValue=std::string()) const
- Return value of init or context param with specified key.*
- int [intValue](#) (const std::string &key) const
- Return int value of init or context param with specified key.*
- bool [booleanValue](#) (const std::string &key) const
- Return value of init or context param with specified key.*
- [model::ServletSecurityPtr](#) [securityForPath](#) (const std::string &url)
- Attempt to return ServletSecurity for path. Designed to be invoked from [servletSecurity\(const string& \)](#).*
- [model::ServletSecurityPtr](#) [servletSecurity](#) (const std::string &url)
- Return the servlet security configuration if any for request path.*
- const std::string & [getDocRoot](#) () const

7.35.1 Detailed Description

The base servlet class from which all concrete servlet instances are to be derived.

Base servlet that handles various types of HTTP requests and delegates them to the appropriate doXxx method. The base implementation just sends a HTTP_NOT_IMPLEMENTED error. Sub-classes must implement the appropriate doXxx methods.

Date

Created 2011/08/18 18:41

Copyright

Copyright © 2011, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[Servlet.h](#) 5629 2014-10-01 10:44:20Z rakesh

7.35.2 Member Typedef Documentation

7.35.2.1 `typedef std::pair<Poco::UInt32,Poco::UInt32> spt::servlet::Servlet::Range` `[protected]`

Pair of the starting and ending byte range specified in request.

7.35.3 Constructor & Destructor Documentation

7.35.3.1 `spt::servlet::Servlet::Servlet ( )`

Default CTOR.

7.35.3.2 `virtual spt::servlet::Servlet::~~Servlet ( )` `[inline],[virtual]`

Virtual DTOR for sub-classes.

7.35.4 Member Function Documentation

7.35.4.1 `virtual bool spt::servlet::Servlet::authenticate ( const Poco::Net::HTTPServerRequest & req )` `[protected],[virtual]`

Sub-classes that need custom authentication protocols must over-ride. Uses [requiresAuthentication](#) to determine whether the requested resource is protected or not. Default implementation uses user and role information in a users.xml file (similar to tomcat-users.xml) that is specified when the server is started.

7.35.4.2 `bool spt::servlet::Servlet::booleanValue ( const std::string & key ) const` `[protected]`

Return value of init or context param with specified key.

Parameters

<i>key</i>	The property name/key to look up.
------------	-----------------------------------

Returns

`bool` Return `true` if found and has value `"true"`

7.35.4.3 `virtual bool spt::servlet::Servlet::checkURI ( Poco::Net::HTTPServerRequest & req, Poco::Net::HTTPServerResponse & resp )` `[protected],[virtual]`

Check the URI specified in client request to ensure it is valid. Validity checks try to protect against common attacks attempted by malicious scripts and hackers.

Parameters

<i>req</i>	The client request object to check
<i>resp</i>	The server response object

Returns

bool Return `true` if the request URI is clean.

Reimplemented in [spt::servlet::FileServlet](#).

7.35.4.4 `bool spt::servlet::Servlet::deflateResponse ( const Poco::Net::HTTPServerRequest & req ) const` `[protected]`

Check to see if request specified deflate compressed response.

Parameters

<i>req</i>	The client request object
------------	---------------------------

Returns

bool Return `true` if Accept-Encoding: deflate was specified.

7.35.4.5 `virtual void spt::servlet::Servlet::doDelete ( Poco::Net::HTTPServerRequest & req, Poco::Net::HTTPServerResponse & resp )` `[protected]`, `[virtual]`

Handle a `DELETE` HTTP request from a client.

Parameters

<i>req</i>	The client request object
<i>resp</i>	The server response object

7.35.4.6 `virtual void spt::servlet::Servlet::doGet ( Poco::Net::HTTPServerRequest & req, Poco::Net::HTTPServerResponse & resp )` `[protected]`, `[virtual]`

Handle a `GET` HTTP request from a client.

Parameters

<i>req</i>	The client request object
<i>resp</i>	The server response object

Reimplemented in [spt::servlet::FileServlet](#).

7.35.4.7 `virtual void spt::servlet::Servlet::doHead ( Poco::Net::HTTPServerRequest & req, Poco::Net::HTTPServerResponse & resp )` `[protected]`, `[virtual]`

Handle a `HEAD` HTTP request from a client.

Parameters

<i>req</i>	The client request object
------------	---------------------------

<i>resp</i>	The server response object
-------------	----------------------------

Reimplemented in [spt::servlet::FileServlet](#).

7.35.4.8 `virtual void spt::servlet::Servlet::doOptions ( Poco::Net::HTTPServerRequest & req, Poco::Net::HTTPServerResponse & resp )` `[protected]`, `[virtual]`

Handle a `OPTIONS` HTTP request from a client.

Parameters

<i>req</i>	The client request object
<i>resp</i>	The server response object

Reimplemented in [spt::servlet::form::FormHandler](#), and [spt::servlet::FileServlet](#).

7.35.4.9 `virtual void spt::servlet::Servlet::doPost ( Poco::Net::HTTPServerRequest & req, Poco::Net::HTTPServerResponse & resp )` `[protected]`, `[virtual]`

Handle a `POST` HTTP request from a client.

Parameters

<i>req</i>	The client request object
<i>resp</i>	The server response object

Reimplemented in [spt::servlet::form::FormHandler](#).

7.35.4.10 `virtual void spt::servlet::Servlet::doPut ( Poco::Net::HTTPServerRequest & req, Poco::Net::HTTPServerResponse & resp )` `[protected]`, `[virtual]`

Handle a `PUT` HTTP request from a client.

Parameters

<i>req</i>	The client request object
<i>resp</i>	The server response object

7.35.4.11 `virtual void spt::servlet::Servlet::doTrace ( Poco::Net::HTTPServerRequest & req, Poco::Net::HTTPServerResponse & resp )` `[protected]`, `[virtual]`

Handle a `TRACE` HTTP request from a client.

Parameters

<i>req</i>	The client request object
<i>resp</i>	The server response object

7.35.4.12 `const std::string& spt::servlet::Servlet::getClassName ( ) const` `[inline]`, `[protected]`

Returns

`std::string` Return the fully qualified class name for this servlet

7.35.4.13 `const std::string spt::servlet::Servlet::getContentType ( const std::string & path ) const` `[protected]`

Return the mime-type for the resource at specified path.

Parameters

<i>path</i>	The path to the file/resource.
-------------	--------------------------------

Returns

const std::string The mime type for the contents.

7.35.4.14 **const model::ContextParams& spt::servlet::Servlet::getContextParams () const** [inline], [protected]

Returns

[model::ContextParams](#) Return the context parameters for virtual host

7.35.4.15 **const std::string& spt::servlet::Servlet::getDocRoot () const** [protected]

Returns

const std::string Return the document root configured for the virtual host

7.35.4.16 **const model::HostConfigurationPtr spt::servlet::Servlet::getHostConfiguration ()** [inline], [protected]

Returns

[model::HostConfigurationPtr](#) Return the host configuration used.

7.35.4.17 **model::HostContextPtr spt::servlet::Servlet::getHostContext ()** [inline], [protected]

Returns

[model::HostContextPtr](#) Return the host context used.

7.35.4.18 **const model::InitParams& spt::servlet::Servlet::getInitParams () const** [inline], [protected]

Returns

[model::InitParams](#) Return the init parameters for the servlet

7.35.4.19 **virtual Poco::Logger& spt::servlet::Servlet::getLogger ()** [inline], [protected], [virtual]

Returns

Poco::Logger Return the logger used by this servlet.

Reimplemented in [spt::servlet::templating::CMSServlet](#), and [spt::servlet::DocRootServlet](#).

7.35.4.20 **const std::string& spt::servlet::Servlet::getName () const** [inline], [protected]

Returns

const std::string Return the name for this servlet as configured.

7.35.4.21 `const std::string * spt::servlet::Servlet::getRealm ( ) const` `[inline], [protected], [virtual]`

Sub-classes that need to specify custom realms must over-ride. The default implementation uses the realm configured under the login-config element in web.xml.

Returns

`const std::string` Return the realm to associate the authentication with.

7.35.4.22 `const Range spt::servlet::Servlet::getRequestedRange ( const Poco::Net::HTTPServerRequest & req ) const` `[protected]`

Return the byte range specified in the request.

Parameters

<i>req</i>	The client request object
------------	---------------------------

Returns

`const Range` The pair of start and end bytes specified in request.

7.35.4.23 `virtual const std::string& spt::servlet::Servlet::getServerIdentifier ( ) const` `[inline], [virtual]`

Returns

`const std::string` Return the server identifier response header.

7.35.4.24 `const model::ServletConfigPtr spt::servlet::Servlet::getServletConfig ( ) const` `[inline], [protected]`

Returns

[model::ServletConfigPtr](#) Return the servlet configuration used.

7.35.4.25 `virtual const std::string spt::servlet::Servlet::getServletName ( ) const` `[inline], [virtual]`

Returns

`std::string` Return a descriptive name of servlet (mainly for logging)

Reimplemented in [spt::servlet::templating::CMSServlet](#), [spt::servlet::form::ContactService](#), [spt::servlet::DocRootServlet](#), and [spt::servlet::search::SearchServlet](#).

7.35.4.26 `bool spt::servlet::Servlet::gzipResponse ( const Poco::Net::HTTPServerRequest & req ) const` `[protected]`

Check to see if request specified gzip compressed response.

Parameters

<i>req</i>	The client request object
------------	---------------------------

Returns

`bool` Return `true` if Accept-Encoding: gzip was specified.

7.35.4.27 `virtual void spt::servlet::Servlet::init ( ) [inline],[virtual]`

Initialise the servlet instance. Invoked from [VirtualHostServer](#).

Reimplemented in [spt::servlet::templating::CMSServlet](#), [spt::servlet::search::SearchServlet](#), and [spt::servlet::DocRootServlet](#).

7.35.4.28 `int spt::servlet::Servlet::intValue ( const std::string & key ) const [protected]`

Return int value of init or context param with specified key.

Parameters

<i>key</i>	The property name/key to look up.
------------	-----------------------------------

Returns

int The configured value or returns the value of `string::npos`

7.35.4.29 `void spt::servlet::Servlet::log ( const std::string & msg ) [inline],[protected]`

Log the specified information message to the [getLogger](#).

Parameters

<i>msg</i>	The message to log with information priority.
------------	---

7.35.4.30 `void spt::servlet::Servlet::log ( const std::string & className, const std::string & method, const std::string & msg ) [inline],[protected]`

Log the specified information message to the [getLogger](#) Prefix the message with the `className` and `method`.

Parameters

<i>className</i>	The class name to use for the log message.
<i>method</i>	The method name to use for the log message.
<i>msg</i>	The information message to log.

7.35.4.31 `bool spt::servlet::Servlet::rangeRequest ( const Poco::Net::HTTPServerRequest & req ) const [protected]`

Check to see if request specified a byte range from full data.

Parameters

<i>req</i>	The client request object
------------	---------------------------

Returns

bool Return `true` if Range was specified in request headers.

7.35.4.32 `bool spt::servlet::Servlet::requiresAuthentication ( const Poco::Net::HTTPServerRequest & req ) [protected]`

Checks rules in web.xml to determine if URI is protected.

Parameters

<i>req</i>	The client request object to check
------------	------------------------------------

Returns

bool Return `true` if the URI requested was configured to require authentication.

7.35.4.33 model::ServletSecurityPtr spt::servlet::Servlet::securityForPath (const std::string & *url*) [protected]

Attempt to return ServletSecurity for path. Designed to be invoked from [servletSecurity\(const string& \)](#).

Parameters

<i>url</i>	The URL which is to be checked
------------	--------------------------------

Returns

[model::ServletSecurityPtr](#) Return the security configuration if found, or an empty shared pointer.

7.35.4.34 void spt::servlet::Servlet::sendErrorResponse (Poco::Net::HTTPResponse::HTTPStatus *status*, const std::string & *message*, Poco::Net::HTTPServerResponse & *resp*) [protected]

Send an error response with the specified status and explanation message to the response.

Parameters

<i>status</i>	The HTTP response status code.
<i>message</i>	The explanation for the error status.
<i>resp</i>	The server response object.

7.35.4.35 void spt::servlet::Servlet::service (Poco::Net::HTTPServerRequest & *req*, Poco::Net::HTTPServerResponse & *resp*)

The primary request/response handling method.

Performs various checks on the request URI, request headers, authentication configuration etc. before handing over the the `run` method which delegates the the `doXYZ` method based on HTTP request method.

Parameters

<i>req</i>	The client request object
<i>resp</i>	The server response object

7.35.4.36 model::ServletSecurityPtr spt::servlet::Servlet::servletSecurity (const std::string & *url*) [protected]

Return the servlet security configuration if any for request path.

Parameters

<i>url</i>	The URL which is to be checked.
------------	---------------------------------

Returns

[model::ServletSecurityPtr](#) Return the security configuration if found, or an empty shared pointer.

7.35.4.37 `void spt::servlet::Servlet::setHostContext ( const model::HostContextPtr hptr )` `[inline]`

Set the host context information for this instance.

Parameters

<i>hptr</i>	The host configuration information.
-------------	-------------------------------------

7.35.4.38 void spt::servlet::Servlet::setServletConfig (const model::ServletConfigPtr *sc*) [inline]

Set the servlet configuration information for this instance.

Parameters

<i>sc</i>	The servlet configuration information.
-----------	--

7.35.4.39 const std::string& spt::servlet::Servlet::stringValue (const std::string & *key*, const std::string & *defaultValue* = std::string()) const [protected]

Return value of init or context param with specified key.

Parameters

<i>key</i>	The property name/key to look up.
<i>defaultValue</i>	The default value to return.

Returns

const std::string The configured value or default value if not found.

7.35.5 Member Data Documentation

7.35.5.1 const std::string spt::servlet::Servlet::ACCEPT_RANGES [static]

The HTTP Accept-Ranges header.

7.35.5.2 const std::string spt::servlet::Servlet::ALLOW [static]

The HTTP Allow header.

7.35.5.3 const std::string spt::servlet::Servlet::CACHE_CONTROL [static]

The HTTP Cache-Control directive.

7.35.5.4 const std::string spt::servlet::Servlet::CONTENT_RANGE [static]

The HTTP Content-Range header.

7.35.5.5 const std::string spt::servlet::Servlet::ETAG_HEADER [static]

The HTTP ETag header.

7.35.5.6 const std::string spt::servlet::Servlet::IF_MODIFIED_SINCE [static]

The HTTP If-Modified-Since header.

7.35.5.7 `const std::string spt::servlet::Servlet::IF_RANGE` `[static]`

The HTTP If-Range header.

7.35.5.8 `const std::string spt::servlet::Servlet::LAST_MODIFIED_HEADER` `[static]`

The HTTP Last-Modified header.

7.35.5.9 `const std::string spt::servlet::Servlet::MUST_REVALIDATE` `[static]`

The HTTP must-revalidate header.

7.35.5.10 `const std::string spt::servlet::Servlet::RANGE` `[static]`

The HTTP Range header.

The documentation for this class was generated from the following file:

- `/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/Servlet.h`

7.36 spt::servlet::model::ServletConfig Class Reference

A simple value object that represents a servlet configuration in a web.xml file. This object stores the information from the servlet and servlet-mapping elements.

```
#include <ServletConfig.h>
```

Public Member Functions

- [ServletConfig](#) (const std::string &name, const std::string &class, const [ContextParams](#) &cparams, const [InitParams](#) &iparams, const [PathMapping](#) &mapping)
Create a new instance with the specified values for the servlet identified by the `sname`.
- const std::string & [getServletName](#) () const
- const std::string & [getServletClass](#) () const
- const [ContextParams](#) & [getContextParams](#) () const
- const [InitParams](#) & [getInitParams](#) () const
- const [PathMapping](#) & [getPathMapping](#) () const

7.36.1 Detailed Description

A simple value object that represents a servlet configuration in a web.xml file. This object stores the information from the servlet and servlet-mapping elements.

Date

Created 2011/09/07 18:41

Copyright

Copyright © 2011, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[ServletConfig.h](#) 5567 2013-02-23 22:13:21Z rakesh

7.36.2 Constructor & Destructor Documentation

7.36.2.1 `spt::servlet::model::ServletConfig ( const std::string & sname, const std::string & sclass, const ContextParams & cparams, const InitParams & iparams, const PathMapping & mapping )` `[inline]`

Create a new instance with the specified values for the servlet identified by the *sname*.

