

# RStudio Launcher Plugin SDK

99.9.9

Generated by Doxygen 1.8.16



<b>1 Hierarchical Index</b>	<b>1</b>
1.1 Class Hierarchy	1
<b>2 Class Index</b>	<b>5</b>
2.1 Class List	5
<b>3 File Index</b>	<b>11</b>
3.1 File List	11
<b>4 Class Documentation</b>	<b>13</b>
4.1 <a href="#">rstudio::launcher_plugins::system::process::AbstractChildProcess Class Reference</a>	13
4.1.1 Detailed Description	14
4.1.2 Constructor & Destructor Documentation	14
4.1.2.1 AbstractChildProcess()	14
4.1.3 Member Function Documentation	14
4.1.3.1 getPid()	14
4.1.3.2 run()	14
4.1.3.3 terminate()	15
4.1.3.4 writeToStdin()	15
4.2 <a href="#">rstudio::launcher_plugins::jobs::AbstractJobRepository Class Reference</a>	15
4.2.1 Detailed Description	16
4.2.2 Constructor & Destructor Documentation	16
4.2.2.1 AbstractJobRepository()	16
4.2.3 Member Function Documentation	16
4.2.3.1 addJob()	16
4.2.3.2 getJob()	17
4.2.3.3 getJobs()	17
4.2.3.4 initialize()	18
4.2.3.5 removeJob()	18
4.3 <a href="#">rstudio::launcher_plugins::jobs::AbstractJobStatusWatcher Class Reference</a>	18
4.3.1 Detailed Description	19
4.3.2 Constructor & Destructor Documentation	19
4.3.2.1 AbstractJobStatusWatcher()	19
4.3.3 Member Function Documentation	19
4.3.3.1 updateJobStatus()	19
4.4 <a href="#">rstudio::launcher_plugins::comms::AbstractLauncherCommunicator Class Reference</a>	20
4.4.1 Detailed Description	21
4.4.2 Constructor & Destructor Documentation	21
4.4.2.1 AbstractLauncherCommunicator()	21
4.4.3 Member Function Documentation	21
4.4.3.1 onDataReceived()	21
4.4.3.2 registerRequestHandler()	22
4.4.3.3 reportError()	22

4.4.3.4 <code>sendResponse()</code>	22
4.4.3.5 <code>shared_from_derived()</code>	23
4.4.3.6 <code>start()</code>	23
4.4.3.7 <code>stop()</code>	23
4.4.3.8 <code>waitForExit()</code>	23
4.5 <code>rstudio::launcher_plugins::AbstractMain</code> Class Reference	24
4.5.1 Detailed Description	24
4.5.2 Member Function Documentation	24
4.5.2.1 <code>run()</code>	24
4.6 <code>rstudio::launcher_plugins::api::AbstractMultiStream&lt; R, Args &gt;</code> Class Template Reference	25
4.6.1 Detailed Description	26
4.6.2 Constructor & Destructor Documentation	26
4.6.2.1 <code>AbstractMultiStream()</code>	26
4.6.3 Member Function Documentation	26
4.6.3.1 <code>addRequest()</code>	26
4.6.3.2 <code>initialize()</code>	28
4.6.3.3 <code>isEmpty()</code>	28
4.6.3.4 <code>onAddRequest()</code>	28
4.6.3.5 <code>onRemoveRequest()</code>	29
4.6.3.6 <code>removeRequest()</code>	29
4.6.3.7 <code>sendResponse()</code> [1/2]	29
4.6.3.8 <code>sendResponse()</code> [2/2]	30
4.6.4 Member Data Documentation	30
4.6.4.1 <code>m_mutex</code>	30
4.7 <code>rstudio::launcher_plugins::api::AbstractOutputStream</code> Class Reference	30
4.7.1 Detailed Description	31
4.7.2 Member Typedef Documentation	31
4.7.2.1 <code>OnComplete</code>	31
4.7.3 Constructor & Destructor Documentation	32
4.7.3.1 <code>AbstractOutputStream()</code>	32
4.7.4 Member Function Documentation	32
4.7.4.1 <code>reportData()</code>	32
4.7.4.2 <code>reportError()</code>	32
4.7.4.3 <code>start()</code>	34
4.7.5 Member Data Documentation	34
4.7.5.1 <code>m_job</code>	34
4.7.5.2 <code>m_outputType</code>	34
4.8 <code>rstudio::launcher_plugins::api::AbstractPluginApi</code> Class Reference	34
4.8.1 Detailed Description	35
4.8.2 Constructor & Destructor Documentation	35
4.8.2.1 <code>AbstractPluginApi()</code>	35
4.8.3 Member Function Documentation	35

4.8.3.1 initialize()	35
4.9 rstudio::launcher_plugins::api::AbstractResourceStream Class Reference	36
4.9.1 Detailed Description	37
4.9.2 Constructor & Destructor Documentation	37
4.9.2.1 AbstractResourceStream()	37
4.9.3 Member Function Documentation	37
4.9.3.1 addRequest()	37
4.9.3.2 initialize()	37
4.9.3.3 reportData()	38
4.9.3.4 reportError()	38
4.9.3.5 setStreamComplete()	38
4.9.4 Member Data Documentation	38
4.9.4.1 m_job	39
4.10 rstudio::launcher_plugins::jobs::AbstractTimedJobStatusWatcher Class Reference	39
4.10.1 Detailed Description	40
4.10.2 Constructor & Destructor Documentation	40
4.10.2.1 AbstractTimedJobStatusWatcher()	40
4.11 rstudio::launcher_plugins::api::AbstractTimedResourceStream Class Reference	40
4.11.1 Constructor & Destructor Documentation	41
4.11.1.1 AbstractTimedResourceStream()	41
4.11.2 Member Function Documentation	41
4.11.2.1 initialize()	41
4.12 rstudio::launcher_plugins::options::AbstractUserProfiles Class Reference	42
4.12.1 Detailed Description	42
4.12.2 Constructor & Destructor Documentation	43
4.12.2.1 AbstractUserProfiles() [1/2]	43
4.12.2.2 AbstractUserProfiles() [2/2]	43
4.12.3 Member Function Documentation	43
4.12.3.1 getValueForUser()	43
4.12.3.2 initialize()	44
4.12.3.3 isValueNotFoundError()	44
4.12.3.4 validateValue() [1/2]	45
4.12.3.5 validateValue() [2/2]	45
4.13 rstudio::launcher_plugins::json::Array Class Reference	46
4.13.1 Detailed Description	48
4.13.2 Constructor & Destructor Documentation	48
4.13.2.1 Array() [1/3]	48
4.13.2.2 Array() [2/3]	48
4.13.2.3 Array() [3/3]	49
4.13.3 Member Function Documentation	49
4.13.3.1 begin()	49
4.13.3.2 end()	49

4.13.3.3 erase() [1/2]	49
4.13.3.4 erase() [2/2]	50
4.13.3.5 getBack()	50
4.13.3.6 getFront()	50
4.13.3.7 getSize()	51
4.13.3.8 getValueAt()	51
4.13.3.9 isEmpty()	51
4.13.3.10 operator=() [1/2]	51
4.13.3.11 operator=() [2/2]	52
4.13.3.12 operator[]()	52
4.13.3.13 parse() [1/2]	53
4.13.3.14 parse() [2/2]	53
4.13.3.15 push_back() [1/12]	53
4.13.3.16 push_back() [2/12]	54
4.13.3.17 push_back() [3/12]	54
4.13.3.18 push_back() [4/12]	54
4.13.3.19 push_back() [5/12]	55
4.13.3.20 push_back() [6/12]	55
4.13.3.21 push_back() [7/12]	55
4.13.3.22 push_back() [8/12]	56
4.13.3.23 push_back() [9/12]	56
4.13.3.24 push_back() [10/12]	56
4.13.3.25 push_back() [11/12]	57
4.13.3.26 push_back() [12/12]	57
4.13.3.27 rbegin()	57
4.13.3.28 rend()	58
4.13.3.29 toSetString()	58
4.13.3.30 toStringPairList()	58
4.13.3.31 toVectorInt()	58
4.13.3.32 toVectorString()	59
4.14 rstudio::launcher_plugins::system::AsioService Class Reference	59
4.14.1 Detailed Description	60
4.14.2 Member Function Documentation	60
4.14.2.1 post()	60
4.14.2.2 setSignalHandler()	60
4.14.2.3 startThreads()	61
4.14.2.4 stop()	61
4.15 rstudio::launcher_plugins::system::AsioStream Class Reference	61
4.15.1 Detailed Description	62
4.15.2 Constructor & Destructor Documentation	62
4.15.2.1 AsioStream()	62
4.15.3 Member Function Documentation	62

4.15.3.1 readBytes()	62
4.15.3.2 writeBytes()	62
4.16 rstudio::launcher_plugins::system::AsyncDeadlineEvent Class Reference	63
4.16.1 Detailed Description	63
4.16.2 Constructor & Destructor Documentation	63
4.16.2.1 AsyncDeadlineEvent() [1/2]	63
4.16.2.2 AsyncDeadlineEvent() [2/2]	64
4.17 rstudio::launcher_plugins::system::process::AsyncProcessCallbacks Struct Reference	64
4.17.1 Detailed Description	65
4.18 rstudio::launcher_plugins::system::AsyncTimedEvent Class Reference	65
4.18.1 Detailed Description	65
4.18.2 Member Function Documentation	65
4.18.2.1 reportError()	65
4.18.2.2 start()	66
4.19 rstudio::launcher_plugins::api::AzureFileMountSource Struct Reference	66
4.19.1 Detailed Description	67
4.19.2 Member Function Documentation	67
4.19.2.1 fromJson()	67
4.19.2.2 getSecretName()	67
4.19.2.3 getShareName()	68
4.20 rstudio::launcher_plugins::api::BootstrapRequest Class Reference	68
4.20.1 Detailed Description	69
4.20.2 Member Function Documentation	69
4.20.2.1 getMajorVersion()	69
4.20.2.2 getMinorVersion()	69
4.20.2.3 getPatchNumber()	70
4.21 rstudio::launcher_plugins::api::BootstrapResponse Class Reference	70
4.21.1 Detailed Description	70
4.21.2 Constructor & Destructor Documentation	70
4.21.2.1 BootstrapResponse()	70
4.21.3 Member Function Documentation	71
4.21.3.1 toJson()	71
4.22 rstudio::launcher_plugins::api::CephFsMountSource Struct Reference	71
4.22.1 Detailed Description	72
4.22.2 Member Function Documentation	72
4.22.2.1 fromJson()	72
4.22.2.2 getMonitors()	73
4.22.2.3 getPath()	73
4.22.2.4 getSecretFile()	73
4.22.2.5 getSecretRef()	74
4.22.2.6 getUser()	74
4.23 rstudio::launcher_plugins::api::ClusterInfoResponse Class Reference	74

4.23.1 Detailed Description	75
4.23.2 Constructor & Destructor Documentation	75
4.23.2.1 ClusterInfoResponse()	75
4.23.3 Member Function Documentation	75
4.23.3.1 toJson()	75
4.24 rstudio::launcher_plugins::api::Container Struct Reference	76
4.24.1 Detailed Description	76
4.24.2 Member Function Documentation	76
4.24.2.1 fromJson()	76
4.24.2.2 toJson()	77
4.24.3 Member Data Documentation	77
4.24.3.1 Image	77
4.24.3.2 RunAsGroupId	77
4.24.3.3 RunAsUserId	77
4.24.3.4 SupplementalGroupIds	77
4.25 rstudio::launcher_plugins::api::ContainerConfiguration Struct Reference	78
4.25.1 Detailed Description	78
4.25.2 Member Data Documentation	78
4.25.2.1 AllowUnknownImages	78
4.25.2.2 ContainerImages	78
4.25.2.3 DefaultImage	78
4.25.2.4 SupportsContainers	79
4.26 rstudio::launcher_plugins::api::ControlJobRequest Class Reference	79
4.26.1 Detailed Description	80
4.26.2 Member Enumeration Documentation	80
4.26.2.1 Operation	80
4.26.3 Member Function Documentation	80
4.26.3.1 getOperation()	80
4.27 rstudio::launcher_plugins::api::ControlJobResponse Class Reference	81
4.27.1 Detailed Description	81
4.27.2 Constructor & Destructor Documentation	81
4.27.2.1 ControlJobResponse()	81
4.27.3 Member Function Documentation	82
4.27.3.1 toJson()	82
4.28 rstudio::launcher_plugins::system::DateTime Class Reference	82
4.28.1 Detailed Description	83
4.28.2 Constructor & Destructor Documentation	83
4.28.2.1 DateTime() [1/3]	83
4.28.2.2 DateTime() [2/3]	83
4.28.2.3 DateTime() [3/3]	84
4.28.3 Member Function Documentation	84
4.28.3.1 fromString()	84



4.28.3.2 operator!=(())	84
4.28.3.3 operator+()	85
4.28.3.4 operator+=(())	85
4.28.3.5 operator-() [1/2]	86
4.28.3.6 operator-() [2/2]	86
4.28.3.7 operator-=(())	86
4.28.3.8 operator<()	87
4.28.3.9 operator<=(())	87
4.28.3.10 operator=() [1/2]	87
4.28.3.11 operator=() [2/2]	88
4.28.3.12 operator==(())	88
4.28.3.13 operator>()	88
4.28.3.14 operator>=(())	89
4.28.3.15 toString() [1/3]	89
4.28.3.16 toString() [2/3]	89
4.28.3.17 toString() [3/3]	90
4.29 rstudio::launcher_plugins::Error Class Reference	90
4.29.1 Detailed Description	92
4.29.2 Constructor & Destructor Documentation	92
4.29.2.1 Error() [1/5]	92
4.29.2.2 Error() [2/5]	92
4.29.2.3 Error() [3/5]	93
4.29.2.4 Error() [4/5]	93
4.29.2.5 Error() [5/5]	93
4.29.3 Member Function Documentation	94
4.29.3.1 addOrUpdateProperty() [1/3]	94
4.29.3.2 addOrUpdateProperty() [2/3]	94
4.29.3.3 addOrUpdateProperty() [3/3]	94
4.29.3.4 addProperty() [1/3]	95
4.29.3.5 addProperty() [2/3]	95
4.29.3.6 addProperty() [3/3]	95
4.29.3.7 asString()	96
4.29.3.8 getCause()	96
4.29.3.9 getCode()	96
4.29.3.10 getLocation()	97
4.29.3.11 getMessage()	97
4.29.3.12 getName()	97
4.29.3.13 getProperties()	97
4.29.3.14 getProperty()	97
4.29.3.15 getSummary()	98
4.29.3.16 isExpected()	98
4.29.3.17 operator bool()	98

4.29.3.18 operator"!"()	99
4.29.3.19 operator"!="()	99
4.29.3.20 operator=="()	100
4.30 rstudio::launcher_plugins::ErrorLocation Class Reference	100
4.30.1 Detailed Description	101
4.30.2 Constructor & Destructor Documentation	101
4.30.2.1 ErrorLocation() [1/3]	101
4.30.2.2 ErrorLocation() [2/3]	101
4.30.2.3 ErrorLocation() [3/3]	102
4.30.3 Member Function Documentation	102
4.30.3.1 asString()	102
4.30.3.2 getFile()	102
4.30.3.3 getFunction()	103
4.30.3.4 getLine()	103
4.30.3.5 hasLocation()	103
4.30.3.6 operator=()	103
4.30.3.7 operator=="()	104
4.31 rstudio::launcher_plugins::ErrorLock Class Reference	104
4.31.1 Detailed Description	105
4.32 rstudio::launcher_plugins::api::ErrorResponse Class Reference	105
4.32.1 Detailed Description	105
4.32.2 Constructor & Destructor Documentation	106
4.32.2.1 ErrorResponse()	106
4.32.3 Member Function Documentation	106
4.32.3.1 toJson()	106
4.33 rstudio::launcher_plugins::api::ExposedPort Struct Reference	106
4.33.1 Detailed Description	107
4.33.2 Member Function Documentation	107
4.33.2.1 fromJson()	107
4.33.2.2 toJson()	107
4.33.3 Member Data Documentation	108
4.33.3.1 Protocol	108
4.33.3.2 PublishedPort	108
4.33.3.3 TargetPort	108
4.34 rstudio::launcher_plugins::logging::FileLogDestination Class Reference	108
4.34.1 Detailed Description	109
4.34.2 Constructor & Destructor Documentation	109
4.34.2.1 FileLogDestination()	109
4.34.3 Member Function Documentation	109
4.34.3.1 getId()	110
4.34.3.2 writeLog()	110
4.35 rstudio::launcher_plugins::logging::FileLogOptions Class Reference	110

4.35.1 Detailed Description	111
4.35.2 Constructor & Destructor Documentation	111
4.35.2.1 FileLogOptions() [1/2]	111
4.35.2.2 FileLogOptions() [2/2]	111
4.35.3 Member Function Documentation	112
4.35.3.1 doRotation()	112
4.35.3.2 getDirectory()	112
4.35.3.3 getFileMode()	112
4.35.3.4 getMaxSizeMb()	113
4.35.3.5 includePid()	113
4.36 rstudio::launcher_plugins::api::FileOutputStream Class Reference	113
4.36.1 Detailed Description	114
4.36.2 Constructor & Destructor Documentation	114
4.36.2.1 FileOutputStream()	114
4.36.3 Member Function Documentation	114
4.36.3.1 start()	114
4.37 rstudio::launcher_plugins::system::FilePath Class Reference	115
4.37.1 Detailed Description	118
4.37.2 Member Typedef Documentation	118
4.37.2.1 RecursiveIterationFunction	118
4.37.3 Member Enumeration Documentation	118
4.37.3.1 MoveType	118
4.37.4 Constructor & Destructor Documentation	119
4.37.4.1 FilePath()	119
4.37.5 Member Function Documentation	119
4.37.5.1 changeFileMode() [1/2]	119
4.37.5.2 changeFileMode() [2/2]	119
4.37.5.3 changeOwnership()	120
4.37.5.4 completeChildPath() [1/2]	120
4.37.5.5 completeChildPath() [2/2]	121
4.37.5.6 completePath()	121
4.37.5.7 copy()	121
4.37.5.8 copyDirectoryRecursive()	122
4.37.5.9 createAliasedPath()	122
4.37.5.10 createDirectory()	123
4.37.5.11 ensureDirectory()	123
4.37.5.12 ensureFile()	123
4.37.5.13 exists() [1/2]	123
4.37.5.14 exists() [2/2]	123
4.37.5.15 getAbsolutePath()	124
4.37.5.16 getAbsolutePathNative()	124
4.37.5.17 getCanonicalPath()	124

4.37.5.18	<a href="#">getChildren()</a>	124
4.37.5.19	<a href="#">getChildrenRecursive()</a>	125
4.37.5.20	<a href="#">getExtension()</a>	125
4.37.5.21	<a href="#">getExtensionLowerCase()</a>	125
4.37.5.22	<a href="#">getFileMode()</a>	126
4.37.5.23	<a href="#">getFilename()</a>	127
4.37.5.24	<a href="#">getLastWriteTime()</a>	127
4.37.5.25	<a href="#">getLexicallyNormalPath()</a>	127
4.37.5.26	<a href="#">getMimeType()</a>	127
4.37.5.27	<a href="#">getParent()</a>	128
4.37.5.28	<a href="#">getRelativePath()</a>	128
4.37.5.29	<a href="#">getSize()</a>	128
4.37.5.30	<a href="#">getSizeRecursive()</a>	129
4.37.5.31	<a href="#">getStem()</a>	129
4.37.5.32	<a href="#">hasExtension()</a>	129
4.37.5.33	<a href="#">hasExtensionLowerCase()</a>	129
4.37.5.34	<a href="#">hasTextMimeType()</a>	130
4.37.5.35	<a href="#">isDirectory()</a>	130
4.37.5.36	<a href="#">isEmpty()</a>	130
4.37.5.37	<a href="#">isEqualCaseInsensitive()</a>	130
4.37.5.38	<a href="#">isEquivalentTo()</a>	131
4.37.5.39	<a href="#">isHidden()</a>	131
4.37.5.40	<a href="#">isReadable()</a>	131
4.37.5.41	<a href="#">isRegularFile()</a>	132
4.37.5.42	<a href="#">isRootPath()</a>	132
4.37.5.43	<a href="#">isSymlink()</a>	132
4.37.5.44	<a href="#">isWithin()</a>	133
4.37.5.45	<a href="#">isWriteable()</a>	133
4.37.5.46	<a href="#">makeCurrent()</a>	133
4.37.5.47	<a href="#">makeCurrentPath()</a>	134
4.37.5.48	<a href="#">move()</a>	134
4.37.5.49	<a href="#">moveIndirect()</a>	134
4.37.5.50	<a href="#">openForRead()</a>	135
4.37.5.51	<a href="#">openForWrite()</a>	135
4.37.5.52	<a href="#">operator"!="()</a>	136
4.37.5.53	<a href="#">operator&lt;()</a>	136
4.37.5.54	<a href="#">operator==()</a>	136
4.37.5.55	<a href="#">remove()</a>	137
4.37.5.56	<a href="#">removeIfExists()</a>	137
4.37.5.57	<a href="#">resetDirectory()</a>	137
4.37.5.58	<a href="#">resolveAliasedPath()</a>	137
4.37.5.59	<a href="#">resolveSymlink()</a>	138

4.37.5.60	safeCurrentPath()	138
4.37.5.61	setLastWriteTime()	138
4.37.5.62	tempFilePath() [1/2]	139
4.37.5.63	tempFilePath() [2/2]	139
4.37.5.64	testWritePermissions()	139
4.37.5.65	uniqueFilePath() [1/2]	140
4.37.5.66	uniqueFilePath() [2/2]	140
4.38	rstudio::launcher_plugins::api::GlusterFsMountSource Struct Reference	141
4.38.1	Member Function Documentation	141
4.38.1.1	fromJson()	141
4.38.1.2	getEndpoints()	142
4.38.1.3	getPath()	142
4.39	rstudio::launcher_plugins::api::HeartbeatResponse Class Reference	142
4.39.1	Detailed Description	143
4.40	rstudio::launcher_plugins::api::HostMountSource Struct Reference	143
4.40.1	Detailed Description	144
4.40.2	Member Function Documentation	144
4.40.2.1	fromJson()	144
4.40.2.2	getPath()	144
4.41	rstudio::launcher_plugins::api::IJobSource Class Reference	145
4.41.1	Detailed Description	146
4.41.2	Constructor & Destructor Documentation	146
4.41.2.1	IJobSource()	146
4.41.3	Member Function Documentation	146
4.41.3.1	cancelJob()	146
4.41.3.2	createOutputStream()	147
4.41.3.3	createResourceStream()	147
4.41.3.4	getConfiguration()	148
4.41.3.5	getNetworkInfo()	148
4.41.3.6	initialize()	149
4.41.3.7	killJob()	149
4.41.3.8	resumeJob()	150
4.41.3.9	stopJob()	150
4.41.3.10	submitJob()	151
4.41.3.11	suspendJob()	151
4.41.4	Member Data Documentation	152
4.41.4.1	m_jobRepository	152
4.41.4.2	m_jobStatusNotifier	152
4.42	rstudio::launcher_plugins::logging::ILogDestination Class Reference	152
4.42.1	Detailed Description	153
4.42.2	Constructor & Destructor Documentation	153
4.42.2.1	ILogDestination()	153

4.42.3 Member Function Documentation	153
4.42.3.1 getId()	154
4.42.3.2 getLogLevel()	154
4.42.3.3 writeLog()	154
4.43 rstudio::launcher_plugins::options::Options::Init Class Reference	154
4.43.1 Detailed Description	155
4.43.2 Constructor & Destructor Documentation	155
4.43.2.1 Init()	155
4.43.3 Member Function Documentation	155
4.43.3.1 operator>() [1/2]	155
4.43.3.2 operator>() [2/2]	156
4.44 rstudio::launcher_plugins::system::posix::IpAddress Struct Reference	157
4.44.1 Detailed Description	157
4.44.2 Member Data Documentation	157
4.44.2.1 Address	157
4.44.2.2 Name	157
4.45 rstudio::launcher_plugins::json::detail::is_json_type< T > Struct Template Reference	157
4.45.1 Detailed Description	158
4.46 rstudio::launcher_plugins::json::Object::Iterator Class Reference	158
4.46.1 Detailed Description	159
4.46.2 Constructor & Destructor Documentation	159
4.46.2.1 Iterator() [1/2]	159
4.46.2.2 Iterator() [2/2]	159
4.46.3 Member Function Documentation	159
4.46.3.1 operator!=(())	159
4.46.3.2 operator*()	160
4.46.3.3 operator++() [1/2]	160
4.46.3.4 operator++() [2/2]	160
4.46.3.5 operator--() [1/2]	161
4.46.3.6 operator--() [2/2]	161
4.46.3.7 operator=()	161
4.46.3.8 operator==(())	161
4.47 rstudio::launcher_plugins::json::Array::Iterator Class Reference	162
4.47.1 Detailed Description	163
4.47.2 Constructor & Destructor Documentation	163
4.47.2.1 Iterator() [1/2]	163
4.47.2.2 Iterator() [2/2]	163
4.47.3 Member Function Documentation	163
4.47.3.1 operator!=(())	163
4.47.3.2 operator*()	164
4.47.3.3 operator++() [1/2]	164
4.47.3.4 operator++() [2/2]	164

4.47.3.5 operator--() [1/2]	164
4.47.3.6 operator--() [2/2]	165
4.47.3.7 operator=()	165
4.47.3.8 operator==()	165
4.48 rstudio::launcher_plugins::api::Job Struct Reference	165
4.48.1 Detailed Description	167
4.48.2 Member Enumeration Documentation	167
4.48.2.1 State	167
4.48.3 Constructor & Destructor Documentation	168
4.48.3.1 Job() [1/2]	168
4.48.3.2 Job() [2/2]	168
4.48.4 Member Function Documentation	168
4.48.4.1 fromJson()	168
4.48.4.2 getJobConfigValue()	169
4.48.4.3 isCompleted()	169
4.48.4.4 matchesTags()	169
4.48.4.5 operator=() [1/2]	170
4.48.4.6 operator=() [2/2]	170
4.48.4.7 stateFromString()	170
4.48.4.8 stateToString()	171
4.48.4.9 toJson()	171
4.48.5 Member Data Documentation	171
4.48.5.1 Arguments	171
4.48.5.2 Cluster	172
4.48.5.3 Command	172
4.48.5.4 Config	172
4.48.5.5 ContainerDetails	172
4.48.5.6 Environment	172
4.48.5.7 Exe	172
4.48.5.8 ExitCode	173
4.48.5.9 ExposedPorts	173
4.48.5.10 Host	173
4.48.5.11 Id	173
4.48.5.12 LastUpdateTime	173
4.48.5.13 Mounts	173
4.48.5.14 Name	173
4.48.5.15 Pid	173
4.48.5.16 PlacementConstraints	174
4.48.5.17 Queues	174
4.48.5.18 ResourceLimits	174
4.48.5.19 StandardErrFile	174
4.48.5.20 StandardIn	174

4.48.5.21 StandardOutFile . . . . .	174
4.48.5.22 Status . . . . .	174
4.48.5.23 StatusMessage . . . . .	174
4.48.5.24 SubmissionTime . . . . .	175
4.48.5.25 Tags . . . . .	175
4.48.5.26 User . . . . .	175
4.48.5.27 WorkingDirectory . . . . .	175
4.49 rstudio::launcher_plugins::api::JobConfig Struct Reference . . . . .	175
4.49.1 Detailed Description . . . . .	176
4.49.2 Member Enumeration Documentation . . . . .	176
4.49.2.1 Type . . . . .	176
4.49.3 Constructor & Destructor Documentation . . . . .	176
4.49.3.1 JobConfig() . . . . .	176
4.49.4 Member Function Documentation . . . . .	177
4.49.4.1 fromJson() . . . . .	177
4.49.4.2 toJson() . . . . .	177
4.49.5 Member Data Documentation . . . . .	177
4.49.5.1 Name . . . . .	177
4.49.5.2 Value . . . . .	178
4.49.5.3 ValueType . . . . .	178
4.50 rstudio::launcher_plugins::api::JobIdRequest Class Reference . . . . .	178
4.50.1 Detailed Description . . . . .	178
4.50.2 Constructor & Destructor Documentation . . . . .	179
4.50.2.1 JobIdRequest() . . . . .	179
4.50.3 Member Function Documentation . . . . .	180
4.50.3.1 getEncodedJobId() . . . . .	180
4.50.3.2 getJobId() . . . . .	180
4.51 rstudio::launcher_plugins::api::JobLock Class Reference . . . . .	180
4.51.1 Detailed Description . . . . .	181
4.51.2 Constructor & Destructor Documentation . . . . .	181
4.51.2.1 JobLock() [1/2] . . . . .	181
4.51.2.2 JobLock() [2/2] . . . . .	181
4.52 rstudio::launcher_plugins::api::JobSourceConfiguration Struct Reference . . . . .	182
4.52.1 Detailed Description . . . . .	182
4.52.2 Member Data Documentation . . . . .	182
4.52.2.1 ContainerConfig . . . . .	182
4.52.2.2 CustomConfig . . . . .	182
4.52.2.3 PlacementConstraints . . . . .	182
4.52.2.4 Queues . . . . .	183
4.52.2.5 ResourceLimits . . . . .	183
4.53 rstudio::launcher_plugins::api::JobStateRequest Class Reference . . . . .	183
4.53.1 Detailed Description . . . . .	184



4.53.2 Member Function Documentation	184
4.53.2.1 getEndTime()	184
4.53.2.2 getFieldSet()	184
4.53.2.3 getStartTime()	185
4.53.2.4 getStatusSet()	185
4.53.2.5 getTagSet()	185
4.54 rstudio::launcher_plugins::api::JobStateResponse Class Reference	186
4.54.1 Detailed Description	186
4.54.2 Constructor & Destructor Documentation	186
4.54.2.1 JobStateResponse()	186
4.54.3 Member Function Documentation	187
4.54.3.1 toJson()	187
4.55 rstudio::launcher_plugins::jobs::JobStatusNotifier Class Reference	187
4.55.1 Detailed Description	188
4.55.2 Member Function Documentation	188
4.55.2.1 subscribe() [1/2]	188
4.55.2.2 subscribe() [2/2]	188
4.55.2.3 updateJob()	190
4.56 rstudio::launcher_plugins::api::JobStatusRequest Class Reference	190
4.56.1 Detailed Description	191
4.56.2 Member Function Documentation	191
4.56.2.1 isCancelRequest()	191
4.57 rstudio::launcher_plugins::api::JobStatusResponse Class Reference	192
4.57.1 Detailed Description	192
4.57.2 Constructor & Destructor Documentation	192
4.57.2.1 JobStatusResponse()	192
4.57.3 Member Function Documentation	193
4.57.3.1 toJson()	193
4.58 rstudio::launcher_plugins::local::LocalJobRepository Class Reference	193
4.58.1 Detailed Description	194
4.58.2 Constructor & Destructor Documentation	194
4.58.2.1 LocalJobRepository()	194
4.58.3 Member Function Documentation	194
4.58.3.1 saveJob()	194
4.58.3.2 setJobOutputPaths()	194
4.59 rstudio::launcher_plugins::local::LocalJobRunner Class Reference	195
4.59.1 Detailed Description	195
4.59.2 Constructor & Destructor Documentation	195
4.59.2.1 LocalJobRunner()	195
4.59.3 Member Function Documentation	196
4.59.3.1 initialize()	196
4.59.3.2 runJob()	196

4.60 rstudio::launcher_plugins::local::LocalJobSource Class Reference . . . . .	197
4.60.1 Detailed Description . . . . .	197
4.60.2 Constructor & Destructor Documentation . . . . .	198
4.60.2.1 LocalJobSource() . . . . .	198
4.60.3 Member Function Documentation . . . . .	198
4.60.3.1 cancelJob() . . . . .	198
4.60.3.2 createOutputStream() . . . . .	199
4.60.3.3 createResourceStream() . . . . .	199
4.60.3.4 getConfiguration() . . . . .	200
4.60.3.5 getNetworkInfo() . . . . .	200
4.60.3.6 initialize() . . . . .	200
4.60.3.7 killJob() . . . . .	201
4.60.3.8 resumeJob() . . . . .	201
4.60.3.9 stopJob() . . . . .	202
4.60.3.10 submitJob() . . . . .	202
4.60.3.11 suspendJob() . . . . .	203
4.61 rstudio::launcher_plugins::local::LocalOptions Class Reference . . . . .	203
4.61.1 Detailed Description . . . . .	204
4.61.2 Member Function Documentation . . . . .	204
4.61.2.1 getInstance() . . . . .	204
4.61.2.2 getNodeConnectionTimeoutSeconds() . . . . .	204
4.61.2.3 getSecureCookieKeyFile() . . . . .	205
4.61.2.4 initialize() . . . . .	205
4.61.2.5 shouldSaveUnspecifiedOutput() . . . . .	205
4.62 rstudio::launcher_plugins::local::LocalPluginApi Class Reference . . . . .	205
4.62.1 Detailed Description . . . . .	206
4.62.2 Constructor & Destructor Documentation . . . . .	206
4.62.2.1 LocalPluginApi() . . . . .	206
4.63 rstudio::launcher_plugins::local::LocalResourceStream Class Reference . . . . .	206
4.63.1 Constructor & Destructor Documentation . . . . .	207
4.63.1.1 LocalResourceStream() . . . . .	207
4.64 rstudio::launcher_plugins::local::LocalSecureCookie Class Reference . . . . .	207
4.64.1 Detailed Description . . . . .	208
4.64.2 Member Function Documentation . . . . .	208
4.64.2.1 getKey() . . . . .	208
4.64.2.2 initialize() . . . . .	208
4.65 rstudio::launcher_plugins::json::Object::Member Class Reference . . . . .	208
4.65.1 Detailed Description . . . . .	209
4.65.2 Constructor & Destructor Documentation . . . . .	209
4.65.2.1 Member() . . . . .	209
4.65.3 Member Function Documentation . . . . .	209
4.65.3.1 getName() . . . . .	209

4.65.3.2 <code>getValue()</code> . . . . .	209
4.66 <code>rstudio::launcher_plugins::api::Mount</code> Struct Reference . . . . .	210
4.66.1 Detailed Description . . . . .	210
4.66.2 Member Function Documentation . . . . .	210
4.66.2.1 <code>fromJson()</code> . . . . .	210
4.66.2.2 <code>toJson()</code> . . . . .	211
4.66.3 Member Data Documentation . . . . .	211
4.66.3.1 Destination . . . . .	211
4.66.3.2 <code>IsReadOnly</code> . . . . .	211
4.66.3.3 Source . . . . .	211
4.67 <code>rstudio::launcher_plugins::api::MountSource</code> Struct Reference . . . . .	212
4.67.1 Detailed Description . . . . .	213
4.67.2 Member Enumeration Documentation . . . . .	213
4.67.2.1 Type . . . . .	213
4.67.3 Member Function Documentation . . . . .	214
4.67.3.1 <code>asAzureFileMountSource()</code> [1/2] . . . . .	214
4.67.3.2 <code>asAzureFileMountSource()</code> [2/2] . . . . .	214
4.67.3.3 <code>asCephFsMountSource()</code> [1/2] . . . . .	214
4.67.3.4 <code>asCephFsMountSource()</code> [2/2] . . . . .	215
4.67.3.5 <code>asGlusterFsMountSource()</code> [1/2] . . . . .	215
4.67.3.6 <code>asGlusterFsMountSource()</code> [2/2] . . . . .	215
4.67.3.7 <code>asHostMountSource()</code> [1/2] . . . . .	216
4.67.3.8 <code>asHostMountSource()</code> [2/2] . . . . .	216
4.67.3.9 <code>asNfsMountSource()</code> [1/2] . . . . .	216
4.67.3.10 <code>asNfsMountSource()</code> [2/2] . . . . .	217
4.67.3.11 <code>fromJson()</code> . . . . .	217
4.67.3.12 <code>isAzureFileMountSource()</code> . . . . .	217
4.67.3.13 <code>isCephFsMountSource()</code> . . . . .	218
4.67.3.14 <code>isGlusterFsMountSource()</code> . . . . .	218
4.67.3.15 <code>isHostMountSource()</code> . . . . .	218
4.67.3.16 <code>isNfsMountSource()</code> . . . . .	218
4.67.3.17 <code>isPassthroughMountSource()</code> . . . . .	219
4.67.3.18 <code>toJson()</code> . . . . .	219
4.67.4 Member Data Documentation . . . . .	219
4.67.4.1 <code>CustomType</code> . . . . .	219
4.67.4.2 <code>SourceObject</code> . . . . .	219
4.67.4.3 <code>SourceType</code> . . . . .	219
4.68 <code>rstudio::launcher_plugins::api::MultiStreamResponse</code> Class Reference . . . . .	220
4.68.1 Detailed Description . . . . .	220
4.68.2 Constructor & Destructor Documentation . . . . .	220
4.68.2.1 <code>MultiStreamResponse()</code> . . . . .	220
4.68.3 Member Function Documentation . . . . .	221

4.68.3.1 toJson()	221
4.69 rstudio::launcher_plugins::api::NetworkInfo Struct Reference	221
4.69.1 Detailed Description	221
4.69.2 Member Data Documentation	222
4.69.2.1 Hostname	222
4.69.2.2 IpAddresses	222
4.70 rstudio::launcher_plugins::api::NetworkRequest Class Reference	222
4.70.1 Detailed Description	223
4.71 rstudio::launcher_plugins::api::NetworkResponse Class Reference	223
4.71.1 Detailed Description	223
4.71.2 Constructor & Destructor Documentation	223
4.71.2.1 NetworkResponse()	223
4.71.3 Member Function Documentation	224
4.71.3.1 toJson()	224
4.72 rstudio::launcher_plugins::api::NfsMountSource Struct Reference	224
4.72.1 Detailed Description	225
4.72.2 Member Function Documentation	225
4.72.2.1 fromJson()	225
4.72.2.2 getHost()	225
4.72.2.3 getPath()	226
4.73 rstudio::launcher_plugins::Noncopyable Class Reference	226
4.73.1 Detailed Description	227
4.74 rstudio::launcher_plugins::json::Object Class Reference	227
4.74.1 Detailed Description	230
4.74.2 Constructor & Destructor Documentation	230
4.74.2.1 Object() [1/3]	230
4.74.2.2 Object() [2/3]	231
4.74.2.3 Object() [3/3]	231
4.74.3 Member Function Documentation	231
4.74.3.1 begin()	231
4.74.3.2 createMember()	231
4.74.3.3 end()	232
4.74.3.4 erase() [1/3]	232
4.74.3.5 erase() [2/3]	232
4.74.3.6 erase() [3/3]	233
4.74.3.7 find() [1/2]	233
4.74.3.8 find() [2/2]	233
4.74.3.9 getSchemaDefaults()	234
4.74.3.10 getSize()	234
4.74.3.11 hasMember() [1/2]	234
4.74.3.12 hasMember() [2/2]	235
4.74.3.13 insert() [1/13]	235

4.74.3.14 insert() [2/13]	235
4.74.3.15 insert() [3/13]	236
4.74.3.16 insert() [4/13]	236
4.74.3.17 insert() [5/13]	236
4.74.3.18 insert() [6/13]	237
4.74.3.19 insert() [7/13]	237
4.74.3.20 insert() [8/13]	237
4.74.3.21 insert() [9/13]	238
4.74.3.22 insert() [10/13]	238
4.74.3.23 insert() [11/13]	238
4.74.3.24 insert() [12/13]	239
4.74.3.25 insert() [13/13]	239
4.74.3.26 isEmpty()	239
4.74.3.27 mergeObjects()	240
4.74.3.28 operator=() [1/2]	240
4.74.3.29 operator=() [2/2]	240
4.74.3.30 operator[]() [1/2]	241
4.74.3.31 operator[]() [2/2]	241
4.74.3.32 parse() [1/2]	241
4.74.3.33 parse() [2/2]	242
4.74.3.34 rbegin()	242
4.74.3.35 rend()	243
4.74.3.36 toStringMap()	243
4.74.3.37 toStringPairList()	243
4.75 rstudio::launcher_plugins::Optional< T > Class Template Reference	244
4.75.1 Detailed Description	244
4.75.2 Constructor & Destructor Documentation	245
4.75.2.1 Optional() [1/4]	245
4.75.2.2 Optional() [2/4]	245
4.75.2.3 Optional() [3/4]	245
4.75.2.4 Optional() [4/4]	246
4.75.3 Member Function Documentation	246
4.75.3.1 getValueOr() [1/2]	246
4.75.3.2 getValueOr() [2/2]	246
4.75.3.3 hasValue()	247
4.75.3.4 operator bool()	247
4.75.3.5 operator"!()	247
4.75.3.6 operator=() [1/4]	247
4.75.3.7 operator=() [2/4]	248
4.75.3.8 operator=() [3/4]	248
4.75.3.9 operator=() [4/4]	249
4.76 rstudio::launcher_plugins::options::Options Class Reference	249

4.76.1 Detailed Description	250
4.76.2 Member Function Documentation	250
4.76.2.1 enableDebugLogging()	251
4.76.2.2 getHeartbeatIntervalSeconds()	251
4.76.2.3 getInstance()	251
4.76.2.4 getJobExpiryHours()	251
4.76.2.5 getLauncherConfigFile()	252
4.76.2.6 getLoggingDir()	252
4.76.2.7 getLogLevel()	252
4.76.2.8 getMaxMessageSize()	252
4.76.2.9 getPluginName()	253
4.76.2.10 getRSandboxPath()	253
4.76.2.11 getScratchPath()	253
4.76.2.12 getServerUser()	253
4.76.2.13 getThreadPoolSize()	254
4.76.2.14 readOptions()	254
4.76.2.15 registerOptions()	254
4.76.2.16 useUnprivilegedMode()	255
4.77 rstudio::launcher_plugins::api::OutputStreamRequest Class Reference	255
4.77.1 Detailed Description	256
4.77.2 Member Function Documentation	256
4.77.2.1 getStreamType()	256
4.77.2.2 isCancelRequest()	256
4.78 rstudio::launcher_plugins::api::OutputStreamResponse Class Reference	257
4.78.1 Detailed Description	257
4.78.2 Constructor & Destructor Documentation	257
4.78.2.1 OutputStreamResponse() [1/2]	257
4.78.2.2 OutputStreamResponse() [2/2]	258
4.78.3 Member Function Documentation	258
4.78.3.1 toJson()	258
4.79 rstudio::launcher_plugins::system::PathScopeImplDeleter Struct Reference	259
4.79.1 Detailed Description	259
4.80 rstudio::launcher_plugins::api::PlacementConstraint Struct Reference	259
4.80.1 Detailed Description	260
4.80.2 Constructor & Destructor Documentation	260
4.80.2.1 PlacementConstraint() [1/2]	260
4.80.2.2 PlacementConstraint() [2/2]	260
4.80.3 Member Function Documentation	261
4.80.3.1 fromJson()	261
4.80.3.2 toJson()	261
4.80.4 Member Data Documentation	261
4.80.4.1 Name	261

4.80.4.2 Value . . . . .	262
4.81 rstudio::launcher_plugins::system::process::ProcessInfo Struct Reference . . . . .	262
4.81.1 Detailed Description . . . . .	262
4.81.2 Member Function Documentation . . . . .	262
4.81.2.1 getProcessInfo() . . . . .	262
4.81.3 Member Data Documentation . . . . .	263
4.81.3.1 Arguments . . . . .	263
4.81.3.2 Executable . . . . .	263
4.81.3.3 Owner . . . . .	263
4.81.3.4 PGrp . . . . .	263
4.81.3.5 Pid . . . . .	263
4.81.3.6 PPid . . . . .	264
4.81.3.7 State . . . . .	264
4.82 rstudio::launcher_plugins::system::process::ProcessOptions Struct Reference . . . . .	264
4.82.1 Detailed Description . . . . .	265
4.82.2 Member Data Documentation . . . . .	265
4.82.2.1 CloseStdIn . . . . .	265
4.82.2.2 Mounts . . . . .	265
4.82.2.3 PamProfile . . . . .	265
4.82.2.4 Password . . . . .	266
4.82.2.5 RunAsUser . . . . .	266
4.82.2.6 UseSandbox . . . . .	266
4.83 rstudio::launcher_plugins::system::process::ProcessResult Struct Reference . . . . .	266
4.83.1 Detailed Description . . . . .	267
4.83.2 Member Data Documentation . . . . .	267
4.83.2.1 ExitCode . . . . .	267
4.83.2.2 StdError . . . . .	267
4.83.2.3 StdOut . . . . .	267
4.84 rstudio::launcher_plugins::system::process::ProcessSupervisor Class Reference . . . . .	267
4.84.1 Detailed Description . . . . .	268
4.84.2 Member Function Documentation . . . . .	268
4.84.2.1 hasRunningChildren() . . . . .	268
4.84.2.2 runAsyncProcess() . . . . .	268
4.84.2.3 waitForExit() . . . . .	269
4.85 rstudio::launcher_plugins::quickstart::QuickStartJobRepository Class Reference . . . . .	269
4.85.1 Constructor & Destructor Documentation . . . . .	269
4.85.1.1 QuickStartJobRepository() . . . . .	269
4.86 rstudio::launcher_plugins::quickstart::QuickStartJobSource Class Reference . . . . .	270
4.86.1 Detailed Description . . . . .	271
4.86.2 Constructor & Destructor Documentation . . . . .	271
4.86.2.1 QuickStartJobSource() . . . . .	271
4.86.3 Member Function Documentation . . . . .	271

4.86.3.1	<a href="#">cancelJob()</a>	271
4.86.3.2	<a href="#">createOutputStream()</a>	272
4.86.3.3	<a href="#">createResourceStream()</a>	272
4.86.3.4	<a href="#">getConfiguration()</a>	273
4.86.3.5	<a href="#">getNetworkInfo()</a>	273
4.86.3.6	<a href="#">initialize()</a>	274
4.86.3.7	<a href="#">killJob()</a>	274
4.86.3.8	<a href="#">resumeJob()</a>	275
4.86.3.9	<a href="#">stopJob()</a>	275
4.86.3.10	<a href="#">submitJob()</a>	276
4.86.3.11	<a href="#">suspendJob()</a>	276
4.87	<a href="#">rstudio::launcher_plugins::quickstart::QuickStartJobStatusWatcher Class Reference</a>	277
4.87.1	<a href="#">Constructor &amp; Destructor Documentation</a>	277
4.87.1.1	<a href="#">QuickStartJobStatusWatcher()</a>	277
4.88	<a href="#">rstudio::launcher_plugins::quickstart::QuickStartOptions Class Reference</a>	278
4.88.1	<a href="#">Detailed Description</a>	278
4.88.2	<a href="#">Member Function Documentation</a>	278
4.88.2.1	<a href="#">getInstance()</a>	278
4.88.2.2	<a href="#">getSampleOption()</a>	279
4.88.2.3	<a href="#">initialize()</a>	279
4.89	<a href="#">rstudio::launcher_plugins::quickstart::QuickStartPluginApi Class Reference</a>	279
4.89.1	<a href="#">Detailed Description</a>	280
4.89.2	<a href="#">Constructor &amp; Destructor Documentation</a>	280
4.89.2.1	<a href="#">QuickStartPluginApi()</a>	280
4.90	<a href="#">rstudio::launcher_plugins::quickstart::QuickStartResourceStream Class Reference</a>	280
4.90.1	<a href="#">Constructor &amp; Destructor Documentation</a>	281
4.90.1.1	<a href="#">QuickStartResourceStream()</a>	281
4.90.2	<a href="#">Member Function Documentation</a>	281
4.90.2.1	<a href="#">initialize()</a>	281
4.91	<a href="#">rstudio::launcher_plugins::system::RemoveOnExitScope Class Reference</a>	281
4.91.1	<a href="#">Detailed Description</a>	282
4.91.2	<a href="#">Constructor &amp; Destructor Documentation</a>	282
4.91.2.1	<a href="#">RemoveOnExitScope()</a>	282
4.92	<a href="#">rstudio::launcher_plugins::api::Request Class Reference</a>	282
4.92.1	<a href="#">Detailed Description</a>	283
4.92.2	<a href="#">Member Enumeration Documentation</a>	283
4.92.2.1	<a href="#">Type</a>	284
4.92.3	<a href="#">Constructor &amp; Destructor Documentation</a>	284
4.92.3.1	<a href="#">Request()</a>	284
4.92.4	<a href="#">Member Function Documentation</a>	284
4.92.4.1	<a href="#">fromJson()</a>	285
4.92.4.2	<a href="#">getId()</a>	285



4.92.4.3 <code>getType()</code> . . . . .	285
4.93 <code>rstudio::launcher_plugins::api::ResourceLimit</code> Struct Reference . . . . .	285
4.93.1 Detailed Description . . . . .	286
4.93.2 Constructor & Destructor Documentation . . . . .	286
4.93.2.1 <code>ResourceLimit()</code> . . . . .	286
4.93.3 Member Function Documentation . . . . .	287
4.93.3.1 <code>fromJson()</code> . . . . .	287
4.93.3.2 <code>toJson()</code> . . . . .	287
4.93.4 Member Data Documentation . . . . .	287
4.93.4.1 <code>DefaultValue</code> . . . . .	287
4.93.4.2 <code>MaxValue</code> . . . . .	287
4.93.4.3 <code>ResourceType</code> . . . . .	288
4.93.4.4 <code>Value</code> . . . . .	288
4.94 <code>rstudio::launcher_plugins::api::ResourceUtilData</code> Struct Reference . . . . .	288
4.94.1 Detailed Description . . . . .	288
4.94.2 Member Data Documentation . . . . .	288
4.94.2.1 <code>CpuPercent</code> . . . . .	288
4.94.2.2 <code>CpuSeconds</code> . . . . .	289
4.94.2.3 <code>ResidentMem</code> . . . . .	289
4.94.2.4 <code>VirtualMem</code> . . . . .	289
4.95 <code>rstudio::launcher_plugins::api::ResourceUtilStreamRequest</code> Class Reference . . . . .	289
4.95.1 Detailed Description . . . . .	290
4.95.2 Member Function Documentation . . . . .	290
4.95.2.1 <code>isCancelRequest()</code> . . . . .	290
4.96 <code>rstudio::launcher_plugins::api::ResourceUtilStreamResponse</code> Class Reference . . . . .	290
4.96.1 Detailed Description . . . . .	291
4.96.2 Constructor & Destructor Documentation . . . . .	291
4.96.2.1 <code>ResourceUtilStreamResponse()</code> . . . . .	291
4.96.3 Member Function Documentation . . . . .	291
4.96.3.1 <code>toJson()</code> . . . . .	291
4.97 <code>rstudio::launcher_plugins::api::Response</code> Class Reference . . . . .	292
4.97.1 Detailed Description . . . . .	293
4.97.2 Member Enumeration Documentation . . . . .	293
4.97.2.1 <code>Type</code> . . . . .	293
4.97.3 Constructor & Destructor Documentation . . . . .	293
4.97.3.1 <code>Response()</code> . . . . .	294
4.97.4 Member Function Documentation . . . . .	294
4.97.4.1 <code>toJson()</code> . . . . .	294
4.98 <code>rstudio::launcher_plugins::system::RestoreCurrentPathScope</code> Class Reference . . . . .	294
4.98.1 Detailed Description . . . . .	295
4.98.2 Constructor & Destructor Documentation . . . . .	295
4.98.2.1 <code>RestoreCurrentPathScope()</code> . . . . .	295

4.99 <a href="#">rstudio::launcher_plugins::api::StreamSequenceId Class Reference</a>	295
4.99.1 Detailed Description	296
4.99.2 Constructor & Destructor Documentation	296
4.99.2.1 <a href="#">StreamSequenceId()</a> [1/3]	296
4.99.2.2 <a href="#">StreamSequenceId()</a> [2/3]	296
4.99.2.3 <a href="#">StreamSequenceId()</a> [3/3]	297
4.99.3 Member Function Documentation	297
4.99.3.1 <a href="#">operator=()</a> [1/2]	297
4.99.3.2 <a href="#">operator=()</a> [2/2]	297
4.99.3.3 <a href="#">toJson()</a>	298
4.100 <a href="#">rstudio::launcher_plugins::api::SubmitJobRequest Class Reference</a>	298
4.100.1 Detailed Description	298
4.100.2 Member Function Documentation	299
4.100.2.1 <a href="#">getJob()</a>	299
4.101 <a href="#">rstudio::launcher_plugins::Success Class Reference</a>	299
4.101.1 Detailed Description	299
4.102 <a href="#">rstudio::launcher_plugins::system::process::SyncChildProcess Class Reference</a>	300
4.102.1 Detailed Description	300
4.102.2 Constructor & Destructor Documentation	300
4.102.2.1 <a href="#">SyncChildProcess()</a>	300
4.102.3 Member Function Documentation	301
4.102.3.1 <a href="#">run()</a>	301
4.102.3.2 <a href="#">writeToStdin()</a>	301
4.103 <a href="#">rstudio::launcher_plugins::system::TimeDuration Class Reference</a>	301
4.103.1 Detailed Description	303
4.103.2 Constructor & Destructor Documentation	303
4.103.2.1 <a href="#">TimeDuration()</a> [1/3]	303
4.103.2.2 <a href="#">TimeDuration()</a> [2/3]	303
4.103.2.3 <a href="#">TimeDuration()</a> [3/3]	303
4.103.3 Member Function Documentation	304
4.103.3.1 <a href="#">getHours()</a>	304
4.103.3.2 <a href="#">getMicroseconds()</a>	304
4.103.3.3 <a href="#">getMinutes()</a>	304
4.103.3.4 <a href="#">getSeconds()</a>	305
4.103.3.5 <a href="#">Hours()</a>	305
4.103.3.6 <a href="#">Infinity()</a>	305
4.103.3.7 <a href="#">isInfinity()</a>	305
4.103.3.8 <a href="#">Microseconds()</a>	306
4.103.3.9 <a href="#">Minutes()</a>	306
4.103.3.10 <a href="#">operator"!=(())</a>	306
4.103.3.11 <a href="#">operator&lt;()</a>	307
4.103.3.12 <a href="#">operator&lt;=()</a>	307

4.103.3.13 operator=() [1/2]	307
4.103.3.14 operator=() [2/2]	308
4.103.3.15 operator==(())	308
4.103.3.16 operator>()	308
4.103.3.17 operator>=()	309
4.103.3.18 Seconds()	309
4.104 rstudio::launcher_plugins::api::ResourceLimit::Type Struct Reference	310
4.104.1 Member Data Documentation	310
4.104.1.1 CPU_COUNT	310
4.104.1.2 CPU_TIME	310
4.104.1.3 MEMORY	310
4.104.1.4 MEMORY_SWAP	310
4.105 rstudio::launcher_plugins::system::User Class Reference	310
4.105.1 Detailed Description	311
4.105.2 Constructor & Destructor Documentation	312
4.105.2.1 User() [1/3]	312
4.105.2.2 User() [2/3]	312
4.105.2.3 User() [3/3]	312
4.105.3 Member Function Documentation	312
4.105.3.1 exists()	313
4.105.3.2 getCurrentUser()	313
4.105.3.3 getGroupId()	313
4.105.3.4 getHomePath()	314
4.105.3.5 getShell()	314
4.105.3.6 getUserFromIdentifier() [1/2]	314
4.105.3.7 getUserFromIdentifier() [2/2]	314
4.105.3.8 getUserHomePath()	315
4.105.3.9 getUserId()	315
4.105.3.10 getUsername()	315
4.105.3.11 isAllUsers()	316
4.105.3.12 isEmpty()	316
4.105.3.13 operator!=(())	316
4.105.3.14 operator=() [1/2]	316
4.105.3.15 operator=() [2/2]	317
4.105.3.16 operator==(())	317
4.106 rstudio::launcher_plugins::api::UserRequest Class Reference	318
4.106.1 Detailed Description	318
4.106.2 Constructor & Destructor Documentation	318
4.106.2.1 UserRequest()	318
4.106.3 Member Function Documentation	319
4.106.3.1 getRequestUsername()	319
4.106.3.2 getUser()	319

4.107 <a href="#">rstudio::launcher_plugins::options::Value&lt; T &gt; Class Template Reference</a>	319
4.107.1 Detailed Description	320
4.107.2 Constructor & Destructor Documentation	321
4.107.2.1 Value()	321
4.107.3 Member Function Documentation	321
4.107.3.1 setDefaultValue()	321
4.108 <a href="#">rstudio::launcher_plugins::json::Value Class Reference</a>	322
4.108.1 Detailed Description	325
4.108.2 Constructor & Destructor Documentation	325
4.108.2.1 Value() [1/12]	325
4.108.2.2 Value() [2/12]	326
4.108.2.3 Value() [3/12]	326
4.108.2.4 Value() [4/12]	326
4.108.2.5 Value() [5/12]	326
4.108.2.6 Value() [6/12]	327
4.108.2.7 Value() [7/12]	327
4.108.2.8 Value() [8/12]	327
4.108.2.9 Value() [9/12]	328
4.108.2.10 Value() [10/12]	328
4.108.2.11 Value() [11/12]	328
4.108.2.12 Value() [12/12]	328
4.108.3 Member Function Documentation	329
4.108.3.1 clone()	329
4.108.3.2 coerce()	329
4.108.3.3 getArray()	329
4.108.3.4 getBool()	330
4.108.3.5 getDouble()	330
4.108.3.6 getFloat()	330
4.108.3.7 getInt()	330
4.108.3.8 getInt64()	331
4.108.3.9 getObject()	331
4.108.3.10 getString()	331
4.108.3.11 getType()	331
4.108.3.12 getUInt()	332
4.108.3.13 getUInt64()	332
4.108.3.14 getValue()	332
4.108.3.15 isArray()	332
4.108.3.16 isBool()	333
4.108.3.17 isDouble()	333
4.108.3.18 isFloat()	333
4.108.3.19 isInt()	333
4.108.3.20 isInt64()	334

4.108.3.21 isNull()	334
4.108.3.22 isObject()	334
4.108.3.23 isString()	334
4.108.3.24 isUInt()	335
4.108.3.25 isUInt64()	335
4.108.3.26 operator!=(())	335
4.108.3.27 operator=() [1/11]	335
4.108.3.28 operator=() [2/11]	336
4.108.3.29 operator=() [3/11]	336
4.108.3.30 operator=() [4/11]	336
4.108.3.31 operator=() [5/11]	338
4.108.3.32 operator=() [6/11]	338
4.108.3.33 operator=() [7/11]	338
4.108.3.34 operator=() [8/11]	340
4.108.3.35 operator=() [9/11]	340
4.108.3.36 operator=() [10/11]	340
4.108.3.37 operator=() [11/11]	342
4.108.3.38 operator==(())	342
4.108.3.39 parse() [1/2]	342
4.108.3.40 parse() [2/2]	344
4.108.3.41 parseAndValidate()	344
4.108.3.42 setValueAtPath() [1/12]	345
4.108.3.43 setValueAtPath() [2/12]	345
4.108.3.44 setValueAtPath() [3/12]	345
4.108.3.45 setValueAtPath() [4/12]	346
4.108.3.46 setValueAtPath() [5/12]	346
4.108.3.47 setValueAtPath() [6/12]	346
4.108.3.48 setValueAtPath() [7/12]	347
4.108.3.49 setValueAtPath() [8/12]	347
4.108.3.50 setValueAtPath() [9/12]	348
4.108.3.51 setValueAtPath() [10/12]	348
4.108.3.52 setValueAtPath() [11/12]	348
4.108.3.53 setValueAtPath() [12/12]	349
4.108.3.54 validate()	349
4.108.3.55 write() [1/2]	349
4.108.3.56 write() [2/2]	350
4.108.3.57 writeFormatted() [1/2]	350
4.108.3.58 writeFormatted() [2/2]	350

## 5 File Documentation 351

5.1 /workspaces/rstudio-launcher-plugin-sdk/sdk/include/json/Json.hpp File Reference	351
5.1.1 Detailed Description	354

5.1.2 Enumeration Type Documentation	354
5.1.2.1 Type	354
5.1.3 Function Documentation	354
5.1.3.1 asJsonType() [1/3]	354
5.1.3.2 asJsonType() [2/3]	355
5.1.3.3 asJsonType() [3/3]	355
5.1.3.4 isMissingMemberError()	356
5.1.3.5 isType()	356
5.1.3.6 jsonReadError()	357
5.1.3.7 readObject() [1/12]	357
5.1.3.8 readObject() [2/12]	358
5.1.3.9 readObject() [3/12]	358
5.1.3.10 readObject() [4/12]	359
5.1.3.11 readObject() [5/12]	359
5.1.3.12 readObject() [6/12]	360
5.1.3.13 readObject() [7/12]	360
5.1.3.14 readObject() [8/12]	361
5.1.3.15 readObject() [9/12]	362
5.1.3.16 readObject() [10/12]	362
5.1.3.17 readObject() [11/12]	363
5.1.3.18 readObject() [12/12]	363
5.1.3.19 toJsonArray() [1/2]	364
5.1.3.20 toJsonArray() [2/2]	364
5.1.3.21 toJsonValue() [1/5]	365
5.1.3.22 toJsonValue() [2/5]	365
5.1.3.23 toJsonValue() [3/5]	366
5.1.3.24 toJsonValue() [4/5]	366
5.1.3.25 toJsonValue() [5/5]	367
5.1.3.26 typeAsString()	367
5.2 /workspaces/rstudio-launcher-plugin-sdk/sdk/include/logging/Logger.hpp File Reference	367
5.2.1 Detailed Description	369
5.2.2 Function Documentation	369
5.2.2.1 addLogDestination() [1/2]	369
5.2.2.2 addLogDestination() [2/2]	370
5.2.2.3 cleanDelimiters()	370
5.2.2.4 cleanDelims()	370
5.2.2.5 logDebugMessage() [1/3]	371
5.2.2.6 logDebugMessage() [2/3]	371
5.2.2.7 logDebugMessage() [3/3]	372
5.2.2.8 logError() [1/2]	372
5.2.2.9 logError() [2/2]	372
5.2.2.10 logErrorAsDebug()	373

5.2.2.11 logErrorAsInfo()	373
5.2.2.12 logErrorAsWarning()	373
5.2.2.13 logErrorMessage() [1/3]	374
5.2.2.14 logErrorMessage() [2/3]	374
5.2.2.15 logErrorMessage() [3/3]	374
5.2.2.16 logInfoMessage() [1/3]	375
5.2.2.17 logInfoMessage() [2/3]	375
5.2.2.18 logInfoMessage() [3/3]	375
5.2.2.19 logWarningMessage() [1/3]	376
5.2.2.20 logWarningMessage() [2/3]	376
5.2.2.21 logWarningMessage() [3/3]	377
5.2.2.22 removeLogDestination()	377
5.2.2.23 setProgramId()	377
5.2.2.24 writeError() [1/2]	378
5.2.2.25 writeError() [2/2]	378
5.3 /workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/Crypto.hpp File Reference	378
5.3.1 Detailed Description	379
5.3.2 Function Documentation	379
5.3.2.1 aesDecrypt()	379
5.3.2.2 aesEncrypt()	380
5.3.2.3 base64Decode() [1/2]	380
5.3.2.4 base64Decode() [2/2]	381
5.3.2.5 base64Encode() [1/2]	381
5.3.2.6 base64Encode() [2/2]	382
5.3.2.7 decryptAndBase64Decode()	382
5.3.2.8 encryptAndBase64Encode()	383
5.3.2.9 random()	383
5.4 /workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/PosixSystem.hpp File Reference	384
5.4.1 Detailed Description	384
5.4.2 Function Documentation	385
5.4.2.1 enableCoreDumps()	385
5.4.2.2 getEnvironmentVariable()	385
5.4.2.3 getIpAddresses()	385
5.4.2.4 ignoreSignal()	386
5.4.2.5 posixCall() [1/2]	386
5.4.2.6 posixCall() [2/2]	387
5.4.2.7 realUserIsRoot()	387
5.4.2.8 restorePrivileges()	387
5.4.2.9 restoreRoot()	388
5.4.2.10 temporarilyDropPrivileges()	388
5.5 /workspaces/rstudio-launcher-plugin-sdk/sdk/include/utils/Functionals.hpp File Reference	388
5.5.1 Detailed Description	388

5.5.2 Typedef Documentation . . . . .	388
5.5.2.1 OnError . . . . .	389
<b>Index</b>	<b>391</b>



# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

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

rstudio::launcher_plugins::api::AbstractOutputStream . . . . .	30
rstudio::launcher_plugins::api::FileOutputStream . . . . .	113
rstudio::launcher_plugins::system::AsioStream . . . . .	61
rstudio::launcher_plugins::system::AsyncDeadlineEvent . . . . .	63
rstudio::launcher_plugins::system::process::AsyncProcessCallbacks . . . . .	64
rstudio::launcher_plugins::system::AsyncTimedEvent . . . . .	65
rstudio::launcher_plugins::api::Container . . . . .	76
rstudio::launcher_plugins::api::ContainerConfiguration . . . . .	78
rstudio::launcher_plugins::system::DateTime . . . . .	82
enable_shared_from_this	
rstudio::launcher_plugins::api::AbstractPluginApi . . . . .	34
rstudio::launcher_plugins::local::LocalPluginApi . . . . .	205
rstudio::launcher_plugins::quickstart::QuickStartPluginApi . . . . .	279
rstudio::launcher_plugins::api::AbstractTimedResourceStream . . . . .	40
rstudio::launcher_plugins::local::LocalResourceStream . . . . .	206
rstudio::launcher_plugins::api::FileOutputStream . . . . .	113
rstudio::launcher_plugins::comms::AbstractLauncherCommunicator . . . . .	20
rstudio::launcher_plugins::jobs::AbstractJobRepository . . . . .	15
rstudio::launcher_plugins::local::LocalJobRepository . . . . .	193
rstudio::launcher_plugins::quickstart::QuickStartJobRepository . . . . .	269
rstudio::launcher_plugins::jobs::AbstractTimedJobStatusWatcher . . . . .	39
rstudio::launcher_plugins::quickstart::QuickStartJobStatusWatcher . . . . .	277
rstudio::launcher_plugins::jobs::JobStatusNotifier . . . . .	187
rstudio::launcher_plugins::local::LocalJobRunner . . . . .	195
rstudio::launcher_plugins::ErrorLocation . . . . .	100
rstudio::launcher_plugins::ErrorLock . . . . .	104
rstudio::launcher_plugins::Error . . . . .	90
rstudio::launcher_plugins::Success . . . . .	299
rstudio::launcher_plugins::api::ExposedPort . . . . .	106
rstudio::launcher_plugins::logging::FileLogOptions . . . . .	110
rstudio::launcher_plugins::system::FilePath . . . . .	115
rstudio::launcher_plugins::api::IJobSource . . . . .	145
rstudio::launcher_plugins::local::LocalJobSource . . . . .	197
rstudio::launcher_plugins::quickstart::QuickStartJobSource . . . . .	270

rstudio::launcher_plugins::options::Options::Init . . . . .	154
rstudio::launcher_plugins::system::posix::IpAddress . . . . .	157
is_base_of	
rstudio::launcher_plugins::json::detail::is_json_type< T > . . . . .	157
iterator	
rstudio::launcher_plugins::json::Array::Iterator . . . . .	162
rstudio::launcher_plugins::json::Object::Iterator . . . . .	158
rstudio::launcher_plugins::api::Job . . . . .	165
rstudio::launcher_plugins::api::JobConfig . . . . .	175
rstudio::launcher_plugins::api::JobSourceConfiguration . . . . .	182
rstudio::launcher_plugins::local::LocalSecureCookie . . . . .	207
rstudio::launcher_plugins::json::Object::Member . . . . .	208
rstudio::launcher_plugins::api::Mount . . . . .	210
rstudio::launcher_plugins::api::MountSource . . . . .	212
rstudio::launcher_plugins::api::AzureFileMountSource . . . . .	66
rstudio::launcher_plugins::api::CephFsMountSource . . . . .	71
rstudio::launcher_plugins::api::GlusterFsMountSource . . . . .	141
rstudio::launcher_plugins::api::HostMountSource . . . . .	143
rstudio::launcher_plugins::api::NfsMountSource . . . . .	224
rstudio::launcher_plugins::api::NetworkInfo . . . . .	221
noncopyable	
rstudio::launcher_plugins::options::Options . . . . .	249
rstudio::launcher_plugins::Noncopyable . . . . .	226
rstudio::launcher_plugins::api::AbstractMultiStream< ResourceUtilStreamResponse, ResourceUtilStreamResponse, Data, bool > . . . . .	25
rstudio::launcher_plugins::api::AbstractResourceStream . . . . .	36
rstudio::launcher_plugins::api::AbstractTimedResourceStream . . . . .	40
rstudio::launcher_plugins::quickstart::QuickStartResourceStream . . . . .	280
rstudio::launcher_plugins::AbstractMain . . . . .	24
rstudio::launcher_plugins::api::AbstractMultiStream< R, Args > . . . . .	25
rstudio::launcher_plugins::api::AbstractPluginApi . . . . .	34
rstudio::launcher_plugins::api::JobLock . . . . .	180
rstudio::launcher_plugins::api::Request . . . . .	282
rstudio::launcher_plugins::api::BootstrapRequest . . . . .	68
rstudio::launcher_plugins::api::UserRequest . . . . .	318
rstudio::launcher_plugins::api::JobIdRequest . . . . .	178
rstudio::launcher_plugins::api::ControlJobRequest . . . . .	79
rstudio::launcher_plugins::api::JobStateRequest . . . . .	183
rstudio::launcher_plugins::api::JobStatusRequest . . . . .	190
rstudio::launcher_plugins::api::NetworkRequest . . . . .	222
rstudio::launcher_plugins::api::OutputStreamRequest . . . . .	255
rstudio::launcher_plugins::api::ResourceUtilStreamRequest . . . . .	289
rstudio::launcher_plugins::api::SubmitJobRequest . . . . .	298
rstudio::launcher_plugins::api::Response . . . . .	292
rstudio::launcher_plugins::api::BootstrapResponse . . . . .	70
rstudio::launcher_plugins::api::ClusterInfoResponse . . . . .	74
rstudio::launcher_plugins::api::ControlJobResponse . . . . .	81
rstudio::launcher_plugins::api::ErrorResponse . . . . .	105
rstudio::launcher_plugins::api::HeartbeatResponse . . . . .	142
rstudio::launcher_plugins::api::JobStateResponse . . . . .	186
rstudio::launcher_plugins::api::MultiStreamResponse . . . . .	220
rstudio::launcher_plugins::api::JobStatusResponse . . . . .	192
rstudio::launcher_plugins::api::ResourceUtilStreamResponse . . . . .	290
rstudio::launcher_plugins::api::NetworkResponse . . . . .	223
rstudio::launcher_plugins::api::OutputStreamResponse . . . . .	257
rstudio::launcher_plugins::comms::AbstractLauncherCommunicator . . . . .	20
rstudio::launcher_plugins::jobs::AbstractJobRepository . . . . .	15

rstudio::launcher_plugins::jobs::AbstractJobStatusWatcher	18
rstudio::launcher_plugins::jobs::AbstractTimedJobStatusWatcher	39
rstudio::launcher_plugins::jobs::JobStatusNotifier	187
rstudio::launcher_plugins::local::LocalOptions	203
rstudio::launcher_plugins::logging::ILogDestination	152
rstudio::launcher_plugins::logging::FileLogDestination	108
rstudio::launcher_plugins::options::AbstractUserProfiles	42
rstudio::launcher_plugins::quickstart::QuickStartOptions	278
rstudio::launcher_plugins::system::AsioService	59
rstudio::launcher_plugins::system::process::AbstractChildProcess	13
rstudio::launcher_plugins::system::process::SyncChildProcess	300
rstudio::launcher_plugins::system::process::ProcessSupervisor	267
rstudio::launcher_plugins::system::RemoveOnExitScope	281
rstudio::launcher_plugins::system::RestoreCurrentPathScope	294
rstudio::launcher_plugins::Optional< T >	244
rstudio::launcher_plugins::Optional< double >	244
rstudio::launcher_plugins::Optional< int >	244
rstudio::launcher_plugins::Optional< pid_t >	244
rstudio::launcher_plugins::Optional< rstudio::launcher_plugins::api::Container >	244
rstudio::launcher_plugins::Optional< rstudio::launcher_plugins::system::DateTime >	244
rstudio::launcher_plugins::Optional< Type >	244
rstudio::launcher_plugins::system::PathScopeImplDeleter	259
rstudio::launcher_plugins::api::PlacementConstraint	259
rstudio::launcher_plugins::system::process::ProcessInfo	262
rstudio::launcher_plugins::system::process::ProcessOptions	264
rstudio::launcher_plugins::system::process::ProcessResult	266
rstudio::launcher_plugins::api::ResourceLimit	285
rstudio::launcher_plugins::api::ResourceUtilData	288
rstudio::launcher_plugins::api::StreamSequenceId	295
rstudio::launcher_plugins::system::TimeDuration	301
rstudio::launcher_plugins::api::ResourceLimit::Type	310
rstudio::launcher_plugins::system::User	310
rstudio::launcher_plugins::options::Value< T >	319
rstudio::launcher_plugins::json::Value	322
rstudio::launcher_plugins::json::Array	46
rstudio::launcher_plugins::json::Object	227



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">rstudio::launcher_plugins::system::process::AbstractChildProcess</a>	
Base class for a child process, which will be launched via rsandbox . . . . .	13
<a href="#">rstudio::launcher_plugins::jobs::AbstractJobRepository</a>	
Stores any jobs currently in the job scheduling system . . . . .	15
<a href="#">rstudio::launcher_plugins::jobs::AbstractJobStatusWatcher</a>	
Manages posting job status updates to the notifier . . . . .	18
<a href="#">rstudio::launcher_plugins::comms::AbstractLauncherCommunicator</a>	
Base class responsible for communicating the the launcher. The type of communication is im- plementation dependent . . . . .	20
<a href="#">rstudio::launcher_plugins::AbstractMain</a>	
Base class for the PluginMain class, which runs the plugin . . . . .	24
<a href="#">rstudio::launcher_plugins::api::AbstractMultiStream&lt; R, Args &gt;</a>	
Manages the sending of streamed responses . . . . .	25
<a href="#">rstudio::launcher_plugins::api::AbstractOutputStream</a>	
Streams job output data to the launcher . . . . .	30
<a href="#">rstudio::launcher_plugins::api::AbstractPluginApi</a>	
Base class for the Launcher Plugin API . . . . .	34
<a href="#">rstudio::launcher_plugins::api::AbstractResourceStream</a>	
Streams job resource utilization data to the Launcher . . . . .	36
<a href="#">rstudio::launcher_plugins::jobs::AbstractTimedJobStatusWatcher</a>	
Responsible for polling job statuses on a timer . . . . .	39
<a href="#">rstudio::launcher_plugins::api::AbstractTimedResourceStream</a>	
	40
<a href="#">rstudio::launcher_plugins::options::AbstractUserProfiles</a>	
Base class which reads an ini-based user profiles file . . . . .	42
<a href="#">rstudio::launcher_plugins::json::Array</a>	
Class which represents a JSON array . . . . .	46
<a href="#">rstudio::launcher_plugins::system::AsioService</a>	
Async input/output class which may be used to manage ASIO operations . . . . .	59
<a href="#">rstudio::launcher_plugins::system::AsioStream</a>	
Class which allows reading from or writing to streams asynchronously . . . . .	61
<a href="#">rstudio::launcher_plugins::system::AsyncDeadlineEvent</a>	
Class which may be used to post async work to be performed at a later time . . . . .	63
<a href="#">rstudio::launcher_plugins::system::process::AsyncProcessCallbacks</a>	
Callbacks that will be invoked when certain events happen in the asynchronous child process .	64
<a href="#">rstudio::launcher_plugins::system::AsyncTimedEvent</a>	
Class which performs an action asynchronously every specified number of seconds . . . . .	65

<a href="#">rstudio::launcher_plugins::api::AzureFileMountSource</a>	
Represents an Azure File <a href="#">Mount</a> Source . . . . .	66
<a href="#">rstudio::launcher_plugins::api::BootstrapRequest</a>	
Represents a bootstrap request received from the Launcher . . . . .	68
<a href="#">rstudio::launcher_plugins::api::BootstrapResponse</a>	
Class which represents a bootstrap response which can be sent to the Launcher in response to a bootstrap request . . . . .	70
<a href="#">rstudio::launcher_plugins::api::CephFsMountSource</a>	
Represents a Ceph File System <a href="#">Mount</a> Source . . . . .	71
<a href="#">rstudio::launcher_plugins::api::ClusterInfoResponse</a>	
Class which represents a cluster info response which should be sent to the Launcher in response to a cluster info request . . . . .	74
<a href="#">rstudio::launcher_plugins::api::Container</a>	
Struct which represents the container to use when launching a containerized job . . . . .	76
<a href="#">rstudio::launcher_plugins::api::ContainerConfiguration</a>	
Describes the container configuration of the <a href="#">Job</a> Source . . . . .	78
<a href="#">rstudio::launcher_plugins::api::ControlJobRequest</a>	
Request from the launcher to control the state of a <a href="#">Job</a> . . . . .	79
<a href="#">rstudio::launcher_plugins::api::ControlJobResponse</a>	
Class which represents the result of a control job operation . . . . .	81
<a href="#">rstudio::launcher_plugins::system::DateTime</a>	
Class which represents a date and time in UTC . . . . .	82
<a href="#">rstudio::launcher_plugins::Error</a>	
Class which represents an error . . . . .	90
<a href="#">rstudio::launcher_plugins::ErrorLocation</a>	
Class which represents the location of an error . . . . .	100
<a href="#">rstudio::launcher_plugins::ErrorLock</a>	
A class which can be derived from in order to prevent child classes from being derived from further . . . . .	104
<a href="#">rstudio::launcher_plugins::api::ErrorResponse</a>	
Class which represents an error response which can be sent to the Launcher in response to any request . . . . .	105
<a href="#">rstudio::launcher_plugins::api::ExposedPort</a>	
Struct which represents an exposed port on a containerized job . . . . .	106
<a href="#">rstudio::launcher_plugins::logging::FileLogDestination</a>	
Class which allows sending log messages to a file . . . . .	108
<a href="#">rstudio::launcher_plugins::logging::FileLogOptions</a>	
Class which represents the options for a file logger . . . . .	110
<a href="#">rstudio::launcher_plugins::api::FileOutputStream</a>	
Streams job output data from a file . . . . .	113
<a href="#">rstudio::launcher_plugins::system::FilePath</a>	
Class which represents a path on the system. May be any type of file (e.g. directory, symlink, regular file, etc.) . . . . .	115
<a href="#">rstudio::launcher_plugins::api::GlusterFsMountSource</a>	
<a href="#">rstudio::launcher_plugins::api::HeartbeatResponse</a>	
Class which represents a heartbeat response which should be sent to the Launcher every configured heartbeat-interval-seconds . . . . .	142
<a href="#">rstudio::launcher_plugins::api::HostMountSource</a>	
Represents a path to mount on the same host as the <a href="#">Job</a> . . . . .	143
<a href="#">rstudio::launcher_plugins::api::IJobSource</a>	
Generic interface for communicating with a <a href="#">Job</a> Source. Implementation is plugin specific . . .	145
<a href="#">rstudio::launcher_plugins::logging::ILogDestination</a>	
Interface which allows a logger to write a log message to a destination . . . . .	152
<a href="#">rstudio::launcher_plugins::options::Options::Init</a>	
Class for initializing <a href="#">Options</a> . . . . .	154
<a href="#">rstudio::launcher_plugins::system::posix::IpAddress</a>	
Represents an IP address . . . . .	157

<a href="#">rstudio::launcher_plugins::json::detail::is_json_type&lt; T &gt;</a>	
Struct which is either a child class of <code>std::true_type</code> or <code>std::false_type</code> depending on whether T is a JSON type (e.g. <a href="#">Value</a> , <a href="#">Object</a> , <a href="#">Array</a> ) or not (e.g. int, bool, string, float, etc.)	157
<a href="#">rstudio::launcher_plugins::json::Object::Iterator</a>	
Class which allows iterating over the members of a JSON object	158
<a href="#">rstudio::launcher_plugins::json::Array::Iterator</a>	
Class which allows iterating over the elements of a JSON array	162
<a href="#">rstudio::launcher_plugins::api::Job</a>	
Structure which represents a job	165
<a href="#">rstudio::launcher_plugins::api::JobConfig</a>	
Struct which represents a custom configuration setting for jobs launched with a given Plugin	175
<a href="#">rstudio::launcher_plugins::api::JobIdRequest</a>	
Base class which should be used for requests that require a <a href="#">Job</a> ID	178
<a href="#">rstudio::launcher_plugins::api::JobLock</a>	
RAII class for locking access to a <a href="#">Job</a> object. Should be used every time a <a href="#">Job</a> is modified	180
<a href="#">rstudio::launcher_plugins::api::JobSourceConfiguration</a>	
Describes the capabilities and configuration of this <a href="#">Job</a> Source	182
<a href="#">rstudio::launcher_plugins::api::JobStateRequest</a>	
Represents a job state request received from the Launcher	183
<a href="#">rstudio::launcher_plugins::api::JobStateResponse</a>	
Class which represents a job state response which can be sent to the Launcher in response to a get or submit job request	186
<a href="#">rstudio::launcher_plugins::jobs::JobStatusNotifier</a>	
Class which notifies subscribers when a job updates	187
<a href="#">rstudio::launcher_plugins::api::JobStatusRequest</a>	
Request from the launcher to begin or end a <a href="#">Job</a> Status Stream	190
<a href="#">rstudio::launcher_plugins::api::JobStatusResponse</a>	
Class which represents a <a href="#">Job</a> Status Stream, either for all jobs or for a specific job	192
<a href="#">rstudio::launcher_plugins::local::LocalJobRepository</a>	
Responsible for job persistence	193
<a href="#">rstudio::launcher_plugins::local::LocalJobRunner</a>	
Runs jobs on the local machine	195
<a href="#">rstudio::launcher_plugins::local::LocalJobSource</a>	
Class which is responsible for running and retrieving information about jobs on the Local system	197
<a href="#">rstudio::launcher_plugins::local::LocalOptions</a>	
Class which stores options specific to the Local Container system	203
<a href="#">rstudio::launcher_plugins::local::LocalPluginApi</a>	
Launcher Plugin API for the Local Plugin	205
<a href="#">rstudio::launcher_plugins::local::LocalResourceStream</a>	
	206
<a href="#">rstudio::launcher_plugins::local::LocalSecureCookie</a>	
Reads and makes available the secure-cookie-key-file specified in the launcher.local.conf file	207
<a href="#">rstudio::launcher_plugins::json::Object::Member</a>	
Class which represents a single member of a JSON object	208
<a href="#">rstudio::launcher_plugins::api::Mount</a>	
Struct which represents an file system mount available to a job	210
<a href="#">rstudio::launcher_plugins::api::MountSource</a>	
Struct which represents the source path of an NFS <a href="#">Mount</a>	212
<a href="#">rstudio::launcher_plugins::api::MultiStreamResponse</a>	
Base class for responses which are returned to multiple stream requests	220
<a href="#">rstudio::launcher_plugins::api::NetworkInfo</a>	
Represents the network information for a job	221
<a href="#">rstudio::launcher_plugins::api::NetworkRequest</a>	
Request from the Launcher to get the network information for a job	222
<a href="#">rstudio::launcher_plugins::api::NetworkResponse</a>	
Class which represents a network information response which should be sent to the Launcher in response to a <a href="#">Job</a> network information request	223
<a href="#">rstudio::launcher_plugins::api::NfsMountSource</a>	
Represents an NFS <a href="#">Mount</a> Source	224

<a href="#">rstudio::launcher_plugins::Noncopyable</a>	
Class which can be inherited from to disallow copying of its child classes	226
<a href="#">rstudio::launcher_plugins::json::Object</a>	
Class which represents a specific type of JSON <a href="#">Value</a> : a JSON object	227
<a href="#">rstudio::launcher_plugins::Optional&lt; T &gt;</a>	
Container class which represents a value that may or may not be set	244
<a href="#">rstudio::launcher_plugins::options::Options</a>	
<a href="#">Options</a> for the plugin	249
<a href="#">rstudio::launcher_plugins::api::OutputStreamRequest</a>	
<a href="#">Request</a> from the launcher to begin or end a <a href="#">Job</a> Output Stream	255
<a href="#">rstudio::launcher_plugins::api::OutputStreamResponse</a>	
Class which represents a <a href="#">Job</a> Output Stream response for a specific job	257
<a href="#">rstudio::launcher_plugins::system::PathScopeImplDeleter</a>	
Struct which implements the deleter for PathScopeImpl	259
<a href="#">rstudio::launcher_plugins::api::PlacementConstraint</a>	
Struct which represents a custom placement constraint for the job	259
<a href="#">rstudio::launcher_plugins::system::process::ProcessInfo</a>	
Represents the details of a process that is running on this machine	262
<a href="#">rstudio::launcher_plugins::system::process::ProcessOptions</a>	
Defines a process that can be run	264
<a href="#">rstudio::launcher_plugins::system::process::ProcessResult</a>	
Represents the result of a synchronous child process	266
<a href="#">rstudio::launcher_plugins::system::process::ProcessSupervisor</a>	
Creates and manages non-blocking child processes	267
<a href="#">rstudio::launcher_plugins::quickstart::QuickStartJobRepository</a>	269
<a href="#">rstudio::launcher_plugins::quickstart::QuickStartJobSource</a>	
Class which is responsible for running and retrieving information about jobs in the job scheduling system	270
<a href="#">rstudio::launcher_plugins::quickstart::QuickStartJobStatusWatcher</a>	277
<a href="#">rstudio::launcher_plugins::quickstart::QuickStartOptions</a>	
Class which defines options for the QuickStart Launcher Plugin	278
<a href="#">rstudio::launcher_plugins::quickstart::QuickStartPluginApi</a>	
Launcher Plugin API for the QuickStart Plugin	279
<a href="#">rstudio::launcher_plugins::quickstart::QuickStartResourceStream</a>	280
<a href="#">rstudio::launcher_plugins::system::RemoveOnExitScope</a>	
RAII class for restoring the current working directory	281
<a href="#">rstudio::launcher_plugins::api::Request</a>	
Base class for all requests which may be received from the Launcher	282
<a href="#">rstudio::launcher_plugins::api::ResourceLimit</a>	
Struct which represents a resource limit for a job	285
<a href="#">rstudio::launcher_plugins::api::ResourceUtilData</a>	
Represents the current resource utilization of a job	288
<a href="#">rstudio::launcher_plugins::api::ResourceUtilStreamRequest</a>	
<a href="#">Request</a> from the launcher to begin or end a Resource Utilization Stream	289
<a href="#">rstudio::launcher_plugins::api::ResourceUtilStreamResponse</a>	
Class which represents a Resource Utilization Stream response for a specific job	290
<a href="#">rstudio::launcher_plugins::api::Response</a>	
Represents the common components of all responses which can be sent the RStudio Launcher	292
<a href="#">rstudio::launcher_plugins::system::RestoreCurrentPathScope</a>	
RAII class for restoring the current working directory	294
<a href="#">rstudio::launcher_plugins::api::StreamSequenceId</a>	
An identifier for a <a href="#">MultiStreamResponse</a>	295
<a href="#">rstudio::launcher_plugins::api::SubmitJobRequest</a>	
Represents a submit job request from the Launcher	298
<a href="#">rstudio::launcher_plugins::Success</a>	
Class which represents a successful operation (i.e. no error)	299
<a href="#">rstudio::launcher_plugins::system::process::SyncChildProcess</a>	
A blocking child process	300



<a href="#">rstudio::launcher_plugins::system::TimeDuration</a>	
Represents an duration of time (e.g. 5 hours, 43 minutes, and 21 seconds) as opposed to a point in time . . . . .	301
<a href="#">rstudio::launcher_plugins::api::ResourceLimit::Type</a> . . . . .	310
<a href="#">rstudio::launcher_plugins::system::User</a>	
Class which represents a system user . . . . .	310
<a href="#">rstudio::launcher_plugins::api::UserRequest</a>	
Base class which should be used by the class of requests which require a username . . . . .	318
<a href="#">rstudio::launcher_plugins::options::Value&lt; T &gt;</a>	
Concrete class which represents an option <a href="#">Value</a> . . . . .	319
<a href="#">rstudio::launcher_plugins::json::Value</a>	
Class which represents a json value . . . . .	322



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all documented files with brief descriptions:

/workspaces/rstudio-launcher-plugin-sdk/plugins/Local/include/ <b>LocalConstants.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/plugins/Local/include/ <b>LocalError.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/plugins/Local/include/ <b>LocalJobRepository.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/plugins/Local/include/ <b>LocalJobRunner.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/plugins/Local/include/ <b>LocalJobSource.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/plugins/Local/include/ <b>LocalOptions.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/plugins/Local/include/ <b>LocalPluginApi.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/plugins/Local/include/ <b>LocalResourceStream.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/plugins/Local/include/ <b>LocalSecureCookie.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/plugins/QuickStart/include/ <b>QuickStartJobRepository.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/plugins/QuickStart/include/ <b>QuickStartJobSource.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/plugins/QuickStart/include/ <b>QuickStartJobStatusWatcher.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/plugins/QuickStart/include/ <b>QuickStartOptions.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/plugins/QuickStart/include/ <b>QuickStartPluginApi.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/plugins/QuickStart/include/ <b>QuickStartResourceStream.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/ <b>AbstractMain.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/ <b>Error.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/ <b>Noncopyable.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/ <b>Optional.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/ <b>Pimpl.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/ <b>SafeConvert.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/ <b>AbstractPluginApi.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/ <b>IJobSource.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/ <b>Job.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/ <b>Request.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/ <b>Response.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/ <b>ResponseTypes.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/stream/ <b>AbstractMultiStream.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/stream/ <b>AbstractOutputStream.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/stream/ <b>AbstractResourceStream.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/stream/ <b>AbstractTimedResourceStream.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/stream/ <b>FileOutputStream.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/comms/ <b>AbstractLauncherCommunicator.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/jobs/ <b>AbstractJobRepository.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/jobs/ <b>AbstractJobStatusWatcher.hpp</b>	??