Parameters

<i>sname</i>	The name used to identify the servlet.
<i>sclass</i>	The fully-qualified class name of the servlet.
<i>cparams</i>	The context parameter values configured in <code>web.xml</code>
<i>iparams</i>	The init-param values for the servlet.
<i>mapping</i>	The paths this servlet instance handles.

7.36.3 Member Function Documentation

7.36.3.1 `const ContextParams& spt::servlet::model::ServletConfig::getContextParams ( ) const` `[inline]`

Returns

const ContextParams The context param values for the host

7.36.3.2 `const InitParams& spt::servlet::model::ServletConfig::getInitParams ( ) const` `[inline]`

Returns

const InitParams The init-param values for the servlet

7.36.3.3 `const PathMapping& spt::servlet::model::ServletConfig::getPathMapping ( ) const` `[inline]`

Returns

const PathMapping The paths the servlet handles

7.36.3.4 `const std::string& spt::servlet::model::ServletConfig::getServletClass ( ) const` `[inline]`

Returns

const std::string The fully qualified class of the servlet.

7.36.3.5 `const std::string& spt::servlet::model::ServletConfig::getServletName ( ) const` `[inline]`

Returns

`const std::string` The name of the servlet.

The documentation for this class was generated from the following file:

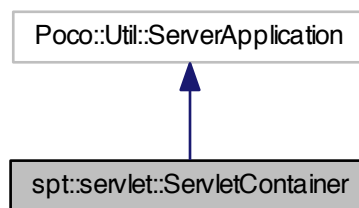
- `/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/ServletConfig.h`

7.37 spt::servlet::ServletContainer Class Reference

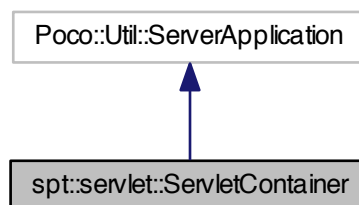
The main servlet container implementation. Loads the properties file and web.xml file from directory specified in constructor. Hands off to [RequestDispatcher](#) to perform the actual request handling.

```
#include <ServletContainer.h>
```

Inheritance diagram for `spt::servlet::ServletContainer`:



Collaboration diagram for `spt::servlet::ServletContainer`:



Public Member Functions

- [ServletContainer](#) (`const std::string &directory`)

Create a new servlet container instance using the configuration file `minnal.xml` in the specified directory.

Protected Member Functions

- void [initialize](#) (Poco::Util::Application &self)
Load the `minnal.xml` file and create a new `model::ContainerConfiguration` with the configuration data. Initialises the logging sub-system through the `LogManager` singleton.
- void [defineOptions](#) (Poco::Util::OptionSet &options)
Set up the command line options supported by this application.
- void [handleOption](#) (const std::string &name, const std::string &value)
Over-ridden to handle the `help` command line option.
- int [main](#) (const std::vector< std::string > &args)
The main entry point into the server application.

7.37.1 Detailed Description

The main servlet container implementation. Loads the properties file and web.xml file from directory specified in constructor. Hands off to [RequestDispatcher](#) to perform the actual request handling.

Date

Created 2011/08/16 18:41

Copyright

Copyright © 2011, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[ServletContainer.h](#) 5567 2013-02-23 22:13:21Z rakesh

7.37.2 Constructor & Destructor Documentation

7.37.2.1 spt::servlet::ServletContainer::ServletContainer (const std::string & directory)

Create a new servlet container instance using the configuration file `minnal.xml` in the specified directory.

Parameters

<i>directory</i>	The configuration directory for the server.
------------------	---

7.37.3 Member Function Documentation

7.37.3.1 void spt::servlet::ServletContainer::defineOptions (Poco::Util::OptionSet & options) [protected]

Set up the command line options supported by this application.

Parameters

<i>options</i>	The collection of options to append to.
----------------	---

7.37.3.2 `void spt::servlet::ServletContainer::handleOption ( const std::string & name, const std::string & value )`
[protected]

Over-ridden to handle the `help` command line option.

Parameters

<i>name</i>	The name of the command line option specified
<i>value</i>	The value for the option.

7.37.3.3 `void spt::servlet::ServletContainer::initialize ( Poco::Util::Application & self )` [protected]

Load the `minnal.xml` file and create a new [model::ContainerConfiguration](#) with the configuration data. Initialises the logging sub-system through the [LogManager](#) singleton.

Parameters

<i>self</i>	The server application instance.
-------------	----------------------------------

7.37.3.4 `int spt::servlet::ServletContainer::main ( const std::vector< std::string > & args )` [protected]

The main entry point into the server application.

Loads the configuration and binds to the `port` configured. Sets up the thread pool with number of threads and maximum queue sizes. Switches the process user if configured. Starts the `Poco::Net::HTTPServer` process after all configuration has been performed.

Parameters

<i>args</i>	The command line argument vector.
-------------	-----------------------------------

Returns

`int` Return the exit status of the process.

The documentation for this class was generated from the following file:

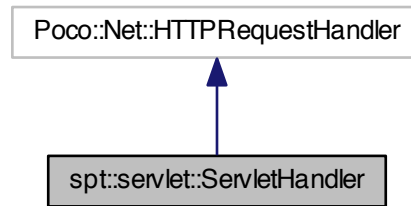
- `/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/ServletContainer.h`

7.38 spt::servlet::ServletHandler Class Reference

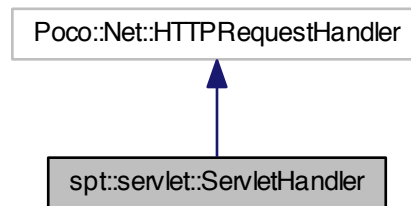
The temporary per request handler instance that is required by the [Poco](#) handler factory.

```
#include <ServletHandler.h>
```


Inheritance diagram for spt::servlet::ServletHandler:



Collaboration diagram for spt::servlet::ServletHandler:



Public Member Functions

- [ServletHandler](#) ([ServletPtr](#) srv)
Create a new instance using the specified servlet instance to perform the request handling.
- virtual [~ServletHandler](#) ()
Delete the `HTMLForm` instance if initialised.
- void [handleRequest](#) (Poco::Net::HTTPServerRequest &req, Poco::Net::HTTPServerResponse &resp)
Handle a client request. Delegate to the [Servlet#service](#) method of the servlet used to initialise this instance.
- Poco::Net::HTTPServerRequest & [getRequest](#) ()
- Poco::Net::HTTPServerResponse & [getResponse](#) ()
- Poco::Net::HTMLForm & [getForm](#) ()

7.38.1 Detailed Description

The temporary per request handler instance that is required by the [Poco](#) handler factory.

A Poco::Net::HTTPRequestHandler that delegates to the servlet instance specified in the CTOR. This class serves as the temporary handler instance that the [Poco](#) handler factory deletes after each request. The [Servlet](#) instance itself is initialised only once for the lifetime of the server in the same way JEE containers do.

Date

Created 2011/08/17 18:41

Copyright

Copyright © 2011, Sans Pareil Technologies, Inc.

Author

Rakesh

Version**Id**

[ServletHandler.h](#) 5567 2013-02-23 22:13:21Z rakesh

7.38.2 Constructor & Destructor Documentation**7.38.2.1 spt::servlet::ServletHandler::ServletHandler (ServletPtr *srv*)**

Create a new instance using the specified servlet instance to perform the request handling.

Parameters

<i>srv</i>	The servlet that will process the request.
------------	--

7.38.2.2 virtual spt::servlet::ServletHandler::~~ServletHandler () [inline],[virtual]

Delete the `HTMLForm` instance if initialised.

7.38.3 Member Function Documentation**7.38.3.1 Poco::Net::HTMLForm& spt::servlet::ServletHandler::getForm ()****Returns**

Poco::Net::HTMLForm The form object from the client request.

7.38.3.2 Poco::Net::HTTPServerRequest & spt::servlet::ServletHandler::getRequest () [inline]**Returns**

Poco::Net::HTTPServerRequest The client request object

7.38.3.3 Poco::Net::HTTPServerResponse & spt::servlet::ServletHandler::getResponse () [inline]**Returns**

Poco::Net::HTTPServerResponse The server response object

7.38.3.4 void spt::servlet::ServletHandler::handleRequest (Poco::Net::HTTPServerRequest & *req*,
Poco::Net::HTTPServerResponse & *resp*)

Handle a client request. Delegate to the [Servlet#service](#) method of the servlet used to initialise this instance.

Parameters

<i>req</i>	The client request object
<i>resp</i>	The server response object

The documentation for this class was generated from the following file:

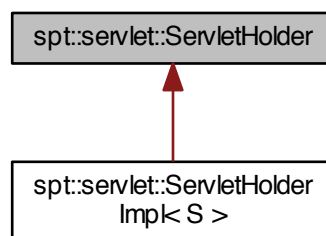
- /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/[ServletHandler.h](#)

7.39 spt::servlet::ServletHolder Struct Reference

A base holder for creating servlet instances.

```
#include <ServletRegistry.h>
```

Inheritance diagram for spt::servlet::ServletHolder:



Public Member Functions

- virtual [~ServletHolder](#) ()
Virtual DTOR for sub-classes.
- virtual Poco::SharedPtr< [Servlet](#) > [create](#) ()=0
Create a new instance of the servlet held in this instance.

7.39.1 Detailed Description

A base holder for creating servlet instances.

7.39.2 Constructor & Destructor Documentation

7.39.2.1 virtual spt::servlet::ServletHolder::~~ServletHolder () [inline],[virtual]

Virtual DTOR for sub-classes.

7.39.3 Member Function Documentation

7.39.3.1 virtual Poco::SharedPtr<[Servlet](#)> spt::servlet::ServletHolder::create () [pure virtual]

Create a new instance of the servlet held in this instance.

Returns

Poco::SharedPtr<Servlet> The shared pointer to newly created servlet instance.

Implemented in [spt::servlet::ServletHolderImpl< S >](#).

The documentation for this struct was generated from the following file:

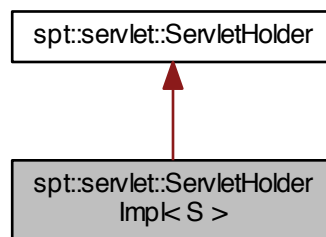
- /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/[ServletRegistry.h](#)

7.40 spt::servlet::ServletHolderImpl< S > Class Template Reference

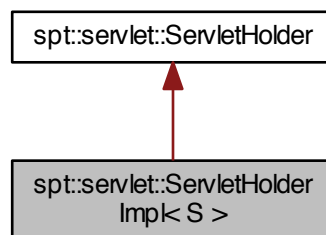
A template class that is used to register a servlet class. Used by the [REGISTER_SERVLET](#) macro to create a new instance and register by class name.

```
#include <ServletRegistry.h>
```

Inheritance diagram for spt::servlet::ServletHolderImpl< S >:



Collaboration diagram for spt::servlet::ServletHolderImpl< S >:

**Public Member Functions**

- [ServletHolderImpl](#) (const std::string &className)

Create a new instance that holds an instance of a servlet with the specified class name.

- [~ServletHolderImpl](#) ()
De-register the held servlet.
- Poco::SharedPtr< [Servlet](#) > [create](#) ()
*Create a new instance of the servlet *held* in this instance.*
- const std::string & [getName](#) () const

7.40.1 Detailed Description

```
template<typename S>class spt::servlet::ServletHolderImpl< S >
```

A template class that is used to register a servlet class. Used by the [REGISTER_SERVLET](#) macro to create a new instance and register by class name.

Template Parameters

S	The servlet that is held in this implementation
-------------------	---

Date

Created 2013/01/10 18:41

Copyright

Copyright © 2013, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[ServletRegistry.h](#) 5567 2013-02-23 22:13:21Z rakesh

7.40.2 Constructor & Destructor Documentation

7.40.2.1 `template<typename S> spt::servlet::ServletHolderImpl< S >::ServletHolderImpl ( const std::string & className ) [inline]`

Create a new instance that holds an instance of a servlet with the specified class name.

Parameters

<i>className</i>	The fully qualified name of the held servlet.
------------------	---

7.40.2.2 `template<typename S> spt::servlet::ServletHolderImpl< S >::~ServletHolderImpl ( ) [inline]`

De-register the held servlet.

7.40.3 Member Function Documentation

7.40.3.1 `template<typename S > Poco::SharedPtr<Servlet> spt::servlet::ServletHolderImpl< S >::create ( )`
`[inline], [virtual]`

Create a new instance of the servlet held in this instance.

Returns

Poco::SharedPtr<Servlet> The shared pointer to newly created servlet instance.

Implements [spt::servlet::ServletHolder](#).

7.40.3.2 `template<typename S > const std::string& spt::servlet::ServletHolderImpl< S >::getName ( ) const`
`[inline]`

Returns

const std::string Return the held servlet class name.

The documentation for this class was generated from the following file:

- /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/[ServletRegistry.h](#)

7.41 spt::servlet::model::ServletSecurity Class Reference

A simple value object that holds the security configuration for a path that has been mapped to a servlet.

```
#include <ServletSecurity.h>
```

Public Member Functions

- [ServletSecurity](#) (const std::string *url, const [ListPtr](#) methods, const [ListPtr](#) constraints)
Create a new instance for the specified url.
- const std::string * [getPath](#) () const
- const [ListPtr](#) [getMethod](#) () const
- const [ListPtr](#) [getAuthConstraint](#) () const

7.41.1 Detailed Description

A simple value object that holds the security configuration for a path that has been mapped to a servlet.

Date

Created 2011/09/04 18:41

Copyright

Copyright © 2011, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[ServletSecurity.h](#) 5567 2013-02-23 22:13:21Z rakesh

7.41.2 Constructor & Destructor Documentation

7.41.2.1 `spt::servlet::model::ServletSecurity ( const std::string * url, const ListPtr methods, const ListPtr constraints ) [inline]`

Create a new instance for the specified url.

Parameters

<i>url</i>	The url to which this security configuration applies.
<i>methods</i>	The HTTP methods for which this configuration applies.
<i>constraints</i>	The constraints to apply for the url/method combination.

7.41.3 Member Function Documentation

7.41.3.1 `const ListPtr spt::servlet::model::ServletSecurity::getAuthConstraint ( ) const [inline]`

Returns

const ListPtr Return constraints the configuration applies

7.41.3.2 `const ListPtr spt::servlet::model::ServletSecurity::getMethod ( ) const [inline]`

Returns

const ListPtr Return HTTP methods for which the configuration applies

7.41.3.3 `const std::string* spt::servlet::model::ServletSecurity::getPath ( ) const [inline]`

Returns

const std::string Return the path for which the configuration applies.

The documentation for this class was generated from the following file:

- /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/[ServletSecurity.h](#)

7.42 spt::servlet::templating::TemplateManager Class Reference

A manager class used to abstract interactions with the templating engine.

```
#include <TemplateManager.h>
```

Public Member Functions

- void `init (spt::servlet::model::ServerConfigurationPtr configuration)`
Initialise the template engine with configuration from the server configuration file (*minnal.xml*).

- void [setInterval](#) (long interval)
- void [addDirectory](#) (const std::string &path)
Add the specified directory to the template engine search path.
- std::string [expand](#) (const std::string &templateName, ctemplate::TemplateDictionary *data)
Convenience method used to return the expanded representation of the specified template using the specified data dictionary.
- void [reload](#) (Poco::Timer &timer)
Callback method invoked by the timer after the timer interval has expired.
- std::string [getFilePath](#) (const std::string &templateName)
Return the filesystem path for the template by searching within the template directory search path(s).

Static Public Member Functions

- static [TemplateManager](#) & [getTemplateManager](#) ()

Protected Member Functions

- ctemplate::TemplateCache * [getCache](#) ()

7.42.1 Detailed Description

A manager class used to abstract interactions with the templating engine.

Minnal uses Google ctemplate as the templating engine. This class exists primarily to run a timer that signals the template engine to reload any templates modified on disk.

Date

Created 2014/09/29 21:50:41

Copyright

Copyright © 2014, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[TemplateManager.h](#) 5632 2014-10-06 22:04:21Z rakesh

Since

1.6.3

7.42.2 Member Function Documentation

7.42.2.1 void spt::servlet::templating::TemplateManager::addDirectory (const std::string & path)

Add the specified directory to the template engine search path.

The first time this method is invoked, sets the specified directory as the template engine root directory. Additional invocations will add the specified directory as alternate root directories in which templates are stored.

Parameters

<i>path</i>	The path from which templates are to be loaded.
-------------	---

7.42.2.2 `std::string spt::servlet::templating::TemplateManager::expand (const std::string & templateName,
ctemplate::TemplateDictionary * data)`

Convenience method used to return the expanded representation of the specified template using the specified data dictionary.

Parameters

<i>templateName</i>	The name of the template that is to be loaded and expanded by the template engine.
<i>data</i>	The data dictionary to use to expand the template.

Returns

The string that represents the expanded template.

7.42.2.3 `ctemplate::TemplateCache* spt::servlet::templating::TemplateManager::getCache ( )` [protected]

7.42.2.4 `std::string spt::servlet::templating::TemplateManager::getFilePath ( const std::string & templateName )`

Return the filesystem path for the template by searching within the template directory search path(s).

Parameters

<i>templateName</i>	The name of the template as specified to the template engine.
---------------------	---

Returns

The path to the location of the template

7.42.2.5 `static TemplateManager& spt::servlet::templating::TemplateManager::getTemplateManager ( )` [static]

Returns

Returns the singleton instance of the template manager.

7.42.2.6 `void spt::servlet::templating::TemplateManager::init (spt::servlet::model::ServerConfigurationPtr
configuration)`

Initialise the template engine with configuration from the server configuration file (`minnal.xml`).

Parameters

<i>configuration</i>	The server configuration information.
----------------------	---------------------------------------

7.42.2.7 `void spt::servlet::templating::TemplateManager::reload ( Poco::Timer & timer )`

Callback method invoked by the timer after the timer interval has expired.

Parameters

<i>timer</i>	The timer instance that invoked the callback
--------------	--

7.42.2.8 void spt::servlet::templating::TemplateManager::setInterval (long *interval*) [inline]

Parameters

<i>interval</i>	The periodic interval at which the timer is to fire
-----------------	---

The documentation for this class was generated from the following file:

- /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/templating/[TemplateManager.h](#)

7.43 spt::servlet::model::User Class Reference

A simple value object that represents a user configured in the users.xml file.

```
#include <User.h>
```

Public Member Functions

- [User](#) (const std::string &name, const std::string &passwd, const [Roles](#) &rls)
Create a new user instance with the specified values.
- const std::string & [getUsername](#) () const
- const std::string & [getPassword](#) () const
- const [Roles](#) & [getRoles](#) () const

7.43.1 Detailed Description

A simple value object that represents a user configured in the users.xml file.

Date

Created 2011/09/12 18:41

Copyright

Copyright © 2011, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[User.h](#) 5567 2013-02-23 22:13:21Z rakesh

7.43.2 Constructor & Destructor Documentation

7.43.2.1 `spt::servlet::model::User::User ( const std::string & name, const std::string & passwd, const Roles & rls )`
`[inline]`

Create a new user instance with the specified values.

Parameters

<i>name</i>	The user/login name of the user.
<i>passwd</i>	The password for the user
<i>rls</i>	The roles assigned to the user.

7.43.3 Member Function Documentation

7.43.3.1 `const std::string& spt::servlet::model::User::getPassword ( ) const` `[inline]`

Returns

`const std::string` Return the password for the user.

7.43.3.2 `const Roles& spt::servlet::model::User::getRoles ( ) const` `[inline]`

Returns

`const Roles` Return the roles assigned to the user.

7.43.3.3 `const std::string& spt::servlet::model::User::getUsername ( ) const` `[inline]`

Returns

`const std::string` Return the user/login name for the user.

The documentation for this class was generated from the following file:

- [/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/User.h](#)

7.44 spt::servlet::model::UserXml Class Reference

Parser for a users.xml file based on tomcat-users.xml format. This is the default user database supported by the container.

```
#include <UserXml.h>
```

Public Member Functions

- void [load](#) (const std::string &path)
- const [UserMap](#) & [getUserMap](#) () const

7.44.1 Detailed Description

Parser for a users.xml file based on tomcat-users.xml format. This is the default user database supported by the container.

Date

Created 2011/09/10 18:41

Copyright

Copyright © 2011, Sans Pareil Technologies, Inc.

Author

Rakesh

Version**Id**

[UserXml.h](#) 5567 2013-02-23 22:13:21Z rakesh

7.44.2 Member Function Documentation

7.44.2.1 `const UserMap& spt::servlet::model::UserXml::getUserMap ( ) const` `[inline]`

Returns

const UserMap Return the map of users parsed from the file

7.44.2.2 `void spt::servlet::model::UserXml::load ( const std::string & path )`

Load users.xml file from specified location.

Exceptions

<i>Poco::Exception</i>	if file not found or invalid
------------------------	------------------------------

The documentation for this class was generated from the following file:

- `/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/UserXml.h`

7.45 spt::servlet::VirtualHostServer Class Reference

Handler for each virtual host handled by Minnal.

```
#include <VirtualHostServer.h>
```

Public Member Functions

- [VirtualHostServer](#) (const std::string &webXml, const std::string &userXml, const [model::HostContextPtr](#) hostContext)
Create a new virtual host using the specified `web.xml`, user configuration and host context information.
- [ServletPtr](#) [createServlet](#) (const Poco::Net::HTTPRequest &request)
Return the servlet mapped to the path, or the default file handler servlet.

7.45.1 Detailed Description

Handler for each virtual host handled by Minnal.

The standard request handler factory that will parse the requested resource path and attempt to retrieve the associated servlet to handle the request. If no servlet has been mapped to the path, the default servlet that handles requests for static resources stored under the doc root is invoked.

Date

Created 2011/08/21 18:41

Copyright

Copyright © 2011, Sans Pareil Technologies, Inc.

Author

Rakesh

Version**Id**

[VirtualHostServer.h](#) 5632 2014-10-06 22:04:21Z rakesh

7.45.2 Constructor & Destructor Documentation

7.45.2.1 `spt::servlet::VirtualHostServer::VirtualHostServer ( const std::string & webXml, const std::string & userXml, const model::HostContextPtr hostContext )`

Create a new virtual host using the specified `web.xml`, user configuration and host context information.

Parameters

<i>webXml</i>	The web.xml file to use to configure this host
<i>userXml</i>	The xml file for authentication and authorisation
<i>hostContext</i>	The host context with other configuration information.

7.45.3 Member Function Documentation

7.45.3.1 `ServletPtr spt::servlet::VirtualHostServer::createServlet ( const Poco::Net::HTTPServerRequest & request )`

Return the servlet mapped to the path, or the default file handler servlet.

Parameters

<i>request</i>	The client request to handle.
----------------	-------------------------------

Returns

`ServletPtr` The servlet instance to handle the request.

The documentation for this class was generated from the following file:

- `/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/VirtualHostServer.h`

7.46 spt::servlet::model::WebXml Class Reference

Parser for a JEE standard web.xml servlet configuration file. Note that not all JEE configuration is supported by the servlet container.

```
#include <WebXml.h>
```

Public Member Functions

- void [load](#) (const std::string &path)
- const [ContextParams](#) & [getContextParams](#) () const
- const [Servlets](#) & [getServlets](#) () const
- const [SecurityConstraints](#) & [getSecurityConstraints](#) () const
- const [LoginConfigPtr](#) [getLoginConfig](#) () const
- const [ListPtr](#) [getWelcomeFiles](#) () const
- const [ErrorPagePtr](#) [getErrorPages](#) () const

7.46.1 Detailed Description

Parser for a JEE standard web.xml servlet configuration file. Note that not all JEE configuration is supported by the servlet container.

Date

Created 2011/09/10 18:41

Copyright

Copyright © 2011, Sans Pareil Technologies, Inc.

Author

Rakesh

Version

Id

[WebXml.h](#) 5567 2013-02-23 22:13:21Z rakesh

7.46.2 Member Function Documentation

7.46.2.1 `const ContextParams& spt::servlet::model::WebXml::getContextParams ( ) const` `[inline]`

Returns

const ContextParams Return the context params parsed

7.46.2.2 `const ErrorPagePtr spt::servlet::model::WebXml::getErrorPages ( ) const` `[inline]`

Returns

const ErrorPagePtr Return the error pages configuration parsed

7.46.2.3 `const LoginConfigPtr spt::servlet::model::WebXml::getLoginConfig ( ) const` `[inline]`

Returns

const LoginConfigPtr Return the login configuration parsed

7.46.2.4 `const SecurityConstraints& spt::servlet::model::WebXml::getSecurityConstraints ( ) const` `[inline]`

Returns

`const SecurityConstraints` Return the security constraints parsed

7.46.2.5 `const Servlets& spt::servlet::model::WebXml::getServlets ( ) const` `[inline]`

Returns

`const Servlets` Return the configured servlets as a hash map of config objects

7.46.2.6 `const ListPtr spt::servlet::model::WebXml::getWelcomeFiles ( ) const` `[inline]`

Returns

`const ListPtr` Return the welcome files list parsed

7.46.2.7 `void spt::servlet::model::WebXml::load ( const std::string & path )`

Load web.xml file from specified location.

Exceptions

<code>Poco::Exception</code>	if file not found or invalid
------------------------------	------------------------------

The documentation for this class was generated from the following file:

- `/Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/WebXml.h`

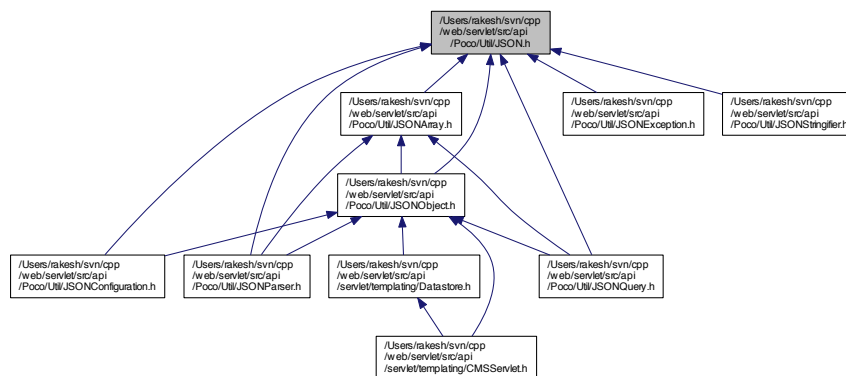
Chapter 8

File Documentation

8.1 /Users/rakesh/svn/cpp/web/servlet/src/api/mainpage.dox File Reference

8.2 /Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSON.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- `#define JSON_API`

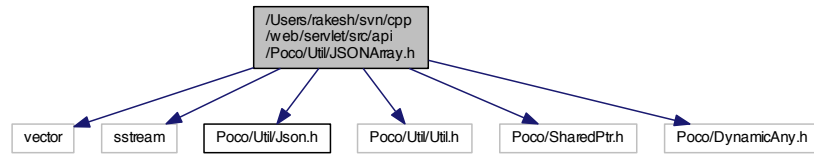
8.2.1 Macro Definition Documentation

8.2.1.1 `#define JSON_API`

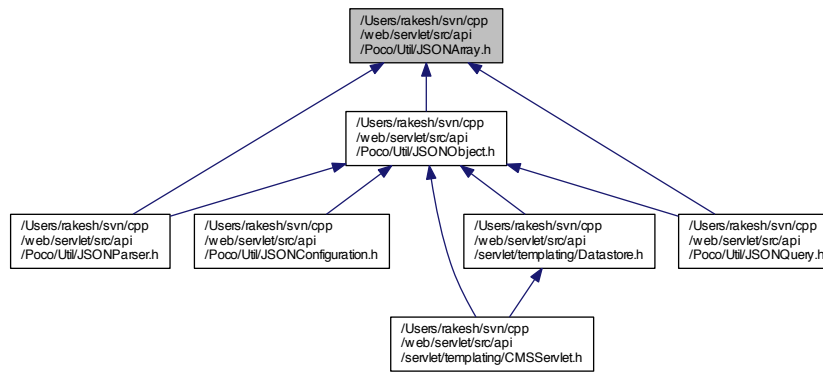
8.3 /Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONArray.h File Reference

```
#include <vector>
#include <sstream>
#include <Poco/Util/Json.h>
#include <Poco/Util/Util.h>
#include <Poco/SharedPtr.h>
#include <Poco/DynamicAny.h>
```

Include dependency graph for JSONArray.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Poco::Util::JSONArray](#)
- class [Poco::DynamicAnyHolderImpl< Util::JSONArray::Ptr >](#)

Namespaces

- [Poco](#)
- [Poco::Util](#)

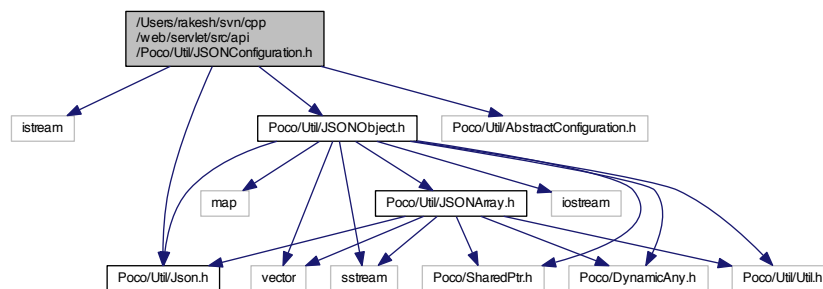
8.4 /Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONConfiguration.h File Reference

```

#include <istream>
#include <Poco/Util/Json.h>
#include <Poco/Util/AbstractConfiguration.h>
#include <Poco/Util/JSONObject.h>

```

Include dependency graph for JSONConfiguration.h:



Classes

- class [Poco::Util::JSONConfiguration](#)

This configuration class extracts configuration properties from a JSON object. An XPath-like syntax for property names is supported to allow full access to the JSON object.

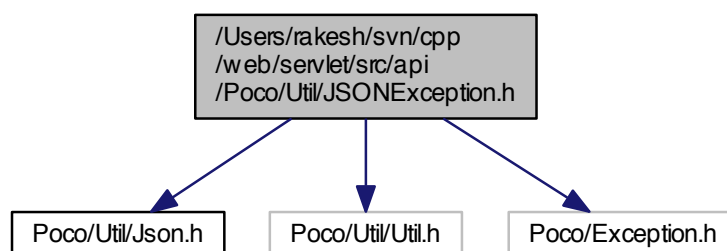
Namespaces

- [Poco](#)
- [Poco::Util](#)

8.5 /Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONException.h File Reference

```
#include <Poco/Util/Json.h>
#include <Poco/Util/Util.h>
#include <Poco/Exception.h>
```

Include dependency graph for JSONException.h:



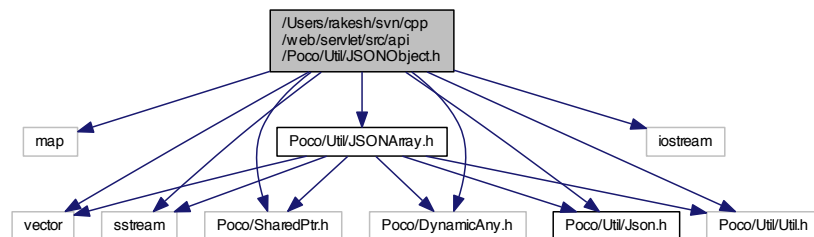
Namespaces

- [Poco](#)
- [Poco::Util](#)

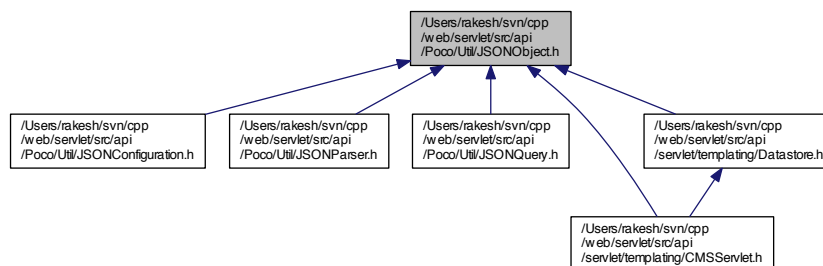
8.6 /Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONObject.h File Reference

```
#include <map>
#include <vector>
#include <iostream>
#include <sstream>
#include <Poco/SharedPtr.h>
#include <Poco/DynamicAny.h>
#include <Poco/Util/Json.h>
#include <Poco/Util/Util.h>
#include <Poco/Util/JSONArray.h>
```

Include dependency graph for JSONObject.h:



This graph shows which files directly or indirectly include this file:



Classes

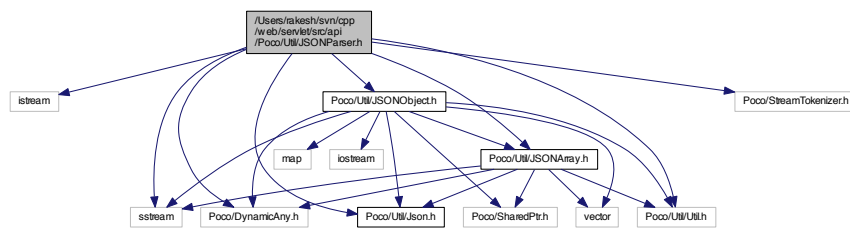
- class [Poco::Util::JSONObject](#)
Represents a JSON object.
- class [Poco::DynamicAnyHolderImpl](#)< [Util::JSONObject::Ptr](#) >

Namespaces

- [Poco](#)
- [Poco::Util](#)

8.7 /Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONParser.h File Reference

```
#include <istream>
#include <sstream>
#include <Poco/DynamicAny.h>
#include <Poco/StreamTokenizer.h>
#include <Poco/Util/Json.h>
#include <Poco/Util/Util.h>
#include <Poco/Util/JSONObject.h>
#include <Poco/Util/JSONArray.h>
Include dependency graph for JSONParser.h:
```



Classes

- class [Poco::Util::JSONParser](#)

A class for passing JSON strings or streams.