/workspaces/rstudio-launcher-plugin-sdk/sdk/include/jobs/ <b>AbstractTimedJobStatusWatcher.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/jobs/ <b>JobStatusNotifier.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/json/ <b>Json.hpp</b>	351
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/logging/ <b>FileLogDestination.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/logging/ <b>ILogDestination.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/logging/ <b>Logger.hpp</b>	367
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/options/ <b>AbstractUserProfiles.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/options/ <b>Options.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/ <b>Asio.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/ <b>Crypto.hpp</b>	378
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/ <b>DateTime.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/ <b>FilePath.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/ <b>PosixSystem.hpp</b>	384
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/ <b>Process.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/ <b>User.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/utls/ <b>FileUtils.hpp</b>	??
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/utls/ <b>Functionals.hpp</b>	388
/workspaces/rstudio-launcher-plugin-sdk/sdk/include/utls/ <b>MutexUtils.hpp</b>	??

## Chapter 4

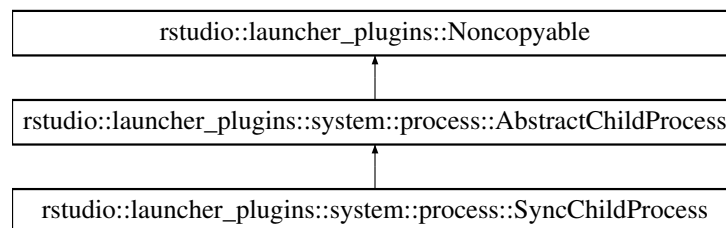
# Class Documentation

### 4.1 rstudio::launcher\_plugins::system::process::AbstractChildProcess Class Reference

Base class for a child process, which will be launched via rsandbox.

```
#include <Process.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::system::process::AbstractChildProcess:



#### Public Member Functions

- virtual `~AbstractChildProcess()` = default  
*Virtual destructor for inheritance.*
- `pid_t getpid()` const  
*Gets the PID of this child process.*
- virtual `Error terminate()`  
*Terminates the child process.*
- virtual `Error writeToStdin(const std::string &in_string, bool in_eof)=0`  
*Writes the specified string to stdin.*

#### Protected Member Functions

- `AbstractChildProcess(const ProcessOptions &in_options)`  
*Constructor.*
- `Error run()`  
*Forks and executes the child process, passing along arguments and environment variables.*
- `PRIVATE_IMPL(m_baseImpl)`

### 4.1.1 Detailed Description

Base class for a child process, which will be launched via rsandbox.

### 4.1.2 Constructor & Destructor Documentation

#### 4.1.2.1 AbstractChildProcess()

```
rstudio::launcher_plugins::system::process::AbstractChildProcess::AbstractChildProcess (
    const ProcessOptions & in_options ) [explicit], [protected]
```

Constructor.

Parameters

<i>in_options</i>	The options for the child process.
-------------------	------------------------------------

### 4.1.3 Member Function Documentation

#### 4.1.3.1 getPid()

```
pid_t rstudio::launcher_plugins::system::process::AbstractChildProcess::getPid ( ) const
```

Gets the PID of this child process.

Returns

The PID of this child process.

#### 4.1.3.2 run()

```
Error rstudio::launcher_plugins::system::process::AbstractChildProcess::run ( ) [protected]
```

Forks and executes the child process, passing along arguments and environment variables.

Returns

**Success** if the child process could be started; **Error** otherwise.

#### 4.1.3.3 terminate()

```
virtual Error rstudio::launcher_plugins::system::process::AbstractChildProcess::terminate ( )  
[virtual]
```

Terminates the child process.

##### Returns

[Success](#) if the child process was terminated; [Error](#) otherwise.

#### 4.1.3.4 writeToStdin()

```
virtual Error rstudio::launcher_plugins::system::process::AbstractChildProcess::writeToStdin (   
    const std::string & in_string,  
    bool in_eof ) [pure virtual]
```

Writes the specified string to stdin.

##### Parameters

<i>in_string</i>	The data to write to stdin.
<i>in_eof</i>	True if this is the last data to write to stdin.

##### Returns

[Success](#) if the data could be written; [Error](#) otherwise.

Implemented in [rstudio::launcher\\_plugins::system::process::SyncChildProcess](#).

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

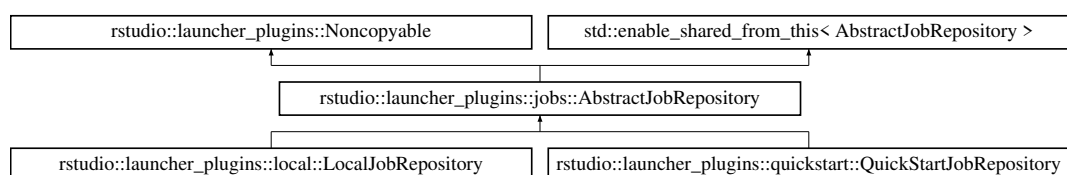
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/Process.hpp

## 4.2 rstudio::launcher\_plugins::jobs::AbstractJobRepository Class Reference

Stores any jobs currently in the job scheduling system.

```
#include <AbstractJobRepository.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::jobs::AbstractJobRepository:



## Public Member Functions

- [AbstractJobRepository](#) (JobStatusNotifierPtr in\_jobStatusNotifier)  
*Constructor.*
- virtual [~AbstractJobRepository](#) ()=default  
*Virtual Destructor, to allow for inheritance, if necessary.*
- void [addJob](#) (const api::JobPtr &in\_job)  
*Adds the job to the repository.*
- api::JobPtr [getJob](#) (const std::string &in\_jobId, const [system::User](#) &in\_user=[system::User\(\)](#)) const  
*Gets the specified job for the specified user from the repository.*
- api::JobList [getJobs](#) (const [system::User](#) &in\_use=[system::User\(\)](#)) const  
*Gets all jobs belonging to the specified user.*
- [Error initialize](#) ()  
*Initializes the [AbstractJobRepository](#).*
- void [removeJob](#) (const std::string &in\_jobId)  
*Removes a job from the repository.*

### 4.2.1 Detailed Description

Stores any jobs currently in the job scheduling system.

### 4.2.2 Constructor & Destructor Documentation

#### 4.2.2.1 AbstractJobRepository()

```
rstudio::launcher_plugins::jobs::AbstractJobRepository::AbstractJobRepository (
    JobStatusNotifierPtr in_jobStatusNotifier ) [explicit]
```

Constructor.

Parameters

<i>in_jobStatusNotifier</i>	The job status notifier. Used to add new jobs.
-----------------------------	--

### 4.2.3 Member Function Documentation

#### 4.2.3.1 addJob()

```
void rstudio::launcher_plugins::jobs::AbstractJobRepository::addJob (
    const api::JobPtr & in_job )
```

Adds the job to the repository.

If the job is already in the repository, nothing will happen.



**Parameters**

<i>in_job</i>	The job to add to the repository.
---------------	-----------------------------------

**4.2.3.2 getJob()**

```
api::JobPtr rstudio::launcher_plugins::jobs::AbstractJobRepository::getJob (
    const std::string & in_jobId,
    const system::User & in_user = system::User() ) const
```

Gets the specified job for the specified user from the repository.

If the job does not belong to the specified user and if the user does not represent "all users", no job will be returned.

**Parameters**

<i>in_jobId</i>	The ID of the job to retrieve.
<i>in_user</i>	The user requesting the job. Default: All users.

**Returns**

The Job, if it could be found; an empty pointer otherwise.

**4.2.3.3 getJobs()**

```
api::JobList rstudio::launcher_plugins::jobs::AbstractJobRepository::getJobs (
    const system::User & in_user = system::User() ) const
```

Gets all jobs belonging to the specified user.

If the user object represents "all users", all jobs will be returned.

**Parameters**

<i>in_user</i>	The user for whom to retrieve all jobs. Default: All users.
----------------	---

**Returns**

All of the jobs belonging to the specified user.

#### 4.2.3.4 initialize()

```
Error rstudio::launcher_plugins::jobs::AbstractJobRepository::initialize ( )
```

Initializes the [AbstractJobRepository](#).

##### Returns

[Success](#) if the repository could be initialized; [Error](#) otherwise.

#### 4.2.3.5 removeJob()

```
void rstudio::launcher_plugins::jobs::AbstractJobRepository::removeJob (
    const std::string & in_jobId )
```

Removes a job from the repository.

If there is no job with the specified id, nothing will happen.

##### Parameters

<i>in_↔ jobId</i>	The ID of the job to remove.
-----------------------	------------------------------

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

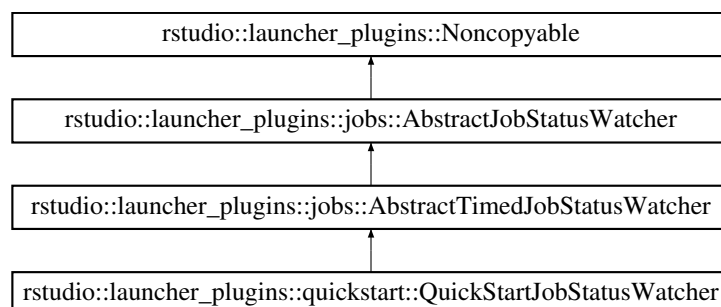
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/jobs/AbstractJobRepository.hpp

## 4.3 rstudio::launcher\_plugins::jobs::AbstractJobStatusWatcher Class Reference

Manages posting job status updates to the notifier.

```
#include <AbstractJobStatusWatcher.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::jobs::AbstractJobStatusWatcher:



## Public Member Functions

- virtual [~AbstractJobStatusWatcher](#) ()=default  
*Virtual destructor for inheritance.*
- virtual [Error start](#) ()=0  
*Starts the job status watcher.*
- virtual void [stop](#) ()=0  
*Stops the job status watcher.*

## Protected Member Functions

- [AbstractJobStatusWatcher](#) (JobRepositoryPtr in\_jobRepository, JobStatusNotifierPtr in\_jobStatusNotifier)  
*Constructor.*
- [Error updateJobStatus](#) (const std::string &in\_jobId, [api::Job::State](#) in\_newStatus, const std::string &in\_statusMessage="", const [system::DateTime](#) &in\_invocationTime=[system::DateTime](#)())  
*Updates the job status for the specified job.*

### 4.3.1 Detailed Description

Manages posting job status updates to the notifier.

### 4.3.2 Constructor & Destructor Documentation

#### 4.3.2.1 AbstractJobStatusWatcher()

```
rstudio::launcher_plugins::jobs::AbstractJobStatusWatcher::AbstractJobStatusWatcher (
    JobRepositoryPtr in_jobRepository,
    JobStatusNotifierPtr in_jobStatusNotifier ) [protected]
```

Constructor.

Parameters

<i>in_jobRepository</i>	The job repository, from which to look-up jobs.
<i>in_jobStatusNotifier</i>	The job status notifier to which to post job updates.

### 4.3.3 Member Function Documentation

#### 4.3.3.1 updateJobStatus()

```
Error rstudio::launcher_plugins::jobs::AbstractJobStatusWatcher::updateJobStatus (
    const std::string & in_jobId,
```

```

    api::Job::State in_newStatus,
    const std::string & in_statusMessage = "",
    const system::DateTime & in_invocationTime = system::DateTime() ) [protected]

```

Updates the job status for the specified job.

Attempts to look up the job in the repository. If the job cannot be found `getJobDetails(...)` will be invoked.

#### Parameters

<code>in_jobId</code>	The ID of the job which should be updated.
<code>in_newStatus</code>	The new status of the job.
<code>in_statusMessage</code>	The new status message of the job, if any.
<code>in_invocationTime</code>	The time at which the job was updated, if different from now.

#### Returns

**Success** if the job could be found and updated; **Error** otherwise.

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

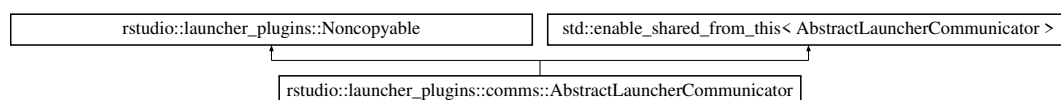
- `/workspaces/rstudio-launcher-plugin-sdk/sdk/include/jobs/AbstractJobStatusWatcher.hpp`

## 4.4 rstudio::launcher\_plugins::comms::AbstractLauncherCommunicator Class Reference

Base class responsible for communicating the the launcher. The type of communication is implementation dependent.

```
#include <AbstractLauncherCommunicator.hpp>
```

Inheritance diagram for `rstudio::launcher_plugins::comms::AbstractLauncherCommunicator`:



### Public Member Functions

- virtual `~AbstractLauncherCommunicator()`=default  
*Virtual destructor to allow for inheritance.*
- void `registerRequestHandler` (std::unique\_ptr< RequestHandler > &&in\_requestHandler)  
*Registers a request handler for all requests.*
- void `sendResponse` (const api::Response &in\_response)  
*Sends the response to the RStudio Launcher.*
- virtual `Error start()`  
*Starts the communicator.*
- virtual void `stop()`  
*Stops the communicator.*
- virtual void `waitForExit()`  
*Blocks until the communicator has successfully stopped.*

## Protected Member Functions

- [AbstractLauncherCommunicator](#) (size\_t in\_maxMessageSize, const [OnError](#) &in\_onError)  
*Constructor.*
- void [reportError](#) (const [Error](#) &in\_error)  
*Reports an error and stops the communicator.*
- void [onDataReceived](#) (const char \*in\_data, size\_t in\_length)  
*Handles data that is received from the RStudio Launcher.*
- template<typename Derived >  
std::shared\_ptr< Derived > [shared\\_from\\_derived](#) ()

### 4.4.1 Detailed Description

Base class responsible for communicating the the launcher. The type of communication is implementation dependent.

### 4.4.2 Constructor & Destructor Documentation

#### 4.4.2.1 AbstractLauncherCommunicator()

```
rstudio::launcher_plugins::comms::AbstractLauncherCommunicator::AbstractLauncherCommunicator (
    size_t in_maxMessageSize,
    const OnError & in_onError ) [protected]
```

Constructor.

##### Parameters

<i>in_maxMessageSize</i>	The maximum allowable size of a message which can be received from or sent to the RStudio Launcher.
<i>in_onError</i>	<a href="#">Error</a> handler to allow the creator of this communicator to receive communications errors.

### 4.4.3 Member Function Documentation

#### 4.4.3.1 onDataReceived()

```
void rstudio::launcher_plugins::comms::AbstractLauncherCommunicator::onDataReceived (
    const char * in_data,
    size_t in_length ) [protected]
```

Handles data that is received from the RStudio Launcher.

## Parameters

<i>in_data</i>	The data received from the RStudio Launcher.
<i>in_length</i>	The length of the data received from the RStudio Launcher.

**4.4.3.2 registerRequestHandler()**

```
void rstudio::launcher_plugins::comms::AbstractLauncherCommunicator::registerRequestHandler (
    std::unique_ptr< RequestHandler > && in_requestHandler )
```

Registers a request handler for all requests.

## Parameters

<i>in_requestHandler</i>	The handler for the request.
--------------------------	------------------------------

**4.4.3.3 reportError()**

```
void rstudio::launcher_plugins::comms::AbstractLauncherCommunicator::reportError (
    const Error & in_error ) [protected]
```

Reports an error and stops the communicator.

## Parameters

<i>in_error</i>	The error to report.
-----------------	----------------------

**4.4.3.4 sendResponse()**

```
void rstudio::launcher_plugins::comms::AbstractLauncherCommunicator::sendResponse (
    const api::Response & in_response )
```

Sends the response to the RStudio Launcher.

## Parameters

<i>in_response</i>	The response to be sent to the RStudio Launcher.
--------------------	--

#### 4.4.3.5 shared\_from\_derived()

```
template<typename Derived >
std::shared_ptr<Derived> rstudio::launcher_plugins::comms::AbstractLauncherCommunicator←
::shared_from_derived ( ) [inline], [protected]
```

@breif Template method which allows classes which inherit [AbstractLauncherCommunicator](#) to get a shared\_ptr to themselves.

##### Template Parameters

<i>Derived</i>	Type of the class which inherits from <a href="#">AbstractLauncherCommunicator</a> .
----------------	--

##### Returns

A shared\_ptr to this derived class.

#### 4.4.3.6 start()

```
virtual Error rstudio::launcher_plugins::comms::AbstractLauncherCommunicator::start ( ) [virtual]
```

Starts the communicator.

Child classes which override this method should also invoke the base method.

##### Returns

[Success](#) if the communicator could be started; [Error](#) otherwise.

#### 4.4.3.7 stop()

```
virtual void rstudio::launcher_plugins::comms::AbstractLauncherCommunicator::stop ( ) [virtual]
```

Stops the communicator.

Child classes which override this method should also invoke the base method.

#### 4.4.3.8 waitForExit()

```
virtual void rstudio::launcher_plugins::comms::AbstractLauncherCommunicator::waitForExit ( )
[virtual]
```

Blocks until the communicator has successfully stopped.

Child classes which override this method should also invoke the base method.

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

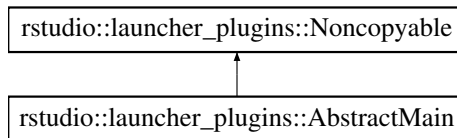
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/comms/AbstractLauncherCommunicator.hpp

## 4.5 rstudio::launcher\_plugins::AbstractMain Class Reference

Base class for the PluginMain class, which runs the plugin.

```
#include <AbstractMain.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::AbstractMain:



### Public Member Functions

- virtual [~AbstractMain](#) ()=default  
*Default destructor.*
- int [run](#) (int in\_argCount, char \*\*in\_argList)  
*Runs the plugin.*

### Protected Member Functions

- [AbstractMain](#) ()  
*Default Constructor.*

#### 4.5.1 Detailed Description

Base class for the PluginMain class, which runs the plugin.

#### 4.5.2 Member Function Documentation

##### 4.5.2.1 run()

```
int rstudio::launcher_plugins::AbstractMain::run (
    int in_argCount,
    char ** in_argList )
```

Runs the plugin.

##### Parameters

<i>in_argCount</i>	The number of arguments in in_argList.
<i>in_argList</i>	The argument list to the program.



**Returns**

0 on a successful exit. A non-zero error code, otherwise.

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

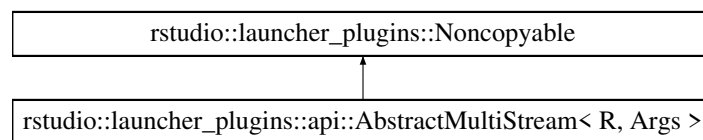
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/AbstractMain.hpp

## 4.6 rstudio::launcher\_plugins::api::AbstractMultiStream< R, Args > Class Template Reference

Manages the sending of streamed responses.

```
#include <AbstractMultiStream.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::AbstractMultiStream< R, Args >:

**Public Member Functions**

- [AbstractMultiStream](#) (comms::AbstractLauncherCommunicatorPtr in\_launcherCommunicator)  
*Constructor.*
- virtual void [addRequest](#) (uint64\_t in\_requestId, const [system::User](#) &in\_requestUser)=0  
*Adds a request to the stream.*
- virtual [Error initialize](#) ()=0  
*Initializes the response stream.*
- bool [isEmpty](#) () const  
*Checks whether there are any requests listening to this stream.*
- virtual void [removeRequest](#) (uint64\_t in\_requestId)  
*Removes a request from the multi-stream response.*

**Protected Member Functions**

- void [onAddRequest](#) (uint64\_t in\_requestId)  
*Adds a new request ID to the multi-stream response.*
- void [onRemoveRequest](#) (uint64\_t in\_requestId)  
*Removes a request from the multi-stream response.*
- void [sendResponse](#) (Args... in\_responseArgs)  
*Sends a response to the Launcher for all requests.*
- void [sendResponse](#) (const std::set< uint64\_t > &in\_requestIds, Args... in\_responseArgs)  
*Sends a response to the Launcher for the specified requests.*

## Protected Attributes

- std::mutex [m\\_mutex](#)

### 4.6.1 Detailed Description

```
template<typename R, typename ... Args>
class rstudio::launcher_plugins::api::AbstractMultiStream< R, Args >
```

Manages the sending of streamed responses.

#### Template Parameters

<i>R</i>	The <a href="#">Response</a> type which should be sent.
----------	---

#### Parameters

<i>Args</i>	The additional constructor parameters of R, besides the request and sequence IDs.
-------------	---

### 4.6.2 Constructor & Destructor Documentation

#### 4.6.2.1 AbstractMultiStream()

```
template<typename R, typename ... Args>
rstudio::launcher_plugins::api::AbstractMultiStream< R, Args >::AbstractMultiStream (
    comms::AbstractLauncherCommunicatorPtr in_launcherCommunicator ) [explicit]
```

Constructor.

#### Parameters

<i>in_launcherCommunicator</i>	The launcher communicator which will send the responses.
--------------------------------	--

### 4.6.3 Member Function Documentation

#### 4.6.3.1 addRequest()

```
template<typename R, typename ... Args>
virtual void rstudio::launcher_plugins::api::AbstractMultiStream< R, Args >::addRequest (
```

```
uint64_t in_requestId,  
const system::User & in_requestUser ) [pure virtual]
```

Adds a request to the stream.

**Parameters**

<i>in_requestId</i>	The ID of the request.
<i>in_requestUser</i>	The user who made the request.

Implemented in [rstudio::launcher\\_plugins::api::AbstractResourceStream](#).

**4.6.3.2 initialize()**

```
template<typename R, typename ... Args>
virtual Error rstudio::launcher_plugins::api::AbstractMultiStream< R, Args >::initialize ( )
[pure virtual]
```

Initializes the response stream.

**Returns**

[Success](#) if the response stream could be initialized; [Error](#) otherwise.

Implemented in [rstudio::launcher\\_plugins::api::AbstractResourceStream](#), [rstudio::launcher\\_plugins::api::AbstractTimedResourceStream](#) and [rstudio::launcher\\_plugins::quickstart::QuickStartResourceStream](#).

**4.6.3.3 isEmpty()**

```
template<typename R, typename ... Args>
bool rstudio::launcher_plugins::api::AbstractMultiStream< R, Args >::isEmpty ( ) const
```

Checks whether there are any requests listening to this stream.

**Returns**

True if this stream has any requests; false otherwise.

**4.6.3.4 onAddRequest()**

```
template<typename R, typename ... Args>
void rstudio::launcher_plugins::api::AbstractMultiStream< R, Args >::onAddRequest (
    uint64_t in_requestId ) [protected]
```

Adds a new request ID to the multi-stream response.

NOTE: The mutex must be held when this is called.

## Parameters

<i>in_↔ requestId</i>	The request ID which is listening to this response stream.
---------------------------	--

**4.6.3.5 onRemoveRequest()**

```
template<typename R, typename ... Args>
void rstudio::launcher_plugins::api::AbstractMultiStream< R, Args >::onRemoveRequest (
    uint64_t in_requestId ) [protected]
```

Removes a request from the multi-stream response.

NOTE: The mutex must be held when this is called.

## Parameters

<i>in_↔ requestId</i>	The request ID which has stopped listening to this response stream.
---------------------------	---

**4.6.3.6 removeRequest()**

```
template<typename R, typename ... Args>
virtual void rstudio::launcher_plugins::api::AbstractMultiStream< R, Args >::removeRequest (
    uint64_t in_requestId ) [virtual]
```

Removes a request from the multi-stream response.

## Parameters

<i>in_↔ requestId</i>	The request ID which has stopped listening to this response stream.
---------------------------	---

**4.6.3.7 sendResponse() [1/2]**

```
template<typename R, typename ... Args>
void rstudio::launcher_plugins::api::AbstractMultiStream< R, Args >::sendResponse (
    Args... in_responseArgs ) [protected]
```

Sends a response to the Launcher for all requests.

NOTE: The mutex must be held when this is called.

## Parameters

<i>in_responseArgs</i>	The details of the response, if any.
------------------------	--------------------------------------

**4.6.3.8 sendResponse() [2/2]**

```
template<typename R, typename ... Args>
void rstudio::launcher_plugins::api::AbstractMultiStream< R, Args >::sendResponse (
    const std::set< uint64_t > & in_requestIds,
    Args... in_responseArgs ) [protected]
```

Sends a response to the Launcher for the specified requests.

NOTE: The mutex must be held when this is called.

## Parameters

<i>in_requestIds</i>	Sends the response only for the specified request IDs.
<i>in_responseArgs</i>	The details of the response, if any.

**4.6.4 Member Data Documentation****4.6.4.1 m\_mutex**

```
template<typename R, typename ... Args>
std::mutex rstudio::launcher_plugins::api::AbstractMultiStream< R, Args >::m_mutex [mutable],
[protected]
```

Mutex to protect shared state of the stream.

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

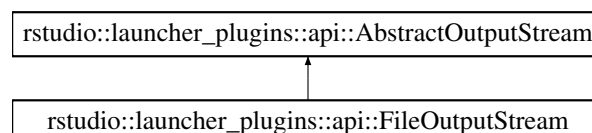
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/stream/AbstractMultiStream.hpp

**4.7 rstudio::launcher\_plugins::api::AbstractOutputStream Class Reference**

Streams job output data to the launcher.

```
#include <AbstractOutputStream.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::AbstractOutputStream:



## Public Types

- typedef std::function< void(uint64\_t)> [OnComplete](#)
- typedef std::function< void(const [Error](#) &)> [OnError](#)
- typedef std::function< void(const std::string &, OutputType, uint64\_t)> [OnOutput](#)

## Public Member Functions

- virtual [~AbstractOutputStream](#) ()=default  
*Virtual destructor for inheritance.*
- virtual [Error](#) [start](#) ()=0  
*Starts the output stream.*
- virtual void [stop](#) ()=0  
*Stops the output stream.*

## Protected Member Functions

- [AbstractOutputStream](#) (OutputType in\_outputType, JobPtr in\_job, OnOutput in\_onOutput, [OnComplete](#) in\_onComplete, OnError in\_onError)  
*Constructor.*
- void [reportData](#) (const std::string &in\_data, OutputType in\_outputType)  
*Reports output to the launcher.*
- void [reportError](#) (const [Error](#) &in\_error)  
*Reports an error to the launcher.*
- void [setStreamComplete](#) ()  
*Notifies the base class that the output stream has completed (i.e. all output of the specified type has been reported).*

## Protected Attributes

- OutputType [m\\_outputType](#)
- JobPtr [m\\_job](#)

### 4.7.1 Detailed Description

Streams job output data to the launcher.

### 4.7.2 Member Typedef Documentation

#### 4.7.2.1 OnComplete

```
typedef std::function<void(uint64_t)> rstudio::launcher\_plugins::api::AbstractOutputStream::OnComplete
```

Definitions for callback functions which will be invoked when certain events occur.

### 4.7.3 Constructor & Destructor Documentation

#### 4.7.3.1 AbstractOutputStream()

```
rstudio::launcher_plugins::api::AbstractOutputStream::AbstractOutputStream (
    OutputType in_outputType,
    JobPtr in_job,
    OnOutput in_onOutput,
    OnComplete in_onComplete,
    OnError in_onError ) [protected]
```

Constructor.

##### Parameters

<i>in_requestId</i>	The ID of the request for which job output should be streamed.
<i>in_outputType</i>	The type of job output to stream.
<i>in_onOutput</i>	Callback function which will be invoked when data is reported.
<i>in_onComplete</i>	Callback function which will be invoked when the stream is complete.
<i>in_onError</i>	Callback function which will be invoked if an error occurs.
<i>in_job</i>	The job for which output should be streamed.

### 4.7.4 Member Function Documentation

#### 4.7.4.1 reportData()

```
void rstudio::launcher_plugins::api::AbstractOutputStream::reportData (
    const std::string & in_data,
    OutputType in_outputType ) [protected]
```

Reports output to the launcher.

##### Parameters

<i>in_data</i>	The output data.
<i>in_outputType</i>	The type of output data.

#### 4.7.4.2 reportError()

```
void rstudio::launcher_plugins::api::AbstractOutputStream::reportError (
    const Error & in_error ) [protected]
```



Reports an error to the launcher.

## Parameters

<code>in_error</code>	The error which occurred.
-----------------------	---------------------------

**4.7.4.3 start()**

```
virtual Error rstudio::launcher_plugins::api::AbstractOutputStream::start ( ) [pure virtual]
```

Starts the output stream.

## Returns

[Success](#) if the stream could be started; [Error](#) otherwise.

Implemented in [rstudio::launcher\\_plugins::api::FileOutputStream](#).

**4.7.5 Member Data Documentation****4.7.5.1 m\_job**

```
JobPtr rstudio::launcher_plugins::api::AbstractOutputStream::m_job [protected]
```

The job for which output should be streamed.

**4.7.5.2 m\_outputType**

```
OutputType rstudio::launcher_plugins::api::AbstractOutputStream::m_outputType [protected]
```

The type of output that should be streamed.

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

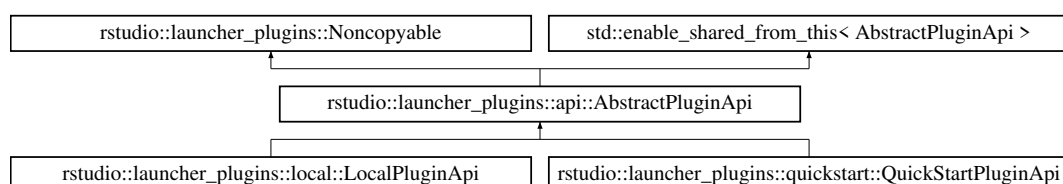
- `/workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/stream/AbstractOutputStream.hpp`

**4.8 rstudio::launcher\_plugins::api::AbstractPluginApi Class Reference**

Base class for the Launcher Plugin API.

```
#include <AbstractPluginApi.hpp>
```

Inheritance diagram for `rstudio::launcher_plugins::api::AbstractPluginApi`:



## Public Member Functions

- virtual [~AbstractPluginApi](#) ()=default  
*Virtual destructor.*
- [Error initialize](#) ()  
*This method initializes the abstract plugin API.*

## Protected Member Functions

- [AbstractPluginApi](#) (std::shared\_ptr< [comms::AbstractLauncherCommunicator](#) > in\_launcherCommunicator)  
*Constructor.*

### 4.8.1 Detailed Description

Base class for the Launcher Plugin API.

### 4.8.2 Constructor & Destructor Documentation

#### 4.8.2.1 AbstractPluginApi()

```
rstudio::launcher_plugins::api::AbstractPluginApi::AbstractPluginApi (
    std::shared_ptr< comms::AbstractLauncherCommunicator > in_launcherCommunicator )
[explicit], [protected]
```

Constructor.

Parameters

<i>in_launcherCommunicator</i>	The communicator to use for sending and receiving messages from the RStudio Launcher.
--------------------------------	---

### 4.8.3 Member Function Documentation

#### 4.8.3.1 initialize()

```
Error rstudio::launcher_plugins::api::AbstractPluginApi::initialize ( )
```

This method initializes the abstract plugin API.

## Returns

[Success](#) if the abstract plugin API was successfully initialized; [Error](#) otherwise.

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

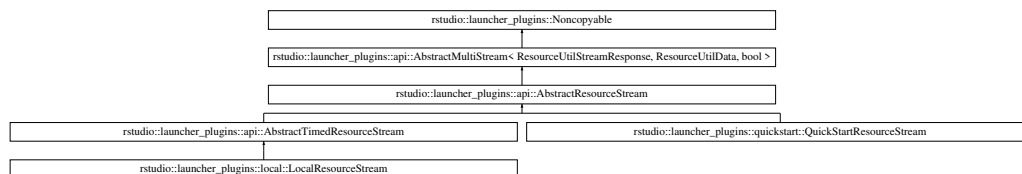
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/AbstractPluginApi.hpp

## 4.9 rstudio::launcher\_plugins::api::AbstractResourceStream Class Reference

Streams job resource utilization data to the Launcher.

```
#include <AbstractResourceStream.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::AbstractResourceStream:



### Public Member Functions

- virtual [~AbstractResourceStream](#) ()=default  
*Virtual destructor for inheritance.*
- void [addRequest](#) (uint64\_t in\_requestId, const [system::User](#) &in\_requestUser) override  
*Adds a request to the stream.*
- virtual [Error initialize](#) ()=0  
*Initializes the resource utilization stream.*
- void [setStreamComplete](#) ()  
*Notifies that the data stream has completed.*

### Protected Member Functions

- [AbstractResourceStream](#) (const ConstJobPtr &in\_job, comms::AbstractLauncherCommunicatorPtr in\_launcherCommunicator)  
*Constructor.*
- void [reportData](#) (const [ResourceUtilData](#) &in\_data)  
*Reports resource utilization data to the Launcher.*
- void [reportError](#) (const [Error](#) &in\_error)  
*Reports an error to the Launcher.*

### Protected Attributes

- const ConstJobPtr [m\\_job](#)  
*The job for which resource utilization metrics should be streamed.*

## 4.9.1 Detailed Description

Streams job resource utilization data to the Launcher.

## 4.9.2 Constructor & Destructor Documentation

### 4.9.2.1 AbstractResourceStream()

```
rstudio::launcher_plugins::api::AbstractResourceStream::AbstractResourceStream (
    const ConstJobPtr & in_job,
    comms::AbstractLauncherCommunicatorPtr in_launcherCommunicator ) [protected]
```

Constructor.

#### Parameters

<i>in_job</i>	The job for which resource utilization metrics should be streamed.
<i>in_launcherCommunicator</i>	The communicator through which messages may be sent to the launcher.

## 4.9.3 Member Function Documentation

### 4.9.3.1 addRequest()

```
void rstudio::launcher_plugins::api::AbstractResourceStream::addRequest (
    uint64_t in_requestId,
    const system::User & in_requestUser ) [override], [virtual]
```

Adds a request to the stream.

#### Parameters

<i>in_requestId</i>	The ID of the request.
<i>in_requestUser</i>	The user who made the request.

Implements [rstudio::launcher\\_plugins::api::AbstractMultiStream< ResourceUtilStreamResponse, ResourceUtilData, bool >](#).

### 4.9.3.2 initialize()

```
virtual Error rstudio::launcher_plugins::api::AbstractResourceStream::initialize ( ) [pure virtual]
```

Initializes the resource utilization stream.

**Returns**

[Success](#) if resource utilization streaming was started correctly; [Error](#) otherwise.

Implements [rstudio::launcher\\_plugins::api::AbstractMultiStream< ResourceUtilStreamResponse, ResourceUtilData, bool >](#).

Implemented in [rstudio::launcher\\_plugins::api::AbstractTimedResourceStream](#), and [rstudio::launcher\\_plugins::quickstart::QuickStartP](#)

**4.9.3.3 reportData()**

```
void rstudio::launcher_plugins::api::AbstractResourceStream::reportData (
    const ResourceUtilData & in_data ) [protected]
```

Reports resource utilization data to the Launcher.

**Parameters**

<i>in_data</i>	The new resource utilization data for this job.
----------------	---

**4.9.3.4 reportError()**

```
void rstudio::launcher_plugins::api::AbstractResourceStream::reportError (
    const Error & in_error ) [protected]
```

Reports an error to the Launcher.

Additional calls to reportError, reportData, or setStreamComplete will be ignored.

**Parameters**

<i>in_error</i>	The error which occurred.
-----------------	---------------------------

**4.9.3.5 setStreamComplete()**

```
void rstudio::launcher_plugins::api::AbstractResourceStream::setStreamComplete ( )
```

Notifies that the data stream has completed.

Additional calls to reportError, reportData, or setStreamComplete will be ignored.

**4.9.4 Member Data Documentation**

## 4.9.4.1 m\_job

```
const ConstJobPtr rstudio::launcher_plugins::api::AbstractResourceStream::m_job [protected]
```

The job for which resource utilization metrics should be streamed.

NOTE: To avoid potential deadlock scenarios, the lock for the mutex on the base class, m\_mutex, must be held when the job lock is acquired, and the job lock must be released before the mutex lock is released. For consistency, it is recommended to use the following block of code to acquire both locks:

```
LOCK_MUTEX_AND_JOB(std::lock_guard, std::mutex, m_mutex, m_job) { // Do tasks which require the Job Lock.
} END_LOCK_MUTEX_AND_JOB
```

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

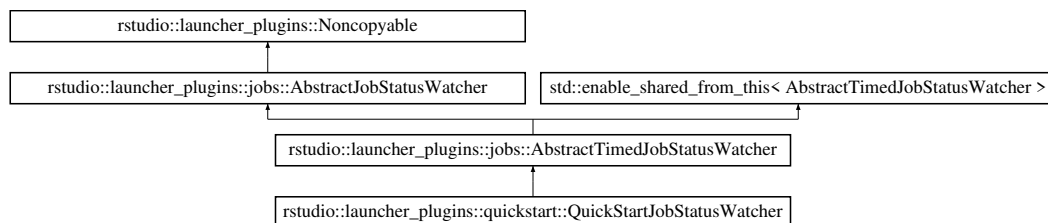
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/stream/AbstractResourceStream.hpp

## 4.10 rstudio::launcher\_plugins::jobs::AbstractTimedJobStatusWatcher Class Reference

Responsible for polling job statuses on a timer.

```
#include <AbstractTimedJobStatusWatcher.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::jobs::AbstractTimedJobStatusWatcher:



### Public Member Functions

- [~AbstractTimedJobStatusWatcher](#) () noexcept override  
*Virtual destructor for inheritance. Invokes [stop\(\)](#).*
- [Error start](#) () final  
*Starts the timed job status watcher.*
- void [stop](#) () final  
*Stops the timed job status watcher.*

### Protected Member Functions

- [AbstractTimedJobStatusWatcher](#) (system::TimeDuration in\_frequency, JobRepositoryPtr in\_jobRepository, JobStatusNotifierPtr in\_jobStatusNotifier)  
*Constructor.*

### 4.10.1 Detailed Description

Responsible for polling job statuses on a timer.

### 4.10.2 Constructor & Destructor Documentation

#### 4.10.2.1 AbstractTimedJobStatusWatcher()

```
rstudio::launcher_plugins::jobs::AbstractTimedJobStatusWatcher::AbstractTimedJobStatusWatcher
(
    system::TimeDuration in_frequency,
    JobRepositoryPtr in_jobRepository,
    JobStatusNotifierPtr in_jobStatusNotifier ) [protected]
```

Constructor.

#### Parameters

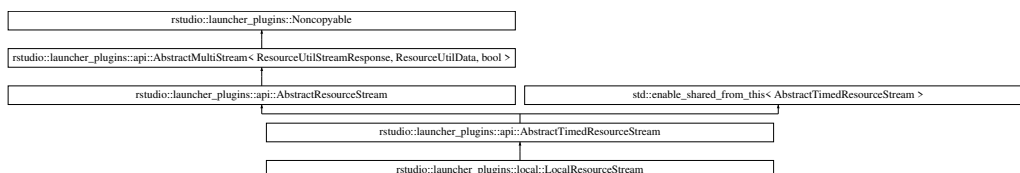
<i>in_frequency</i>	The frequency at which job statuses should be polled.
<i>in_jobRepository</i>	The job repository, from which to look-up jobs.
<i>in_jobStatusNotifier</i>	The job status notifier to which to post job updates.

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/jobs/AbstractTimedJobStatusWatcher.hpp

## 4.11 rstudio::launcher\_plugins::api::AbstractTimedResourceStream Class Reference

Inheritance diagram for rstudio::launcher\_plugins::api::AbstractTimedResourceStream:



### Public Member Functions

- virtual [~AbstractTimedResourceStream](#) ()  
*Virtual destructor.*
- [Error initialize](#) () override  
*Initializes the timed resource utilization stream.*



## Protected Member Functions

- [AbstractTimedResourceStream](#) ([system::TimeDuration](#) in\_frequency, const ConstJobPtr &in\_job, comms::AbstractLauncherCommunicatorPtr in\_launcherCommunicator)

*Constructor.*

## Additional Inherited Members

### 4.11.1 Constructor & Destructor Documentation

#### 4.11.1.1 AbstractTimedResourceStream()

```
rstudio::launcher_plugins::api::AbstractTimedResourceStream::AbstractTimedResourceStream (
    system::TimeDuration in_frequency,
    const ConstJobPtr & in_job,
    comms::AbstractLauncherCommunicatorPtr in_launcherCommunicator ) [protected]
```

Constructor.

#### Parameters

<i>in_frequency</i>	The frequency at which job resource utilization metrics should be polled.
<i>in_job</i>	The job for which resource utilization metrics should be streamed.
<i>in_launcherCommunicator</i>	The communicator through which messages may be sent to the launcher.

### 4.11.2 Member Function Documentation

#### 4.11.2.1 initialize()

```
Error rstudio::launcher_plugins::api::AbstractTimedResourceStream::initialize ( ) [override],
[virtual]
```

Initializes the timed resource utilization stream.

#### Returns

[Success](#) if resource utilization streaming was started correctly; [Error](#) otherwise.

Implements [rstudio::launcher\\_plugins::api::AbstractResourceStream](#).

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

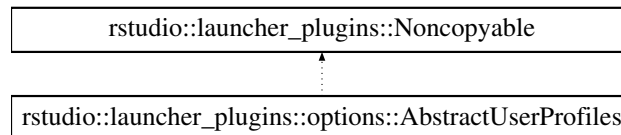
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/stream/AbstractTimedResourceStream.hpp

## 4.12 rstudio::launcher\_plugins::options::AbstractUserProfiles Class Reference

Base class which reads an ini-based user profiles file.

```
#include <AbstractUserProfiles.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::options::AbstractUserProfiles:



### Public Member Functions

- virtual [~AbstractUserProfiles](#) ()=default  
*Default virtual destructor for inheritance.*
- [Error initialize](#) ()  
*Initializes the user profiles. Must be called before attempting to retrieve configuration values.*

### Protected Member Functions

- [AbstractUserProfiles](#) ()  
*Default constructor.*
- [AbstractUserProfiles](#) (const std::string &in\_pluginName)  
*Constructor.*
- template<typename T >  
[Error getValueForUser](#) (const std::string &in\_valueName, const [system::User](#) &in\_user, T &out\_value) const  
*Gets the value with the specified name for the given user, based on the profiles configuration file.*
- template<typename T >  
[Error validateValue](#) (const std::string &in\_valueName) const  
*Parses all occurrences of the configuration setting with name in\_valueName to validate it is correctly formatted.*
- [Error validateValue](#) (const std::string &in\_valueName, const CustomValueValidator &in\_validator) const  
*Validates all occurrences of a value which has a non-standard type.*

### Static Protected Member Functions

- static bool [isValueNotFoundError](#) (const [Error](#) &in\_error)  
*Checks whether the error indicates that the configuration value was not found.*

#### 4.12.1 Detailed Description

Base class which reads an ini-based user profiles file.

## 4.12.2 Constructor & Destructor Documentation

### 4.12.2.1 AbstractUserProfiles() [1/2]

```
rstudio::launcher_plugins::options::AbstractUserProfiles::AbstractUserProfiles ( ) [protected]
```

Default constructor.

This constructor should only be used if the inheriting class overrides `getConfigurationFileName`.

If this constructor is used, the user profiles file will be `/etc/rstudio/<getConfigurationFileName()>.conf`.

### 4.12.2.2 AbstractUserProfiles() [2/2]

```
rstudio::launcher_plugins::options::AbstractUserProfiles::AbstractUserProfiles (
    const std::string & in_pluginName ) [explicit], [protected]
```

Constructor.

If this constructor is used, the user profiles file will be `/etc/rstudio/launcher.<in_pluginName>.profiles.conf`.

Parameters

<code>in_pluginName</code>	The lower-case name of the plugin, to be used to set the configuration file name.
----------------------------	---

## 4.12.3 Member Function Documentation

### 4.12.3.1 getValueForUser()

```
template<typename T >
Error rstudio::launcher_plugins::options::AbstractUserProfiles::getValueForUser (
    const std::string & in_valueName,
    const system::User & in_user,
    T & out_value ) const [protected]
```

Gets the value with the specified name for the given user, based on the profiles configuration file.

This template method is instantiated in the CPP for a fixed set of types. Supported types: `std::string` `int32_t` `uint32_t` `int64_t` `uint64_t` `float` `bool` `std::set<U>`, where U is one of the types above `std::vector<U>`, where U is one of the types above (except `std::set`) `std::map<U, V>` where U and V are any two of the types above

If additional types are required, this method should be invoked using `T=std::string` and the caller may parse the string value as desired.

### Template Parameters

<i>T</i>	The type of the value, as defined above.
----------	--

### Parameters

<i>in_valueName</i>	The name of the value to retrieve for the specified user.
<i>in_user</i>	The user for which to retrieve the value.
<i>out_value</i>	The requested value, if no error occurred.

### Returns

**Success** if the requested value could be found for the given user and could be parsed to type *T*; **Error** otherwise.

#### 4.12.3.2 initialize()

```
Error rstudio::launcher_plugins::options::AbstractUserProfiles::initialize ( )
```

Initializes the user profiles. Must be called before attempting to retrieve configuration values.

### Returns

**Success** if the user profiles file could be opened for read and parsed; **Error** otherwise.

#### 4.12.3.3 isValueNotFoundError()

```
static bool rstudio::launcher_plugins::options::AbstractUserProfiles::isValueNotFoundError (
    const Error & in_error ) [static], [protected]
```

Checks whether the error indicates that the configuration value was not found.

This method may be used to handle missing values differently than invalid values. For example, the Plugin developer may choose to use a default value if none was specified in the configuration file.

### Parameters

<i>in_error</i>	The error to check.
-----------------	---------------------

### Returns

True if this error is a value-not-found error; false otherwise.

#### 4.12.3.4 validateValue() [1/2]

```
template<typename T >
Error rstudio::launcher_plugins::options::AbstractUserProfiles::validateValue (
    const std::string & in_valueName ) const [protected]
```

Parses all occurrences of the configuration setting with name `in_valueName` to validate it is correctly formatted.

##### Template Parameters

<i>T</i>	The type of the value. This is a precompiled templated function. Possible types are described in the documentation of <code>getValueForUser</code> .
----------	--

##### Parameters

<i>in_valueName</i>	The name of the value to validate.
---------------------	------------------------------------

##### Returns

**Success** if all occurrences of `in_valueName` within the profiles configuration file could be parsed; **Error** otherwise.

#### 4.12.3.5 validateValue() [2/2]

```
Error rstudio::launcher_plugins::options::AbstractUserProfiles::validateValue (
    const std::string & in_valueName,
    const CustomValueValidator & in_validator ) const [protected]
```

Validates all occurrences of a value which has a non-standard type.

##### Parameters

<i>in_valueName</i>	The name of the value to validate.
<i>in_validator</i>	The custom validation function. This should return success if the value could be parsed correctly and an error otherwise.

##### Returns

**Success** if all occurrences of `in_valueName` within the profiles configuration file could be parsed; **Error** otherwise.

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

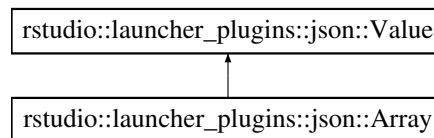
- `/workspaces/rstudio-launcher-plugin-sdk/sdk/include/options/AbstractUserProfiles.hpp`

## 4.13 rstudio::launcher\_plugins::json::Array Class Reference

Class which represents a JSON array.

```
#include <Json.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::json::Array:



### Classes

- class [Iterator](#)  
*Class which allows iterating over the elements of a JSON array.*

### Public Types

- typedef [Value](#) [value\\_type](#)  
*Typedef required for the inheritance of `std::iterator` with a `value_type` of [json::Value](#).*
- typedef `std::reverse_iterator< Iterator >` [Reverseliterator](#)  
*Reverse iterator for a JSON array.*

### Public Member Functions

- [Array](#) ()  
*Constructs an empty JSON array.*
- [Array](#) (const StringPairList &in\_strPairs)  
*Constructs a JSON array from a list of string pairs as an array of strings in the format "key=value".*
- [Array](#) (const [Array](#) &in\_other)  
*Copy constructor.*
- [Array](#) ([Array](#) &&in\_other) noexcept  
*Move constructor.*
- [Array](#) & operator= (const [Array](#) &in\_other)  
*Assignment operator.*
- [Array](#) & operator= ([Array](#) &&in\_other) noexcept  
*Move operator.*
- [Value](#) operator[] (size\_t in\_index) const  
*Accessor operator. Gets the JSON value at the specified position in the array.*
- [Iterator](#) begin () const  
*Gets an iterator pointing to the first member of this array.*
- [Iterator](#) end () const  
*Gets an iterator after the last member of this array.*
- [Reverseliterator](#) rbegin () const  
*Gets an iterator pointing to the last member of this array, which iterates in the reverse direction.*
- [Reverseliterator](#) rend () const

*Gets an iterator before the first member of this array, which can be compared with an other Reverseliterator to determine when reverse iteration has ended.*

- void **clear** ()  
*Clears the JSON array.*
- **Iterator erase** (const **Iterator** &in\_itr)  
*Erases the member specified by the provided iterator.*
- **Iterator erase** (const **Iterator** &in\_first, const **Iterator** &in\_last)  
*Erases a range of member specified by the provided iterators to the first and last members to erase.*
- **Value getBack** () const  
*Gets the value at the back of the JSON array.*
- **Value getFront** () const  
*Gets the value at the front of the JSON array.*
- **Value getValueAt** (size\_t in\_index) const  
*Gets the value at the specified index of the JSON array.*
- size\_t **getSize** () const  
*Gets the number of values in the JSON array.*
- bool **isEmpty** () const  
*Checks whether the JSON array is empty.*
- **Error parse** (const char \*in\_jsonStr) override  
*Parses the JSON string into this array.*
- **Error parse** (const std::string &in\_jsonStr) override  
*Parses the JSON string into this array.*
- void **push\_back** (const **Value** &in\_value)  
*Pushes the value onto the end of the JSON array.*
- void **push\_back** (bool in\_value)  
*Pushes the value onto the end of the JSON array.*
- void **push\_back** (double in\_value)  
*Pushes the value onto the end of the JSON array.*
- void **push\_back** (float in\_value)  
*Pushes the value onto the end of the JSON array.*
- void **push\_back** (int in\_value)  
*Pushes the value onto the end of the JSON array.*
- void **push\_back** (int64\_t in\_value)  
*Pushes the value onto the end of the JSON array.*
- void **push\_back** (const char \*in\_value)  
*Pushes the value onto the end of the JSON array.*
- void **push\_back** (const std::string &in\_value)  
*Pushes the value onto the end of the JSON array.*
- void **push\_back** (unsigned int in\_value)  
*Pushes the value onto the end of the JSON array.*
- void **push\_back** (uint64\_t in\_value)  
*Pushes the value onto the end of the JSON array.*
- void **push\_back** (const **Array** &in\_value)  
*Pushes the value onto the end of the JSON array.*
- void **push\_back** (const **Object** &in\_value)  
*Pushes the value onto the end of the JSON array.*
- bool **toSetString** (std::set< std::string > &out\_set) const  
*Converts this JSON array to a set of strings.*
- StringPairList **toStringPairList** () const  
*Converts this array into a vector of string pairs. Elements of the form "key=value" will be parsed into the pair < "key", "value">. Elements which are not in that format (e.g. "value") will be parsed as < "value", "">. Any elements of the array which are not strings will be skipped.*

- bool [toVectorInt](#) (std::vector< int > &out\_set) const  
*Converts this JSON array to a vector of integers.*
- bool [toVectorString](#) (std::vector< std::string > &out\_set) const  
*Converts this JSON array to a vector of strings.*

## Friends

- class [Value](#)

## Additional Inherited Members

### 4.13.1 Detailed Description

Class which represents a JSON array.

### 4.13.2 Constructor & Destructor Documentation

#### 4.13.2.1 [Array\(\)](#) [1/3]

```
rstudio::launcher_plugins::json::Array::Array (
    const StringPairList & in_strPairs ) [explicit]
```

Constructs a JSON array from a list of string pairs as an array of strings in the format "key=value".

#### Parameters

<i>in_strPairs</i>	The list of string pairs from which to construct this array.
--------------------	--

#### 4.13.2.2 [Array\(\)](#) [2/3]

```
rstudio::launcher_plugins::json::Array::Array (
    const Array & in_other )
```

Copy constructor.

#### Parameters

<i>in_other</i>	The JSON array to copy from.
-----------------	------------------------------



### 4.13.2.3 Array() [3/3]

```
rstudio::launcher_plugins::json::Array::Array (
    Array && in_other ) [noexcept]
```

Move constructor.

Parameters

<code>in_other</code>	The JSON array to move to this <a href="#">Object</a> .
-----------------------	---

## 4.13.3 Member Function Documentation

### 4.13.3.1 begin()

```
Iterator rstudio::launcher_plugins::json::Array::begin ( ) const
```

Gets an iterator pointing to the first member of this array.

Returns

An iterator pointing to the first member of this array.

### 4.13.3.2 end()

```
Iterator rstudio::launcher_plugins::json::Array::end ( ) const
```

Gets an iterator after the last member of this array.

Returns

An iterator after the last member of this array.

### 4.13.3.3 erase() [1/2]

```
Iterator rstudio::launcher_plugins::json::Array::erase (
    const Iterator & in_first,
    const Iterator & in_last )
```

Erases a range of member specified by the provided iterators to the first and last members to erase.

**Parameters**

<i>in_first</i>	The iterator pointing to the first member to erase.
<i>in_last</i>	The iterator pointing to the last member to erase.

**Returns**

An iterator pointing to the member immediately after the last erased member.

**4.13.3.4 erase() [2/2]**

```
Iterator rstudio::launcher_plugins::json::Array::erase (
    const Iterator & in_itr )
```

Erases the member specified by the provided iterator.

**Parameters**

<i>in_itr</i>	The iterator pointing to the member to erase.
---------------	---

**Returns**

An iterator pointing to the member immediately after the erased member.

**4.13.3.5 getBack()**

```
Value rstudio::launcher_plugins::json::Array::getBack ( ) const
```

Gets the value at the back of the JSON array.

**Returns**

The value at the back of the JSON array or an empty value, if the array is empty.

**4.13.3.6 getFront()**

```
Value rstudio::launcher_plugins::json::Array::getFront ( ) const
```

Gets the value at the front of the JSON array.

**Returns**

The value at the front of the JSON array or an empty value, if the array is empty.

#### 4.13.3.7 getSize()

```
size_t rstudio::launcher_plugins::json::Array::getSize ( ) const
```

Gets the number of values in the JSON array.

##### Returns

The number of values in the JSON array.

#### 4.13.3.8 getValueAt()

```
Value rstudio::launcher_plugins::json::Array::getValueAt (
    size_t in_index ) const
```

Gets the value at the specified index of the JSON array.

##### Parameters

<i>in_index</i>	The index of the value to retrieve.
-----------------	-------------------------------------

##### Returns

The value at the specified index or an empty value if the index is out of bounds.

#### 4.13.3.9 isEmpty()

```
bool rstudio::launcher_plugins::json::Array::isEmpty ( ) const
```

Checks whether the JSON array is empty.

##### Returns

True if the JSON array has no members; false otherwise.

#### 4.13.3.10 operator=() [1/2]

```
Array& rstudio::launcher_plugins::json::Array::operator= (
    Array && in_other ) [noexcept]
```

Move operator.

**Parameters**

<i>in_other</i>	The JSON <a href="#">Array</a> to move from.
-----------------	--

**Returns**

A reference to this JSON [Array](#).

**4.13.3.11 operator=()** [2/2]

```
Array& rstudio::launcher_plugins::json::Array::operator= (
    const Array & in_other )
```

Assignment operator.

**Parameters**

<i>in_other</i>	The JSON array to copy from.
-----------------	------------------------------

**Returns**

A reference to this JSON array.

**4.13.3.12 operator[]()**

```
Value rstudio::launcher_plugins::json::Array::operator[] (
    size_t in_index ) const
```

Accessor operator. Gets the JSON value at the specified position in the array.

**Parameters**

<i>in_index</i>	The position of the element to access.
-----------------	--

**Exceptions**

<i>std::out_of_range</i>	If <i>in_index</i> is greater than or equal to the value returned by <a href="#">getSize()</a> .
--------------------------	--

**Returns**

The value of the member with the specified name, if it exists; empty JSON value otherwise.

**4.13.3.13 parse()** [1/2]

```
Error rstudio::launcher_plugins::json::Array::parse (
    const char * in_jsonStr ) [override], [virtual]
```

Parses the JSON string into this array.

**Parameters**

<i>in_jsonStr</i>	The JSON string to parse.
-------------------	---------------------------

**Returns**

**Success** on successful parse when the resulting JSON value is a JSON [Array](#); error otherwise (e.g. Parse↵  
Error).

Reimplemented from [rstudio::launcher\\_plugins::json::Value](#).

**4.13.3.14 parse()** [2/2]

```
Error rstudio::launcher_plugins::json::Array::parse (
    const std::string & in_jsonStr ) [override], [virtual]
```

Parses the JSON string into this array.

**Parameters**

<i>in_jsonStr</i>	The JSON string to parse.
-------------------	---------------------------

**Returns**

**Success** on successful parse when the resulting JSON value is a JSON [Array](#); error otherwise (e.g. Parse↵  
Error).

Reimplemented from [rstudio::launcher\\_plugins::json::Value](#).

**4.13.3.15 push\_back()** [1/12]

```
void rstudio::launcher_plugins::json::Array::push_back (
    bool in_value )
```

Pushes the value onto the end of the JSON array.

**MAINTENANCE NOTE:** This method must be named in the STL style to work with STL functions and types such as `std::back_inserter`.

## Parameters

<i>in_value</i>	The value to push onto the end of the JSON array.
-----------------	---

**4.13.3.16 push\_back()** [2/12]

```
void rstudio::launcher_plugins::json::Array::push_back (
    const Array & in_value )
```

Pushes the value onto the end of the JSON array.

MAINTENANCE NOTE: This method must be named in the STL style to work with STL functions and types such as `std::back_inserter`.

## Parameters

<i>in_value</i>	The value to push onto the end of the JSON array.
-----------------	---

**4.13.3.17 push\_back()** [3/12]

```
void rstudio::launcher_plugins::json::Array::push_back (
    const char * in_value )
```

Pushes the value onto the end of the JSON array.

MAINTENANCE NOTE: This method must be named in the STL style to work with STL functions and types such as `std::back_inserter`.

## Parameters

<i>in_value</i>	The value to push onto the end of the JSON array.
-----------------	---

**4.13.3.18 push\_back()** [4/12]

```
void rstudio::launcher_plugins::json::Array::push_back (
    const Object & in_value )
```

Pushes the value onto the end of the JSON array.

MAINTENANCE NOTE: This method must be named in the STL style to work with STL functions and types such as `std::back_inserter`.

## Parameters

<i>in_value</i>	The value to push onto the end of the JSON array.
-----------------	---

**4.13.3.19 push\_back()** [5/12]

```
void rstudio::launcher_plugins::json::Array::push_back (
    const std::string & in_value )
```

Pushes the value onto the end of the JSON array.

MAINTENANCE NOTE: This method must be named in the STL style to work with STL functions and types such as `std::back_inserter`.

## Parameters

<i>in_value</i>	The value to push onto the end of the JSON array.
-----------------	---

**4.13.3.20 push\_back()** [6/12]

```
void rstudio::launcher_plugins::json::Array::push_back (
    const Value & in_value )
```

Pushes the value onto the end of the JSON array.

MAINTENANCE NOTE: This method must be named in the STL style to work with STL functions and types such as `std::back_inserter`.

## Parameters

<i>in_value</i>	The value to push onto the end of the JSON array.
-----------------	---

**4.13.3.21 push\_back()** [7/12]

```
void rstudio::launcher_plugins::json::Array::push_back (
    double in_value )
```

Pushes the value onto the end of the JSON array.

MAINTENANCE NOTE: This method must be named in the STL style to work with STL functions and types such as `std::back_inserter`.

## Parameters

<i>in_value</i>	The value to push onto the end of the JSON array.
-----------------	---

**4.13.3.22 push\_back()** [8/12]

```
void rstudio::launcher_plugins::json::Array::push_back (
    float in_value )
```

Pushes the value onto the end of the JSON array.

MAINTENANCE NOTE: This method must be named in the STL style to work with STL functions and types such as `std::back_inserter`.

## Parameters

<i>in_value</i>	The value to push onto the end of the JSON array.
-----------------	---

**4.13.3.23 push\_back()** [9/12]

```
void rstudio::launcher_plugins::json::Array::push_back (
    int in_value )
```

Pushes the value onto the end of the JSON array.

MAINTENANCE NOTE: This method must be named in the STL style to work with STL functions and types such as `std::back_inserter`.

## Parameters

<i>in_value</i>	The value to push onto the end of the JSON array.
-----------------	---

**4.13.3.24 push\_back()** [10/12]

```
void rstudio::launcher_plugins::json::Array::push_back (
    int64_t in_value )
```

Pushes the value onto the end of the JSON array.

MAINTENANCE NOTE: This method must be named in the STL style to work with STL functions and types such as `std::back_inserter`.



## Parameters

<i>in_value</i>	The value to push onto the end of the JSON array.
-----------------	---

**4.13.3.25 push\_back()** [11/12]

```
void rstudio::launcher_plugins::json::Array::push_back (
    uint64_t in_value )
```

Pushes the value onto the end of the JSON array.

MAINTENANCE NOTE: This method must be named in the STL style to work with STL functions and types such as `std::back_inserter`.

## Parameters

<i>in_value</i>	The value to push onto the end of the JSON array.
-----------------	---

**4.13.3.26 push\_back()** [12/12]

```
void rstudio::launcher_plugins::json::Array::push_back (
    unsigned int in_value )
```

Pushes the value onto the end of the JSON array.

MAINTENANCE NOTE: This method must be named in the STL style to work with STL functions and types such as `std::back_inserter`.

## Parameters

<i>in_value</i>	The value to push onto the end of the JSON array.
-----------------	---

**4.13.3.27 rbegin()**

```
ReverseIterator rstudio::launcher_plugins::json::Array::rbegin ( ) const
```

Gets an iterator pointing to the last member of this array, which iterates in the reverse direction.

## Returns

A reverse iterator pointing to the last member of this array.

#### 4.13.3.28 `rend()`

```
ReverseIterator rstudio::launcher_plugins::json::Array::rend ( ) const
```

Gets an iterator before the first member of this array, which can be compared with an other Reverseliterator to determine when reverse iteration has ended.

##### Returns

An iterator before the first member of this array.

#### 4.13.3.29 `toSetString()`

```
bool rstudio::launcher_plugins::json::Array::toSetString (
    std::set< std::string > & out_set ) const
```

Converts this JSON array to a set of strings.

##### Parameters

<code>out_set</code>	The set of strings.
----------------------	---------------------

##### Returns

True if this array could be converted to a set of strings; false otherwise.

#### 4.13.3.30 `toStringPairList()`

```
StringPairList rstudio::launcher_plugins::json::Array::toStringPairList ( ) const
```

Converts this array into a vector of string pairs. Elements of the form "key=value" will be parsed into the pair <"key", "value">. Elements which are not in that format (e.g. "value") will be parsed as <"value", "">. Any elements of the array which are not strings will be skipped.

##### Returns

The string elements of this JSON array as key value pairs.

#### 4.13.3.31 `toVectorInt()`

```
bool rstudio::launcher_plugins::json::Array::toVectorInt (
    std::vector< int > & out_set ) const
```

Converts this JSON array to a vector of integers.

## Parameters

<i>out_set</i>	The vector of integers.
----------------	-------------------------

## Returns

True if this array could be converted to a vector of integers; false otherwise.

## 4.13.3.32 toVectorString()

```
bool rstudio::launcher_plugins::json::Array::toVectorString (
    std::vector< std::string > & out_set ) const
```

Converts this JSON array to a vector of strings.

## Parameters

<i>out_set</i>	The vector of strings.
----------------	------------------------

## Returns

True if this array could be converted to a vector of strings; false otherwise.

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

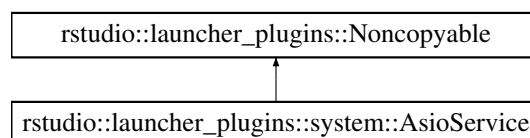
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/json/[Json.hpp](#)

## 4.14 rstudio::launcher\_plugins::system::AsioService Class Reference

Async input/output class which may be used to manage ASIO operations.

```
#include <Asio.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::system::AsioService:



## Static Public Member Functions

- static void `post` (const AsioFunction &in\_work)  
*Posts a job to be completed by this ASIO Service.*
- static void `setSignalHandler` (const OnSignal &in\_onSignal)  
*Sets the signal handler on the ASIO service.*
- static void `startThreads` (size\_t in\_numThreads)  
*Creates and adds the specified number of worker threads to the ASIO service.*
- static void `stop` ()  
*Stops the ASIO Service.*
- static void `waitForExit` ()

## Additional Inherited Members

### 4.14.1 Detailed Description

Async input/output class which may be used to manage ASIO operations.

### 4.14.2 Member Function Documentation

#### 4.14.2.1 `post()`

```
static void rstudio::launcher_plugins::system::AsioService::post (
    const AsioFunction & in_work ) [static]
```

Posts a job to be completed by this ASIO Service.

##### Parameters

<code>in_work</code>	The job to be posted to the ASIO Service.
----------------------	---

#### 4.14.2.2 `setSignalHandler()`

```
static void rstudio::launcher_plugins::system::AsioService::setSignalHandler (
    const OnSignal & in_onSignal ) [static]
```

Sets the signal handler on the ASIO service.

The ASIO service will manage signals sent to the process. The signal handler provided here will be invoked when a signal is received.

## Parameters

<code>in_onSignal</code>	The function to invoke when a signal is received.
--------------------------	---

**4.14.2.3 startThreads()**

```
static void rstudio::launcher_plugins::system::AsioService::startThreads (
    size_t in_numThreads ) [static]
```

Creates and adds the specified number of worker threads to the ASIO service.

## Parameters

<code>in_numThreads</code>	The number of worker threads to add to the ASIO service.
----------------------------	--

**4.14.2.4 stop()**

```
static void rstudio::launcher_plugins::system::AsioService::stop ( ) [static]
```

Stops the ASIO Service.

Calling this function will stop all async IO operations for the whole Plugin. IMPORTANT: Only call this function from the main thread immediately prior to shutdown.

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/Asio.hpp

**4.15 rstudio::launcher\_plugins::system::AsioStream Class Reference**

Class which allows reading from or writing to streams asynchronously.

```
#include <Asio.hpp>
```

**Public Member Functions**

- [AsioStream](#) (int in\_streamHandle)  
*Constructor.*
- [~AsioStream](#) () noexcept  
*Destructor. Closes the stream.*
- void [close](#) () noexcept  
*Closes the stream. Nothing may be read from or written to the stream after this is called.*
- void [readBytes](#) (const OnReadBytes &in\_onReadBytes, const [OnError](#) &in\_onError)  
*Attempts to read bytes from this ASIO stream.*
- void [writeBytes](#) (const std::string &in\_data, const [OnError](#) &in\_onError, const AsioFunction &in\_onFinished)  
*Writing=AsioFunction()*  
*Writes the provided data to the stream asynchronously.*

### 4.15.1 Detailed Description

Class which allows reading from or writing to streams asynchronously.

### 4.15.2 Constructor & Destructor Documentation

#### 4.15.2.1 AsioStream()

```
rstudio::launcher_plugins::system::AsioStream::AsioStream (
    int in_streamHandle ) [explicit]
```

Constructor.

##### Parameters

<i>in_streamHandle</i>	The handle of the stream for which to create this ASIO stream descriptor.
------------------------	---

### 4.15.3 Member Function Documentation

#### 4.15.3.1 readBytes()

```
void rstudio::launcher_plugins::system::AsioStream::readBytes (
    const OnReadBytes & in_onReadBytes,
    const OnError & in_onError )
```

Attempts to read bytes from this ASIO stream.

##### Parameters

<i>in_onReadBytes</i>	Callback function which will be invoked on successful read.
<i>in_onError</i>	Callback function which will be invoked if an error occurs.

#### 4.15.3.2 writeBytes()

```
void rstudio::launcher_plugins::system::AsioStream::writeBytes (
    const std::string & in_data,
    const OnError & in_onError,
    const AsioFunction & in_onFinishedWriting = AsioFunction() )
```

Writes the provided data to the stream asynchronously.

This method is thread safe. Each provided block of data will be written to the stream in full before a new one begins.

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/Asio.hpp

## 4.16 rstudio::launcher\_plugins::system::AsyncDeadlineEvent Class Reference

Class which may be used to post async work to be performed at a later time.

```
#include <Asio.hpp>
```

### Public Member Functions

- [AsyncDeadlineEvent](#) (const AsioFunction &in\_work, const [DateTime](#) &in\_deadlineTime)  
*Constructor.*
- [AsyncDeadlineEvent](#) (const AsioFunction &in\_work, const [TimeDuration](#) &in\_waitTime)  
*Constructor.*
- [~AsyncDeadlineEvent](#) ()  
*Destructor. The event will be canceled if this invoked before the deadline time.*
- void [cancel](#) ()  
*Cancels the event, if invoked before the deadline time.*
- void [start](#) ()  
*Starts waiting for the event.*

### 4.16.1 Detailed Description

Class which may be used to post async work to be performed at a later time.

### 4.16.2 Constructor & Destructor Documentation

#### 4.16.2.1 AsyncDeadlineEvent() [1/2]

```
rstudio::launcher_plugins::system::AsyncDeadlineEvent::AsyncDeadlineEvent (
    const AsioFunction & in_work,
    const DateTime & in_deadlineTime )
```

Constructor.

## Parameters

<i>in_work</i>	The work to be performed when the deadline time is reached.
<i>in_deadlineTime</i>	The time at which the work should be performed.

**4.16.2.2 AsyncDeadlineEvent() [2/2]**

```
rstudio::launcher_plugins::system::AsyncDeadlineEvent::AsyncDeadlineEvent (
    const AsioFunction & in_work,
    const TimeDuration & in_waitTime )
```

Constructor.

## Parameters

<i>in_work</i>	The work to be performed when the deadline time is reached.
<i>in_waitTime</i>	The amount of time, from the invocation time, that should pass before the work is performed.

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/Asio.hpp

## 4.17 rstudio::launcher\_plugins::system::process::AsyncProcess↩ Callbacks Struct Reference

Callbacks that will be invoked when certain events happen in the asynchronous child process.

```
#include <Process.hpp>
```

**Public Attributes**

- OnErrorCallback [OnError](#)  
*Callback invoked if the asynchronous child process encounters an error.*
- OnExitCallback [OnExit](#)  
*Callback invoked when the asynchronous child process exits.*
- OnOutputCallback [OnStandardError](#)  
*Callback invoked when the asynchronous child process writes to standard error.*
- OnOutputCallback [OnStandardOutput](#)  
*Callback invoked when the asynchronous child process writes to standard out.*



### 4.17.1 Detailed Description

Callbacks that will be invoked when certain events happen in the asynchronous child process.

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/Process.hpp

## 4.18 rstudio::launcher\_plugins::system::AsyncTimedEvent Class Reference

Class which performs an action asynchronously every specified number of seconds.

```
#include <Asio.hpp>
```

### Public Member Functions

- [AsyncTimedEvent](#) ()  
*Default constructor.*
- void [start](#) (const [TimeDuration](#) &in\_timeDuration, const AsioFunction &in\_event)  
*Starts performing the specified event every in\_timeDuration.*
- void [cancel](#) ()  
*Cancels the timed event.*
- void [reportError](#) (const [Error](#) &in\_error)  
*Reports a fatal error to the [AsyncTimedEvent](#) instance. Invoking this will cause the timed event to stop running.*

### 4.18.1 Detailed Description

Class which performs an action asynchronously every specified number of seconds.

### 4.18.2 Member Function Documentation

#### 4.18.2.1 reportError()

```
void rstudio::launcher_plugins::system::AsyncTimedEvent::reportError (  
    const Error & in_error )
```

Reports a fatal error to the [AsyncTimedEvent](#) instance. Invoking this will cause the timed event to stop running.

This method should be invoked by the event passed to start if a fatal error occurs.

## Parameters

<code>in_error</code>	The error which occurred.
-----------------------	---------------------------

**4.18.2.2 start()**

```
void rstudio::launcher_plugins::system::AsyncTimedEvent::start (
    const TimeDuration & in_timeDuration,
    const AsioFunction & in_event )
```

Starts performing the specified event every `in_timeDuration`.

This function may only be called once per instance. Restarting a canceled or otherwise stopped timed event will not work. Instead, a new instance should be created and started.

## Parameters

<code>in_timeDuration</code>	The amount of time to wait between each event.
<code>in_event</code>	The action to perform every <code>in_timeDuration</code> .

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

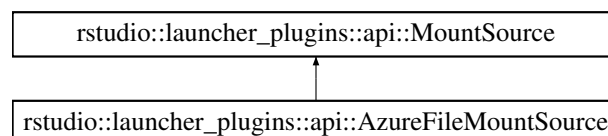
- `/workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/Asio.hpp`

**4.19 rstudio::launcher\_plugins::api::AzureFileMountSource Struct Reference**

Represents an Azure File [Mount](#) Source.

```
#include <Job.hpp>
```

Inheritance diagram for `rstudio::launcher_plugins::api::AzureFileMountSource`:

**Public Member Functions**

- `std::string getSecretName () const`  
*Gets the name of the Azure Secret used to connect to the Azure File [Mount](#) Source.*
- `std::string getShareName () const`  
*Gets the name of the share in Azure to be mounted.*

## Static Public Member Functions

- static [Error fromJson](#) (const [json::Object](#) &in\_json, [AzureFileMountSource](#) &out\_mountSource)  
*Constructs an AzureMountSource from a JSON object which represents the mount source.*

## Friends

- class [MountSource](#)

## Additional Inherited Members

### 4.19.1 Detailed Description

Represents an Azure File [Mount](#) Source.

### 4.19.2 Member Function Documentation

#### 4.19.2.1 fromJson()

```
static Error rstudio::launcher_plugins::api::AzureFileMountSource::fromJson (  
    const json::Object & in_json,  
    AzureFileMountSource & out_mountSource ) [static]
```

Constructs an AzureMountSource from a JSON object which represents the mount source.

#### Parameters

<i>in_json</i>	The JSON object which represents the mount source.
<i>out_mountSource</i>	The populated mount source value. Not valid if an error is returned.

#### Returns

[Success](#) if in\_json could be parsed as an AzureMountSource; [Error](#) otherwise.

#### 4.19.2.2 getSecretName()

```
std::string rstudio::launcher_plugins::api::AzureFileMountSource::getSecretName ( ) const
```

Gets the name of the Azure Secret used to connect to the Azure File [Mount](#) Source.

**Exceptions**

<code>std::logic_error</code>	if the 'secretName' field cannot be found.
-------------------------------	--

**Returns**

The name of the Azure Secret used to connect to the Azure File [Mount](#) Source.

**4.19.2.3 getShareName()**

```
std::string rstudio::launcher_plugins::api::AzureFileMountSource::getShareName ( ) const
```

Gets the name of the share in Azure to be mounted.

**Exceptions**

<code>std::logic_error</code>	if the 'shareName' field cannot be found.
-------------------------------	---

**Returns**

The name of the share in Azure to be mounted.

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

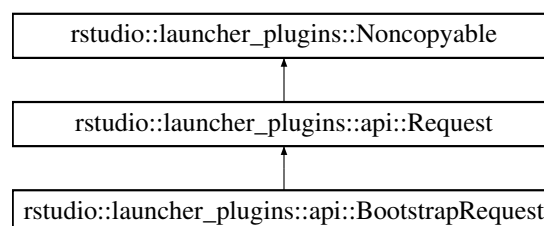
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Job.hpp

**4.20 rstudio::launcher\_plugins::api::BootstrapRequest Class Reference**

Represents a bootstrap request received from the Launcher.

```
#include <Request.hpp>
```

Inheritance diagram for `rstudio::launcher_plugins::api::BootstrapRequest`:



## Public Member Functions

- int [getMajorVersion](#) () const  
*Gets the major version of the RStudio Launcher that sent this bootstrap request.*
- int [getMinorVersion](#) () const  
*Gets the minor version of the RStudio Launcher that sent this bootstrap request.*
- int [getPatchNumber](#) () const  
*Gets the patch number of the RStudio Launcher that sent this bootstrap request.*

## Friends

- class **Request**

## Additional Inherited Members

### 4.20.1 Detailed Description

Represents a bootstrap request received from the Launcher.

### 4.20.2 Member Function Documentation

#### 4.20.2.1 [getMajorVersion\(\)](#)

```
int rstudio::launcher_plugins::api::BootstrapRequest::getMajorVersion ( ) const
```

Gets the major version of the RStudio Launcher that sent this bootstrap request.

#### Returns

The major version of the RStudio Launcher.

#### 4.20.2.2 [getMinorVersion\(\)](#)

```
int rstudio::launcher_plugins::api::BootstrapRequest::getMinorVersion ( ) const
```

Gets the minor version of the RStudio Launcher that sent this bootstrap request.

#### Returns

The minor version of the RStudio Launcher.

#### 4.20.2.3 getPatchNumber()

```
int rstudio::launcher_plugins::api::BootstrapRequest::getPatchNumber ( ) const
```

Gets the patch number of the RStudio Launcher that sent this bootstrap request.

##### Returns

The patch number of the RStudio Launcher.

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

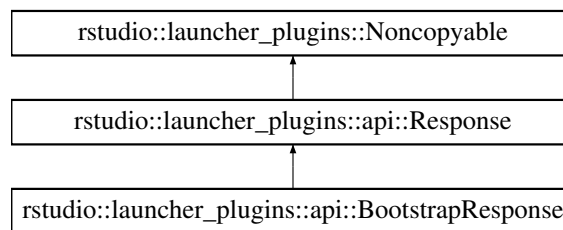
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Request.hpp

## 4.21 rstudio::launcher\_plugins::api::BootstrapResponse Class Reference

Class which represents a bootstrap response which can be sent to the Launcher in response to a bootstrap request.

```
#include <Response.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::BootstrapResponse:



### Public Member Functions

- [BootstrapResponse](#) (uint64\_t in\_requestId)  
*Constructor.*
- [json::Object toJson](#) () const override  
*Converts this bootstrap response to a JSON object.*

### Additional Inherited Members

#### 4.21.1 Detailed Description

Class which represents a bootstrap response which can be sent to the Launcher in response to a bootstrap request.

#### 4.21.2 Constructor & Destructor Documentation

##### 4.21.2.1 BootstrapResponse()

```
rstudio::launcher_plugins::api::BootstrapResponse::BootstrapResponse (
    uint64_t in_requestId ) [explicit]
```

Constructor.

## Parameters

<i>in_↵</i> <i>requestId</i>	The ID of the bootstrap request for which this bootstrap response is being sent.
---------------------------------	--

### 4.21.3 Member Function Documentation

#### 4.21.3.1 toJson()

```
json::Object rstudio::launcher_plugins::api::BootstrapResponse::toJson ( ) const [override],  
[virtual]
```

Converts this bootstrap response to a JSON object.

## Returns

The JSON object which represents this bootstrap response.

Reimplemented from [rstudio::launcher\\_plugins::api::Response](#).

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

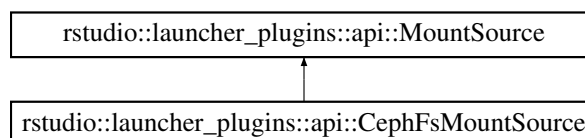
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Response.hpp

## 4.22 rstudio::launcher\_plugins::api::CephFsMountSource Struct Reference

Represents a Ceph File System [Mount](#) Source.

```
#include <Job.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::CephFsMountSource:



## Public Member Functions

- `std::vector< std::string > getMonitors () const`  
*Gets the list of Ceph monitor addresses.*
- `std::string getPath () const`  
*Gets the path to mount.*
- `std::string getUser () const`  
*Gets the user to mount the path as.*
- `std::string getSecretFile () const`  
*Gets the location of the file which contains the Ceph keyring for authentication.*
- `std::string getSecretRef () const`  
*Gets the reference to the Ceph authentication secrets which override the Secret File.*

## Static Public Member Functions

- static `Error fromJson (const json::Object &in_json, CephFsMountSource &out_mountSource)`  
*Constructs a [CephFsMountSource](#) from a JSON object which represents the mount source.*

## Friends

- class `MountSource`

## Additional Inherited Members

### 4.22.1 Detailed Description

Represents a Ceph File System [Mount](#) Source.

### 4.22.2 Member Function Documentation

#### 4.22.2.1 fromJson()

```
static Error rstudio::launcher_plugins::api::CephFsMountSource::fromJson (
    const json::Object & in_json,
    CephFsMountSource & out_mountSource ) [static]
```

Constructs a [CephFsMountSource](#) from a JSON object which represents the mount source.

#### Parameters

<i>in_json</i>	The JSON object which represents the mount source.
<i>out_mountSource</i>	The populated mount source value. Not valid if an error is returned.



**Returns**

[Success](#) if in\_json could be parsed as a [CephFsMountSource](#); [Error](#) otherwise.

**4.22.2.2 getMonitors()**

```
std::vector<std::string> rstudio::launcher_plugins::api::CephFsMountSource::getMonitors ( )  
const
```

Gets the list of Ceph monitor addresses.

**Exceptions**

<code>std::logic_error</code>	if the 'monitors' field cannot be found.
-------------------------------	--

**Returns**

The list of Ceph monitor addresses.

**4.22.2.3 getPath()**

```
std::string rstudio::launcher_plugins::api::CephFsMountSource::getPath ( ) const
```

Gets the path to mount.

**Returns**

The path to mount.

**4.22.2.4 getSecretFile()**

```
std::string rstudio::launcher_plugins::api::CephFsMountSource::getSecretFile ( ) const
```

Gets the location of the file which contains the Ceph keyring for authentication.

**Returns**

The location of the file which contains the Ceph keyring for authentication.

#### 4.22.2.5 getSecretRef()

```
std::string rstudio::launcher_plugins::api::CephFsMountSource::getSecretRef ( ) const
```

Gets the reference to the Ceph authentication secrets which override the Secret File.

##### Returns

The reference to the Ceph authentication secrets which override the Secret File.

#### 4.22.2.6 getUser()

```
std::string rstudio::launcher_plugins::api::CephFsMountSource::getUser ( ) const
```

Gets the user to mount the path as.

##### Returns

The user to mount the path as.

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

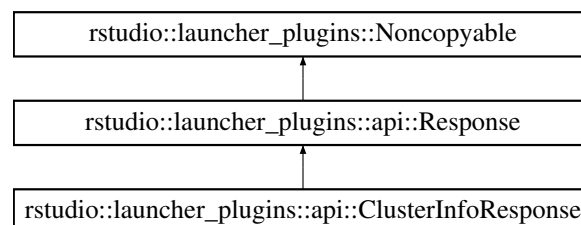
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Job.hpp

## 4.23 rstudio::launcher\_plugins::api::ClusterInfoResponse Class Reference

Class which represents a cluster info response which should be sent to the Launcher in response to a cluster info request.

```
#include <Response.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::ClusterInfoResponse:



### Public Member Functions

- [ClusterInfoResponse](#) (uint64\_t in\_requestId, const [JobSourceConfiguration](#) &in\_configuration)  
*Constructor.*
- [json::Object toJson](#) ( ) const override  
*Converts this cluster info response to a JSON object.*

## Additional Inherited Members

### 4.23.1 Detailed Description

Class which represents a cluster info response which should be sent to the Launcher in response to a cluster info request.

### 4.23.2 Constructor & Destructor Documentation

#### 4.23.2.1 ClusterInfoResponse()

```
rstudio::launcher_plugins::api::ClusterInfoResponse::ClusterInfoResponse (
    uint64_t in_requestId,
    const JobSourceConfiguration & in_configuration )
```

Constructor.

##### Parameters

<i>in_requestId</i>	The ID of the request for which this response is being sent.
<i>in_configuration</i>	The configuration and capabilities of the cluster.

### 4.23.3 Member Function Documentation

#### 4.23.3.1 toJson()

```
json::Object rstudio::launcher_plugins::api::ClusterInfoResponse::toJson ( ) const [override],
[virtual]
```

Converts this cluster info response to a JSON object.

##### Returns

The JSON object which represents this cluster info response.

Reimplemented from [rstudio::launcher\\_plugins::api::Response](#).

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Response.hpp

## 4.24 rstudio::launcher\_plugins::api::Container Struct Reference

Struct which represents the container to use when launching a containerized job.

```
#include <Job.hpp>
```

### Public Member Functions

- [json::Object toJson](#) () const  
*Converts this [Container](#) to a JSON object which represents it.*

### Static Public Member Functions

- static [Error fromJson](#) (const [json::Object](#) &in\_json, [Container](#) &out\_container)  
*Constructs a [Container](#) from a JSON object which represents the container.*

### Public Attributes

- std::string [Image](#)
- [Optional](#)< int > [RunAsUserId](#)
- [Optional](#)< int > [RunAsGroupId](#)
- std::vector< int > [SupplementalGroupIds](#)

#### 4.24.1 Detailed Description

Struct which represents the container to use when launching a containerized job.

#### 4.24.2 Member Function Documentation

##### 4.24.2.1 fromJson()

```
static Error rstudio::launcher_plugins::api::Container::fromJson (
    const json::Object & in_json,
    Container & out_container ) [static]
```

Constructs a [Container](#) from a JSON object which represents the container.

##### Parameters

<i>in_json</i>	The JSON object which represents the container.
<i>out_container</i>	The populated container value. Not valid if an error is returned.

#### Returns

[Success](#) if in\_json could be parsed as a [Container](#); [Error](#) otherwise.

#### 4.24.2.2 toJson()

```
json::Object rstudio::launcher_plugins::api::Container::toJson ( ) const
```

Converts this [Container](#) to a JSON object which represents it.

#### Returns

The JSON object which represents this [Container](#).

### 4.24.3 Member Data Documentation

#### 4.24.3.1 Image

```
std::string rstudio::launcher_plugins::api::Container::Image
```

The name of the image to use.

#### 4.24.3.2 RunAsGroupId

```
Optional<int> rstudio::launcher_plugins::api::Container::RunAsGroupId
```

The optional group ID to run the container as.

#### 4.24.3.3 RunAsUserId

```
Optional<int> rstudio::launcher_plugins::api::Container::RunAsUserId
```

The optional user ID to run the container as.

#### 4.24.3.4 SupplementalGroupIds

```
std::vector<int> rstudio::launcher_plugins::api::Container::SupplementalGroupIds
```

The optional set of supplemental group IDs for the run-as user, to pass to the container on launch.

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Job.hpp

## 4.25 rstudio::launcher\_plugins::api::ContainerConfiguration Struct Reference

Describes the container configuration of the [Job](#) Source.

```
#include <IJobSource.hpp>
```

### Public Member Functions

- [ContainerConfiguration](#) ()  
*Default constructor.*

### Public Attributes

- bool [AllowUnknownImages](#)
- std::set< std::string > [ContainerImages](#)
- std::string [DefaultImage](#)
- bool [SupportsContainers](#)

### 4.25.1 Detailed Description

Describes the container configuration of the [Job](#) Source.

### 4.25.2 Member Data Documentation

#### 4.25.2.1 AllowUnknownImages

```
bool rstudio::launcher_plugins::api::ContainerConfiguration::AllowUnknownImages
```

Whether users may select unknown images when launching a job.

#### 4.25.2.2 ContainerImages

```
std::set<std::string> rstudio::launcher_plugins::api::ContainerConfiguration::ContainerImages
```

The list of known images.

#### 4.25.2.3 DefaultImage

```
std::string rstudio::launcher_plugins::api::ContainerConfiguration::DefaultImage
```

The default image.

#### 4.25.2.4 SupportsContainers

```
bool rstudio::launcher_plugins::api::ContainerConfiguration::SupportsContainers
```

Whether this [Job](#) Source supports containers. Default: false.

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

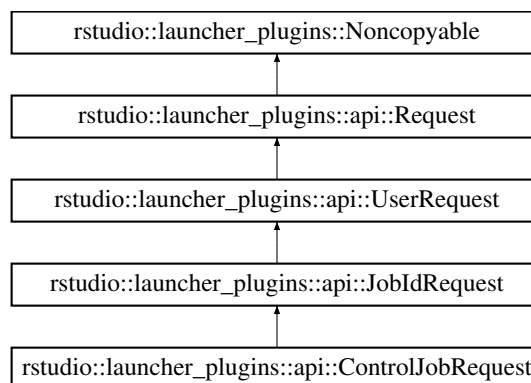
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/IJobSource.hpp

## 4.26 rstudio::launcher\_plugins::api::ControlJobRequest Class Reference

[Request](#) from the launcher to control the state of a [Job](#).

```
#include <Request.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::ControlJobRequest:



### Public Types

- enum [Operation](#) {  
[Operation::FIRST](#) = 0, **SUSPEND** = 0, [Operation::RESUME](#) = 1, [Operation::STOP](#) = 2,  
[Operation::KILL](#) = 3, [Operation::CANCEL](#) = 4, [Operation::INVALID](#) }

### Public Member Functions

- [Operation](#) [getOperation](#) () const  
*Gets the control job action which should be taken.*

### Friends

- class **Request**

## Additional Inherited Members

### 4.26.1 Detailed Description

[Request](#) from the launcher to control the state of a [Job](#).

### 4.26.2 Member Enumeration Documentation

#### 4.26.2.1 Operation

```
enum rstudio::launcher_plugins::api::ControlJobRequest::Operation [strong]
```

Enumerator

FIRST	Indicates that the job should be suspended. This operation should be equivalent to sending SIGSTOP.
RESUME	Indicates that the job should be resumed. This operation should be equivalent to sending SIGCONT.
STOP	Indicates that the job should be stopped. This operation should be equivalent to sending SIGTERM.
KILL	Indicates that the job should be killed. This operation should be equivalent to sending SIGKILL.
CANCEL	Indicates that a pending job should be canceled, if possible.
INVALID	This value must always be last for input validation purposes.

### 4.26.3 Member Function Documentation

#### 4.26.3.1 getOperation()

```
Operation rstudio::launcher_plugins::api::ControlJobRequest::getOperation ( ) const
```

Gets the control job action which should be taken.

Returns

The control job action which should be taken.

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Request.hpp

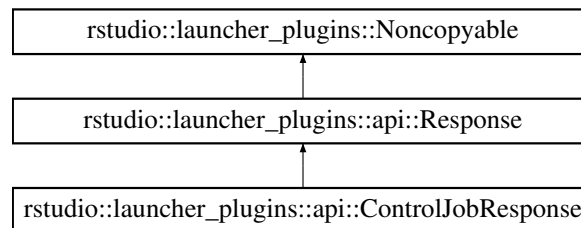


## 4.27 rstudio::launcher\_plugins::api::ControlJobResponse Class Reference

Class which represents the result of a control job operation.

```
#include <Response.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::ControlJobResponse:



### Public Member Functions

- [ControlJobResponse](#) (uint64\_t in\_requestId, std::string in\_statusMessage, bool in\_isComplete)  
*Constructor.*
- [json::Object toJson](#) () const override  
*Converts this control job response to a JSON object.*

### Additional Inherited Members

#### 4.27.1 Detailed Description

Class which represents the result of a control job operation.

#### 4.27.2 Constructor & Destructor Documentation

##### 4.27.2.1 ControlJobResponse()

```
rstudio::launcher_plugins::api::ControlJobResponse::ControlJobResponse (
    uint64_t in_requestId,
    std::string in_statusMessage,
    bool in_isComplete )
```

Constructor.

##### Parameters

<i>in_requestId</i>	The ID of the request for which this response is being sent.
<i>in_statusMessage</i>	A message describing the status of the control job operation that was requested.
<i>in_isComplete</i>	Whether the request control job operation has completed (true) or not (false).

Generated by Doxygen

### 4.27.3 Member Function Documentation

#### 4.27.3.1 toJson()

```
json::Object rstudio::launcher_plugins::api::ControlJobResponse::toJson ( ) const [override],
[virtual]
```

Converts this control job response to a JSON object.

#### Returns

The JSON object which represents this control job response.

Reimplemented from [rstudio::launcher\\_plugins::api::Response](#).

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Response.hpp

## 4.28 rstudio::launcher\_plugins::system::DateTime Class Reference

Class which represents a date and time in UTC.

```
#include <DateTime.hpp>
```

### Public Member Functions

- [DateTime](#) ()  
*Constructor.*
- [DateTime](#) (const [DateTime](#) &in\_other)  
*Copy constructor.*
- [DateTime](#) ([DateTime](#) &&in\_other) noexcept  
*Move constructor.*
- [DateTime](#) & operator= (const [DateTime](#) &in\_other)  
*Assignment operator.*
- [DateTime](#) & operator= ([DateTime](#) &&in\_other) noexcept  
*Move operator.*
- [TimeDuration](#) operator- (const [DateTime](#) &in\_other) const  
*Subtracts two DateTime's to produce an TimeDuration.*
- [DateTime](#) operator- (const [TimeDuration](#) &in\_intervalTime) const  
*Subtracts the given TimeDuration from a copy of this DateTime.*
- [DateTime](#) & operator-= (const [TimeDuration](#) &in\_intervalTime)  
*Subtracts the given TimeDuration from this DateTime.*
- [DateTime](#) operator+ (const [TimeDuration](#) &in\_intervalTime) const  
*Adds the given TimeDuration to a copy of this DateTime.*
- [DateTime](#) & operator+= (const [TimeDuration](#) &in\_intervalTime)

- Adds the given [TimeDuration](#) to this [DateTime](#).*
- bool [operator==](#) (const [DateTime](#) &in\_other) const  
*Equality operator.*
- bool [operator!=](#) (const [DateTime](#) &in\_other) const  
*Inequality operator.*
- bool [operator<](#) (const [DateTime](#) &in\_other) const  
*Less than operator.*
- bool [operator<=](#) (const [DateTime](#) &in\_other) const  
*Less than operator.*
- bool [operator>](#) (const [DateTime](#) &in\_other) const  
*Less than operator.*
- bool [operator>=](#) (const [DateTime](#) &in\_other) const  
*Less than operator.*
- std::string [toString](#) () const  
*Converts this [DateTime](#) to an ISO 8601 time string.*
- std::string [toString](#) (const char \*in\_format) const  
*Converts this [DateTime](#) to a string representation defined by the provided format.*
- std::string [toString](#) (const std::string &in\_format) const  
*Converts this [DateTime](#) to a string representation defined by the provided format.*

## Static Public Member Functions

- static [Error fromString](#) (const std::string &in\_timeStr, [DateTime](#) &out\_dateTime)  
*Constructs a [DateTime](#) from an ISO 8601 string representation. The string must be in UTC time.*

### 4.28.1 Detailed Description

Class which represents a date and time in UTC.

### 4.28.2 Constructor & Destructor Documentation

#### 4.28.2.1 [DateTime\(\)](#) [1/3]

```
rstudio::launcher_plugins::system::DateTime::DateTime ( )
```

Constructor.

Creates a date time which represents the time at which it was created.

#### 4.28.2.2 [DateTime\(\)](#) [2/3]

```
rstudio::launcher_plugins::system::DateTime::DateTime (
    const DateTime & in_other )
```

Copy constructor.

## Parameters

<i>in_other</i>	The <a href="#">DateTime</a> to copied into this.
-----------------	---

**4.28.2.3 DateTime() [3/3]**

```
rstudio::launcher_plugins::system::DateTime::DateTime (
    DateTime && in_other ) [noexcept]
```

Move constructor.

## Parameters

<i>in_other</i>	The <a href="#">DateTime</a> to moved into this.
-----------------	--

**4.28.3 Member Function Documentation****4.28.3.1 fromString()**

```
static Error rstudio::launcher_plugins::system::DateTime::fromString (
    const std::string & in_timeStr,
    DateTime & out_dateTime ) [static]
```

Constructs a [DateTime](#) from an ISO 8601 string representation. The string must be in UTC time.

Valid format: "%Y-%m-%dT%H:%M:%S%F%ZP" e.g. "2020-03-05T14:33:15.008765Z" e.g. "1995-10-31T02:06:22+8:00" (fractional seconds are 0) e.g. "1988-12-25T23:23:23.054321MST-06" e.g. "1972-04-18T00:01:51PS-T-08PDT+01,M4.1.0/02:00,M10.5.0/02:00" (Full Posix Time Zone String)

## Parameters

<i>in_timeStr</i>	The string representation of the <a href="#">DateTime</a> to construct.
<i>out_dateTime</i>	The newly constructed <a href="#">DateTime</a> , if no error occurs.

## Returns

[Success](#) if *in\_timeStr* is a valid ISO 8601 representation of a date and time; [Error](#) otherwise.

**4.28.3.2 operator"!=(**

```
bool rstudio::launcher_plugins::system::DateTime::operator!= (
    const DateTime & in_other ) const
```

Inequality operator.

#### Parameters

<i>in_other</i>	The <a href="#">DateTime</a> to compare against this.
-----------------	---

#### Returns

True if this [DateTime](#) and *in\_other* represent the different times; false otherwise.

### 4.28.3.3 operator+()

```
DateTime rstudio::launcher_plugins::system::DateTime::operator+ (  
    const TimeDuration & in_intervalTime ) const
```

Adds the given [TimeDuration](#) to a copy of this [DateTime](#).

#### Parameters

<i>in_intervalTime</i>	The interval time to add to this <a href="#">DateTime</a> .
------------------------	---

#### Returns

The new [DateTime](#), which is this value of [DateTime](#) plus the specified [TimeDuration](#).

### 4.28.3.4 operator+=()

```
DateTime& rstudio::launcher_plugins::system::DateTime::operator+= (  
    const TimeDuration & in_intervalTime )
```

Adds the given [TimeDuration](#) to this [DateTime](#).

#### Parameters

<i>in_intervalTime</i>	The interval time to add to this <a href="#">DateTime</a> .
------------------------	---

#### Returns

A reference to this [DateTime](#).

**4.28.3.5 operator-() [1/2]**

```
TimeDuration rstudio::launcher_plugins::system::DateTime::operator- (
    const DateTime & in_other ) const
```

Subtracts two [DateTimes](#) to produce an [TimeDuration](#).

**Parameters**

<i>in_other</i>	The date time to subtract from this.
-----------------	--------------------------------------

**Returns**

An interval time representing the difference between this [DateTime](#) and *in\_other*.

**4.28.3.6 operator-() [2/2]**

```
DateTime rstudio::launcher_plugins::system::DateTime::operator- (
    const TimeDuration & in_intervalTime ) const
```

Subtracts the given [TimeDuration](#) from a copy of this [DateTime](#).

**Parameters**

<i>in_intervalTime</i>	The interval time to subtract from this <a href="#">DateTime</a> .
------------------------	--

**Returns**

A reference to this [DateTime](#).

**4.28.3.7 operator-=()**

```
DateTime& rstudio::launcher_plugins::system::DateTime::operator-= (
    const TimeDuration & in_intervalTime )
```

Subtracts the given [TimeDuration](#) from this [DateTime](#).

**Parameters**

<i>in_intervalTime</i>	The interval time to subtract from this <a href="#">DateTime</a> .
------------------------	--

**Returns**

The new [DateTime](#), which is this value of [DateTime](#) minus the specified [TimeDuration](#).

#### 4.28.3.8 operator<()

```
bool rstudio::launcher_plugins::system::DateTime::operator< (
    const DateTime & in_other ) const
```

Less than operator.

##### Parameters

<i>in_other</i>	The <a href="#">DateTime</a> to compare against this.
-----------------	---

##### Returns

True if this [DateTime](#) is an earlier time than *in\_other*; false otherwise.

#### 4.28.3.9 operator<=()

```
bool rstudio::launcher_plugins::system::DateTime::operator<= (
    const DateTime & in_other ) const
```

Less than operator.

##### Parameters

<i>in_other</i>	The <a href="#">DateTime</a> to compare against this.
-----------------	---

##### Returns

True if this [DateTime](#) is an earlier time or the same time as *in\_other*; false otherwise.

#### 4.28.3.10 operator=() [1/2]

```
DateTime& rstudio::launcher_plugins::system::DateTime::operator= (
    const DateTime & in_other )
```

Assignment operator.

##### Parameters

<i>in_other</i>	The <a href="#">DateTime</a> to assign to this.
-----------------	---

**Returns**

A reference to this [DateTime](#).

**4.28.3.11 operator=()** [2/2]

```
DateTime& rstudio::launcher_plugins::system::DateTime::operator= (
    DateTime && in_other ) [noexcept]
```

Move operator.

**Parameters**

<i>in_other</i>	The <a href="#">DateTime</a> to move to this.
-----------------	---

**Returns**

A reference to this [DateTime](#).

**4.28.3.12 operator==()**

```
bool rstudio::launcher_plugins::system::DateTime::operator== (
    const DateTime & in_other ) const
```

Equality operator.

**Parameters**

<i>in_other</i>	The <a href="#">DateTime</a> to compare against this.
-----------------	---

**Returns**

True if this [DateTime](#) and *in\_other* represent the exact same moment in time; false otherwise.

**4.28.3.13 operator>()**

```
bool rstudio::launcher_plugins::system::DateTime::operator> (
    const DateTime & in_other ) const
```

Less than operator.



## Parameters

<i>in_other</i>	The <a href="#">DateTime</a> to compare against this.
-----------------	---

## Returns

True if this [DateTime](#) is a later time than *in\_other*; false otherwise.

**4.28.3.14 operator>=()**

```
bool rstudio::launcher_plugins::system::DateTime::operator>= (
    const DateTime & in_other ) const
```

Less than operator.

## Parameters

<i>in_other</i>	The <a href="#">DateTime</a> to compare against this.
-----------------	---

## Returns

True if this [DateTime](#) is the same time or a later time than *in\_other*; false otherwise.

**4.28.3.15 toString() [1/3]**

```
std::string rstudio::launcher_plugins::system::DateTime::toString ( ) const
```

Converts this [DateTime](#) to an ISO 8601 time string.

## Returns

This [DateTime](#) as an ISO 8601 string representation.

**4.28.3.16 toString() [2/3]**

```
std::string rstudio::launcher_plugins::system::DateTime::toString (
    const char * in_format ) const
```

Converts this [DateTime](#) to a string representation defined by the provided format.

## Parameters

<i>in_format</i>	The time format string, as documented in the 'Date-Time Support' section of the 'Advanced Features' chapter of the RStudio Launcher Plugin SDK Developer's Guide.
------------------	---

## Returns

This [DateTime](#), as a string with the specified format.

**4.28.3.17 toString() [3/3]**

```
std::string rstudio::launcher_plugins::system::DateTime::toString (
    const std::string & in_format ) const
```

Converts this [DateTime](#) to a string representation defined by the provided format.

## Parameters

<i>in_format</i>	The time format string, as documented in the 'Date-Time Support' section of the 'Advanced Features' chapter of the RStudio Launcher Plugin SDK Developer's Guide.
------------------	---

## Returns

This [DateTime](#), as a string with the specified format.

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

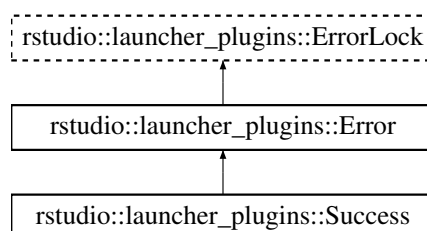
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/DateTime.hpp

**4.29 rstudio::launcher\_plugins::Error Class Reference**

Class which represents an error.

```
#include <Error.hpp>
```

Inheritance diagram for `rstudio::launcher_plugins::Error`:



## Public Member Functions

- [Error](#) ()=default  
*Default constructor.*
- [Error](#) (const [Error](#) &in\_other)  
*Copy constructor.*
- [Error](#) (std::string in\_name, int in\_code, const [ErrorLocation](#) &in\_location)  
*Constructor.*
- [Error](#) (std::string in\_name, int in\_code, const [Error](#) &in\_cause, const [ErrorLocation](#) &in\_location)  
*Constructor.*
- [Error](#) (std::string in\_name, int in\_code, std::string in\_message, const [ErrorLocation](#) &in\_location)  
*Constructor.*
- [Error](#) (std::string in\_name, int in\_code, std::string in\_message, const [Error](#) &in\_cause, const [ErrorLocation](#) &in\_location)  
*Constructor.*
- [~Error](#) () override=default  
*Non-virtual destructor because only [Success](#) inherits [Error](#) and it will keep [Error](#) lightweight.*
- [operator bool](#) () const  
*Overloaded operator bool to allow Errors to be treated as boolean values.*
- [operator!](#) () const  
*Overloaded operator ! to allow Errors to be treated as boolean values.*
- [operator==](#) (const [Error](#) &in\_other) const  
*Equality operator. Two errors are equal if their codes and names are the same.*
- [operator!=](#) (const [Error](#) &in\_other) const  
*Inequality operator. Two errors are equal if their codes and names are the same.*
- [void addOrUpdateProperty](#) (const std::string &in\_name, const std::string &in\_value)  
*Add or updates a property of this error. If any properties with the specified name exist, they will all be updated.*
- [void addOrUpdateProperty](#) (const std::string &in\_name, const [system::FilePath](#) &in\_value)  
*Add or updates a property of this error. If any properties with the specified name exist, they will all be updated.*
- [void addOrUpdateProperty](#) (const std::string &in\_name, int in\_value)  
*Add or updates a property of this error. If any properties with the specified name exist, they will all be updated.*
- [void addProperty](#) (const std::string &in\_name, const std::string &in\_value)  
*Adds a property of this error. If a property with the same name already exists, a duplicate will be added.*
- [void addProperty](#) (const std::string &in\_name, const [system::FilePath](#) &in\_value)  
*Adds a property of this error. If a property with the same name already exists, a duplicate will be added.*
- [void addProperty](#) (const std::string &in\_name, int in\_value)  
*Adds a property of this error. If a property with the same name already exists, a duplicate will be added.*
- [std::string asString](#) () const  
*Formats the error as a string.*
- [const Error & getCause](#) () const  
*Gets the error which caused this error.*
- [int getCode](#) () const  
*Gets the error code.*
- [const ErrorLocation & getLocation](#) () const  
*Gets the location where the error occurred.*
- [const std::string & getMessage](#) () const  
*Gets the error message.*
- [const std::string & getName](#) () const  
*Gets the name of the error.*
- [const ErrorProperties & getProperties](#) () const  
*Gets the custom properties of the error.*

- `std::string getProperty (const std::string &name) const`  
*Gets a custom property of this error.*
- `std::string getSummary () const`  
*Gets the cause of the error.*
- `bool isExpected () const`  
*Gets whether this error was expected or not.*
- `void setExpected ()`  
*Sets the property that indicates that this error was expected. Errors are unexpected by default; only unexpected errors will be logged. Expected errors can be marked as such to suppress logging of those errors.*

### 4.29.1 Detailed Description

Class which represents an error.

This class should not be derived from since it is returned by value throughout the SDK. Instead, create helper functions for each "subclass" of `Error` that would be desired.

### 4.29.2 Constructor & Destructor Documentation

#### 4.29.2.1 Error() [1/5]

```
rstudio::launcher_plugins::Error::Error (
    const Error & in_other )
```

Copy constructor.

##### Parameters

<code>in_other</code>	The error to copy.
-----------------------	--------------------

#### 4.29.2.2 Error() [2/5]

```
rstudio::launcher_plugins::Error::Error (
    std::string in_name,
    int in_code,
    const ErrorLocation & in_location )
```

Constructor.

##### Parameters

<code>in_name</code>	A contextual or categorical name for the error. (e.g. "RequestNotSupported")
<code>in_code</code>	The non-zero error code. Note that an error code of zero indicates success. (e.g. 1)
<code>in_location</code>	The location of the error.

**4.29.2.3 Error()** [3/5]

```
rstudio::launcher_plugins::Error::Error (
    std::string in_name,
    int in_code,
    const Error & in_cause,
    const ErrorLocation & in_location )
```

Constructor.

**Parameters**

<i>in_name</i>	A contextual or categorical name for the error. (e.g. "RequestNotSupported")
<i>in_code</i>	The non-zero error code. Note that an error code of zero indicates success. (e.g. 1)
<i>in_cause</i>	The error which caused this error.
<i>in_location</i>	The location of the error.

**4.29.2.4 Error()** [4/5]

```
rstudio::launcher_plugins::Error::Error (
    std::string in_name,
    int in_code,
    std::string in_message,
    const ErrorLocation & in_location )
```

Constructor.

**Parameters**

<i>in_name</i>	A contextual or categorical name for the error. (e.g. "RequestNotSupported")
<i>in_code</i>	The non-zero error code. Note that an error code of zero indicates success. (e.g. 1)
<i>in_message</i>	The detailed error message. (e.g. "The JobNetworkRequest is not supported by this plugin.")
<i>in_location</i>	The location of the error.

**4.29.2.5 Error()** [5/5]

```
rstudio::launcher_plugins::Error::Error (
    std::string in_name,
    int in_code,
    std::string in_message,
    const Error & in_cause,
    const ErrorLocation & in_location )
```

Constructor.

## Parameters

<i>in_name</i>	A contextual or categorical name for the error. (e.g. "RequestNotSupported")
<i>in_code</i>	The non-zero error code. Note that an error code of zero indicates success. (e.g. 1)
<i>in_message</i>	The detailed error message. (e.g. "The JobNetworkRequest is not supported by this plugin.")
<i>in_cause</i>	The error which caused this error.
<i>in_location</i>	The location of the error.

### 4.29.3 Member Function Documentation

#### 4.29.3.1 addOrUpdateProperty() [1/3]

```
void rstudio::launcher_plugins::Error::addOrUpdateProperty (
    const std::string & in_name,
    const std::string & in_value )
```

Add or updates a property of this error. If any properties with the specified name exist, they will all be updated.

## Parameters

<i>in_name</i>	The name of the property to add or update.
<i>in_value</i>	The new value of the property.

#### 4.29.3.2 addOrUpdateProperty() [2/3]

```
void rstudio::launcher_plugins::Error::addOrUpdateProperty (
    const std::string & in_name,
    const system::FilePath & in_value )
```

Add or updates a property of this error. If any properties with the specified name exist, they will all be updated.

## Parameters

<i>in_name</i>	The name of the property to add or update.
<i>in_value</i>	The new value of the property.

#### 4.29.3.3 addOrUpdateProperty() [3/3]

```
void rstudio::launcher_plugins::Error::addOrUpdateProperty (
    const std::string & in_name,
    int in_value )
```

Add or updates a property of this error. If any properties with the specified name exist, they will all be updated.

#### Parameters

<i>in_name</i>	The name of the property to add or update.
<i>in_value</i>	The new value of the property.

#### 4.29.3.4 addProperty() [1/3]

```
void rstudio::launcher_plugins::Error::addProperty (
    const std::string & in_name,
    const std::string & in_value )
```

Adds a property of this error. If a property with the same name already exists, a duplicate will be added.

#### Parameters

<i>in_name</i>	The name of the property to add or update.
<i>in_value</i>	The new value of the property.

#### 4.29.3.5 addProperty() [2/3]

```
void rstudio::launcher_plugins::Error::addProperty (
    const std::string & in_name,
    const system::FilePath & in_value )
```

Adds a property of this error. If a property with the same name already exists, a duplicate will be added.

#### Parameters

<i>in_name</i>	The name of the property to add or update.
<i>in_value</i>	The new value of the property.

#### 4.29.3.6 addProperty() [3/3]

```
void rstudio::launcher_plugins::Error::addProperty (
    const std::string & in_name,
    int in_value )
```

Adds a property of this error. If a property with the same name already exists, a duplicate will be added.

**Parameters**

<i>in_name</i>	The name of the property to add or update.
<i>in_value</i>	The new value of the property.

**4.29.3.7 asString()**

```
std::string rstudio::launcher_plugins::Error::asString ( ) const
```

Formats the error as a string.

**Returns**

The error formatted as a string.

**4.29.3.8 getCause()**

```
const Error& rstudio::launcher_plugins::Error::getCause ( ) const
```

Gets the error which caused this error.

**Returns**

The error which caused this error.

**4.29.3.9 getCode()**

```
int rstudio::launcher_plugins::Error::getCode ( ) const
```

Gets the error code.

**Returns**

The error code.



#### 4.29.3.10 getLocation()

```
const ErrorLocation& rstudio::launcher_plugins::Error::getLocation ( ) const
```

Gets the location where the error occurred.

##### Returns

The location where the error occurred.

#### 4.29.3.11 getMessage()

```
const std::string& rstudio::launcher_plugins::Error::getMessage ( ) const
```

Gets the error message.

##### Returns

The error message.

#### 4.29.3.12 getName()

```
const std::string& rstudio::launcher_plugins::Error::getName ( ) const
```

Gets the name of the error.

##### Returns

The name of the error.

#### 4.29.3.13 getProperties()

```
const ErrorProperties& rstudio::launcher_plugins::Error::getProperties ( ) const
```

Gets the custom properties of the error.

##### Returns

The custom properties of this error.

#### 4.29.3.14 getProperty()

```
std::string rstudio::launcher_plugins::Error::getProperty (
    const std::string & name ) const
```

Gets a custom property of this error.

**Parameters**

<i>name</i>	The name of the property to retrieve.
-------------	---------------------------------------

**Returns**

The value of the specified property, if it exists; empty string otherwise.

**4.29.3.15 getSummary()**

```
std::string rstudio::launcher_plugins::Error::getSummary ( ) const
```

Gets the cause of the error.

**Returns**

The cause of the error.

**4.29.3.16 isExpected()**

```
bool rstudio::launcher_plugins::Error::isExpected ( ) const
```

Gets whether this error was expected or not.

**Returns**

True if this error was expected; false otherwise.

**4.29.3.17 operator bool()**

```
rstudio::launcher_plugins::Error::operator bool ( ) const [explicit]
```

Overloaded operator bool to allow Errors to be treated as boolean values.

**Returns**

True if there is an error; false otherwise.

#### 4.29.3.18 operator"!"()

```
bool rstudio::launcher_plugins::Error::operator! ( ) const
```

Overloaded operator ! to allow Errors to be treated as boolean values.

##### Returns

True if there is not an error; false otherwise.

#### 4.29.3.19 operator"!="(

```
bool rstudio::launcher_plugins::Error::operator!= (
    const Error & in_other ) const
```

Inequality operator. Two errors are equal if their codes and names are the same.

**Parameters**

<i>in_other</i>	The error to compare with this error.
-----------------	---------------------------------------

**Returns**

True if *in\_other* is not equal to this error; false otherwise.

**4.29.3.20 operator==()**

```
bool rstudio::launcher_plugins::Error::operator== (
    const Error & in_other ) const
```

Equality operator. Two errors are equal if their codes and names are the same.

**Parameters**

<i>in_other</i>	The error to compare with this error.
-----------------	---------------------------------------

**Returns**

True if *in\_other* is equal to this error; false otherwise.

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/Error.hpp

**4.30 rstudio::launcher\_plugins::ErrorLocation Class Reference**

Class which represents the location of an error.

```
#include <Error.hpp>
```

**Public Member Functions**

- [ErrorLocation](#) ()  
*Default constructor.*
- [ErrorLocation](#) (const [ErrorLocation](#) &*in\_other*)  
*Copy constructor.*
- [ErrorLocation](#) ([ErrorLocation](#) &&*in\_other*) noexcept  
*Move constructor.*
- [ErrorLocation](#) (const char \**in\_function*, const char \**in\_file*, long *in\_line*)  
*Constructor.*
- [ErrorLocation](#) & [operator=](#) (const [ErrorLocation](#) &*in\_other*)

- Assignment operator.*
- bool `operator==` (const `ErrorLocation` &in\_location) const
- Equality comparison operator.*
- std::string `asString` () const
- Formats the error location as a string.*
- const std::string & `getFile` () const
- Gets the file where the error occurred.*
- const std::string & `getFunction` () const
- Gets the function where the error occurred.*
- long `getLine` () const
- Gets the line where the error occurred.*
- bool `hasLocation` () const
- Checks whether the location is set.*

### 4.30.1 Detailed Description

Class which represents the location of an error.

### 4.30.2 Constructor & Destructor Documentation

#### 4.30.2.1 ErrorLocation() [1/3]

```
rstudio::launcher_plugins::ErrorLocation::ErrorLocation (
    const ErrorLocation & in_other )
```

Copy constructor.

Parameters

<code>in_other</code>	The error location to move to this.
-----------------------	-------------------------------------

#### 4.30.2.2 ErrorLocation() [2/3]

```
rstudio::launcher_plugins::ErrorLocation::ErrorLocation (
    ErrorLocation && in_other ) [noexcept]
```

Move constructor.

Parameters

<code>in_other</code>	The error location to move to this.
-----------------------	-------------------------------------

#### 4.30.2.3 ErrorLocation() [3/3]

```
rstudio::launcher_plugins::ErrorLocation::ErrorLocation (
    const char * in_function,
    const char * in_file,
    long in_line )
```

Constructor.

##### Parameters

<i>in_function</i>	The function in which the error occurred.
<i>in_file</i>	The file in which the error occurred.
<i>in_line</i>	The line at which the error occurred.

### 4.30.3 Member Function Documentation

#### 4.30.3.1 asString()

```
std::string rstudio::launcher_plugins::ErrorLocation::asString ( ) const
```

Formats the error location as a string.

##### Returns

The error location formatted as a string.

#### 4.30.3.2 getFile()

```
const std::string& rstudio::launcher_plugins::ErrorLocation::getFile ( ) const
```

Gets the file where the error occurred.

##### Returns

The file where the error occurred.

#### 4.30.3.3 getFunction()

```
const std::string& rstudio::launcher_plugins::ErrorLocation::getFunction ( ) const
```

Gets the function where the error occurred.

##### Returns

The function where the error occurred.

#### 4.30.3.4 getLine()

```
long rstudio::launcher_plugins::ErrorLocation::getLine ( ) const
```

Gets the line where the error occurred.

##### Returns

The line where the error occurred.

#### 4.30.3.5 hasLocation()

```
bool rstudio::launcher_plugins::ErrorLocation::hasLocation ( ) const
```

Checks whether the location is set.

##### Returns

True if a location has been set; false otherwise.

#### 4.30.3.6 operator=()

```
ErrorLocation& rstudio::launcher_plugins::ErrorLocation::operator= (
    const ErrorLocation & in_other )
```

Assignment operator.

##### Parameters

<i>in_other</i>	The location to copy to this location.
-----------------	--

**Returns**

A reference to this location.

**4.30.3.7 operator==()**

```
bool rstudio::launcher_plugins::ErrorLocation::operator== (
    const ErrorLocation & in_location ) const
```

Equality comparison operator.

**Parameters**

<i>in_location</i>	The location to compare this location with.
--------------------	---

**Returns**

True if *in\_location* is the same as this location; false otherwise.

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

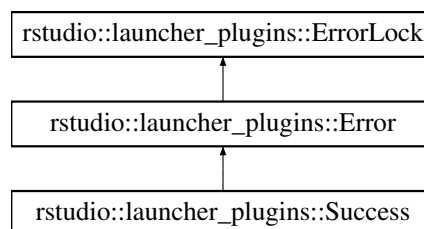
- `/workspaces/rstudio-launcher-plugin-sdk/sdk/include/Error.hpp`

**4.31 rstudio::launcher\_plugins::ErrorLock Class Reference**

A class which can be derived from in order to prevent child classes from being derived from further.

```
#include <Error.hpp>
```

Inheritance diagram for `rstudio::launcher_plugins::ErrorLock`:

**Friends**

- class **Error**
- class **Success**



### 4.31.1 Detailed Description

A class which can be derived from in order to prevent child classes from being derived from further.

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

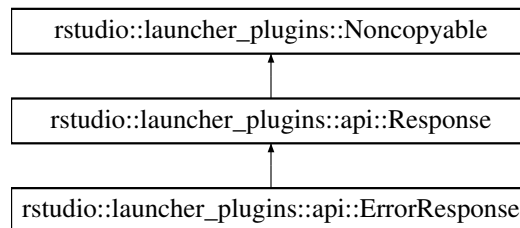
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/Error.hpp

## 4.32 rstudio::launcher\_plugins::api::ErrorResponse Class Reference

Class which represents an error response which can be sent to the Launcher in response to any request.

```
#include <Response.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::ErrorResponse:



### Public Types

- enum **Type** {  
**INVALID\_RESPONSE** = -1, **UNKNOWN** = 0, **REQUEST\_NOT\_SUPPORTED** = 1, **INVALID\_REQUEST** = 2,  
**JOB\_NOT\_FOUND** = 3, **PLUGIN\_RESTARTED** = 4, **TIMEOUT** = 5, **JOB\_NOT\_RUNNING** = 6,  
**JOB\_OUTPUT\_NOT\_FOUND** = 7, **INVALID\_JOB\_STATE** = 8, **JOB\_CONTROL\_FAILURE** = 9, **UNSUPPORTED\_VERSION** = 10 }

### Public Member Functions

- [ErrorResponse](#) (uint64\_t in\_requestId, [Type](#) in\_errorType, std::string in\_errorMessage)  
*Constructor.*
- [json::Object toJson](#) () const override  
*Converts this error response to a JSON object.*

### Additional Inherited Members

#### 4.32.1 Detailed Description

Class which represents an error response which can be sent to the Launcher in response to any request.

## 4.32.2 Constructor & Destructor Documentation

### 4.32.2.1 ErrorResponse()

```
rstudio::launcher_plugins::api::ErrorResponse::ErrorResponse (
    uint64_t in_requestId,
    Type in_errorType,
    std::string in_errorMessage )
```

Constructor.

#### Parameters

<i>in_requestId</i>	The ID of the request for which this error is being returned.
<i>in_errorType</i>	The type of error which occurred while processing the request.
<i>in_errorMessage</i>	A message describing the details of the error.

## 4.32.3 Member Function Documentation

### 4.32.3.1 toJson()

```
json::Object rstudio::launcher_plugins::api::ErrorResponse::toJson ( ) const [override],
[virtual]
```

Converts this error response to a JSON object.

#### Returns

The JSON object which represents this error response.

Reimplemented from [rstudio::launcher\\_plugins::api::Response](#).

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Response.hpp

## 4.33 rstudio::launcher\_plugins::api::ExposedPort Struct Reference

Struct which represents an exposed port on a containerized job.

```
#include <Job.hpp>
```

## Public Member Functions

- `json::Object toJson () const`  
Converts this *ExposedPort* to a JSON object which represents it.

## Static Public Member Functions

- static `Error fromJson (const json::Object &in_json, ExposedPort &out_exposedPort)`  
Constructs an *ExposedPort* from a JSON object which represents the exposed port.

## Public Attributes

- `Optional< int > PublishedPort`
- `std::string Protocol`
- `int TargetPort`

### 4.33.1 Detailed Description

Struct which represents an exposed port on a containerized job.

### 4.33.2 Member Function Documentation

#### 4.33.2.1 fromJson()

```
static Error rstudio::launcher_plugins::api::ExposedPort::fromJson (
    const json::Object & in_json,
    ExposedPort & out_exposedPort ) [static]
```

Constructs an *ExposedPort* from a JSON object which represents the exposed port.

#### Parameters

<i>in_json</i>	The JSON object which represents the exposed port.
<i>out_exposedPort</i>	The populated exposed port value. Not valid if an error is returned.

#### Returns

*Success* if *in\_json* could be parsed as an *ExposedPort*; *Error* otherwise.

#### 4.33.2.2 toJson()

```
json::Object rstudio::launcher_plugins::api::ExposedPort::toJson () const
```

Converts this *ExposedPort* to a JSON object which represents it.

**Returns**

The JSON object which represents this [ExposedPort](#).

**4.33.3 Member Data Documentation****4.33.3.1 Protocol**

```
std::string rstudio::launcher_plugins::api::ExposedPort::Protocol
```

The protocol of the port.

**4.33.3.2 PublishedPort**

```
Optional<int> rstudio::launcher_plugins::api::ExposedPort::PublishedPort
```

The published port.

**4.33.3.3 TargetPort**

```
int rstudio::launcher_plugins::api::ExposedPort::TargetPort
```

The target port.

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

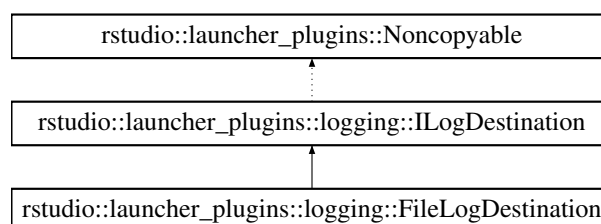
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Job.hpp

**4.34 rstudio::launcher\_plugins::logging::FileLogDestination Class Reference**

Class which allows sending log messages to a file.

```
#include <FileLogDestination.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::logging::FileLogDestination:



## Public Member Functions

- [FileLogDestination](#) (unsigned int in\_id, [LogLevel](#) in\_logLevel, const std::string &in\_programId, [FileLogOptions](#) in\_logOptions)  
*Constructor.*
- [~FileLogDestination](#) () override  
*Destructor.*
- unsigned int [getId](#) () const override  
*Gets the unique ID of this file log destination.*
- void [reload](#) () override  
*Reloads the log destination. Ensures that the log does not have any stale file handles.*
- void [writeLog](#) ([LogLevel](#) in\_logLevel, const std::string &in\_message) override  
*Writes a message to the log file.*

## Additional Inherited Members

### 4.34.1 Detailed Description

Class which allows sending log messages to a file.

### 4.34.2 Constructor & Destructor Documentation

#### 4.34.2.1 FileLogDestination()

```
rstudio::launcher_plugins::logging::FileLogDestination::FileLogDestination (
    unsigned int in_id,
    LogLevel in_logLevel,
    const std::string & in_programId,
    FileLogOptions in_logOptions )
```

Constructor.

#### Parameters

<i>in_id</i>	The ID of this log destination. Must be unique for each file log destination and > 100.
<i>in_logLevel</i>	The most detailed level of log to be written to this log file.
<i>in_programId</i>	The ID of this program.
<i>in_logOptions</i>	The options for log file creation and management.

If the log file cannot be opened, no logs will be written to the file. If there are other log destinations registered an error will be logged regarding the failure.

### 4.34.3 Member Function Documentation

#### 4.34.3.1 getId()

```
unsigned int rstudio::launcher_plugins::logging::FileLogDestination::getId ( ) const [override],
[virtual]
```

Gets the unique ID of this file log destination.

##### Returns

The unique ID of this file log destination.

Implements [rstudio::launcher\\_plugins::logging::ILogDestination](#).

#### 4.34.3.2 writeLog()

```
void rstudio::launcher_plugins::logging::FileLogDestination::writeLog (
    LogLevel in_logLevel,
    const std::string & in_message ) [override], [virtual]
```

Writes a message to the log file.

##### Parameters

<i>in_logLevel</i>	The log level of the message to write. Filtering is done prior to this call. This is for informational purposes only.
<i>in_message</i>	The message to write to the log file.

Implements [rstudio::launcher\\_plugins::logging::ILogDestination](#).

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/logging/FileLogDestination.hpp

## 4.35 rstudio::launcher\_plugins::logging::FileLogOptions Class Reference

Class which represents the options for a file logger.

```
#include <FileLogDestination.hpp>
```

### Public Member Functions

- [FileLogOptions](#) (system::FilePath in\_directory)  
*Constructor.*

- [FileLogOptions](#) ([system::FilePath](#) in\_directory, std::string in\_fileMode, double in\_maxSizeMb, bool in\_doRotation, bool in\_includePid)  
*Constructor.*
- const [system::FilePath](#) & [getDirectory](#) () const  
*Gets the directory where log files should be written.*
- const std::string & [getFileMode](#) () const  
*Gets the permissions with which log files should be created.*
- double [getMaxSizeMb](#) () const  
*Gets the maximum size of log files, in MB.*
- bool [doRotation](#) () const  
*Returns whether or not to rotate log files before overwriting them.*
- bool [includePid](#) () const  
*Returns whether or not to include the PID in the logs.*

### 4.35.1 Detailed Description

Class which represents the options for a file logger.

### 4.35.2 Constructor & Destructor Documentation

#### 4.35.2.1 FileLogOptions() [1/2]

```
rstudio::launcher_plugins::logging::FileLogOptions::FileLogOptions (
    system::FilePath in_directory )
```

Constructor.

This constructor is intentionally not explicit to allow for conversion from [system::FilePath](#) to [FileLogOptions](#).

Parameters

<i>in_directory</i>	The directory in which to create log files.
---------------------	---

#### 4.35.2.2 FileLogOptions() [2/2]

```
rstudio::launcher_plugins::logging::FileLogOptions::FileLogOptions (
    system::FilePath in_directory,
    std::string in_fileMode,
    double in_maxSizeMb,
    bool in_doRotation,
    bool in_includePid )
```

Constructor.

**Parameters**

<i>in_directory</i>	The directory in which to create log files.
<i>in_fileMode</i>	The permissions to set on log files.
<i>in_maxSizeMb</i>	The maximum size of log files, in MB, before they are rotated and/or overwritten.
<i>in_doRotation</i>	Whether to rotate log files or not.
<i>in_includePid</i>	Whether to include the PID of the process in the logs.

**4.35.3 Member Function Documentation****4.35.3.1 doRotation()**

```
bool rstudio::launcher_plugins::logging::FileLogOptions::doRotation ( ) const
```

Returns whether or not to rotate log files before overwriting them.

**Returns**

True if log files should be rotated; false otherwise.

**4.35.3.2 getDirectory()**

```
const system::FilePath& rstudio::launcher_plugins::logging::FileLogOptions::getDirectory ( )  
const
```

Gets the directory where log files should be written.

**Returns**

The directory where log files should be written.

**4.35.3.3 getFileMode()**

```
const std::string& rstudio::launcher_plugins::logging::FileLogOptions::getFileMode ( ) const
```

Gets the permissions with which log files should be created.

**Returns**

The permissions with which log files should be created.



#### 4.35.3.4 getMaxSizeMb()

```
double rstudio::launcher_plugins::logging::FileLogOptions::getMaxSizeMb ( ) const
```

Gets the maximum size of log files, in MB.

##### Returns

The maximum size of log files, in MB.

#### 4.35.3.5 includePid()

```
bool rstudio::launcher_plugins::logging::FileLogOptions::includePid ( ) const
```

Returns whether or not to include the PID in the logs.

##### Returns

True if the PID should be included in the logs; false otherwise.

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

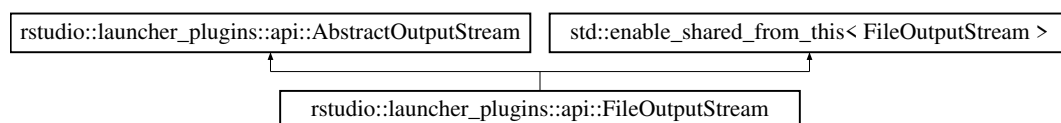
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/logging/FileLogDestination.hpp

## 4.36 rstudio::launcher\_plugins::api::FileOutputStream Class Reference

Streams job output data from a file.

```
#include <FileOutputStream.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::FileOutputStream:



### Public Member Functions

- [FileOutputStream](#) (OutputType in\_outputType, api::JobPtr in\_job, AbstractOutputStream::OnOutput in\_onOutput, [AbstractOutputStream::OnComplete](#) in\_onComplete, AbstractOutputStream::OnError in\_onError, [system::TimeDuration](#) in\_maxWaitTime=[system::TimeDuration::Seconds](#)(10))  
*Constructor.*
- virtual [~FileOutputStream](#) ()=default  
*Virtual destructor for inheritance.*
- [Error start](#) () override  
*Starts the output stream.*
- void [stop](#) () override  
*Stops the output stream.*

## Additional Inherited Members

### 4.36.1 Detailed Description

Streams job output data from a file.

### 4.36.2 Constructor & Destructor Documentation

#### 4.36.2.1 FileOutputStream()

```
rstudio::launcher_plugins::api::FileOutputStream::FileOutputStream (
    OutputType in_outputType,
    api::JobPtr in_job,
    AbstractOutputStream::OnOutput in_onOutput,
    AbstractOutputStream::OnComplete in_onComplete,
    AbstractOutputStream::OnError in_onError,
    system::TimeDuration in_maxWaitTime = system::TimeDuration::Seconds(10) )
```

Constructor.

#### Parameters

<i>in_outputType</i>	The type of job output to stream.
<i>in_job</i>	The job for which output should be streamed.
<i>in_onOutput</i>	Callback function which will be invoked when data is reported.
<i>in_onComplete</i>	Callback function which will be invoked when the stream is complete.
<i>in_onError</i>	Callback function which will be invoked if an error occurs.
<i>in_maxWaitTime</i>	The maximum amount of time to wait for the output files to be created before reporting an error.

### 4.36.3 Member Function Documentation

#### 4.36.3.1 start()

```
Error rstudio::launcher_plugins::api::FileOutputStream::start ( ) [override], [virtual]
```

Starts the output stream.

#### Returns

[Success](#) if the stream could be started; [Error](#) otherwise.

Implements [rstudio::launcher\\_plugins::api::AbstractOutputStream](#).

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/stream/FileOutputStream.hpp

## 4.37 rstudio::launcher\_plugins::system::FilePath Class Reference

Class which represents a path on the system. May be any type of file (e.g. directory, symlink, regular file, etc.)

```
#include <FilePath.hpp>
```

### Public Types

- enum [MoveType](#) { [MoveDirect](#), [MoveCrossDevice](#) }  
*Enum which represents the type of move to perform.*
- typedef std::function< bool(int, const [FilePath](#) &)> [RecursivelterationFunction](#)  
*Function which recursively iterates over [FilePath](#) objects.*

### Public Member Functions

- [FilePath](#) ()  
*Default constructor.*
- [FilePath](#) (const std::string &in\_absolutePath)  
*Constructor.*
- bool [operator==](#) (const [FilePath](#) &in\_other) const  
*Comparison operator. File paths are equal if their absolute representations are equal.*
- bool [operator!=](#) (const [FilePath](#) &in\_other) const  
*Comparison operator. File paths are equal if their absolute representations are equal.*
- bool [operator<](#) (const [FilePath](#) &in\_other) const  
*Less-than operator to establish natural order. The natural order is based on the absolute representation of the paths.*
- [Error](#) [changeFileMode](#) (const std::string &in\_fileModeStr) const  
*Changes the file mode to the specified file mode.*
- [Error](#) [changeFileMode](#) (FileMode in\_fileMode, bool in\_setStickyBit=false) const  
*Changes the file mode to the specified file mode.*
- [Error](#) [changeOwnership](#) (const [system::User](#) &in\_newUser, bool in\_recursive=false, const [RecursivelterationFunction](#) &in\_shouldChown=[RecursivelterationFunction](#)()) const  
*Changes the ownership of the file or directory to the specified user.*
- [FilePath](#) [completeChildPath](#) (const std::string &in\_filePath) const  
*Gets the provided relative path as a child of this path.*
- [Error](#) [completeChildPath](#) (const std::string &in\_filePath, [FilePath](#) &out\_childPath) const  
*Gets the provided relative path as a child of this path.*
- [FilePath](#) [completePath](#) (const std::string &in\_filePath) const  
*Completes the provided path relative to this path. If the provided path is not relative, it will be returned as is. Relative paths such as "." are permitted.*
- [Error](#) [copy](#) (const [FilePath](#) &in\_targetPath, bool overwrite=false) const  
*Copies this file path to the specified location.*
- [Error](#) [copyDirectoryRecursive](#) (const [FilePath](#) &in\_targetPath, bool overwrite=false) const  
*Copies this directory recursively to the specified location.*
- [Error](#) [createDirectory](#) (const std::string &in\_filePath) const  
*Creates the specified directory.*
- [Error](#) [ensureDirectory](#) () const  
*Creates this directory, if it does not exist.*
- [Error](#) [ensureFile](#) () const  
*Creates this file, if it does not exist.*

- `bool exists () const`  
*Checks whether this file path exists in the file system.*
- `std::string getAbsolutePath () const`  
*Gets the full absolute representation of this file path.*
- `std::string getAbsolutePathNative () const`  
*Gets the full absolute representation of this file path in native format.*
- `std::string getCanonicalPath () const`  
*Gets the canonical representation of this file path. The path must exist so that its canonical location on disk can be obtained.*
- `Error getChildren (std::vector< FilePath > &out_filePaths) const`  
*Gets the children of this directory. Sub-directories will not be traversed.*
- `Error getChildrenRecursive (const RecursivelterationFunction &in_iterationFunction) const`  
*Gets the children of this directory recursively. Sub-directories will be traversed.*
- `std::string getExtension () const`  
*Gets the extension of the file, including the leading '.'.*
- `std::string getExtensionLowerCase () const`  
*Gets the extension of the file in lower case, including the leading '.'.*
- `Error getFileMode (FileMode &out_fileMode) const`  
*Gets the posix file mode of this file or directory.*
- `std::string getFilename () const`  
*Gets only the name of the file, including the extension.*
- `std::time_t getLastWriteTime () const`  
*Get the last time this file path was written.*
- `std::string getLexicallyNormalPath () const`  
*Gets the lexically normal representation of this file path, with . and .. components resolved and/or removed.*
- `std::string getMimeType (const std::string &in_defaultType="text/plain") const`  
*Gets the mime content type of this file.*
- `FilePath getParent () const`  
*Gets the parent directory of this file path.*
- `std::string getRelativePath (const FilePath &in_parentPath) const`  
*Gets the representation of this path, relative to the provided path.*
- `uintmax_t getSize () const`  
*Gets the size of this file path in bytes.*
- `uintmax_t getSizeRecursive () const`  
*Gets the size of this file path and all sub-directories and files in it, in bytes.*
- `std::string getStem () const`  
*Gets only the name of the file, excluding the extension.*
- `bool hasExtension (const std::string &in_extension) const`  
*Checks whether this file has the specified extension.*
- `bool hasExtensionLowerCase (const std::string &in_extension) const`  
*Checks whether this file has the specified extension when it is converted to lower case.*
- `bool hasTextMimeType () const`  
*Checks whether this file has a text mime content type.*
- `bool isDirectory () const`  
*Checks whether this file path is a directory.*
- `bool isEmpty () const`  
*Checks whether this file path contains a path or not.*
- `bool isEquivalentTo (const FilePath &in_other) const`  
*Checks whether this file path points to the same location in the filesystem as the specified file path.*
- `bool isHidden () const`  
*Checks whether this file path is a hidden file or directory.*

- [Error isReadable](#) (bool &out\_readable) const  
*Checks whether this file path is readable.*
- bool [isRegularFile](#) () const  
*Checks whether this file path is a regular file.*
- bool [isSymlink](#) () const  
*Checks whether this file path is a symbolic link.*
- bool [isWithin](#) (const [FilePath](#) &in\_scopePath) const  
*Checks whether this file path is within the specified file path.*
- [Error isWriteable](#) (bool &out\_writeable) const  
*Checks whether this file path is writeable.*
- [Error makeCurrentPath](#) (bool in\_autoCreate=false) const  
*Changes the current working directory to location represented by this file path.*
- [Error move](#) (const [FilePath](#) &in\_targetPath, [MoveType](#) in\_type=[MoveCrossDevice](#), bool overwrite=false) const  
*Moves the current directory to the specified directory.*
- [Error moveIndirect](#) (const [FilePath](#) &in\_targetPath, bool overwrite=false) const  
*Performs an indirect move by copying this directory to the target and then deleting this directory.*
- [Error openForRead](#) (std::shared\_ptr< std::istream > &out\_stream) const  
*Opens this file for read.*
- [Error openForWrite](#) (std::shared\_ptr< std::ostream > &out\_stream, bool in\_truncate=true) const  
*Opens this file for write.*
- [Error remove](#) () const  
*Removes this file or directory from the filesystem.*
- [Error removeIfExists](#) () const  
*Removes this file or directory from the filesystem, if it exists.*
- [Error resetDirectory](#) () const  
*Removes the directory represented by this [FilePath](#), if it exists, and recreates it.*
- [FilePath resolveSymlink](#) () const  
*Resolves this symbolic link to the location to which it is pointing. If this [FilePath](#) is not a symbolic link, the original [FilePath](#) is returned.*
- void [setLastWriteTime](#) (std::time\_t in\_time=:time(nullptr)) const  
*Sets the last time that this file was modified to the specified time.*
- [Error testWritePermissions](#) () const  
*Checks if a file can be written to by opening the file.*

## Static Public Member Functions

- static std::string [createAliasedPath](#) (const [FilePath](#) &in\_filePath, const [FilePath](#) &in\_userHomePath)  
*Creates a path in which the user home path will be replaced by the ~ alias.*
- static bool [exists](#) (const std::string &in\_filePath)  
*Checks whether the specified path exists.*
- static bool [isEqualCaseInsensitive](#) (const [FilePath](#) &in\_filePath1, const [FilePath](#) &in\_filePath2)  
*Checks whether the two provided files are equal, ignoring case. Two files are equal if their absolute paths are equal.*
- static bool [isRootPath](#) (const std::string &in\_filePath)  
*Checks whether the specified path is a root path or a relative path.*
- static [Error makeCurrent](#) (const std::string &in\_filePath)  
*Changes the current working directory to the specified path.*
- static [FilePath resolveAliasedPath](#) (const std::string &in\_aliasedPath, const [FilePath](#) &in\_userHomePath)  
*Resolves the '~' alias within the path to the user's home path.*
- static [FilePath safeCurrentPath](#) (const [FilePath](#) &in\_revertToPath)

*Checks whether the current working directory exists. If it does not, moves the current working directory to the specified path.*

- static [Error tempFilePath](#) ([FilePath](#) &out\_filePath)  
*Creates a randomly named file in the temp directory.*
- static [Error tempFilePath](#) (const std::string &in\_extension, [FilePath](#) &out\_filePath)  
*Creates a randomly named file with the specified extension in the temp directory.*
- static [Error uniqueFilePath](#) (const std::string &in\_basePath, [FilePath](#) &out\_filePath)  
*Creates a file with a random name in the specified directory.*
- static [Error uniqueFilePath](#) (const std::string &in\_basePath, const std::string &in\_extension, [FilePath](#) &out\_filePath)  
*Creates a file with a random name and the specified extension in the specified directory.*

### 4.37.1 Detailed Description

Class which represents a path on the system. May be any type of file (e.g. directory, symlink, regular file, etc.)

### 4.37.2 Member Typedef Documentation

#### 4.37.2.1 RecursiveIterationFunction

```
typedef std::function<bool(int, const FilePath&)> rstudio::launcher\_plugins::system::FilePath::RecursiveIterationFunction
```

Function which recursively iterates over [FilePath](#) objects.

#### Parameters

<i>int</i>	The depth of the iteration.
<a href="#">FilePath</a>	The current <a href="#">FilePath</a> object in the recursive iteration.

#### Returns

True if the computation can continue; false otherwise.

### 4.37.3 Member Enumeration Documentation

#### 4.37.3.1 MoveType

```
enum rstudio::launcher\_plugins::system::FilePath::MoveType
```

Enum which represents the type of move to perform.

## Enumerator

MoveDirect	Attempt to perform an ordinary move
MoveCrossDevice	Perform an ordinary move, but fallback to copy/delete on cross-device errors

## 4.37.4 Constructor &amp; Destructor Documentation

## 4.37.4.1 FilePath()

```
rstudio::launcher_plugins::system::FilePath::FilePath (
    const std::string & in_absolutePath ) [explicit]
```

Constructor.

## Parameters

<i>in_absolutePath</i>	The string representation of the path.
------------------------	--

## 4.37.5 Member Function Documentation

## 4.37.5.1 changeFileMode() [1/2]

```
Error rstudio::launcher_plugins::system::FilePath::changeFileMode (
    const std::string & in_fileModeStr ) const
```

Changes the file mode to the specified file mode.

## Parameters

<i>in_fileModeStr</i>	The posix file mode string. e.g. rwxr-xr-x.
-----------------------	---

## Returns

[Success](#) if the file mode could be changed; [Error](#) otherwise.

## 4.37.5.2 changeFileMode() [2/2]

```
Error rstudio::launcher_plugins::system::FilePath::changeFileMode (
    FileMode in_fileMode,
    bool in_setStickyBit = false ) const
```

Changes the file mode to the specified file mode.

#### Parameters

<i>in_fileMode</i>	The new file mode.
<i>in_setStickyBit</i>	Whether to set the sticky bit on this file.

#### Returns

[Success](#) if the file mode could be changed; [Error](#) otherwise.

### 4.37.5.3 changeOwnership()

```
Error rstudio::launcher_plugins::system::FilePath::changeOwnership (
    const system::User & in_newUser,
    bool in_recursive = false,
    const RecursiveIterationFunction & in_shouldChown = RecursiveIterationFunction()
) const
```

Changes the ownership of the file or directory to the specified user.

#### Parameters

<i>in_newUser</i>	The user who should own the file.
<i>in_recursive</i>	If this <a href="#">FilePath</a> is a directory, whether to recursively change ownership on all files and directories within this directory.
<i>in_shouldChown</i>	A recursive iteration function which allows the caller to filter files and directories. If a file or directory should have its ownership changed, this function should return true.

#### Returns

[Success](#) if the file, and optionally all nested files and directories, had their ownership changed; [Error](#) otherwise.

### 4.37.5.4 completeChildPath() [1/2]

```
FilePath rstudio::launcher_plugins::system::FilePath::completeChildPath (
    const std::string & in_filePath ) const
```

Gets the provided relative path as a child of this path.

#### Parameters

<i>in_filePath</i>	The path to get as a child of this path. Must be a relative path.
--------------------	---



## Returns

The completed child path, or this path if the provided path was not relative or another error occurred.

## 4.37.5.5 completeChildPath() [2/2]

```
Error rstudio::launcher_plugins::system::FilePath::completeChildPath (
    const std::string & in_filePath,
    FilePath & out_childPath ) const
```

Gets the provided relative path as a child of this path.

## Parameters

<i>in_filePath</i>	The path to get as a child of this path. Must be a relative path that refers to a path strictly within this one (i.e. ".." isn't allowed)
<i>out_childPath</i>	The completed child path. Not valid if an error is returned.

## Returns

[Success](#) if the child path could be completed; [Error](#) otherwise.

## 4.37.5.6 completePath()

```
FilePath rstudio::launcher_plugins::system::FilePath::completePath (
    const std::string & in_filePath ) const
```

Completes the provided path relative to this path. If the provided path is not relative, it will be returned as is. Relative paths such as ".." are permitted.

## Parameters

<i>in_filePath</i>	<i>in_filePath</i> The path to complete.
--------------------	--

## Returns

The completed path if the provided path was relative, or the provided path if it was not relative.

## 4.37.5.7 copy()

```
Error rstudio::launcher_plugins::system::FilePath::copy (
    const FilePath & in_targetPath,
    bool overwrite = false ) const
```

Copies this file path to the specified location.

## Parameters

<i>in_targetPath</i>	The location to copy this file path to.
<i>overwrite</i>	Whether to overwrite the file if one exists in target path.

## Returns

[Success](#) if the copy could be completed; [Error](#) otherwise.

**4.37.5.8 copyDirectoryRecursive()**

```
Error rstudio::launcher_plugins::system::FilePath::copyDirectoryRecursive (
    const FilePath & in_targetPath,
    bool overwrite = false ) const
```

Copies this directory recursively to the specified location.

## Parameters

<i>in_targetPath</i>	The location to which to copy this directory and its contents.
<i>overwrite</i>	Whether to overwrite the file if one exists in target path.

## Returns

[Success](#) if the copy could be completed; [Error](#) otherwise.

**4.37.5.9 createAliasedPath()**

```
static std::string rstudio::launcher_plugins::system::FilePath::createAliasedPath (
    const FilePath & in_filePath,
    const FilePath & in_userHomePath ) [static]
```

Creates a path in which the user home path will be replaced by the ~ alias.

## Parameters

<i>in_filePath</i>	The path to convert to an aliased path.
<i>in_userHomePath</i>	The user home path.

## Returns

If the path is within the user home path, an aliased path; the original path otherwise.

**4.37.5.10 createDirectory()**

```
Error rstudio::launcher_plugins::system::FilePath::createDirectory (
    const std::string & in_filePath ) const
```

Creates the specified directory.

**Parameters**

<code>in_filePath</code>	The directory to create, relative to this directory.
--------------------------	--

**Returns**

**Success** if the directory could be created; **Error** if it could not be created for any reason.

**4.37.5.11 ensureDirectory()**

```
Error rstudio::launcher_plugins::system::FilePath::ensureDirectory ( ) const
```

Creates this directory, if it does not exist.

**Returns**

**Success** if the directory could be created or it exists already; **Error** otherwise.

**4.37.5.12 ensureFile()**

```
Error rstudio::launcher_plugins::system::FilePath::ensureFile ( ) const
```

Creates this file, if it does not exist.

**Returns**

**Success** if the file could be created or it exists already; **Error** otherwise.

**4.37.5.13 exists() [1/2]**

```
bool rstudio::launcher_plugins::system::FilePath::exists ( ) const
```

Checks whether this file path exists in the file system.

**Returns**

True if this file path exists; false otherwise.

**4.37.5.14 exists() [2/2]**

```
static bool rstudio::launcher_plugins::system::FilePath::exists (
    const std::string & in_filePath ) [static]
```

Checks whether the specified path exists.

**Parameters**

<i>in_filePath</i>	The path to check.
--------------------	--------------------

**Returns**

True if the specified path exists; false otherwise.

**4.37.5.15 getAbsolutePath()**

```
std::string rstudio::launcher_plugins::system::FilePath::getAbsolutePath ( ) const
```

Gets the full absolute representation of this file path.

**Returns**

The absolute representation of this file path.

**4.37.5.16 getAbsolutePathNative()**

```
std::string rstudio::launcher_plugins::system::FilePath::getAbsolutePathNative ( ) const
```

Gets the full absolute representation of this file path in native format.

**Returns**

The absolute representation of this file path in native format.

**4.37.5.17 getCanonicalPath()**

```
std::string rstudio::launcher_plugins::system::FilePath::getCanonicalPath ( ) const
```

Gets the canonical representation of this file path. The path must exist so that its canonical location on disk can be obtained.

**Returns**

The canonical representation of this file path.

**4.37.5.18 getChildren()**