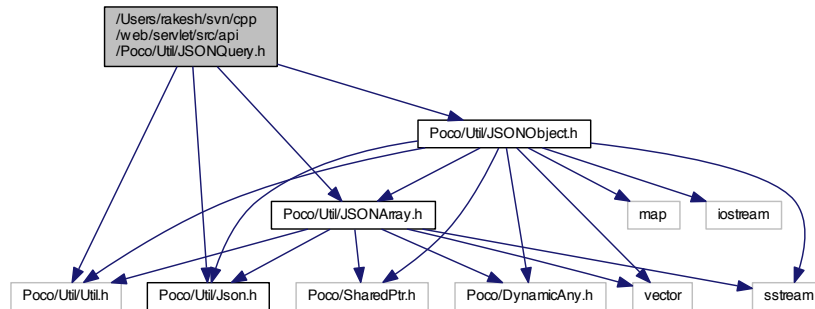
Namespaces

- [Poco](#)
- [Poco::Util](#)

8.8 /Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONQuery.h File Reference

```
#include <Poco/Util/Json.h>
#include <Poco/Util/Util.h>
#include <Poco/Util/JSONObject.h>
#include <Poco/Util/JSONArray.h>
```

Include dependency graph for JSONQuery.h:



Classes

- class [Poco::Util::JSONQuery](#)

Class that can be used to search for a value in a JSON object or array.

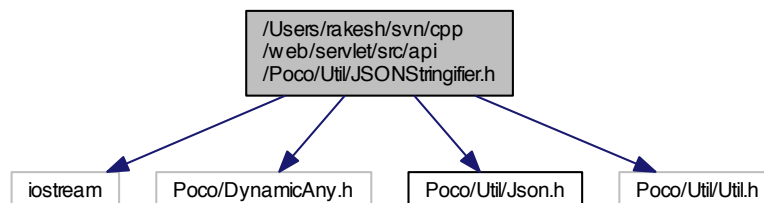
Namespaces

- [Poco](#)
- [Poco::Util](#)

8.9 /Users/rakesh/svn/cpp/web/servlet/src/api/Poco/Util/JSONStringifier.h File Reference

```
#include <iostream>
#include <Poco/DynamicAny.h>
#include <Poco/Util/Json.h>
#include <Poco/Util/Util.h>
```

Include dependency graph for JSONStringifier.h:



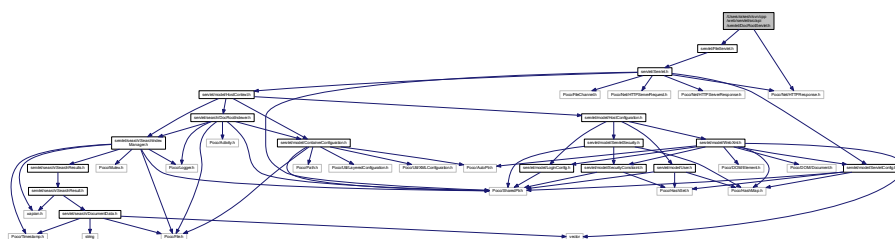
Classes

- class [Poco::Util::JSONStringifier](#)

Helper class for creating a String from a JSON object or array.

- Poco
- Poco::Util

```
#include <servlet/FileServlet.h>
#include <Poco/Net/HTTPResponse.h>
Include dependency graph for DocRootServlet.h:
```

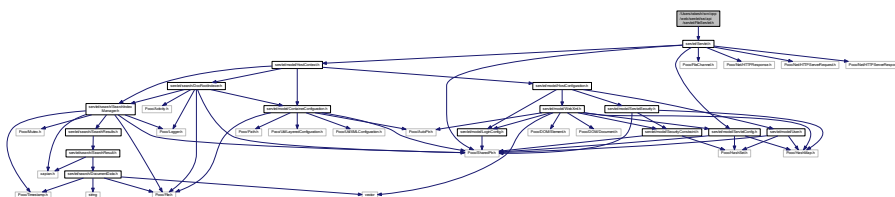


- class `spt::servlet::DocRootServlet`
Default servlet for serving static files.

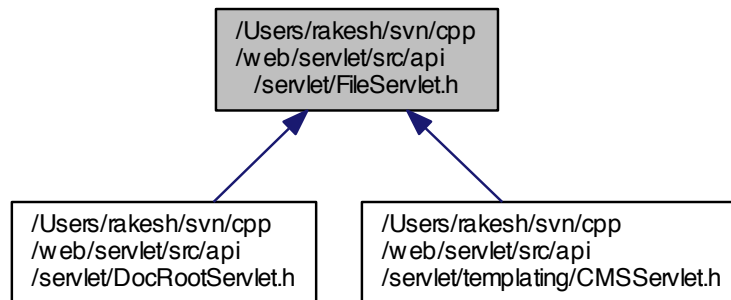
- `spt`
- `spt::servlet`

Minimal C++ `Servlet` Container implementation.

```
#include <servlet/Servlet.h>
Include dependency graph for FileServlet.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [spt::servlet::FileServlet](#)

Abstract base servlet for serving files.

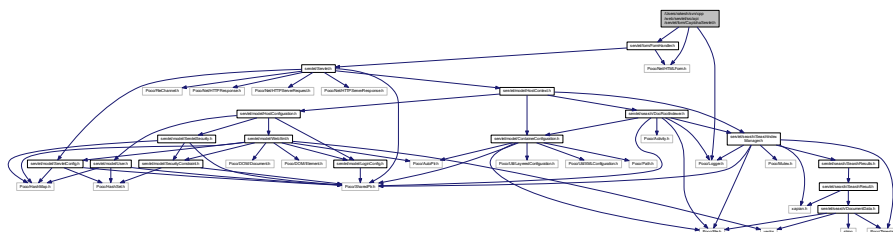
Namespaces

- [spt](#)
- [spt::servlet](#)

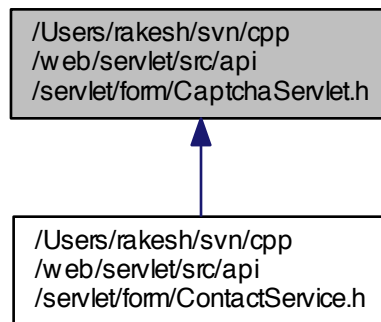
Minnal C++ [Servlet](#) Container implementation.

8.12 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/form/CaptchaServlet.h File Reference

```
#include <servlet/form/FormHandler.h>
#include <Poco/Logger.h>
#include <Poco/Net/HTMLForm.h>
Include dependency graph for CaptchaServlet.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [spt::servlet::form::CaptchaServlet](#)

A base servlet class that adds support for captcha validation. Sub-classes should use the [validateCaptcha](#) method to validate the client captcha response in addition to any other business rules.

Namespaces

- [spt](#)
- [spt::servlet](#)

Minnal C++ [Servlet](#) Container implementation.

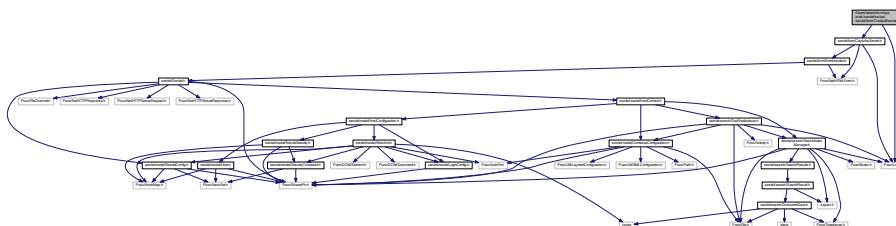
- [spt::servlet::form](#)

Servlets and supporting classes for form handling.

8.13 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/form/ContactService.h File Reference

```
#include <servlet/form/CaptchaServlet.h>
#include <Poco/Logger.h>
```

Include dependency graph for ContactService.h:



Classes

- class [spt::servlet::form::ContactService](#)

A handler for contact forms.

Namespaces

- [spt](#)
- [spt::servlet](#)

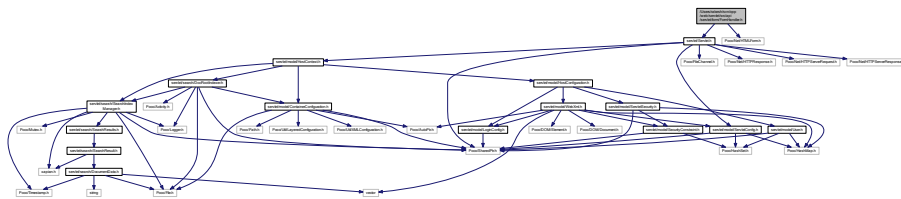
Minnal C++ [Servlet](#) Container implementation.

- [spt::servlet::form](#)

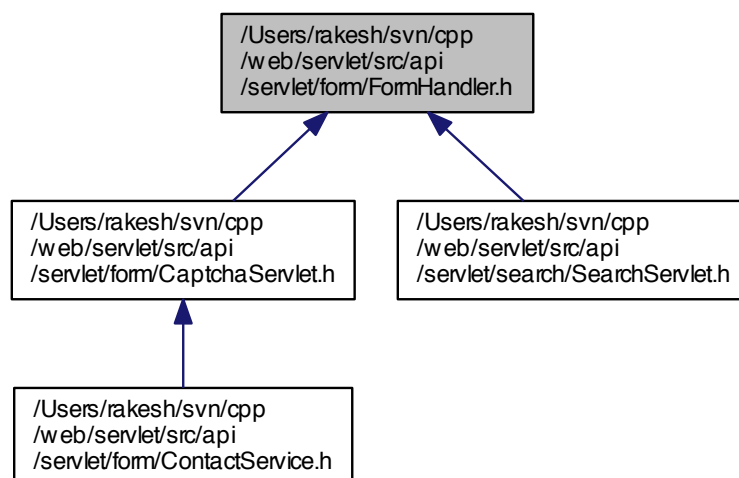
Servlets and supporting classes for form handling.

8.14 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/form/FormHandler.h File Reference

```
#include <servlet/Servlet.h>
#include <Poco/Net/HTMLForm.h>
Include dependency graph for FormHandler.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [spt::servlet::form::FormHandler](#)

A base form handler servlet.

Namespaces

- [spt](#)
- [spt::servlet](#)

Minnal C++ [Servlet](#) Container implementation.

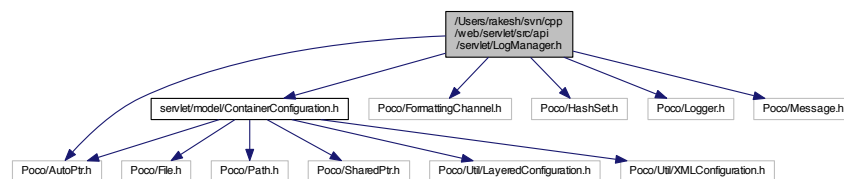
- [spt::servlet::form](#)

Servlets and supporting classes for form handling.

8.15 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/LogManager.h File Reference

```
#include <servlet/model/ContainerConfiguration.h>
#include <Poco/AutoPtr.h>
#include <Poco/FormattingChannel.h>
#include <Poco/HashSet.h>
#include <Poco/Logger.h>
#include <Poco/Message.h>
```

Include dependency graph for LogManager.h:



Classes

- class [spt::servlet::LogManager](#)

A singleton class that is used to set up the loggers for the container.

Namespaces

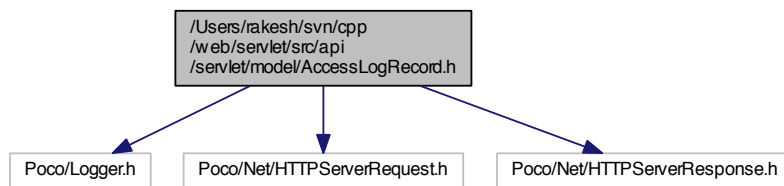
- [spt](#)
- [spt::servlet](#)

Minnal C++ [Servlet](#) Container implementation.

8.16 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/AccessLogRecord.h File Reference

```
#include <Poco/Logger.h>
#include <Poco/Net/HTTPServerRequest.h>
#include <Poco/Net/HTTPServerResponse.h>
```

Include dependency graph for AccessLogRecord.h:



Classes

- class `spt::servlet::model::AccessLogRecord`

Utility class that provides support for generating virtual host access logs in common log format.

Namespaces

- `spt`
- `spt::servlet`

Minnal C++ [Servlet](#) Container implementation.

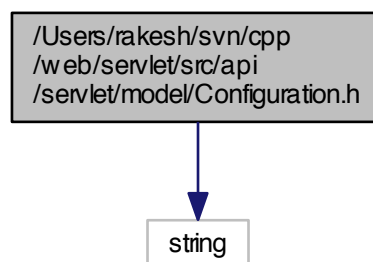
- `spt::servlet::model`

Namespace for value objects used to represent various configuration information for the server. Minnal is configured through a server wide configuration file, and additional `web.xml` style configuration files for each virtual host configured in the server.

8.17 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/Configuration.h File Reference