```
Error rstudio::launcher_plugins::system::FilePath::getChildren (
    std::vector< FilePath > & out_filePaths ) const
```

Gets the children of this directory. Sub-directories will not be traversed.

## Parameters

<code>out_filePaths</code>	The children of this directory.
----------------------------	---------------------------------

## Returns

**Success** if the children could be retrieved; **Error** otherwise (e.g. if this path does not exist).

**4.37.5.19 getChildrenRecursive()**

```
Error rstudio::launcher_plugins::system::FilePath::getChildrenRecursive (
    const RecursiveIterationFunction & in_iterationFunction ) const
```

Gets the children of this directory recursively. Sub-directories will be traversed.

## Parameters

<code>in_iterationFunction</code>	The function to perform for each child of this directory.
-----------------------------------	---

## Returns

**Success** if the children could be iterated; **Error** otherwise (e.g. if tis path does not exist).

**4.37.5.20 getExtension()**

```
std::string rstudio::launcher_plugins::system::FilePath::getExtension ( ) const
```

Gets the extension of the file, including the leading '.'.

## Returns

The extension of the file.

**4.37.5.21 getExtensionLowerCase()**

```
std::string rstudio::launcher_plugins::system::FilePath::getExtensionLowerCase ( ) const
```

Gets the extension of the file in lower case, including the leading '.'.

## Returns

The extension of the file in lower case.

#### 4.37.5.22 getFileMode()

```
Error rstudio::launcher_plugins::system::FilePath::getFileMode (
    FileMode & out_fileMode ) const
```

Gets the posix file mode of this file or directory.

## Parameters

<code>out_fileMode</code>	The file mode of this file or directory. Invalid if an error is returned.
---------------------------	---

## Returns

[Success](#) if the file mode could be retrieved; [Error](#) otherwise.

**4.37.5.23 getFilename()**

```
std::string rstudio::launcher_plugins::system::FilePath::getFilename ( ) const
```

Gets only the name of the file, including the extension.

## Returns

The name of the file, including the extension.

**4.37.5.24 getLastWriteTime()**

```
std::time_t rstudio::launcher_plugins::system::FilePath::getLastWriteTime ( ) const
```

Get the last time this file path was written.

## Returns

The time of the last write.

**4.37.5.25 getLexicallyNormalPath()**

```
std::string rstudio::launcher_plugins::system::FilePath::getLexicallyNormalPath ( ) const
```

Gets the lexically normal representation of this file path, with . and .. components resolved and/or removed.

## Returns

The lexically normal representation of this file path.

**4.37.5.26 getMimeType()**

```
std::string rstudio::launcher_plugins::system::FilePath::getMimeType (
    const std::string & in_defaultType = "text/plain" ) const
```

Gets the mime content type of this file.

## Parameters

<i>in_defaultType</i>	The default mime content type to return if this file does not have a mime content type. Default: "text/plain".
-----------------------	--

## Returns

The mime content type of this file, or the default type if the file does not have a mime content type.

**4.37.5.27 getParent()**

```
FilePath rstudio::launcher_plugins::system::FilePath::getParent ( ) const
```

Gets the parent directory of this file path.

## Returns

The parent directory of this file path.

**4.37.5.28 getRelativePath()**

```
std::string rstudio::launcher_plugins::system::FilePath::getRelativePath (
    const FilePath & in_parentPath ) const
```

Gets the representation of this path, relative to the provided path.

## Parameters

<i>in_parentPath</i>	The parent of this path.
----------------------	--------------------------

## Returns

The representation of this path, relative to the provided parent, or empty if this path is not within the provided parent.

**4.37.5.29 getSize()**

```
uintmax_t rstudio::launcher_plugins::system::FilePath::getSize ( ) const
```

Gets the size of this file path in bytes.

## Returns

The size of this file path in bytes.



#### 4.37.5.30 getSizeRecursive()

```
uintmax_t rstudio::launcher_plugins::system::FilePath::getSizeRecursive ( ) const
```

Gets the size of this file path and all sub-directories and files in it, in bytes.

##### Returns

The size of this file path and all sub-directories and files in it, in bytes.

#### 4.37.5.31 getStem()

```
std::string rstudio::launcher_plugins::system::FilePath::getStem ( ) const
```

Gets only the name of the file, excluding the extension.

##### Returns

The name of the file, excluding the extension.

#### 4.37.5.32 hasExtension()

```
bool rstudio::launcher_plugins::system::FilePath::hasExtension (
    const std::string & in_extension ) const
```

Checks whether this file has the specified extension.

##### Parameters

<i>in_extension</i>	The extension to check this file for.
---------------------	---------------------------------------

##### Returns

True if the extension of this file matches the specified extension; false otherwise.

#### 4.37.5.33 hasExtensionLowerCase()

```
bool rstudio::launcher_plugins::system::FilePath::hasExtensionLowerCase (
    const std::string & in_extension ) const
```

Checks whether this file has the specified extension when it is converted to lower case.

**Parameters**

<i>in_extension</i>	The extension to check this file for.
---------------------	---------------------------------------

**Returns**

True if the lower case extension of this file matches the specified extension; false otherwise.

**4.37.5.34 hasTextMimeType()**

```
bool rstudio::launcher_plugins::system::FilePath::hasTextMimeType ( ) const
```

Checks whether this file has a text mime content type.

**Returns**

True if this file has a text mime content type; false otherwise.

**4.37.5.35 isDirectory()**

```
bool rstudio::launcher_plugins::system::FilePath::isDirectory ( ) const
```

Checks whether this file path is a directory.

**Returns**

True if this file path is a directory; false otherwise.

**4.37.5.36 isEmpty()**

```
bool rstudio::launcher_plugins::system::FilePath::isEmpty ( ) const
```

Checks whether this file path contains a path or not.

**Returns**

True if this file path does not contain a path; false otherwise.

**4.37.5.37 isEqualCaseInsensitive()**

```
static bool rstudio::launcher_plugins::system::FilePath::isEqualCaseInsensitive (
    const FilePath & in_filePath1,
    const FilePath & in_filePath2 ) [static]
```

Checks whether the two provided files are equal, ignoring case. Two files are equal if their absolute paths are equal.

## Parameters

<i>in_filePath1</i>	The first file to compare.
<i>in_filePath2</i>	The second file to compare.

## Returns

True if the absolute representations of the paths are equal, case insensitively; false otherwise.

**4.37.5.38 isEquivalentTo()**

```
bool rstudio::launcher_plugins::system::FilePath::isEquivalentTo (
    const FilePath & in_other ) const
```

Checks whether this file path points to the same location in the filesystem as the specified file path.

## Parameters

<i>in_other</i>	The file path to which to compare this file path to.
-----------------	--

## Returns

True if this file path points to the same location in the filesystem as the specified file path; false otherwise.

**4.37.5.39 isHidden()**

```
bool rstudio::launcher_plugins::system::FilePath::isHidden ( ) const
```

Checks whether this file path is a hidden file or directory.

## Returns

True if this file path is a hidden file or directory; false otherwise.

**4.37.5.40 isReadable()**

```
Error rstudio::launcher_plugins::system::FilePath::isReadable (
    bool & out_readable ) const
```

Checks whether this file path is readable.

## Parameters

<i>out_readable</i>	True if this file path is readable by the current effective user; false if it is not. Invalid if this method returns an error.
---------------------	--

## Returns

[Success](#) if the readability of this file could be checked; [Error](#) otherwise. (e.g. EACCES).

**4.37.5.41 isRegularFile()**

```
bool rstudio::launcher_plugins::system::FilePath::isRegularFile ( ) const
```

Checks whether this file path is a regular file.

## Returns

True if this file path is a regular file; false otherwise.

**4.37.5.42 isRootPath()**

```
static bool rstudio::launcher_plugins::system::FilePath::isRootPath (
    const std::string & in_filePath ) [static]
```

Checks whether the specified path is a root path or a relative path.

## Parameters

<i>in_filePath</i>	The path to check.
--------------------	--------------------

## Returns

True if the path is a root path; false if the path is a relative path.

**4.37.5.43 isSymlink()**

```
bool rstudio::launcher_plugins::system::FilePath::isSymlink ( ) const
```

Checks whether this file path is a symbolic link.

## Returns

True if this file path is a symbolic link; false otherwise.

#### 4.37.5.44 isWithin()

```
bool rstudio::launcher_plugins::system::FilePath::isWithin (
    const FilePath & in_scopePath ) const
```

Checks whether this file path is within the specified file path.

##### Parameters

<i>in_scopePath</i>	The potential parent path.
---------------------	----------------------------

##### Returns

True if this file path is within the specified path, or if the two paths are equal; false otherwise.

#### 4.37.5.45 isWriteable()

```
Error rstudio::launcher_plugins::system::FilePath::isWriteable (
    bool & out_writeable ) const
```

Checks whether this file path is writeable.

##### Parameters

<i>out_writeable</i>	True if this file path is writeable by the current effective user; false if it is not. Invalid if this method returns an error.
----------------------	---

##### Returns

**Success** if the writeability of this file could be checked; **Error** otherwise. (e.g. EACCES).

#### 4.37.5.46 makeCurrent()

```
static Error rstudio::launcher_plugins::system::FilePath::makeCurrent (
    const std::string & in_filePath ) [static]
```

Changes the current working directory to the specified path.

##### Parameters

<i>in_filePath</i>	The path to which to change the current working directory.
--------------------	--

**Returns**

[Success](#) if `in_path` exists and can be moved to; [Error](#) otherwise.

**4.37.5.47 makeCurrentPath()**

```
Error rstudio::launcher_plugins::system::FilePath::makeCurrentPath (
    bool in_autoCreate = false ) const
```

Changes the current working directory to location represented by this file path.

**Parameters**

<i>in_autoCreate</i>	Controls whether to create the location represented by this file path if it does not exist. Default: <code>false</code> .
----------------------	---

**Returns**

[Success](#) if the working directory was changed; [Error](#) otherwise.

**4.37.5.48 move()**

```
Error rstudio::launcher_plugins::system::FilePath::move (
    const FilePath & in_targetPath,
    MoveType in_type = MoveCrossDevice,
    bool overwrite = false ) const
```

Moves the current directory to the specified directory.

**Parameters**

<i>in_targetPath</i>	The location to which to move this directory.
<i>in_type</i>	The type of move to perform, direct or cross device. See <a href="#">MoveType</a> for more details. Default: <code>MoveCrossDevice</code> .
<i>overwrite</i>	Whether to overwrite the file if one exists in target path.

**Returns**

[Success](#) if this directory could be moved to the target; [Error](#) otherwise.

**4.37.5.49 moveIndirect()**

```
Error rstudio::launcher_plugins::system::FilePath::moveIndirect (
    const FilePath & in_targetPath,
    bool overwrite = false ) const
```

Performs an indirect move by copying this directory to the target and then deleting this directory.

#### Parameters

<i>in_targetPath</i>	The location to which to move this directory.
<i>overwrite</i>	Whether to overwrite the file if one exists in target path.

#### Returns

[Success](#) if this directory could be moved to the target; [Error](#) otherwise.

#### 4.37.5.50 openForRead()

```
Error rstudio::launcher_plugins::system::FilePath::openForRead (  
    std::shared_ptr< std::istream > & out_stream ) const
```

Opens this file for read.

#### Parameters

<i>out_stream</i>	The input stream for this open file.
-------------------	--------------------------------------

#### Returns

[Success](#) if the file was opened; system error otherwise (e.g. EPERM, ENOENT, etc.)

#### 4.37.5.51 openForWrite()

```
Error rstudio::launcher_plugins::system::FilePath::openForWrite (  
    std::shared_ptr< std::ostream > & out_stream,  
    bool in_truncate = true ) const
```

Opens this file for write.

#### Parameters

<i>out_stream</i>	The output stream for this open file.
<i>in_truncate</i>	Whether to truncate the existing contents of the file. Default: true.

#### Returns

[Success](#) if the file was opened; system error otherwise (e.g. EPERM, ENOENT, etc.)

#### 4.37.5.52 operator"!=()

```
bool rstudio::launcher_plugins::system::FilePath::operator!= (
    const FilePath & in_other ) const
```

Comparison operator. File paths are equal if their absolute representations are equal.

##### Parameters

<i>in_other</i>	The file path to compare with this file path.
-----------------	---

##### Returns

True if the file paths are not equal; false otherwise.

#### 4.37.5.53 operator<()

```
bool rstudio::launcher_plugins::system::FilePath::operator< (
    const FilePath & in_other ) const
```

Less-than operator to establish natural order. The natural order is based on the absolute representation of the paths.

##### Parameters

<i>in_other</i>	The path to which to compare this path.
-----------------	---

##### Returns

True if the absolute representation of this path is less, alphabetically, than the absolute representation of the other path; false otherwise.

#### 4.37.5.54 operator==(())

```
bool rstudio::launcher_plugins::system::FilePath::operator==(
    const FilePath & in_other ) const
```

Comparison operator. File paths are equal if their absolute representations are equal.

##### Parameters

<i>in_other</i>	The file path to compare with this file path.
-----------------	---



**Returns**

True if the file paths are equal; false otherwise.

**4.37.5.55 remove()**

```
Error rstudio::launcher_plugins::system::FilePath::remove ( ) const
```

Removes this file or directory from the filesystem.

**Returns**

**Success** if the file or directory was removed; **Error** otherwise.

**4.37.5.56 removeIfExists()**

```
Error rstudio::launcher_plugins::system::FilePath::removeIfExists ( ) const
```

Removes this file or directory from the filesystem, if it exists.

**Returns**

**Success** if the file or directory was removed, or if the file did not exist; **Error** otherwise.

**4.37.5.57 resetDirectory()**

```
Error rstudio::launcher_plugins::system::FilePath::resetDirectory ( ) const
```

Removes the directory represented by this [FilePath](#), if it exists, and recreates it.

**Returns**

**Success** if the directory was able to be created freshly; **Error** otherwise.

**4.37.5.58 resolveAliasedPath()**

```
static FilePath rstudio::launcher_plugins::system::FilePath::resolveAliasedPath (
    const std::string & in_aliasedPath,
    const FilePath & in_userHomePath ) [static]
```

Resolves the '~' alias within the path to the user's home path.

## Parameters

<i>in_aliasedPath</i>	The aliased path to resolve.
<i>in_userHomePath</i>	The user's home path.

## Returns

The resolved path.

**4.37.5.59 resolveSymlink()**

```
FilePath rstudio::launcher_plugins::system::FilePath::resolveSymlink ( ) const
```

Resolves this symbolic link to the location to which it is pointing. If this [FilePath](#) is not a symbolic link, the original [FilePath](#) is returned.

## Returns

The resolved symbolic link, or this path if it is not a symbolic link.

**4.37.5.60 safeCurrentPath()**

```
static FilePath rstudio::launcher_plugins::system::FilePath::safeCurrentPath (
    const FilePath & in_revertToPath ) [static]
```

Checks whether the current working directory exists. If it does not, moves the current working directory to the specified path.

## Parameters

<i>in_revertToPath</i>	The path to revert to if the current working directory no longer exists.
------------------------	--

## Returns

The current working directory.

**4.37.5.61 setLastWriteTime()**

```
void rstudio::launcher_plugins::system::FilePath::setLastWriteTime (
    std::time_t in_time = ::time(nullptr) ) const
```

Sets the last time that this file was modified to the specified time.

## Parameters

<i>in_time</i>	The time to which to set the last write time of this file. Default: now.
----------------	--

**4.37.5.62 tempFilePath()** [1/2]

```
static Error rstudio::launcher_plugins::system::FilePath::tempFilePath (
    const std::string & in_extension,
    FilePath & out_filePath ) [static]
```

Creates a randomly named file with the specified extension in the temp directory.

## Parameters

<i>in_extension</i>	The extension with which to create the file. The extension should include the leading '.', e.g. '.zip'.
<i>out_filePath</i>	The absolute path of the newly created file, or an empty file path if the file could not be created.

## Returns

**Success** if the file could be created; **Error** otherwise.

**4.37.5.63 tempFilePath()** [2/2]

```
static Error rstudio::launcher_plugins::system::FilePath::tempFilePath (
    FilePath & out_filePath ) [static]
```

Creates a randomly named file in the temp directory.

## Parameters

<i>out_filePath</i>	The absolute path of the newly created file, or an empty file path if the file could not be created.
---------------------	--

## Returns

**Success** if the file could be created; **Error** otherwise.

**4.37.5.64 testWritePermissions()**

```
Error rstudio::launcher_plugins::system::FilePath::testWritePermissions ( ) const
```

Checks if a file can be written to by opening the file.

To be successful, the file must already exist on the system. If write access is not absolutely necessary, use `isWritable` instead.

## Returns

[Success](#) if file can be written to; system error otherwise (e.g. EPERM, ENOENT, etc.)

**4.37.5.65 uniqueFilePath() [1/2]**

```
static Error rstudio::launcher_plugins::system::FilePath::uniqueFilePath (
    const std::string & in_basePath,
    const std::string & in_extension,
    FilePath & out_filePath ) [static]
```

Creates a file with a random name and the specified extension in the specified directory.

## Parameters

<i>in_basePath</i>	The path at which to create the file.
<i>in_extension</i>	The extension with which to create the file. The extension should include the leading '.', e.g. '.zip'.
<i>out_filePath</i>	The absolute path of the newly created file, or an empty file path if the file could not be created.

## Returns

[Success](#) if the file could be created; [Error](#) otherwise.

**4.37.5.66 uniqueFilePath() [2/2]**

```
static Error rstudio::launcher_plugins::system::FilePath::uniqueFilePath (
    const std::string & in_basePath,
    FilePath & out_filePath ) [static]
```

Creates a file with a random name in the specified directory.

## Parameters

<i>in_basePath</i>	The path at which to create the file.
<i>out_filePath</i>	The absolute path of the newly created file, or an empty file path if the file could not be created.

## Returns

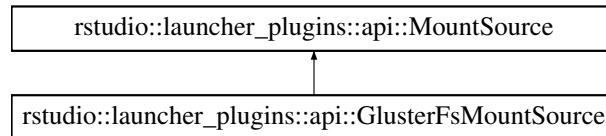
[Success](#) if the file could be created; [Error](#) otherwise.

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/FilePath.hpp

## 4.38 rstudio::launcher\_plugins::api::GlusterFsMountSource Struct Reference

Inheritance diagram for rstudio::launcher\_plugins::api::GlusterFsMountSource:



### Public Member Functions

- [GlusterFsMountSource](#) ()  
*Constructor.*
- `std::string` [getEndpoints](#) () const  
*Gets the name of the endpoints object that represents a Gluster cluster configuration.*
- `std::string` [getPath](#) () const  
*Gets the name of the GlusterFs volume mount.*

### Static Public Member Functions

- static [Error fromJson](#) (const `json::Object` &in\_json, [GlusterFsMountSource](#) &out\_mountSource)  
*Constructs a [GlusterFsMountSource](#) from a JSON object which represents the mount source.*

### Additional Inherited Members

#### 4.38.1 Member Function Documentation

##### 4.38.1.1 fromJson()

```
static Error rstudio::launcher_plugins::api::GlusterFsMountSource::fromJson (
    const json::Object & in_json,
    GlusterFsMountSource & out_mountSource ) [static]
```

Constructs a [GlusterFsMountSource](#) from a JSON object which represents the mount source.

#### Parameters

<i>in_json</i>	The JSON object which represents the mount source.
<i>out_mountSource</i>	The populated mount source value. Not valid if an error is returned.

**Returns**

[Success](#) if `in_json` could be parsed as a [GlusterFsMountSource](#); [Error](#) otherwise.

**4.38.1.2 getEndpoints()**

```
std::string rstudio::launcher_plugins::api::GlusterFsMountSource::getEndpoints ( ) const
```

Gets the name of the endpoints object that represents a Gluster cluster configuration.

**Exceptions**

<code>std::logic_error</code>	if the 'endpoints' field cannot be found.
-------------------------------	---

**Returns**

The name of the endpoints object that represents a Gluster cluster configuration

**4.38.1.3 getPath()**

```
std::string rstudio::launcher_plugins::api::GlusterFsMountSource::getPath ( ) const
```

Gets the name of the GlusterFs volume mount.

**Exceptions**

<code>std::logic_error</code>	if the 'path' field cannot be found.
-------------------------------	--------------------------------------

**Returns**

The name of the GlusterFs volume to mount.

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

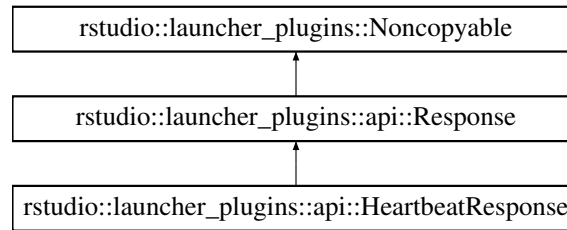
- `/workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Job.hpp`

## 4.39 rstudio::launcher\_plugins::api::HeartbeatResponse Class Reference

Class which represents a heartbeat response which should be sent to the Launcher every configured heartbeat-interval-seconds.

```
#include <Response.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::HeartbeatResponse:



## Public Member Functions

- [HeartbeatResponse](#) ()  
*Constructor.*

## Additional Inherited Members

### 4.39.1 Detailed Description

Class which represents a heartbeat response which should be sent to the Launcher every configured heartbeat-interval-seconds.

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

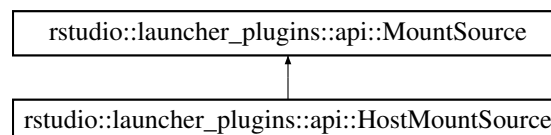
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Response.hpp

## 4.40 rstudio::launcher\_plugins::api::HostMountSource Struct Reference

Represents a path to mount on the same host as the [Job](#).

```
#include <Job.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::HostMountSource:



## Public Member Functions

- `std::string` [getPath](#) () const  
*Gets the path on the current host to be mounted.*

## Static Public Member Functions

- static [Error fromJson](#) (const [json::Object](#) &in\_json, [HostMountSource](#) &out\_mountSource)  
*Constructs a [HostMountSource](#) from a JSON object which represents the mount source.*

## Friends

- class [MountSource](#)

## Additional Inherited Members

### 4.40.1 Detailed Description

Represents a path to mount on the same host as the [Job](#).

### 4.40.2 Member Function Documentation

#### 4.40.2.1 fromJson()

```
static Error rstudio::launcher_plugins::api::HostMountSource::fromJson (
    const json::Object & in_json,
    HostMountSource & out_mountSource ) [static]
```

Constructs a [HostMountSource](#) from a JSON object which represents the mount source.

#### Parameters

<i>in_json</i>	The JSON object which represents the mount source.
<i>out_mountSource</i>	The populated mount source value. Not valid if an error is returned.

#### Returns

[Success](#) if in\_json could be parsed as a [HostMountSource](#); [Error](#) otherwise.

#### 4.40.2.2 getPath()

```
std::string rstudio::launcher_plugins::api::HostMountSource::getPath ( ) const
```

Gets the path on the current host to be mounted.



## Exceptions

<code>std::logic_error</code>	if the 'path' field cannot be found.
-------------------------------	--------------------------------------

## Returns

The path on the current host to be mounted.

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

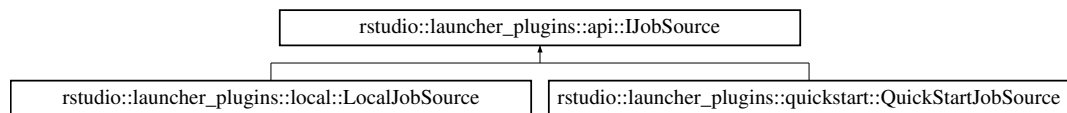
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Job.hpp

## 4.41 rstudio::launcher\_plugins::api::IJobSource Class Reference

Generic interface for communicating with a [Job](#) Source. Implementation is plugin specific.

```
#include <IJobSource.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::IJobSource:



### Public Member Functions

- virtual `~IJobSource()`=default  
*Virtual Destructor.*
- virtual `Error initialize()`=0  
*Initializes the Job Source.*
- virtual `bool cancelJob(JobPtr in_job, bool &out_isComplete, std::string &out_statusMessage)`=0  
*Cancels a pending job.*
- virtual `Error getConfiguration(const system::User &in_user, JobSourceConfiguration &out_configuration)`  
const =0  
*Gets the configuration and capabilities of this Job Source for the specified user.*
- virtual `Error getNetworkInfo(JobPtr in_job, NetworkInfo &out_networkInfo)` const =0  
*Gets the network information for the specified job.*
- virtual `bool killJob(JobPtr in_job, bool &out_isComplete, std::string &out_statusMessage)`=0  
*Forcibly kills a running job.*
- virtual `bool resumeJob(JobPtr in_job, bool &out_isComplete, std::string &out_statusMessage)`=0  
*Resumes a suspended job.*
- virtual `bool stopJob(JobPtr in_job, bool &out_isComplete, std::string &out_statusMessage)`=0  
*Stops a running job.*
- virtual `bool suspendJob(JobPtr in_job, bool &out_isComplete, std::string &out_statusMessage)`=0  
*Suspends a running job.*
- virtual `Error submitJob(JobPtr io_job, bool &out_wasInvalidRequest)` const =0  
*Submits a job to the Job Scheduling System.*
- virtual `Error createOutputStream(OutputType in_outputType, JobPtr in_job, AbstractOutputStream::OnOutput in_onOutput, AbstractOutputStream::OnComplete in_onComplete, AbstractOutputStream::OnError in_onError, OutputStreamPtr &out_outputStream)`=0  
*Creates an output stream for the specified job.*
- virtual `Error createResourceStream(ConstJobPtr in_job, comms::AbstractLauncherCommunicatorPtr in_launcherCommunicator, AbstractResourceStreamPtr &out_resourceStream)`=0  
*Creates a resource utilization metric stream for the specified job.*

## Protected Member Functions

- [IJobSource](#) (jobs::JobRepositoryPtr in\_jobRepository, jobs::JobStatusNotifierPtr in\_jobStatusNotifier)  
*Constructor.*

## Protected Attributes

- jobs::JobRepositoryPtr [m\\_jobRepository](#)
- jobs::JobStatusNotifierPtr [m\\_jobStatusNotifier](#)

### 4.41.1 Detailed Description

Generic interface for communicating with a [Job](#) Source. Implementation is plugin specific.

### 4.41.2 Constructor & Destructor Documentation

#### 4.41.2.1 IJobSource()

```
rstudio::launcher_plugins::api::IJobSource::IJobSource (
    jobs::JobRepositoryPtr in_jobRepository,
    jobs::JobStatusNotifierPtr in_jobStatusNotifier ) [inline], [protected]
```

Constructor.

#### Parameters

<i>in_jobRepository</i>	The job repository, from which to look up jobs.
<i>in_jobStatusNotifier</i>	The job status notifier to which to post or from which to receive job status updates.

### 4.41.3 Member Function Documentation

#### 4.41.3.1 cancelJob()

```
virtual bool rstudio::launcher_plugins::api::IJobSource::cancelJob (
    JobPtr in_job,
    bool & out_isComplete,
    std::string & out_statusMessage ) [pure virtual]
```

Cancels a pending job.

This method will not be invoked unless the job is currently pending. The [Job](#) lock will be held when this method is invoked.

## Parameters

<i>in_job</i>	The job to be canceled.
<i>out_isComplete</i>	Whether the cancel operation completed successfully (true) or not (false).
<i>out_statusMessage</i>	The status message of the cancel operation, if any.

## Returns

False if the cancel operation is not supported; true otherwise.

Implemented in [rstudio::launcher\\_plugins::local::LocalJobSource](#), and [rstudio::launcher\\_plugins::quickstart::QuickStartJobSource](#).

## 4.41.3.2 createOutputStream()

```
virtual Error rstudio::launcher_plugins::api::IJobSource::createOutputStream (
    OutputType in_outputType,
    JobPtr in_job,
    AbstractOutputStream::OnOutput in_onOutput,
    AbstractOutputStream::OnComplete in_onComplete,
    AbstractOutputStream::OnError in_onError,
    OutputStreamPtr & out_outputStream ) [pure virtual]
```

Creates an output stream for the specified job.

## Parameters

<i>in_outputType</i>	The type of job output to stream.
<i>in_job</i>	The job for which output should be streamed.
<i>in_onOutput</i>	Callback function which will be invoked when data is reported.
<i>in_onComplete</i>	Callback function which will be invoked when the stream is complete.
<i>in_onError</i>	Callback function which will be invoked if an error occurs.
<i>out_outputStream</i>	The newly created output stream, on <a href="#">Success</a> .

## Returns

[Success](#) if the output stream could be created; [Error](#) otherwise.

Implemented in [rstudio::launcher\\_plugins::quickstart::QuickStartJobSource](#), and [rstudio::launcher\\_plugins::local::LocalJobSource](#).

## 4.41.3.3 createResourceStream()

```
virtual Error rstudio::launcher_plugins::api::IJobSource::createResourceStream (
    ConstJobPtr in_job,
    comms::AbstractLauncherCommunicatorPtr in_launcherCommunicator,
    AbstractResourceStreamPtr & out_resourceStream ) [pure virtual]
```

Creates a resource utilization metric stream for the specified job.

## Parameters

<i>in_job</i>	The job for which resource utilization metrics should be streamed.
<i>in_launcherCommunicator</i>	The communicator with which to send responses to the Launcher.
<i>out_resourceStream</i>	The newly created resource utilization metric stream, on <a href="#">Success</a> .

## Returns

Success if the stream could be created; the [Error](#) that occurred otherwise.

Implemented in [rstudio::launcher\\_plugins::quickstart::QuickStartJobSource](#), and [rstudio::launcher\\_plugins::local::LocalJobSource](#).

## 4.41.3.4 getConfiguration()

```
virtual Error rstudio::launcher_plugins::api::IJobSource::getConfiguration (
    const system::User & in_user,
    JobSourceConfiguration & out_configuration ) const [pure virtual]
```

Gets the configuration and capabilities of this [Job](#) Source for the specified user.

This function controls the options that will be available to users when launching jobs.

NOTE: Many of the values here should most likely be controllable by Launcher administrators when they configure the Launcher. For more details, see the RStudio Launcher Plugin SDK QuickStart Guide TODO #7.

## Parameters

<i>in_user</i>	The user who made the request to see the configuration and capabilities of the Cluster. This may be used to return a different configuration based on any configured user profiles. For more information about user profiles, see the 'User Profiles' subsection of the 'Advanced Features' section of the RStudio Launcher Plugin SDK Developer's Guide.
<i>out_configuration</i>	The configuration and capabilities of this <a href="#">Job</a> Source, for the specified user.

## Returns

[Success](#) if the configuration and capabilities for this [Job](#) Source could be populated; [Error](#) otherwise.

Implemented in [rstudio::launcher\\_plugins::quickstart::QuickStartJobSource](#), and [rstudio::launcher\\_plugins::local::LocalJobSource](#).

## 4.41.3.5 getNetworkInfo()

```
virtual Error rstudio::launcher_plugins::api::IJobSource::getNetworkInfo (
    JobPtr in_job,
    NetworkInfo & out_networkInfo ) const [pure virtual]
```

Gets the network information for the specified job.

## Parameters

<i>in_job</i>	The job for which to retrieve network information.
<i>out_networkInfo</i>	The network information of the specified job, if no error occurred.

## Returns

[Success](#) if the network information could be retrieved; [Error](#) otherwise.

Implemented in [rstudio::launcher\\_plugins::quickstart::QuickStartJobSource](#), and [rstudio::launcher\\_plugins::local::LocalJobSource](#).

#### 4.41.3.6 initialize()

```
virtual Error rstudio::launcher_plugins::api::IJobSource::initialize ( ) [pure virtual]
```

Initializes the [Job](#) Source.

This function should return an error if communication with the job source fails.

## Returns

[Success](#) if the job source could be initialized; [Error](#) otherwise.

Implemented in [rstudio::launcher\\_plugins::local::LocalJobSource](#), and [rstudio::launcher\\_plugins::quickstart::QuickStartJobSource](#).

#### 4.41.3.7 killJob()

```
virtual bool rstudio::launcher_plugins::api::IJobSource::killJob (
    JobPtr in_job,
    bool & out_isComplete,
    std::string & out_statusMessage ) [pure virtual]
```

Forcibly kills a running job.

This method should perform the equivalent of sending a SIGKILL to a process. This method will not be invoked unless the job is currently running. The [Job](#) lock will be held when this method is invoked.

## Parameters

<i>in_job</i>	The job to be killed.
<i>out_isComplete</i>	Whether the kill operation completed successfully (true) or not (false).
<i>out_statusMessage</i>	The status message of the kill operation, if any.

**Returns**

False if the kill operation is not supported; true otherwise.

Implemented in [rstudio::launcher\\_plugins::quickstart::QuickStartJobSource](#), and [rstudio::launcher\\_plugins::local::LocalJobSource](#).

**4.41.3.8 resumeJob()**

```
virtual bool rstudio::launcher_plugins::api::IJobSource::resumeJob (
    JobPtr in_job,
    bool & out_isComplete,
    std::string & out_statusMessage ) [pure virtual]
```

Resumes a suspended job.

This method should perform the equivalent of sending a SIGCONT to a process. This method will not be invoked unless the job is currently suspended. The [Job](#) lock will be held when this method is invoked.

**Parameters**

<i>in_job</i>	The job to be resumed.
<i>out_isComplete</i>	Whether the resume operation completed successfully (true) or not (false).
<i>out_statusMessage</i>	The status message of the resume operation, if any.

**Returns**

False if the resume operation is not supported; true otherwise.

Implemented in [rstudio::launcher\\_plugins::quickstart::QuickStartJobSource](#), and [rstudio::launcher\\_plugins::local::LocalJobSource](#).

**4.41.3.9 stopJob()**

```
virtual bool rstudio::launcher_plugins::api::IJobSource::stopJob (
    JobPtr in_job,
    bool & out_isComplete,
    std::string & out_statusMessage ) [pure virtual]
```

Stops a running job.

This method should perform the equivalent of sending a SIGTERM to a process. This method will not be invoked unless the job is currently running. The [Job](#) lock will be held when this method is invoked.

**Parameters**

<i>in_job</i>	The job to be stopped.
<i>out_isComplete</i>	Whether the stop operation completed successfully (true) or not (false).
<i>out_statusMessage</i>	The status message of the stop operation, if any.

### Returns

False if the stop operation is not supported; true otherwise.

Implemented in [rstudio::launcher\\_plugins::local::LocalJobSource](#), and [rstudio::launcher\\_plugins::quickstart::QuickStartJobSource](#).

#### 4.41.3.10 submitJob()

```
virtual Error rstudio::launcher_plugins::api::IJobSource::submitJob (
    JobPtr io_job,
    bool & out_wasInvalidRequest ) const [pure virtual]
```

Submits a job to the [Job](#) Scheduling System.

### Parameters

<i>io_job</i>	The <a href="#">Job</a> to be submitted. On successful submission, the <a href="#">Job</a> should be updated with relevant details, such as the ID of the job, the Submission time, the actual <a href="#">Job</a> Queue (if applicable), and the current status.
<i>out_wasInvalidRequest</i>	Whether the requested <a href="#">Job</a> was invalid, based on the features supported by the <a href="#">Job</a> Scheduling System.

### Returns

[Success](#) if the job could be submitted to the [Job](#) Scheduling System; [Error](#) otherwise.

Implemented in [rstudio::launcher\\_plugins::quickstart::QuickStartJobSource](#), and [rstudio::launcher\\_plugins::local::LocalJobSource](#).

#### 4.41.3.11 suspendJob()

```
virtual bool rstudio::launcher_plugins::api::IJobSource::suspendJob (
    JobPtr in_job,
    bool & out_isComplete,
    std::string & out_statusMessage ) [pure virtual]
```

Suspends a running job.

This method should perform the equivalent of sending a SIGSTOP to a process. A suspended job should be able to be resumed at a later time. This method will not be invoked unless the job is currently running. The [Job](#) lock will be held when this method is invoked.

### Parameters

<i>in_job</i>	The job to be suspended.
<i>out_isComplete</i>	Whether the suspend operation completed successfully (true) or not (false).
<i>out_statusMessage</i>	The status message of the suspend operation, if any.

**Returns**

False if the suspend operation is not supported; true otherwise.

Implemented in [rstudio::launcher\\_plugins::local::LocalJobSource](#), and [rstudio::launcher\\_plugins::quickstart::QuickStartJobSource](#).

**4.41.4 Member Data Documentation****4.41.4.1 m\_jobRepository**

```
jobs::JobRepositoryPtr rstudio::launcher_plugins::api::IJobSource::m_jobRepository [protected]
```

The job repository, from which to look up jobs.

**4.41.4.2 m\_jobStatusNotifier**

```
jobs::JobStatusNotifierPtr rstudio::launcher_plugins::api::IJobSource::m_jobStatusNotifier [protected]
```

The job status notifier to which to post or from which to receive job status updates.

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

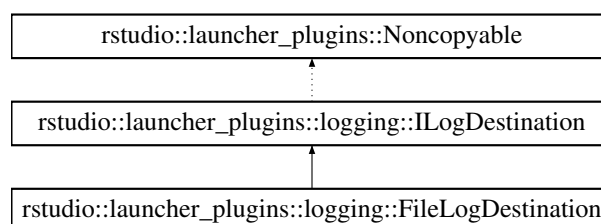
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/IJobSource.hpp

**4.42 rstudio::launcher\_plugins::logging::ILogDestination Class Reference**

Interface which allows a logger to write a log message to a destination.

```
#include <ILogDestination.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::logging::ILogDestination:





## Public Member Functions

- [ILogDestination](#) ([LogLevel](#) in\_logLevel)  
*Constructor.*
- virtual [~ILogDestination](#) ()=default  
*Virtual destructor to allow for inheritance.*
- virtual unsigned int [getId](#) () const =0  
*Gets the unique ID of the log destination.*
- virtual void [reload](#) ()=0  
*Reloads the log destination. Ensures that the log does not have any stale file handles.*
- [LogLevel](#) [getLogLevel](#) ()  
*Gets the maximum level of logs that will be written to this log destination.*
- virtual void [writeLog](#) ([LogLevel](#) in\_logLevel, const std::string &in\_message)=0  
*Writes a message to this log destination.*

## Protected Attributes

- [LogLevel](#) m\_logLevel  
*The maximum level of log messages to write for this logger.*

### 4.42.1 Detailed Description

Interface which allows a logger to write a log message to a destination.

Log destinations IDs 0 - 100 are reserved for SDK provided log destinations.

### 4.42.2 Constructor & Destructor Documentation

#### 4.42.2.1 ILogDestination()

```
rstudio::launcher_plugins::logging::ILogDestination::ILogDestination (
    LogLevel in_logLevel ) [inline], [explicit]
```

Constructor.

Parameters

<a href="#">in_logLevel</a>	The most detailed level of log to be written to this log destination.
-----------------------------	---

### 4.42.3 Member Function Documentation

#### 4.42.3.1 getId()

```
virtual unsigned int rstudio::launcher_plugins::logging::ILogDestination::getId ( ) const
[pure virtual]
```

Gets the unique ID of the log destination.

##### Returns

The unique ID of the log destination.

Implemented in [rstudio::launcher\\_plugins::logging::FileLogDestination](#).

#### 4.42.3.2 getLogLevel()

```
LogLevel rstudio::launcher_plugins::logging::ILogDestination::getLogLevel ( ) [inline]
```

Gets the maximum level of logs that will be written to this log destination.

##### Returns

This log destination's maximum log level.

#### 4.42.3.3 writeLog()

```
virtual void rstudio::launcher_plugins::logging::ILogDestination::writeLog (
    LogLevel in_logLevel,
    const std::string & in_message ) [pure virtual]
```

Writes a message to this log destination.

##### Parameters

<i>in_logLevel</i>	The log level of the message to write.
<i>in_message</i>	The message to write to the destination.

Implemented in [rstudio::launcher\\_plugins::logging::FileLogDestination](#).

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/logging/ILogDestination.hpp

## 4.43 rstudio::launcher\_plugins::options::Options::Init Class Reference

Class for initializing [Options](#).

```
#include <Options.hpp>
```

## Public Member Functions

- [Init](#) ([Options](#) &in\_owner)  
*Helper class which initializes [Options](#).*
- template<class T >  
[Init](#) & [operator](#)() (const char \*in\_name, [Value](#)< T > &in\_value, const char \*in\_description)  
*Operator which initializes a specific option value.*
- template<class T >  
[Init](#) & [operator](#)() (const char \*in\_name, [Value](#)< T > &&in\_value, const char \*in\_description)  
*Operator which initializes a specific option value.*

### 4.43.1 Detailed Description

Class for initializing [Options](#).

### 4.43.2 Constructor & Destructor Documentation

#### 4.43.2.1 Init()

```
rstudio::launcher_plugins::options::Options::Init::Init (
    Options & in_owner ) [explicit]
```

Helper class which initializes [Options](#).

#### Parameters

<i>in_owner</i>	The <a href="#">Options</a> object which owns this init object.
-----------------	---

### 4.43.3 Member Function Documentation

#### 4.43.3.1 operator>() [1/2]

```
template<class T >
Init& rstudio::launcher_plugins::options::Options::Init::operator() (
    const char * in_name,
    Value< T > && in_value,
    const char * in_description )
```

Operator which initializes a specific option value.

#### Template Parameters

<i>T</i>	The type of the option.
----------	-------------------------

#### Parameters

<i>in_name</i>	The name of the option.
<i>in_value</i>	The value object, which holds the default and the storage object. The <a href="#">Value</a> object is not usable after this call.
<i>in_description</i>	The description of the option.

#### Returns

A reference to this [Init](#) object.

#### 4.43.3.2 operator>() [2/2]

```
template<class T >
Init& rstudio::launcher_plugins::options::Options::Init::operator() (
    const char * in_name,
    Value< T > & in_value,
    const char * in_description )
```

Operator which initializes a specific option value.

#### Template Parameters

<i>T</i>	The type of the option.
----------	-------------------------

#### Parameters

<i>in_name</i>	The name of the option.
<i>in_value</i>	The value object, which holds the default and the storage object. The <a href="#">Value</a> object is not usable after this call.
<i>in_description</i>	The description of the option.

#### Returns

A reference to this [Init](#) object.

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/options/Options.hpp

## 4.44 rstudio::launcher\_plugins::system::posix::IpAddress Struct Reference

Represents an IP address.

```
#include <PosixSystem.hpp>
```

### Public Attributes

- std::string [Name](#)
- std::string [Address](#)

#### 4.44.1 Detailed Description

Represents an IP address.

#### 4.44.2 Member Data Documentation

##### 4.44.2.1 Address

```
std::string rstudio::launcher_plugins::system::posix::IpAddress::Address
```

The address of the IP address.

##### 4.44.2.2 Name

```
std::string rstudio::launcher_plugins::system::posix::IpAddress::Name
```

The name of the IP address.

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

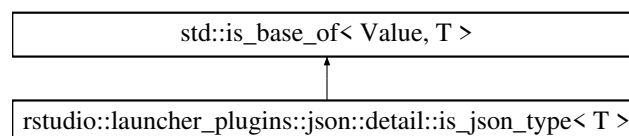
- [/workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/PosixSystem.hpp](#)

## 4.45 rstudio::launcher\_plugins::json::detail::is\_json\_type< T > Struct Template Reference

Struct which is either a child class of std::true\_type or std::false\_type depending on whether T is a JSON type (e.g. [Value](#), [Object](#), [Array](#)) or not (e.g. int, bool, string, float, etc.).

```
#include <Json.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::json::detail::is\_json\_type< T >:



### 4.45.1 Detailed Description

```
template<typename T>
struct rstudio::launcher_plugins::json::detail::is_json_type< T >
```

Struct which is either a child class of `std::true_type` or `std::false_type` depending on whether `T` is a JSON type (e.g. [Value](#), [Object](#), [Array](#)) or not (e.g. `int`, `bool`, `string`, `float`, etc.).

#### Template Parameters

<code>T</code>	The type of the object for which to test the type.
----------------	--

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

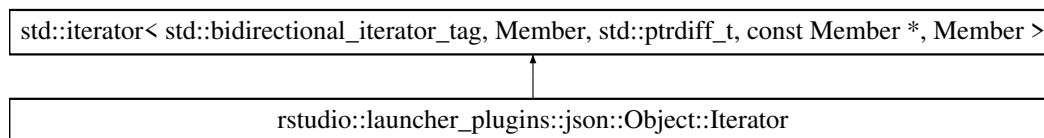
- `/workspaces/rstudio-launcher-plugin-sdk/sdk/include/json/Json.hpp`

## 4.46 rstudio::launcher\_plugins::json::Object::Iterator Class Reference

Class which allows iterating over the members of a JSON object.

```
#include <Json.hpp>
```

Inheritance diagram for `rstudio::launcher_plugins::json::Object::Iterator`:



### Public Member Functions

- [Iterator](#) (const [Object](#) \*in\_parent, std::ptrdiff\_t in\_startPos=0)  
*Constructor.*
- [Iterator](#) (const [Iterator](#) &in\_other)=default  
*Copy constructor.*
- [Iterator](#) & [operator=](#) (const [Iterator](#) &in\_other)  
*Assignment operator.*
- [Iterator](#) & [operator++](#) ()  
*Pre-increment operator.*
- [Iterator](#) & [operator--](#) ()  
*Pre-decrement operator.*
- [Iterator](#) [operator++](#) (int)  
*Post-increment operator.*
- [Iterator](#) [operator--](#) (int)  
*Post-decrement operator.*
- bool [operator==](#) (const [Iterator](#) &in\_other) const  
*Equality operator.*
- bool [operator!=](#) (const [Iterator](#) &in\_other) const  
*Inequality operator.*
- reference [operator\\*](#) () const  
*Dereference operator.*

## Friends

- class **Object**

### 4.46.1 Detailed Description

Class which allows iterating over the members of a JSON object.

### 4.46.2 Constructor & Destructor Documentation

#### 4.46.2.1 Iterator() [1/2]

```
rstudio::launcher_plugins::json::Object::Iterator::Iterator (
    const Object * in_parent,
    std::ptrdiff_t in_startPos = 0 ) [explicit]
```

Constructor.

##### Parameters

<i>in_parent</i>	The parent object which will be iterated.
<i>in_startPos</i>	The starting position of the iterator. Default: the first member.

#### 4.46.2.2 Iterator() [2/2]

```
rstudio::launcher_plugins::json::Object::Iterator::Iterator (
    const Iterator & in_other ) [default]
```

Copy constructor.

##### Parameters

<i>in_other</i>	The iterator to copy.
-----------------	-----------------------

### 4.46.3 Member Function Documentation

#### 4.46.3.1 operator"!="()

```
bool rstudio::launcher_plugins::json::Object::Iterator::operator!= (
    const Iterator & in_other ) const
```

Inequality operator.

#### Returns

True if this iterator is not the same as `in_other`; false otherwise.

#### 4.46.3.2 `operator*()`

```
reference rstudio::launcher_plugins::json::Object::Iterator::operator* ( ) const
```

Dereference operator.

#### Returns

A reference to the value this iterator is currently pointing at.

#### 4.46.3.3 `operator++()` [1/2]

```
Iterator& rstudio::launcher_plugins::json::Object::Iterator::operator++ ( )
```

Pre-increment operator.

#### Returns

A reference to this operator, incremented by one position.

#### 4.46.3.4 `operator++()` [2/2]

```
Iterator rstudio::launcher_plugins::json::Object::Iterator::operator++ (
    int )
```

Post-increment operator.

#### Returns

A copy of this operator prior to this increment.



#### 4.46.3.5 operator--() [1/2]

```
Iterator& rstudio::launcher_plugins::json::Object::Iterator::operator-- ( )
```

Pre-decrement operator.

##### Returns

A reference to this operator, decremented by one position.

#### 4.46.3.6 operator--() [2/2]

```
Iterator rstudio::launcher_plugins::json::Object::Iterator::operator-- (
    int )
```

Post-decrement operator.

##### Returns

A copy of this operator prior to this decrement.

#### 4.46.3.7 operator=()

```
Iterator& rstudio::launcher_plugins::json::Object::Iterator::operator= (
    const Iterator & in_other )
```

Assignment operator.

##### Parameters

<i>in_other</i>	The iterator to copy.
-----------------	-----------------------

##### Returns

A reference to this iterator.

#### 4.46.3.8 operator==()

```
bool rstudio::launcher_plugins::json::Object::Iterator::operator== (
    const Iterator & in_other ) const
```

Equality operator.

**Returns**

True if this iterator is the same as `in_other`; false otherwise.

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

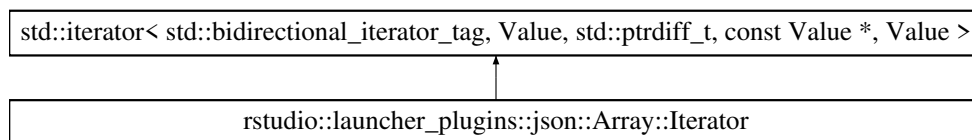
- `/workspaces/rstudio-launcher-plugin-sdk/sdk/include/json/Json.hpp`

## 4.47 rstudio::launcher\_plugins::json::Array::Iterator Class Reference

Class which allows iterating over the elements of a JSON array.

```
#include <Json.hpp>
```

Inheritance diagram for `rstudio::launcher_plugins::json::Array::Iterator`:

**Public Member Functions**

- `Iterator` (`const Array *``in_parent`, `std::ptrdiff_t` `in_startPos`=0)  
*Constructor.*
- `Iterator` (`const Iterator &``in_other`)=default  
*Copy constructor.*
- `Iterator & operator=` (`const Iterator &``in_other`)  
*Assignment operator.*
- `Iterator & operator++` ()  
*Pre-increment operator.*
- `Iterator & operator--` ()  
*Pre-decrement operator.*
- `Iterator operator++` (int)  
*Post-increment operator.*
- `Iterator operator--` (int)  
*Post-decrement operator.*
- `bool operator==` (`const Iterator &``in_other`) const  
*Equality operator.*
- `bool operator!=` (`const Iterator &``in_other`) const  
*Inequality operator.*
- `reference operator*` () const  
*Dereference operator.*

**Friends**

- class `Array`

### 4.47.1 Detailed Description

Class which allows iterating over the elements of a JSON array.

### 4.47.2 Constructor & Destructor Documentation

#### 4.47.2.1 Iterator() [1/2]

```
rstudio::launcher_plugins::json::Array::Iterator::Iterator (
    const Array * in_parent,
    std::ptrdiff_t in_startPos = 0 ) [explicit]
```

Constructor.

##### Parameters

<i>in_parent</i>	The parent array which will be iterated.
<i>in_startPos</i>	The starting position of the iterator. Default: the first member.

#### 4.47.2.2 Iterator() [2/2]

```
rstudio::launcher_plugins::json::Array::Iterator::Iterator (
    const Iterator & in_other ) [default]
```

Copy constructor.

##### Parameters

<i>in_other</i>	The iterator to copy.
-----------------	-----------------------

### 4.47.3 Member Function Documentation

#### 4.47.3.1 operator!=(=)

```
bool rstudio::launcher_plugins::json::Array::Iterator::operator!= (
    const Iterator & in_other ) const
```

Inequality operator.

**Returns**

True if this iterator is not the same as `in_other`; false otherwise.

**4.47.3.2 operator\*()**

```
reference rstudio::launcher_plugins::json::Array::Iterator::operator* ( ) const
```

Dereference operator.

**Returns**

A reference to the value this iterator is currently pointing at.

**4.47.3.3 operator++() [1/2]**

```
Iterator& rstudio::launcher_plugins::json::Array::Iterator::operator++ ( )
```

Pre-increment operator.

**Returns**

A reference to this operator, incremented by one position.

**4.47.3.4 operator++() [2/2]**

```
Iterator rstudio::launcher_plugins::json::Array::Iterator::operator++ (
    int )
```

Post-increment operator.

**Returns**

A copy of this operator prior to this increment.

**4.47.3.5 operator--() [1/2]**

```
Iterator& rstudio::launcher_plugins::json::Array::Iterator::operator-- ( )
```

Pre-decrement operator.

**Returns**

A reference to this operator, decremented by one position.

#### 4.47.3.6 operator--() [2/2]

```
Iterator rstudio::launcher_plugins::json::Array::Iterator::operator-- (
    int )
```

Post-decrement operator.

##### Returns

A copy of this operator prior to this decrement.

#### 4.47.3.7 operator=()

```
Iterator& rstudio::launcher_plugins::json::Array::Iterator::operator= (
    const Iterator & in_other )
```

Assignment operator.

##### Parameters

<i>in_other</i>	The iterator to copy.
-----------------	-----------------------

##### Returns

A reference to this iterator.

#### 4.47.3.8 operator==()

```
bool rstudio::launcher_plugins::json::Array::Iterator::operator== (
    const Iterator & in_other ) const
```

Equality operator.

##### Returns

True if this iterator is the same as *in\_other*; false otherwise.

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/json/Json.hpp

## 4.48 rstudio::launcher\_plugins::api::Job Struct Reference

Structure which represents a job.

```
#include <Job.hpp>
```

## Public Types

- enum [State](#) {  
[State::CANCELED](#), [State::FAILED](#), [State::FINISHED](#), [State::KILLED](#),  
[State::PENDING](#), [State::RUNNING](#), [State::SUSPENDED](#), [State::UNKNOWN](#) }

## Public Member Functions

- [Job](#) ()  
*Constructor.*
- [Job](#) (const [Job](#) &in\_other)  
*Copy constructor.*
- [Job](#) ([Job](#) &&in\_other) noexcept  
*Move constructor.*
- [Job](#) & [operator=](#) (const [Job](#) &in\_other)  
*Assignment operator.*
- [Job](#) & [operator=](#) ([Job](#) &&in\_other) noexcept  
*Move operator.*
- [Optional](#)< std::string > [getJobConfigValue](#) (const std::string &in\_name) const  
*Gets a job configuration value, if it exists.*
- bool [isCompleted](#) () const  
*Checks whether the job has completed (i.e. the job's state is a completed state).*
- bool [matchesTags](#) (const std::set< std::string > &in\_tags) const  
*Checks whether the job has all of the supplied tags.*
- [json::Object](#) [toJson](#) () const  
*Converts this [Job](#) to a JSON object which represents it.*

## Static Public Member Functions

- static [Error](#) [fromJson](#) (const [json::Object](#) &in\_json, [Job](#) &out\_job)  
*Constructs a [Job](#) from a JSON object which represents the job.*
- static [Error](#) [stateFromString](#) (const std::string &in\_statusString, [State](#) &out\_status)  
*Converts a status string into its equivalent [Job::State](#) enum value.*
- static std::string [stateToString](#) ([State](#) in\_status)  
*Converts a [Job::State](#) enum value into its string representation.*

## Public Attributes

- std::vector< std::string > [Arguments](#)
- std::string [Cluster](#)
- std::string [Command](#)  
*The shell command to run.*
- JobConfigList [Config](#)
- [Optional](#)< [Container](#) > [ContainerDetails](#)
- EnvironmentList [Environment](#)
- std::string [Exe](#)  
*The executable to run.*
- [Optional](#)< int > [ExitCode](#)
- ExposedPortList [ExposedPorts](#)
- std::string [Host](#)

- std::string [Id](#)
- [Optional](#)< [system::DateTime](#) > [LastUpdateTime](#)
- MountList [Mounts](#)
- std::string [Name](#)
- [Optional](#)< pid\_t > [Pid](#)
- PlacementConstraintList [PlacementConstraints](#)
- std::set< std::string > [Queues](#)
- ResourceLimitList [ResourceLimits](#)
- std::string [StandardIn](#)
- std::string [StandardErrFile](#)
- std::string [StandardOutFile](#)
- [State](#) [Status](#)
- std::string [StatusMessage](#)
- [system::DateTime](#) [SubmissionTime](#)
- std::set< std::string > [Tags](#)
- [system::User](#) [User](#)
- std::string [WorkingDirectory](#)

## Friends

- class [JobLock](#)

### 4.48.1 Detailed Description

Structure which represents a job.

### 4.48.2 Member Enumeration Documentation

#### 4.48.2.1 State

```
enum rstudio::launcher\_plugins::api::Job::State [strong]
```

##### Enumerator

CANCELED	The job was canceled by the user.
FAILED	The job failed to launch.
FINISHED	The job finished running, successfully or not.
KILLED	The job was killed.
PENDING	The job is queued in the job scheduling system and has not started yet.
RUNNING	The job is currently running.
SUSPENDED	The job has been suspended.
UNKNOWN	The job status is unknown.

### 4.48.3 Constructor & Destructor Documentation

#### 4.48.3.1 Job() [1/2]

```
rstudio::launcher_plugins::api::Job::Job (  
    const Job & in_other )
```

Copy constructor.

##### Parameters

<i>in_other</i>	The job to copy.
-----------------	------------------

#### 4.48.3.2 Job() [2/2]

```
rstudio::launcher_plugins::api::Job::Job (  
    Job && in_other ) [noexcept]
```

Move constructor.

##### Parameters

<i>in_other</i>	The job to move into this job.
-----------------	--------------------------------

### 4.48.4 Member Function Documentation

#### 4.48.4.1 fromJson()

```
static Error rstudio::launcher_plugins::api::Job::fromJson (  
    const json::Object & in_json,  
    Job & out_job ) [static]
```

Constructs a Job from a JSON object which represents the job.

##### Parameters

<i>in_json</i>	The JSON object which represents the job.
<i>out_job</i>	The populated job value. Not valid if an error is returned.



**Returns**

[Success](#) if `in_json` could be parsed as a [Job](#); [Error](#) otherwise.

**4.48.4.2 getJobConfigValue()**

```
Optional<std::string> rstudio::launcher_plugins::api::Job::getJobConfigValue (
    const std::string & in_name ) const
```

Gets a job configuration value, if it exists.

**Parameters**

<i>in_name</i>	The name of the configuration option to retrieve.
----------------	---

**Returns**

The value of the configuration option, if any.

**4.48.4.3 isCompleted()**

```
bool rstudio::launcher_plugins::api::Job::isCompleted ( ) const
```

Checks whether the job has completed (i.e. the job's state is a completed state).

**Returns**

True if the job has completed; false otherwise.

**4.48.4.4 matchesTags()**

```
bool rstudio::launcher_plugins::api::Job::matchesTags (
    const std::set< std::string > & in_tags ) const
```

Checks whether the job has all of the supplied tags.

**Parameters**

<i>in_tags</i>	The desired set of tags to filter jobs by.
----------------	--

**Returns**

True if this job has all of the supplied tags; false otherwise.

**4.48.4.5 operator=()** [1/2]

```
Job& rstudio::launcher_plugins::api::Job::operator= (
    const Job & in_other )
```

Assignment operator.

**Parameters**

<i>in_other</i>	The <a href="#">Job</a> to copy into this <a href="#">Job</a> .
-----------------	---

**Returns**

A reference to this [Job](#).

**4.48.4.6 operator=()** [2/2]

```
Job& rstudio::launcher_plugins::api::Job::operator= (
    Job && in_other ) [noexcept]
```

Move operator.

**Parameters**

<i>in_other</i>	The <a href="#">Job</a> to move into this <a href="#">Job</a> .
-----------------	---

**Returns**

A reference to this [Job](#).

**4.48.4.7 stateFromString()**

```
static Error rstudio::launcher_plugins::api::Job::stateFromString (
    const std::string & in_statusString,
    State & out_status ) [static]
```

Converts a status string into its equivalent [Job::State](#) enum value.

## Parameters

<i>in_statusString</i>	The string to convert.
<i>out_status</i>	The converted status, if no error occurred.

## Returns

[Success](#) if *in\_statusString* is a valid job state; [Error](#) otherwise.

#### 4.48.4.8 stateToString()

```
static std::string rstudio::launcher_plugins::api::Job::stateToString (
    State in_status ) [static]
```

Converts a [Job::State](#) enum value into its string representation.

## Parameters

<i>in_status</i>	The <a href="#">Job::State</a> value to be converted to string.
------------------	---

## Returns

The string representation of the specified [Job::State](#).

#### 4.48.4.9 toJson()

```
json::Object rstudio::launcher_plugins::api::Job::toJson ( ) const
```

Converts this [Job](#) to a JSON object which represents it.

## Returns

The JSON object which represents this [Job](#).

### 4.48.5 Member Data Documentation

#### 4.48.5.1 Arguments

```
std::vector<std::string> rstudio::launcher_plugins::api::Job::Arguments
```

The arguments to supply to the Command or Exe.

#### 4.48.5.2 Cluster

```
std::string rstudio::launcher_plugins::api::Job::Cluster
```

The name of the cluster which should run this job.

#### 4.48.5.3 Command

```
std::string rstudio::launcher_plugins::api::Job::Command
```

The shell command to run.

This should be run using a shell such as `/bin/sh`, as opposed to `Exe`, which should be invoked directly.

Only one of `Command` and `Exe` may be set per job. Jobs which have both set should have been rejected by the Launcher.

#### 4.48.5.4 Config

```
JobConfigList rstudio::launcher_plugins::api::Job::Config
```

The custom job scheduling specific configuration options that were set by the user for this job.

#### 4.48.5.5 ContainerDetails

```
Optional<Container> rstudio::launcher_plugins::api::Job::ContainerDetails
```

The container to run the job in. Only used for containerized jobs.

#### 4.48.5.6 Environment

```
EnvironmentList rstudio::launcher_plugins::api::Job::Environment
```

Environment variables to set on the job's run environment.

#### 4.48.5.7 Exe

```
std::string rstudio::launcher_plugins::api::Job::Exe
```

The executable to run.

This should be invoked directly. It is the user's responsibility to ensure that the provided executable is either fully qualified or on the `PATH` within the job environment.

Only one of `Command` and `Exe` may be set per job. Jobs which have both set should have been rejected by the Launcher.

#### 4.48.5.8 ExitCode

`Optional<int> rstudio::launcher_plugins::api::Job::ExitCode`

The exit code of the job, if applicable.

#### 4.48.5.9 ExposedPorts

`ExposedPortList rstudio::launcher_plugins::api::Job::ExposedPorts`

The ports which were exposed for this job. Only used with containerized jobs.

#### 4.48.5.10 Host

`std::string rstudio::launcher_plugins::api::Job::Host`

The host on which the job was or is being run.

#### 4.48.5.11 Id

`std::string rstudio::launcher_plugins::api::Job::Id`

The unique ID of the job in the scheduling system.

#### 4.48.5.12 LastUpdateTime

`Optional<system::DateTime> rstudio::launcher_plugins::api::Job::LastUpdateTime`

The last time the job was updated.

#### 4.48.5.13 Mounts

`MountList rstudio::launcher_plugins::api::Job::Mounts`

The file system mounts to set when launching this job.

#### 4.48.5.14 Name

`std::string rstudio::launcher_plugins::api::Job::Name`

The name of the job.

#### 4.48.5.15 Pid

`Optional<pid_t> rstudio::launcher_plugins::api::Job::Pid`

The PID of the job, if applicable.

#### 4.48.5.16 PlacementConstraints

`PlacementConstraintList rstudio::launcher_plugins::api::Job::PlacementConstraints`

Custom placement constraints for the job scheduling system that were set by the user for this job.

#### 4.48.5.17 Queues

`std::set<std::string> rstudio::launcher_plugins::api::Job::Queues`

The set of queues on which this job may be run, or the queue which ran the job.

#### 4.48.5.18 ResourceLimits

`ResourceLimitList rstudio::launcher_plugins::api::Job::ResourceLimits`

The resource limits that were set by the user for this job.

#### 4.48.5.19 StandardErrFile

`std::string rstudio::launcher_plugins::api::Job::StandardErrFile`

The file to which the job's standard error output was written.

#### 4.48.5.20 StandardIn

`std::string rstudio::launcher_plugins::api::Job::StandardIn`

Data which should be supplied to the job via standard in.

#### 4.48.5.21 StandardOutFile

`std::string rstudio::launcher_plugins::api::Job::StandardOutFile`

The file to which the job's standard output was written.

#### 4.48.5.22 Status

`State rstudio::launcher_plugins::api::Job::Status`

The status of the job.

#### 4.48.5.23 StatusMessage

`std::string rstudio::launcher_plugins::api::Job::StatusMessage`

The reason for the status, if any.

#### 4.48.5.24 SubmissionTime

`system::DateTime` `rstudio::launcher_plugins::api::Job::SubmissionTime`

The time at which the job was submitted to the job scheduling system.

#### 4.48.5.25 Tags

`std::set<std::string>` `rstudio::launcher_plugins::api::Job::Tags`

The tags which were set on the job by the user. Can be used for filtering jobs based on tags.

#### 4.48.5.26 User

`system::User` `rstudio::launcher_plugins::api::Job::User`

The user who ran the job.

#### 4.48.5.27 WorkingDirectory

`std::string` `rstudio::launcher_plugins::api::Job::WorkingDirectory`

The working directory from which to run the job.

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

- `/workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Job.hpp`

## 4.49 rstudio::launcher\_plugins::api::JobConfig Struct Reference

Struct which represents a custom configuration setting for jobs launched with a given Plugin.

```
#include <Job.hpp>
```

### Public Types

- enum `Type` { `Type::ENUM`, `Type::FLOAT`, `Type::INT`, `Type::STRING` }

### Public Member Functions

- `JobConfig` ()=default  
*Default constructor.*
- `JobConfig` (std::string in\_name, `Type` in\_type)  
*Constructor.*
- `json::Object toJson` () const  
*Converts this `JobConfig` to a JSON object which represents it.*

## Static Public Member Functions

- static [Error fromJson](#) (const [json::Object](#) &in\_json, [JobConfig](#) &out\_jobConfig)  
*Constructs a [JobConfig](#) from a JSON object which represents the job config.*

## Public Attributes

- std::string [Name](#)
- [Optional](#)< [Type](#) > [ValueType](#)
- std::string [Value](#)

### 4.49.1 Detailed Description

Struct which represents a custom configuration setting for jobs launched with a given Plugin.

[JobConfig](#) values should be used only when there is a necessary per-job configuration that cannot be covered by another aspect of the [Job](#) structure, such as a [ResourceLimit](#) or [PlacementConstraint](#).

### 4.49.2 Member Enumeration Documentation

#### 4.49.2.1 Type

```
enum rstudio::launcher\_plugins::api::JobConfig::Type [strong]
```

Enumerator

ENUM	Enumeration type.
FLOAT	Floating point value type.
INT	Integer type.
STRING	String type.

### 4.49.3 Constructor & Destructor Documentation

#### 4.49.3.1 JobConfig()

```
rstudio::launcher\_plugins::api::JobConfig::JobConfig (  
    std::string in_name,  
    Type in_type )
```

Constructor.



## Parameters

<i>in_name</i>	The name of the custom job configuration value.
<i>in_type</i>	The type of the custom job configuration value.

## 4.49.4 Member Function Documentation

### 4.49.4.1 fromJson()

```
static Error rstudio::launcher_plugins::api::JobConfig::fromJson (
    const json::Object & in_json,
    JobConfig & out_jobConfig ) [static]
```

Constructs a [JobConfig](#) from a JSON object which represents the job config.

## Parameters

<i>in_json</i>	The JSON object which represents the job config.
<i>out_jobConfig</i>	The populated job config value. Not valid if an error is returned.

## Returns

[Success](#) if in\_json could be parsed as a [JobConfig](#); [Error](#) otherwise.

### 4.49.4.2 toJson()

```
json::Object rstudio::launcher_plugins::api::JobConfig::toJson ( ) const
```

Converts this [JobConfig](#) to a JSON object which represents it.

## Returns

The JSON object which represents this [JobConfig](#).

## 4.49.5 Member Data Documentation

### 4.49.5.1 Name

```
std::string rstudio::launcher_plugins::api::JobConfig::Name
```

The name of the custom job configuration value.

#### 4.49.5.2 Value

```
std::string rstudio::launcher_plugins::api::JobConfig::Value
```

The value of the custom job configuration value.

#### 4.49.5.3 ValueType

```
Optional<Type> rstudio::launcher_plugins::api::JobConfig::ValueType
```

The type of the custom job configuration value.

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

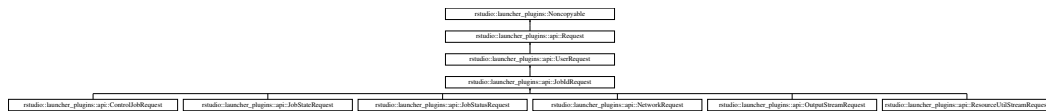
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Job.hpp

## 4.50 rstudio::launcher\_plugins::api::JobIdRequest Class Reference

Base class which should be used for requests that require a [Job](#) ID.

```
#include <Request.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::JobIdRequest:



### Public Member Functions

- const std::string & [getJobId](#) () const  
*Gets the ID of the job for which this request was made.*
- const std::string & [getEncodedJobId](#) () const  
*Gets the ID of the job for which this request was made.*

### Protected Member Functions

- [JobIdRequest](#) ([Request::Type](#) in\_type, const [json::Object](#) &in\_requestJson)  
*Constructor.*

### Additional Inherited Members

#### 4.50.1 Detailed Description

Base class which should be used for requests that require a [Job](#) ID.

## 4.50.2 Constructor & Destructor Documentation

### 4.50.2.1 JobIdRequest()

```
rstudio::launcher_plugins::api::JobIdRequest::JobIdRequest (
    Request::Type in_type,
    const json::Object & in_requestJson ) [protected]
```

Constructor.

## Parameters

<i>in_type</i>	The type of the user request.
<i>in_requestJson</i>	The JSON Object which represents the job ID request.

### 4.50.3 Member Function Documentation

#### 4.50.3.1 getEncodedJobId()

```
const std::string& rstudio::launcher_plugins::api::JobIdRequest::getEncodedJobId ( ) const
```

Gets the ID of the job for which this request was made.

## Returns

The ID of the job for which this request was made.

#### 4.50.3.2 getJobId()

```
const std::string& rstudio::launcher_plugins::api::JobIdRequest::getJobId ( ) const
```

Gets the ID of the job for which this request was made.

## Returns

The ID of the job for which this request was made.

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

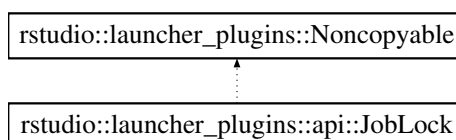
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Request.hpp

## 4.51 rstudio::launcher\_plugins::api::JobLock Class Reference

RAII class for locking access to a [Job](#) object. Should be used every time a [Job](#) is modified.

```
#include <Job.hpp>
```

Inheritance diagram for `rstudio::launcher_plugins::api::JobLock`:



## Public Member Functions

- [JobLock](#) (JobPtr in\_job)  
*Constructor.*
- [JobLock](#) (ConstJobPtr in\_job)  
*Constructor.*

### 4.51.1 Detailed Description

RAII class for locking access to a [Job](#) object. Should be used every time a [Job](#) is modified.

### 4.51.2 Constructor & Destructor Documentation

#### 4.51.2.1 JobLock() [1/2]

```
rstudio::launcher_plugins::api::JobLock::JobLock (
    JobPtr in_job ) [explicit]
```

Constructor.

May throw a `std::system_error`.

##### Parameters

<code>in_job</code>	The job to lock.
---------------------	------------------

#### 4.51.2.2 JobLock() [2/2]

```
rstudio::launcher_plugins::api::JobLock::JobLock (
    ConstJobPtr in_job ) [explicit]
```

Constructor.

May throw a `std::system_error`.

##### Parameters

<code>in_job</code>	The job to lock.
---------------------	------------------

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

- `/workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Job.hpp`

## 4.52 rstudio::launcher\_plugins::api::JobSourceConfiguration Struct Reference

Describes the capabilities and configuration of this [Job](#) Source.

```
#include <IJobSource.hpp>
```

### Public Attributes

- [ContainerConfiguration](#) ContainerConfig
- JobConfigList [CustomConfig](#)
- PlacementConstraintList [PlacementConstraints](#)
- std::set< std::string > [Queues](#)
- ResourceLimitList [ResourceLimits](#)

### 4.52.1 Detailed Description

Describes the capabilities and configuration of this [Job](#) Source.

### 4.52.2 Member Data Documentation

#### 4.52.2.1 ContainerConfig

```
ContainerConfiguration rstudio::launcher_plugins::api::JobSourceConfiguration::ContainerConfig
```

The container configuration of this [Job](#) Source.

#### 4.52.2.2 CustomConfig

```
JobConfigList rstudio::launcher_plugins::api::JobSourceConfiguration::CustomConfig
```

The custom configuration values supported by this [Job](#) Source.

#### 4.52.2.3 PlacementConstraints

```
PlacementConstraintList rstudio::launcher_plugins::api::JobSourceConfiguration::Placement↔  
Constraints
```

The set of job placement constraints which may be set when launching a job.

#### 4.52.2.4 Queues

```
std::set<std::string> rstudio::launcher_plugins::api::JobSourceConfiguration::Queues
```

The set of queues on which jobs may be run.

#### 4.52.2.5 ResourceLimits

```
ResourceLimitList rstudio::launcher_plugins::api::JobSourceConfiguration::ResourceLimits
```

The set of resource limit types, optionally with maximum and default values, which user may set when launching a job.

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

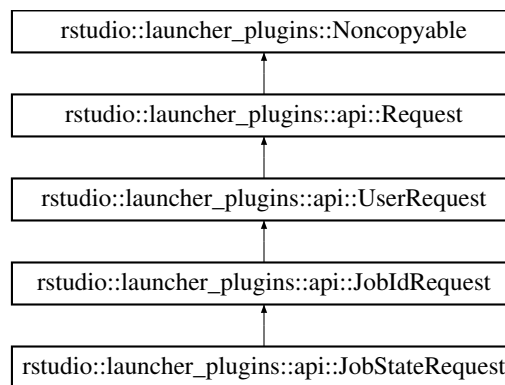
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/IJobSource.hpp

## 4.53 rstudio::launcher\_plugins::api::JobStateRequest Class Reference

Represents a job state request received from the Launcher.

```
#include <Request.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::JobStateRequest:



### Public Member Functions

- [Error getEndTime](#) (Optional< [system::DateTime](#) > &out\_endTime) const  
*Gets the end of the date range for this request.*
- const [Optional](#)< std::set< std::string > > & [getFieldSet](#) () const  
*Gets the set of [Job](#) fields which should be included in the response.*
- [Error getStartTime](#) (Optional< [system::DateTime](#) > &out\_endTime) const  
*Gets the start of the date range for this request.*
- [Error getStatusSet](#) (Optional< std::set< [Job::State](#) > > &out\_statuses) const  
*Gets the set of [Job](#) statuses by which to filter the returned list of jobs.*
- const [Optional](#)< std::set< std::string > > & [getTagSet](#) () const  
*Gets the set of [Job](#) tags by which to filter the returned list of jobs.*

## Friends

- class **Request**

## Additional Inherited Members

### 4.53.1 Detailed Description

Represents a job state request received from the Launcher.

### 4.53.2 Member Function Documentation

#### 4.53.2.1 `getEndTime()`

```
Error rstudio::launcher_plugins::api::JobStateRequest::getEndTime (
    Optional< system::DateTime > & out_endTime ) const
```

Gets the end of the date range for this request.

If this value is set, only jobs which were submitted before this DateTime should be returned in the response.

#### Parameters

<i>out_endTime</i>	The end time, if it was set and the string value could be parsed as a DateTime correctly.
--------------------	---

#### Returns

**Success** if the value was set and could be parsed correctly, or if the value was not set; **Error** otherwise.

#### 4.53.2.2 `getFieldSet()`

```
const Optional<std::set<std::string> >& rstudio::launcher_plugins::api::JobStateRequest↵
::getFieldSet ( ) const
```

Gets the set of **Job** fields which should be included in the response.

If this value is set, only the fields which are included in this set should be returned in the response. ID will always be returned, as it is required.

#### Returns

The optional set of **Job** fields to include in the response.



### 4.53.2.3 getStartTime()

```
Error rstudio::launcher_plugins::api::JobStateRequest::getStartTime (
    Optional< system::DateTime > & out_endTime ) const
```

Gets the start of the date range for this request.

If this value is set, only jobs which were submitted after this DateTime should be returned in the response.

#### Parameters

<i>out_startTime</i>	The start time, if it was set and the string value could be parsed as a DateTime correctly.
----------------------	---

#### Returns

[Success](#) if the value was set and could be parsed correctly, or if the value was not set; [Error](#) otherwise.

### 4.53.2.4 getStatusSet()

```
Error rstudio::launcher_plugins::api::JobStateRequest::getStatusSet (
    Optional< std::set< Job::State > > & out_statuses ) const
```

Gets the set of [Job](#) statuses by which to filter the returned list of jobs.

If this value is set, only the jobs which have one of the specified states should be returned in the response.

#### Parameters

<i>out_statuses</i>	The set of statuses to filter by, if any were set and they could all be parsed as <a href="#">Job::State</a> values correctly.
---------------------	--

#### Returns

[Success](#) if the value was set and could be parsed correctly, or if the value was not set; [Error](#) otherwise.

### 4.53.2.5 getTagSet()

```
const Optional<std::set<std::string> >& rstudio::launcher_plugins::api::JobStateRequest↵
::getTagSet ( ) const
```

Gets the set of [Job](#) tags by which to filter the returned list of jobs.

If this value is set, only the jobs which have one of the specified states should be returned in the response.

#### Returns

The optional set of [Job](#) statuses by which to filter the returned list of jobs.

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

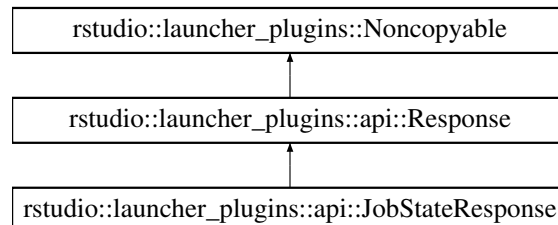
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Request.hpp

## 4.54 rstudio::launcher\_plugins::api::JobStateResponse Class Reference

Class which represents a job state response which can be sent to the Launcher in response to a get or submit job request.

```
#include <Response.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::JobStateResponse:



### Public Member Functions

- **JobStateResponse** (uint64\_t in\_requestId, JobList in\_jobs, **Optional**< std::set< std::string > > in\_jobFields=**Optional**< std::set< std::string > >())  
*Constructor.*
- **json::Object toJson** () const override  
*Converts this job state response to a JSON object.*

### Additional Inherited Members

#### 4.54.1 Detailed Description

Class which represents a job state response which can be sent to the Launcher in response to a get or submit job request.

#### 4.54.2 Constructor & Destructor Documentation

##### 4.54.2.1 JobStateResponse()

```
rstudio::launcher_plugins::api::JobStateResponse::JobStateResponse (
    uint64_t in_requestId,
    JobList in_jobs,
    Optional< std::set< std::string > > in_jobFields = Optional< std::set< std::string > >() )
```

Constructor.

## Parameters

<i>in_requestId</i>	The ID of the request for which this job state response is being sent.
<i>in_jobs</i>	The jobs to be returned to the Launcher.
<i>in_jobFields</i>	The optional set of job fields to include for each job. If this is not set, all fields will be returned.

### 4.54.3 Member Function Documentation

#### 4.54.3.1 toJson()

```
json::Object rstudio::launcher_plugins::api::JobStateResponse::toJson ( ) const [override],
[virtual]
```

Converts this job state response to a JSON object.

## Returns

The JSON object which represents this job state response.

Reimplemented from [rstudio::launcher\\_plugins::api::Response](#).

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

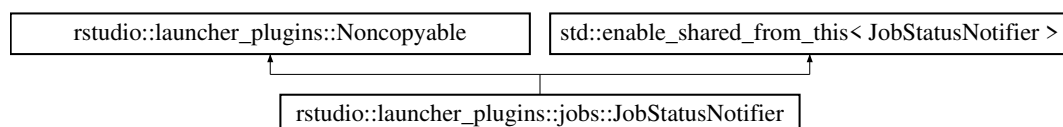
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Response.hpp

## 4.55 rstudio::launcher\_plugins::jobs::JobStatusNotifier Class Reference

Class which notifies subscribers when a job updates.