```
#include <string>
```

Include dependency graph for Configuration.h:



Namespaces

- [spt](#)
- [spt::servlet](#)

Minnal C++ [Servlet](#) Container implementation.

Functions

- `std::string & spt::servlet::getVersionNumber ()`
Return the current version number for Minnal.
- `std::string & spt::servlet::getServerIdentifier ()`
Return the standard server identifier which will be sent back as part of standard HTTP response header.
- `std::string & spt::servlet::getServerPort ()`
Return the key used to configure the port on which the server port listens to in the `minnal.xml` file.
- `std::string & spt::servlet::getServerUser ()`
Return the key used to configure the user used to own the servlet container process. The process is meant to be started by the `root` user, and once the process has bound to the `port`, the effective group and user id of the process will be switched to the configured user and group values.
- `std::string & spt::servlet::getServerGroup ()`
Return the key used to configure the group used to own the servlet container process. The process is meant to be started by the `root` user, and once the process has bound to the `port`, the effective group and user id of the process will be switched to the configured user and group values.
- `std::string & spt::servlet::getConnectionKeepAlive ()`
Return the key used to configure whether `keep-alive` support is enabled for client connections.
- `std::string & spt::servlet::getMaxQueued ()`
Return the maximum size of request queue to be maintained by the server before rejecting client connections.
- `std::string & spt::servlet::getMaxThreads ()`
Return the key used to configure the maximum number of worker threads the server process will maintain to serve requests.
- `std::string & spt::servlet::getMimeTypeMapping ()`
Return the key used to configure the mime type mapping file used to look up the mime-type for files.
- `std::string & spt::servlet::getServletLoggerFile ()`
Return the key used to configure the name of the log file for the virtual host.
- `std::string & spt::servlet::getServletLogRotation ()`
Return the key used to configure the log file rotation interval for the virtual host.
- `std::string & spt::servlet::getServletLoggerArchive ()`
Return the key used to configure the log file archival policy for the virtual host.
- `std::string & spt::servlet::getServletLoggerCompress ()`
Return the key used to configure whether the archived log files are to be compressed for the virtual host.
- `std::string & spt::servlet::getServletLoggerPurgeCount ()`
Return the key used to configure the number of archived log files to be maintained for the virtual host.
- `std::string & spt::servlet::getServletLoggerTimes ()`
Return the key used to configure the time zone to be used for log messages for the virtual host.
- `std::string & spt::servlet::getServletLoggerPattern ()`
Return the key used to configure the log record format for log messages for the virtual host.
- `std::string & spt::servlet::getCacheCompressedFiles ()`
Return the key used to configure whether the server should attempt to cache compressed versions of documents served to clients.
- `std::string & spt::servlet::getDocumentRoot ()`
Return the key used to configure the document root directory for the virtual host.
- `std::string & spt::servlet::getDocumentRootCache ()`
Return the key used to configure the cache directory for the virtual host.

- `std::string` & `spt::servlet::getDenyQueryStrings ()`

Return the key used to configure whether the server should deny all requests that have query strings for the virtual host.

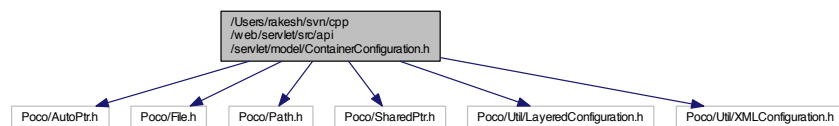
- `std::string` & `spt::servlet::getDocrootHandler ()`

Return the key used to configure the servlet used to server static files for the virtual host.

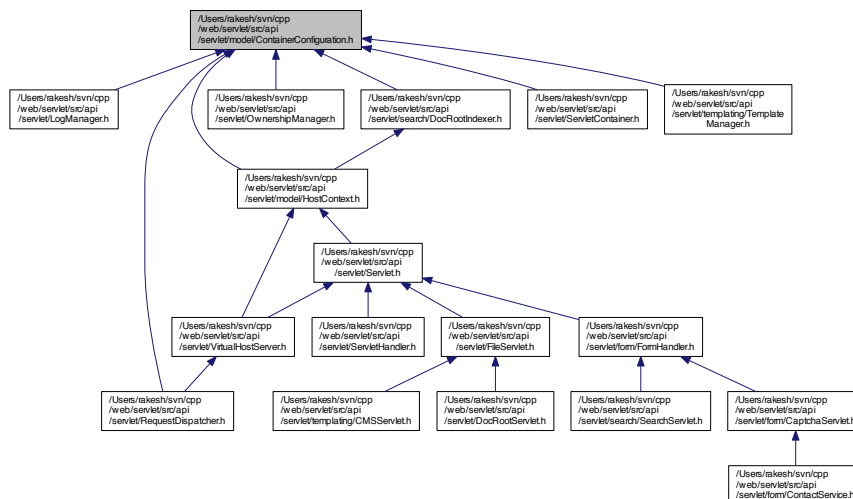
8.18 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/ContainerConfiguration.h File Reference

```
#include <Poco/AutoPtr.h>
#include <Poco/File.h>
#include <Poco/Path.h>
#include <Poco/SharedPtr.h>
#include <Poco/Util/LayeredConfiguration.h>
#include <Poco/Util/XMLConfiguration.h>
```

Include dependency graph for ContainerConfiguration.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `spt::servlet::model::ContainerConfiguration`

Value object that stores the various configuration objects used to manage the servlet container.

Namespaces

- `spt`
- `spt::servlet`

*Minnal C++ **Servlet** Container implementation.*

- `spt::servlet::model`

Namespace for value objects used to represent various configuration information for the server. Minnal is configured through a server wide configuration file, and additional `web.xml` style configuration files for each virtual host configured in the server.

Typedefs

- `typedef Poco::AutoPtr
< Poco::Util::XMLConfiguration > spt::servlet::model::ServerConfigurationPtr`

Auto pointer to the xml configuration data for the server.

- typedef
Poco::Util::LayeredConfiguration [spt::servlet::model::MimeTypes](#)

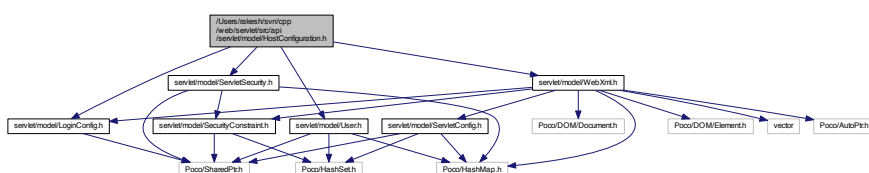
The mime-type configuration data.

- `typedef Poco::SharedPtr
< ContainerConfiguration > spt::servlet::model::ContainerConfigurationPtr`

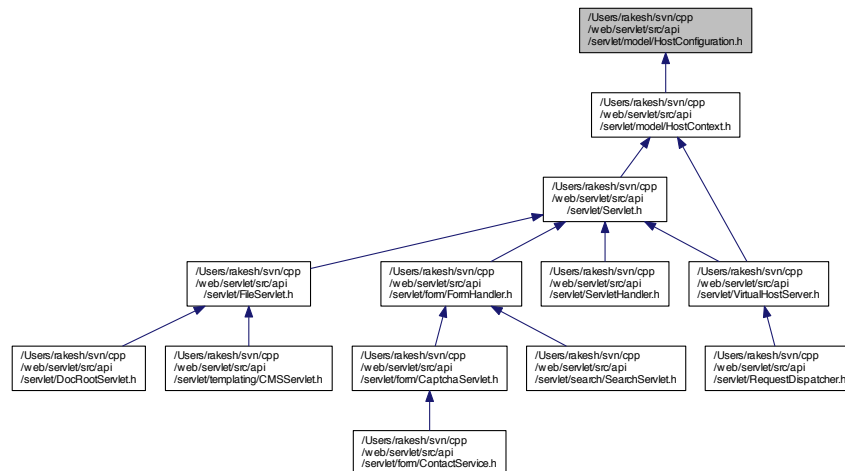
Shared pointer to a container configuration instance.

8.19 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/HostConfiguration.h File
Reference

```
#include <servlet/model/LoginConfig.h>
#include <servlet/model/ServletSecurity.h>
#include <servlet/model/User.h>
#include <servlet/model/WebXml.h>
Include dependency graph for HostConfiguration.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [spt::servlet::model::HostConfiguration](#)

A value object that represents the configured values for a virtual host in the container. The configuration is parsed from the web.xml and user.xml files configured for a virtual host.

Namespaces

- [spt](#)
- [spt::servlet](#)

Minnal C++ [Servlet](#) Container implementation.

- [spt::servlet::model](#)

Namespace for value objects used to represent various configuration information for the server. Minnal is configured through a server wide configuration file, and additional web.xml style configuration files for each virtual host configured in the server.

Typedefs

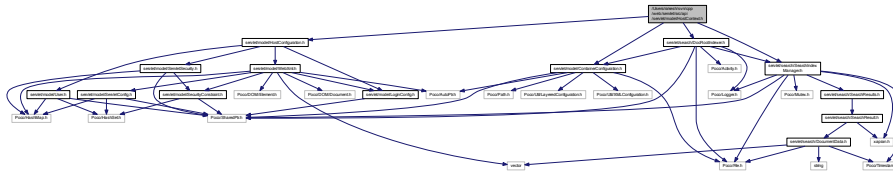
- typedef Poco::SharedPtr
< HostConfiguration > [spt::servlet::model::HostConfigurationPtr](#)

Shared pointer to a host configuration instance.

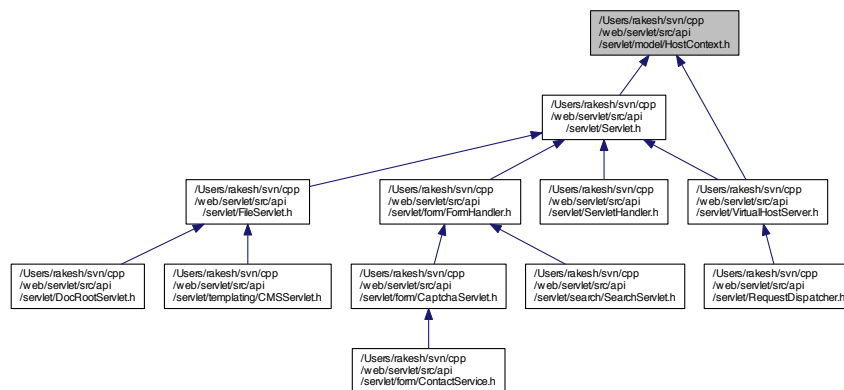
8.20 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/HostContext.h File Reference

```
#include <servlet/model/ContainerConfiguration.h>
#include <servlet/model/HostConfiguration.h>
#include <servlet/search/SearchIndexManager.h>
#include <servlet/search/DocRootIndexer.h>
```

Include dependency graph for HostContext.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [spt::servlet::model::HostContext](#)

A value object that stores the information pertinent to a virtual host.

Namespaces

- [spt](#)
- [spt::servlet](#)

Minnal C++ [Servlet](#) Container implementation.

- [spt::servlet::model](#)

Namespace for value objects used to represent various configuration information for the server. Minnal is configured through a server wide configuration file, and additional `web.xml` style configuration files for each virtual host configured in the server.

Typedefs

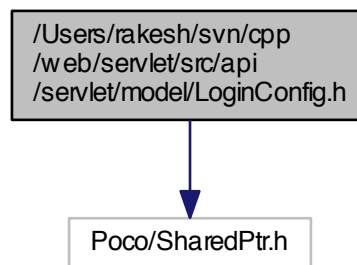
- typedef [Poco::SharedPtr](#)
< [HostContext](#) > [spt::servlet::model::HostContextPtr](#)

Shared pointer to a host context instance.

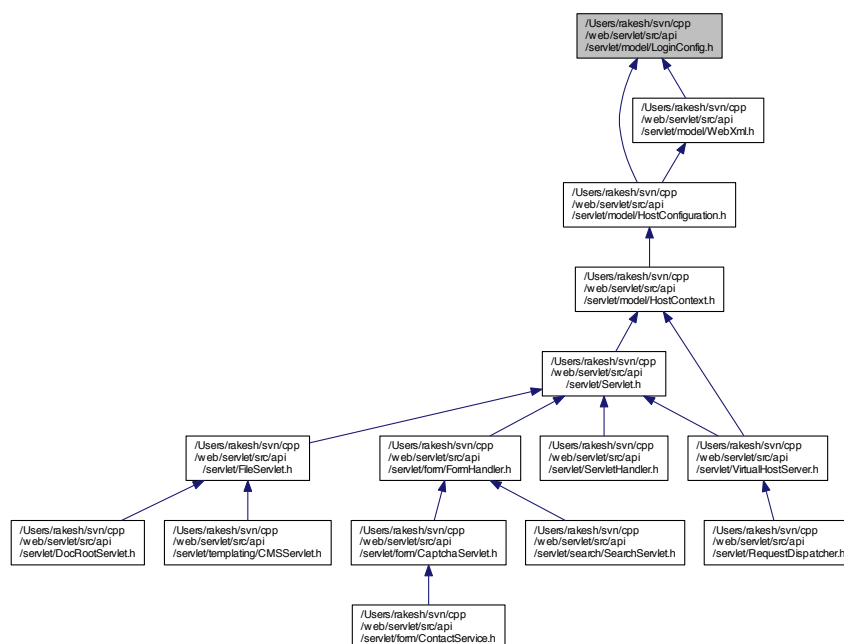
8.21 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/LoginConfig.h File Reference

```
#include <Poco/SharedPtr.h>
```

Include dependency graph for LoginConfig.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [spt::servlet::model::LoginConfig](#)

A simple value object that represents a login-config element in a web.xml file.

Namespaces

- [spt](#)
- [spt::servlet](#)

Minnal C++ [Servlet](#) Container implementation.

- [spt::servlet::model](#)

Namespace for value objects used to represent various configuration information for the server. Minnal is configured through a server wide configuration file, and additional `web.xml` style configuration files for each virtual host configured in the server.

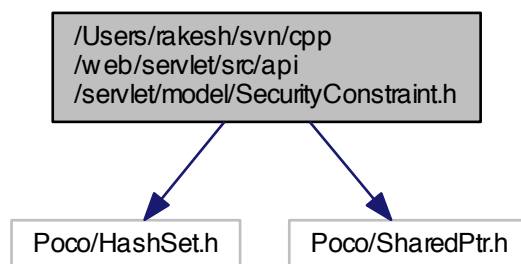
Typedefs

- typedef Poco::SharedPtr
< LoginConfig > [spt::servlet::model::LoginConfigPtr](#)

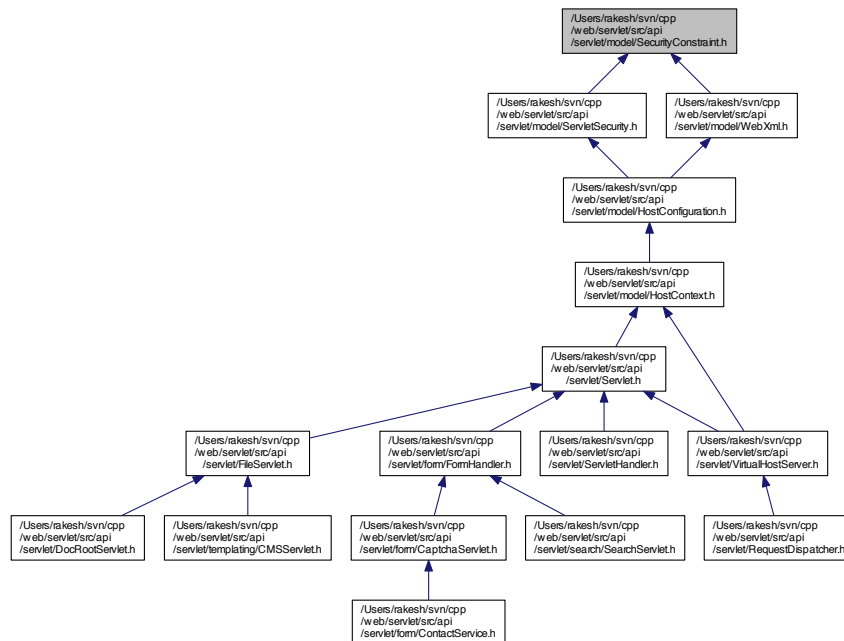
Shared pointer to a login configuration instance.