```
#include <JobStatusNotifier.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::jobs::JobStatusNotifier:



## Public Member Functions

- [JobStatusNotifier](#) ()  
*Constructor.*
- SubscriptionHandle [subscribe](#) (const OnJobStatusUpdate &in\_onJobStatusUpdate)  
*Subscribes to all jobs.*
- SubscriptionHandle [subscribe](#) (const std::string &in\_jobId, const OnJobStatusUpdate &in\_onJobStatusUpdate)  
*Subscribes to a specific job.*
- void [updateJob](#) (const api::JobPtr &in\_job, [api::Job::State](#) in\_newStatus, const std::string &in\_statusMessage="", const [system::DateTime](#) &in\_invocationTime=[system::DateTime\(\)](#))  
*Updates the status of a job with a new status and optionally a new status message.*

## Friends

- class **Subscription**

### 4.55.1 Detailed Description

Class which notifies subscribers when a job updates.

### 4.55.2 Member Function Documentation

#### 4.55.2.1 [subscribe\(\)](#) [1/2]

```
SubscriptionHandle rstudio::launcher_plugins::jobs::JobStatusNotifier::subscribe (
    const OnJobStatusUpdate & in_onJobStatusUpdate )
```

Subscribes to all jobs.

#### Parameters

<i>in_onJobStatusUpdate</i>	The function to be invoked when any job is updated.
-----------------------------	---

#### Returns

A handle to the job subscription. To end the subscription, allow the handle to fall out of scope.

#### 4.55.2.2 [subscribe\(\)](#) [2/2]

```
SubscriptionHandle rstudio::launcher_plugins::jobs::JobStatusNotifier::subscribe (
    const std::string & in_jobId,
    const OnJobStatusUpdate & in_onJobStatusUpdate )
```

Subscribes to a specific job.

## Parameters

<i>in_jobId</i>	The ID of the job to subscribe to.
<i>in_onJobStatusUpdate</i>	The function to be invoked when the job is updated.

## Returns

A handle to the job subscription. To end the subscription, allow the handle to fall out of scope.

**4.55.2.3 updateJob()**

```
void rstudio::launcher_plugins::jobs::JobStatusNotifier::updateJob (
    const api::JobPtr & in_job,
    api::Job::State in_newStatus,
    const std::string & in_statusMessage = "",
    const system::DateTime & in_invocationTime = system::DateTime() )
```

Updates the status of a job with a new status and optionally a new status message.

## Parameters

<i>in_job</i>	The job to be updated.
<i>in_newStatus</i>	The new status of the job.
<i>in_statusMessage</i>	The new status message of the job. Default: "".
<i>in_invocationTime</i>	The time at which this method was invoked. Default: Current Time. If there is concern about time differences between the RStudio Launcher Host and the Job Scheduling System, this may be overridden with the time from the point of view of the Job Scheduling System.

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

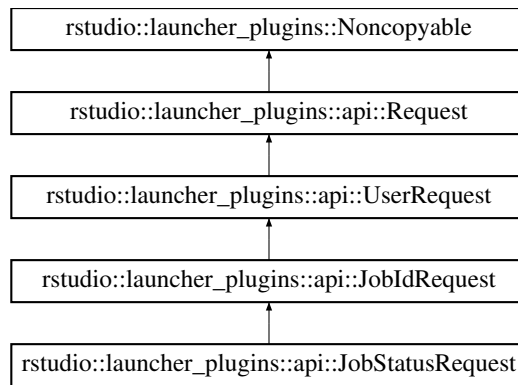
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/jobs/JobStatusNotifier.hpp

**4.56 rstudio::launcher\_plugins::api::JobStatusRequest Class Reference**

[Request](#) from the launcher to begin or end a [Job](#) Status Stream.

```
#include <Request.hpp>
```

Inheritance diagram for `rstudio::launcher_plugins::api::JobStatusRequest`:



## Public Member Functions

- bool `isCancelRequest` () const  
*Gets whether the [Job](#) Status Stream should be started (false) or ended (true).*

## Friends

- class `Request`

## Additional Inherited Members

### 4.56.1 Detailed Description

[Request](#) from the launcher to begin or end a [Job](#) Status Stream.

### 4.56.2 Member Function Documentation

#### 4.56.2.1 isCancelRequest()

```
bool rstudio::launcher_plugins::api::JobStatusRequest::isCancelRequest ( ) const
```

Gets whether the [Job](#) Status Stream should be started (false) or ended (true).

#### Returns

True if the stream should be canceled; false if it should be started.

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

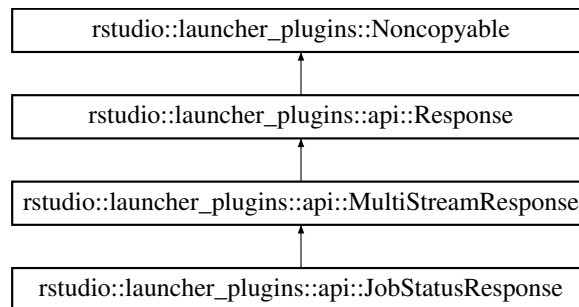
- `/workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Request.hpp`

## 4.57 rstudio::launcher\_plugins::api::JobStatusResponse Class Reference

Class which represents a [Job](#) Status Stream, either for all jobs or for a specific job.

```
#include <Response.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::JobStatusResponse:



### Public Member Functions

- [JobStatusResponse](#) (StreamSequences in\_sequences, const JobPtr &in\_job)  
*Constructor.*
- [json::Object toJson](#) () const override  
*Converts this job status response to a JSON object.*

### Additional Inherited Members

#### 4.57.1 Detailed Description

Class which represents a [Job](#) Status Stream, either for all jobs or for a specific job.

#### 4.57.2 Constructor & Destructor Documentation

##### 4.57.2.1 JobStatusResponse()

```
rstudio::launcher_plugins::api::JobStatusResponse::JobStatusResponse (
    StreamSequences in_sequences,
    const JobPtr & in_job )
```

Constructor.

##### Parameters

<i>in_sequences</i>	The stream sequences for which this response will be sent.
<i>in_job</i>	The job that was updated.



### 4.57.3 Member Function Documentation

#### 4.57.3.1 toJson()

```
json::Object rstudio::launcher_plugins::api::JobStatusResponse::toJson ( ) const [override],
[virtual]
```

Converts this job status response to a JSON object.

#### Returns

The JSON object which represents this job status response.

Reimplemented from [rstudio::launcher\\_plugins::api::MultiStreamResponse](#).

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

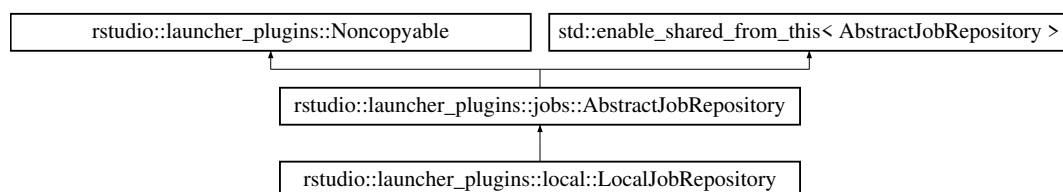
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Response.hpp

## 4.58 rstudio::launcher\_plugins::local::LocalJobRepository Class Reference

Responsible for job persistence.

```
#include <LocalJobRepository.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::local::LocalJobRepository:



### Public Member Functions

- [LocalJobRepository](#) (const std::string &in\_hostname, jobs::JobStatusNotifierPtr in\_notifier)  
*Constructor.*
- void [saveJob](#) (api::JobPtr in\_job) const  
*Saves a job to disk.*
- [Error setJobOutputPaths](#) (api::JobPtr io\_job) const  
*Sets the default output paths for the specified job.*

### 4.58.1 Detailed Description

Responsible for job persistence.

### 4.58.2 Constructor & Destructor Documentation

#### 4.58.2.1 LocalJobRepository()

```
rstudio::launcher_plugins::local::LocalJobRepository::LocalJobRepository (
    const std::string & in_hostname,
    jobs::JobStatusNotifierPtr in_notifier )
```

Constructor.

##### Parameters

<i>in_hostname</i>	The hostname of machine which is hosting this instance of the Local Plugin.
<i>in_notifier</i>	The job status notifier from which to receive job status update notifications.

### 4.58.3 Member Function Documentation

#### 4.58.3.1 saveJob()

```
void rstudio::launcher_plugins::local::LocalJobRepository::saveJob (
    api::JobPtr in_job ) const
```

Saves a job to disk.

##### Parameters

<i>in_job</i>	The job to be saved.
---------------	----------------------

#### 4.58.3.2 setJobOutputPaths()

```
Error rstudio::launcher_plugins::local::LocalJobRepository::setJobOutputPaths (
    api::JobPtr io_job ) const
```

Sets the default output paths for the specified job.

## Parameters

<code>io_job</code>	The job to modify.
---------------------	--------------------

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

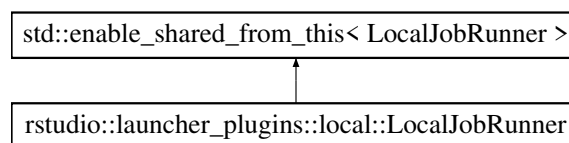
- /workspaces/rstudio-launcher-plugin-sdk/plugins/Local/include/LocalJobRepository.hpp

## 4.59 rstudio::launcher\_plugins::local::LocalJobRunner Class Reference

Runs jobs on the local machine.

```
#include <LocalJobRunner.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::local::LocalJobRunner:



### Public Member Functions

- [LocalJobRunner](#) (const std::string &in\_hostname, jobs::JobStatusNotifierPtr in\_notifier, std::shared\_ptr< [LocalJobRepository](#) > in\_jobRepository)  
*Constructor.*
- [Error initialize](#) ()  
*Initializes the job runner.*
- [Error runJob](#) (api::JobPtr &io\_job, bool &out\_wasInvalidJob)  
*Runs the specified job.*

#### 4.59.1 Detailed Description

Runs jobs on the local machine.

#### 4.59.2 Constructor & Destructor Documentation

##### 4.59.2.1 LocalJobRunner()

```
rstudio::launcher_plugins::local::LocalJobRunner::LocalJobRunner (
    const std::string & in_hostname,
    jobs::JobStatusNotifierPtr in_notifier,
    std::shared_ptr< LocalJobRepository > in_jobRepository )
```

Constructor.

## Parameters

<i>in_hostname</i>	The hostname of the machine on which jobs will be run (this machine).
<i>in_notifier</i>	The job status notifier, for posting job status updates.
<i>in_jobRepository</i>	The job repository, for saving jobs and job output.

## 4.59.3 Member Function Documentation

### 4.59.3.1 initialize()

```
Error rstudio::launcher_plugins::local::LocalJobRunner::initialize ( )
```

Initializes the job runner.

## Returns

**Success** if the job runner could be initialized; **Error** otherwise.

### 4.59.3.2 runJob()

```
Error rstudio::launcher_plugins::local::LocalJobRunner::runJob (
    api::JobPtr & io_job,
    bool & out_wasInvalidJob )
```

Runs the specified job.

## Parameters

<i>io_job</i>	The Job to be run.
<i>out_wasInvalidJob</i>	Whether the error that occurred was because the requested Job was invalid.

## Returns

**Success** if the job could be run; **Error** otherwise.

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

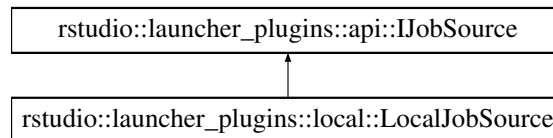
- /workspaces/rstudio-launcher-plugin-sdk/plugins/Local/include/LocalJobRunner.hpp

## 4.60 rstudio::launcher\_plugins::local::LocalJobSource Class Reference

Class which is responsible for running and retrieving information about jobs on the Local system.

```
#include <LocalJobSource.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::local::LocalJobSource:



### Public Member Functions

- [LocalJobSource](#) (std::string in\_hostname, jobs::JobStatusNotifierPtr in\_jobStatusNotifier, std::shared\_ptr< [LocalJobRepository](#) > in\_jobRepository)  
*Constructor.*
- [Error initialize](#) () override  
*Initializes the Local Job Source.*
- bool [cancelJob](#) (api::JobPtr in\_job, bool &out\_isComplete, std::string &out\_statusMessage) override  
*Cancels a pending job.*
- [Error getConfiguration](#) (const [system::User](#) &, [api::JobSourceConfiguration](#) &out\_configuration) const override  
*Gets the configuration and capabilities of the Local Job Source.*
- [Error getNetworkInfo](#) (api::JobPtr in\_job, [api::NetworkInfo](#) &out\_networkInfo) const override  
*Gets the network information for the specified job.*
- bool [killJob](#) (api::JobPtr in\_job, bool &out\_isComplete, std::string &out\_statusMessage) override  
*Forcibly kills a running job.*
- bool [resumeJob](#) (api::JobPtr in\_job, bool &out\_isComplete, std::string &out\_statusMessage) override  
*Resumes a suspended job.*
- bool [stopJob](#) (api::JobPtr in\_job, bool &out\_isComplete, std::string &out\_statusMessage) override  
*Stops a running job.*
- bool [suspendJob](#) (api::JobPtr in\_job, bool &out\_isComplete, std::string &out\_statusMessage) override  
*Suspends a running job.*
- [Error submitJob](#) (api::JobPtr io\_job, bool &out\_wasInvalidRequest) const override  
*Runs a job on the local instance.*
- [Error createOutputStream](#) (api::OutputType in\_outputType, api::JobPtr in\_job, api::AbstractOutputStream↵::OnOutput in\_onOutput, [api::AbstractOutputStream::OnComplete](#) in\_onComplete, api::AbstractOutput↵Stream::OnError in\_onError, api::OutputStreamPtr &out\_outputStream) override  
*Creates a file output stream for the specified job.*
- [Error createResourceStream](#) (api::ConstJobPtr in\_job, comms::AbstractLauncherCommunicatorPtr in\_↵launcherCommunicator, api::AbstractResourceStreamPtr &out\_resourceStream) override  
*Creates a resource utilization metric stream for the specified job.*

### Additional Inherited Members

#### 4.60.1 Detailed Description

Class which is responsible for running and retrieving information about jobs on the Local system.

## 4.60.2 Constructor & Destructor Documentation

### 4.60.2.1 LocalJobSource()

```
rstudio::launcher_plugins::local::LocalJobSource::LocalJobSource (
    std::string in_hostname,
    jobs::JobStatusNotifierPtr in_jobStatusNotifier,
    std::shared_ptr< LocalJobRepository > in_jobRepository )
```

Constructor.

#### Parameters

<i>in_hostname</i>	The name of the host running this instance of the Local Plugin.
<i>in_jobStatusNotifier</i>	The job status notifier to which to post or from which to receive job status updates.
<i>in_jobRepository</i>	The job repository, from which to look up jobs.

## 4.60.3 Member Function Documentation

### 4.60.3.1 cancelJob()

```
bool rstudio::launcher_plugins::local::LocalJobSource::cancelJob (
    api::JobPtr in_job,
    bool & out_isComplete,
    std::string & out_statusMessage ) [override], [virtual]
```

Cancels a pending job.

This method will not be invoked unless the job is currently pending. The Job lock will be held when this method is invoked.

#### Parameters

<i>in_job</i>	The job to be canceled.
<i>out_isComplete</i>	Whether the cancel operation completed successfully (true) or not (false).
<i>out_statusMessage</i>	The status message of the cancel operation, if any.

#### Returns

False if the cancel operation is not supported; true otherwise.

Implements [rstudio::launcher\\_plugins::api::IJobSource](#).

### 4.60.3.2 createOutputStream()

```
Error rstudio::launcher_plugins::local::LocalJobSource::createOutputStream (
    api::OutputType in_outputType,
    api::JobPtr in_job,
    api::AbstractOutputStream::OnOutput in_onOutput,
    api::AbstractOutputStream::OnComplete in_onComplete,
    api::AbstractOutputStream::OnError in_onError,
    api::OutputStreamPtr & out_outputStream ) [override], [virtual]
```

Creates a file output stream for the specified job.

#### Parameters

<i>in_outputType</i>	The type of job output to stream.
<i>in_job</i>	The job for which output should be streamed.
<i>in_onOutput</i>	Callback function which will be invoked when data is reported.
<i>in_onComplete</i>	Callback function which will be invoked when the stream is complete.
<i>in_onError</i>	Callback function which will be invoked if an error occurs.
<i>out_outputStream</i>	The newly created output stream, on <a href="#">Success</a> .

#### Returns

[Success](#) if the output stream could be created; [Error](#) otherwise.

Implements [rstudio::launcher\\_plugins::api::IJobSource](#).

### 4.60.3.3 createResourceStream()

```
Error rstudio::launcher_plugins::local::LocalJobSource::createResourceStream (
    api::ConstJobPtr in_job,
    comms::AbstractLauncherCommunicatorPtr in_launcherCommunicator,
    api::AbstractResourceStreamPtr & out_resourceStream ) [override], [virtual]
```

Creates a resource utilization metric stream for the specified job.

#### Parameters

<i>in_job</i>	The job for which resource utilization metrics should be streamed.
<i>in_launcherCommunicator</i>	The communicator with which to send responses to the Launcher.
<i>out_resourceStream</i>	The newly created resource utilization metric stream, on <a href="#">Success</a> .

#### Returns

Success if the stream could be created; the [Error](#) that occurred otherwise.

Implements [rstudio::launcher\\_plugins::api::IJobSource](#).

#### 4.60.3.4 getConfiguration()

```
Error rstudio::launcher_plugins::local::LocalJobSource::getConfiguration (
    const system::User & ,
    api::JobSourceConfiguration & out_configuration ) const [override], [virtual]
```

Gets the configuration and capabilities of the Local Job Source.

The Local Job Source only has two custom configuration values. It does not support resource limits, placement constraints, queues, or containers.

##### Parameters

<i>out_configuration</i>	The configuration and capabilities of this Job Source, for the specified user.
--------------------------	--

##### Returns

[Success](#) if the configuration and capabilities for this Job Source could be populated; [Error](#) otherwise.

Implements [rstudio::launcher\\_plugins::api::IJobSource](#).

#### 4.60.3.5 getNetworkInfo()

```
Error rstudio::launcher_plugins::local::LocalJobSource::getNetworkInfo (
    api::JobPtr in_job,
    api::NetworkInfo & out_networkInfo ) const [override], [virtual]
```

Gets the network information for the specified job.

##### Parameters

<i>in_job</i>	The job for which to retrieve network information.
<i>out_networkInfo</i>	The network information of the specified job, if no error occurred.

##### Returns

[Success](#) if the network information could be retrieved; [Error](#) otherwise.

Implements [rstudio::launcher\\_plugins::api::IJobSource](#).

#### 4.60.3.6 initialize()

```
Error rstudio::launcher_plugins::local::LocalJobSource::initialize ( ) [override], [virtual]
```

Initializes the Local Job Source.

This function initializes the file-based job storage and communications with other Local plugins which are part of this Launcher cluster.



### Returns

[Success](#) if the job source could be initialized; [Error](#) otherwise.

Implements [rstudio::launcher\\_plugins::api::IJobSource](#).

#### 4.60.3.7 killJob()

```
bool rstudio::launcher_plugins::local::LocalJobSource::killJob (
    api::JobPtr in_job,
    bool & out_isComplete,
    std::string & out_statusMessage ) [override], [virtual]
```

Forcibly kills a running job.

This method should perform the equivalent of sending a SIGKILL to a process. This method will not be invoked unless the job is currently running. The Job lock will be held when this method is invoked.

### Parameters

<i>in_job</i>	The job to be canceled.
<i>out_isComplete</i>	Whether the kill operation completed successfully (true) or not (false).
<i>out_statusMessage</i>	The status message of the kill operation, if any.

### Returns

False if the kill operation is not supported; true otherwise.

Implements [rstudio::launcher\\_plugins::api::IJobSource](#).

#### 4.60.3.8 resumeJob()

```
bool rstudio::launcher_plugins::local::LocalJobSource::resumeJob (
    api::JobPtr in_job,
    bool & out_isComplete,
    std::string & out_statusMessage ) [override], [virtual]
```

Resumes a suspended job.

This method should perform the equivalent of sending a SIGCONT to a process. This method will not be invoked unless the job is currently suspended. The Job lock will be held when this method is invoked.

### Parameters

<i>in_job</i>	The job to be canceled.
<i>out_isComplete</i>	Whether the resume operation completed successfully (true) or not (false).
<i>out_statusMessage</i>	The status message of the resume operation, if any.

**Returns**

False if the resume operation is not supported; true otherwise.

Implements [rstudio::launcher\\_plugins::api::IJobSource](#).

**4.60.3.9 stopJob()**

```
bool rstudio::launcher_plugins::local::LocalJobSource::stopJob (
    api::JobPtr in_job,
    bool & out_isComplete,
    std::string & out_statusMessage ) [override], [virtual]
```

Stops a running job.

This method should perform the equivalent of sending a SIGTERM to a process. This method will not be invoked unless the job is currently running. The Job lock will be held when this method is invoked.

**Parameters**

<i>in_job</i>	The job to be canceled.
<i>out_isComplete</i>	Whether the stop operation completed successfully (true) or not (false).
<i>out_statusMessage</i>	The status message of the stop operation, if any.

**Returns**

False if the stop operation is not supported; true otherwise.

Implements [rstudio::launcher\\_plugins::api::IJobSource](#).

**4.60.3.10 submitJob()**

```
Error rstudio::launcher_plugins::local::LocalJobSource::submitJob (
    api::JobPtr io_job,
    bool & out_wasInvalidRequest ) const [override], [virtual]
```

Runs a job on the local instance.

**Parameters**

<i>io_job</i>	The Job to be submitted.
<i>out_wasInvalidRequest</i>	Whether the requested Job was invalid, based on the features supported by the Job Scheduling System.

**Returns**

[Success](#) if the job could be submitted to the Job Scheduling System; [Error](#) otherwise.

Implements [rstudio::launcher\\_plugins::api::IJobSource](#).

**4.60.3.11 suspendJob()**

```
bool rstudio::launcher_plugins::local::LocalJobSource::suspendJob (
    api::JobPtr in_job,
    bool & out_isComplete,
    std::string & out_statusMessage ) [override], [virtual]
```

Suspends a running job.

This method should perform the equivalent of sending a SIGSTOP to a process. A suspended job should be able to be resumed at a later time. This method will not be invoked unless the job is currently running. The Job lock will be held when this method is invoked.

**Parameters**

<i>in_job</i>	The job to be suspended.
<i>out_isComplete</i>	Whether the suspend operation completed successfully (true) or not (false).
<i>out_statusMessage</i>	The status message of the suspend operation, if any.

**Returns**

False if the suspend operation is not supported; true otherwise.

Implements [rstudio::launcher\\_plugins::api::IJobSource](#).

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

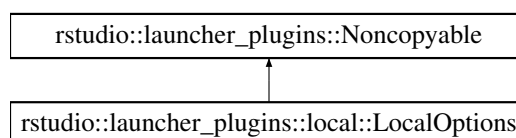
- /workspaces/rstudio-launcher-plugin-sdk/plugins/Local/include/LocalJobSource.hpp

**4.61 rstudio::launcher\_plugins::local::LocalOptions Class Reference**

Class which stores options specific to the Local Container system.

```
#include <LocalOptions.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::local::LocalOptions:



## Public Member Functions

- `size_t getNodeConnectionTimeoutSeconds () const`  
*Gets the number of seconds that can elapse before an attempted connection to another local node will be timed out.*
- `const system::FilePath & getSecureCookieKeyFile () const`  
*Gets the secure cookie key file to use for decrypting PAM passwords.*
- `void initialize ()`  
*Method which initializes [LocalOptions](#). This method should be called exactly once, before the options file is read.*
- `bool shouldSaveUnspecifiedOutput () const`  
*Gets whether to save output for a job when the output path has not been specified.*

## Static Public Member Functions

- `static LocalOptions & getInstance ()`  
*Gets the single instance of [LocalOptions](#) for the plugin.*

### 4.61.1 Detailed Description

Class which stores options specific to the Local Container system.

### 4.61.2 Member Function Documentation

#### 4.61.2.1 getInstance()

```
static LocalOptions& rstudio::launcher_plugins::local::LocalOptions::getInstance ( ) [static]
```

Gets the single instance of [LocalOptions](#) for the plugin.

#### Returns

The single instance of [LocalOptions](#) for the plugin.

#### 4.61.2.2 getNodeConnectionTimeoutSeconds()

```
size_t rstudio::launcher_plugins::local::LocalOptions::getNodeConnectionTimeoutSeconds ( )  
const
```

Gets the number of seconds that can elapse before an attempted connection to another local node will be timed out.

#### Returns

The timeout for connecting to other local nodes, in seconds.

### 4.61.2.3 getSecureCookieKeyFile()

```
const system::FilePath& rstudio::launcher_plugins::local::LocalOptions::getSecureCookieKeyFile
( ) const
```

Gets the secure cookie key file to use for decrypting PAM passwords.

#### Returns

The secure cookie key file to use for decrypting PAM passwords.

### 4.61.2.4 initialize()

```
void rstudio::launcher_plugins::local::LocalOptions::initialize ( )
```

Method which initializes [LocalOptions](#). This method should be called exactly once, before the options file is read.

This is where Local Options are registered with the Options object.

### 4.61.2.5 shouldSaveUnspecifiedOutput()

```
bool rstudio::launcher_plugins::local::LocalOptions::shouldSaveUnspecifiedOutput ( ) const
```

Gets whether to save output for a job when the output path has not been specified.

#### Returns

True if job output should be saved when no output path was specified; false otherwise.

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

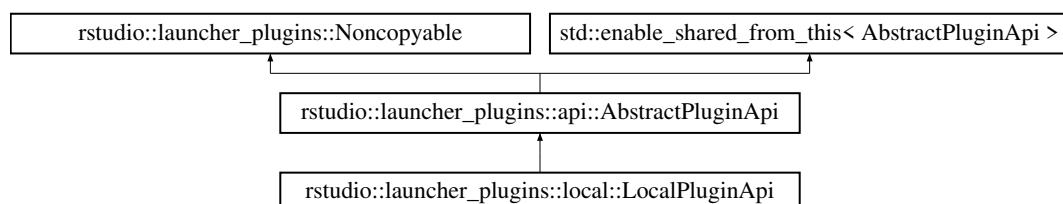
- /workspaces/rstudio-launcher-plugin-sdk/plugins/Local/include/LocalOptions.hpp

## 4.62 rstudio::launcher\_plugins::local::LocalPluginApi Class Reference

Launcher Plugin API for the Local Plugin.

```
#include <LocalPluginApi.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::local::LocalPluginApi:



## Public Member Functions

- [LocalPluginApi](#) (std::string in\_hostname, std::shared\_ptr< [comms::AbstractLauncherCommunicator](#) > in\_launcherCommunicator)

*Constructor.*

## Additional Inherited Members

### 4.62.1 Detailed Description

Launcher Plugin API for the Local Plugin.

### 4.62.2 Constructor & Destructor Documentation

#### 4.62.2.1 LocalPluginApi()

```
rstudio::launcher_plugins::local::LocalPluginApi::LocalPluginApi (
    std::string in_hostname,
    std::shared_ptr< comms::AbstractLauncherCommunicator > in_launcherCommunicator )
```

Constructor.

#### Parameters

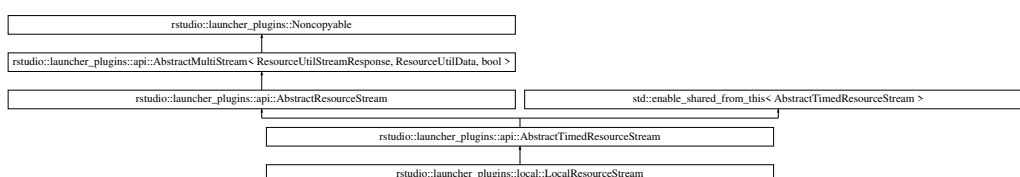
<i>in_hostname</i>	The name of the host running this instance of the Local Plugin.
<i>in_launcherCommunicator</i>	The communicator to use for sending and receiving messages from the RStudio Launcher.

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

- /workspaces/rstudio-launcher-plugin-sdk/plugins/Local/include/LocalPluginApi.hpp

## 4.63 rstudio::launcher\_plugins::local::LocalResourceStream Class Reference

Inheritance diagram for rstudio::launcher\_plugins::local::LocalResourceStream:



## Public Member Functions

- [LocalResourceStream](#) ([system::TimeDuration](#) in\_frequency, const [api::ConstJobPtr](#) &in\_job, [comms::AbstractLauncherCommunicatorPtr](#) in\_launcherCommunicator)

*Constructor.*

## Additional Inherited Members

### 4.63.1 Constructor & Destructor Documentation

#### 4.63.1.1 LocalResourceStream()

```
rstudio::launcher_plugins::local::LocalResourceStream::LocalResourceStream (
    system::TimeDuration in_frequency,
    const api::ConstJobPtr & in_job,
    comms::AbstractLauncherCommunicatorPtr in_launcherCommunicator )
```

Constructor.

#### Parameters

<i>in_frequency</i>	The frequency at which job resource utilization metrics should be polled.
<i>in_job</i>	The job for which resource utilization metrics should be streamed.
<i>in_launcherCommunicator</i>	The communicator through which messages may be sent to the launcher.

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

- /workspaces/rstudio-launcher-plugin-sdk/plugins/Local/include/LocalResourceStream.hpp

## 4.64 rstudio::launcher\_plugins::local::LocalSecureCookie Class Reference

Reads and makes available the secure-cookie-key-file specified in the launcher.local.conf file.

```
#include <LocalSecureCookie.hpp>
```

## Public Member Functions

- [LocalSecureCookie](#) ()=default  
*Constructor.*
- [Error initialize](#) ()  
*Reads and validates the secure-cookie-key from the location specified in the options.*
- const std::string & [getKey](#) () const  
*Gets the secure cookie key.*

### 4.64.1 Detailed Description

Reads and makes available the secure-cookie-key-file specified in the launcher.local.conf file.

### 4.64.2 Member Function Documentation

#### 4.64.2.1 getKey()

```
const std::string& rstudio::launcher_plugins::local::LocalSecureCookie::getKey ( ) const
```

Gets the secure cookie key.

##### Returns

The secure cookie key.

#### 4.64.2.2 initialize()

```
Error rstudio::launcher_plugins::local::LocalSecureCookie::initialize ( )
```

Reads and validates the secure-cookie-key from the location specified in the options.

##### Returns

[Success](#) if the secure-cookie-key exists and was valid; [Error](#) otherwise.

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

- /workspaces/rstudio-launcher-plugin-sdk/plugins/Local/include/LocalSecureCookie.hpp

## 4.65 rstudio::launcher\_plugins::json::Object::Member Class Reference

Class which represents a single member of a JSON object.

```
#include <Json.hpp>
```

### Public Member Functions

- [Member](#) ()=default  
*Default constructor.*
- [Member](#) (const std::shared\_ptr< Impl > &in\_impl)  
*Creates a [Member](#) object via its private implementation.*
- const std::string & [getName](#) () const  
*Gets the name of the member.*
- [Value](#) [getValue](#) () const  
*Gets the value of the member.*



## Friends

- class `Iterator`

### 4.65.1 Detailed Description

Class which represents a single member of a JSON object.

### 4.65.2 Constructor & Destructor Documentation

#### 4.65.2.1 Member()

```
rstudio::launcher_plugins::json::Object::Member::Member (  
    const std::shared_ptr< Impl > & in_impl ) [explicit]
```

Creates a [Member](#) object via its private implementation.

##### Parameters

<i>in_impl</i>	The private implementation of the member.
----------------	---

### 4.65.3 Member Function Documentation

#### 4.65.3.1 getName()

```
const std::string& rstudio::launcher_plugins::json::Object::Member::getName ( ) const
```

Gets the name of the member.

##### Returns

The name of the member.

#### 4.65.3.2 getValue()

```
Value rstudio::launcher_plugins::json::Object::Member::getValue ( ) const
```

Gets the value of the member.

##### Returns

The value of the member.

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

- `/workspaces/rstudio-launcher-plugin-sdk/sdk/include/json/Json.hpp`

## 4.66 rstudio::launcher\_plugins::api::Mount Struct Reference

Struct which represents an file system mount available to a job.

```
#include <Job.hpp>
```

### Public Member Functions

- [json::Object toJson](#) () const  
*Converts this [Mount](#) to a JSON object which represents it.*

### Static Public Member Functions

- static [Error fromJson](#) (const [json::Object](#) &in\_json, [Mount](#) &out\_mount)  
*Constructs a [Mount](#) from a JSON object which represents the mount.*

### Public Attributes

- std::string [Destination](#)
- bool [IsReadOnly](#)
- [MountSource](#) [Source](#)

#### 4.66.1 Detailed Description

Struct which represents an file system mount available to a job.

#### 4.66.2 Member Function Documentation

##### 4.66.2.1 fromJson()

```
static Error rstudio::launcher_plugins::api::Mount::fromJson (
    const json::Object & in_json,
    Mount & out_mount ) [static]
```

Constructs a [Mount](#) from a JSON object which represents the mount.

##### Parameters

<i>in_json</i>	The JSON object which represents the mount.
<i>out_mount</i>	The populated mount value. Not valid if an error is returned.

#### Returns

[Success](#) if `in_json` could be parsed as a [Mount](#); [Error](#) otherwise.

#### 4.66.2.2 toJson()

```
json::Object rstudio::launcher_plugins::api::Mount::toJson ( ) const
```

Converts this [Mount](#) to a JSON object which represents it.

#### Returns

The JSON object which represents this [Mount](#).

### 4.66.3 Member Data Documentation

#### 4.66.3.1 Destination

```
std::string rstudio::launcher_plugins::api::Mount::Destination
```

The path to which to mount the source path.

#### 4.66.3.2 IsReadOnly

```
bool rstudio::launcher_plugins::api::Mount::IsReadOnly
```

Whether the mounted path is read only.

#### 4.66.3.3 Source

```
MountSource rstudio::launcher_plugins::api::Mount::Source
```

The source to mount.

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

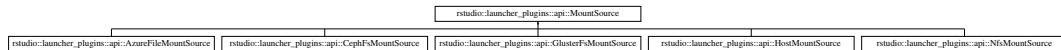
- `/workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Job.hpp`

## 4.67 rstudio::launcher\_plugins::api::MountSource Struct Reference

Struct which represents the source path of an NFS [Mount](#).

```
#include <Job.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::MountSource:



### Public Types

- enum [Type](#) {  
[Type::AZURE\\_FILE](#), [Type::CEPH\\_FS](#), [Type::GLUSTER\\_FS](#), [Type::HOST](#),  
[Type::NFS](#), [Type::PASSTHROUGH](#) }

*Constants representing the support types of [MountSource](#).*

### Public Member Functions

- virtual [~MountSource](#) ()=default  
*Virtual destructor for inheritance.*
- [AzureFileMountSource](#) & [asAzureFileMountSource](#) ()  
*Gets this [MountSource](#) as an [AzureFileMountSource](#).*
- const [AzureFileMountSource](#) & [asAzureFileMountSource](#) () const  
*Gets this [MountSource](#) as an [AzureFileMountSource](#).*
- [CephFsMountSource](#) & [asCephFsMountSource](#) ()  
*Gets this [MountSource](#) as an [CephFsMountSource](#).*
- const [CephFsMountSource](#) & [asCephFsMountSource](#) () const  
*Gets this [MountSource](#) as an [CephFsMountSource](#).*
- [GlusterFsMountSource](#) & [asGlusterFsMountSource](#) ()  
*Gets this [MountSource](#) as an [GlusterFsMountSource](#).*
- const [GlusterFsMountSource](#) & [asGlusterFsMountSource](#) () const  
*Gets this [MountSource](#) as an [GlusterFsMountSource](#).*
- [HostMountSource](#) & [asHostMountSource](#) ()  
*Gets this [MountSource](#) as an [HostMountSource](#).*
- const [HostMountSource](#) & [asHostMountSource](#) () const  
*Gets this [MountSource](#) as an [HostMountSource](#).*
- [NfsMountSource](#) & [asNfsMountSource](#) ()  
*Gets this [MountSource](#) as an [NfsMountSource](#).*
- const [NfsMountSource](#) & [asNfsMountSource](#) () const  
*Gets this [MountSource](#) as an [NfsMountSource](#).*
- bool [isAzureFileMountSource](#) () const  
*Checks whether this [MountSource](#) is an [AzureFileMountSource](#).*
- bool [isCephFsMountSource](#) () const  
*Checks whether this [MountSource](#) is an [CephFsMountSource](#).*
- bool [isGlusterFsMountSource](#) () const  
*Checks whether this [MountSource](#) is an [GlusterFsMountSource](#).*
- bool [isHostMountSource](#) () const

- Checks whether this [MountSource](#) is an [HostMountSource](#).  
• bool [isNfsMountSource](#) () const  
Checks whether this [MountSource](#) is an [NfsMountSource](#).
- bool [isPassthroughMountSource](#) () const  
Checks whether this [MountSource](#) is an [PassthroughMountSource](#).
- [json::Object toJson](#) () const  
Converts this [NfsMountSource](#) to a JSON object which represents it.

## Static Public Member Functions

- static [Error fromJson](#) (const [json::Object](#) &in\_json, [MountSource](#) &out\_mountSource)  
Constructs a [MountSource](#) from a JSON object which represents the mount source.

## Public Attributes

- std::string [CustomType](#)
- [json::Object](#) [SourceObject](#)
- [Type](#) [SourceType](#)

### 4.67.1 Detailed Description

Struct which represents the source path of an NFS [Mount](#).

### 4.67.2 Member Enumeration Documentation

#### 4.67.2.1 Type

```
enum rstudio::launcher_plugins::api::MountSource::Type [strong]
```

Constants representing the support types of [MountSource](#).

#### Enumerator

AZURE_FILE	Represents an Azure File <a href="#">Mount</a> Source.
CEPH_FS	Represents a Ceph File System <a href="#">Mount</a> Source.
GLUSTER_FS	Represents a Gluster File System <a href="#">Mount</a> Source.
HOST	Represents a Host <a href="#">Mount</a> Source.
NFS	Represents an NFS <a href="#">Mount</a> Source.
PASSTHROUGH	Represents a <a href="#">Mount</a> Source that will be passed through to the Plugin to handle.

### 4.67.3 Member Function Documentation

#### 4.67.3.1 asAzureFileMountSource() [1/2]

```
AzureFileMountSource& rstudio::launcher_plugins::api::MountSource::asAzureFileMountSource ( )
```

Gets this [MountSource](#) as an [AzureFileMountSource](#).

##### Exceptions

<code>std::logic_error</code>	if <a href="#">isAzureFileMountSource()</a> would return false.
-------------------------------	---

##### Returns

This [MountSource](#) as an [AzureFileMountSource](#).

#### 4.67.3.2 asAzureFileMountSource() [2/2]

```
const AzureFileMountSource& rstudio::launcher_plugins::api::MountSource::asAzureFileMountSource ( ) const
```

Gets this [MountSource](#) as an [AzureFileMountSource](#).

##### Exceptions

<code>std::logic_error</code>	if <a href="#">isAzureFileMountSource()</a> would return false.
-------------------------------	---

##### Returns

This [MountSource](#) as an [AzureFileMountSource](#).

#### 4.67.3.3 asCephFsMountSource() [1/2]

```
CephFsMountSource& rstudio::launcher_plugins::api::MountSource::asCephFsMountSource ( )
```

Gets this [MountSource](#) as an [CephFsMountSource](#).

##### Exceptions

<code>std::logic_error</code>	if <a href="#">isCephFsMountSource()</a> would return false.
-------------------------------	--

**Returns**

This [MountSource](#) as an [CephFsMountSource](#).

**4.67.3.4 asCephFsMountSource() [2/2]**

```
const CephFsMountSource& rstudio::launcher_plugins::api::MountSource::asCephFsMountSource ( )
const
```

Gets this [MountSource](#) as an [CephFsMountSource](#).

**Exceptions**

<code>std::logic_error</code>	if <a href="#">isCephFsMountSource()</a> would return false.
-------------------------------	--

**Returns**

This [MountSource](#) as an [CephFsMountSource](#).

**4.67.3.5 asGlusterFsMountSource() [1/2]**

```
GlusterFsMountSource& rstudio::launcher_plugins::api::MountSource::asGlusterFsMountSource ( )
```

Gets this [MountSource](#) as an [GlusterFsMountSource](#).

**Exceptions**

<code>std::logic_error</code>	if <a href="#">isGlusterFsMountSource()</a> would return false.
-------------------------------	---

**Returns**

This [MountSource](#) as an [GlusterFsMountSource](#).

**4.67.3.6 asGlusterFsMountSource() [2/2]**

```
const GlusterFsMountSource& rstudio::launcher_plugins::api::MountSource::asGlusterFsMount↵
Source ( ) const
```

Gets this [MountSource](#) as an [GlusterFsMountSource](#).

## Exceptions

<code>std::logic_error</code>	if <code>isGlusterFsMountSource()</code> would return false.
-------------------------------	--

## Returns

This [MountSource](#) as an [GlusterFsMountSource](#).

**4.67.3.7 asHostMountSource()** [1/2]

```
HostMountSource& rstudio::launcher_plugins::api::MountSource::asHostMountSource ( )
```

Gets this [MountSource](#) as an [HostMountSource](#).

## Exceptions

<code>std::logic_error</code>	if <code>isHostMountSource()</code> would return false.
-------------------------------	---

## Returns

This [MountSource](#) as an [HostMountSource](#).

**4.67.3.8 asHostMountSource()** [2/2]

```
const HostMountSource& rstudio::launcher_plugins::api::MountSource::asHostMountSource ( )
const
```

Gets this [MountSource](#) as an [HostMountSource](#).

## Exceptions

<code>std::logic_error</code>	if <code>isHostMountSource()</code> would return false.
-------------------------------	---

## Returns

This [MountSource](#) as an [HostMountSource](#).

**4.67.3.9 asNfsMountSource()** [1/2]

```
NfsMountSource& rstudio::launcher_plugins::api::MountSource::asNfsMountSource ( )
```

Gets this [MountSource](#) as an [NfsMountSource](#).



## Exceptions

<code>std::logic_error</code>	if <code>isNfsMountSource()</code> would return false.
-------------------------------	--

## Returns

This `MountSource` as an `NfsMountSource`.

**4.67.3.10 asNfsMountSource()** [2/2]

```
const NfsMountSource& rstudio::launcher_plugins::api::MountSource::asNfsMountSource ( ) const
```

Gets this `MountSource` as an `NfsMountSource`.

## Exceptions

<code>std::logic_error</code>	if <code>isNfsMountSource()</code> would return false.
-------------------------------	--

## Returns

This `MountSource` as an `NfsMountSource`.

**4.67.3.11 fromJson()**

```
static Error rstudio::launcher_plugins::api::MountSource::fromJson (
    const json::Object & in_json,
    MountSource & out_mountSource ) [static]
```

Constructs a `MountSource` from a JSON object which represents the mount source.

## Parameters

<code>in_json</code>	The JSON object which represents the mount source.
<code>out_mountSource</code>	The populated mount source value. Not valid if an error is returned.

## Returns

`Success` if `in_json` could be parsed as a `MountSource`; `Error` otherwise.

**4.67.3.12 isAzureFileMountSource()**

```
bool rstudio::launcher_plugins::api::MountSource::isAzureFileMountSource ( ) const
```

Checks whether this `MountSource` is an `AzureFileMountSource`.

**Returns**

True if this [MountSource](#) is an [AzureFileMountSource](#); false otherwise.

**4.67.3.13 isCephFsMountSource()**

```
bool rstudio::launcher_plugins::api::MountSource::isCephFsMountSource ( ) const
```

Checks whether this [MountSource](#) is an [CephFsMountSource](#).

**Returns**

True if this [MountSource](#) is an [CephFsMountSource](#); false otherwise.

**4.67.3.14 isGlusterFsMountSource()**

```
bool rstudio::launcher_plugins::api::MountSource::isGlusterFsMountSource ( ) const
```

Checks whether this [MountSource](#) is an [GlusterFsMountSource](#).

**Returns**

True if this [MountSource](#) is an [GlusterFsMountSource](#); false otherwise.

**4.67.3.15 isHostMountSource()**

```
bool rstudio::launcher_plugins::api::MountSource::isHostMountSource ( ) const
```

Checks whether this [MountSource](#) is an [HostMountSource](#).

**Returns**

True if this [MountSource](#) is an [HostMountSource](#); false otherwise.

**4.67.3.16 isNfsMountSource()**

```
bool rstudio::launcher_plugins::api::MountSource::isNfsMountSource ( ) const
```

Checks whether this [MountSource](#) is an [NfsMountSource](#).

**Returns**

True if this [MountSource](#) is an [NfsMountSource](#); false otherwise.

#### 4.67.3.17 isPassthroughMountSource()

```
bool rstudio::launcher_plugins::api::MountSource::isPassthroughMountSource ( ) const
```

Checks whether this [MountSource](#) is an PassthroughMountSource.

##### Returns

True if this [MountSource](#) is an PassthroughMountSource; false otherwise.

#### 4.67.3.18 toJson()

```
json::Object rstudio::launcher_plugins::api::MountSource::toJson ( ) const
```

Converts this [NfsMountSource](#) to a JSON object which represents it.

##### Returns

The JSON object which represents this [NfsMountSource](#).

### 4.67.4 Member Data Documentation

#### 4.67.4.1 CustomType

```
std::string rstudio::launcher_plugins::api::MountSource::CustomType
```

An optional field, to represent a custom mount source type. If the type field is not recongized SourceType will be set to [MountSource::Type::PASSTHROUGH](#) and CustomType will hold the original value of the field.

#### 4.67.4.2 SourceObject

```
json::Object rstudio::launcher_plugins::api::MountSource::SourceObject
```

The JSON object that describes the [Mount](#) Source for this [Mount](#).

#### 4.67.4.3 SourceType

```
Type rstudio::launcher_plugins::api::MountSource::SourceType
```

The type of the [Mount](#) Source for this [Mount](#).

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

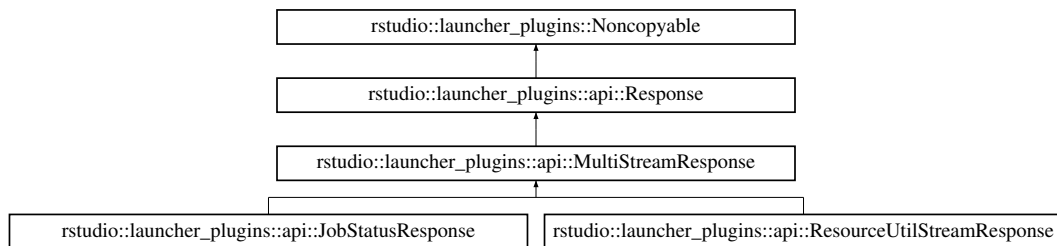
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Job.hpp

## 4.68 rstudio::launcher\_plugins::api::MultiStreamResponse Class Reference

Base class for responses which are returned to multiple stream requests.

```
#include <Response.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::MultiStreamResponse:



### Public Member Functions

- [json::Object toJson](#) () const override  
*Converts this [MultiStreamResponse](#) to a JSON object.*

### Protected Member Functions

- [MultiStreamResponse](#) ([Type](#) in\_responseType, StreamSequences in\_sequences)  
*Constructor.*

### Additional Inherited Members

#### 4.68.1 Detailed Description

Base class for responses which are returned to multiple stream requests.

#### 4.68.2 Constructor & Destructor Documentation

##### 4.68.2.1 MultiStreamResponse()

```
rstudio::launcher_plugins::api::MultiStreamResponse::MultiStreamResponse (
    Type in_responseType,
    StreamSequences in_sequences ) [protected]
```

Constructor.

## Parameters

<i>in_responseType</i>	The type of the base class.
<i>in_sequences</i>	The sequence IDs for which this response should be sent.

### 4.68.3 Member Function Documentation

#### 4.68.3.1 `toJson()`

```
json::Object rstudio::launcher_plugins::api::MultiStreamResponse::toJson ( ) const [override],  
[virtual]
```

Converts this [MultiStreamResponse](#) to a JSON object.

## Returns

The JSON object which represents this [MultiStreamResponse](#).

Reimplemented from [rstudio::launcher\\_plugins::api::Response](#).

Reimplemented in [rstudio::launcher\\_plugins::api::ResourceUtilStreamResponse](#), and [rstudio::launcher\\_plugins::api::JobStatusResponse](#).

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

- `/workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Response.hpp`

## 4.69 `rstudio::launcher_plugins::api::NetworkInfo` Struct Reference

Represents the network information for a job.

```
#include <ResponseTypes.hpp>
```

### Public Attributes

- `std::string` [Hostname](#)
- `std::vector< std::string >` [IpAddresses](#)

#### 4.69.1 Detailed Description

Represents the network information for a job.

## 4.69.2 Member Data Documentation

### 4.69.2.1 Hostname

```
std::string rstudio::launcher_plugins::api::NetworkInfo::Hostname
```

The hostname of the machine running the job.

### 4.69.2.2 IpAddresses

```
std::vector<std::string> rstudio::launcher_plugins::api::NetworkInfo::IpAddresses
```

The IP Address(es) of the machine running the job.

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

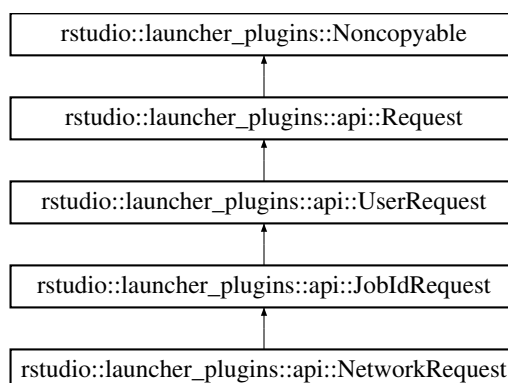
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/ResponseTypes.hpp

## 4.70 rstudio::launcher\_plugins::api::NetworkRequest Class Reference

[Request](#) from the Launcher to get the network information for a job.

```
#include <Request.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::NetworkRequest:



## Friends

- class **Request**

## Additional Inherited Members

### 4.70.1 Detailed Description

[Request](#) from the Launcher to get the network information for a job.

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

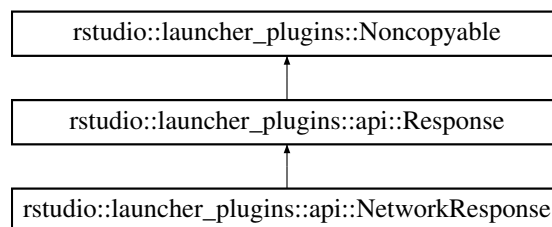
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Request.hpp

## 4.71 rstudio::launcher\_plugins::api::NetworkResponse Class Reference

Class which represents a network information response which should be sent to the Launcher in response to a [Job](#) network information request.

```
#include <Response.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::NetworkResponse:



## Public Member Functions

- [NetworkResponse](#) (uint64\_t in\_requestId, [NetworkInfo](#) in\_networkInfo)  
*Constructor.*
- [json::Object toJson](#) () const override  
*Converts this cluster info response to a JSON object.*

## Additional Inherited Members

### 4.71.1 Detailed Description

Class which represents a network information response which should be sent to the Launcher in response to a [Job](#) network information request.

### 4.71.2 Constructor & Destructor Documentation

#### 4.71.2.1 NetworkResponse()

```
rstudio::launcher_plugins::api::NetworkResponse::NetworkResponse (
    uint64_t in_requestId,
    NetworkInfo in_networkInfo )
```

Constructor.

## Parameters

<i>in_requestId</i>	The ID of the request for which this response is being sent.
<i>in_networkInfo</i>	The network information for the requested <a href="#">Job</a> .

### 4.71.3 Member Function Documentation

#### 4.71.3.1 toJson()

```
json::Object rstudio::launcher_plugins::api::NetworkResponse::toJson ( ) const [override],  
[virtual]
```

Converts this cluster info response to a JSON object.

## Returns

The JSON object which represents this network info response.

Reimplemented from [rstudio::launcher\\_plugins::api::Response](#).

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

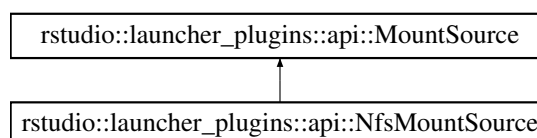
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Response.hpp

## 4.72 rstudio::launcher\_plugins::api::NfsMountSource Struct Reference

Represents an NFS [Mount](#) Source.

```
#include <Job.hpp>
```

Inheritance diagram for `rstudio::launcher_plugins::api::NfsMountSource`:



### Public Member Functions

- `std::string getHost () const`  
*Gets the NFS host.*
- `std::string getPath () const`  
*Gets the path on the NFS host to be mounted.*



## Static Public Member Functions

- static [Error fromJson](#) (const [json::Object](#) &in\_json, [NfsMountSource](#) &out\_mountSource)  
*Constructs an [NfsMountSource](#) from a JSON object which represents the mount source.*

## Friends

- class [MountSource](#)

## Additional Inherited Members

### 4.72.1 Detailed Description

Represents an NFS [Mount](#) Source.

### 4.72.2 Member Function Documentation

#### 4.72.2.1 fromJson()

```
static Error rstudio::launcher_plugins::api::NfsMountSource::fromJson (
    const json::Object & in_json,
    NfsMountSource & out_mountSource ) [static]
```

Constructs an [NfsMountSource](#) from a JSON object which represents the mount source.

#### Parameters

<i>in_json</i>	The JSON object which represents the mount source.
<i>out_mountSource</i>	The populated mount source value. Not valid if an error is returned.

#### Returns

[Success](#) if in\_json could be parsed as an [NfsMountSource](#); [Error](#) otherwise.

#### 4.72.2.2 getHost()

```
std::string rstudio::launcher_plugins::api::NfsMountSource::getHost ( ) const
```

Gets the NFS host.

**Exceptions**

<code>std::logic_error</code>	if the 'host' field cannot be found.
-------------------------------	--------------------------------------

**Returns**

The NFS host.

**4.72.2.3 getPath()**

```
std::string rstudio::launcher_plugins::api::NfsMountSource::getPath ( ) const
```

Gets the path on the NFS host to be mounted.

**Exceptions**

<code>std::logic_error</code>	if the 'path' field cannot be found.
-------------------------------	--------------------------------------

**Returns**

The path on the NFS host to be mounted.

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

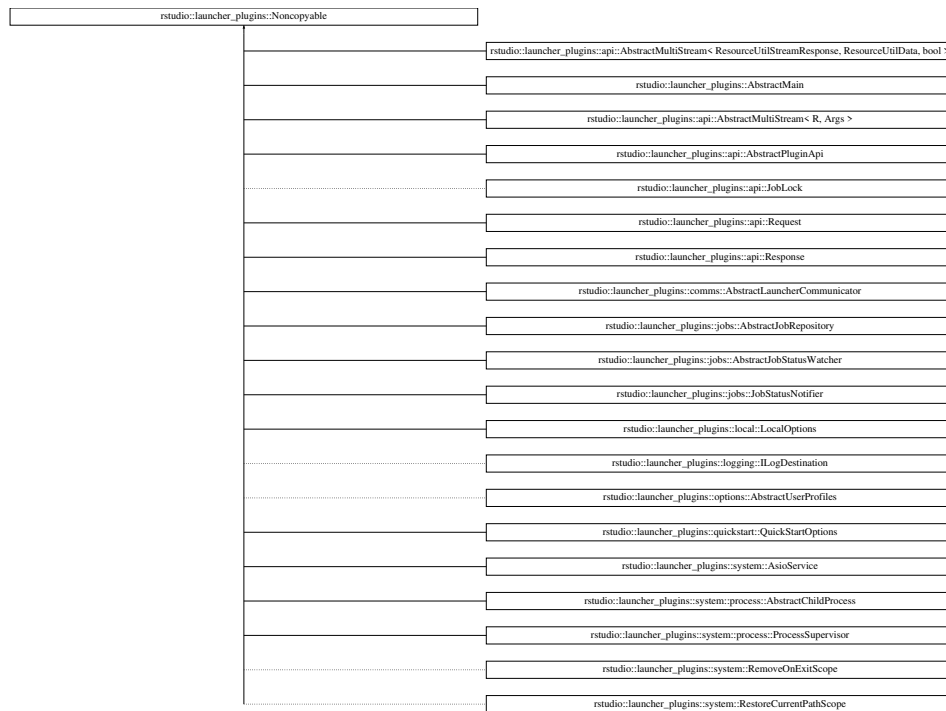
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Job.hpp

**4.73 rstudio::launcher\_plugins::Noncopyable Class Reference**

Class which can be inherited from to disallow copying of its child classes.

```
#include <Noncopyable.hpp>
```

Inheritance diagram for `rstudio::launcher_plugins::Noncopyable`:



## Public Member Functions

- **Noncopyable** ()=default  
*Default constructor.*
- **Noncopyable** (const **Noncopyable** &)=delete  
*Deleted copy constructor.*
- **Noncopyable** & **operator=** (const **Noncopyable** &)=delete  
*Deleted assignment operator.*

### 4.73.1 Detailed Description

Class which can be inherited from to disallow copying of its child classes.

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

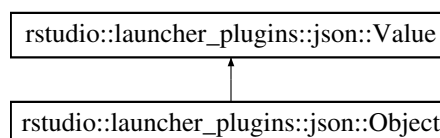
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/Noncopyable.hpp

## 4.74 rstudio::launcher\_plugins::json::Object Class Reference

Class which represents a specific type of JSON **Value**: a JSON object.

```
#include <Json.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::json::Object:



## Classes

- class [Iterator](#)  
*Class which allows iterating over the members of a JSON object.*
- class [Member](#)  
*Class which represents a single member of a JSON object.*

## Public Types

- typedef std::reverse\_iterator< [Iterator](#) > [Reverseliterator](#)  
*Reverse iterator for a JSON object.*

## Public Member Functions

- [Object](#) ()  
*Constructs an empty JSON object.*
- [Object](#) (const StringPairList &in\_strPairs)  
*Constructs a JSON object from a list of string pairs.*
- [Object](#) (const [Object](#) &in\_other)  
*Copy constructor.*
- [Object](#) ([Object](#) &&in\_other) noexcept  
*Move constructor.*
- [Object](#) & operator= (const [Object](#) &in\_other)  
*Assignment operator.*
- [Object](#) & operator= ([Object](#) &&in\_other) noexcept  
*Move operator.*
- [Value operator\[\]](#) (const char \*in\_name)  
*Accessor operator. Gets the value a member of this JSON object by name. If no such object exists, an empty JSON value will be returned.*
- [Value operator\[\]](#) (const std::string &in\_name)  
*Accessor operator. Gets the value a member of this JSON object by name. If no such object exists, an empty JSON value will be returned.*
- [Iterator find](#) (const char \*in\_name) const  
*Finds a JSON member by name.*
- [Iterator find](#) (const std::string &in\_name) const  
*Finds a JSON member by name.*
- [Iterator begin](#) () const  
*Gets an iterator pointing to the first member of this object.*
- [Iterator end](#) () const  
*Gets an iterator after the last member of this object.*
- [Reverseliterator rbegin](#) () const  
*Gets an iterator pointing to the last member of this object, which iterates in the reverse direction.*
- [Reverseliterator rend](#) () const  
*Gets an iterator before the first member of this object, which can be compared with an other Reverseliterator to determine when reverse iteration has ended.*
- void [clear](#) ()  
*Clears the JSON object.*
- bool [erase](#) (const char \*in\_name)  
*Erases a member by name.*
- bool [erase](#) (const std::string &in\_name)

- Erases a member by name.*

  - `Iterator erase` (const `Iterator` &in\_itr)

*Erases the member specified by the provided iterator.*
- `size_t getSize` () const

*Gets the number of members in the JSON object.*
- `bool hasMember` (const char \*in\_name) const

*Checks whether this object has a member with the specified name.*
- `bool hasMember` (const std::string &in\_name) const

*Checks whether this object has a member with the specified name.*
- `void insert` (const std::string &in\_name, const `Value` &in\_value)

*Inserts the specified member into this JSON object. If an object with the same name already exists, it will be overridden.*
- `void insert` (const std::string &in\_name, bool in\_value)

*Inserts the specified member into this JSON object. If an object with the same name already exists, it will be overridden.*
- `void insert` (const std::string &in\_name, double in\_value)

*Inserts the specified member into this JSON object. If an object with the same name already exists, it will be overridden.*
- `void insert` (const std::string &in\_name, float in\_value)

*Inserts the specified member into this JSON object. If an object with the same name already exists, it will be overridden.*
- `void insert` (const std::string &in\_name, int in\_value)

*Inserts the specified member into this JSON object. If an object with the same name already exists, it will be overridden.*
- `void insert` (const std::string &in\_name, int64\_t in\_value)

*Inserts the specified member into this JSON object. If an object with the same name already exists, it will be overridden.*
- `void insert` (const std::string &in\_name, const char \*in\_value)

*Inserts the specified member into this JSON object. If an object with the same name already exists, it will be overridden.*
- `void insert` (const std::string &in\_name, const std::string &in\_value)

*Inserts the specified member into this JSON object. If an object with the same name already exists, it will be overridden.*
- `void insert` (const std::string &in\_name, unsigned int in\_value)

*Inserts the specified member into this JSON object. If an object with the same name already exists, it will be overridden.*
- `void insert` (const std::string &in\_name, uint64\_t in\_value)

*Inserts the specified member into this JSON object. If an object with the same name already exists, it will be overridden.*
- `void insert` (const std::string &in\_name, const `Array` &in\_value)

*Inserts the specified member into this JSON object. If an object with the same name already exists, it will be overridden.*
- `void insert` (const std::string &in\_name, const `Object` &in\_value)

*Inserts the specified member into this JSON object. If an object with the same name already exists, it will be overridden.*
- `void insert` (const `Member` &in\_member)

*Inserts the specified member into this JSON object. If an object with the same name already exists, it will be overridden.*
- `bool isEmpty` () const

*Checks whether the JSON object is empty.*
- `Error parse` (const char \*in\_jsonStr) override

*Parses the JSON string into this object.*
- `Error parse` (const std::string &in\_jsonStr) override

*Parses the JSON string into this object.*

- bool [toStringMap](#) (StringListMap &out\_map) const  
*Converts this JSON object to a map with string keys and a list of string values.*
- StringPairList [toStringPairList](#) () const  
*Converts this JSON object to a list of string pairs.*

## Static Public Member Functions

- static [Member](#) [createMember](#) (const std::string &in\_name, const [Value](#) &in\_value)  
*Creates a JSON object from the given name and JSON value.*
- static [Error](#) [getSchemaDefaults](#) (const std::string &in\_schema, [Object](#) &out\_schemaDefaults)  
*Creates a JSON object which represents the schema defaults of the provided JSON schema string.*
- static [Object](#) [mergeObjects](#) (const [Object](#) &in\_base, const [Object](#) &in\_overlay)  
*Merges two JSON objects together. Conflicts between the base and the overlay will be resolved by taking the value in the overlay.*

## Friends

- class [Value](#)
- class [Iterator](#)

## Additional Inherited Members

### 4.74.1 Detailed Description

Class which represents a specific type of JSON [Value](#): a JSON object.

### 4.74.2 Constructor & Destructor Documentation

#### 4.74.2.1 [Object\(\)](#) [1/3]

```
rstudio::launcher_plugins::json::Object::Object (
    const StringPairList & in_strPairs ) [explicit]
```

Constructs a JSON object from a list of string pairs.

#### Parameters

<i>in_strPairs</i>	The list of string pairs from which to construct this object.
--------------------	---

#### 4.74.2.2 Object() [2/3]

```
rstudio::launcher_plugins::json::Object::Object (
    const Object & in_other )
```

Copy constructor.

##### Parameters

<i>in_other</i>	The JSON object to copy from.
-----------------	-------------------------------

#### 4.74.2.3 Object() [3/3]

```
rstudio::launcher_plugins::json::Object::Object (
    Object && in_other ) [noexcept]
```

Move constructor.

##### Parameters

<i>in_other</i>	The JSON object to move to this <a href="#">Object</a> .
-----------------	--

### 4.74.3 Member Function Documentation

#### 4.74.3.1 begin()

```
Iterator rstudio::launcher_plugins::json::Object::begin ( ) const
```

Gets an iterator pointing to the first member of this object.

##### Returns

An iterator pointing to the first member of this object.

#### 4.74.3.2 createMember()

```
static Member rstudio::launcher_plugins::json::Object::createMember (
    const std::string & in_name,
    const Value & in_value ) [static]
```

Creates a JSON object from the given name and JSON value.

**Parameters**

<i>in_name</i>	The name of the JSON object.
<i>in_value</i>	The value of the JSON object.

**Returns**

The newly created member.

**4.74.3.3 end()**

```
Iterator rstudio::launcher_plugins::json::Object::end ( ) const
```

Gets an iterator after the last member of this object.

**Returns**

An iterator after the last member of this object.

**4.74.3.4 erase() [1/3]**

```
bool rstudio::launcher_plugins::json::Object::erase (
    const char * in_name )
```

Erases a member by name.

**Parameters**

<i>in_name</i>	The name of the member to erase.
----------------	----------------------------------

**Returns**

True if a member was erased; false otherwise.

**4.74.3.5 erase() [2/3]**

```
Iterator rstudio::launcher_plugins::json::Object::erase (
    const Iterator & in_itr )
```

Erases the member specified by the provided iterator.



## Parameters

<i>in</i> ↔ _itr	The iterator pointing to the member to erase.
---------------------	---

## Returns

An iterator pointing to the member immediately after the erased member.

**4.74.3.6 erase()** [3/3]

```
bool rstudio::launcher_plugins::json::Object::erase (  
    const std::string & in_name )
```

Erases a member by name.

## Parameters

<i>in_name</i>	The name of the member to erase.
----------------	----------------------------------

## Returns

True if a member was erased; false otherwise.

**4.74.3.7 find()** [1/2]

```
Iterator rstudio::launcher_plugins::json::Object::find (  
    const char * in_name ) const
```

Finds a JSON member by name.

## Parameters

<i>in_name</i>	The name of the member to find.
----------------	---------------------------------

## Returns

If such a member exists, an iterator pointing to that member; the end iterator otherwise.

**4.74.3.8 find()** [2/2]

```
Iterator rstudio::launcher_plugins::json::Object::find (  
    const std::string & in_name ) const
```

Finds a JSON member by name.

#### Parameters

<i>in_name</i>	The name of the member to find.
----------------	---------------------------------

#### Returns

If such a member exists, an iterator pointing to that member; the end iterator otherwise.

#### 4.74.3.9 getSchemaDefaults()

```
static Error rstudio::launcher_plugins::json::Object::getSchemaDefaults (
    const std::string & in_schema,
    Object & out_schemaDefaults ) [static]
```

Creates a JSON object which represents the schema defaults of the provided JSON schema string.

#### Parameters

<i>in_schema</i>	The JSON schema string to parse into a JSON object.
<i>out_schemaDefaults</i>	The parsed schema defaults. This object is not valid if an error is returned.

#### Returns

**Success** if *in\_schema* could be parsed; **Error** otherwise.

#### 4.74.3.10 getSize()

```
size_t rstudio::launcher_plugins::json::Object::getSize ( ) const
```

Gets the number of members in the JSON object.

#### Returns

The number of members in the JSON object.

#### 4.74.3.11 hasMember() [1/2]

```
bool rstudio::launcher_plugins::json::Object::hasMember (
    const char * in_name ) const
```

Checks whether this object has a member with the specified name.

## Parameters

<i>in_name</i>	The name of the member for which to check.
----------------	--

## Returns

True if a member with the specified name exists; false otherwise.

**4.74.3.12 hasMember()** [2/2]

```
bool rstudio::launcher_plugins::json::Object::hasMember (
    const std::string & in_name ) const
```

Checks whether this object has a member with the specified name.

## Parameters

<i>in_name</i>	The name of the member for which to check.
----------------	--

## Returns

True if a member with the specified name exists; false otherwise.

**4.74.3.13 insert()** [1/13]

```
void rstudio::launcher_plugins::json::Object::insert (
    const Member & in_member )
```

Inserts the specified member into this JSON object. If an object with the same name already exists, it will be overridden.

## Parameters

<i>in_member</i>	The member to insert.
------------------	-----------------------

**4.74.3.14 insert()** [2/13]

```
void rstudio::launcher_plugins::json::Object::insert (
    const std::string & in_name,
    bool in_value )
```

Inserts the specified member into this JSON object. If an object with the same name already exists, it will be overridden.

## Parameters

<i>in_name</i>	The name of the JSON value to insert.
<i>in_value</i>	The value to insert.

**4.74.3.15 insert()** [3/13]

```
void rstudio::launcher_plugins::json::Object::insert (
    const std::string & in_name,
    const Array & in_value )
```

Inserts the specified member into this JSON object. If an object with the same name already exists, it will be overridden.

## Parameters

<i>in_name</i>	The name of the JSON value to insert.
<i>in_value</i>	The value to insert.

**4.74.3.16 insert()** [4/13]

```
void rstudio::launcher_plugins::json::Object::insert (
    const std::string & in_name,
    const char * in_value )
```

Inserts the specified member into this JSON object. If an object with the same name already exists, it will be overridden.

## Parameters

<i>in_name</i>	The name of the JSON value to insert.
<i>in_value</i>	The value to insert.

**4.74.3.17 insert()** [5/13]

```
void rstudio::launcher_plugins::json::Object::insert (
    const std::string & in_name,
    const Object & in_value )
```

Inserts the specified member into this JSON object. If an object with the same name already exists, it will be overridden.

## Parameters

<i>in_name</i>	The name of the JSON value to insert.
<i>in_value</i>	The value to insert.

**4.74.3.18 insert()** [6/13]

```
void rstudio::launcher_plugins::json::Object::insert (
    const std::string & in_name,
    const std::string & in_value )
```

Inserts the specified member into this JSON object. If an object with the same name already exists, it will be overridden.

## Parameters

<i>in_name</i>	The name of the JSON value to insert.
<i>in_value</i>	The value to insert.

**4.74.3.19 insert()** [7/13]

```
void rstudio::launcher_plugins::json::Object::insert (
    const std::string & in_name,
    const Value & in_value )
```

Inserts the specified member into this JSON object. If an object with the same name already exists, it will be overridden.

## Parameters

<i>in_name</i>	The name of the JSON value to insert.
<i>in_value</i>	The value to insert.

**4.74.3.20 insert()** [8/13]

```
void rstudio::launcher_plugins::json::Object::insert (
    const std::string & in_name,
    double in_value )
```

Inserts the specified member into this JSON object. If an object with the same name already exists, it will be overridden.

## Parameters

<i>in_name</i>	The name of the JSON value to insert.
<i>in_value</i>	The value to insert.

**4.74.3.21 insert()** [9/13]

```
void rstudio::launcher_plugins::json::Object::insert (
    const std::string & in_name,
    float in_value )
```

Inserts the specified member into this JSON object. If an object with the same name already exists, it will be overridden.

## Parameters

<i>in_name</i>	The name of the JSON value to insert.
<i>in_value</i>	The value to insert.

**4.74.3.22 insert()** [10/13]

```
void rstudio::launcher_plugins::json::Object::insert (
    const std::string & in_name,
    int in_value )
```

Inserts the specified member into this JSON object. If an object with the same name already exists, it will be overridden.

## Parameters

<i>in_name</i>	The name of the JSON value to insert.
<i>in_value</i>	The value to insert.

**4.74.3.23 insert()** [11/13]

```
void rstudio::launcher_plugins::json::Object::insert (
    const std::string & in_name,
    int64_t in_value )
```

Inserts the specified member into this JSON object. If an object with the same name already exists, it will be overridden.

## Parameters

<i>in_name</i>	The name of the JSON value to insert.
<i>in_value</i>	The value to insert.

**4.74.3.24 insert()** [12/13]

```
void rstudio::launcher_plugins::json::Object::insert (
    const std::string & in_name,
    uint64_t in_value )
```

Inserts the specified member into this JSON object. If an object with the same name already exists, it will be overridden.

## Parameters

<i>in_name</i>	The name of the JSON value to insert.
<i>in_value</i>	The value to insert.

**4.74.3.25 insert()** [13/13]

```
void rstudio::launcher_plugins::json::Object::insert (
    const std::string & in_name,
    unsigned int in_value )
```

Inserts the specified member into this JSON object. If an object with the same name already exists, it will be overridden.

## Parameters

<i>in_name</i>	The name of the JSON value to insert.
<i>in_value</i>	The value to insert.

**4.74.3.26 isEmpty()**

```
bool rstudio::launcher_plugins::json::Object::isEmpty ( ) const
```

Checks whether the JSON object is empty.

## Returns

True if the JSON object has no members; false otherwise.

#### 4.74.3.27 mergeObjects()

```
static Object rstudio::launcher_plugins::json::Object::mergeObjects (
    const Object & in_base,
    const Object & in_overlay ) [static]
```

Merges two JSON objects together. Conflicts between the base and the overlay will be resolved by taking the value in the overlay.

##### Parameters

<i>in_base</i>	The base object to merge.
<i>in_overlay</i>	The overlay object to merge with the base.

##### Returns

The merged object.

#### 4.74.3.28 operator=() [1/2]

```
Object& rstudio::launcher_plugins::json::Object::operator= (
    const Object & in_other )
```

Assignment operator.

##### Parameters

<i>in_other</i>	The JSON object to copy from.
-----------------	-------------------------------

##### Returns

A reference to this JSON object.

#### 4.74.3.29 operator=() [2/2]

```
Object& rstudio::launcher_plugins::json::Object::operator= (
    Object && in_other ) [noexcept]
```

Move operator.

##### Parameters

<i>in_other</i>	The JSON object to move from.
-----------------	-------------------------------



**Returns**

A reference to this JSON object.

**4.74.3.30 operator[]()** [1/2]

```
Value rstudio::launcher_plugins::json::Object::operator[] (
    const char * in_name )
```

Accessor operator. Gets the value a member of this JSON object by name. If no such object exists, an empty JSON value will be returned.

**Parameters**

<i>in_name</i>	The name of the member to access.
----------------	-----------------------------------

**Returns**

The value of the member with the specified name, if it exists; empty JSON value otherwise.

**4.74.3.31 operator[]()** [2/2]

```
Value rstudio::launcher_plugins::json::Object::operator[] (
    const std::string & in_name )
```

Accessor operator. Gets the value a member of this JSON object by name. If no such object exists, an empty JSON value will be returned.

**Parameters**

<i>in_name</i>	The name of the member to access.
----------------	-----------------------------------

**Returns**

The value of the member with the specified name, if it exists; empty JSON value otherwise.

**4.74.3.32 parse()** [1/2]

```
Error rstudio::launcher_plugins::json::Object::parse (
    const char * in_jsonStr ) [override], [virtual]
```

Parses the JSON string into this object.

## Parameters

<i>in_jsonStr</i>	The JSON string to parse.
-------------------	---------------------------

## Returns

[Success](#) on successful parse when the resulting JSON value is a JSON [Object](#); error otherwise (e.g. [ParseError](#)).

Reimplemented from [rstudio::launcher\\_plugins::json::Value](#).

**4.74.3.33 parse()** [2/2]

```
Error rstudio::launcher_plugins::json::Object::parse (
    const std::string & in_jsonStr ) [override], [virtual]
```

Parses the JSON string into this object.

## Parameters

<i>in_jsonStr</i>	The JSON string to parse.
-------------------	---------------------------

## Returns

[Success](#) on successful parse when the resulting JSON value is a JSON [Object](#); error otherwise (e.g. [ParseError](#)).

Reimplemented from [rstudio::launcher\\_plugins::json::Value](#).

**4.74.3.34 rbegin()**

```
ReverseIterator rstudio::launcher_plugins::json::Object::rbegin ( ) const
```

Gets an iterator pointing to the last member of this object, which iterates in the reverse direction.

## Returns

A reverse iterator pointing to the last member of this object.

#### 4.74.3.35 `rend()`

```
ReverseIterator rstudio::launcher_plugins::json::Object::rend ( ) const
```

Gets an iterator before the first member of this object, which can be compared with an other `ReverseIterator` to determine when reverse iteration has ended.

##### Returns

An iterator before the first member of this object.

#### 4.74.3.36 `toStringMap()`

```
bool rstudio::launcher_plugins::json::Object::toStringMap (
    StringListMap & out_map ) const
```

Converts this JSON object to a map with string keys and a list of string values.

##### Parameters

<code>out_map</code>	The converted map, on success.
----------------------	--------------------------------

##### Returns

True if conversion succeeded; false otherwise.

#### 4.74.3.37 `toStringPairList()`

```
StringPairList rstudio::launcher_plugins::json::Object::toStringPairList ( ) const
```

Converts this JSON object to a list of string pairs.

NOTE: This method will skip any members whose values are not string type.

##### Returns

The string pairs represented in this object.

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

- `/workspaces/rstudio-launcher-plugin-sdk/sdk/include/json/Json.hpp`

## 4.75 rstudio::launcher\_plugins::Optional< T > Class Template Reference

Container class which represents a value that may or may not be set.

```
#include <Optional.hpp>
```

### Public Member Functions

- [Optional](#) ()=default  
*Default constructor.*
- [Optional](#) (T \*in\_value)  
*Constructor.*
- [Optional](#) (const T &in\_value)  
*Constructor.*
- [Optional](#) (const [Optional](#) &in\_other)  
*Copy constructor.*
- [Optional](#) ([Optional](#) &&in\_other) noexcept  
*Move constructor.*
- [operator bool](#) () const  
*Boolean operator.*
- bool [operator!](#) () const  
*Not boolean operator.*
- [Optional](#) & [operator=](#) (const [Optional](#) &in\_other)  
*Assignment operator.*
- [Optional](#) & [operator=](#) ([Optional](#) &&in\_other) noexcept  
*Assignment operator.*
- [Optional](#) & [operator=](#) (T \*in\_value)  
*Assignment operator.*
- [Optional](#) & [operator=](#) (const T &in\_value)  
*Assginment operator.*
- const T & [getValueOr](#) (const T &in\_default) const  
*Gets the value of this optional, or the provided default value if this optional has no value.*
- T & [getValueOr](#) (T &in\_default)  
*Gets the value of this optional, or the provided default value if this optional has no value.*
- bool [hasValue](#) () const  
*Checks whether this optional has a value.*

### 4.75.1 Detailed Description

```
template<typename T>
class rstudio::launcher_plugins::Optional< T >
```

Container class which represents a value that may or may not be set.

#### Template Parameters

<i>T</i>	The type of the optional value.
----------	---------------------------------

## 4.75.2 Constructor & Destructor Documentation

### 4.75.2.1 Optional() [1/4]

```
template<typename T>
rstudio::launcher_plugins::Optional< T >::Optional (
    T * in_value ) [inline], [explicit]
```

Constructor.

#### Parameters

<i>in_value</i>	The value to set on this optional. The optional takes ownership of this value.
-----------------	--

### 4.75.2.2 Optional() [2/4]

```
template<typename T>
rstudio::launcher_plugins::Optional< T >::Optional (
    const T & in_value ) [inline], [explicit]
```

Constructor.

#### Parameters

<i>in_value</i>	The value to set on this optional.
-----------------	------------------------------------

### 4.75.2.3 Optional() [3/4]

```
template<typename T>
rstudio::launcher_plugins::Optional< T >::Optional (
    const Optional< T > & in_other ) [inline]
```

Copy constructor.

#### Parameters

<i>in_other</i>	The optional value to copy.
-----------------	-----------------------------

#### 4.75.2.4 Optional() [4/4]

```
template<typename T>
rstudio::launcher_plugins::Optional< T >::Optional (
    Optional< T > && in_other ) [inline], [noexcept]
```

Move constructor.

##### Parameters

<i>in_other</i>	The optional value to move into this optional value.
-----------------	--

### 4.75.3 Member Function Documentation

#### 4.75.3.1 getValueOr() [1/2]

```
template<typename T>
const T& rstudio::launcher_plugins::Optional< T >::getValueOr (
    const T & in_default ) const [inline]
```

Gets the value of this optional, or the provided default value if this optional has no value.

##### Parameters

<i>in_default</i>	The default value to use if this optional has no value.
-------------------	---

##### Returns

The value of this optional, or the provided default value if this optional has no value.

#### 4.75.3.2 getValueOr() [2/2]

```
template<typename T>
T& rstudio::launcher_plugins::Optional< T >::getValueOr (
    T & in_default ) [inline]
```

Gets the value of this optional, or the provided default value if this optional has no value.

##### Parameters

<i>in_default</i>	The default value to use if this optional has no value.
-------------------	---

**Returns**

The value of this optional, or the provided default value if this optional has no value.

**4.75.3.3 `hasValue()`**

```
template<typename T>
bool rstudio::launcher_plugins::Optional< T >::hasValue ( ) const [inline]
```

Checks whether this optional has a value.

**Returns**

True if this optional has a value; false otherwise.

**4.75.3.4 `operator bool()`**

```
template<typename T>
rstudio::launcher_plugins::Optional< T >::operator bool ( ) const [inline], [explicit]
```

Boolean operator.

**Returns**

True if this optional has a value; false otherwise.

**4.75.3.5 `operator"!()`**

```
template<typename T>
bool rstudio::launcher_plugins::Optional< T >::operator! ( ) const [inline]
```

Not boolean operator.

**Returns**

True if this optional does not have a value; false otherwise.

**4.75.3.6 `operator=()` [1/4]**

```
template<typename T>
Optional& rstudio::launcher_plugins::Optional< T >::operator= (
    const Optional< T > & in_other ) [inline]
```

Assignment operator.

**Parameters**

<i>in_other</i>	The optional value to copy to this optional.
-----------------	--

**Returns**

A reference to this optional.

**4.75.3.7 operator=()** [2/4]

```
template<typename T>
Optional& rstudio::launcher_plugins::Optional< T >::operator= (
    const T & in_value ) [inline]
```

Assginment operator.

**Parameters**

<i>in_value</i>	The value to assign to this optional.
-----------------	---------------------------------------

**Returns**

A reference to this optional.

**4.75.3.8 operator=()** [3/4]

```
template<typename T>
Optional& rstudio::launcher_plugins::Optional< T >::operator= (
    Optional< T > && in_other ) [inline], [noexcept]
```

Assignment operator.

**Parameters**

<i>in_other</i>	The optional value to copy to this optional.
-----------------	--

**Returns**

A reference to this optional.



## 4.75.3.9 operator=() [4/4]

```
template<typename T>
Optional& rstudio::launcher_plugins::Optional< T >::operator= (
    T * in_value ) [inline]
```

Assignment operator.

## Parameters

<i>in_value</i>	The value to assign to this optional. The optional will take ownership of the value.
-----------------	--

## Returns

A reference to this optional.

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

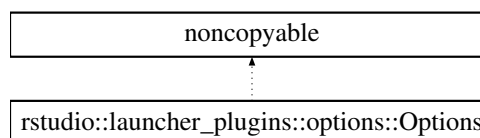
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/Optional.hpp

## 4.76 rstudio::launcher\_plugins::options::Options Class Reference

[Options](#) for the plugin.

```
#include <Options.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::options::Options:



## Classes

- class [Init](#)

*Class for initializing [Options](#).*

## Public Member Functions

- [Init registerOptions](#) ()  
*Allows the caller to register their options using the [Init](#) helper object.*
- [Error readOptions](#) (int in\_argc, const char \*const in\_argv[], const [system::FilePath](#) &in\_location)  
*Reads the option file, loading all registered options.*
- [system::TimeDuration getJobExpiryHours](#) () const  
*Gets the number of hours after which finished jobs expire and should be pruned from the plugin.*
- [system::TimeDuration getHeartbeatIntervalSeconds](#) () const  
*Gets the number of seconds between heartbeats.*
- [logging::LogLevel getLogLevel](#) () const  
*Gets the maximum level of log messages to write.*
- const [system::FilePath](#) & [getLauncherConfigFile](#) () const  
*Gets the location of the configuration file for the RStudio Job Launcher.*
- size\_t [getMaxMessageSize](#) () const  
*Gets the maximum allowable size of messages which can be used in communications with the RStudio Launcher.*
- const std::string & [getPluginName](#) () const  
*Gets the name the administrator gave to this instance of the Plugin in the launcher.conf file.*
- const [system::FilePath](#) & [getRSandboxPath](#) () const  
*Gets the path to the rsandbox executable provided by the RStudio Server Pro installation.*
- const [system::FilePath](#) & [getLoggingDir](#) () const  
*Gets path where debug logs should be written.*
- const [system::FilePath](#) & [getScratchPath](#) () const  
*Gets the scratch path to which log files and other plugin data may be written.*
- [Error getServerUser](#) ([system::User](#) &out\_serverUser) const  
*Gets the user to run as when root privileges are dropped.*
- size\_t [getThreadPoolSize](#) () const  
*Gets the size of the thread pool.*
- bool [useUnprivilegedMode](#) () const  
*Gets whether the plugin should run in single-user unprivileged mode.*
- bool [enableDebugLogging](#) () const  
*Gets whether debug logging is activated.*

## Static Public Member Functions

- static [Options](#) & [getInstance](#) ()  
*Gets the single instance of [Options](#) for the plugin.*

### 4.76.1 Detailed Description

[Options](#) for the plugin.

### 4.76.2 Member Function Documentation

#### 4.76.2.1 enableDebugLogging()

```
bool rstudio::launcher_plugins::options::Options::enableDebugLogging ( ) const
```

Gets whether debug logging is activated.

##### Returns

True if the enableDebugLogging is true; false otherwise.

#### 4.76.2.2 getHeartbeatIntervalSeconds()

```
system::TimeDuration rstudio::launcher_plugins::options::Options::getHeartbeatIntervalSeconds  
( ) const
```

Gets the number of seconds between heartbeats.

##### Returns

The number of seconds between heartbeats.

#### 4.76.2.3 getInstance()

```
static Options& rstudio::launcher_plugins::options::Options::getInstance ( ) [static]
```

Gets the single instance of [Options](#) for the plugin.

##### Returns

The single instance of [Options](#) for the plugin.

#### 4.76.2.4 getJobExpiryHours()

```
system::TimeDuration rstudio::launcher_plugins::options::Options::getJobExpiryHours ( ) const
```

Gets the number of hours after which finished jobs expire and should be pruned from the plugin.

##### Returns

The number of hours after which finished jobs expire and should be pruned from the plugin.

#### 4.76.2.5 getLauncherConfigFile()

```
const system::FilePath& rstudio::launcher_plugins::options::Options::getLauncherConfigFile ( )  
const
```

Gets the location of the configuration file for the RStudio Job Launcher.

This is useful if the plugin implementation requires knowledge of the Job Launcher's configuration. Most plugin implementations will not need this value.

##### Returns

The location of the configuration file for the RStudio Job Launcher.

#### 4.76.2.6 getLoggingDir()

```
const system::FilePath& rstudio::launcher_plugins::options::Options::getLoggingDir ( ) const
```

Gets path where debug logs should be written.

##### Returns

The path where debug logs should be written.

#### 4.76.2.7 getLogLevel()

```
logging::LogLevel rstudio::launcher_plugins::options::Options::getLogLevel ( ) const
```

Gets the maximum level of log messages to write.

##### Returns

The maximum level of log messages to write.

#### 4.76.2.8 getMaxMessageSize()

```
size_t rstudio::launcher_plugins::options::Options::getMaxMessageSize ( ) const
```

Gets the maximum allowable size of messages which can be used in communications with the RStudio Launcher.

It is not recommended to change this value directly in the Plugin's configuration file. Changes to this value will be propagated from the RStudio Launcher to all Plugins.

##### Returns

The maximum allowable size of messages which can be used in communications with the RStudio Launcher.

#### 4.76.2.9 getPluginName()

```
const std::string& rstudio::launcher_plugins::options::Options::getPluginName ( ) const
```

Gets the name the administrator gave to this instance of the Plugin in the launcher.conf file.

This value may be useful if a plugin implementation needs to be aware of other instances of the same Plugin in a load balanced scenario.

##### Returns

The name the administrator gave to this instance of the Plugin in the launcher.conf file.

#### 4.76.2.10 getRSandboxPath()

```
const system::FilePath& rstudio::launcher_plugins::options::Options::getRSandboxPath ( ) const
```

Gets the path to the rsandbox executable provided by the RStudio Server Pro installation.

If RStudio Server Pro is installed to the default location, this value does not need to be set.

##### Returns

The path to the rsandbox executable.

#### 4.76.2.11 getScratchPath()

```
const system::FilePath& rstudio::launcher_plugins::options::Options::getScratchPath ( ) const
```

Gets the scratch path to which log files and other plugin data may be written.

Note that this does not include job output. Job output should be written in the location specified by the user when the job is run.

##### Returns

The scratch path to which log files and other plugin data may be written.

#### 4.76.2.12 getServerUser()

```
Error rstudio::launcher_plugins::options::Options::getServerUser (
    system::User & out_serverUser ) const
```

Gets the user to run as when root privileges are dropped.

## Parameters

<code>out_serverUser</code>	The server user, if it exists.
-----------------------------	--------------------------------

## Returns

[Success](#) if the server user exists; error otherwise.

**4.76.2.13 getThreadPoolSize()**

```
size_t rstudio::launcher_plugins::options::Options::getThreadPoolSize ( ) const
```

Gets the size of the thread pool.

## Returns

The size of the thread pool.

**4.76.2.14 readOptions()**

```
Error rstudio::launcher_plugins::options::Options::readOptions (
    int in_argc,
    const char *const in_argv[],
    const system::FilePath & in_location )
```

Reads the option file, loading all registered options.

[registerOptions\(\)](#) must be called before this is called in order to include additional options.

## Parameters

<code>in_argc</code>	The count of command line arguments.
<code>in_argv</code>	The command line arguments.
<code>in_location</code>	The location of the configuration file. Must exist.

## Returns

[Success](#) if all required options were read and no parsing errors occurred; [Error](#) otherwise.

**4.76.2.15 registerOptions()**

```
Init rstudio::launcher_plugins::options::Options::registerOptions ( )
```

Allows the caller to register their options using the [Init](#) helper object.

**Returns**

The [Init](#) helper object with which options can be registered.

**4.76.2.16 useUnprivilegedMode()**

```
bool rstudio::launcher_plugins::options::Options::useUnprivilegedMode ( ) const
```

Gets whether the plugin should run in single-user unprivileged mode.

**Returns**

True if the plugin should run in uprivileged mode; false otherwise.

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

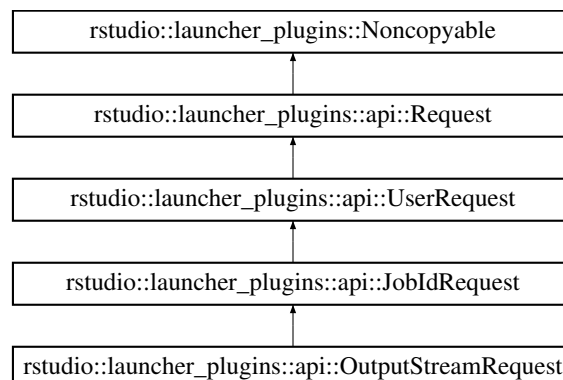
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/options/Options.hpp

## 4.77 rstudio::launcher\_plugins::api::OutputStreamRequest Class Reference

[Request](#) from the launcher to begin or end a [Job](#) Output Stream.

```
#include <Request.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::OutputStreamRequest:

**Public Member Functions**

- OutputType [getStreamType](#) () const  
*Gets the type of Output that should be streamed.*
- bool [isCancelRequest](#) () const  
*Gets whether the [Job](#) Output Stream should be started (false) or ended (true).*

## Friends

- class **Request**

## Additional Inherited Members

### 4.77.1 Detailed Description

[Request](#) from the launcher to begin or end a [Job](#) Output Stream.

### 4.77.2 Member Function Documentation

#### 4.77.2.1 `getStreamType()`

```
OutputType rstudio::launcher_plugins::api::OutputStreamRequest::getStreamType ( ) const
```

Gets the type of Output that should be streamed.

#### Returns

The type of Output that should be streamed.

#### 4.77.2.2 `isCancelRequest()`

```
bool rstudio::launcher_plugins::api::OutputStreamRequest::isCancelRequest ( ) const
```

Gets whether the [Job](#) Output Stream should be started (false) or ended (true).

#### Returns

True if the stream should be canceled; false if it should be started.

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

- `/workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Request.hpp`

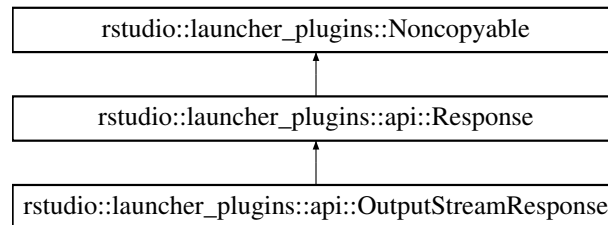


## 4.78 rstudio::launcher\_plugins::api::OutputStreamResponse Class Reference

Class which represents a [Job](#) Output Stream response for a specific job.

```
#include <Response.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::OutputStreamResponse:



### Public Member Functions

- [OutputStreamResponse](#) (uint64\_t in\_requestId, uint64\_t in\_sequenceId, std::string in\_output, OutputType in\_outputType)  
*Constructor.*
- [OutputStreamResponse](#) (uint64\_t in\_requestId, uint64\_t in\_sequenceId)  
*Constructor. Represents the last (complete notification) response of the output stream.*
- [json::Object toJson](#) () const override  
*Converts this output stream response to a JSON object.*

### Additional Inherited Members

#### 4.78.1 Detailed Description

Class which represents a [Job](#) Output Stream response for a specific job.

#### 4.78.2 Constructor & Destructor Documentation

##### 4.78.2.1 OutputStreamResponse() [1/2]

```
rstudio::launcher_plugins::api::OutputStreamResponse::OutputStreamResponse (
    uint64_t in_requestId,
    uint64_t in_sequenceId,
    std::string in_output,
    OutputType in_outputType )
```

Constructor.

## Parameters

<i>in_requestId</i>	The ID of the request for which this response is being sent.
<i>in_↔ sequenceId</i>	The ID of this output in the sequence of responses for this request.
<i>in_output</i>	The output to send to the Launcher.
<i>in_outputType</i>	The type of output being sent.

**4.78.2.2 OutputStreamResponse() [2/2]**

```
rstudio::launcher_plugins::api::OutputStreamResponse::OutputStreamResponse (
    uint64_t in_requestId,
    uint64_t in_sequenceId )
```

Constructor. Represents the last (complete notification) response of the output stream.

## Parameters

<i>in_requestId</i>	The ID of the request for which this response is being sent.
<i>in_↔ sequenceId</i>	The ID of this output in the sequence of responses for this request.

**4.78.3 Member Function Documentation****4.78.3.1 toJson()**

```
json::Object rstudio::launcher_plugins::api::OutputStreamResponse::toJson ( ) const [override],
[virtual]
```

Converts this output stream response to a JSON object.

## Returns

The JSON object which represents this output stream response.

Reimplemented from [rstudio::launcher\\_plugins::api::Response](#).

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

- `/workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Response.hpp`

## 4.79 rstudio::launcher\_plugins::system::PathScopImplDeleter Struct Reference

Struct which implements the deleter for PathScopImpl.

```
#include <FilePath.hpp>
```

### Public Member Functions

- void [operator\(\)](#) (PathScopImpl \*)  
*Deletion operator.*

#### 4.79.1 Detailed Description

Struct which implements the deleter for PathScopImpl.

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/FilePath.hpp

## 4.80 rstudio::launcher\_plugins::api::PlacementConstraint Struct Reference

Struct which represents a custom placement constraint for the job.

```
#include <Job.hpp>
```

### Public Member Functions

- [PlacementConstraint](#) ()=default  
*Default constructor.*
- [PlacementConstraint](#) (std::string in\_name)  
*Constructor.*
- [PlacementConstraint](#) (std::string in\_name, std::string in\_value)  
*Constructor.*
- [json::Object toJson](#) () const  
*Converts this [PlacementConstraint](#) to a JSON object which represents it.*

### Static Public Member Functions

- static [Error fromJson](#) (const [json::Object](#) &in\_json, [PlacementConstraint](#) &out\_placementConstraint)  
*Constructs a [PlacementConstraint](#) from a JSON object which represents the placement constraint.*

## Public Attributes

- std::string [Name](#)
- std::string [Value](#)

### 4.80.1 Detailed Description

Struct which represents a custom placement constraint for the job.

This may be used to allow users to request other resource limits than those supported by [ResourceLimit](#), or it may be used for any other constraint that can affect where a job is run.

There should be a [PlacementConstraint](#) for every value of a given constraint type. For example, if the constraint is the AWS region and the allowed AWS regions are us-east-1, us-west-1, and us-west-2, there should be the following PlacementConstraints in the ClusterInfo response: { "name": "region", "value": "us-east-1" } { "name": "region", "value": "us-west-1" } { "name": "region", "value": "us-west-2" }

For more details, see [ClusterInfoResponse](#) or [PlacementConstraint](#) in the RStudio [Job Launcher Documentation](#): <https://docs.rstudio.com/job-launcher/latest/creating-plugins.html#plugin-messages>.

### 4.80.2 Constructor & Destructor Documentation

#### 4.80.2.1 PlacementConstraint() [1/2]

```
rstudio::launcher_plugins::api::PlacementConstraint::PlacementConstraint (
    std::string in_name ) [explicit]
```

Constructor.

Creates a free-form placement constraint, which allows the user to enter any text value.

Parameters

<i>in_name</i>	The name of the placement constraint.
----------------	---------------------------------------

#### 4.80.2.2 PlacementConstraint() [2/2]

```
rstudio::launcher_plugins::api::PlacementConstraint::PlacementConstraint (
    std::string in_name,
    std::string in_value )
```

Constructor.

Creates an enumeration placement constraint, which allows to

## Parameters

<i>in_name</i>	The name of the placement constraint.
<i>in_value</i>	One of the possible values for the placement constraint with the specified name.

### 4.80.3 Member Function Documentation

#### 4.80.3.1 fromJson()

```
static Error rstudio::launcher_plugins::api::PlacementConstraint::fromJson (
    const json::Object & in_json,
    PlacementConstraint & out_placementConstraint ) [static]
```

Constructs a [PlacementConstraint](#) from a JSON object which represents the placement constraint.

## Parameters

<i>in_json</i>	The JSON object which represents the placement constraint.
<i>out_placementConstraint</i>	The populated placement constraint value. Not valid if an error is returned.

## Returns

[Success](#) if *in\_json* could be parsed as a [PlacementConstraint](#); [Error](#) otherwise.

#### 4.80.3.2 toJson()

```
json::Object rstudio::launcher_plugins::api::PlacementConstraint::toJson ( ) const
```

Converts this [PlacementConstraint](#) to a JSON object which represents it.

## Returns

The JSON object which represents this [PlacementConstraint](#).

### 4.80.4 Member Data Documentation

#### 4.80.4.1 Name

```
std::string rstudio::launcher_plugins::api::PlacementConstraint::Name
```

The name of this placement constraint.

#### 4.80.4.2 Value

```
std::string rstudio::launcher_plugins::api::PlacementConstraint::Value
```

The value of this placement constraint.

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Job.hpp

## 4.81 rstudio::launcher\_plugins::system::process::ProcessInfo Struct Reference

Represents the details of a process that is running on this machine.

```
#include <Process.hpp>
```

### Public Member Functions

- [ProcessInfo](#) ()  
*Constructor.*

### Static Public Member Functions

- static [Error](#) [getProcessInfo](#) (pid\_t in\_pid, [ProcessInfo](#) &out\_info)  
*Gets the process information for the process with the specified PID.*

### Public Attributes

- std::vector< std::string > [Arguments](#)
- std::string [Executable](#)
- [User Owner](#)
- pid\_t [PGrp](#)
- pid\_t [Pid](#)
- pid\_t [PPid](#)
- std::string [State](#)

#### 4.81.1 Detailed Description

Represents the details of a process that is running on this machine.

#### 4.81.2 Member Function Documentation

##### 4.81.2.1 getProcessInfo()

```
static Error rstudio::launcher_plugins::system::process::ProcessInfo::getProcessInfo (
    pid_t in_pid,
    ProcessInfo & out_info ) [static]
```

Gets the process information for the process with the specified PID.

## Parameters

<i>in_pid</i>	The PID of the process for which to retrieve the details.
<i>out_info</i>	The details of the specified process, if no error occurs.

## Returns

[Success](#) if the process information could be retrieved; [Error](#) otherwise.

### 4.81.3 Member Data Documentation

#### 4.81.3.1 Arguments

```
std::vector<std::string> rstudio::launcher_plugins::system::process::ProcessInfo::Arguments
```

The arguments that were passed to the process.

#### 4.81.3.2 Executable

```
std::string rstudio::launcher_plugins::system::process::ProcessInfo::Executable
```

The executable that was run.

#### 4.81.3.3 Owner

```
User rstudio::launcher_plugins::system::process::ProcessInfo::Owner
```

The process' owner.

#### 4.81.3.4 PGrp

```
pid_t rstudio::launcher_plugins::system::process::ProcessInfo::PGrp
```

The process group ID of the process.

#### 4.81.3.5 Pid

```
pid_t rstudio::launcher_plugins::system::process::ProcessInfo::Pid
```

The PID of the process.

#### 4.81.3.6 PPid

```
pid_t rstudio::launcher_plugins::system::process::ProcessInfo::PPid
```

The PID of the process' parent process.

#### 4.81.3.7 State

```
std::string rstudio::launcher_plugins::system::process::ProcessInfo::State
```

The current state of the process.

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/Process.hpp

## 4.82 rstudio::launcher\_plugins::system::process::ProcessOptions Struct Reference

Defines a process that can be run.

```
#include <Process.hpp>
```

### Public Member Functions

- [ProcessOptions](#) ()  
*Constructor.*

### Public Attributes

- std::vector< std::string > [Arguments](#)  
*The arguments of the process. Each argument will be escaped using single quotations so that the values are always interpreted literally. No expansion of environment variables or backslashes will be performed.*
- bool [CloseStdIn](#)  
*Whether to close write end of the standard input stream after the specified StandardInput is written. Default: true.*
- api::EnvironmentList [Environment](#)  
*The environment variables which should be available to the process. If PATH is not set, it will be added to the environment with the same value as the PATH of this process.*
- std::string [Executable](#)  
*The executable or shell command to run. This value should either be an absolute path, or it should be located in one of the locations in the PATH environment variable.*
- bool [IsShellCommand](#)  
*True if the executable is a shell command; False otherwise. Default: false.*
- api::MountList [Mounts](#)  
*The set of mounts to be applied for the child process. Only mounts with a HostMountSource type will be applied. All other mounts will be ignored.*
- std::string [PamProfile](#)  
*The PAM profile to load, if any.*



- `std::string Password`  
*The password of the user running the job, if any.*
- `User RunAsUser`  
*The user who the job should be run as.*
- `std::string StandardInput`  
*The standard input that should be sent to the process.*
- `FilePath StandardOutputFile`  
*The file to which to write standard output. If not set, standard output will not be redirected.*
- `FilePath StandardErrorFile`  
*The file to which to write standard error. If not set, standard error will not be redirected.*
- `bool UseSandbox`  
*Whether to use the rsandbox executable to launch the child in sandbox environment or launch the child process directly.*
- `FilePath WorkingDirectory`  
*The directory from which to run the process. Must exist and be accessible by the RunAsUser.*

### 4.82.1 Detailed Description

Defines a process that can be run.

### 4.82.2 Member Data Documentation

#### 4.82.2.1 CloseStdIn

```
bool rstudio::launcher_plugins::system::process::ProcessOptions::CloseStdIn
```

Whether to close write end of the standard input stream after the specified StandardInput is written. Default: true.

If UseSandbox is true, this value will be ignored and treated as true.

#### 4.82.2.2 Mounts

```
api::MountList rstudio::launcher_plugins::system::process::ProcessOptions::Mounts
```

The set of mounts to be applied for the child process. Only mounts with a HostMountSource type will be applied. All other mounts will be ignored.

Mounts will be ignored if UseSandbox is false.

#### 4.82.2.3 PamProfile

```
std::string rstudio::launcher_plugins::system::process::ProcessOptions::PamProfile
```

The PAM profile to load, if any.

PamProfile will be ignored if UseSandbox is false.

#### 4.82.2.4 Password

```
std::string rstudio::launcher_plugins::system::process::ProcessOptions::Password
```

The password of the user running the job, if any.

Password will be ignored if UseSandbox is false.

#### 4.82.2.5 RunAsUser

```
User rstudio::launcher_plugins::system::process::ProcessOptions::RunAsUser
```

The user who the job should be run as.

The job may fail to run if the RunAsUser is empty.

#### 4.82.2.6 UseSandbox

```
bool rstudio::launcher_plugins::system::process::ProcessOptions::UseSandbox
```

Whether to use the rsandbox executable to launch the child in sandbox environment or launch the child process directly.

If this value is true, CloseStdIn will be ignored and treated as true.

The following values will be ignored if UseSandbox is false:

- Mounts
- PamProfile
- Password

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/Process.hpp

### 4.83 rstudio::launcher\_plugins::system::process::ProcessResult Struct Reference

Represents the result of a synchronous child process.

```
#include <Process.hpp>
```

#### Public Member Functions

- [ProcessResult](#) ()  
*Constructor.*

## Public Attributes

- int [ExitCode](#)
- std::string [StdError](#)
- std::string [StdOut](#)

### 4.83.1 Detailed Description

Represents the result of a synchronous child process.

### 4.83.2 Member Data Documentation

#### 4.83.2.1 ExitCode

```
int rstudio::launcher_plugins::system::process::ProcessResult::ExitCode
```

The exit code of the process.

#### 4.83.2.2 StdError

```
std::string rstudio::launcher_plugins::system::process::ProcessResult::StdError
```

Standard error output from the process.

#### 4.83.2.3 StdOut

```
std::string rstudio::launcher_plugins::system::process::ProcessResult::StdOut
```

Standard output from the process.

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

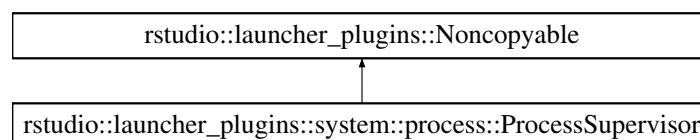
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/Process.hpp

## 4.84 rstudio::launcher\_plugins::system::process::ProcessSupervisor Class Reference

Creates and manages non-blocking child processes.

```
#include <Process.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::system::process::ProcessSupervisor:



## Static Public Member Functions

- static bool `hasRunningChildren` ()  
*Checks whether the supervisor is tracking any processes which have not exited yet.*
- static `Error` `runAsyncProcess` (const `ProcessOptions` &in\_options, const `AsyncProcessCallbacks` &in\_callbacks, std::shared\_ptr< `AbstractChildProcess` > \*out\_childProcess=nullptr)  
*Runs a child process asynchronously.*
- static void `terminateAll` ()  
*Terminates all running children forcefully.*
- static bool `waitForExit` (const `TimeDuration` &in\_maxWaitTime=`TimeDuration::Infinity`())  
*Waits for all child processes to exit.*

## Additional Inherited Members

### 4.84.1 Detailed Description

Creates and manages non-blocking child processes.

### 4.84.2 Member Function Documentation

#### 4.84.2.1 `hasRunningChildren()`

```
static bool rstudio::launcher_plugins::system::process::ProcessSupervisor::hasRunningChildren
( ) [static]
```

Checks whether the supervisor is tracking any processes which have not exited yet.

#### Returns

True if there are any children running; false otherwise.

#### 4.84.2.2 `runAsyncProcess()`

```
static Error rstudio::launcher_plugins::system::process::ProcessSupervisor::runAsyncProcess (
    const ProcessOptions & in_options,
    const AsyncProcessCallbacks & in_callbacks,
    std::shared_ptr< AbstractChildProcess > * out_childProcess = nullptr ) [static]
```

Runs a child process asynchronously.

#### Parameters

<code>in_options</code>	The options for the child process.
<code>in_callbacks</code>	The callbacks to invoke when output is written, an error occurs, or the process exits.
<code>out_childProcess</code>	The child process, if no error occurs on startup.

## Returns

[Success](#) if the child process could be started; [Error](#) otherwise.

## 4.84.2.3 waitForExit()

```
static bool rstudio::launcher_plugins::system::process::ProcessSupervisor::waitForExit (
    const TimeDuration & in_maxWaitTime = TimeDuration::Infinity() ) [static]
```

Waits for all child processes to exit.

## Parameters

<i>in_maxWaitTime</i>	The maximum amount of time to wait for the child processes to exit. Default: no limit.
-----------------------	--

## Returns

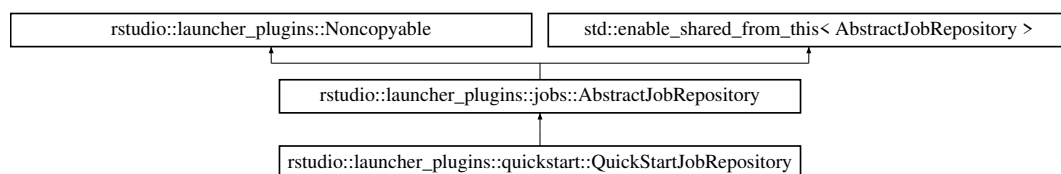
True if the function exited because the timeout was reached; false if the function exited because all child processes exited.

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/Process.hpp

## 4.85 rstudio::launcher\_plugins::quickstart::QuickStartJobRepository Class Reference

Inheritance diagram for rstudio::launcher\_plugins::quickstart::QuickStartJobRepository:



### Public Member Functions

- [QuickStartJobRepository](#) (jobs::JobStatusNotifierPtr in\_notifier)  
*Constructor.*

### 4.85.1 Constructor & Destructor Documentation

#### 4.85.1.1 QuickStartJobRepository()

```
rstudio::launcher_plugins::quickstart::QuickStartJobRepository::QuickStartJobRepository (
    jobs::JobStatusNotifierPtr in_notifier ) [explicit]
```

Constructor.

## Parameters

<code>in_jobStatusNotifier</code>	The job status notifier. Used to add new jobs.
-----------------------------------	--

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

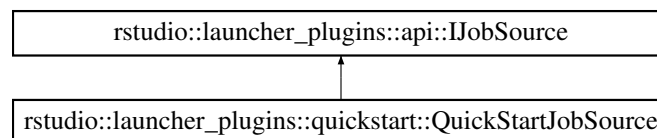
- /workspaces/rstudio-launcher-plugin-sdk/plugins/QuickStart/include/QuickStartJobRepository.hpp

## 4.86 rstudio::launcher\_plugins::quickstart::QuickStartJobSource Class Reference

Class which is responsible for running and retrieving information about jobs in the job scheduling system.

```
#include <QuickStartJobSource.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::quickstart::QuickStartJobSource:



### Public Member Functions

- [QuickStartJobSource](#) (const jobs::JobRepositoryPtr &in\_jobRepository, const jobs::JobStatusNotifierPtr &in\_jobStatusNotifier)  
*Constructor.*
- [Error initialize](#) () override  
*Initializes the Job Source.*
- bool [cancelJob](#) (api::JobPtr in\_job, bool &out\_isComplete, std::string &out\_statusMessage) override  
*Cancels a pending job.*
- [Error getConfiguration](#) (const system::User &in\_user, [api::JobSourceConfiguration](#) &out\_configuration) const override  
*Gets the configuration and capabilities of this Job Source for the specified user.*
- [Error getNetworkInfo](#) (api::JobPtr in\_job, [api::NetworkInfo](#) &out\_networkInfo) const override  
*Gets the network information for the specified job.*
- bool [killJob](#) (api::JobPtr in\_job, bool &out\_isComplete, std::string &out\_statusMessage) override  
*Forcibly kills a running job.*
- bool [resumeJob](#) (api::JobPtr in\_job, bool &out\_isComplete, std::string &out\_statusMessage) override  
*Resumes a suspended job.*
- bool [stopJob](#) (api::JobPtr in\_job, bool &out\_isComplete, std::string &out\_statusMessage) override  
*Stops a running job.*
- bool [suspendJob](#) (api::JobPtr in\_job, bool &out\_isComplete, std::string &out\_statusMessage) override  
*Suspends a running job.*
- [Error submitJob](#) (api::JobPtr io\_job, bool &out\_wasInvalidRequest) const override  
*Submits a job to the Job Scheduling System.*
- [Error createOutputStream](#) (api::OutputType in\_outputType, api::JobPtr in\_job, api::AbstractOutputStream↔::OnOutput in\_onOutput, [api::AbstractOutputStream::OnComplete](#) in\_onComplete, api::AbstractOutput↔Stream::OnError in\_onError, api::OutputStreamPtr &out\_outputStream) override  
*Creates an output stream for the specified job.*
- [Error createResourceStream](#) (api::ConstJobPtr in\_job, comms::AbstractLauncherCommunicatorPtr in\_↔launcherCommunicator, api::AbstractResourceStreamPtr &out\_resourceStream) override  
*Creates a resource utilization metric stream for the specified job.*

## Additional Inherited Members

### 4.86.1 Detailed Description

Class which is responsible for running and retrieving information about jobs in the job scheduling system.

### 4.86.2 Constructor & Destructor Documentation

#### 4.86.2.1 QuickStartJobSource()

```
rstudio::launcher_plugins::quickstart::QuickStartJobSource::QuickStartJobSource (
    const jobs::JobRepositoryPtr & in_jobRepository,
    const jobs::JobStatusNotifierPtr & in_jobStatusNotifier )
```

Constructor.

##### Parameters

<i>in_jobRepository</i>	The job repository, from which to look up jobs.
<i>in_jobStatusNotifier</i>	The job status notifier to which to post or from which to receive job status updates.

### 4.86.3 Member Function Documentation

#### 4.86.3.1 cancelJob()

```
bool rstudio::launcher_plugins::quickstart::QuickStartJobSource::cancelJob (
    api::JobPtr in_job,
    bool & out_isComplete,
    std::string & out_statusMessage ) [override], [virtual]
```

Cancels a pending job.

This method will not be invoked unless the job is currently pending. The Job lock will be held when this method is invoked.

##### Parameters

<i>in_job</i>	The job to be canceled.
<i>out_isComplete</i>	Whether the cancel operation completed successfully (true) or not (false).
<i>out_statusMessage</i>	The status message of the cancel operation, if any.

**Returns**

False if the cancel operation is not supported; true otherwise.

Implements [rstudio::launcher\\_plugins::api::IJobSource](#).

**4.86.3.2 createOutputStream()**

```
Error rstudio::launcher_plugins::quickstart::QuickStartJobSource::createOutputStream (
    api::OutputType in_outputType,
    api::JobPtr in_job,
    api::AbstractOutputStream::OnOutput in_onOutput,
    api::AbstractOutputStream::OnComplete in_onComplete,
    api::AbstractOutputStream::OnError in_onError,
    api::OutputStreamPtr & out_outputStream ) [override], [virtual]
```

Creates an output stream for the specified job.

**Parameters**

<i>in_outputType</i>	The type of job output to stream.
<i>in_job</i>	The job for which output should be streamed.
<i>in_onOutput</i>	Callback function which will be invoked when data is reported.
<i>in_onComplete</i>	Callback function which will be invoked when the stream is complete.
<i>in_onError</i>	Callback function which will be invoked if an error occurs.
<i>out_outputStream</i>	The newly created output stream, on <a href="#">Success</a> .

**Returns**

[Success](#) if the output stream could be created; [Error](#) otherwise.

Implements [rstudio::launcher\\_plugins::api::IJobSource](#).

**4.86.3.3 createResourceStream()**

```
Error rstudio::launcher_plugins::quickstart::QuickStartJobSource::createResourceStream (
    api::ConstJobPtr in_job,
    comms::AbstractLauncherCommunicatorPtr in_launcherCommunicator,
    api::AbstractResourceStreamPtr & out_resourceStream ) [override], [virtual]
```

Creates a resource utilization metric stream for the specified job.

**Parameters**

<i>in_job</i>	The job for which resource utilization metrics should be streamed.
<i>in_launcherCommunicator</i>	The communicator with which to send responses to the Launcher.
<i>out_resourceStream</i>	The newly created resource utilization metric stream, on <a href="#">Success</a> .



**Returns**

Success if the stream could be created; the [Error](#) that occurred otherwise.

Implements [rstudio::launcher\\_plugins::api::IJobSource](#).

**4.86.3.4 getConfiguration()**

```
Error rstudio::launcher_plugins::quickstart::QuickStartJobSource::getConfiguration (
    const system::User & in_user,
    api::JobSourceConfiguration & out_configuration ) const [override], [virtual]
```

Gets the configuration and capabilities of this Job Source for the specified user.

This function controls the options that will be available to users when launching jobs.

NOTE: Many of the values here should most likely be controllable by Launcher administrators when they configure the Launcher. For more details, see the RStudio Launcher Plugin SDK QuickStart Guide TODO #7.

**Parameters**

<i>in_user</i>	The user who made the request to see the configuration and capabilities of the Cluster. This may be used to return a different configuration based on any configured user profiles. For more information about user profiles, see the 'User Profiles' subsection of the 'Advanced Features' section of the RStudio Launcher Plugin SDK Developer's Guide.
<i>out_configuration</i>	The configuration and capabilities of this Job Source, for the specified user.

**Returns**

[Success](#) if the configuration and capabilities for this Job Source could be populated; [Error](#) otherwise.

Implements [rstudio::launcher\\_plugins::api::IJobSource](#).

**4.86.3.5 getNetworkInfo()**

```
Error rstudio::launcher_plugins::quickstart::QuickStartJobSource::getNetworkInfo (
    api::JobPtr in_job,
    api::NetworkInfo & out_networkInfo ) const [override], [virtual]
```

Gets the network information for the specified job.

**Parameters**

<i>in_job</i>	The job for which to retrieve network information.
<i>out_networkInfo</i>	The network information of the specified job, if no error occurred.

**Returns**

[Success](#) if the network information could be retrieved; [Error](#) otherwise.

Implements [rstudio::launcher\\_plugins::api::IJobSource](#).

**4.86.3.6 initialize()**

```
Error rstudio::launcher_plugins::quickstart::QuickStartJobSource::initialize ( ) [override],  
[virtual]
```

Initializes the Job Source.

This function should return an error if communication with the job source fails.

**Returns**

[Success](#) if the job source could be initialized; [Error](#) otherwise.

Implements [rstudio::launcher\\_plugins::api::IJobSource](#).

**4.86.3.7 killJob()**

```
bool rstudio::launcher_plugins::quickstart::QuickStartJobSource::killJob (  
    api::JobPtr in_job,  
    bool & out_isComplete,  
    std::string & out_statusMessage ) [override], [virtual]
```

Forcibly kills a running job.

This method should perform the equivalent of sending a SIGKILL to a process. This method will not be invoked unless the job is currently running. The Job lock will be held when this method is invoked.

**Parameters**

<i>in_job</i>	The job to be killed.
<i>out_isComplete</i>	Whether the kill operation completed successfully (true) or not (false).
<i>out_statusMessage</i>	The status message of the kill operation, if any.

**Returns**

False if the kill operation is not supported; true otherwise.

Implements [rstudio::launcher\\_plugins::api::IJobSource](#).

### 4.86.3.8 resumeJob()

```
bool rstudio::launcher_plugins::quickstart::QuickStartJobSource::resumeJob (
    api::JobPtr in_job,
    bool & out_isComplete,
    std::string & out_statusMessage ) [override], [virtual]
```

Resumes a suspended job.

This method should perform the equivalent of sending a SIGCONT to a process. This method will not be invoked unless the job is currently suspended. The Job lock will be held when this method is invoked.

#### Parameters

<i>in_job</i>	The job to be resumed.
<i>out_isComplete</i>	Whether the stop operation completed successfully (true) or not (false).
<i>out_statusMessage</i>	The status message of the stop operation, if any.

#### Returns

False if the stop operation is not supported; true otherwise.

Implements [rstudio::launcher\\_plugins::api::IJobSource](#).

### 4.86.3.9 stopJob()

```
bool rstudio::launcher_plugins::quickstart::QuickStartJobSource::stopJob (
    api::JobPtr in_job,
    bool & out_isComplete,
    std::string & out_statusMessage ) [override], [virtual]
```

Stops a running job.

This method should perform the equivalent of sending a SIGTERM to a process. This method will not be invoked unless the job is currently running. The Job lock will be held when this method is invoked.

#### Parameters

<i>in_job</i>	The job to be canceled.
<i>out_statusMessage</i>	The status message of the cancel operation, if any.

#### Returns

True if the job was stopped; false otherwise.

Implements [rstudio::launcher\\_plugins::api::IJobSource](#).

#### 4.86.3.10 submitJob()

```
Error rstudio::launcher_plugins::quickstart::QuickStartJobSource::submitJob (
    api::JobPtr io_job,
    bool & out_wasInvalidRequest ) const [override], [virtual]
```

Submits a job to the Job Scheduling System.

##### Parameters

<i>io_job</i>	The Job to be submitted. On successful submission, the Job should be updated with relevant details, such as the ID of the job, the Submission time, the actual Job Queue (if applicable), and the current status.
<i>out_wasInvalidRequest</i>	Whether the requested Job was invalid, based on the features supported by the Job Scheduling System.

##### Returns

**Success** if the job could be submitted to the Job Scheduling System; **Error** otherwise.

Implements [rstudio::launcher\\_plugins::api::IJobSource](#).

#### 4.86.3.11 suspendJob()

```
bool rstudio::launcher_plugins::quickstart::QuickStartJobSource::suspendJob (
    api::JobPtr in_job,
    bool & out_isComplete,
    std::string & out_statusMessage ) [override], [virtual]
```

Suspends a running job.

This method should perform the equivalent of sending a SIGSTOP to a process. A suspended job should be able to be resumed at a later time. This method will not be invoked unless the job is currently running. The Job lock will be held when this method is invoked.

##### Parameters

<i>in_job</i>	The job to be suspended.
<i>out_isComplete</i>	Whether the suspend operation completed successfully (true) or not (false).
<i>out_statusMessage</i>	The status message of the suspend operation, if any.

##### Returns

False if the suspend operation is not supported; true otherwise.

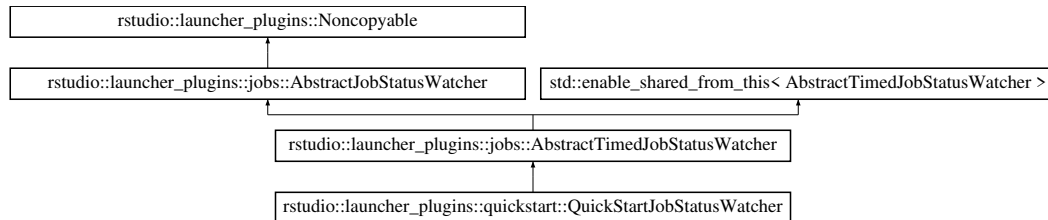
Implements [rstudio::launcher\\_plugins::api::IJobSource](#).

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

- /workspaces/rstudio-launcher-plugin-sdk/plugins/QuickStart/include/QuickStartJobSource.hpp

## 4.87 rstudio::launcher\_plugins::quickstart::QuickStartJobStatusWatcher Class Reference

Inheritance diagram for rstudio::launcher\_plugins::quickstart::QuickStartJobStatusWatcher:



### Public Member Functions

- [QuickStartJobStatusWatcher](#) ([system::TimeDuration](#) in\_frequency, jobs::JobRepositoryPtr in\_jobRepository, jobs::JobStatusNotifierPtr in\_jobStatusNotifier)

*Constructor.*

### Additional Inherited Members

#### 4.87.1 Constructor & Destructor Documentation

##### 4.87.1.1 QuickStartJobStatusWatcher()

```

rstudio::launcher_plugins::quickstart::QuickStartJobStatusWatcher::QuickStartJobStatusWatcher
(
    system::TimeDuration in_frequency,
    jobs::JobRepositoryPtr in_jobRepository,
    jobs::JobStatusNotifierPtr in_jobStatusNotifier )

```

*Constructor.*

#### Parameters

<i>in_frequency</i>	The frequency at which job statuses should be polled.
<i>in_jobRepository</i>	The job repository, from which to look-up jobs.
<i>in_jobStatusNotifier</i>	The job status notifier to which to post job updates.

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

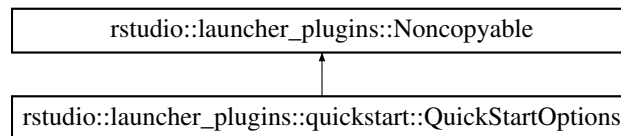
- /workspaces/rstudio-launcher-plugin-sdk/plugins/QuickStart/include/QuickStartJobStatusWatcher.hpp

## 4.88 rstudio::launcher\_plugins::quickstart::QuickStartOptions Class Reference

Class which defines options for the QuickStart Launcher Plugin.

```
#include <QuickStartOptions.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::quickstart::QuickStartOptions:



### Public Member Functions

- bool [getSampleOption](#) () const  
*Gets the value of the sample option.*
- void [initialize](#) ()  
*Method which initializes [QuickStartOptions](#). This method should be called exactly once, before the options file is read.*

### Static Public Member Functions

- static [QuickStartOptions](#) & [getInstance](#) ()  
*Gets the single instance of [QuickStartOptions](#) for the plugin.*

#### 4.88.1 Detailed Description

Class which defines options for the QuickStart Launcher Plugin.

#### 4.88.2 Member Function Documentation

##### 4.88.2.1 getInstance()

```
static QuickStartOptions& rstudio::launcher_plugins::quickstart::QuickStartOptions::get←  
Instance ( ) [static]
```

Gets the single instance of [QuickStartOptions](#) for the plugin.

##### Returns

The single instance of [QuickStartOptions](#) for the plugin.

### 4.88.2.2 getSampleOption()

```
bool rstudio::launcher_plugins::quickstart::QuickStartOptions::getSampleOption ( ) const
```

Gets the value of the sample option.

#### Returns

The value of the sample option.

### 4.88.2.3 initialize()

```
void rstudio::launcher_plugins::quickstart::QuickStartOptions::initialize ( )
```

Method which initializes [QuickStartOptions](#). This method should be called exactly once, before the options file is read.

This is where QuickStart Options are registered with the Options object.

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

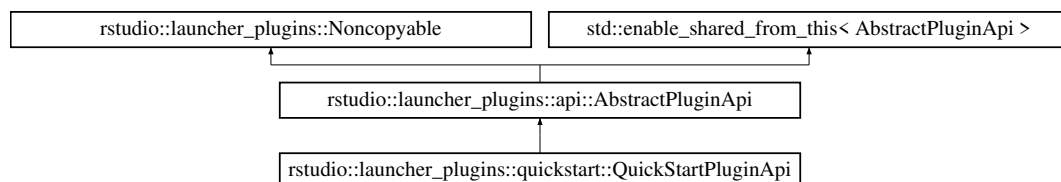
- /workspaces/rstudio-launcher-plugin-sdk/plugins/QuickStart/include/QuickStartOptions.hpp

## 4.89 rstudio::launcher\_plugins::quickstart::QuickStartPluginApi Class Reference

Launcher Plugin API for the QuickStart Plugin.

```
#include <QuickStartPluginApi.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::quickstart::QuickStartPluginApi:



### Public Member Functions

- [QuickStartPluginApi](#) (std::shared\_ptr< [comms::AbstractLauncherCommunicator](#) > in\_launcherCommunicator)  
*Constructor.*

## Additional Inherited Members

### 4.89.1 Detailed Description

Launcher Plugin API for the QuickStart Plugin.

### 4.89.2 Constructor & Destructor Documentation

#### 4.89.2.1 QuickStartPluginApi()

```
rstudio::launcher_plugins::quickstart::QuickStartPluginApi::QuickStartPluginApi (
    std::shared_ptr< comms::AbstractLauncherCommunicator > in_launcherCommunicator )
[explicit]
```

Constructor.

Parameters

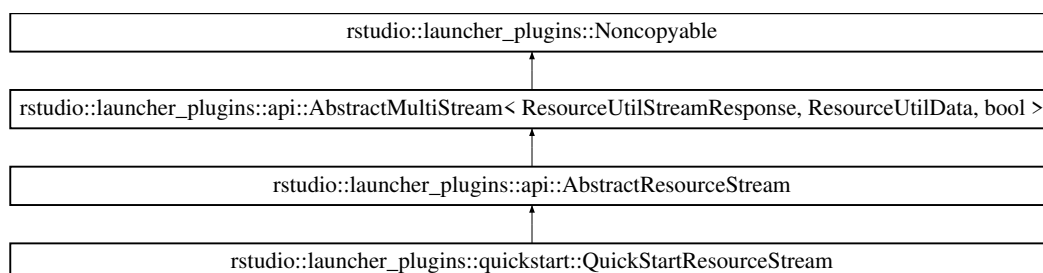
<i>in_launcherCommunicator</i>	The communicator to use for sending and receiving messages from the RStudio Launcher.
--------------------------------	---

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

- /workspaces/rstudio-launcher-plugin-sdk/plugins/QuickStart/include/QuickStartPluginApi.hpp

## 4.90 rstudio::launcher\_plugins::quickstart::QuickStartResourceStream Class Reference

Inheritance diagram for rstudio::launcher\_plugins::quickstart::QuickStartResourceStream:



### Public Member Functions

- [QuickStartResourceStream](#) (const api::ConstJobPtr &in\_job, comms::AbstractLauncherCommunicatorPtr in\_launcherCommunicator)  
*Constructor.*
- [Error initialize](#) () override  
*Initializes the resource utilization stream.*



## Additional Inherited Members

### 4.90.1 Constructor & Destructor Documentation

#### 4.90.1.1 QuickStartResourceStream()

```
rstudio::launcher_plugins::quickstart::QuickStartResourceStream::QuickStartResourceStream (
    const api::ConstJobPtr & in_job,
    comms::AbstractLauncherCommunicatorPtr in_launcherCommunicator )
```

Constructor.

Parameters

<i>in_job</i>	The job for which resource utilization metrics should be streamed.
<i>in_launcherCommunicator</i>	The communicator through which messages may be sent to the launcher.

### 4.90.2 Member Function Documentation

#### 4.90.2.1 initialize()

```
Error rstudio::launcher_plugins::quickstart::QuickStartResourceStream::initialize ( ) [override],
[virtual]
```

Initializes the resource utilization stream.

Returns

[Success](#) if resource utilization streaming was started correctly; [Error](#) otherwise.

Implements [rstudio::launcher\\_plugins::api::AbstractResourceStream](#).

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

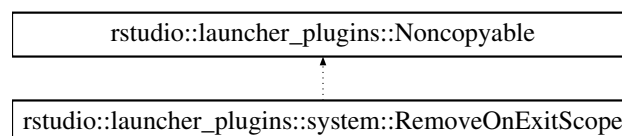
- /workspaces/rstudio-launcher-plugin-sdk/plugins/QuickStart/include/QuickStartResourceStream.hpp

## 4.91 rstudio::launcher\_plugins::system::RemoveOnExitScope Class Reference

RAII class for restoring the current working directory.

```
#include <FilePath.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::system::RemoveOnExitScope:



## Public Member Functions

- [RemoveOnExitScope](#) ([FilePath](#) in\_restorePath, [ErrorLocation](#) in\_location)  
*Constructor.*
- virtual [~RemoveOnExitScope](#) ()  
*Destructor. Removes the path that was provided in the constructor from the filesystem.*

### 4.91.1 Detailed Description

RAII class for restoring the current working directory.

### 4.91.2 Constructor & Destructor Documentation

#### 4.91.2.1 RemoveOnExitScope()

```
rstudio::launcher_plugins::system::RemoveOnExitScope::RemoveOnExitScope (
    FilePath in_restorePath,
    ErrorLocation in_location )
```

Constructor.

#### Parameters

<i>in_restorePath</i>	The path to which to restore the current working directory on destruction of this object.
<i>in_location</i>	The location where this object was constructed, for logging purposes.

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/FilePath.hpp

## 4.92 rstudio::launcher\_plugins::api::Request Class Reference

Base class for all requests which may be received from the Launcher.

```
#include <Request.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::Request:



## Public Types

- enum [Type](#) {  
[Type::HEARTBEAT](#) = 0, [Type::BOOTSTRAP](#) = 1, [Type::SUBMIT\\_JOB](#) = 2, [Type::GET\\_JOB](#) = 3,  
[Type::GET\\_JOB\\_STATUS](#) = 4, [Type::CONTROL\\_JOB](#) = 5, [Type::GET\\_JOB\\_OUTPUT](#) = 6, [Type::GET\\_JOB\\_RESOURCE\\_UTILIZATION](#) = 7,  
[Type::GET\\_JOB\\_NETWORK](#) = 8, [Type::GET\\_CLUSTER\\_INFO](#) = 9, [Type::INVALID](#) }

## Public Member Functions

- virtual [~Request](#) ()=default  
*Virtual destructor for inheritance.*
- uint64\_t [getId](#) () const  
*Gets the ID of this request.*
- [Type](#) [getType](#) () const  
*Gets the request type.*

## Static Public Member Functions

- static [Error fromJson](#) (const [json::Object](#) &in\_requestJson, std::shared\_ptr< [Request](#) > &out\_request)  
*Converts a [Json::Object](#) into the appropriate [Request](#) object.*

## Protected Member Functions

- [Request](#) ([Type](#) in\_requestType, const [json::Object](#) &in\_requestJson)  
*Constructor.*
- PRIVATE\_IMPL** (m\_baseImpl)

### 4.92.1 Detailed Description

Base class for all requests which may be received from the Launcher.

### 4.92.2 Member Enumeration Documentation

---

## Enumerator

---

### 4.92.2.1 Type

```
enum rstudio::launcher_plugins::api::Request::Type [strong]
```

#### Enumerator

HEARTBEAT	Heartbeat request
BOOTSTRAP	Bootstrap request
SUBMIT_JOB	Submit <a href="#">Job</a> request
GET_JOB	Get <a href="#">Job</a> request
GET_JOB_STATUS	Get <a href="#">Job</a> Status request
CONTROL_JOB	Control <a href="#">Job</a> request
GET_JOB_OUTPUT	Get <a href="#">Job</a> Output request
GET_JOB_RESOURCE_UTIL	Get <a href="#">Job</a> Resource Utilization request
GET_JOB_NETWORK	Get <a href="#">Job</a> Network information request
GET_CLUSTER_INFO	Get Cluster Info request
INVALID	Invalid request. Should not be received. Always the last element of this enum for comparison purposes.

### 4.92.3 Constructor & Destructor Documentation

#### 4.92.3.1 Request()

```
rstudio::launcher_plugins::api::Request::Request (
    Type in_requestType,
    const json::Object & in_requestJson ) [explicit], [protected]
```

Constructor.

#### Parameters

<i>in_requestType</i>	The type of the request.
<i>in_requestJson</i>	The JSON object representing the request.

### 4.92.4 Member Function Documentation

#### 4.92.4.1 fromJson()

```
static Error rstudio::launcher_plugins::api::Request::fromJson (
    const json::Object & in_requestJson,
    std::shared_ptr< Request > & out_request ) [static]
```

Converts a [Json::Object](#) into the appropriate [Request](#) object.

##### Parameters

<i>in_requestJson</i>	The json object which represents a request from the Launcher.
<i>out_request</i>	The converted request object.

##### Returns

[Success](#) if the provided json Object was valid; [Error](#) otherwise.

#### 4.92.4.2 getId()

```
uint64_t rstudio::launcher_plugins::api::Request::getId ( ) const
```

Gets the ID of this request.

##### Returns

The ID of this request.

#### 4.92.4.3 getType()

```
Type rstudio::launcher_plugins::api::Request::getType ( ) const
```

Gets the request type.

##### Returns

The type of the request.

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Request.hpp

## 4.93 rstudio::launcher\_plugins::api::ResourceLimit Struct Reference

Struct which represents a resource limit for a job.

```
#include <Job.hpp>
```

## Classes

- struct [Type](#)

## Public Member Functions

- [ResourceLimit](#) ()=default  
*Default constructor.*
- [ResourceLimit](#) (std::string in\_limitType, std::string in\_maxValue="", std::string in\_defaultValue="")  
*Constructor.*
- [json::Object toJson](#) () const  
*Converts this [ResourceLimit](#) to a JSON object which represents it.*

## Static Public Member Functions

- static [Error fromJson](#) (const [json::Object](#) &in\_json, [ResourceLimit](#) &out\_resourceLimit)  
*Constructs a [ResourceLimit](#) from a JSON object which represents the resource limit.*

## Public Attributes

- std::string [ResourceType](#)
- std::string [Value](#)
- std::string [MaxValue](#)
- std::string [DefaultValue](#)

### 4.93.1 Detailed Description

Struct which represents a resource limit for a job.

### 4.93.2 Constructor & Destructor Documentation

#### 4.93.2.1 ResourceLimit()

```
rstudio::launcher_plugins::api::ResourceLimit::ResourceLimit (
    std::string in_limitType,
    std::string in_maxValue = "",
    std::string in_defaultValue = "" ) [explicit]
```

Constructor.

#### Parameters

<i>in_limitType</i>	The type of the resource limit.
<i>in_maxValue</i>	The maximum value of the resource limit. Default: no maximum.
<i>in_defaultValue</i>	The default value of the resource limit. Default: no default.

### 4.93.3 Member Function Documentation

#### 4.93.3.1 fromJson()

```
static Error rstudio::launcher_plugins::api::ResourceLimit::fromJson (
    const json::Object & in_json,
    ResourceLimit & out_resourceLimit ) [static]
```

Constructs a [ResourceLimit](#) from a JSON object which represents the resource limit.

##### Parameters

<i>in_json</i>	The JSON object which represents the resource limit.
<i>out_resourceLimit</i>	The populated resource limit value. Not valid if an error is returned.

##### Returns

[Success](#) if in\_json could be parsed as a [ResourceLimit](#); [Error](#) otherwise.

#### 4.93.3.2 toJson()

```
json::Object rstudio::launcher_plugins::api::ResourceLimit::toJson ( ) const
```

Converts this [ResourceLimit](#) to a JSON object which represents it.

##### Returns

The JSON object which represents this [ResourceLimit](#).

### 4.93.4 Member Data Documentation

#### 4.93.4.1 DefaultValue

```
std::string rstudio::launcher_plugins::api::ResourceLimit::DefaultValue
```

The default value that will be set for this type of resource.

#### 4.93.4.2 MaxValue

```
std::string rstudio::launcher_plugins::api::ResourceLimit::MaxValue
```

The maximum value that can be set for this type of resource.

#### 4.93.4.3 ResourceType

```
std::string rstudio::launcher_plugins::api::ResourceLimit::ResourceType
```

The type of resource to limit.

#### 4.93.4.4 Value

```
std::string rstudio::launcher_plugins::api::ResourceLimit::Value
```

The value of the resource limit.

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Job.hpp

### 4.94 rstudio::launcher\_plugins::api::ResourceUtilData Struct Reference

Represents the current resource utilization of a job.

```
#include <ResponseTypes.hpp>
```

#### Public Attributes

- [Optional](#)< double > [CpuPercent](#)
- [Optional](#)< double > [CpuSeconds](#)
- [Optional](#)< double > [VirtualMem](#)
- [Optional](#)< double > [ResidentMem](#)

#### 4.94.1 Detailed Description

Represents the current resource utilization of a job.

#### 4.94.2 Member Data Documentation

##### 4.94.2.1 CpuPercent

```
Optional<double> rstudio::launcher_plugins::api::ResourceUtilData::CpuPercent
```

The percentage of the CPU(s) that are currently in use by the [Job](#).



#### 4.94.2.2 CpuSeconds

`Optional<double> rstudio::launcher_plugins::api::ResourceUtilData::CpuSeconds`

The total CPU time in seconds that the [Job](#) has used, from the start until now.

#### 4.94.2.3 ResidentMem

`Optional<double> rstudio::launcher_plugins::api::ResourceUtilData::ResidentMem`

The size of resident (actual/physical) memory currently in use by the [Job](#), in MB.

#### 4.94.2.4 VirtualMem

`Optional<double> rstudio::launcher_plugins::api::ResourceUtilData::VirtualMem`

The amount of virtual memory currently in use by the [Job](#), in MB.

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

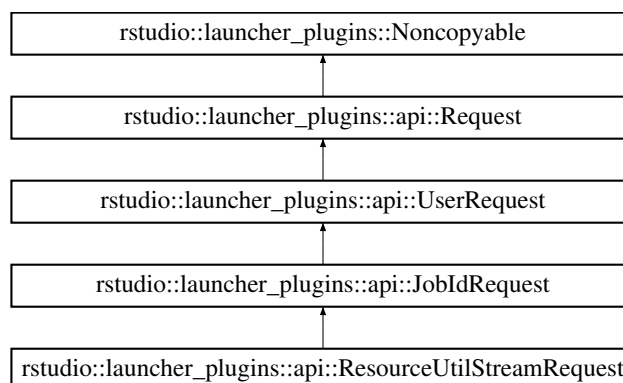
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/ResponseTypes.hpp

## 4.95 rstudio::launcher\_plugins::api::ResourceUtilStreamRequest Class Reference

[Request](#) from the launcher to begin or end a Resource Utilization Stream.

```
#include <Request.hpp>
```

Inheritance diagram for `rstudio::launcher_plugins::api::ResourceUtilStreamRequest`:



### Public Member Functions

- `bool isCancelRequest () const`  
Gets whether the Resource Utilization Stream should be started (false) or ended (true).

## Friends

- class **Request**

## Additional Inherited Members

### 4.95.1 Detailed Description

[Request](#) from the launcher to begin or end a Resource Utilization Stream.

### 4.95.2 Member Function Documentation

#### 4.95.2.1 isCancelRequest()

```
bool rstudio::launcher_plugins::api::ResourceUtilStreamRequest::isCancelRequest ( ) const
```

Gets whether the Resource Utilization Stream should be started (false) or ended (true).

#### Returns

True if the stream should be canceled; false if it should be started.

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

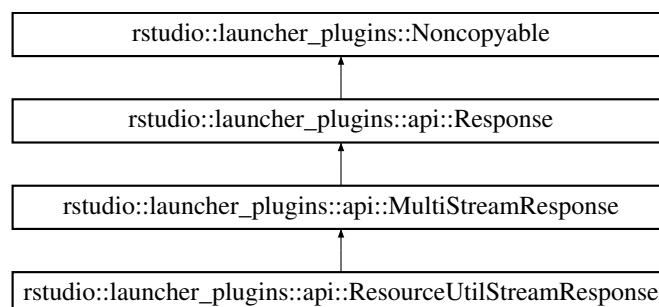
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Request.hpp

## 4.96 rstudio::launcher\_plugins::api::ResourceUtilStreamResponse Class Reference

Class which represents a Resource Utilization Stream response for a specific job.

```
#include <Response.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::ResourceUtilStreamResponse:



## Public Member Functions

- [ResourceUtilStreamResponse](#) (StreamSequences in\_sequences, const [ResourceUtilData](#) &in\_resourceData, bool in\_isComplete)  
*Constructor.*
- [json::Object toJson](#) () const override  
*Converts this resource utilization stream response to a JSON object.*

## Additional Inherited Members

### 4.96.1 Detailed Description

Class which represents a Resource Utilization Stream response for a specific job.

### 4.96.2 Constructor & Destructor Documentation

#### 4.96.2.1 ResourceUtilStreamResponse()

```
rstudio::launcher_plugins::api::ResourceUtilStreamResponse::ResourceUtilStreamResponse (
    StreamSequences in_sequences,
    const ResourceUtilData & in_resourceData,
    bool in_isComplete )
```

Constructor.

#### Parameters

<i>in_sequences</i>	The stream sequences for which this response will be sent.
<i>in_resourceData</i>	The current resource utilization of the job for which the sequenced requests were made.
<i>in_isComplete</i>	Whether the stream is complete (true) or not (false).

### 4.96.3 Member Function Documentation

#### 4.96.3.1 toJson()

```
json::Object rstudio::launcher_plugins::api::ResourceUtilStreamResponse::toJson ( ) const
[override], [virtual]
```

Converts this resource utilization stream response to a JSON object.

**Returns**

The JSON object which represents this resource utilization stream response.

Reimplemented from [rstudio::launcher\\_plugins::api::MultiStreamResponse](#).

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

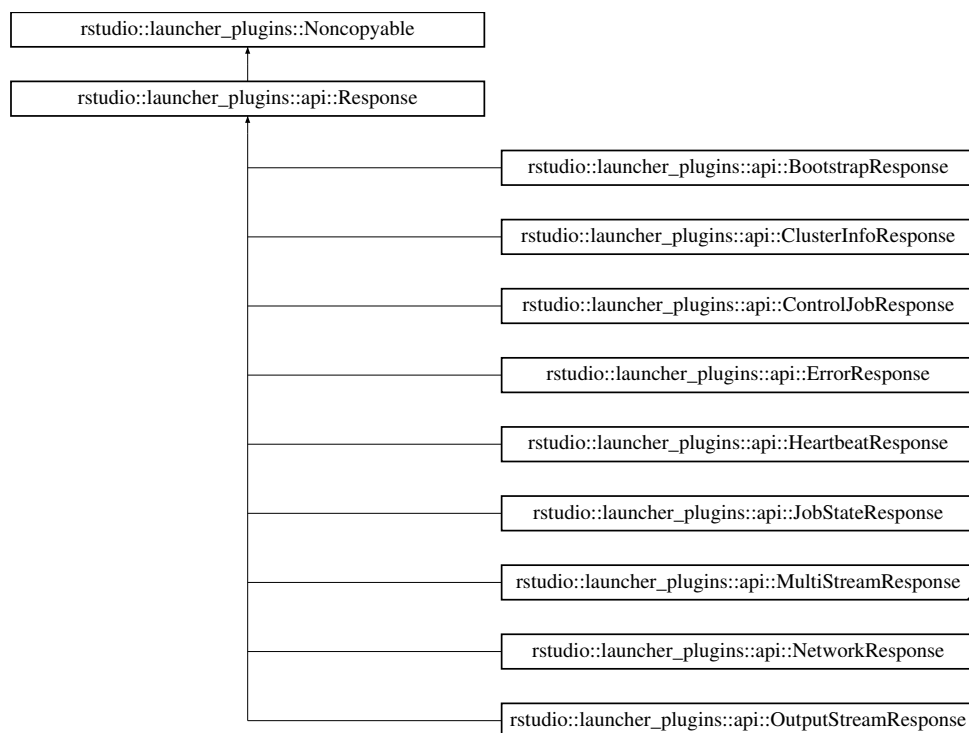
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Response.hpp

## 4.97 rstudio::launcher\_plugins::api::Response Class Reference

Represents the common components of all responses which can be sent the RStudio Launcher.

```
#include <Response.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::Response:



### Public Member Functions

- virtual [~Response](#) ()=default  
*Virtual destructor to allow for inheritance.*
- virtual [json::Object toJson](#) () const  
*Converts this response to a JSON object.*

## Protected Types

- enum [Type](#) {  
[Type::ERROR](#) = -1, [Type::HEARTBEAT](#) = 0, [Type::BOOTSTRAP](#) = 1, [Type::JOB\\_STATE](#) = 2,  
[Type::JOB\\_STATUS](#) = 3, [Type::CONTROL\\_JOB](#) = 4, [Type::JOB\\_OUTPUT](#) = 5, [Type::JOB\\_RESOURCE\\_UTIL](#)  
= 6,  
[Type::JOB\\_NETWORK](#) = 7, [Type::CLUSTER\\_INFO](#) = 8 }

## Protected Member Functions

- [Response](#) ([Type](#) in\_responseType, uint64\_t in\_requestId)  
*Constructor.*

### 4.97.1 Detailed Description

Represents the common components of all responses which can be sent the RStudio Launcher.

### 4.97.2 Member Enumeration Documentation

#### 4.97.2.1 Type

```
enum rstudio::launcher_plugins::api::Response::Type [strong], [protected]
```

##### Enumerator

ERROR	<a href="#">Error</a> response
HEARTBEAT	Heartbeat response
BOOTSTRAP	Bootstrap response
JOB_STATE	<a href="#">Job</a> State response
JOB_STATUS	<a href="#">Job</a> Status response
CONTROL_JOB	Control <a href="#">Job</a> response
JOB_OUTPUT	Control <a href="#">Job</a> output response
JOB_RESOURCE_UTIL	<a href="#">Job</a> Resource utilization response
JOB_NETWORK	<a href="#">Job</a> Network information response
CLUSTER_INFO	Cluster Info response

### 4.97.3 Constructor & Destructor Documentation

#### 4.97.3.1 Response()

```
rstudio::launcher_plugins::api::Response::Response (
    Type in_responseType,
    uint64_t in_requestId ) [protected]
```

Constructor.

##### Parameters

<i>in_responseType</i>	The type of response to be constructed.
<i>in_requestId</i>	The ID of the request for which this response is being sent.

#### 4.97.4 Member Function Documentation

##### 4.97.4.1 toJson()

```
virtual json::Object rstudio::launcher_plugins::api::Response::toJson ( ) const [virtual]
```

Converts this response to a JSON object.

##### Returns

The JSON object which represents this response.

Reimplemented in [rstudio::launcher\\_plugins::api::ClusterInfoResponse](#), [rstudio::launcher\\_plugins::api::NetworkResponse](#), [rstudio::launcher\\_plugins::api::ResourceUtilStreamResponse](#), [rstudio::launcher\\_plugins::api::OutputStreamResponse](#), [rstudio::launcher\\_plugins::api::ControlJobResponse](#), [rstudio::launcher\\_plugins::api::JobStatusResponse](#), [rstudio::launcher\\_plugins::api::BootstrapResponse](#), [rstudio::launcher\\_plugins::api::ErrorResponse](#), and [rstudio::launcher\\_plugins::api::...](#)

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

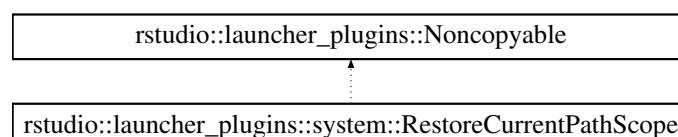
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Response.hpp

## 4.98 rstudio::launcher\_plugins::system::RestoreCurrentPathScope Class Reference

RAII class for restoring the current working directory.

```
#include <FilePath.hpp>
```

Inheritance diagram for `rstudio::launcher_plugins::system::RestoreCurrentPathScope`:



## Public Member Functions

- [RestoreCurrentPathScope](#) ([FilePath](#) in\_restorePath, [ErrorLocation](#) in\_location)  
*Constructor.*
- virtual [~RestoreCurrentPathScope](#) ()  
*Destructor. Returns the working directory to the original path.*

### 4.98.1 Detailed Description

RAII class for restoring the current working directory.

### 4.98.2 Constructor & Destructor Documentation

#### 4.98.2.1 RestoreCurrentPathScope()

```
rstudio::launcher_plugins::system::RestoreCurrentPathScope::RestoreCurrentPathScope (
    FilePath in_restorePath,
    ErrorLocation in_location )
```

Constructor.

#### Parameters

<i>in_restorePath</i>	The path to which to restore the current working directory on destruction of this object.
<i>in_location</i>	The location where this object was constructed, for logging purposes.

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/FilePath.hpp

## 4.99 rstudio::launcher\_plugins::api::StreamSequenceld Class Reference

An identifier for a [MultiStreamResponse](#).

```
#include <ResponseTypes.hpp>
```

## Public Member Functions

- [StreamSequenceld](#) (uint64\_t in\_requestId, uint64\_t in\_sequenceld)  
*Constructor.*
- [StreamSequenceld](#) (const [StreamSequenceld](#) &in\_other)  
*Copy constructor.*
- [StreamSequenceld](#) ([StreamSequenceld](#) &&in\_other) noexcept

- *Move constructor.*
- `~StreamSequenceId()` = default
- *Destructor.*
- `StreamSequenceId & operator= (const StreamSequenceId &in_other)`
- *Assignment operator.*
- `StreamSequenceId & operator= (StreamSequenceId &&in_other) noexcept`
- *Move operator.*
- `json::Object toJson () const`
- *Converts this `StreamSequenceId` to a JSON Object.*

### 4.99.1 Detailed Description

An identifier for a [MultiStreamResponse](#).

### 4.99.2 Constructor & Destructor Documentation

#### 4.99.2.1 StreamSequenceId() [1/3]

```
rstudio::launcher_plugins::api::StreamSequenceId::StreamSequenceId (
    uint64_t in_requestId,
    uint64_t in_sequenceId )
```

Constructor.

Parameters

<i>in_requestId</i>	The ID of the request for which this response will be streamed.
<i>in_↵ sequenceId</i>	The ID of this response in the sequence of streamed responses for the specified request.

#### 4.99.2.2 StreamSequenceId() [2/3]

```
rstudio::launcher_plugins::api::StreamSequenceId::StreamSequenceId (
    const StreamSequenceId & in_other )
```

Copy constructor.

Parameters

<i>in_other</i>	The <a href="#">StreamSequenceId</a> to be copied into this.
-----------------	--



### 4.99.2.3 StreamSequenceId() [3/3]

```
rstudio::launcher_plugins::api::StreamSequenceId::StreamSequenceId (
    StreamSequenceId && in_other ) [noexcept]
```

Move constructor.

#### Parameters

<i>in_other</i>	The <a href="#">StreamSequenceId</a> to be moved into this.
-----------------	---

## 4.99.3 Member Function Documentation

### 4.99.3.1 operator=() [1/2]

```
StreamSequenceId& rstudio::launcher_plugins::api::StreamSequenceId::operator= (
    const StreamSequenceId & in_other )
```

Assignment operator.

#### Parameters

<i>in_other</i>	The <a href="#">StreamSequenceId</a> to be copied into this.
-----------------	--

#### Returns

A reference to this [StreamSequenceId](#).

### 4.99.3.2 operator=() [2/2]

```
StreamSequenceId& rstudio::launcher_plugins::api::StreamSequenceId::operator= (
    StreamSequenceId && in_other ) [noexcept]
```

Move operator.

#### Parameters

<i>in_other</i>	The <a href="#">StreamSequenceId</a> to be moved into this.
-----------------	---

#### Returns

A reference to this [StreamSequenceId](#).

### 4.99.3.3 toJson()

```
json::Object rstudio::launcher_plugins::api::StreamSequenceId::toJson ( ) const
```

Converts this [StreamSequenceId](#) to a JSON Object.

#### Returns

The JSON object which represents this [StreamSequenceId](#).

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

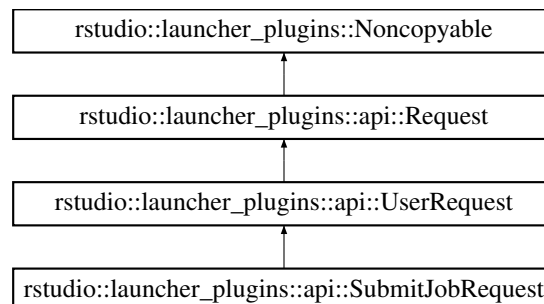
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/ResponseTypes.hpp

## 4.100 rstudio::launcher\_plugins::api::SubmitJobRequest Class Reference

Represents a submit job request from the Launcher.

```
#include <Request.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::SubmitJobRequest:



### Public Member Functions

- JobPtr [getJob](#) ()  
*Gets the job that should be submitted to the [Job](#) Scheduling System.*

### Friends

- class **Request**

### Additional Inherited Members

#### 4.100.1 Detailed Description

Represents a submit job request from the Launcher.

## 4.100.2 Member Function Documentation

### 4.100.2.1 getJob()

```
JobPtr rstudio::launcher_plugins::api::SubmitJobRequest::getJob ( )
```

Gets the job that should be submitted to the [Job](#) Scheduling System.

#### Returns

The job that should be submitted to the [Job](#) Scheduling System.

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

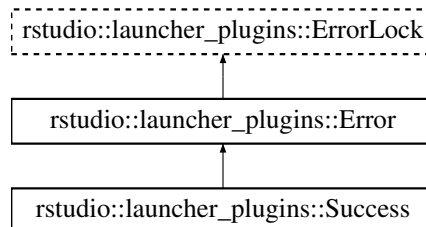
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Request.hpp

## 4.101 rstudio::launcher\_plugins::Success Class Reference

Class which represents a successful operation (i.e. no error).

```
#include <Error.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::Success:



### Public Member Functions

- [Success](#) ()  
*Constructor.*

### 4.101.1 Detailed Description

Class which represents a successful operation (i.e. no error).

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/Error.hpp

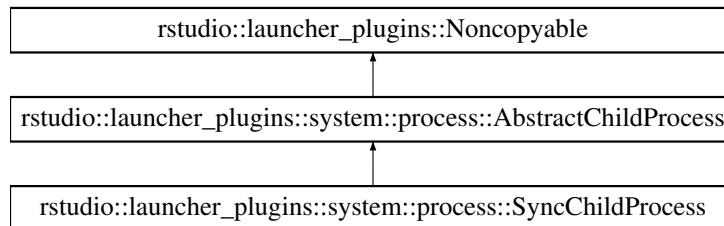
## 4.102 rstudio::launcher\_plugins::system::process::SyncChildProcess

### Class Reference

A blocking child process.

```
#include <Process.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::system::process::SyncChildProcess:



### Public Member Functions

- [SyncChildProcess](#) (const [ProcessOptions](#) &in\_options)  
*Constructor.*
- [Error run](#) ([ProcessResult](#) &out\_result)  
*Runs the child process, blocking until it completes.*
- [Error writeToStdin](#) (const std::string &in\_string, bool in\_eof) override  
*Writes the specified string to stdin.*

### Additional Inherited Members

#### 4.102.1 Detailed Description

A blocking child process.

#### 4.102.2 Constructor & Destructor Documentation

##### 4.102.2.1 SyncChildProcess()

```
rstudio::launcher_plugins::system::process::SyncChildProcess::SyncChildProcess (
    const ProcessOptions & in_options ) [explicit]
```

Constructor.

#### Parameters

<i>in_options</i>	The options for the child process.
-------------------	------------------------------------

## 4.102.3 Member Function Documentation

### 4.102.3.1 run()

```
Error rstudio::launcher_plugins::system::process::SyncChildProcess::run (
    ProcessResult & out_result )
```

Runs the child process, blocking until it completes.

#### Parameters

<i>out_result</i>	The process result.
-------------------	---------------------

#### Returns

[Success](#) if the child process could be started; [Error](#) otherwise.

### 4.102.3.2 writeToStdin()

```
Error rstudio::launcher_plugins::system::process::SyncChildProcess::writeToStdin (
    const std::string & in_string,
    bool in_eof ) [override], [virtual]
```

Writes the specified string to stdin.

#### Parameters

<i>in_string</i>	The data to write to stdin.
<i>in_eof</i>	True if this is the last data to write to stdin.

#### Returns

[Success](#) if the data could be written; [Error](#) otherwise.

Implements [rstudio::launcher\\_plugins::system::process::AbstractChildProcess](#).

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/Process.hpp

## 4.103 rstudio::launcher\_plugins::system::TimeDuration Class Reference

Represents an duration of time (e.g. 5 hours, 43 minutes, and 21 seconds) as opposed to a point in time.