8.22 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/SecurityConstraint.h File Reference

```
#include <Poco/HashSet.h>
#include <Poco/SharedPtr.h>
Include dependency graph for SecurityConstraint.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [spt::servlet::model::SecurityConstraint](#)

A simple value object used to store security constraint declarations in a web.xml file.

Namespaces

- [spt](#)
- [spt::servlet](#)

Minnal C++ [Servlet](#) Container implementation.

- [spt::servlet::model](#)

Namespace for value objects used to represent various configuration information for the server. Minnal is configured through a server wide configuration file, and additional `web.xml` style configuration files for each virtual host configured in the server.

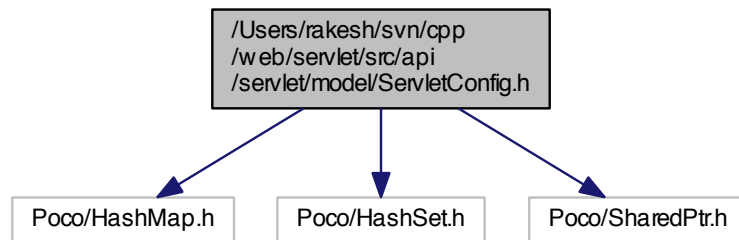
Typedefs

- typedef Poco::HashSet
< std::string > [spt::servlet::model::ListEntry](#)
Set of string values.
- typedef Poco::SharedPtr
< ListEntry > [spt::servlet::model::ListPtr](#)
Shared pointer to a set of string values.
- typedef Poco::SharedPtr
< SecurityConstraint > [spt::servlet::model::SecurityConstraintPtr](#)
Shared pointer to a security constraint instance.

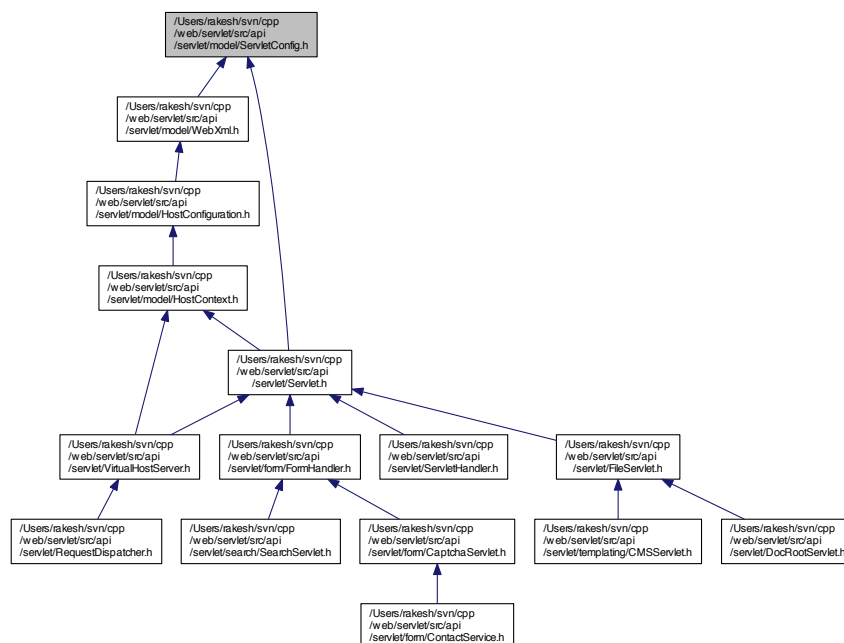
8.23 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/ServletConfig.h File Reference

```
#include <Poco/HashMap.h>
#include <Poco/HashSet.h>
#include <Poco/SharedPtr.h>
```

Include dependency graph for ServletConfig.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [spt::servlet::model::ServletConfig](#)

A simple value object that represents a servlet configuration in a web.xml file. This object stores the information from the servlet and servlet-mapping elements.

Namespaces

- [spt](#)
- [spt::servlet](#)

Minnal C++ [Servlet](#) Container implementation.

- [spt::servlet::model](#)

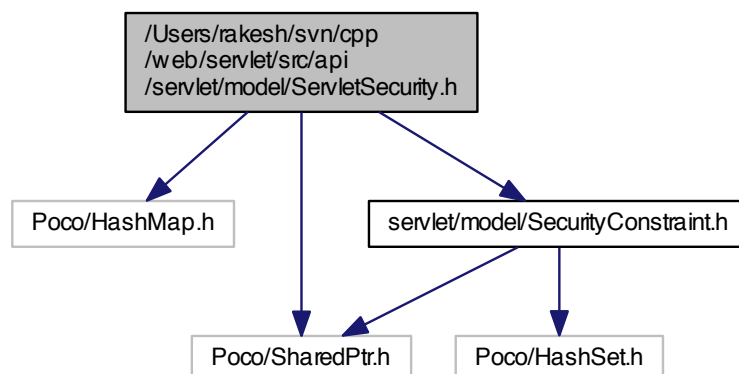
Namespace for value objects used to represent various configuration information for the server. Minnal is configured through a server wide configuration file, and additional `web.xml` style configuration files for each virtual host configured in the server.

Typedefs

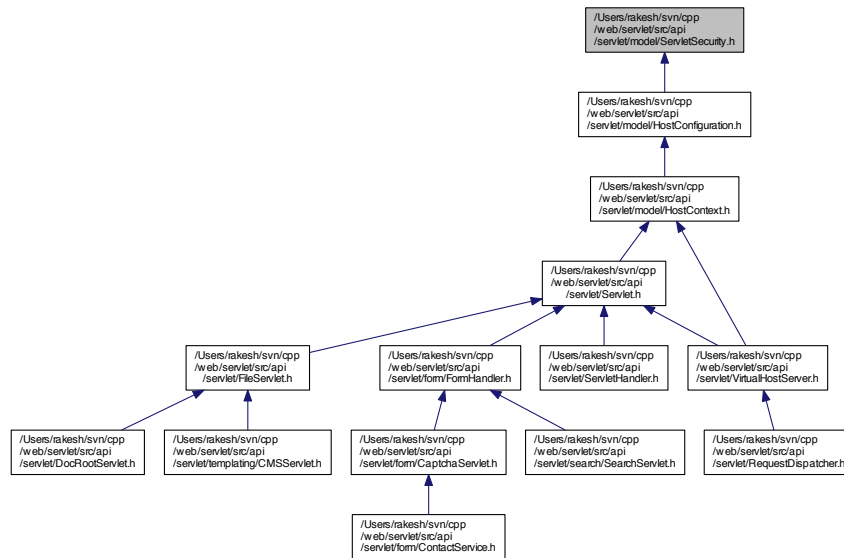
- typedef Poco::HashMap
< std::string, std::string > [spt::servlet::model::ContextParams](#)
Mapping for context-param name-value pairs for each virtual host.
- typedef Poco::HashMap
< std::string, std::string > [spt::servlet::model::InitParams](#)
Mapping of init-param name-value pairs for each configured servlet.
- typedef Poco::HashSet
< std::string, Poco::Hash
< std::string > > [spt::servlet::model::PathMapping](#)
Set of paths mapped for a servlet.
- typedef Poco::SharedPtr
< ServletConfig > [spt::servlet::model::ServletConfigPtr](#)
Shared pointer to a servlet configuration instance.

8.24 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/ServletSecurity.h File Reference

```
#include <Poco/HashMap.h>
#include <Poco/SharedPtr.h>
#include <servlet/model/SecurityConstraint.h>
Include dependency graph for ServletSecurity.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [spt::servlet::model::ServletSecurity](#)

A simple value object that holds the security configuration for a path that has been mapped to a servlet.

Namespaces

- [spt](#)
- [spt::servlet](#)

Minnal C++ [Servlet](#) Container implementation.

- [spt::servlet::model](#)

Namespace for value objects used to represent various configuration information for the server. Minnal is configured through a server wide configuration file, and additional `web.xml` style configuration files for each virtual host configured in the server.

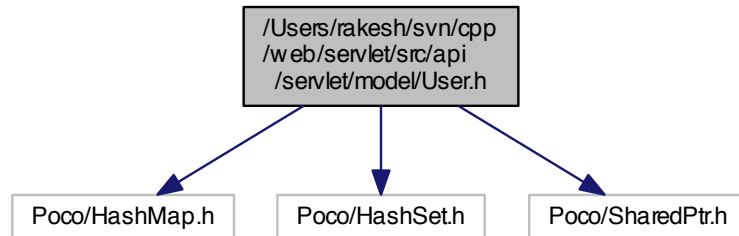
Typedefs

- typedef Poco::SharedPtr
< ServletSecurity > [spt::servlet::model::ServletSecurityPtr](#)
Shared pointer to a servlet security instance.
- typedef Poco::HashMap
< std::string,
ServletSecurityPtr > [spt::servlet::model::ServletSecurityMap](#)
Map for path to servlet security configuration look up.

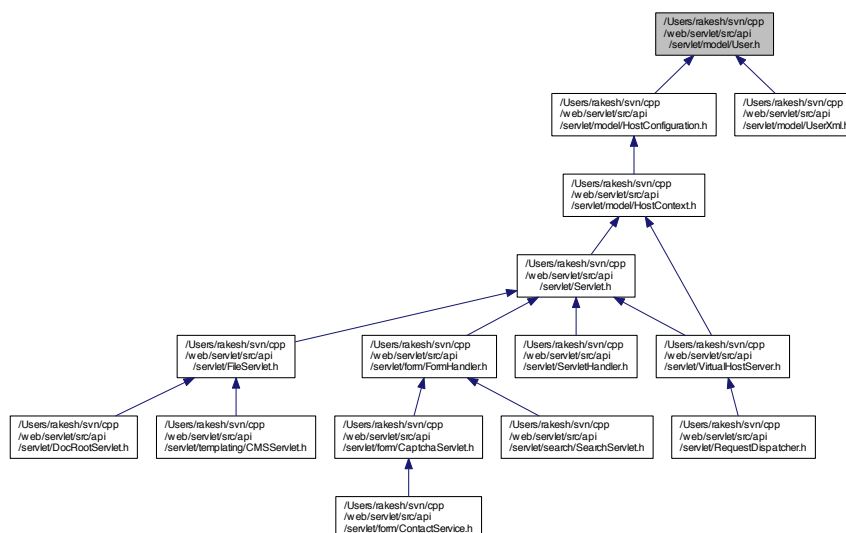
8.25 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/User.h File Reference

```
#include <Poco/HashMap.h>
```

```
#include <Poco/HashSet.h>
#include <Poco/SharedPtr.h>
Include dependency graph for User.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [spt::servlet::model::User](#)

A simple value object that represents a user configured in the users.xml file.

Namespaces

- [spt](#)
- [spt::servlet](#)

Minnal C++ [Servlet](#) Container implementation.

- [spt::servlet::model](#)

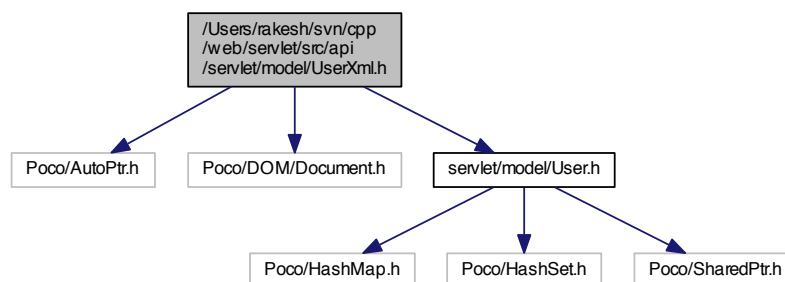
Namespace for value objects used to represent various configuration information for the server. Minnal is configured through a server wide configuration file, and additional `web.xml` style configuration files for each virtual host configured in the server.

Typedefs

- typedef Poco::HashSet
< std::string > [spt::servlet::model::Roles](#)
The roles used to categorise users. Usually used for authorisation.
- typedef Poco::SharedPtr< User > [spt::servlet::model::UserPtr](#)
Shared pointer to a user instance.
- typedef Poco::HashMap
< std::string, UserPtr > [spt::servlet::model::UserMap](#)
Map for looking up a user instance by username.

8.26 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/UserXml.h File Reference

```
#include <Poco/AutoPtr.h>
#include <Poco/DOM/Document.h>
#include <servlet/model/User.h>
Include dependency graph for UserXml.h:
```



Classes

- class [spt::servlet::model::UserXml](#)
Parser for a users.xml file based on tomcat-users.xml format. This is the default user database supported by the container.

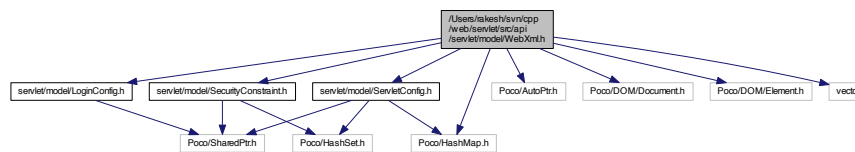
Namespaces

- [spt](#)
- [spt::servlet](#)
Minnal C++ [Servlet](#) Container implementation.
- [spt::servlet::model](#)
Namespace for value objects used to represent various configuration information for the server. Minnal is configured through a server wide configuration file, and additional `web.xml` style configuration files for each virtual host configured in the server.

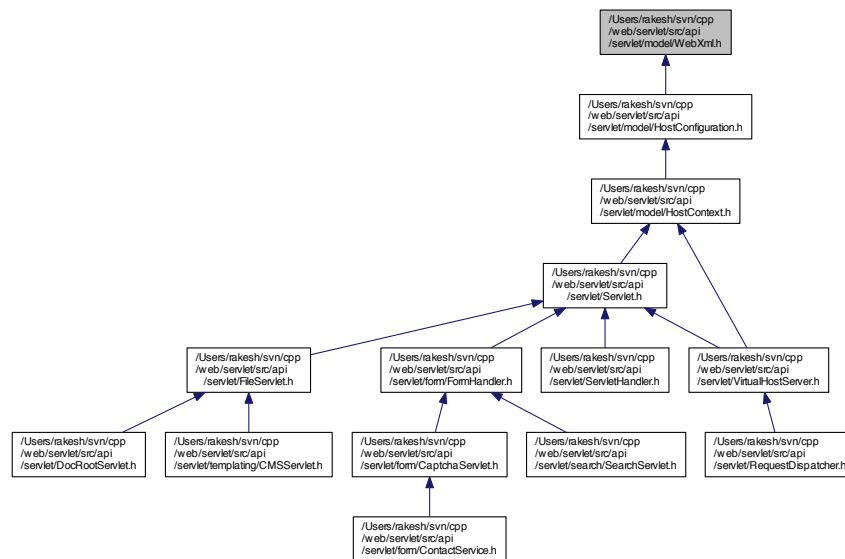
8.27 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/model/WebXml.h File Reference

```
#include <servlet/model/LoginConfig.h>
#include <servlet/model/ServletConfig.h>
#include <servlet/model/SecurityConstraint.h>
#include <Poco/AutoPtr.h>
#include <Poco/HashMap.h>
#include <Poco/DOM/Document.h>
#include <Poco/DOM/Element.h>
#include <vector>
```

Include dependency graph for WebXml.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [spt::servlet::model::WebXml](#)

Parser for a JEE standard web.xml servlet configuration file. Note that not all JEE configuration is supported by the servlet container.

Namespaces

- [spt](#)
- [spt::servlet](#)

Minnal C++ [Servlet](#) Container implementation.

- [spt::servlet::model](#)

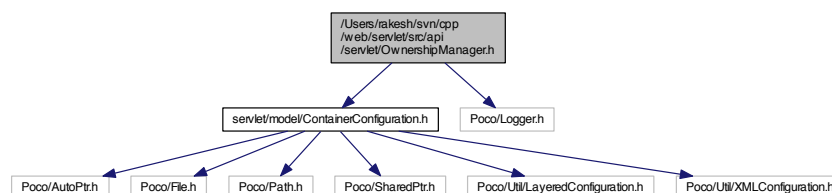
Namespace for value objects used to represent various configuration information for the server. Minnal is configured through a server wide configuration file, and additional `web.xml` style configuration files for each virtual host configured in the server.

Typedefs

- typedef Poco::HashMap
< std::string,
ServletConfigPtr > [spt::servlet::model::Servlets](#)
Map for servlet name to servlet configuration information.
- typedef Poco::HashMap< int,
std::string > [spt::servlet::model::ErrorPage](#)
Map for error code to error page lookup.
- typedef Poco::SharedPtr
< ErrorPage > [spt::servlet::model::ErrorPagePtr](#)
Shared pointer for error page configuration instance.
- typedef std::vector
< SecurityConstraintPtr > [spt::servlet::model::SecurityConstraints](#)
A collection of security constraints.

8.28 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/OwnershipManager.h File Reference

```
#include <servlet/model/ContainerConfiguration.h>
#include <Poco/Logger.h>
Include dependency graph for OwnershipManager.h:
```



Classes

- class [spt::servlet::OwnershipManager](#)

A utility class used to handle switching of user from root to unprivileged user after the server has started up and bound to the desired port.

Namespaces

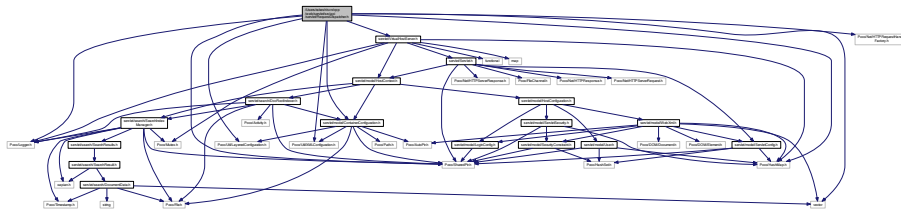
- [spt](#)
- [spt::servlet](#)

Minnal C++ [Servlet](#) Container implementation.

8.29 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/RequestDispatcher.h File Reference

```
#include <servlet/model/ContainerConfiguration.h>
#include <servlet/VirtualHostServer.h>
#include <Poco/HashMap.h>
#include <Poco/Logger.h>
#include <Poco/SharedPtr.h>
#include <Poco/Net/HTTPRequestHandlerFactory.h>
#include <Poco/Util/LayeredConfiguration.h>
#include <Poco/Util/XMLConfiguration.h>
#include <vector>
```

Include dependency graph for RequestDispatcher.h:



Classes

- class [spt::servlet::RequestDispatcher](#)

The standard request handler factory that will parse the Host HTTP header and delegate to the appropriate VirtualHost to process the request. The first configured VirtualHost will be used if no Host header has been specified.

Namespaces

- [spt](#)
- [spt::servlet](#)

Minnal C++ [Servlet](#) Container implementation.

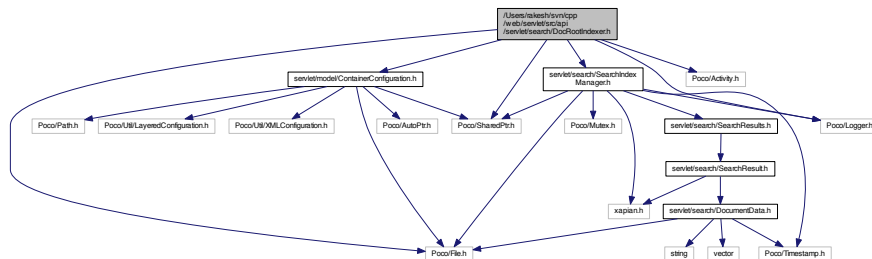
Typedefs

- typedef Poco::SharedPtr
< VirtualHostServer > [spt::servlet::VirtualHostPtr](#)
Shared pointer to a virtual host instance.
- typedef std::vector
< VirtualHostPtr > [spt::servlet::VirtualHostList](#)
A collection of virtual hosts.
- typedef Poco::HashMap
< std::string, VirtualHostPtr > [spt::servlet::VirtualHostMap](#)
Mapping of virtual hosts instances to their names.

8.30 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/DocRootIndexer.h File Reference

```
#include <servlet/search/SearchIndexManager.h>
```

Include dependency graph for DocRootIndexer.h:



```

graph TD
    Root["/Users/rakesh/svn/cpp  
/web/serveit/src/api  
/serveit/search/DocRootIndexer.h"] --> HostContext["/Users/rakesh/svn/cpp  
/web/serveit/src/api  
/serveit/model/HostContext.h"]
    HostContext --> Servlet["/Users/rakesh/svn/cpp  
/web/serveit/src/api  
/serveit/Servlet.h"]
    HostContext --> ServletHandler["/Users/rakesh/svn/cpp  
/web/serveit/src/api  
/serveit/ServletHandler.h"]
    Servlet --> FileServlet["/Users/rakesh/svn/cpp  
/web/serveit/src/api  
/serveit/FileServlet.h"]
    Servlet --> FormHandler["/Users/rakesh/svn/cpp  
/web/serveit/src/api  
/serveit/Form/FormHandler.h"]
    Servlet --> ServletHandler2["/Users/rakesh/svn/cpp  
/web/serveit/src/api  
/serveit/ServletHandler.h"]
    Servlet --> VirtualHostServer["/Users/rakesh/svn/cpp  
/web/serveit/src/api  
/serveit/VirtualHostServer.h"]
    FormHandler --> ContactService["/Users/rakesh/svn/cpp  
/web/serveit/src/api  
/serveit/Form/ContactService.h"]
    FormHandler --> CaptchaServlet["/Users/rakesh/svn/cpp  
/web/serveit/src/api  
/serveit/Form/CaptchaServlet.h"]
    FormHandler --> ContactService2["/Users/rakesh/svn/cpp  
/web/serveit/src/api  
/serveit/Form/ContactService.h"]
    VirtualHostServer --> RequestDispatcher1["/Users/rakesh/svn/cpp  
/web/serveit/src/api  
/serveit/RequestDispatcher.h"]
    VirtualHostServer --> RequestDispatcher2["/Users/rakesh/svn/cpp  
/web/serveit/src/api  
/serveit/RequestDispatcher.h"]
  
```

- class `spt::servlet::search::DocRootIndexer`

Namespaces

- Minnal C++ **Servlet** Container implementation.

- Namespace for classes that add support for full-text search indexing of content served by the server.*

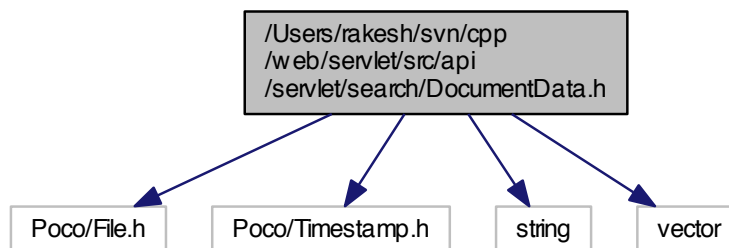
Typedefs

- typedef Poco::SharedPtr
< DocRootIndexer > [spt::servlet::search::DocRootIndexerPtr](#)

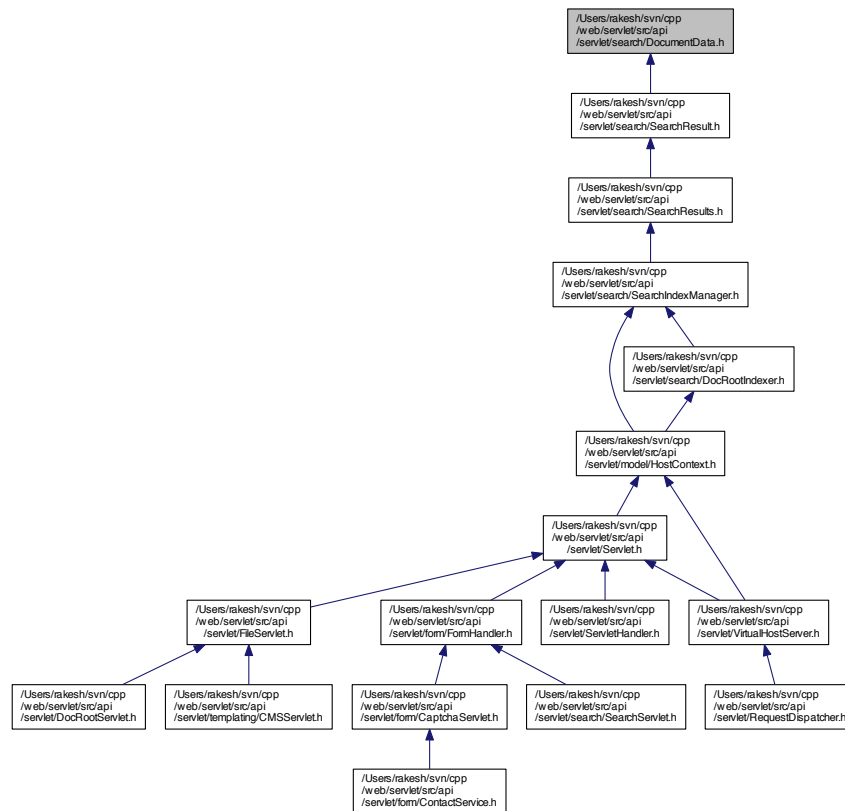
Shared pointer to a document root indexer instance.

8.31 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/DocumentData.h File Reference

```
#include <Poco/File.h>
#include <Poco/Timestamp.h>
#include <string>
#include <vector>
Include dependency graph for DocumentData.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [spt::servlet::search::DocumentData](#)

A simple POD used to store data relevant to a document indexed in the Xapian search engine.

Namespaces

- [spt](#)
- [spt::servlet](#)

Minnal C++ [Servlet](#) Container implementation.

- [spt::servlet::search](#)

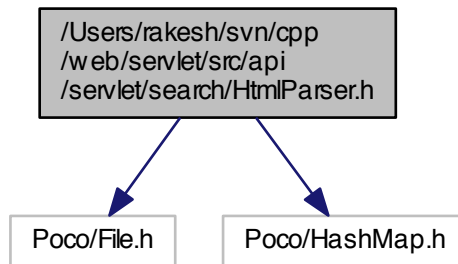
Namespace for classes that add support for full-text search indexing of content served by the server.

Functions

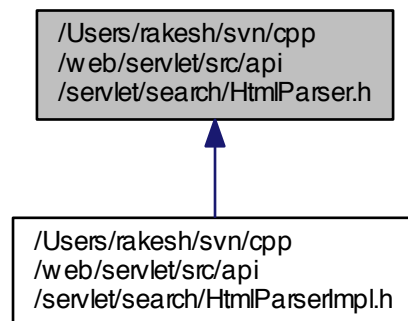
- bool [spt::servlet::search::operator==](#) (const DocumentData &lhs, const DocumentData &rhs)
Compare two document data instances for equality.
- bool [spt::servlet::search::operator<](#) (const DocumentData &lhs, const DocumentData &rhs)
Compare two document data instances for ordering. Compares by the file names, followed by modification timestamp if they have same name.
- bool [spt::servlet::search::indexable](#) (const std::string &contentType)
Common function used to determine if the specified content type is indexable by the search engine.

8.32 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/HtmlParser.h File Reference

```
#include <Poco/File.h>
#include <Poco/HashMap.h>
Include dependency graph for HtmlParser.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class `spt::servlet::search::HtmlParser`
A parser for extracting the plain text contents and some meta-data from a HTML string.

Namespaces

- `spt`
- `spt::servlet`
Minnal C++ *Servlet* Container implementation.

- [spt::servlet::search](#)

Namespace for classes that add support for full-text search indexing of content served by the server.

Functions

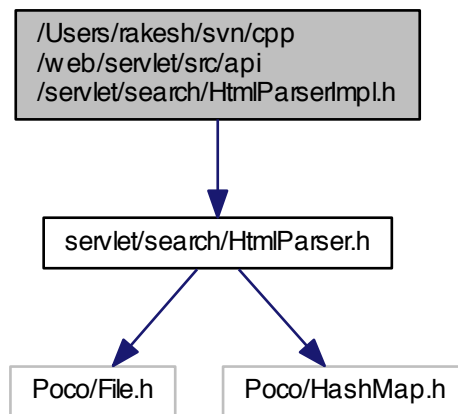
- void [spt::servlet::search::convertToUTF8](#) (std::string &text, const std::string &charset)

Convert the specified text in the specified charset to UTF-8.

8.33 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/HtmlParserImpl.h File Reference

```
#include <servlet/search/HtmlParser.h>
```

Include dependency graph for HtmlParserImpl.h:



Classes

- class [spt::servlet::search::HtmlParserImpl](#)

A more robust HTML parser implementation based on the Xapian Omega search engine project.

Namespaces

- [spt](#)
- [spt::servlet](#)

Minnal C++ [Servlet](#) Container implementation.

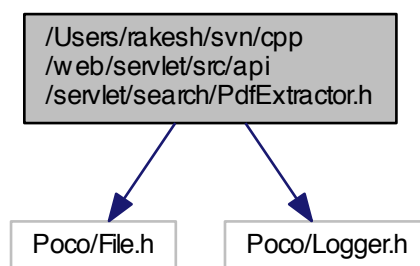
- [spt::servlet::search](#)

Namespace for classes that add support for full-text search indexing of content served by the server.

8.34 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/NamedEntities.h File Reference

8.35 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/PdfExtractor.h File Reference

```
#include <Poco/File.h>
#include <Poco/Logger.h>
Include dependency graph for PdfExtractor.h:
```



Classes

- class [spt::servlet::search::PdfExtractor](#)

Extract text content from PDF files using pdftotext (part of xpdf or the newer poppler system). Return the extracted plain text content.

Namespaces

- [spt](#)
- [spt::servlet](#)

Minnal C++ [Servlet](#) Container implementation.

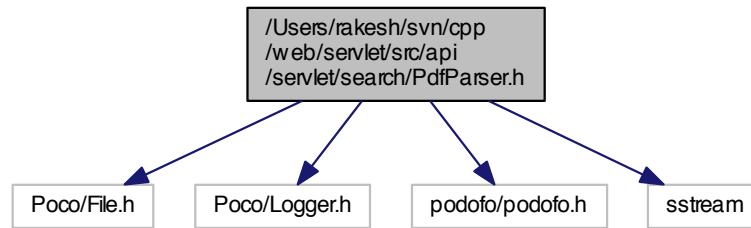
- [spt::servlet::search](#)

Namespace for classes that add support for full-text search indexing of content served by the server.

8.36 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/PdfParser.h File Reference

```
#include <Poco/File.h>
#include <Poco/Logger.h>
#include <podof/podof.h>
#include <sstream>
```

Include dependency graph for PdfParser.h:



Classes

- class [spt::servlet::search::PdfParser](#)

A parser for extracting the text contents of a PDF file using PoDoFo.

Namespaces

- [spt](#)
- [spt::servlet](#)

Minnal C++ [Servlet](#) Container implementation.

- [spt::servlet::search](#)

Namespace for classes that add support for full-text search indexing of content served by the server.

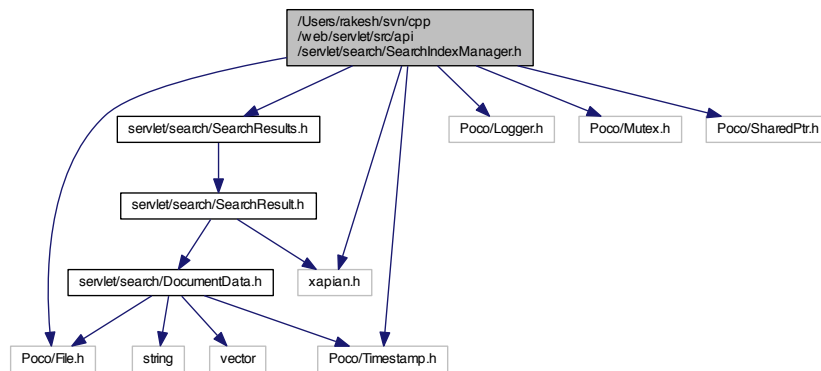
8.37 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/SearchIndexManager.h File Reference