```
#include <DateTime.hpp>
```

## Public Member Functions

- [TimeDuration](#) (int64\_t in\_hours=0, int64\_t in\_minutes=0, int64\_t in\_seconds=0, int64\_t in\_microseconds=0)  
*Constructor.*
- [TimeDuration](#) (const [TimeDuration](#) &in\_other)  
*Copy constructor.*
- [TimeDuration](#) ([TimeDuration](#) &&in\_other) noexcept  
*Move constructor.*
- [~TimeDuration](#) ()=default  
*Destructor.*
- [TimeDuration](#) & operator= (const [TimeDuration](#) &in\_other)  
*Assignment operator.*
- [TimeDuration](#) & operator= ([TimeDuration](#) &&in\_other) noexcept  
*Move operator.*
- bool operator== (const [TimeDuration](#) &in\_other) const  
*Equality comparison operator.*
- bool operator!= (const [TimeDuration](#) &in\_other) const  
*Inequality comparison operator.*
- bool operator< (const [TimeDuration](#) &in\_other) const  
*Less-than comparison operator.*
- bool operator<= (const [TimeDuration](#) &in\_other) const  
*Less-than-equal comparison operator.*
- bool operator> (const [TimeDuration](#) &in\_other) const  
*Greater-than comparison operator.*
- bool operator>= (const [TimeDuration](#) &in\_other) const  
*Greater-than-equal comparison operator.*
- bool isInfinity () const  
*Checks whether this [TimeDuration](#) represents "any amount of time".*
- int64\_t getHours () const  
*Gets the number of hours in this [TimeDuration](#).*
- int64\_t getMinutes () const  
*Gets the number of minutes in this [TimeDuration](#).*
- int64\_t getSeconds () const  
*Gets the number of seconds in this [TimeDuration](#).*
- int64\_t getMicroseconds () const  
*Gets the number of days in this [TimeDuration](#).*

## Static Public Member Functions

- static [TimeDuration](#) Infinity ()  
*Constructs a [TimeDuration](#) which represents "any amount of time". Use with caution.*
- static [TimeDuration](#) Hours (int64\_t in\_hours)  
*Constructs an [TimeDuration](#) which represents the specified number of hours.*
- static [TimeDuration](#) Minutes (int64\_t in\_minutes)  
*Constructs an [TimeDuration](#) which represents the specified number of minutes.*
- static [TimeDuration](#) Seconds (int64\_t in\_seconds)  
*Constructs an [TimeDuration](#) which represents the specified number of seconds.*
- static [TimeDuration](#) Microseconds (int64\_t in\_microseconds)  
*Constructs an [TimeDuration](#) which represents the specified number of microseconds.*

## Friends

- class `DateTime`

### 4.103.1 Detailed Description

Represents an duration of time (e.g. 5 hours, 43 minutes, and 21 seconds) as opposed to a point in time.

### 4.103.2 Constructor & Destructor Documentation

#### 4.103.2.1 TimeDuration() [1/3]

```
rstudio::launcher_plugins::system::TimeDuration::TimeDuration (
    int64_t in_hours = 0,
    int64_t in_minutes = 0,
    int64_t in_seconds = 0,
    int64_t in_microseconds = 0 ) [explicit]
```

Constructor.

##### Parameters

<i>in_hours</i>	The number of hours in this <a href="#">TimeDuration</a> .
<i>in_minutes</i>	The number of minutes in this <a href="#">TimeDuration</a> .
<i>in_seconds</i>	The number of seconds in this <a href="#">TimeDuration</a> .
<i>in_microseconds</i>	The number of microseconds in this <a href="#">TimeDuration</a> .

#### 4.103.2.2 TimeDuration() [2/3]

```
rstudio::launcher_plugins::system::TimeDuration::TimeDuration (
    const TimeDuration & in_other )
```

Copy constructor.

##### Parameters

<i>in_other</i>	The <a href="#">TimeDuration</a> to copy into this <a href="#">TimeDuration</a> .
-----------------	---

#### 4.103.2.3 TimeDuration() [3/3]

```
rstudio::launcher_plugins::system::TimeDuration::TimeDuration (
```

```
TimeDuration && in_other ) [noexcept]
```

Move constructor.

#### Parameters

<code>in_other</code>	The <a href="#">TimeDuration</a> to move into this <a href="#">TimeDuration</a> .
-----------------------	---

### 4.103.3 Member Function Documentation

#### 4.103.3.1 getHours()

```
int64_t rstudio::launcher_plugins::system::TimeDuration::getHours ( ) const
```

Gets the number of hours in this [TimeDuration](#).

#### Returns

The number of hours in this [TimeDuration](#).

#### 4.103.3.2 getMicroseconds()

```
int64_t rstudio::launcher_plugins::system::TimeDuration::getMicroseconds ( ) const
```

Gets the number of days in this [TimeDuration](#).

#### Returns

The number of days in this [TimeDuration](#).

#### 4.103.3.3 getMinutes()

```
int64_t rstudio::launcher_plugins::system::TimeDuration::getMinutes ( ) const
```

Gets the number of minutes in this [TimeDuration](#).

#### Returns

The number of minutes in this [TimeDuration](#).



#### 4.103.3.4 getSeconds()

```
int64_t rstudio::launcher_plugins::system::TimeDuration::getSeconds ( ) const
```

Gets the number of seconds in this [TimeDuration](#).

##### Returns

The number of seconds in this [TimeDuration](#).

#### 4.103.3.5 Hours()

```
static TimeDuration rstudio::launcher_plugins::system::TimeDuration::Hours (
    int64_t in_hours ) [static]
```

Constructs an [TimeDuration](#) which represents the specified number of hours.

##### Parameters

<i>in_hours</i>	The number of hours which should be represented by the <a href="#">TimeDuration</a> .
-----------------	---

##### Returns

The new [TimeDuration](#).

#### 4.103.3.6 Infinity()

```
static TimeDuration rstudio::launcher_plugins::system::TimeDuration::Infinity ( ) [static]
```

Constructs a [TimeDuration](#) which represents "any amount of time". Use with caution.

##### Returns

The new [TimeDuration](#).

#### 4.103.3.7 isInfinity()

```
bool rstudio::launcher_plugins::system::TimeDuration::isInfinity ( ) const
```

Checks whether this [TimeDuration](#) represents "any amount of time".

##### Returns

True if this [TimeDuration](#) is "Infinity"; false otherwise.

#### 4.103.3.8 Microseconds()

```
static TimeDuration rstudio::launcher_plugins::system::TimeDuration::Microseconds (
    int64_t in_microseconds ) [static]
```

Constructs an [TimeDuration](#) which represents the specified number of microseconds.

##### Parameters

<i>in_microseconds</i>	The number of microseconds which should be represented by the <a href="#">TimeDuration</a> .
------------------------	--

##### Returns

The new [TimeDuration](#).

#### 4.103.3.9 Minutes()

```
static TimeDuration rstudio::launcher_plugins::system::TimeDuration::Minutes (
    int64_t in_minutes ) [static]
```

Constructs an [TimeDuration](#) which represents the specified number of minutes.

##### Parameters

<i>in_minutes</i>	The number of minutes which should be represented by the <a href="#">TimeDuration</a> .
-------------------	---

##### Returns

The new [TimeDuration](#).

#### 4.103.3.10 operator"!="()

```
bool rstudio::launcher_plugins::system::TimeDuration::operator!= (
    const TimeDuration & in_other ) const
```

Inequality comparison operator.

##### Parameters

<i>in_other</i>	The <a href="#">TimeDuration</a> to compare against.
-----------------	--

##### Returns

False if *in\_other* has the same values as this [TimeDuration](#); true otherwise.

#### 4.103.3.11 operator<()

```
bool rstudio::launcher_plugins::system::TimeDuration::operator< (
    const TimeDuration & in_other ) const
```

Less-than comparison operator.

##### Parameters

<i>in_other</i>	The <a href="#">TimeDuration</a> to compare against.
-----------------	--

##### Returns

True if this [TimeDuration](#) is less than *in\_other*; false otherwise.

#### 4.103.3.12 operator<=()

```
bool rstudio::launcher_plugins::system::TimeDuration::operator<= (
    const TimeDuration & in_other ) const
```

Less-than-equal comparison operator.

##### Parameters

<i>in_other</i>	The <a href="#">TimeDuration</a> to compare against.
-----------------	--

##### Returns

True if this [TimeDuration](#) is less than or equal to *in\_other*; false otherwise.

#### 4.103.3.13 operator=() [1/2]

```
TimeDuration& rstudio::launcher_plugins::system::TimeDuration::operator= (
    const TimeDuration & in_other )
```

Assignment operator.

##### Parameters

<i>in_other</i>	The <a href="#">TimeDuration</a> to copy into this <a href="#">TimeDuration</a> .
-----------------	---

**Returns**

A reference to this [TimeDuration](#).

**4.103.3.14 operator=()** [2/2]

```
TimeDuration& rstudio::launcher_plugins::system::TimeDuration::operator= (  
    TimeDuration && in_other ) [noexcept]
```

Move operator.

**Parameters**

<i>in_other</i>	The <a href="#">TimeDuration</a> to move into this <a href="#">TimeDuration</a> .
-----------------	---

**Returns**

A reference to this [TimeDuration](#).

**4.103.3.15 operator==()**

```
bool rstudio::launcher_plugins::system::TimeDuration::operator== (  
    const TimeDuration & in_other ) const
```

Equality comparison operator.

**Parameters**

<i>in_other</i>	The <a href="#">TimeDuration</a> to compare against.
-----------------	--

**Returns**

True if *in\_other* has the same values as this [TimeDuration](#); false otherwise.

**4.103.3.16 operator>()**

```
bool rstudio::launcher_plugins::system::TimeDuration::operator> (  
    const TimeDuration & in_other ) const
```

Greater-than comparison operator.

## Parameters

<i>in_other</i>	The <a href="#">TimeDuration</a> to compare against.
-----------------	--

## Returns

True if this [TimeDuration](#) is greater than *in\_other*; false otherwise.

**4.103.3.17 operator>=()**

```
bool rstudio::launcher_plugins::system::TimeDuration::operator>= (
    const TimeDuration & in_other ) const
```

Greater-than-equal comparison operator.

## Parameters

<i>in_other</i>	The <a href="#">TimeDuration</a> to compare against.
-----------------	--

## Returns

True if this [TimeDuration](#) is greater than or equal to *in\_other*; false otherwise.

**4.103.3.18 Seconds()**

```
static TimeDuration rstudio::launcher_plugins::system::TimeDuration::Seconds (
    int64_t in_seconds ) [static]
```

Constructs an [TimeDuration](#) which represents the specified number of seconds.

## Parameters

<i>in_seconds</i>	The number of seconds which should be represented by the <a href="#">TimeDuration</a> .
-------------------	---

## Returns

The new [TimeDuration](#).

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/DateTime.hpp

## 4.104 rstudio::launcher\_plugins::api::ResourceLimit::Type Struct Reference

### Static Public Attributes

- static const char \*const [CPU\\_COUNT](#)
- static const char \*const [CPU\\_TIME](#)
- static const char \*const [MEMORY](#)
- static const char \*const [MEMORY\\_SWAP](#)

### 4.104.1 Member Data Documentation

#### 4.104.1.1 CPU\_COUNT

```
const char* const rstudio::launcher_plugins::api::ResourceLimit::Type::CPU_COUNT [static]
```

The required number of CPUs for a job.

#### 4.104.1.2 CPU\_TIME

```
const char* const rstudio::launcher_plugins::api::ResourceLimit::Type::CPU_TIME [static]
```

The required amount of CPU time for a job, in seconds.

#### 4.104.1.3 MEMORY

```
const char* const rstudio::launcher_plugins::api::ResourceLimit::Type::MEMORY [static]
```

The required amount of memory for a job, in MB.

#### 4.104.1.4 MEMORY\_SWAP

```
const char* const rstudio::launcher_plugins::api::ResourceLimit::Type::MEMORY_SWAP [static]
```

The required amount of swap space for a job, in MB.

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

- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Job.hpp

## 4.105 rstudio::launcher\_plugins::system::User Class Reference

Class which represents a system user.

```
#include <User.hpp>
```

## Public Member Functions

- [User](#) (bool in\_isEmpty=false)  
*Constructor.*
- [User](#) (const [User](#) &in\_other)  
*Copy constructor.*
- [User](#) ([User](#) &&in\_other) noexcept=default  
*Move constructor.*
- [User](#) & [operator=](#) (const [User](#) &in\_other)  
*Overloaded assignment operator.*
- [User](#) & [operator=](#) ([User](#) &&in\_other) noexcept=default  
*Overloaded move operator.*
- bool [operator==](#) (const [User](#) &in\_other) const  
*Equality operator.*
- bool [operator!=](#) (const [User](#) &in\_other) const  
*Inequality operator.*
- bool [exists](#) () const  
*Checks whether the user represented by this object exists.*
- bool [isAllUsers](#) () const  
*Returns whether this object represents all users or not. See the default constructor for more details.*
- bool [isEmpty](#) () const  
*Checks whether this user is empty or not.*
- const [FilePath](#) & [getHomePath](#) () const  
*Gets the user home path, if it exists.*
- [GidType](#) [getGroupId](#) () const  
*Gets the ID of this user's primary group.*
- const std::string & [getShell](#) () const  
*Returns the login shell of this user.*
- [UidType](#) [getUserId](#) () const  
*Gets the ID of this user.*
- const std::string & [getUsername](#) () const  
*Returns the name of this user.*

## Static Public Member Functions

- static [FilePath](#) [getUserHomePath](#) (const std::string &in\_envOverride=std::string())  
*Gets the user home path, as set in the environment.*
- static [Error](#) [getCurrentUser](#) ([User](#) &out\_currentUser)  
*Gets the current user.*
- static [Error](#) [getUserFromIdentifier](#) (const std::string &in\_username, [User](#) &out\_user)  
*Gets a user from its username.*
- static [Error](#) [getUserFromIdentifier](#) ([UidType](#) in\_userId, [User](#) &out\_user)  
*Gets a user from its user ID.*

### 4.105.1 Detailed Description

Class which represents a system user.

## 4.105.2 Constructor & Destructor Documentation

### 4.105.2.1 User() [1/3]

```
rstudio::launcher_plugins::system::User::User (
    bool in_isEmpty = false ) [explicit]
```

Constructor.

Creates a user object which is either empty or represents all users.

Parameters

<i>in_isEmpty</i>	True to create an empty user; False to create a user which represents all users. Default: false.
-------------------	--

### 4.105.2.2 User() [2/3]

```
rstudio::launcher_plugins::system::User::User (
    const User & in_other )
```

Copy constructor.

Parameters

<i>in_other</i>	The user to copy.
-----------------	-------------------

### 4.105.2.3 User() [3/3]

```
rstudio::launcher_plugins::system::User::User (
    User && in_other ) [default], [noexcept]
```

Move constructor.

Parameters

<i>in_other</i>	The user to move into this <a href="#">User</a> .
-----------------	---

## 4.105.3 Member Function Documentation



#### 4.105.3.1 exists()

```
bool rstudio::launcher_plugins::system::User::exists ( ) const
```

Checks whether the user represented by this object exists.

If this is an empty user, or is a user object which represents all users, this method will return false as it does not represent a user which exists on the system.

##### Returns

True if this user exists; false otherwise.

#### 4.105.3.2 getCurrentUser()

```
static Error rstudio::launcher_plugins::system::User::getCurrentUser (
    User & out_currentUser ) [static]
```

Gets the current user.

##### Parameters

<i>out_currentUser</i>	The user this process is currently executing on behalf of. This object will be the empty user if this function returns an error.
------------------------	--

##### Returns

[Success](#) if the user could be retrieved; [Error](#) otherwise.

#### 4.105.3.3 getGroupId()

```
GidType rstudio::launcher_plugins::system::User::getGroupId ( ) const
```

Gets the ID of this user's primary group.

##### Returns

The ID of this user's primary group.

**4.105.3.4 getHomePath()**

```
const FilePath& rstudio::launcher_plugins::system::User::getHomePath ( ) const
```

Gets the user home path, if it exists.

**Returns**

The user's home path, if it exists; empty path otherwise.

**4.105.3.5 getShell()**

```
const std::string& rstudio::launcher_plugins::system::User::getShell ( ) const
```

Returns the login shell of this user.

**Returns**

The login shell of this user.

**4.105.3.6 getUserFromIdentifier() [1/2]**

```
static Error rstudio::launcher_plugins::system::User::getUserFromIdentifier (
    const std::string & in_username,
    User & out_user ) [static]
```

Gets a user from its username.

**Parameters**

<i>in_username</i>	The name of the user to create.
<i>out_user</i>	The created user.

**Returns**

**Success** if the user could be retrieved; **Error** otherwise.

**4.105.3.7 getUserFromIdentifier() [2/2]**

```
static Error rstudio::launcher_plugins::system::User::getUserFromIdentifier (
    UidType in_userId,
    User & out_user ) [static]
```

Gets a user from its user ID.

## Parameters

<i>in_↔ userId</i>	The ID of the user to create.
<i>out_user</i>	The created user.

## Returns

**Success** if the user could be retrieved; **Error** otherwise.

**4.105.3.8** `getUserHomePath()`

```
static FilePath rstudio::launcher_plugins::system::User::getUserHomePath (
    const std::string & in_envOverride = std::string() ) [static]
```

Gets the user home path, as set in the environment.

## Parameters

<i>in_envOverride</i>	If set, overrides the name of the environment variable to use as the user's home path. Multiple overrides may be specified by delimiting them with ' ' in order of precedence.
-----------------------	--

## Returns

The user home path, as set in the environment.

**4.105.3.9** `getUserId()`

```
UidType rstudio::launcher_plugins::system::User::getUserId ( ) const
```

Gets the ID of this user.

## Returns

The ID of this user.

**4.105.3.10** `getUsername()`

```
const std::string& rstudio::launcher_plugins::system::User::getUsername ( ) const
```

Returns the name of this user.

## Returns

The name of this user ("\*" for all users).

#### 4.105.3.11 isAllUsers()

```
bool rstudio::launcher_plugins::system::User::isAllUsers ( ) const
```

Returns whether this object represents all users or not. See the default constructor for more details.

##### Returns

True if this object represents all users; false otherwise.

#### 4.105.3.12 isEmpty()

```
bool rstudio::launcher_plugins::system::User::isEmpty ( ) const
```

Checks whether this user is empty or not.

##### Returns

True if this is user is empty; False otherwise.

#### 4.105.3.13 operator"!=()"

```
bool rstudio::launcher_plugins::system::User::operator!= (
    const User & in_other ) const
```

Inequality operator.

##### Parameters

<i>in_other</i>	The user to compare with this user.
-----------------	-------------------------------------

##### Returns

False if this user and *in\_other* have the same user ID; true otherwise.

#### 4.105.3.14 operator=() [1/2]

```
User& rstudio::launcher_plugins::system::User::operator= (
    const User & in_other )
```

Overloaded assignment operator.

## Parameters

<i>in_other</i>	The user to copy to this one.
-----------------	-------------------------------

## Returns

This user.

**4.105.3.15 operator=()** [2/2]

```
User& rstudio::launcher_plugins::system::User::operator= (  
    User && in_other ) [default], [noexcept]
```

Overloaded move operator.

## Parameters

<i>in_other</i>	The user to move to this one.
-----------------	-------------------------------

## Returns

This user.

**4.105.3.16 operator==()**

```
bool rstudio::launcher_plugins::system::User::operator== (  
    const User & in_other ) const
```

Equality operator.

## Parameters

<i>in_other</i>	The user to compare with this user.
-----------------	-------------------------------------

## Returns

True if this user and *in\_other* have the same user ID; false otherwise.

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

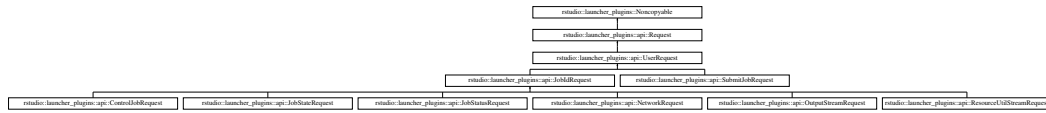
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/User.hpp

## 4.106 rstudio::launcher\_plugins::api::UserRequest Class Reference

Base class which should be used by the class of requests which require a username.

```
#include <Request.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::api::UserRequest:



### Public Member Functions

- const [system::User](#) & [getUser](#) () const  
*Gets the user who initiated this request.*
- const std::string & [getRequestUsername](#) () const  
*Gets the actual username that was used when the request was submitted.*

### Protected Member Functions

- [UserRequest](#) ([Request::Type](#) in\_type, const [json::Object](#) &in\_requestJson)  
*Constructor.*

### Friends

- class [Request](#)

### Additional Inherited Members

#### 4.106.1 Detailed Description

Base class which should be used by the class of requests which require a username.

#### 4.106.2 Constructor & Destructor Documentation

##### 4.106.2.1 UserRequest()

```
rstudio::launcher_plugins::api::UserRequest::UserRequest (
    Request::Type in_type,
    const json::Object & in_requestJson ) [explicit], [protected]
```

Constructor.

## Parameters

<code>in_type</code>	The type of the user request.
<code>in_requestJson</code>	The JSON Object which represents the user request.

### 4.106.3 Member Function Documentation

#### 4.106.3.1 `getRequestUsername()`

```
const std::string& rstudio::launcher_plugins::api::UserRequest::getRequestUsername ( ) const
```

Gets the actual username that was used when the request was submitted.

This value is only useful for auditing purposes and should not be required by most plugins.

## Returns

The actual username that was used when the request was submitted.

#### 4.106.3.2 `getUser()`

```
const system::User& rstudio::launcher_plugins::api::UserRequest::getUser ( ) const
```

Gets the user who initiated this request.

If an admin user made this request, this object may represent all users (check by calling `User::isAllUsers()`). In that case, information for all users should be returned.

## Returns

The user who initiated this request.

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

- `/workspaces/rstudio-launcher-plugin-sdk/sdk/include/api/Request.hpp`

## 4.107 `rstudio::launcher_plugins::options::Value< T >` Class Template Reference

Concrete class which represents an option [Value](#).

```
#include <Options.hpp>
```

## Public Member Functions

- [Value](#) ()  
*Default Constructor.*
- [Value](#) (T &io\_storeTo)  
*Constructor which takes an object to store the value to.*
- [Value](#) & [setDefaultValue](#) (const T &in\_defaultValue)  
*Sets the default value of the option.*

## Friends

- class [Options](#)

### 4.107.1 Detailed Description

```
template<class T>  
class rstudio::launcher_plugins::options::Value< T >
```

Concrete class which represents an option [Value](#).

This class supports the following template types:

- bool
- char
- unsigned char
- short
- unsigned short
- int
- unsigned int
- long
- unsigned long
- long long
- unsigned long long
- float
- double
- long double
- std::string
- rstudio::launcher\_plugins::logging::LogLevel
- [rstudio::launcher\\_plugins::system::FilePath](#)
- [rstudio::launcher\\_plugins::system::User](#)

If a custom type is needed, treat the option value as a string and do the parsing and conversion from the string value.



## Template Parameters

<i>T</i>	The type of the option value.
----------	-------------------------------

## 4.107.2 Constructor & Destructor Documentation

### 4.107.2.1 Value()

```
template<class T>
rstudio::launcher_plugins::options::Value< T >::Value (
    T & io_storeTo ) [explicit]
```

Constructor which takes an object to store the value to.

## Parameters

<i>io_storeTo</i>	The object to store the option value to. The caller is responsible for ensuring that this object is alive when the option file is parsed.
-------------------	---

## 4.107.3 Member Function Documentation

### 4.107.3.1 setDefaultValue()

```
template<class T>
Value& rstudio::launcher_plugins::options::Value< T >::setDefaultValue (
    const T & in_defaultValue )
```

Sets the default value of the option.

## Parameters

<i>in_defaultValue</i>	The default value of the option.
------------------------	----------------------------------

## Returns

A reference to this value.

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

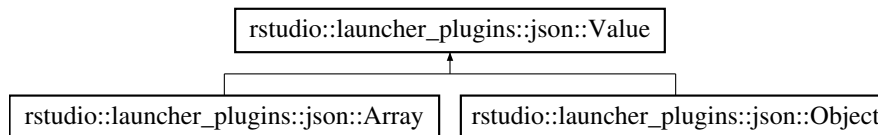
- /workspaces/rstudio-launcher-plugin-sdk/sdk/include/options/Options.hpp

## 4.108 rstudio::launcher\_plugins::json::Value Class Reference

Class which represents a json value.

```
#include <Json.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::json::Value:



### Public Member Functions

- [Value](#) ()  
*Constructor.*
- [Value](#) (ValueImplPtr in\_valueImpl)  
*Constructor. Creates a JSON value from a Value::Impl object.*
- [Value](#) (const [Value](#) &in\_other)  
*Copy constructor.*
- [Value](#) ([Value](#) &&in\_other) noexcept  
*Move constructor.*
- [Value](#) (bool in\_value)  
*Conversion constructor.*
- [Value](#) (double in\_value)  
*Conversion constructor.*
- [Value](#) (float in\_value)  
*Conversion constructor.*
- [Value](#) (int in\_value)  
*Conversion constructor.*
- [Value](#) (int64\_t in\_value)  
*Conversion constructor.*
- [Value](#) (const char \*in\_value)  
*Conversion constructor.*
- [Value](#) (const std::string &in\_value)  
*Conversion constructor.*
- [Value](#) (unsigned int in\_value)  
*Conversion constructor.*
- [Value](#) (uint64\_t in\_value)  
*Conversion constructor.*
- virtual [~Value](#) ()=default  
*Virtual destructor.*
- [Value](#) & [operator=](#) (const [Value](#) &in\_other)  
*Assignment operator from Value.*
- [Value](#) & [operator=](#) ([Value](#) &&in\_other) noexcept  
*Move operator.*
- [Value](#) & [operator=](#) (bool in\_value)  
*Assignment operator.*

- **Value & operator=** (double in\_value)  
*Assignment operator.*
- **Value & operator=** (float in\_value)  
*Assignment operator.*
- **Value & operator=** (int in\_value)  
*Assignment operator.*
- **Value & operator=** (int64\_t in\_value)  
*Assignment operator.*
- **Value & operator=** (const char \*in\_value)  
*Assignment operator.*
- **Value & operator=** (const std::string &in\_value)  
*Assignment operator.*
- **Value & operator=** (unsigned int in\_value)  
*Assignment operator.*
- **Value & operator=** (uint64\_t in\_value)  
*Assignment operator.*
- **bool operator==** (const **Value** &in\_other) const  
*Equality operator.*
- **bool operator!=** (const **Value** &in\_other) const  
*Inequality operator.*
- **Value clone** () const  
*Makes a copy of this JSON value.*
- **Error coerce** (const std::string &in\_schema, std::vector< std::string > &out\_propViolations)  
*Attempts to coerce a JSON object to conform to the given schema by discarding non-conforming properties.*
- **Array getArray** () const  
*Gets the value as a JSON array. If the call to `getType()` does not return `Type::ARRAY`, this method is invalid.*
- **bool getBool** () const  
*Gets the value as a bool. If the call to `getType()` does not return `Type::BOOL`, this method is invalid.*
- **double getDouble** () const  
*Gets the value as a double. If the call to `getType()` does not return `Type::DOUBLE`, this method is invalid.*
- **float getFloat** () const  
*Gets the value as a float. If the call to `getType()` does not return `Type::FLOAT`, this method is invalid.*
- **int getInt** () const  
*Gets the value as an int. If the call to `getType()` does not return `Type::INT`, this method is invalid.*
- **int64\_t getInt64** () const  
*Gets the value as an int64. If the call to `getType()` does not return `Type::INT64`, this method is invalid.*
- **Object getObject** () const  
*Gets the value as a JSON object. If the call to `getType()` does not return `Type::OBJECT`, this method is invalid.*
- **std::string getString** () const  
*Gets the value as a string. If the call to `getType()` does not return `Type::STRING`, this method is invalid.*
- **Type getType** () const  
*Gets the type of this value.*
- **unsigned int getUInt** () const  
*Gets the value as an unsigned int. If the call to `getType()` does not return `Type::UINT`, this method is invalid.*
- **uint64\_t getUInt64** () const  
*Gets the value as an uint64. If the call to `getType()` does not return `Type::UINT64`, this method is invalid.*
- **template<typename T>**  
**T getValue** () const  
*Gets this JSON value as the specified type.*
- **bool isArray** () const  
*Checks whether the value is a JSON array or not.*

- `bool isBool () const`  
*Checks whether the value is a boolean value or not.*
- `bool isDouble () const`  
*Checks whether the value is a double value or not.*
- `bool isFloat () const`  
*Checks whether the value is a float value or not.*
- `bool isInt () const`  
*Checks whether the value is an int 32 value or not.*
- `bool isInt64 () const`  
*Checks whether the value is an int 64 value or not.*
- `bool isObject () const`  
*Checks whether the value is a JSON object or not.*
- `bool isString () const`  
*Checks whether the value is a string value or not.*
- `bool isNull () const`  
*Checks whether the value is null or not.*
- `bool isUInt () const`  
*Checks whether the value is an unsigned int 32 value or not.*
- `bool isUInt64 () const`  
*Checks whether the value is an unsigned int 64 value or not.*
- `virtual Error parse (const char *in_jsonStr)`  
*Parses the JSON string into this value.*
- `virtual Error parse (const std::string &in_jsonStr)`  
*Parses the JSON string into this value.*
- `Error parseAndValidate (const std::string &in_jsonStr, const std::string &in_schema)`  
*Parses the JSON string and validates it against the schema.*
- `Error setValueAtPath (const std::string &in_pointerPath, const json::Value &in_value)`  
*Sets a value within the current value based on the specified JSON Pointer path.*
- `Error setValueAtPath (const std::string &in_pointerPath, bool in_value)`  
*Sets a value within the current value based on the specified JSON Pointer path.*
- `Error setValueAtPath (const std::string &in_pointerPath, double in_value)`  
*Sets a value within the current value based on the specified JSON Pointer path.*
- `Error setValueAtPath (const std::string &in_pointerPath, float in_value)`  
*Sets a value within the current value based on the specified JSON Pointer path.*
- `Error setValueAtPath (const std::string &in_pointerPath, int in_value)`  
*Sets a value within the current value based on the specified JSON Pointer path.*
- `Error setValueAtPath (const std::string &in_pointerPath, int64_t in_value)`  
*Sets a value within the current value based on the specified JSON Pointer path.*
- `Error setValueAtPath (const std::string &in_pointerPath, const char *in_value)`  
*Sets a value within the current value based on the specified JSON Pointer path.*
- `Error setValueAtPath (const std::string &in_pointerPath, const std::string &in_value)`  
*Sets a value within the current value based on the specified JSON Pointer path.*
- `Error setValueAtPath (const std::string &in_pointerPath, unsigned int in_value)`  
*Sets a value within the current value based on the specified JSON Pointer path.*
- `Error setValueAtPath (const std::string &in_pointerPath, uint64_t in_value)`  
*Sets a value within the current value based on the specified JSON Pointer path.*
- `Error setValueAtPath (const std::string &in_pointerPath, const Array &in_value)`  
*Sets a value within the current value based on the specified JSON Pointer path.*
- `Error setValueAtPath (const std::string &in_pointerPath, const Object &in_value)`  
*Sets a value within the current value based on the specified JSON Pointer path.*
- `Error validate (const std::string &in_schema) const`

*Validates this JSON value against a schema.*

- `std::string write () const`

*Writes this value to a string.*

- `void write (std::ostream &io_ostream) const`

*Writes this value to the specified output stream.*

- `std::string writeFormatted () const`

*Writes and formats this value to a string.*

- `void writeFormatted (std::ostream &io_ostream) const`

*Writes and formats this value to the specified output stream.*

## Protected Types

- `typedef std::shared_ptr< Impl > ValueImplPtr`

*Convenience typedef for the type of the private implementation of [json::Value](#).*

## Protected Member Functions

- `PRIVATE_IMPL_SHARED (m_impl)`

*Private implementation of [Value](#).*

## Friends

- class `Array`

### 4.108.1 Detailed Description

Class which represents a json value.

### 4.108.2 Constructor & Destructor Documentation

#### 4.108.2.1 Value() [1/12]

```
rstudio::launcher_plugins::json::Value::Value (
    ValueImplPtr in_valueImpl ) [explicit]
```

Constructor. Creates a JSON value from a Value::Impl object.

Parameters

<i>in_valueImpl</i>	The Value::Impl object to use for the creation of this JSON value.
---------------------	--

#### 4.108.2.2 Value() [2/12]

```
rstudio::launcher_plugins::json::Value::Value (
    const Value & in_other )
```

Copy constructor.

##### Parameters

<i>in_other</i>	The value to copy.
-----------------	--------------------

#### 4.108.2.3 Value() [3/12]

```
rstudio::launcher_plugins::json::Value::Value (
    Value && in_other ) [noexcept]
```

Move constructor.

##### Parameters

<i>in_other</i>	The value to move from.
-----------------	-------------------------

#### 4.108.2.4 Value() [4/12]

```
rstudio::launcher_plugins::json::Value::Value (
    bool in_value ) [explicit]
```

Conversion constructor.

##### Parameters

<i>in_value</i>	The literal value to set this JSON Value to.
-----------------	--

#### 4.108.2.5 Value() [5/12]

```
rstudio::launcher_plugins::json::Value::Value (
    double in_value ) [explicit]
```

Conversion constructor.

## Parameters

<i>in_value</i>	The literal value to set this JSON <a href="#">Value</a> to.
-----------------	--

**4.108.2.6 Value()** [6/12]

```
rstudio::launcher_plugins::json::Value::Value (  
    float in_value ) [explicit]
```

Conversion constructor.

## Parameters

<i>in_value</i>	The literal value to set this JSON <a href="#">Value</a> to.
-----------------	--

**4.108.2.7 Value()** [7/12]

```
rstudio::launcher_plugins::json::Value::Value (  
    int in_value ) [explicit]
```

Conversion constructor.

## Parameters

<i>in_value</i>	The literal value to set this JSON <a href="#">Value</a> to.
-----------------	--

**4.108.2.8 Value()** [8/12]

```
rstudio::launcher_plugins::json::Value::Value (  
    int64_t in_value ) [explicit]
```

Conversion constructor.

## Parameters

<i>in_value</i>	The literal value to set this JSON <a href="#">Value</a> to.
-----------------	--

#### 4.108.2.9 Value() [9/12]

```
rstudio::launcher_plugins::json::Value::Value (
    const char * in_value ) [explicit]
```

Conversion constructor.

##### Parameters

<i>in_value</i>	The literal value to set this JSON <a href="#">Value</a> to.
-----------------	--

#### 4.108.2.10 Value() [10/12]

```
rstudio::launcher_plugins::json::Value::Value (
    const std::string & in_value ) [explicit]
```

Conversion constructor.

##### Parameters

<i>in_value</i>	The literal value to set this JSON <a href="#">Value</a> to.
-----------------	--

#### 4.108.2.11 Value() [11/12]

```
rstudio::launcher_plugins::json::Value::Value (
    unsigned int in_value ) [explicit]
```

Conversion constructor.

##### Parameters

<i>in_value</i>	The literal value to set this JSON <a href="#">Value</a> to.
-----------------	--

#### 4.108.2.12 Value() [12/12]

```
rstudio::launcher_plugins::json::Value::Value (
    uint64_t in_value ) [explicit]
```

Conversion constructor.

##### Parameters

<i>in_value</i>	The literal value to set this JSON <a href="#">Value</a> to.
-----------------	--



### 4.108.3 Member Function Documentation

#### 4.108.3.1 clone()

```
Value rstudio::launcher_plugins::json::Value::clone ( ) const
```

Makes a copy of this JSON value.

##### Returns

A copy of this JSON value.

#### 4.108.3.2 coerce()

```
Error rstudio::launcher_plugins::json::Value::coerce (
    const std::string & in_schema,
    std::vector< std::string > & out_propViolations )
```

Attempts to coerce a JSON object to conform to the given schema by discarding non-conforming properties.

##### Parameters

<i>in_schema</i>	The schema to validate this value against.
<i>out_propViolations</i>	The names of the properties that did not conform to the schema.

##### Returns

[Success](#) if this JSON value matches the schema after coercion; [Error](#) otherwise.

#### 4.108.3.3 getArray()

```
Array rstudio::launcher_plugins::json::Value::getArray ( ) const
```

Gets the value as a JSON array. If the call to [getType\(\)](#) does not return `Type::ARRAY`, this method is invalid.

##### Returns

The value as a JSON array.

#### 4.108.3.4 getBool()

```
bool rstudio::launcher_plugins::json::Value::getBool ( ) const
```

Gets the value as a bool. If the call to [getType\(\)](#) does not return `Type::BOOL`, this method is invalid.

##### Returns

The value as a bool.

#### 4.108.3.5 getDouble()

```
double rstudio::launcher_plugins::json::Value::getDouble ( ) const
```

Gets the value as a double. If the call to [getType\(\)](#) does not return `Type::DOUBLE`, this method is invalid.

##### Returns

The value as a double.

#### 4.108.3.6 getFloat()

```
float rstudio::launcher_plugins::json::Value::getFloat ( ) const
```

Gets the value as a float. If the call to [getType\(\)](#) does not return `Type::FLOAT`, this method is invalid.

##### Returns

The value as a float.

#### 4.108.3.7 getInt()

```
int rstudio::launcher_plugins::json::Value::getInt ( ) const
```

Gets the value as an int. If the call to [getType\(\)](#) does not return `Type::INT`, this method is invalid.

##### Returns

The value as an int.

#### 4.108.3.8 getInt64()

```
int64_t rstudio::launcher_plugins::json::Value::getInt64 ( ) const
```

Gets the value as an int64. If the call to [getType\(\)](#) does not return `Type::INT64`, this method is invalid.

##### Returns

The value as an int64.

#### 4.108.3.9 getObject()

```
Object rstudio::launcher_plugins::json::Value::getObject ( ) const
```

Gets the value as a JSON object. IF the call to [getType\(\)](#) does not return `Type::OBJECT`, this method is invalid.

##### Returns

The value as a JSON object.

#### 4.108.3.10 getString()

```
std::string rstudio::launcher_plugins::json::Value::getString ( ) const
```

Gets the value as a string. If the call to [getType\(\)](#) does not return `Type::STRING`, this method is invalid.

##### Returns

The value as a string.

#### 4.108.3.11 getType()

```
Type rstudio::launcher_plugins::json::Value::getType ( ) const
```

Gets the type of this value.

##### Returns

The type of this value.

#### 4.108.3.12 getUInt()

```
unsigned int rstudio::launcher_plugins::json::Value::getUInt ( ) const
```

Gets the value as an unsigned int. If the call to [getType\(\)](#) does not return `Type::UINT`, this method is invalid.

##### Returns

The value as an unsigned int.

#### 4.108.3.13 getUInt64()

```
uint64_t rstudio::launcher_plugins::json::Value::getUInt64 ( ) const
```

Gets the value as an uint64. If the call to [getType\(\)](#) does not return `Type::UINT64`, this method is invalid.

##### Returns

The value as an uint64.

#### 4.108.3.14 getValue()

```
template<typename T >  
T rstudio::launcher_plugins::json::Value::getValue ( ) const
```

Gets this JSON value as the specified type.

Before calling this method, the appropriate `is<T>` method should return true.

##### Template Parameters

<i>T</i>	The type to retrieve this value as.
----------	-------------------------------------

##### Returns

This value as an object of type `T`.

#### 4.108.3.15 isArray()

```
bool rstudio::launcher_plugins::json::Value::isArray ( ) const
```

Checks whether the value is a JSON array or not.

**Returns**

True if the value is a JSON array; false otherwise.

**4.108.3.16 isBool()**

```
bool rstudio::launcher_plugins::json::Value::isBool ( ) const
```

Checks whether the value is a boolean value or not.

**Returns**

True if the value is a boolean value; false otherwise.

**4.108.3.17 isDouble()**

```
bool rstudio::launcher_plugins::json::Value::isDouble ( ) const
```

Checks whether the value is a double value or not.

**Returns**

True if the value is a double value; false otherwise.

**4.108.3.18 isFloat()**

```
bool rstudio::launcher_plugins::json::Value::isFloat ( ) const
```

Checks whether the value is a float value or not.

**Returns**

True if the value is a float value; false otherwise.

**4.108.3.19 isInt()**

```
bool rstudio::launcher_plugins::json::Value::isInt ( ) const
```

Checks whether the value is an int 32 value or not.

**Returns**

True if the value is an int 32 value; false otherwise.

#### 4.108.3.20 isInt64()

```
bool rstudio::launcher_plugins::json::Value::isInt64 ( ) const
```

Checks whether the value is an int 64 value or not.

##### Returns

True if the value is an int 64 value; false otherwise.

#### 4.108.3.21 isNull()

```
bool rstudio::launcher_plugins::json::Value::isNull ( ) const
```

Checks whether the value is null or not.

##### Returns

True if the value is null; false otherwise.

#### 4.108.3.22 isObject()

```
bool rstudio::launcher_plugins::json::Value::isObject ( ) const
```

Checks whether the value is a JSON object or not.

##### Returns

True if the value is a JSON object; false otherwise.

#### 4.108.3.23 isString()

```
bool rstudio::launcher_plugins::json::Value::isString ( ) const
```

Checks whether the value is a string value or not.

##### Returns

True if the value is a string value; false otherwise.

#### 4.108.3.24 isUInt()

```
bool rstudio::launcher_plugins::json::Value::isUInt ( ) const
```

Checks whether the value is an unsigned int 32 value or not.

##### Returns

True if the value is an unsigned int 32 value; false otherwise.

#### 4.108.3.25 isUInt64()

```
bool rstudio::launcher_plugins::json::Value::isUInt64 ( ) const
```

Checks whether the value is an unsigned int 64 value or not.

##### Returns

True if the value is an unsigned int 64 value; false otherwise.

#### 4.108.3.26 operator"!=()"

```
bool rstudio::launcher_plugins::json::Value::operator!= (
    const Value & in_other ) const
```

Inequality operator.

##### Parameters

<i>in_other</i>	The value to compare this value to.
-----------------	-------------------------------------

##### Returns

True if the two values are not the same; false if they are the same.

#### 4.108.3.27 operator=() [1/11]

```
Value& rstudio::launcher_plugins::json::Value::operator= (
    bool in_value )
```

Assignment operator.

## Parameters

<i>in_value</i>	The literal value to set this JSON <a href="#">Value</a> to.
-----------------	--

## Returns

A reference to this value.

**4.108.3.28 operator=()** [2/11]

```
Value& rstudio::launcher_plugins::json::Value::operator= (
    const char * in_value )
```

Assignment operator.

## Parameters

<i>in_value</i>	The literal value to set this JSON <a href="#">Value</a> to.
-----------------	--

## Returns

A reference to this value.

**4.108.3.29 operator=()** [3/11]

```
Value& rstudio::launcher_plugins::json::Value::operator= (
    const std::string & in_value )
```

Assignment operator.

## Parameters

<i>in_value</i>	The literal value to set this JSON <a href="#">Value</a> to.
-----------------	--

## Returns

A reference to this value.

**4.108.3.30 operator=()** [4/11]

```
Value& rstudio::launcher_plugins::json::Value::operator= (
    const Value & in_other )
```



Assignment operator from [Value](#).

**Parameters**

<i>in_other</i>	The value to copy to this value.
-----------------	----------------------------------

**Returns**

A reference to this value.

**4.108.3.31 operator=()** [5/11]

```
Value& rstudio::launcher_plugins::json::Value::operator= (
    double in_value )
```

Assignment operator.

**Parameters**

<i>in_value</i>	The literal value to set this JSON <a href="#">Value</a> to.
-----------------	--

**Returns**

A reference to this value.

**4.108.3.32 operator=()** [6/11]

```
Value& rstudio::launcher_plugins::json::Value::operator= (
    float in_value )
```

Assignment operator.

**Parameters**

<i>in_value</i>	The literal value to set this JSON <a href="#">Value</a> to.
-----------------	--

**Returns**

A reference to this value.

**4.108.3.33 operator=()** [7/11]

```
Value& rstudio::launcher_plugins::json::Value::operator= (
    int in_value )
```

Assignment operator.

## Parameters

<i>in_value</i>	The literal value to set this JSON <a href="#">Value</a> to.
-----------------	--

## Returns

A reference to this value.

**4.108.3.34 operator=()** [8/11]

```
Value& rstudio::launcher_plugins::json::Value::operator= (
    int64_t in_value )
```

Assignment operator.

## Parameters

<i>in_value</i>	The literal value to set this JSON <a href="#">Value</a> to.
-----------------	--

## Returns

A reference to this value.

**4.108.3.35 operator=()** [9/11]

```
Value& rstudio::launcher_plugins::json::Value::operator= (
    uint64_t in_value )
```

Assignment operator.

## Parameters

<i>in_value</i>	The literal value to set this JSON <a href="#">Value</a> to.
-----------------	--

## Returns

A reference to this value.

**4.108.3.36 operator=()** [10/11]

```
Value& rstudio::launcher_plugins::json::Value::operator= (
    unsigned int in_value )
```

Assignment operator.

**Parameters**

<i>in_value</i>	The literal value to set this JSON <a href="#">Value</a> to.
-----------------	--

**Returns**

A reference to this value.

**4.108.3.37 operator=()** [11/11]

```
Value& rstudio::launcher_plugins::json::Value::operator= (
    Value && in_other ) [noexcept]
```

Move operator.

**Parameters**

<i>in_other</i>	The value to move to this value.
-----------------	----------------------------------

**Returns**

A reference to this value.

**4.108.3.38 operator==()**

```
bool rstudio::launcher_plugins::json::Value::operator== (
    const Value & in_other ) const
```

Equality operator.

**Parameters**

<i>in_other</i>	The value to compare this value to.
-----------------	-------------------------------------

**Returns**

True if the two values are the same; false otherwise.

**4.108.3.39 parse()** [1/2]

```
virtual Error rstudio::launcher_plugins::json::Value::parse (
    const char * in_jsonStr ) [virtual]
```

Parses the JSON string into this value.

## Parameters

<i>in_jsonStr</i>	The JSON string to parse.
-------------------	---------------------------

## Returns

[Success](#) on successful parse; error otherwise (e.g. [ParseError](#))

Reimplemented in [rstudio::launcher\\_plugins::json::Array](#), and [rstudio::launcher\\_plugins::json::Object](#).

**4.108.3.40 parse() [2/2]**

```
virtual Error rstudio::launcher_plugins::json::Value::parse (
    const std::string & in_jsonStr ) [virtual]
```

Parses the JSON string into this value.

## Parameters

<i>in_jsonStr</i>	The JSON string to parse.
-------------------	---------------------------

## Returns

[Success](#) on successful parse; error otherwise (e.g. [ParseError](#))

Reimplemented in [rstudio::launcher\\_plugins::json::Array](#), and [rstudio::launcher\\_plugins::json::Object](#).

**4.108.3.41 parseAndValidate()**

```
Error rstudio::launcher_plugins::json::Value::parseAndValidate (
    const std::string & in_jsonStr,
    const std::string & in_schema )
```

Parses the JSON string and validates it against the schema.

## Parameters

<i>in_jsonStr</i>	The JSON string to parse.
<i>in_schema</i>	The schema to validate the JSON value against.

## Returns

[Success](#) if the string could be parsed and the parsed object matches the schema; [Error](#) otherwise.



**4.108.3.42 setValueAtPath() [1/12]**

```
Error rstudio::launcher_plugins::json::Value::setValueAtPath (
    const std::string & in_pointerPath,
    bool in_value )
```

Sets a value within the current value based on the specified JSON Pointer path.

**Parameters**

<i>in_pointerPath</i>	The JSON Pointer path.
<i>in_value</i>	The JSON value to set at the path. This value is copied before being set.

**Returns**

[Success](#) if the pointer is valid; [Error](#) otherwise.

**4.108.3.43 setValueAtPath() [2/12]**

```
Error rstudio::launcher_plugins::json::Value::setValueAtPath (
    const std::string & in_pointerPath,
    const Array & in_value )
```

Sets a value within the current value based on the specified JSON Pointer path.

**Parameters**

<i>in_pointerPath</i>	The JSON Pointer path.
<i>in_value</i>	The JSON value to set at the path. This value is copied before being set.

**Returns**

[Success](#) if the pointer is valid; [Error](#) otherwise.

**4.108.3.44 setValueAtPath() [3/12]**

```
Error rstudio::launcher_plugins::json::Value::setValueAtPath (
    const std::string & in_pointerPath,
    const char * in_value )
```

Sets a value within the current value based on the specified JSON Pointer path.

**Parameters**

<i>in_pointerPath</i>	The JSON Pointer path.
<i>in_value</i>	The JSON value to set at the path. This value is copied before being set.

## Returns

[Success](#) if the pointer is valid; [Error](#) otherwise.

**4.108.3.45 setValueAtPath()** [4/12]

```
Error rstudio::launcher_plugins::json::Value::setValueAtPath (
    const std::string & in_pointerPath,
    const json::Value & in_value )
```

Sets a value within the current value based on the specified JSON Pointer path.

## Parameters

<i>in_pointerPath</i>	The JSON Pointer path.
<i>in_value</i>	The JSON value to set at the path. This value is copied before being set.

## Returns

[Success](#) if the pointer is valid; [Error](#) otherwise.

**4.108.3.46 setValueAtPath()** [5/12]

```
Error rstudio::launcher_plugins::json::Value::setValueAtPath (
    const std::string & in_pointerPath,
    const Object & in_value )
```

Sets a value within the current value based on the specified JSON Pointer path.

## Parameters

<i>in_pointerPath</i>	The JSON Pointer path.
<i>in_value</i>	The JSON value to set at the path. This value is copied before being set.

## Returns

[Success](#) if the pointer is valid; [Error](#) otherwise.

**4.108.3.47 setValueAtPath()** [6/12]

```
Error rstudio::launcher_plugins::json::Value::setValueAtPath (
    const std::string & in_pointerPath,
    const std::string & in_value )
```

Sets a value within the current value based on the specified JSON Pointer path.

## Parameters

<i>in_pointerPath</i>	The JSON Pointer path.
<i>in_value</i>	The JSON value to set at the path. This value is copied before being set.

## Returns

[Success](#) if the pointer is valid; [Error](#) otherwise.

**4.108.3.48 setValueAtPointerPath()** [7/12]

```
Error rstudio::launcher_plugins::json::Value::setValueAtPointerPath (
    const std::string & in_pointerPath,
    double in_value )
```

Sets a value within the current value based on the specified JSON Pointer path.

## Parameters

<i>in_pointerPath</i>	The JSON Pointer path.
<i>in_value</i>	The JSON value to set at the path. This value is copied before being set.

## Returns

[Success](#) if the pointer is valid; [Error](#) otherwise.

**4.108.3.49 setValueAtPointerPath()** [8/12]

```
Error rstudio::launcher_plugins::json::Value::setValueAtPointerPath (
    const std::string & in_pointerPath,
    float in_value )
```

Sets a value within the current value based on the specified JSON Pointer path.

## Parameters

<i>in_pointerPath</i>	The JSON Pointer path.
<i>in_value</i>	The JSON value to set at the path. This value is copied before being set.

## Returns

[Success](#) if the pointer is valid; [Error](#) otherwise.

**4.108.3.50 setValueAtPath()** [9/12]

```
Error rstudio::launcher_plugins::json::Value::setValueAtPath (
    const std::string & in_pointerPath,
    int in_value )
```

Sets a value within the current value based on the specified JSON Pointer path.

**Parameters**

<i>in_pointerPath</i>	The JSON Pointer path.
<i>in_value</i>	The JSON value to set at the path. This value is copied before being set.

**Returns**

**Success** if the pointer is valid; **Error** otherwise.

**4.108.3.51 setValueAtPath()** [10/12]

```
Error rstudio::launcher_plugins::json::Value::setValueAtPath (
    const std::string & in_pointerPath,
    int64_t in_value )
```

Sets a value within the current value based on the specified JSON Pointer path.

**Parameters**

<i>in_pointerPath</i>	The JSON Pointer path.
<i>in_value</i>	The JSON value to set at the path. This value is copied before being set.

**Returns**

**Success** if the pointer is valid; **Error** otherwise.

**4.108.3.52 setValueAtPath()** [11/12]

```
Error rstudio::launcher_plugins::json::Value::setValueAtPath (
    const std::string & in_pointerPath,
    uint64_t in_value )
```

Sets a value within the current value based on the specified JSON Pointer path.

**Parameters**

<i>in_pointerPath</i>	The JSON Pointer path.
<i>in_value</i>	The JSON value to set at the path. This value is copied before being set.

## Returns

[Success](#) if the pointer is valid; [Error](#) otherwise.

**4.108.3.53 setValueAtPath()** [12/12]

```
Error rstudio::launcher_plugins::json::Value::setValueAtPath (
    const std::string & in_pointerPath,
    unsigned int in_value )
```

Sets a value within the current value based on the specified JSON Pointer path.

## Parameters

<i>in_pointerPath</i>	The JSON Pointer path.
<i>in_value</i>	The JSON value to set at the path. This value is copied before being set.

## Returns

[Success](#) if the pointer is valid; [Error](#) otherwise.

**4.108.3.54 validate()**

```
Error rstudio::launcher_plugins::json::Value::validate (
    const std::string & in_schema ) const
```

Validates this JSON value against a schema.

## Parameters

<i>in_schema</i>	The schema to validate this value against.
------------------	--

## Returns

[Success](#) if this JSON value matches the schema; the [Error](#) that occurred during validation otherwise.

**4.108.3.55 write()** [1/2]

```
std::string rstudio::launcher_plugins::json::Value::write ( ) const
```

Writes this value to a string.

## Returns

The string representation of this value.

**4.108.3.56 write()** [2/2]

```
void rstudio::launcher_plugins::json::Value::write (
    std::ostream & io_ostream ) const
```

Writes this value to the specified output stream.

**Parameters**

<i>io_ostream</i>	The output stream to which to write this value.
-------------------	---

**4.108.3.57 writeFormatted()** [1/2]

```
std::string rstudio::launcher_plugins::json::Value::writeFormatted ( ) const
```

Writes and formats this value to a string.

**Returns**

The formatted string representation of this value.

**4.108.3.58 writeFormatted()** [2/2]

```
void rstudio::launcher_plugins::json::Value::writeFormatted (
    std::ostream & io_ostream ) const
```

Writes and formats this value to the specified output stream.

**Parameters**

<i>io_ostream</i>	The output stream to which to write this value.
-------------------	---

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

- [/workspaces/rstudio-launcher-plugin-sdk/sdk/include/json/Json.hpp](#)

## Chapter 5

# File Documentation

### 5.1 /workspaces/rstudio-launcher-plugin-sdk/sdk/include/json/Json.hpp File Reference

```
#include <map>
#include <ostream>
#include <set>
#include <sstream>
#include <utility>
#include <vector>
#include <Optional.hpp>
#include <Error.hpp>
#include <logging/Logger.hpp>
#include <PImpl.hpp>
```

#### Classes

- class [rstudio::launcher\\_plugins::json::Value](#)  
*Class which represents a json value.*
- class [rstudio::launcher\\_plugins::json::Object](#)  
*Class which represents a specific type of JSON [Value](#): a JSON object.*
- class [rstudio::launcher\\_plugins::json::Object::Member](#)  
*Class which represents a single member of a JSON object.*
- class [rstudio::launcher\\_plugins::json::Object::Iterator](#)  
*Class which allows iterating over the members of a JSON object.*
- class [rstudio::launcher\\_plugins::json::Array](#)  
*Class which represents a JSON array.*
- class [rstudio::launcher\\_plugins::json::Array::Iterator](#)  
*Class which allows iterating over the elements of a JSON array.*
- struct [rstudio::launcher\\_plugins::json::detail::is\\_json\\_type< T >](#)  
*Struct which is either a child class of `std::true_type` or `std::false_type` depending on whether `T` is a JSON type (e.g. [Value](#), [Object](#), [Array](#)) or not (e.g. `int`, `bool`, `string`, `float`, etc.).*

## Typedefs

- typedef std::vector< std::pair< std::string, std::string > > **rstudio::launcher\_plugins::json::StringPair**↔  
**List**
- typedef std::map< std::string, std::vector< std::string > > **rstudio::launcher\_plugins::json::StringList**↔  
**Map**

## Enumerations

- enum **rstudio::launcher\_plugins::json::Type** {  
  **ARRAY, BOOL, INTEGER, OBJECT,**  
  **STRING, REAL, NULL\_TYPE, UNKNOWN** }  
  *Enum which represents the type of a json value.*
- enum **rstudio::launcher\_plugins::json::JsonReadError** { **SUCCESS** = 0, **MISSING\_MEMBER** = 1, **INVALID\_TYPE** = 2 }  
  *Errors which may occur while reading values from JSON objects.*

## Functions

- template<typename T >  
  bool **rstudio::launcher\_plugins::json::isType** (const Value &in\_value)  
    *Checks whether the specified JSON value is of the type specified in the template parameter.*
- std::string **rstudio::launcher\_plugins::json::typeAsString** (Type in\_type)
- std::ostream & **rstudio::launcher\_plugins::json::operator<<** (std::ostream &io\_ostream, Type in\_type)
- template<typename T >  
  Type **rstudio::launcher\_plugins::json::detail::asJsonType** (const T &in\_object, std::true\_type)  
    *Internal utility function. Gets the type of the object as a JSON type, if the object is a JSON type (e.g. [Value](#), [Object](#), [Array](#)).*
- template<typename T >  
  Type **rstudio::launcher\_plugins::json::detail::asJsonType** (const T &in\_object, std::false\_type)  
    *Internal utility function. Gets the type of the object as a JSON type, if the object is not a JSON type (e.g. int, bool, string, float, etc.).*
- template<typename T >  
  Value **rstudio::launcher\_plugins::json::detail::toJsonValue** (const T &in\_value)  
    *Internal utility function. Converts a C/C++ value to a JSON value.*
- template<typename T >  
  Value **rstudio::launcher\_plugins::json::detail::toJsonValue** (const Optional< T > &in\_value)  
    *Internal utility function. Converts a C/C++ optional value to a JSON value.*
- template<typename T >  
  Value **rstudio::launcher\_plugins::json::detail::toJsonValue** (const std::vector< T > &in\_vector)  
    *Internal utility function. Converts a vector value to a JSON array value.*
- template<typename T >  
  Value **rstudio::launcher\_plugins::json::detail::toJsonValue** (const std::set< T > &in\_set)  
    *Internal utility function. Converts a set value to a JSON array value.*
- template<typename T >  
  Type **rstudio::launcher\_plugins::json::asJsonType** (const T &in\_object)  
    *Gets the JSON type of the object.*
- std::string **rstudio::launcher\_plugins::json::typeAsString** (const Value &in\_value)  
    *Gets the type of the JSON value as a string.*
- template<typename T >  
  Value **rstudio::launcher\_plugins::json::toJsonValue** (const T &in\_value)  
    *Converts a C/C++ value to a JSON value.*



- template<typename T >  
Array [rstudio::launcher\\_plugins::json::toJsonArray](#) (const std::vector< T > &in\_vector)  
*Converts a vector value to a JSON array value.*
- template<typename T >  
Array [rstudio::launcher\\_plugins::json::toJsonArray](#) (const std::set< T > &in\_set)  
*Converts a set value to a JSON array value.*
- Error [rstudio::launcher\\_plugins::json::jsonReadError](#) (JsonReadError in\_errorCode, const std::string &in\_message, const ErrorLocation &in\_errorLocation)  
*Creates a JSON read error.*
- bool [rstudio::launcher\\_plugins::json::isMissingMemberError](#) (const Error &in\_error)  
*Checks whether the supplied error is a "missing memeber" error.*
- template<typename T >  
Error [rstudio::launcher\\_plugins::json::readObject](#) (const Object &in\_object, const std::string &in\_name, T &out\_value)  
*Reads a member from an object.*
- template<typename T >  
Error [rstudio::launcher\\_plugins::json::readObject](#) (const Object &in\_object, const std::string &in\_name, Optional< T > &out\_value)  
*Reads a member from an object.*
- template<typename T >  
Error [rstudio::launcher\\_plugins::json::readObject](#) (const Object &in\_object, const std::string &in\_name, std::vector< T > &out\_values)  
*Reads an array member from an object.*
- template<typename T >  
Error [rstudio::launcher\\_plugins::json::readObject](#) (const Object &in\_object, const std::string &in\_name, std::set< T > &out\_values)  
*Reads an array member from an object.*
- template<typename T >  
Error [rstudio::launcher\\_plugins::json::readObject](#) (const Object &in\_object, const std::string &in\_name, Optional< std::vector< T > > &out\_values)  
*Reads an optional array member from an object.*
- template<typename T >  
Error [rstudio::launcher\\_plugins::json::readObject](#) (const Object &in\_object, const std::string &in\_name, Optional< std::set< T > > &out\_values)  
*Reads an optional array member from an object.*
- template<typename T , typename... Args>  
Error [rstudio::launcher\\_plugins::json::readObject](#) (const Object &in\_object, const std::string &in\_name, T &out\_value, Args &... io\_args)  
*Reads multiple members from an object.*
- template<typename T , typename... Args>  
Error [rstudio::launcher\\_plugins::json::readObject](#) (const Object &in\_object, const std::string &in\_name, Optional< T > &out\_value, Args &... io\_args)  
*Reads multiple members from an object.*
- template<typename T , typename... Args>  
Error [rstudio::launcher\\_plugins::json::readObject](#) (const Object &in\_object, const std::string &in\_name, std::vector< T > &out\_values, Args &... io\_args)  
*Reads multiple members from an object.*
- template<typename T , typename... Args>  
Error [rstudio::launcher\\_plugins::json::readObject](#) (const Object &in\_object, const std::string &in\_name, std::set< T > &out\_values, Args &... io\_args)  
*Reads multiple members from an object.*
- template<typename T , typename... Args>  
Error [rstudio::launcher\\_plugins::json::readObject](#) (const Object &in\_object, const std::string &in\_name, Optional< std::vector< T > > &out\_value, Args &... io\_args)

*Reads multiple members from an object.*

- `template<typename T, typename... Args>`  
 Error `rstudio::launcher_plugins::json::readObject` (const Object &in\_object, const std::string &in\_name, Optional< std::set< T > > &out\_value, Args &... io\_args)

*Reads multiple members from an object.*

### 5.1.1 Detailed Description

JSON classes and utility functions.

### 5.1.2 Enumeration Type Documentation

#### 5.1.2.1 Type

```
enum rstudio::launcher_plugins::json::Type [strong]
```

Enum which represents the type of a json value.

Represents the type of error to send to the RStudio Launcher.

Enum which represents the type of a Response.

Enum which represents the type of a Request.

The type of resource limit.

Enum which represents the Type of a JobConfig value.

The last enum value, INVALID, must always be the last value and is used to validate the received request.

Types are defined as described in the RStudio Launcher API Documentation. See <https://docs.rstudio.com/job-launcher/latest/creating-plugins.html#plugin-messages> for more details.

Types are defined as described in the RStudio Launcher API Documentation. See <https://docs.rstudio.com/job-launcher/latest/creating-plugins.html#plugin-messages> for more details.

### 5.1.3 Function Documentation

#### 5.1.3.1 asJsonType() [1/3]

```
template<typename T >
Type rstudio::launcher_plugins::json::asJsonType (
    const T & in_object )
```

Gets the JSON type of the object.

### Template Parameters

<i>T</i>	The C/C++ type of the object.
----------	-------------------------------

### Parameters

<i>in_object</i>	The object for which to retrieve the type.
------------------	--

### Returns

The JSON type of the specified object.

#### 5.1.3.2 asJsonType() [2/3]

```
template<typename T >
Type rstudio::launcher_plugins::json::detail::asJsonType (
    const T & in_object,
    std::false_type )
```

Internal utility function. Gets the type of the object as a JSON type, if the object is not a JSON type (e.g. int, bool, string, float, etc.).

### Template Parameters

<i>T</i>	The type of <i>in_object</i> .
----------	--------------------------------

### Parameters

<i>in_object</i>	The object to get the type of.
------------------	--------------------------------

### Returns

The JSON type of *in\_object*.

#### 5.1.3.3 asJsonType() [3/3]

```
template<typename T >
Type rstudio::launcher_plugins::json::detail::asJsonType (
    const T & in_object,
    std::true_type )
```

Internal utility function. Gets the type of the object as a JSON type, if the object is a JSON type (e.g. Value, Object, Array).

**Template Parameters**

<i>T</i>	The type of <code>in_object</code> .
----------	--------------------------------------

**Parameters**

<i>in_object</i>	The object to get the type of.
------------------	--------------------------------

**Returns**

The JSON type of `in_object`.

**5.1.3.4 isMissingMemberError()**

```
bool rstudio::launcher_plugins::json::isMissingMemberError (
    const Error & in_error )
```

Checks whether the supplied error is a "missing memeber" error.

**Parameters**

<i>in_error</i>	The error to check.
-----------------	---------------------

**Returns**

True if the error is a missing member error; False otherwise.

**5.1.3.5 isType()**

```
template<typename T >
bool rstudio::launcher_plugins::json::isType (
    const Value & in_value )
```

Checks whether the specified JSON value is of the type specified in the template parameter.

**Template Parameters**

<i>T</i>	The type to check the JSON value against.
----------	---

**Parameters**

<i>in_value</i>	The value of which to check the type.
-----------------	---------------------------------------

**Returns**

True if `in_value` is of type `T`; false otherwise.

**5.1.3.6 jsonReadError()**

```
Error rstudio::launcher_plugins::json::jsonReadError (
    JsonReadError in_errorCode,
    const std::string & in_message,
    const ErrorLocation & in_errorLocation )
```

Creates a JSON read error.

**Parameters**

<i>in_errorCode</i>	The code of the error to create.
<i>in_message</i>	The message of the error.
<i>in_errorLocation</i>	The location at which the error occurred.

**Returns**

The newly created JSON read error.

**5.1.3.7 readObject() [1/12]**

```
template<typename T , typename... Args>
Error rstudio::launcher_plugins::json::readObject (
    const Object & in_object,
    const std::string & in_name,
    Optional< std::set< T > > & out_value,
    Args &... io_args )
```

Reads multiple members from an object.

**Template Parameters**

<i>T</i>	The type of the values of the array member to read.
<i>Args</i>	The template parameter pack for the remaining members.

**Parameters**

<i>in_object</i>	The object from which to read the members.
<i>in_name</i>	The name of the first member to be read.
<i>out_values</i>	The set of unique values of the array member to be read, if no error occurs.
<i>io_args</i>	The parameter pack of the remaining members to be read.

**Returns**

Success if the all the elements of the array member have valid types, and if all other members exist and have valid types; Error otherwise.

**5.1.3.8 readObject() [2/12]**

```
template<typename T >
Error rstudio::launcher_plugins::json::readObject (
    const Object & in_object,
    const std::string & in_name,
    Optional< std::set< T > > & out_values )
```

Reads an optional array member from an object.

**Template Parameters**

<i>T</i>	The type of values of the array member.
----------	---

**Parameters**

<i>in_object</i>	The object from which the member should be read.
<i>in_name</i>	The name of the member to read.
<i>out_values</i>	The set of unique values of the array member, if no error occurs.

**Returns**

Success if the values of the member are of type T; Error otherwise.

**5.1.3.9 readObject() [3/12]**

```
template<typename T , typename... Args>
Error rstudio::launcher_plugins::json::readObject (
    const Object & in_object,
    const std::string & in_name,
    Optional< std::vector< T > > & out_value,
    Args &... io_args )
```

Reads multiple members from an object.

**Template Parameters**

<i>T</i>	The type of the values of the array member to read.
<i>Args</i>	The template parameter pack for the remaining members.

**Parameters**

<i>in_object</i>	The object from which to read the members.
<i>in_name</i>	The name of the first member to be read.
<i>out_values</i>	The values of the array member to be read, if no error occurs.
<i>io_args</i>	The parameter pack of the remaining members to be read.

**Returns**

Success if the all the elements of the array member have valid types, and if all other members exist and have valid types; Error otherwise.

**5.1.3.10 readObject() [4/12]**

```
template<typename T >
Error rstudio::launcher_plugins::json::readObject (
    const Object & in_object,
    const std::string & in_name,
    Optional< std::vector< T > > & out_values )
```

Reads an optional array member from an object.

**Template Parameters**

<i>T</i>	The type of values of the array member.
----------	---

**Parameters**

<i>in_object</i>	The object from which the member should be read.
<i>in_name</i>	The name of the member to read.
<i>out_values</i>	The values of the array member, if no error occurs.

**Returns**

Success if the values of the member are of type T; Error otherwise.

**5.1.3.11 readObject() [5/12]**

```
template<typename T >
Error rstudio::launcher_plugins::json::readObject (
    const Object & in_object,
    const std::string & in_name,
    Optional< T > & out_value )
```

Reads a member from an object.

## Template Parameters

<i>T</i>	The type of the member.
----------	-------------------------

## Parameters

<i>in_object</i>	The object from which the member should be read.
<i>in_name</i>	The name of the member to read.
<i>out_value</i>	The value of the member, if no error occurs.

## Returns

Success if the member could be found and is of type *T*; Error otherwise.

## 5.1.3.12 readObject() [6/12]

```
template<typename T , typename... Args>
Error rstudio::launcher_plugins::json::readObject (
    const Object & in_object,
    const std::string & in_name,
    Optional< T > & out_value,
    Args &... io_args )
```

Reads multiple members from an object.

## Template Parameters

<i>T</i>	The type of the first member to read.
<i>Args</i>	The template parameter pack for the remaining members.

## Parameters

<i>in_object</i>	The object from which to read the members.
<i>in_name</i>	The name of the first member to be read.
<i>out_value</i>	The value of the first member to be read, if no error occurs.
<i>io_args</i>	The parameter pack of the remaining members to be read.

## Returns

Success if all the members exist and have valid types; Error otherwise.

## 5.1.3.13 readObject() [7/12]

```
template<typename T >
Error rstudio::launcher_plugins::json::readObject (
```



```
const Object & in_object,
const std::string & in_name,
std::set< T > & out_values )
```

Reads an array member from an object.

#### Template Parameters

<i>T</i>	The type of values of the array member.
----------	---

#### Parameters

<i>in_object</i>	The object from which the member should be read.
<i>in_name</i>	The name of the member to read.
<i>out_values</i>	The set of unique values of the array member, if no error occurs.

#### Returns

Success if the member could be found and its values are of type *T*; Error otherwise.

#### 5.1.3.14 readObject() [8/12]

```
template<typename T , typename... Args>
Error rstudio::launcher_plugins::json::readObject (
    const Object & in_object,
    const std::string & in_name,
    std::set< T > & out_values,
    Args &... io_args )
```

Reads multiple members from an object.

#### Template Parameters

<i>T</i>	The type of the values of the array member to read.
<i>Args</i>	The template parameter pack for the remaining members.

#### Parameters

<i>in_object</i>	The object from which to read the members.
<i>in_name</i>	The name of the first member to be read.
<i>out_values</i>	The set of unique values of the array member to be read, if no error occurs.
<i>io_args</i>	The parameter pack of the remaining members to be read.

#### Returns

Success if the array member exists and all its elements have valid types, and if all other members exist and have valid types; Error otherwise.

### 5.1.3.15 readObject() [9/12]

```
template<typename T >
Error rstudio::launcher_plugins::json::readObject (
    const Object & in_object,
    const std::string & in_name,
    std::vector< T > & out_values )
```

Reads an array member from an object.

#### Template Parameters

<i>T</i>	The type of values of the array member.
----------	---

#### Parameters

<i>in_object</i>	The object from which the member should be read.
<i>in_name</i>	The name of the member to read.
<i>out_values</i>	The values of the array member, if no error occurs.

#### Returns

Success if the member could be found and its values are of type T; Error otherwise.

### 5.1.3.16 readObject() [10/12]

```
template<typename T , typename... Args>
Error rstudio::launcher_plugins::json::readObject (
    const Object & in_object,
    const std::string & in_name,
    std::vector< T > & out_values,
    Args &... io_args )
```

Reads multiple members from an object.

#### Template Parameters

<i>T</i>	The type of the values of the array member to read.
<i>Args</i>	The template parameter pack for the remaining members.

#### Parameters

<i>in_object</i>	The object from which to read the members.
<i>in_name</i>	The name of the first member to be read.
<i>out_values</i>	The values of the array member to be read, if no error occurs.

## Parameters

<i>io_args</i>	The parameter pack of the remaining members to be read.
----------------	---

## Returns

Success if the array member exists and all its elements have valid types, and if all other members exist and have valid types; Error otherwise.

**5.1.3.17 readObject()** [11/12]

```
template<typename T >
Error rstudio::launcher_plugins::json::readObject (
    const Object & in_object,
    const std::string & in_name,
    T & out_value )
```

Reads a member from an object.

## Template Parameters

<i>T</i>	The type of the member.
----------	-------------------------

## Parameters

<i>in_object</i>	The object from which the member should be read.
<i>in_name</i>	The name of the member to read.
<i>out_value</i>	The value of the member, if no error occurs.

## Returns

Success if the member could be found and is of type T; Error otherwise.

**5.1.3.18 readObject()** [12/12]

```
template<typename T , typename... Args>
Error rstudio::launcher_plugins::json::readObject (
    const Object & in_object,
    const std::string & in_name,
    T & out_value,
    Args &... io_args )
```

Reads multiple members from an object.

**Template Parameters**

<i>T</i>	The type of the first member to read.
<i>Args</i>	The template parameter pack for the remaining members.

**Parameters**

<i>in_object</i>	The object from which to read the members.
<i>in_name</i>	The name of the first member to be read.
<i>out_value</i>	The value of the first member to be read, if no error occurs.
<i>io_args</i>	The parameter pack of the remaining members to be read.

**Returns**

Success if all the members exist and have valid types; Error otherwise.

**5.1.3.19 toJsonArray() [1/2]**

```
template<typename T >
Array rstudio::launcher_plugins::json::toJSONArray (
    const std::set< T > & in_set )
```

Converts a set value to a JSON array value.

**Template Parameters**

<i>T</i>	The C/C++ type of the set elements.
----------	-------------------------------------

**Parameters**

<i>in_set</i>	The set value to convert to a JSON array value.
---------------	---

**Returns**

The converted JSON array value.

**5.1.3.20 toJsonArray() [2/2]**

```
template<typename T >
Array rstudio::launcher_plugins::json::toJSONArray (
    const std::vector< T > & in_vector )
```

Converts a vector value to a JSON array value.

## Template Parameters

<i>T</i>	The C/C++ type of the vector elements.
----------	--

## Parameters

<i>in_vector</i>	The vector value to convert to a JSON array value.
------------------	--

## Returns

The converted JSON array value.

**5.1.3.21 toJsonValue() [1/5]**

```
template<typename T >
Value rstudio::launcher_plugins::json::detail::toJsonValue (
    const Optional< T > & in_value ) [inline]
```

Internal utility function. Converts a C/C++ optional value to a JSON value.

## Template Parameters

<i>T</i>	The C/C++ type of the value to convert.
----------	---

## Parameters

<i>in_value</i>	The optional value to convert to a JSON value.
-----------------	--

## Returns

The converted JSON value.

**5.1.3.22 toJsonValue() [2/5]**

```
template<typename T >
Value rstudio::launcher_plugins::json::detail::toJsonValue (
    const std::set< T > & in_set ) [inline]
```

Internal utility function. Converts a set value to a JSON array value.

## Template Parameters

<i>T</i>	The C/C++ type of the set elements.
----------	-------------------------------------

**Parameters**

<i>in_set</i>	The set value to convert to a JSON array value.
---------------	---

**Returns**

The converted JSON array value.

**5.1.3.23 toJsonValue() [3/5]**

```
template<typename T >
Value rstudio::launcher_plugins::json::detail::toJsonValue (
    const std::vector< T > & in_vector ) [inline]
```

Internal utility function. Converts a vector value to a JSON array value.

**Template Parameters**

<i>T</i>	The C/C++ type of the vector elements.
----------	--

**Parameters**

<i>in_vector</i>	The vector value to convert to a JSON array value.
------------------	--

**Returns**

The converted JSON array value.

**5.1.3.24 toJsonValue() [4/5]**

```
template<typename T >
Value rstudio::launcher_plugins::json::detail::toJsonValue (
    const T & in_value ) [inline]
```

Internal utility function. Converts a C/C++ value to a JSON value.

**Template Parameters**

<i>T</i>	The C/C++ type of the value to convert.
----------	---

**Parameters**

<i>in_value</i>	The value to convert to a JSON value.
-----------------	---------------------------------------

**Returns**

The converted JSON value.

**5.1.3.25 toJsonValue() [5/5]**

```
template<typename T >
Value rstudio::launcher_plugins::json::toJsonValue (
    const T & in_value ) [inline]
```

Converts a C/C++ value to a JSON value.

**Template Parameters**

<i>T</i>	The C/C++ type of the value to convert.
----------	---

**Parameters**

<i>in_value</i>	The value to convert to a JSON value.
-----------------	---------------------------------------

**Returns**

The converted JSON value.

**5.1.3.26 typeAsString()**

```
std::string rstudio::launcher_plugins::json::typeAsString (
    const Value & in_value ) [inline]
```

Gets the type of the JSON value as a string.

**Parameters**

<i>in_value</i>	The JSON value for which retrieve the type as a string.
-----------------	---

**Returns**

The type of the JSON value, as a string.

## 5.2 /workspaces/rstudio-launcher-plugin-sdk/sdk/include/logging/↵ Logger.hpp File Reference

```
#include <map>
```

```
#include <memory>
#include <ostream>
#include <string>
#include <PImpl.hpp>
```

## Enumerations

- enum `rstudio::launcher_plugins::logging::LogLevel` {  
**OFF** = 0, **ERR** = 1, **WARN** = 2, **INFO** = 3,  
**DEBUG** = 4 }  
*Enum which represents the level of detail at which to log messages.*

## Functions

- std::string `rstudio::launcher_plugins::logging::cleanDelimiters` (const std::string &in\_str)  
*Helper function which cleans the log delimiter character from a string.*
- void `rstudio::launcher_plugins::logging::setProgramId` (const std::string &in\_programId)  
*Sets the program ID for the logger.*
- void `rstudio::launcher_plugins::logging::addLogDestination` (const std::shared\_ptr< ILogDestination > &in\_↵  
destination)  
*Adds an un-sectioned log destination to the logger.*
- void `rstudio::launcher_plugins::logging::addLogDestination` (const std::shared\_ptr< ILogDestination > &in\_↵  
destination, const std::string &in\_section)  
*Adds a sectioned log destination to the logger.*
- std::string `rstudio::launcher_plugins::logging::cleanDelims` (const std::string &in\_toClean)  
*Replaces logging delimiters with ' ' in the specified string.*
- void `rstudio::launcher_plugins::logging::logError` (const Error &in\_error)  
*Logs an error to all registered destinations.*
- void `rstudio::launcher_plugins::logging::logError` (const Error &in\_error, const ErrorLocation &in\_location)  
*Logs an error to all registered destinations.*
- void `rstudio::launcher_plugins::logging::logErrorAsWarning` (const Error &in\_error)  
*Logs an error as a warning to all registered destinations.*
- void `rstudio::launcher_plugins::logging::logErrorAsInfo` (const Error &in\_error)  
*Logs an error as an info message to all registered destinations.*
- void `rstudio::launcher_plugins::logging::logErrorAsDebug` (const Error &in\_error)  
*Logs an error as a debug message to all registered destinations.*
- void `rstudio::launcher_plugins::logging::logErrorMessage` (const std::string &in\_message, const std::string  
&in\_section=std::string())  
*Logs an error to all registered destinations.*
- void `rstudio::launcher_plugins::logging::logErrorMessage` (const std::string &in\_message, const Error↵  
Location &in\_loggedFrom)  
*Logs an error to all registered destinations.*
- void `rstudio::launcher_plugins::logging::logErrorMessage` (const std::string &in\_message, const std::string  
&in\_section, const ErrorLocation &in\_loggedFrom)  
*Logs an error to all registered destinations.*
- void `rstudio::launcher_plugins::logging::logWarningMessage` (const std::string &in\_message, const std::string  
&in\_section=std::string())  
*Logs a warning message to all registered destinations.*
- void `rstudio::launcher_plugins::logging::logWarningMessage` (const std::string &in\_message, const Error↵  
Location &in\_loggedFrom)



- Logs a warning message to all registered destinations.*
- void `rstudio::launcher_plugins::logging::logWarningMessage` (const std::string &in\_message, const std::string &in\_section, const ErrorLocation &in\_loggedFrom)
- Logs a warning message to all registered destinations.*
- void `rstudio::launcher_plugins::logging::logDebugMessage` (const std::string &in\_message, const std::string &in\_section=std::string())
- Logs a debug message to all registered destinations.*
- void `rstudio::launcher_plugins::logging::logDebugMessage` (const std::string &in\_message, const ErrorLocation &in\_loggedFrom)
- Logs a debug message to all registered destinations.*
- void `rstudio::launcher_plugins::logging::logDebugMessage` (const std::string &in\_message, const std::string &in\_section, const ErrorLocation &in\_loggedFrom)
- Logs a debug message to all registered destinations.*
- void `rstudio::launcher_plugins::logging::logInfoMessage` (const std::string &in\_message, const std::string &in\_section=std::string())
- Logs an info message to all registered destinations.*
- void `rstudio::launcher_plugins::logging::logInfoMessage` (const std::string &in\_message, const ErrorLocation &in\_loggedFrom)
- Logs an info message to all registered destinations.*
- void `rstudio::launcher_plugins::logging::logInfoMessage` (const std::string &in\_message, const std::string &in\_section, const ErrorLocation &in\_loggedFrom)
- Logs an info message to all registered destinations.*
- void `rstudio::launcher_plugins::logging::reloadAllLogDestinations` ()
- Reloads all log destinations. May be used after fork to prevent stale file handles.*
- void `rstudio::launcher_plugins::logging::removeLogDestination` (unsigned int in\_destinationId, const std::string &in\_section=std::string())
- Removes a log destination from the logger.*
- std::ostream & `rstudio::launcher_plugins::logging::writeError` (const Error &in\_error, std::ostream &io\_os)
- Writes an error to the specified output stream.*
- std::string `rstudio::launcher_plugins::logging::writeError` (const Error &in\_error)
- Writes an error to string.*

## Variables

- constexpr char `rstudio::launcher_plugins::logging::s_delim` = ';'
  - Log delimiting character which may be used for custom log formatting.*

## 5.2.1 Detailed Description

Logging definitions and functions.

## 5.2.2 Function Documentation

### 5.2.2.1 addLogDestination() [1/2]