```

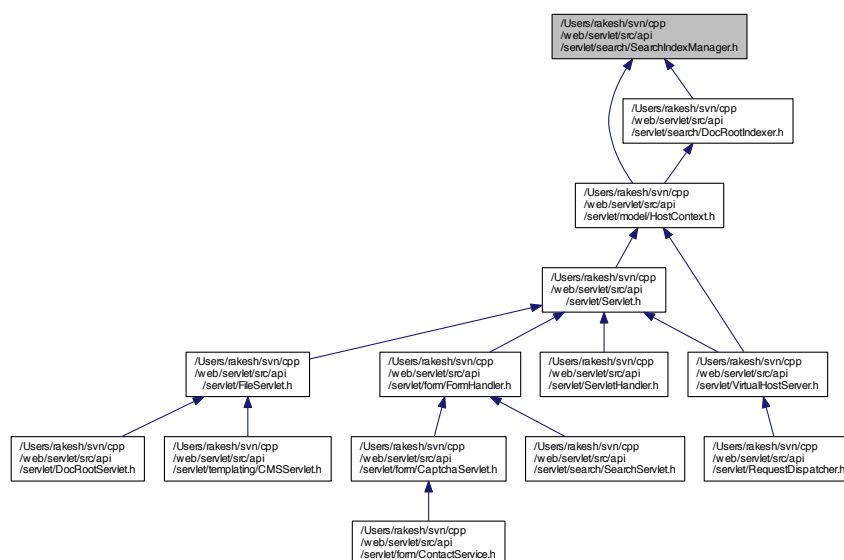
#include <servlet/search/SearchResults.h>
#include <Poco/File.h>
#include <Poco/Logger.h>
#include <Poco/Mutex.h>
#include <Poco/SharedPtr.h>
#include <Poco/TimeStamp.h>
#include <xapian.h>

```

Include dependency graph for SearchIndexManager.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [spt::servlet::search::SearchIndexManager](#)

A manager class for maintaining the Xapian search engine index. Usually used to manage indices for one site/virtual host.

Namespaces

- [spt](#)
- [spt::servlet](#)

Minnal C++ [Servlet](#) Container implementation.

- [spt::servlet::search](#)

Namespace for classes that add support for full-text search indexing of content served by the server.

Typedefs

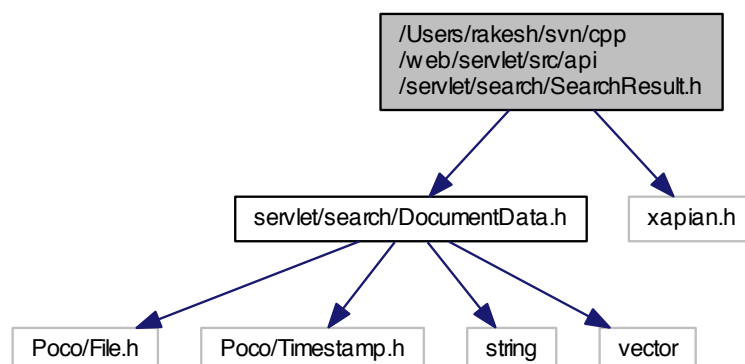
- typedef Poco::SharedPtr
< SearchIndexManager > [spt::servlet::search::SearchIndexManagerPtr](#)

Shared pointer to a search index manager.

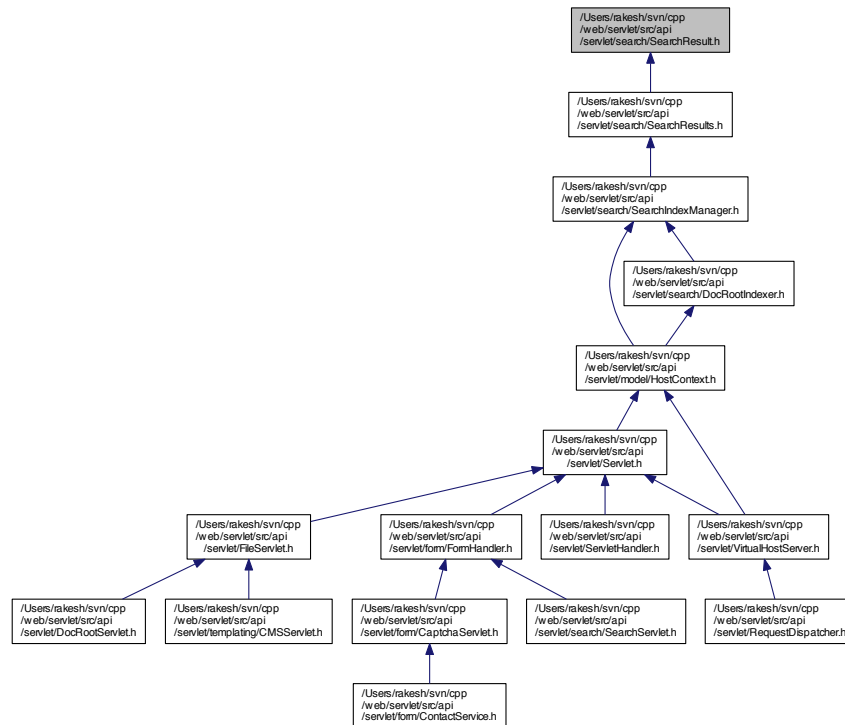
8.38 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/SearchResult.h File Reference

```
#include <servlet/search/DocumentData.h>  
#include <xapian.h>
```

Include dependency graph for SearchResult.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [spt::servlet::search::SearchResult](#)

A wrapper class that encapsulates a document that is part of the search results returned by Xapian for a query.

Namespaces

- [spt](#)
- [spt::servlet](#)

Minnal C++ [Servlet](#) Container implementation.

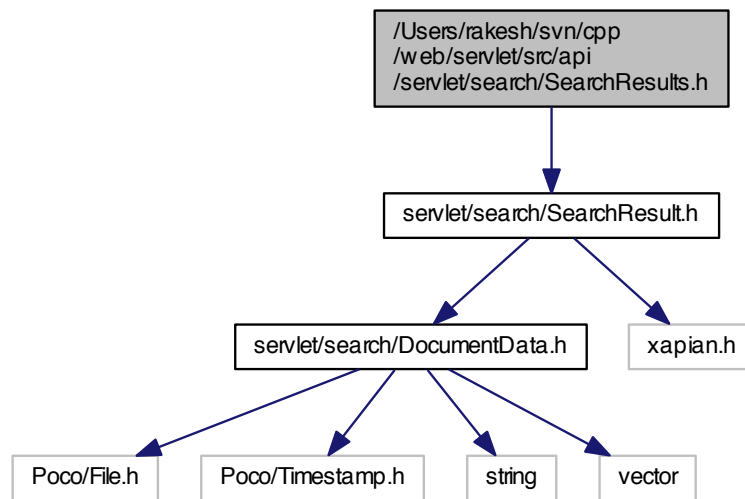
- [spt::servlet::search](#)

Namespace for classes that add support for full-text search indexing of content served by the server.

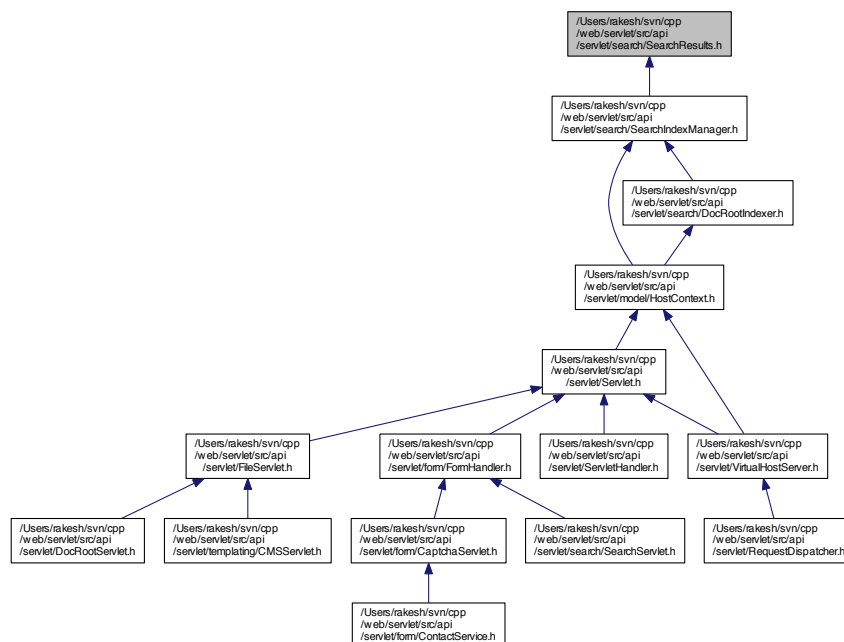
8.39 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/search/SearchResults.h File Reference

```
#include <servlet/search/SearchResult.h>
```


Include dependency graph for SearchResults.h:



This graph shows which files directly or indirectly include this file:

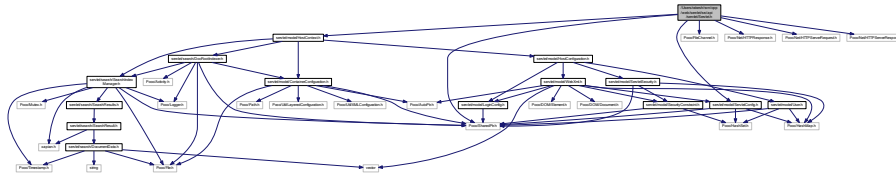


Classes

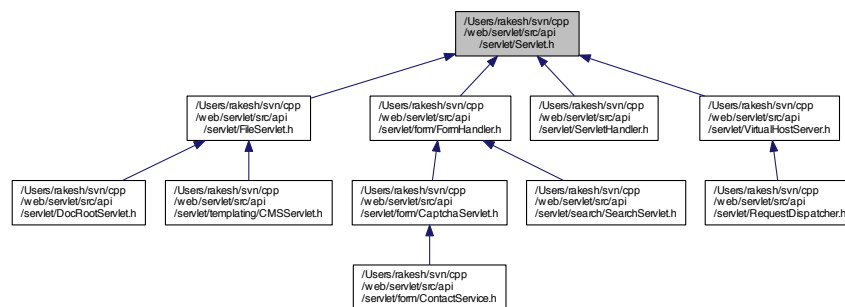
- class [spt::servlet::search::SearchResults](#)

A wrapper class that encapsulates the search results returned by Xapian for a query.

Include dependency graph for Servlet.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [spt::servlet::Servlet](#)

The base servlet class from which all concrete servlet instances are to be derived.

Namespaces

- [spt](#)
- [spt::servlet](#)

Minnal C++ [Servlet](#) Container implementation.

Typedefs

- typedef [Poco::SharedPtr< Servlet >](#) [spt::servlet::ServletPtr](#)

Shared pointer to a servlet instance.

8.42 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/ServletContainer.h File Reference

```
#include <servlet/model/ContainerConfiguration.h>
#include <Poco/Logger.h>
#include <Poco/Util/ServerApplication.h>
#include <Poco/Util/OptionSet.h>
```

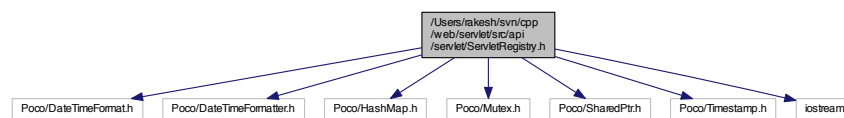

- [spt::servlet](#)

Minnal C++ [Servlet](#) Container implementation.

8.44 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/ServletRegistry.h File Reference

```
#include <Poco/DateTimeFormat.h>
#include <Poco/DateTimeFormatter.h>
#include <Poco/HashMap.h>
#include <Poco/Mutex.h>
#include <Poco/SharedPtr.h>
#include <Poco/Timestamp.h>
#include <iostream>
```

Include dependency graph for ServletRegistry.h:



Classes

- struct [spt::servlet::ServletHolder](#)

A base holder for creating servlet instances.

- class [spt::servlet::ServletHolderImpl< S >](#)

A template class that is used to register a servlet class. Used by the [REGISTER_SERVLET](#) macro to create a new instance and register by class name.

Namespaces

- [spt](#)
- [spt::servlet](#)

Minnal C++ [Servlet](#) Container implementation.

- [spt::servlet::ServletRegistry](#)

Namespace that provides functions for registering servlets. Servlets registered may be fetched from the registry by the fully qualified class name.

Macros

- `#define REGISTER\_SERVLET(className) static spt::servlet::ServletHolderImpl<className> t(#className);`

Macro to register a servlet with specified fully qualified class name.

Typedefs

- `typedef Poco::HashMap< std::string, ServletHolder * > spt::servlet::ServletRegistry::ServletMap`

Map of servlet class name to servlet holder instance.

Functions

- void [spt::servlet::ServletRegistry::writeTimestamp](#) ()
Utility method to output a formatted current timestamp to `std::cout`.
- ServletMap & [spt::servlet::ServletRegistry::getRegisteredServlets](#) ()
Return the map used to registered servlets.
- void [spt::servlet::ServletRegistry::registerServlet](#) (const std::string &className, ServletHolder *servlet)
Register a servlet with the specified fully qualified class name, and specified holder class that can be used to create new instances of the servlet.
- Poco::SharedPtr< Servlet > [spt::servlet::ServletRegistry::getServlet](#) (const std::string &className)
Return a pointer to the servlet that was registered with the specified fully qualified class name.
- void [spt::servlet::ServletRegistry::deRegisterServlet](#) (const std::string &className)
Use to clean up any resources held by the servlet. Normally used only during server shutdown, or reload process.

8.44.1 Macro Definition Documentation

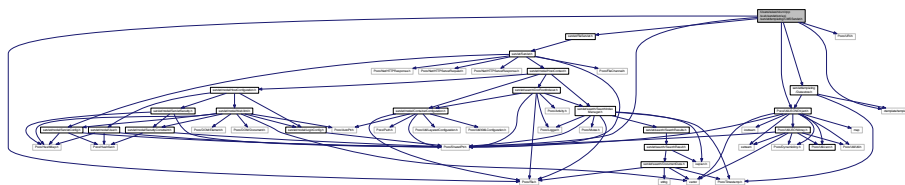
8.44.1.1 `#define REGISTER_SERVLET( className ) static spt::servlet::ServletHolderImpl<className> t(#className);`

Macro to register a servlet with specified fully qualified class name.

8.45 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/templating/CMSServlet.h File Reference

```
#include <servlet/FileServlet.h>
#include <servlet/templating/Datastore.h>
#include <Poco/File.h>
#include <Poco/SharedPtr.h>
#include <Poco/URI.h>
#include <Poco/Util/JSONObject.h>
#include <ctemplate/template.h>
```

Include dependency graph for CMSServlet.h:



Classes

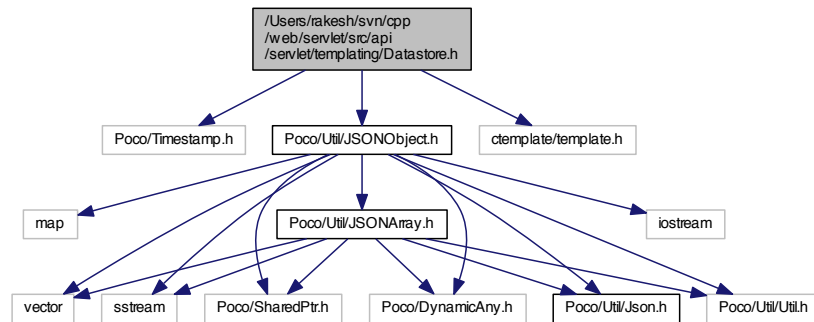
- class [spt::servlet::templating::CMSServlet](#)
The [Servlet](#) that builds dynamic output based on files with extension `cms` stored under the virtual host document root.

Namespaces

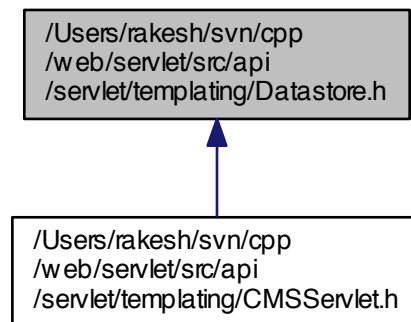
- [spt](#)
- [spt::servlet](#)
Minnal C++ [Servlet](#) Container implementation.
- [spt::servlet::templating](#)
Templating system for Minnal servlet container.

8.46 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/templating/Datastore.h File Reference

```
#include <Poco/Timestamp.h>
#include <Poco/Util/JSONObject.h>
#include <ctemplate/template.h>
Include dependency graph for Datastore.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [spt::servlet::templating::Datastore](#)

Factory class for interacting with backend systems used to store/retrieve data used to fill templates.

Namespaces

- [spt](#)
- [spt::servlet](#)

Minnal C++ [Servlet](#) Container implementation.

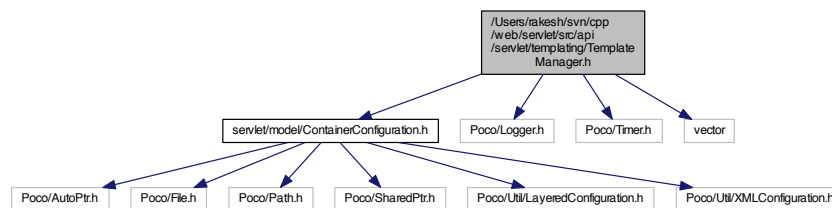
- [spt::servlet::templating](#)

Templating system for Minnal servlet container.

8.47 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/templating/TemplateManager.h File Reference

```
#include <servlet/model/ContainerConfiguration.h>
#include <Poco/Logger.h>
#include <Poco/Timer.h>
#include <vector>
```

Include dependency graph for TemplateManager.h:



Classes

- class [spt::servlet::templating::TemplateManager](#)

A manager class used to abstract interactions with the templating engine.

Namespaces

- [ctemplate](#)
- [spt](#)
- [spt::servlet](#)

Minnal C++ [Servlet](#) Container implementation.

- [spt::servlet::templating](#)

Templating system for Minnal servlet container.

8.48 /Users/rakesh/svn/cpp/web/servlet/src/api/servlet/VirtualHostServer.h File Reference

```
#include <servlet/Servlet.h>
#include <servlet/model/HostContext.h>
#include <Poco/HashMap.h>
#include <Poco/Logger.h>
#include <Poco/Mutex.h>
#include <functional>
#include <map>
```


Index

ctemplate, [13](#)

Poco, [13](#)

spt, [13](#)