```
void rstudio::launcher_plugins::logging::addLogDestination (
    const std::shared_ptr< ILogDestination > & in_destination )
```

Adds an un-sectioned log destination to the logger.

If a duplicate destination is added, the duplicate will be ignored.

**Parameters**

<i>in_destination</i>	The destination to add.
-----------------------	-------------------------

**5.2.2.2 addLogDestination() [2/2]**

```
void rstudio::launcher_plugins::logging::addLogDestination (
    const std::shared_ptr< ILogDestination > & in_destination,
    const std::string & in_section )
```

Adds a sectioned log destination to the logger.

If a duplicate destination is added, the duplicate will be ignored. The log destination will be registered to only record log messages with that section. A log destination may be added to multiple sections and as an un-sectioned log.

**Parameters**

<i>in_destination</i>	The destination to add.
<i>in_section</i>	The name of the log section to which this logger is assigned.

**5.2.2.3 cleanDelimiters()**

```
std::string rstudio::launcher_plugins::logging::cleanDelimiters (
    const std::string & in_str )
```

Helper function which cleans the log delimiter character from a string.

**Parameters**

<i>in_str</i>	The string to be cleaned
---------------	--------------------------

**Returns**

The cleaned string.

**5.2.2.4 cleanDelims()**

```
std::string rstudio::launcher_plugins::logging::cleanDelims (
    const std::string & in_toClean )
```

Replaces logging delimiters with ' ' in the specified string.

### Parameters

<i>in_toClean</i>	The string from which to clean logging delimiters.
-------------------	--

### Returns

The cleaned string.

#### 5.2.2.5 logDebugMessage() [1/3]

```
void rstudio::launcher_plugins::logging::logDebugMessage (
    const std::string & in_message,
    const ErrorLocation & in_loggedFrom )
```

Logs a debug message to all registered destinations.

If no destinations are registered, no log will be written. If the configured log level is below LogLevel::DEBUG, no log will be written.

### Parameters

<i>in_message</i>	The message to log as a debug message.
<i>in_location</i>	The location from which the error message was logged.

#### 5.2.2.6 logDebugMessage() [2/3]

```
void rstudio::launcher_plugins::logging::logDebugMessage (
    const std::string & in_message,
    const std::string & in_section,
    const ErrorLocation & in_loggedFrom )
```

Logs a debug message to all registered destinations.

If no destinations are registered, no log will be written. If the configured log level is below LogLevel::DEBUG, no log will be written.

### Parameters

<i>in_message</i>	The message to log as a debug message.
<i>in_section</i>	The section of the log that the message belongs in.
<i>in_location</i>	The location from which the error message was logged.

### 5.2.2.7 logDebugMessage() [3/3]

```
void rstudio::launcher_plugins::logging::logDebugMessage (
    const std::string & in_message,
    const std::string & in_section = std::string() )
```

Logs a debug message to all registered destinations.

If no destinations are registered, no log will be written. If the configured log level is below LogLevel::DEBUG, no log will be written.

#### Parameters

<i>in_message</i>	The message to log as a debug message.
<i>in_section</i>	The section of the log that the message belongs in. Default: no section.

### 5.2.2.8 logError() [1/2]

```
void rstudio::launcher_plugins::logging::logError (
    const Error & in_error )
```

Logs an error to all registered destinations.

If no destinations are registered, no log will be written. If the configured log level is below LogLevel::ERR, no log will be written.

#### Parameters

<i>in_error</i>	The error to log.
-----------------	-------------------

### 5.2.2.9 logError() [2/2]

```
void rstudio::launcher_plugins::logging::logError (
    const Error & in_error,
    const ErrorLocation & in_location )
```

Logs an error to all registered destinations.

If no destinations are registered, no log will be written. If the configured log level is below LogLevel::ERR, no log will be written.

#### Parameters

<i>in_error</i>	The error to log.
<i>in_errorLocation</i>	A location higher in the stack than the ErrorLocation in in_error. Provides more context.

#### 5.2.2.10 logErrorAsDebug()

```
void rstudio::launcher_plugins::logging::logErrorAsDebug (
    const Error & in_error )
```

Logs an error as a debug message to all registered destinations.

If no destinations are registered, no log will be written. If the configured log level is below `LogLevel::DEBUG`, no log will be written.

##### Parameters

<i>in_error</i>	The error to log as a debug message.
-----------------	--------------------------------------

#### 5.2.2.11 logErrorAsInfo()

```
void rstudio::launcher_plugins::logging::logErrorAsInfo (
    const Error & in_error )
```

Logs an error as an info message to all registered destinations.

If no destinations are registered, no log will be written. If the configured log level is below `LogLevel::INFO`, no log will be written.

##### Parameters

<i>in_error</i>	The error to log as an info message.
-----------------	--------------------------------------

#### 5.2.2.12 logErrorAsWarning()

```
void rstudio::launcher_plugins::logging::logErrorAsWarning (
    const Error & in_error )
```

Logs an error as a warning to all registered destinations.

If no destinations are registered, no log will be written. If the configured log level is below `LogLevel::WARN`, no log will be written.

##### Parameters

<i>in_error</i>	The error to log as a warning.
-----------------	--------------------------------

**5.2.2.13 logErrorMessage() [1/3]**

```
void rstudio::launcher_plugins::logging::logErrorMessage (
    const std::string & in_message,
    const ErrorLocation & in_loggedFrom )
```

Logs an error to all registered destinations.

If no destinations are registered, no log will be written. If the configured log level is below LogLevel::ERR, no log will be written.

**Parameters**

<i>in_message</i>	The message to log as an error.
<i>in_location</i>	The location from which the error message was logged.

**5.2.2.14 logErrorMessage() [2/3]**

```
void rstudio::launcher_plugins::logging::logErrorMessage (
    const std::string & in_message,
    const std::string & in_section,
    const ErrorLocation & in_loggedFrom )
```

Logs an error to all registered destinations.

If no destinations are registered, no log will be written. If the configured log level is below LogLevel::ERR, no log will be written.

**Parameters**

<i>in_message</i>	The message to log as an error.
<i>in_section</i>	The section of the log that the message belongs in.
<i>in_location</i>	The location from which the error message was logged.

**5.2.2.15 logErrorMessage() [3/3]**

```
void rstudio::launcher_plugins::logging::logErrorMessage (
    const std::string & in_message,
    const std::string & in_section = std::string() )
```

Logs an error to all registered destinations.

If no destinations are registered, no log will be written. If the configured log level is below LogLevel::ERR, no log will be written.

## Parameters

<i>in_message</i>	The message to log as an error.
<i>in_section</i>	The section of the log that the message belongs in. Default: no section.

**5.2.2.16 logInfoMessage()** [1/3]

```
void rstudio::launcher_plugins::logging::logInfoMessage (
    const std::string & in_message,
    const ErrorLocation & in_loggedFrom )
```

Logs an info message to all registered destinations.

If no destinations are registered, no log will be written. If the configured log level is below `LogLevel::INFO`, no log will be written.

## Parameters

<i>in_message</i>	The message to log as an info message.
<i>in_location</i>	The location from which the error message was logged.

**5.2.2.17 logInfoMessage()** [2/3]

```
void rstudio::launcher_plugins::logging::logInfoMessage (
    const std::string & in_message,
    const std::string & in_section,
    const ErrorLocation & in_loggedFrom )
```

Logs an info message to all registered destinations.

If no destinations are registered, no log will be written. If the configured log level is below `LogLevel::INFO`, no log will be written.

## Parameters

<i>in_message</i>	The message to log as an info message.
<i>in_section</i>	The section of the log that the message belongs in.
<i>in_location</i>	The location from which the error message was logged.

**5.2.2.18 logInfoMessage()** [3/3]

```
void rstudio::launcher_plugins::logging::logInfoMessage (
```

```
const std::string & in_message,
const std::string & in_section = std::string() )
```

Logs an info message to all registered destinations.

If no destinations are registered, no log will be written. If the configured log level is below `LogLevel::INFO`, no log will be written.

#### Parameters

<i>in_message</i>	The message to log as an info message.
<i>in_section</i>	The section of the log that the message belongs in. Default: no section.

### 5.2.2.19 logWarningMessage() [1/3]

```
void rstudio::launcher_plugins::logging::logWarningMessage (
    const std::string & in_message,
    const ErrorLocation & in_loggedFrom )
```

Logs a warning message to all registered destinations.

If no destinations are registered, no log will be written. If the configured log level is below `LogLevel::WARN`, no log will be written.

#### Parameters

<i>in_message</i>	The message to log as a warning.
<i>in_location</i>	The location from which the error message was logged.

### 5.2.2.20 logWarningMessage() [2/3]

```
void rstudio::launcher_plugins::logging::logWarningMessage (
    const std::string & in_message,
    const std::string & in_section,
    const ErrorLocation & in_loggedFrom )
```

Logs a warning message to all registered destinations.

If no destinations are registered, no log will be written. If the configured log level is below `LogLevel::WARN`, no log will be written.

#### Parameters

<i>in_message</i>	The message to log as a warning.
<i>in_section</i>	The section of the log that the message belongs in.
<i>in_location</i>	The location from which the error message was logged.



#### 5.2.2.21 logWarningMessage() [3/3]

```
void rstudio::launcher_plugins::logging::logWarningMessage (
    const std::string & in_message,
    const std::string & in_section = std::string() )
```

Logs a warning message to all registered destinations.

If no destinations are registered, no log will be written. If the configured log level is below LogLevel::WARN, no log will be written.

##### Parameters

<i>in_message</i>	The message to log as a warning.
<i>in_section</i>	The section of the log that the message belongs in. Default: no section.

#### 5.2.2.22 removeLogDestination()

```
void rstudio::launcher_plugins::logging::removeLogDestination (
    unsigned int in_destinationId,
    const std::string & in_section = std::string() )
```

Removes a log destination from the logger.

If a log destination does not exist with the given ID, no destination will be removed. The log will be removed from the set of default log destinations as well as any sections it has been registered to, if *in\_section* is empty. If *in\_section* is not empty, the log will be removed only from that section.

##### Parameters

<i>in_↔ destinationId</i>	The ID of the destination to remove.
<i>in_section</i>	The name of the section from which to remove the log. Default: all sections.

#### 5.2.2.23 setProgramId()

```
void rstudio::launcher_plugins::logging::setProgramId (
    const std::string & in_programId )
```

Sets the program ID for the logger.

## Parameters

<i>in_↵ programId</i>	The ID of the program.
---------------------------	------------------------

**5.2.2.24 writeError()** [1/2]

```
std::string rstudio::launcher_plugins::logging::writeError (
    const Error & in_error )
```

Writes an error to string.

## Parameters

<i>in_error</i>	The error to write.
-----------------	---------------------

## Returns

The error as a string.

**5.2.2.25 writeError()** [2/2]

```
std::ostream& rstudio::launcher_plugins::logging::writeError (
    const Error & in_error,
    std::ostream & io_os )
```

Writes an error to the specified output stream.

## Parameters

<i>in_error</i>	The error to write.
<i>io_os</i>	The output stream to which to write the error.

## Returns

A reference to the specified output stream.

## 5.3 /workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/↵ Crypto.hpp File Reference

```
#include <string>
#include <vector>
```

## Functions

- Error `rstudio::launcher_plugins::system::crypto::aesDecrypt` (const std::vector< unsigned char > &in\_data, const std::vector< unsigned char > &in\_key, const std::vector< unsigned char > &in\_iv, std::vector< unsigned char > &out\_decrypted)  
*AES decrypts the specified data using the specified initialization vector.*
- Error `rstudio::launcher_plugins::system::crypto::aesEncrypt` (const std::vector< unsigned char > &in\_data, const std::vector< unsigned char > &in\_key, const std::vector< unsigned char > &in\_iv, std::vector< unsigned char > &out\_encrypted)  
*AES encrypts the specified data using the specified initialization vector.*
- Error `rstudio::launcher_plugins::system::crypto::base64Decode` (const std::string &in\_data, std::vector< unsigned char > &out\_decoded)  
*Base-64 decodes a string.*
- Error `rstudio::launcher_plugins::system::crypto::base64Decode` (const std::string &in\_data, std::string &out\_decoded)  
*Base-64 decodes a string.*
- Error `rstudio::launcher_plugins::system::crypto::base64Encode` (const std::vector< unsigned char > &in\_data, std::string &out\_encoded)  
*Base-64 encodes a string.*
- Error `rstudio::launcher_plugins::system::crypto::base64Encode` (const unsigned char \*in\_data, int in\_length, std::string &out\_encoded)  
*Base-64 encodes a string.*
- Error `rstudio::launcher_plugins::system::crypto::decryptAndBase64Decode` (const std::string &in\_input, const std::string &in\_key, const std::string &in\_ivStr, std::string &out\_decrypted)  
*Base-64 decodes and then decrypts an AES encrypted string with the specified initialization vector, which is also base-64 encoded.*
- Error `rstudio::launcher_plugins::system::crypto::encryptAndBase64Encode` (const std::string &in\_input, const std::string &in\_key, std::string &out\_iv, std::string &out\_encrypted)  
*AES encrypts and then base-64 encodes the specified string using the given key. Also generates and base-64 encodes an initialization vector which is used in the encryption of the input.*
- Error `rstudio::launcher_plugins::system::crypto::random` (uint32\_t in\_length, std::vector< unsigned char > &out\_randomData)  
*Generates random bytes of the specified length.*

### 5.3.1 Detailed Description

Cryptographic Utilities.

### 5.3.2 Function Documentation

#### 5.3.2.1 aesDecrypt()

```
Error rstudio::launcher_plugins::system::crypto::aesDecrypt (
    const std::vector< unsigned char > & in_data,
    const std::vector< unsigned char > & in_key,
    const std::vector< unsigned char > & in_iv,
    std::vector< unsigned char > & out_decrypted )
```

AES decrypts the specified data using the specified initialization vector.

This function is the inverse of aesEncrypt.

**Parameters**

<i>in_data</i>	The data to be decrypted.
<i>in_key</i>	The key with which to decrypt the data.
<i>in_iv</i>	The initialization vector that was used during encryption.
<i>out_decrypted</i>	The decrypted data.

**Returns**

Success if the data could be AES decrypted; Error otherwise.

**5.3.2.2 aesEncrypt()**

```
Error rstudio::launcher_plugins::system::crypto::aesEncrypt (
    const std::vector< unsigned char > & in_data,
    const std::vector< unsigned char > & in_key,
    const std::vector< unsigned char > & in_iv,
    std::vector< unsigned char > & out_encrypted )
```

AES encrypts the specified data using the specified initialization vector.

This function is the inverse of aesDecrypt.

**Parameters**

<i>in_data</i>	The data to be encrypted.
<i>in_key</i>	The key with which to encrypt the data.
<i>in_iv</i>	The initialization vector to use during encryption.
<i>out_encrypted</i>	The encrypted data.

**Returns**

Success if the data could be AES encrypted; Error otherwise.

**5.3.2.3 base64Decode() [1/2]**

```
Error rstudio::launcher_plugins::system::crypto::base64Decode (
    const std::string & in_data,
    std::string & out_decoded )
```

Base-64 decodes a string.

This function is the inverse of base64Encode.

**Parameters**

<i>in_data</i>	The base-64 encoded data to be decoded.
<i>out_decoded</i>	The decoded data.

**Returns**

Success if the data could be base-64 decoded; Error otherwise.

**5.3.2.4 base64Decode() [2/2]**

```
Error rstudio::launcher_plugins::system::crypto::base64Decode (  
    const std::string & in_data,  
    std::vector< unsigned char > & out_decoded )
```

Base-64 decodes a string.

This function is the inverse of base64Encode.

**Parameters**

<i>in_data</i>	The base-64 encoded data to be decoded.
<i>out_decoded</i>	The decoded data.

**Returns**

Success if the data could be base-64 decoded; Error otherwise.

**5.3.2.5 base64Encode() [1/2]**

```
Error rstudio::launcher_plugins::system::crypto::base64Encode (  
    const std::vector< unsigned char > & in_data,  
    std::string & out_encoded )
```

Base-64 encodes a string.

This function is the inverse of base64Decode.

**Parameters**

<i>in_data</i>	The string data to be encoded.
<i>out_encoded</i>	The base-64 encoded string.

### Returns

Success if the data could be base-64 encoded; Error otherwise.

#### 5.3.2.6 base64Encode() [2/2]

```
Error rstudio::launcher_plugins::system::crypto::base64Encode (
    const unsigned char * in_data,
    int in_length,
    std::string & out_encoded )
```

Base-64 encodes a string.

This function is the inverse of base64Decode.

### Parameters

<i>in_data</i>	The string data to be encoded.
<i>in_length</i>	The length of in_data.
<i>out_encoded</i>	The base-64 encoded string.

### Returns

Success if the data could be base-64 encoded; Error otherwise.

#### 5.3.2.7 decryptAndBase64Decode()

```
Error rstudio::launcher_plugins::system::crypto::decryptAndBase64Decode (
    const std::string & in_input,
    const std::string & in_key,
    const std::string & in_ivStr,
    std::string & out_decrypted )
```

Base-64 decodes and then decrypts an AES encrypted string with the specified initialization vector, which is also base-64 encoded.

This function is the inverse of encryptAndBase64Encode.

### Parameters

<i>in_input</i>	The base-64 encoded AES encrypted string.
<i>in_key</i>	The key with which to decrypt the string.
<i>in_ivStr</i>	The base-64 encrypted initialization vector.
<i>out_decrypted</i>	The base-64 decoded and decrypted string.

**Returns**

Success if `in_input` could be base-64 decoded and decrypted; Error otherwise.

**5.3.2.8 encryptAndBase64Encode()**

```
Error rstudio::launcher_plugins::system::crypto::encryptAndBase64Encode (
    const std::string & in_input,
    const std::string & in_key,
    std::string & out_iv,
    std::string & out_encrypted )
```

AES encrypts and then base-64 encodes the specified string using the given key. Also generates and base-64 encodes an initialization vector which is used in the encryption of the input.

This function is the inverse of `decryptAndBase64Decode`.

**Parameters**

<i>in_input</i>	The string to encrypt and base-64 encode.
<i>in_key</i>	The key with which to encrypt the string.
<i>out_iv</i>	The generated base-64 encoded initialization vector.
<i>out_encrypted</i>	The encrypted and base-64 encoded string.

**Returns**

Success if the string could be encrypted and base-64 encoded; Error otherwise.

**5.3.2.9 random()**

```
Error rstudio::launcher_plugins::system::crypto::random (
    uint32_t in_length,
    std::vector< unsigned char > & out_randomData )
```

Generates random bytes of the specified length.

This function uses openssl to generate random data. Summarized from the openssl documentation: The bytes are generated using a cryptographically secure pseudo random generator. The quality of the randomness is determined by the operating system's entropy source. If an entropy source fails or isn't available, an error will be returned.

**Parameters**

<i>in_length</i>	The number of bytes of random data to generate.
<i>out_randomData</i>	The random data.

## Returns

Success if the random data could be generated; Error otherwise.

## 5.4 /workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/Posix↵ System.hpp File Reference

```
#include <functional>
#include <Error.hpp>
```

### Classes

- struct [rstudio::launcher\\_plugins::system::posix::IpAddress](#)  
*Represents an IP address.*

### Functions

- Error [rstudio::launcher\\_plugins::system::posix::enableCoreDumps](#) ()  
*Enables core dumps for this process.*
- std::string [rstudio::launcher\\_plugins::system::posix::getEnvironmentVariable](#) (const std::string &in\_name)  
*Gets an environment variable from the system.*
- launcher\_plugins::Error [rstudio::launcher\\_plugins::system::posix::getIpAddresses](#) (std::vector< IpAddress > &out\_addresses, bool in\_includeIPv6=false)  
*Gets the IP addresses of the machine running this process.*
- Error [rstudio::launcher\\_plugins::system::posix::ignoreSignal](#) (int in\_signal)  
*Ignores a particular signal for this process.*
- template<typename T >  
T [rstudio::launcher\\_plugins::system::posix::posixCall](#) (const std::function< T()> &in\_posixFunction)  
*Makes a posix call and handles EINTR retries.*
- template<typename T >  
Error [rstudio::launcher\\_plugins::system::posix::posixCall](#) (const std::function< T()> &in\_posixFunction, const ErrorLocation &in\_errorLocation, T \*out\_result=nullptr)  
*Makes a posix call and handles EINTR retries.*
- bool [rstudio::launcher\\_plugins::system::posix::realUserIsRoot](#) ()  
*Checks whether the real user (not the effective user) running this process is root.*
- Error [rstudio::launcher\\_plugins::system::posix::restoreRoot](#) ()  
*Restores root privileges.*
- Error [rstudio::launcher\\_plugins::system::posix::restorePrivileges](#) ()  
*Restores privileges of the previous user, whose privileges were dropped by calling temporarilyDropPrivileges.*
- Error [rstudio::launcher\\_plugins::system::posix::temporarilyDropPrivileges](#) (const User &in\_user)  
*Temporarily drops privileges from root to the requested user.*

#### 5.4.1 Detailed Description

Posix System Utilities.



## 5.4.2 Function Documentation

### 5.4.2.1 enableCoreDumps()

```
Error rstudio::launcher_plugins::system::posix::enableCoreDumps ( )
```

Enables core dumps for this process.

#### Returns

Success if core dumps could be enabled; Error otherwise.

### 5.4.2.2 getEnvironmentVariable()

```
std::string rstudio::launcher_plugins::system::posix::getEnvironmentVariable (
    const std::string & in_name )
```

Gets an environment variable from the system.

#### Parameters

<i>in_name</i>	The name of the environment variable.
----------------	---------------------------------------

#### Returns

The value of the environment variable, if it exists; empty string otherwise.

### 5.4.2.3 getIpAddresses()

```
launcher_plugins::Error rstudio::launcher_plugins::system::posix::getIpAddresses (
    std::vector< IPAddress > & out_addresses,
    bool in_includeIPv6 = false )
```

Gets the IP addresses of the machine running this process.

#### Parameters

<i>out_addresses</i>	The IP addresses of the machine running this process.
<i>in_includeIPv6</i>	Whether or not to include IPv6 addresses. Default: false.

**Returns**

Success if the IP addresses could be retrieved; Error otherwise.

**5.4.2.4 ignoreSignal()**

```
Error rstudio::launcher_plugins::system::posix::ignoreSignal (
    int in_signal )
```

Ignores a particular signal for this process.

**Parameters**

<i>in_signal</i>	The signal to ignore.
------------------	-----------------------

**Returns**

Success if the specified signal could be ignored; Error otherwise.

**5.4.2.5 posixCall() [1/2]**

```
template<typename T >
T rstudio::launcher_plugins::system::posix::posixCall (
    const std::function< T()> & in_posixFunction )
```

Makes a posix call and handles EINTR retries.

Only for use with functions that return -1 on error and set errno.

**Template Parameters**

<i>T</i>	The return type of the function to be called.
----------	---

**Parameters**

<i>in_posixFunction</i>	The function to call.
-------------------------	-----------------------

**Returns**

The return value of the provided function.

### 5.4.2.6 posixCall() [2/2]

```
template<typename T >
Error rstudio::launcher_plugins::system::posix::posixCall (
    const std::function< T()> & in_posixFunction,
    const ErrorLocation & in_errorLocation,
    T * out_result = nullptr )
```

Makes a posix call and handles EINTR retries.

Only for use with functions that return -1 on error and set errno.

#### Template Parameters

<i>T</i>	The return type of the function to be called.
----------	---

#### Parameters

<i>in_posixFunction</i>	The function to call.
<i>in_errorLocation</i>	The location at which this function was invoked.
<i>out_result</i>	Optional output parameter on which the result of in_posixFunction will be set.

#### Returns

Success if the posix function was invoked and ran successfully; Error otherwise.

### 5.4.2.7 realUserIsRoot()

```
bool rstudio::launcher_plugins::system::posix::realUserIsRoot ( )
```

Checks whether the real user (not the effective user) running this process is root.

#### Returns

True if the real user is root; false otherwise.

### 5.4.2.8 restorePrivileges()

```
Error rstudio::launcher_plugins::system::posix::restorePrivileges ( )
```

Restores privileges of the previous user, whose privileges were dropped by calling temporarilyDropPrivileges.

#### Returns

Success if privileges could be restored; Error otherwise.

#### 5.4.2.9 `restoreRoot()`

```
Error rstudio::launcher_plugins::system::posix::restoreRoot ( )
```

Restores root privileges.

##### Returns

Success if root privileges could be restored; Error otherwise.

#### 5.4.2.10 `temporarilyDropPrivileges()`

```
Error rstudio::launcher_plugins::system::posix::temporarilyDropPrivileges (
    const User & in_user )
```

Temporarily drops privileges from root to the requested user.

##### Parameters

<code>in_user</code>	The user to which to drop privileges.
----------------------	---------------------------------------

##### Returns

Success if privileges could be dropped to the requested user; Error otherwise.

## 5.5 `/workspaces/rstudio-launcher-plugin-sdk/sdk/include/utils/Functionals.hpp` File Reference

```
#include <functional>
#include <Error.hpp>
```

### Typedefs

- typedef std::function< void(const Error &)> `rstudio::launcher_plugins::OnError`

#### 5.5.1 Detailed Description

Defines types for common callback functions within the SDK

#### 5.5.2 Typedef Documentation

### 5.5.2.1 OnError

```
typedef std::function<void(const Error&)> rstudio::launcher_plugins::OnError
```

Callback function which will be invoked when an error occurs.



# Index

`/workspaces/rstudio-launcher-plugin-sdk/sdk/include/json/Json.hpp`, 351  
`/workspaces/rstudio-launcher-plugin-sdk/sdk/include/logging/Logger.hpp`, 367  
`/workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/Crypto.hpp`, 378  
`/workspaces/rstudio-launcher-plugin-sdk/sdk/include/system/PosixSystem.hpp`, 384  
`/workspaces/rstudio-launcher-plugin-sdk/sdk/include/utls/Address.hpp`, 388  
`AbstractChildProcess`  
    `rstudio::launcher_plugins::system::process::AbstractChildProcess`, 14  
`AbstractJobRepository`  
    `rstudio::launcher_plugins::jobs::AbstractJobRepository`, 16  
`AbstractJobStatusWatcher`  
    `rstudio::launcher_plugins::jobs::AbstractJobStatusWatcher`, 19  
`AbstractLauncherCommunicator`  
    `rstudio::launcher_plugins::comms::AbstractLauncherCommunicator`, 21  
`AbstractMultiStream`  
    `rstudio::launcher_plugins::api::AbstractMultiStream<R, Args>`, 26  
`AbstractOutputStream`  
    `rstudio::launcher_plugins::api::AbstractOutputStream`, 32  
`AbstractPluginApi`  
    `rstudio::launcher_plugins::api::AbstractPluginApi`, 35  
`AbstractResourceStream`  
    `rstudio::launcher_plugins::api::AbstractResourceStream`, 37  
`AbstractTimedJobStatusWatcher`  
    `rstudio::launcher_plugins::jobs::AbstractTimedJobStatusWatcher`, 40  
`AbstractTimedResourceStream`  
    `rstudio::launcher_plugins::api::AbstractTimedResourceStream`, 41  
`AbstractUserProfiles`  
    `rstudio::launcher_plugins::options::AbstractUserProfiles`, 43  
`addJob`  
    `rstudio::launcher_plugins::jobs::AbstractJobRepository`, 16  
`addLogDestination`  
    `Logger.hpp`, 369, 370  
`addOrUpdateProperty`  
    `rstudio::launcher_plugins::Error`, 94  
    `addProperty`  
    `rstudio::launcher_plugins::Error`, 95  
    `addRequest`  
    `rstudio::launcher_plugins::api::AbstractMultiStream<R, Args>`, 26  
    `rstudio::launcher_plugins::api::AbstractResourceStream`, 37  
    `rstudio::launcher_plugins::system::posix::IpAddress`, 157  
    `aesDecrypt`  
    `rstudio::launcher_plugins::system::process::AbstractChildProcess`, 379  
    `aesEncrypt`  
    `Crypto.hpp`, 380  
    `AllowUnknownImages`  
    `rstudio::launcher_plugins::api::ContainerConfiguration`, 78  
    `Arguments`  
    `rstudio::launcher_plugins::api::Job`, 171  
    `rstudio::launcher_plugins::system::process::ProcessInfo`, 203  
    `Array`  
    `rstudio::launcher_plugins::json::Array`, 48  
    `asAzureFileMountSource`  
    `rstudio::launcher_plugins::api::MountSource`, 214  
    `asCephFsMountSource`  
    `rstudio::launcher_plugins::api::MountSource`, 214, 215  
    `asGlusterFsMountSource`  
    `rstudio::launcher_plugins::api::MountSource`, 215  
    `asHostMountSource`  
    `rstudio::launcher_plugins::api::MountSource`, 216  
    `asioStream`  
    `rstudio::launcher_plugins::system::AsioStream`, 62  
    `asJsonType`  
    `rstudio::launcher_plugins::api::MountSource`, 354, 355  
    `asNfsMountSource`  
    `rstudio::launcher_plugins::api::MountSource`, 216  
    `asString`  
    `rstudio::launcher_plugins::Error`, 96  
    `rstudio::launcher_plugins::ErrorLocation`, 102  
    `AsyncDeadlineEvent`  
    `rstudio::launcher_plugins::system::AsyncDeadlineEvent`, 63, 64  
    `AZURE_FILE`  
    `rstudio::launcher_plugins::api::MountSource`, 213  
    `base64Decode`

- Crypto.hpp, [380](#), [381](#)
- base64Encode
  - Crypto.hpp, [381](#), [382](#)
- begin
  - rstudio::launcher\_plugins::json::Array, [49](#)
  - rstudio::launcher\_plugins::json::Object, [231](#)
- BOOTSTRAP
  - rstudio::launcher\_plugins::api::Request, [284](#)
  - rstudio::launcher\_plugins::api::Response, [293](#)
- BootstrapResponse
  - rstudio::launcher\_plugins::api::BootstrapResponse, [70](#)
- CANCEL
  - rstudio::launcher\_plugins::api::ControlJobRequest, [80](#)
- CANCELED
  - rstudio::launcher\_plugins::api::Job, [167](#)
- cancelJob
  - rstudio::launcher\_plugins::api::IJobSource, [146](#)
  - rstudio::launcher\_plugins::local::LocalJobSource, [198](#)
  - rstudio::launcher\_plugins::quickstart::QuickStartJobSource, [271](#)
- CEPH\_FS
  - rstudio::launcher\_plugins::api::MountSource, [213](#)
- changeFileMode
  - rstudio::launcher\_plugins::system::FilePath, [119](#)
- changeOwnership
  - rstudio::launcher\_plugins::system::FilePath, [120](#)
- cleanDelimiters
  - Logger.hpp, [370](#)
- cleanDelims
  - Logger.hpp, [370](#)
- clone
  - rstudio::launcher\_plugins::json::Value, [329](#)
- CloseStdIn
  - rstudio::launcher\_plugins::system::process::ProcessOptions, [265](#)
- Cluster
  - rstudio::launcher\_plugins::api::Job, [171](#)
- CLUSTER\_INFO
  - rstudio::launcher\_plugins::api::Response, [293](#)
- ClusterInfoResponse
  - rstudio::launcher\_plugins::api::ClusterInfoResponse, [75](#)
- coerce
  - rstudio::launcher\_plugins::json::Value, [329](#)
- Command
  - rstudio::launcher\_plugins::api::Job, [172](#)
- completeChildPath
  - rstudio::launcher\_plugins::system::FilePath, [120](#), [121](#)
- completePath
  - rstudio::launcher\_plugins::system::FilePath, [121](#)
- Config
  - rstudio::launcher\_plugins::api::Job, [172](#)
- ContainerConfig
  - rstudio::launcher\_plugins::api::JobSourceConfiguration, [182](#)
- ContainerDetails
  - rstudio::launcher\_plugins::api::Job, [172](#)
- ContainerImages
  - rstudio::launcher\_plugins::api::ContainerConfiguration, [78](#)
- CONTROL\_JOB
  - rstudio::launcher\_plugins::api::Request, [284](#)
  - rstudio::launcher\_plugins::api::Response, [293](#)
- ControlJobResponse
  - rstudio::launcher\_plugins::api::ControlJobResponse, [81](#)
- copy
  - rstudio::launcher\_plugins::system::FilePath, [121](#)
- copyDirectoryRecursive
  - rstudio::launcher\_plugins::system::FilePath, [122](#)
- CPU\_COUNT
  - rstudio::launcher\_plugins::api::ResourceLimit::Type, [310](#)
- CPU\_TIME
  - rstudio::launcher\_plugins::api::ResourceLimit::Type, [310](#)
- CpuPercent
  - rstudio::launcher\_plugins::api::ResourceUtilData, [288](#)
- CpuSeconds
  - rstudio::launcher\_plugins::api::ResourceUtilData, [288](#)
- createAliasedPath
  - rstudio::launcher\_plugins::system::FilePath, [122](#)
- createDirectory
  - rstudio::launcher\_plugins::system::FilePath, [122](#)
- createMember
  - rstudio::launcher\_plugins::json::Object, [231](#)
- createOutputStream
  - rstudio::launcher\_plugins::api::IJobSource, [147](#)
  - rstudio::launcher\_plugins::local::LocalJobSource, [198](#)
  - rstudio::launcher\_plugins::quickstart::QuickStartJobSource, [272](#)
- createResourceStream
  - rstudio::launcher\_plugins::api::IJobSource, [147](#)
  - rstudio::launcher\_plugins::local::LocalJobSource, [199](#)
  - rstudio::launcher\_plugins::quickstart::QuickStartJobSource, [272](#)
- Crypto.hpp
  - aesDecrypt, [379](#)
  - aesEncrypt, [380](#)
  - base64Decode, [380](#), [381](#)
  - base64Encode, [381](#), [382](#)
  - decryptAndBase64Decode, [382](#)
  - encryptAndBase64Encode, [383](#)
  - random, [383](#)
- CustomConfig
  - rstudio::launcher\_plugins::api::JobSourceConfiguration, [182](#)



- CustomType
  - rstudio::launcher\_plugins::api::MountSource, 219
- DateTime
  - rstudio::launcher\_plugins::system::DateTime, 83, 84
- decryptAndBase64Decode
  - Crypto.hpp, 382
- DefaultImage
  - rstudio::launcher\_plugins::api::ContainerConfiguration, 78
- DefaultValue
  - rstudio::launcher\_plugins::api::ResourceLimit, 287
- Destination
  - rstudio::launcher\_plugins::api::Mount, 211
- doRotation
  - rstudio::launcher\_plugins::logging::FileLogOptions, 112
- enableCoreDumps
  - PosixSystem.hpp, 385
- enableDebugLogging
  - rstudio::launcher\_plugins::options::Options, 250
- encryptAndBase64Encode
  - Crypto.hpp, 383
- end
  - rstudio::launcher\_plugins::json::Array, 49
  - rstudio::launcher\_plugins::json::Object, 232
- ensureDirectory
  - rstudio::launcher\_plugins::system::FilePath, 123
- ensureFile
  - rstudio::launcher\_plugins::system::FilePath, 123
- ENUM
  - rstudio::launcher\_plugins::api::JobConfig, 176
- Environment
  - rstudio::launcher\_plugins::api::Job, 172
- erase
  - rstudio::launcher\_plugins::json::Array, 49, 50
  - rstudio::launcher\_plugins::json::Object, 232, 233
- ERROR
  - rstudio::launcher\_plugins::api::Response, 293
- Error
  - rstudio::launcher\_plugins::Error, 92, 93
- ErrorLocation
  - rstudio::launcher\_plugins::ErrorLocation, 101, 102
- ErrorResponse
  - rstudio::launcher\_plugins::api::ErrorResponse, 106
- Exe
  - rstudio::launcher\_plugins::api::Job, 172
- Executable
  - rstudio::launcher\_plugins::system::process::ProcessInfo, 263
- exists
  - rstudio::launcher\_plugins::system::FilePath, 123
  - rstudio::launcher\_plugins::system::User, 312
- ExitCode
  - rstudio::launcher\_plugins::api::Job, 172
  - rstudio::launcher\_plugins::system::process::ProcessInfo, 267
- ExposedPorts
  - rstudio::launcher\_plugins::api::Job, 173
- FAILED
  - rstudio::launcher\_plugins::api::Job, 167
- FileLogDestination
  - rstudio::launcher\_plugins::logging::FileLogDestination, 109
- FileLogOptions
  - rstudio::launcher\_plugins::logging::FileLogOptions, 111
- FileOutputStream
  - rstudio::launcher\_plugins::api::FileOutputStream, 114
- FilePath
  - rstudio::launcher\_plugins::system::FilePath, 119
- find
  - rstudio::launcher\_plugins::json::Object, 233
- FINISHED
  - rstudio::launcher\_plugins::api::Job, 167
- FIRST
  - rstudio::launcher\_plugins::api::ControlJobRequest, 80
- FLOAT
  - rstudio::launcher\_plugins::api::JobConfig, 176
- fromJson
  - rstudio::launcher\_plugins::api::AzureFileMountSource, 67
  - rstudio::launcher\_plugins::api::CephFsMountSource, 72
  - rstudio::launcher\_plugins::api::Container, 76
  - rstudio::launcher\_plugins::api::ExposedPort, 107
  - rstudio::launcher\_plugins::api::GlusterFsMountSource, 141
  - rstudio::launcher\_plugins::api::HostMountSource, 144
  - rstudio::launcher\_plugins::api::Job, 168
  - rstudio::launcher\_plugins::api::JobConfig, 177
  - rstudio::launcher\_plugins::api::Mount, 210
  - rstudio::launcher\_plugins::api::MountSource, 217
  - rstudio::launcher\_plugins::api::NfsMountSource, 225
  - rstudio::launcher\_plugins::api::PlacementConstraint, 261
  - rstudio::launcher\_plugins::api::Request, 284
  - rstudio::launcher\_plugins::api::ResourceLimit, 287
- fromString
  - rstudio::launcher\_plugins::system::DateTime, 84
- Functionals.hpp
  - OnError, 388
- GET\_CLUSTER\_INFO
  - rstudio::launcher\_plugins::api::Request, 284
- GET\_JOB
  - rstudio::launcher\_plugins::api::Request, 284
- GET\_JOB\_NETWORK
  - rstudio::launcher\_plugins::api::Request, 284
- GET\_JOB\_OUTPUT
  - rstudio::launcher\_plugins::api::Request, 284

- GET\_JOB\_RESOURCE\_UTIL
  - rstudio::launcher\_plugins::api::Request, [284](#)
- GET\_JOB\_STATUS
  - rstudio::launcher\_plugins::api::Request, [284](#)
- getAbsolutePath
  - rstudio::launcher\_plugins::system::FilePath, [124](#)
- getAbsolutePathNative
  - rstudio::launcher\_plugins::system::FilePath, [124](#)
- getArray
  - rstudio::launcher\_plugins::json::Value, [329](#)
- getBack
  - rstudio::launcher\_plugins::json::Array, [50](#)
- getBool
  - rstudio::launcher\_plugins::json::Value, [329](#)
- getCanonicalPath
  - rstudio::launcher\_plugins::system::FilePath, [124](#)
- getCause
  - rstudio::launcher\_plugins::Error, [96](#)
- getChildren
  - rstudio::launcher\_plugins::system::FilePath, [124](#)
- getChildrenRecursive
  - rstudio::launcher\_plugins::system::FilePath, [125](#)
- getCode
  - rstudio::launcher\_plugins::Error, [96](#)
- getConfiguration
  - rstudio::launcher\_plugins::api::IJobSource, [148](#)
  - rstudio::launcher\_plugins::local::LocalJobSource, [199](#)
  - rstudio::launcher\_plugins::quickstart::QuickStartJobSource, [273](#)
- getCurrentUser
  - rstudio::launcher\_plugins::system::User, [313](#)
- getDirectory
  - rstudio::launcher\_plugins::logging::FileLogOptions, [112](#)
- getDouble
  - rstudio::launcher\_plugins::json::Value, [330](#)
- getEncodedJobId
  - rstudio::launcher\_plugins::api::JobIdRequest, [180](#)
- getEndpoints
  - rstudio::launcher\_plugins::api::GlusterFsMountSource, [142](#)
- getEndTime
  - rstudio::launcher\_plugins::api::JobStateRequest, [184](#)
- getEnvironmentVariable
  - PosixSystem.hpp, [385](#)
- getExtension
  - rstudio::launcher\_plugins::system::FilePath, [125](#)
- getExtensionLowerCase
  - rstudio::launcher\_plugins::system::FilePath, [125](#)
- getFieldSet
  - rstudio::launcher\_plugins::api::JobStateRequest, [184](#)
- getFile
  - rstudio::launcher\_plugins::ErrorLocation, [102](#)
- getFileMode
  - rstudio::launcher\_plugins::logging::FileLogOptions, [112](#)
  - rstudio::launcher\_plugins::system::FilePath, [125](#)
- getFilename
  - rstudio::launcher\_plugins::system::FilePath, [127](#)
- getFloat
  - rstudio::launcher\_plugins::json::Value, [330](#)
- getFront
  - rstudio::launcher\_plugins::json::Array, [50](#)
- getFunction
  - rstudio::launcher\_plugins::ErrorLocation, [102](#)
- getGroupId
  - rstudio::launcher\_plugins::system::User, [313](#)
- getHeartbeatIntervalSeconds
  - rstudio::launcher\_plugins::options::Options, [251](#)
- getHomePath
  - rstudio::launcher\_plugins::system::User, [313](#)
- getHost
  - rstudio::launcher\_plugins::api::NfsMountSource, [225](#)
- getHours
  - rstudio::launcher\_plugins::system::TimeDuration, [304](#)
- getId
  - rstudio::launcher\_plugins::api::Request, [285](#)
  - rstudio::launcher\_plugins::logging::FileLogDestination, [109](#)
  - rstudio::launcher\_plugins::logging::ILogDestination, [153](#)
- getInstance
  - rstudio::launcher\_plugins::local::LocalOptions, [204](#)
  - rstudio::launcher\_plugins::options::Options, [251](#)
  - rstudio::launcher\_plugins::quickstart::QuickStartOptions, [278](#)
- getInt
  - rstudio::launcher\_plugins::json::Value, [330](#)
- getInt64
  - rstudio::launcher\_plugins::json::Value, [330](#)
- getIpAddresses
  - PosixSystem.hpp, [385](#)
- getJob
  - rstudio::launcher\_plugins::api::SubmitJobRequest, [299](#)
  - rstudio::launcher\_plugins::jobs::AbstractJobRepository, [17](#)
- getJobConfigValue
  - rstudio::launcher\_plugins::api::Job, [169](#)
- getJobExpiryHours
  - rstudio::launcher\_plugins::options::Options, [251](#)
- getJobId
  - rstudio::launcher\_plugins::api::JobIdRequest, [180](#)
- getJobs
  - rstudio::launcher\_plugins::jobs::AbstractJobRepository, [17](#)
- getKey
  - rstudio::launcher\_plugins::local::LocalSecureCookie, [208](#)
- getLastWriteTime

- rstudio::launcher\_plugins::system::FilePath, 127
- getLauncherConfigFile
  - rstudio::launcher\_plugins::options::Options, 251
- getLexicallyNormalPath
  - rstudio::launcher\_plugins::system::FilePath, 127
- getLine
  - rstudio::launcher\_plugins::ErrorLocation, 103
- getLocation
  - rstudio::launcher\_plugins::Error, 96
- getLoggingDir
  - rstudio::launcher\_plugins::options::Options, 252
- getLogLevel
  - rstudio::launcher\_plugins::logging::ILogDestination, 154
  - rstudio::launcher\_plugins::options::Options, 252
- getMajorVersion
  - rstudio::launcher\_plugins::api::BootstrapRequest, 69
- getMaxMessageSize
  - rstudio::launcher\_plugins::options::Options, 252
- getMaxSizeMb
  - rstudio::launcher\_plugins::logging::FileLogOptions, 112
- getMessage
  - rstudio::launcher\_plugins::Error, 97
- getMicroseconds
  - rstudio::launcher\_plugins::system::TimeDuration, 304
- getMimeType
  - rstudio::launcher\_plugins::system::FilePath, 127
- getMinorVersion
  - rstudio::launcher\_plugins::api::BootstrapRequest, 69
- getMinutes
  - rstudio::launcher\_plugins::system::TimeDuration, 304
- getMonitors
  - rstudio::launcher\_plugins::api::CephFsMountSource, 73
- getName
  - rstudio::launcher\_plugins::Error, 97
  - rstudio::launcher\_plugins::json::Object::Member, 209
- getNetworkInfo
  - rstudio::launcher\_plugins::api::IJobSource, 148
  - rstudio::launcher\_plugins::local::LocalJobSource, 200
  - rstudio::launcher\_plugins::quickstart::QuickStartJobSource, 273
- getNodeConnectionTimeoutSeconds
  - rstudio::launcher\_plugins::local::LocalOptions, 204
- getObject
  - rstudio::launcher\_plugins::json::Value, 331
- getOperation
  - rstudio::launcher\_plugins::api::ControlJobRequest, 80
- getParent
  - rstudio::launcher\_plugins::system::FilePath, 128
- getPatchNumber
  - rstudio::launcher\_plugins::api::BootstrapRequest, 69
- getPath
  - rstudio::launcher\_plugins::api::CephFsMountSource, 73
  - rstudio::launcher\_plugins::api::GlusterFsMountSource, 142
  - rstudio::launcher\_plugins::api::HostMountSource, 144
  - rstudio::launcher\_plugins::api::NfsMountSource, 226
- getPid
  - rstudio::launcher\_plugins::system::process::AbstractChildProcess, 14
- getPluginName
  - rstudio::launcher\_plugins::options::Options, 252
- getProcessInfo
  - rstudio::launcher\_plugins::system::process::ProcessInfo, 262
- getProperties
  - rstudio::launcher\_plugins::Error, 97
- getProperty
  - rstudio::launcher\_plugins::Error, 97
- getRelativePath
  - rstudio::launcher\_plugins::system::FilePath, 128
- getRequestUsername
  - rstudio::launcher\_plugins::api::UserRequest, 319
- getRSandboxPath
  - rstudio::launcher\_plugins::options::Options, 253
- getSampleOption
  - rstudio::launcher\_plugins::quickstart::QuickStartOptions, 278
- getSchemaDefaults
  - rstudio::launcher\_plugins::json::Object, 234
- getScratchPath
  - rstudio::launcher\_plugins::options::Options, 253
- getSeconds
  - rstudio::launcher\_plugins::system::TimeDuration, 304
- getSecretFile
  - rstudio::launcher\_plugins::api::CephFsMountSource, 73
- getSecretName
  - rstudio::launcher\_plugins::api::AzureFileMountSource, 67
- getSecretRef
  - rstudio::launcher\_plugins::api::CephFsMountSource, 73
- getSecureCookieKeyFile
  - rstudio::launcher\_plugins::local::LocalOptions, 204
- getServerUser
  - rstudio::launcher\_plugins::options::Options, 253
- getShareName
  - rstudio::launcher\_plugins::api::AzureFileMountSource, 68
- getShell
  - rstudio::launcher\_plugins::system::User, 314

- getSize
  - rstudio::launcher\_plugins::json::Array, 50
  - rstudio::launcher\_plugins::json::Object, 234
  - rstudio::launcher\_plugins::system::FilePath, 128
- getSizeRecursive
  - rstudio::launcher\_plugins::system::FilePath, 128
- getStartTime
  - rstudio::launcher\_plugins::api::JobStateRequest, 184
- getStatusSet
  - rstudio::launcher\_plugins::api::JobStateRequest, 185
- getStem
  - rstudio::launcher\_plugins::system::FilePath, 129
- getStreamType
  - rstudio::launcher\_plugins::api::OutputStreamRequest, HEARTBEAT 256
- getString
  - rstudio::launcher\_plugins::json::Value, 331
- getSummary
  - rstudio::launcher\_plugins::Error, 98
- getTagSet
  - rstudio::launcher\_plugins::api::JobStateRequest, 185
- getThreadPoolSize
  - rstudio::launcher\_plugins::options::Options, 254
- getType
  - rstudio::launcher\_plugins::api::Request, 285
  - rstudio::launcher\_plugins::json::Value, 331
- getUInt
  - rstudio::launcher\_plugins::json::Value, 331
- getUInt64
  - rstudio::launcher\_plugins::json::Value, 332
- getUser
  - rstudio::launcher\_plugins::api::CephFsMountSource, 74
  - rstudio::launcher\_plugins::api::UserRequest, 319
- getUserFromIdentifier
  - rstudio::launcher\_plugins::system::User, 314
- getUserHomePath
  - rstudio::launcher\_plugins::system::User, 315
- getUserId
  - rstudio::launcher\_plugins::system::User, 315
- getUsername
  - rstudio::launcher\_plugins::system::User, 315
- getValue
  - rstudio::launcher\_plugins::json::Object::Member, 209
  - rstudio::launcher\_plugins::json::Value, 332
- getValueAt
  - rstudio::launcher\_plugins::json::Array, 51
- getValueForUser
  - rstudio::launcher\_plugins::options::AbstractUserProfiles, 43
- getValueOr
  - rstudio::launcher\_plugins::Optional< T >, 246
- GLUSTER\_FS
  - rstudio::launcher\_plugins::api::MountSource, 213
- hasExtension
  - rstudio::launcher\_plugins::system::FilePath, 129
- hasExtensionLowerCase
  - rstudio::launcher\_plugins::system::FilePath, 129
- hasLocation
  - rstudio::launcher\_plugins::ErrorLocation, 103
- hasMember
  - rstudio::launcher\_plugins::json::Object, 234, 235
- hasRunningChildren
  - rstudio::launcher\_plugins::system::process::ProcessSupervisor, 268
- hasTextMimeType
  - rstudio::launcher\_plugins::system::FilePath, 130
- hasValue
  - rstudio::launcher\_plugins::Optional< T >, 247
  - rstudio::launcher\_plugins::api::Request, 284
  - rstudio::launcher\_plugins::api::Response, 293
- HOST
  - rstudio::launcher\_plugins::api::MountSource, 213
- Host
  - rstudio::launcher\_plugins::api::Job, 173
- Hostname
  - rstudio::launcher\_plugins::api::NetworkInfo, 222
- Hours
  - rstudio::launcher\_plugins::system::TimeDuration, 305
- Id
  - rstudio::launcher\_plugins::api::Job, 173
- ignoreSignal
  - PosixSystem.hpp, 386
- IJobSource
  - rstudio::launcher\_plugins::api::IJobSource, 146
- ILogDestination
  - rstudio::launcher\_plugins::logging::ILogDestination, 153
- Image
  - rstudio::launcher\_plugins::api::Container, 77
- includePid
  - rstudio::launcher\_plugins::logging::FileLogOptions, 113
- Infinity
  - rstudio::launcher\_plugins::system::TimeDuration, 305
- Init
  - rstudio::launcher\_plugins::options::Options::Init, 155
- initialize
  - rstudio::launcher\_plugins::api::AbstractMultiStream< R, Args >, 28
  - rstudio::launcher\_plugins::api::AbstractPluginApi, 35
  - rstudio::launcher\_plugins::api::AbstractResourceStream, 37
  - rstudio::launcher\_plugins::api::AbstractTimedResourceStream, 41
  - rstudio::launcher\_plugins::api::IJobSource, 149

- rstudio::launcher\_plugins::jobs::AbstractJobRepository, 17
  - rstudio::launcher\_plugins::local::LocalJobRunner, 196
  - rstudio::launcher\_plugins::local::LocalJobSource, 200
  - rstudio::launcher\_plugins::local::LocalOptions, 205
  - rstudio::launcher\_plugins::local::LocalSecureCookie, 208
  - rstudio::launcher\_plugins::options::AbstractUserProfiles, 44
  - rstudio::launcher\_plugins::quickstart::QuickStartJobSource, 274
  - rstudio::launcher\_plugins::quickstart::QuickStartOptions, 279
  - rstudio::launcher\_plugins::quickstart::QuickStartResourceStream, 281
- insert
  - rstudio::launcher\_plugins::json::Object, 235–239
- INT
  - rstudio::launcher\_plugins::api::JobConfig, 176
- INVALID
  - rstudio::launcher\_plugins::api::ControlJobRequest, 80
  - rstudio::launcher\_plugins::api::Request, 284
- IpAddresses
  - rstudio::launcher\_plugins::api::NetworkInfo, 222
- isAllUsers
  - rstudio::launcher\_plugins::system::User, 315
- isArray
  - rstudio::launcher\_plugins::json::Value, 332
- isAzureFileMountSource
  - rstudio::launcher\_plugins::api::MountSource, 217
- isBool
  - rstudio::launcher\_plugins::json::Value, 333
- isCancelRequest
  - rstudio::launcher\_plugins::api::JobStatusRequest, 191
  - rstudio::launcher\_plugins::api::OutputStreamRequest, 256
  - rstudio::launcher\_plugins::api::ResourceUtilStreamRequest, 290
- isCephFsMountSource
  - rstudio::launcher\_plugins::api::MountSource, 218
- isCompleted
  - rstudio::launcher\_plugins::api::Job, 169
- isDirectory
  - rstudio::launcher\_plugins::system::FilePath, 130
- isDouble
  - rstudio::launcher\_plugins::json::Value, 333
- isEmpty
  - rstudio::launcher\_plugins::api::AbstractMultiStream<R, Args>, 28
  - rstudio::launcher\_plugins::json::Array, 51
  - rstudio::launcher\_plugins::json::Object, 239
  - rstudio::launcher\_plugins::system::FilePath, 130
  - rstudio::launcher\_plugins::system::User, 316
- isEqualCaseInsensitive
  - rstudio::launcher\_plugins::system::FilePath, 130
  - isEquivalentTo
    - rstudio::launcher\_plugins::system::FilePath, 131
  - isExpected
    - rstudio::launcher\_plugins::Error, 98
  - isFloat
    - rstudio::launcher\_plugins::json::Value, 333
  - isGlusterFsMountSource
    - rstudio::launcher\_plugins::api::MountSource, 218
  - isHidden
    - rstudio::launcher\_plugins::system::FilePath, 131
  - isHostMountSource
    - rstudio::launcher\_plugins::api::MountSource, 218
  - isInfinity
    - rstudio::launcher\_plugins::system::TimeDuration, 305
  - isInt
    - rstudio::launcher\_plugins::json::Value, 333
  - isInt64
    - rstudio::launcher\_plugins::json::Value, 333
  - isMissingMemberError
    - Json.hpp, 356
  - isNfsMountSource
    - rstudio::launcher\_plugins::api::MountSource, 218
  - isNull
    - rstudio::launcher\_plugins::json::Value, 334
  - isObject
    - rstudio::launcher\_plugins::json::Value, 334
  - isPassthroughMountSource
    - rstudio::launcher\_plugins::api::MountSource, 218
  - isReadable
    - rstudio::launcher\_plugins::system::FilePath, 131
  - IsReadOnly
    - rstudio::launcher\_plugins::api::Mount, 211
  - isRegularFile
    - rstudio::launcher\_plugins::system::FilePath, 132
  - isRootPath
    - rstudio::launcher\_plugins::system::FilePath, 132
  - isString
    - rstudio::launcher\_plugins::json::Value, 334
  - isSymlink
    - rstudio::launcher\_plugins::system::FilePath, 132
  - isType
    - Json.hpp, 356
  - isUInt
    - rstudio::launcher\_plugins::json::Value, 334
  - isUInt64
    - rstudio::launcher\_plugins::json::Value, 335
  - isValueNotFoundError
    - rstudio::launcher\_plugins::options::AbstractUserProfiles, 44
  - isWithin
    - rstudio::launcher\_plugins::system::FilePath, 132
  - isWriteable
    - rstudio::launcher\_plugins::system::FilePath, 133
  - Iterator
    - rstudio::launcher\_plugins::json::Array::Iterator, 163

- rstudio::launcher\_plugins::json::Object::Iterator, 159
- Job
  - rstudio::launcher\_plugins::api::Job, 168
- JOB\_NETWORK
  - rstudio::launcher\_plugins::api::Response, 293
- JOB\_OUTPUT
  - rstudio::launcher\_plugins::api::Response, 293
- JOB\_RESOURCE\_UTIL
  - rstudio::launcher\_plugins::api::Response, 293
- JOB\_STATE
  - rstudio::launcher\_plugins::api::Response, 293
- JOB\_STATUS
  - rstudio::launcher\_plugins::api::Response, 293
- JobConfig
  - rstudio::launcher\_plugins::api::JobConfig, 176
- JobIdRequest
  - rstudio::launcher\_plugins::api::JobIdRequest, 179
- JobLock
  - rstudio::launcher\_plugins::api::JobLock, 181
- JobStateResponse
  - rstudio::launcher\_plugins::api::JobStateResponse, 186
- JobStatusResponse
  - rstudio::launcher\_plugins::api::JobStatusResponse, 192
- Json.hpp
  - asJsonType, 354, 355
  - isMissingMemberError, 356
  - isType, 356
  - jsonReadError, 357
  - readObject, 357–363
  - toJsonArray, 364
  - toJsonValue, 365–367
  - Type, 354
  - typeAsString, 367
- jsonReadError
  - Json.hpp, 357
- KILL
  - rstudio::launcher\_plugins::api::ControlJobRequest, 80
- KILLED
  - rstudio::launcher\_plugins::api::Job, 167
- killJob
  - rstudio::launcher\_plugins::api::IJobSource, 149
  - rstudio::launcher\_plugins::local::LocalJobSource, 201
  - rstudio::launcher\_plugins::quickstart::QuickStartJobSource, 274
- LastUpdateTime
  - rstudio::launcher\_plugins::api::Job, 173
- LocalJobRepository
  - rstudio::launcher\_plugins::local::LocalJobRepository, 194
- LocalJobRunner
  - rstudio::launcher\_plugins::local::LocalJobRunner, 195
- LocalJobSource
  - rstudio::launcher\_plugins::local::LocalJobSource, 198
- LocalPluginApi
  - rstudio::launcher\_plugins::local::LocalPluginApi, 206
- LocalResourceStream
  - rstudio::launcher\_plugins::local::LocalResourceStream, 207
- logDebugMessage
  - Logger.hpp, 371
- logError
  - Logger.hpp, 372
- logErrorAsDebug
  - Logger.hpp, 373
- logErrorAsInfo
  - Logger.hpp, 373
- logErrorAsWarning
  - Logger.hpp, 373
- logErrorMessage
  - Logger.hpp, 373, 374
- Logger.hpp
  - addLogDestination, 369, 370
  - cleanDelimiters, 370
  - cleanDelims, 370
  - logDebugMessage, 371
  - logError, 372
  - logErrorAsDebug, 373
  - logErrorAsInfo, 373
  - logErrorAsWarning, 373
  - logErrorMessage, 373, 374
  - logInfoMessage, 375
  - logWarningMessage, 376, 377
  - removeLogDestination, 377
  - setProgramId, 377
  - writeError, 378
- logInfoMessage
  - Logger.hpp, 375
- logWarningMessage
  - Logger.hpp, 376, 377
- m\_job
  - rstudio::launcher\_plugins::api::AbstractOutputStream, 34
  - rstudio::launcher\_plugins::api::AbstractResourceStream, 38
- m\_jobRepository
  - rstudio::launcher\_plugins::api::IJobSource, 152
- m\_jobStatusNotifier
  - rstudio::launcher\_plugins::api::IJobSource, 152
- m\_mutex
  - rstudio::launcher\_plugins::api::AbstractMultiStream< R, Args >, 30
- m\_outputType
  - rstudio::launcher\_plugins::api::AbstractOutputStream, 34
- makeCurrent



- rstudio::launcher\_plugins::system::FilePath, 133
- makeCurrentPath
  - rstudio::launcher\_plugins::system::FilePath, 134
- matchesTags
  - rstudio::launcher\_plugins::api::Job, 169
- MaxValue
  - rstudio::launcher\_plugins::api::ResourceLimit, 287
- Member
  - rstudio::launcher\_plugins::json::Object::Member, 209
- MEMORY
  - rstudio::launcher\_plugins::api::ResourceLimit::Type, 310
- MEMORY\_SWAP
  - rstudio::launcher\_plugins::api::ResourceLimit::Type, 310
- mergeObjects
  - rstudio::launcher\_plugins::json::Object, 239
- Microseconds
  - rstudio::launcher\_plugins::system::TimeDuration, 305
- Minutes
  - rstudio::launcher\_plugins::system::TimeDuration, 306
- Mounts
  - rstudio::launcher\_plugins::api::Job, 173
  - rstudio::launcher\_plugins::system::process::ProcessOptions, 265
- move
  - rstudio::launcher\_plugins::system::FilePath, 134
- MoveCrossDevice
  - rstudio::launcher\_plugins::system::FilePath, 119
- MoveDirect
  - rstudio::launcher\_plugins::system::FilePath, 119
- moveIndirect
  - rstudio::launcher\_plugins::system::FilePath, 134
- MoveType
  - rstudio::launcher\_plugins::system::FilePath, 118
- MultiStreamResponse
  - rstudio::launcher\_plugins::api::MultiStreamResponse, 220
- Name
  - rstudio::launcher\_plugins::api::Job, 173
  - rstudio::launcher\_plugins::api::JobConfig, 177
  - rstudio::launcher\_plugins::api::PlacementConstraint, 261
  - rstudio::launcher\_plugins::system::posix::IpAddress, 157
- NetworkResponse
  - rstudio::launcher\_plugins::api::NetworkResponse, 223
- NFS
  - rstudio::launcher\_plugins::api::MountSource, 213
- Object
  - rstudio::launcher\_plugins::json::Object, 230, 231
- onAddRequest
  - rstudio::launcher\_plugins::api::AbstractMultiStream< R, Args >, 28
- OnComplete
  - rstudio::launcher\_plugins::api::AbstractOutputStream, 31
- onDataReceived
  - rstudio::launcher\_plugins::comms::AbstractLauncherCommunicator, 21
- OnError
  - Functionals.hpp, 388
- onRemoveRequest
  - rstudio::launcher\_plugins::api::AbstractMultiStream< R, Args >, 29
- openForRead
  - rstudio::launcher\_plugins::system::FilePath, 135
- openForWrite
  - rstudio::launcher\_plugins::system::FilePath, 135
- Operation
  - rstudio::launcher\_plugins::api::ControlJobRequest, 80
- operator bool
  - rstudio::launcher\_plugins::Error, 98
  - rstudio::launcher\_plugins::Optional< T >, 247
- operator!
  - rstudio::launcher\_plugins::Error, 98
  - rstudio::launcher\_plugins::Optional< T >, 247
- operator!=
  - rstudio::launcher\_plugins::Error, 99
  - rstudio::launcher\_plugins::json::Array::Iterator, 163
  - rstudio::launcher\_plugins::json::Object::Iterator, 159
  - rstudio::launcher\_plugins::json::Value, 335
  - rstudio::launcher\_plugins::system::DateTime, 84
  - rstudio::launcher\_plugins::system::FilePath, 135
  - rstudio::launcher\_plugins::system::TimeDuration, 306
  - rstudio::launcher\_plugins::system::User, 316
- operator<
  - rstudio::launcher\_plugins::system::DateTime, 87
  - rstudio::launcher\_plugins::system::FilePath, 136
  - rstudio::launcher\_plugins::system::TimeDuration, 307
- operator<=
  - rstudio::launcher\_plugins::system::DateTime, 87
  - rstudio::launcher\_plugins::system::TimeDuration, 307
- operator>
  - rstudio::launcher\_plugins::system::DateTime, 88
  - rstudio::launcher\_plugins::system::TimeDuration, 308
- operator>=
  - rstudio::launcher\_plugins::system::DateTime, 89
  - rstudio::launcher\_plugins::system::TimeDuration, 309
- operator\*
  - rstudio::launcher\_plugins::json::Array::Iterator, 164
  - rstudio::launcher\_plugins::json::Object::Iterator, 160

- operator()
  - rstudio::launcher\_plugins::options::Options::Init, 155, 156
- operator+
  - rstudio::launcher\_plugins::system::DateTime, 85
- operator++
  - rstudio::launcher\_plugins::json::Array::Iterator, 164
  - rstudio::launcher\_plugins::json::Object::Iterator, 160
- operator+=
  - rstudio::launcher\_plugins::system::DateTime, 85
- operator-
  - rstudio::launcher\_plugins::system::DateTime, 85, 86
- operator--
  - rstudio::launcher\_plugins::json::Array::Iterator, 164
  - rstudio::launcher\_plugins::json::Object::Iterator, 160, 161
- operator-=
  - rstudio::launcher\_plugins::system::DateTime, 86
- operator=
  - rstudio::launcher\_plugins::api::Job, 170
  - rstudio::launcher\_plugins::api::StreamSequenceId, 297
  - rstudio::launcher\_plugins::ErrorLocation, 103
  - rstudio::launcher\_plugins::json::Array, 51, 52
  - rstudio::launcher\_plugins::json::Array::Iterator, 165
  - rstudio::launcher\_plugins::json::Object, 240
  - rstudio::launcher\_plugins::json::Object::Iterator, 161
  - rstudio::launcher\_plugins::json::Value, 335, 336, 338, 340, 342
  - rstudio::launcher\_plugins::Optional< T >, 247, 248
  - rstudio::launcher\_plugins::system::DateTime, 87, 88
  - rstudio::launcher\_plugins::system::TimeDuration, 307, 308
  - rstudio::launcher\_plugins::system::User, 316, 317
- operator==
  - rstudio::launcher\_plugins::Error, 100
  - rstudio::launcher\_plugins::ErrorLocation, 104
  - rstudio::launcher\_plugins::json::Array::Iterator, 165
  - rstudio::launcher\_plugins::json::Object::Iterator, 161
  - rstudio::launcher\_plugins::json::Value, 342
  - rstudio::launcher\_plugins::system::DateTime, 88
  - rstudio::launcher\_plugins::system::FilePath, 136
  - rstudio::launcher\_plugins::system::TimeDuration, 308
  - rstudio::launcher\_plugins::system::User, 317
- operator[]
  - rstudio::launcher\_plugins::json::Array, 52
  - rstudio::launcher\_plugins::json::Object, 241
- Optional
  - rstudio::launcher\_plugins::Optional< T >, 245
- OutputStreamResponse
  - rstudio::launcher\_plugins::api::OutputStreamResponse, 257, 258
- Owner
  - rstudio::launcher\_plugins::system::process::ProcessInfo, 263
- PamProfile
  - rstudio::launcher\_plugins::system::process::ProcessOptions, 265
- parse
  - rstudio::launcher\_plugins::json::Array, 52, 53
  - rstudio::launcher\_plugins::json::Object, 241, 242
  - rstudio::launcher\_plugins::json::Value, 342, 344
- parseAndValidate
  - rstudio::launcher\_plugins::json::Value, 344
- PASSTHROUGH
  - rstudio::launcher\_plugins::api::MountSource, 213
- Password
  - rstudio::launcher\_plugins::system::process::ProcessOptions, 265
- PENDING
  - rstudio::launcher\_plugins::api::Job, 167
- PGrp
  - rstudio::launcher\_plugins::system::process::ProcessInfo, 263
- Pid
  - rstudio::launcher\_plugins::api::Job, 173
  - rstudio::launcher\_plugins::system::process::ProcessInfo, 263
- PlacementConstraint
  - rstudio::launcher\_plugins::api::PlacementConstraint, 260
- PlacementConstraints
  - rstudio::launcher\_plugins::api::Job, 173
  - rstudio::launcher\_plugins::api::JobSourceConfiguration, 182
- posixCall
  - PosixSystem.hpp, 386
- PosixSystem.hpp
  - enableCoreDumps, 385
  - getEnvironmentVariable, 385
  - getIpAddresses, 385
  - ignoreSignal, 386
  - posixCall, 386
  - realUserIsRoot, 387
  - restorePrivileges, 387
  - restoreRoot, 387
  - temporarilyDropPrivileges, 388
- post
  - rstudio::launcher\_plugins::system::asioService, 60
- PPid
  - rstudio::launcher\_plugins::system::process::ProcessInfo, 263
- Protocol
  - rstudio::launcher\_plugins::api::ExposedPort, 108
- PublishedPort
  - rstudio::launcher\_plugins::api::ExposedPort, 108
- push\_back
  - rstudio::launcher\_plugins::json::Array, 53–57
- Queues



- rstudio::launcher\_plugins::api::Job, [174](#)
  - rstudio::launcher\_plugins::api::JobSourceConfiguration, [182](#)
- QuickStartJobRepository
  - rstudio::launcher\_plugins::quickstart::QuickStartJobRepository, [269](#)
- QuickStartJobSource
  - rstudio::launcher\_plugins::quickstart::QuickStartJobSource, [271](#)
- QuickStartJobStatusWatcher
  - rstudio::launcher\_plugins::quickstart::QuickStartJobStatusWatcher, [277](#)
- QuickStartPluginApi
  - rstudio::launcher\_plugins::quickstart::QuickStartPluginApi, [280](#)
- QuickStartResourceStream
  - rstudio::launcher\_plugins::quickstart::QuickStartResourceStream, [281](#)
- random
  - Crypto.hpp, [383](#)
- rbegin
  - rstudio::launcher\_plugins::json::Array, [57](#)
  - rstudio::launcher\_plugins::json::Object, [242](#)
- readBytes
  - rstudio::launcher\_plugins::system::AsioStream, [62](#)
- readObject
  - Json.hpp, [357–363](#)
- readOptions
  - rstudio::launcher\_plugins::options::Options, [254](#)
- realUsersRoot
  - PosixSystem.hpp, [387](#)
- RecursivelterationFunction
  - rstudio::launcher\_plugins::system::FilePath, [118](#)
- registerOptions
  - rstudio::launcher\_plugins::options::Options, [254](#)
- registerRequestHandler
  - rstudio::launcher\_plugins::comms::AbstractLauncherComms, [22](#)
- remove
  - rstudio::launcher\_plugins::system::FilePath, [137](#)
- removeIfExists
  - rstudio::launcher\_plugins::system::FilePath, [137](#)
- removeJob
  - rstudio::launcher\_plugins::jobs::AbstractJobRepository, [18](#)
- removeLogDestination
  - Logger.hpp, [377](#)
- RemoveOnExitScope
  - rstudio::launcher\_plugins::system::RemoveOnExitScope, [282](#)
- removeRequest
  - rstudio::launcher\_plugins::api::AbstractMultiStream< R, Args >, [29](#)
- rend
  - rstudio::launcher\_plugins::json::Array, [57](#)
  - rstudio::launcher\_plugins::json::Object, [242](#)
- reportData
  - rstudio::launcher\_plugins::api::AbstractOutputStream, [32](#)
  - rstudio::launcher\_plugins::api::AbstractResourceStream, [38](#)
  - rstudio::launcher\_plugins::api::AbstractOutputStream, [32](#)
  - rstudio::launcher\_plugins::api::AbstractResourceStream, [38](#)
  - rstudio::launcher\_plugins::comms::AbstractLauncherCommunicator, [22](#)
  - rstudio::launcher\_plugins::system::AsyncTimedEvent, [65](#)
  - rstudio::launcher\_plugins::api::Request, [284](#)
  - resetDirectory
    - rstudio::launcher\_plugins::system::FilePath, [137](#)
  - ResidentMem
    - rstudio::launcher\_plugins::api::ResourceUtilData, [289](#)
  - resolveAliasedPath
    - rstudio::launcher\_plugins::system::FilePath, [137](#)
  - resolveSymlink
    - rstudio::launcher\_plugins::system::FilePath, [138](#)
  - ResourceLimit
    - rstudio::launcher\_plugins::api::ResourceLimit, [286](#)
  - ResourceLimits
    - rstudio::launcher\_plugins::api::Job, [174](#)
    - rstudio::launcher\_plugins::api::JobSourceConfiguration, [183](#)
  - ResourceType
    - rstudio::launcher\_plugins::api::ResourceLimit, [287](#)
  - ResourceUtilStreamResponse
    - rstudio::launcher\_plugins::api::ResourceUtilStreamResponse, [291](#)
  - Response
    - rstudio::launcher\_plugins::api::Response, [293](#)
  - RestoreCurrentPathScope
    - rstudio::launcher\_plugins::system::RestoreCurrentPathScope, [295](#)
  - restorePrivileges
    - PosixSystem.hpp, [387](#)
  - restoreRoot
    - PosixSystem.hpp, [387](#)
  - RESUME
    - rstudio::launcher\_plugins::api::ControlJobRequest, [80](#)
  - resumeJob
    - rstudio::launcher\_plugins::api::IJobSource, [150](#)
    - rstudio::launcher\_plugins::local::LocalJobSource, [201](#)
    - rstudio::launcher\_plugins::quickstart::QuickStartJobSource, [274](#)
  - rstudio::launcher\_plugins::AbstractMain, [24](#)
  - run, [24](#)
  - rstudio::launcher\_plugins::api::AbstractMultiStream< R, Args >, [25](#)
  - AbstractMultiStream, [26](#)

- addRequest, 26
- initialize, 28
- isEmpty, 28
- m\_mutex, 30
- onAddRequest, 28
- onRemoveRequest, 29
- removeRequest, 29
- sendResponse, 29, 30
- rstudio::launcher\_plugins::api::AbstractOutputStream, 30
  - AbstractOutputStream, 32
  - m\_job, 34
  - m\_outputType, 34
  - OnComplete, 31
  - reportData, 32
  - reportError, 32
  - start, 34
- rstudio::launcher\_plugins::api::AbstractPluginApi, 34
  - AbstractPluginApi, 35
  - initialize, 35
- rstudio::launcher\_plugins::api::AbstractResourceStream, 36
  - AbstractResourceStream, 37
  - addRequest, 37
  - initialize, 37
  - m\_job, 38
  - reportData, 38
  - reportError, 38
  - setStreamComplete, 38
- rstudio::launcher\_plugins::api::AbstractTimedResourceStream, 40
  - AbstractTimedResourceStream, 41
  - initialize, 41
- rstudio::launcher\_plugins::api::AzureFileMountSource, 66
  - fromJson, 67
  - getSecretName, 67
  - getShareName, 68
- rstudio::launcher\_plugins::api::BootstrapRequest, 68
  - getMajorVersion, 69
  - getMinorVersion, 69
  - getPatchNumber, 69
- rstudio::launcher\_plugins::api::BootstrapResponse, 70
  - BootstrapResponse, 70
  - toJson, 71
- rstudio::launcher\_plugins::api::CephFsMountSource, 71
  - fromJson, 72
  - getMonitors, 73
  - getPath, 73
  - getSecretFile, 73
  - getSecretRef, 73
  - getUser, 74
- rstudio::launcher\_plugins::api::ClusterInfoResponse, 74
  - ClusterInfoResponse, 75
  - toJson, 75
- rstudio::launcher\_plugins::api::Container, 76
  - fromJson, 76
  - Image, 77
  - RunAsGroupId, 77
  - RunAsUserId, 77
  - SupplementalGroupIds, 77
  - toJson, 77
- rstudio::launcher\_plugins::api::ContainerConfiguration, 78
  - AllowUnknownImages, 78
  - ContainerImages, 78
  - DefaultImage, 78
  - SupportsContainers, 78
- rstudio::launcher\_plugins::api::ControlJobRequest, 79
  - CANCEL, 80
  - FIRST, 80
  - getOperation, 80
  - INVALID, 80
  - KILL, 80
  - Operation, 80
  - RESUME, 80
  - STOP, 80
- rstudio::launcher\_plugins::api::ControlJobResponse, 81
  - ControlJobResponse, 81
  - toJson, 82
- rstudio::launcher\_plugins::api::ErrorResponse, 105
  - ErrorResponse, 106
  - toJson, 106
- rstudio::launcher\_plugins::api::ExposedPort, 106
  - fromJson, 107
  - Protocol, 108
  - PublishedPort, 108
  - TargetPort, 108
  - toJson, 107
- rstudio::launcher\_plugins::api::FileOutputStream, 113
  - FileOutputStream, 114
  - start, 114
- rstudio::launcher\_plugins::api::GlusterFsMountSource, 141
  - fromJson, 141
  - getEndpoints, 142
  - getPath, 142
- rstudio::launcher\_plugins::api::HeartbeatResponse, 142
- rstudio::launcher\_plugins::api::HostMountSource, 143
  - fromJson, 144
  - getPath, 144
- rstudio::launcher\_plugins::api::IJobSource, 145
  - cancelJob, 146
  - createOutputStream, 147
  - createResourceStream, 147
  - getConfiguration, 148
  - getNetworkInfo, 148
  - IJobSource, 146
  - initialize, 149
  - killJob, 149
  - m\_jobRepository, 152
  - m\_jobStatusNotifier, 152
  - resumeJob, 150
  - stopJob, 150
  - submitJob, 151
  - suspendJob, 151

- rstudio::launcher\_plugins::api::Job, 165
  - Arguments, 171
  - CANCELED, 167
  - Cluster, 171
  - Command, 172
  - Config, 172
  - ContainerDetails, 172
  - Environment, 172
  - Exe, 172
  - ExitCode, 172
  - ExposedPorts, 173
  - FAILED, 167
  - FINISHED, 167
  - fromJson, 168
  - getJobConfigValue, 169
  - Host, 173
  - Id, 173
  - isCompleted, 169
  - Job, 168
  - KILLED, 167
  - LastUpdateTime, 173
  - matchesTags, 169
  - Mounts, 173
  - Name, 173
  - operator=, 170
  - PENDING, 167
  - Pid, 173
  - PlacementConstraints, 173
  - Queues, 174
  - ResourceLimits, 174
  - RUNNING, 167
  - StandardErrFile, 174
  - StandardIn, 174
  - StandardOutFile, 174
  - State, 167
  - stateFromString, 170
  - stateToString, 171
  - Status, 174
  - StatusMessage, 174
  - SubmissionTime, 174
  - SUSPENDED, 167
  - Tags, 175
  - toJson, 171
  - UNKNOWN, 167
  - User, 175
  - WorkingDirectory, 175
- rstudio::launcher\_plugins::api::JobConfig, 175
  - ENUM, 176
  - FLOAT, 176
  - fromJson, 177
  - INT, 176
  - JobConfig, 176
  - Name, 177
  - STRING, 176
  - toJson, 177
  - Type, 176
  - Value, 177
  - ValueType, 178
- rstudio::launcher\_plugins::api::JobIdRequest, 178
  - getEncodedJobId, 180
  - getJobId, 180
  - JobIdRequest, 179
- rstudio::launcher\_plugins::api::JobLock, 180
  - JobLock, 181
- rstudio::launcher\_plugins::api::JobSourceConfiguration, 182
  - ContainerConfig, 182
  - CustomConfig, 182
  - PlacementConstraints, 182
  - Queues, 182
  - ResourceLimits, 183
- rstudio::launcher\_plugins::api::JobStateRequest, 183
  - getEndTime, 184
  - getFieldSet, 184
  - getStartTime, 184
  - getStatusSet, 185
  - getTagSet, 185
- rstudio::launcher\_plugins::api::JobStateResponse, 186
  - JobStateResponse, 186
  - toJson, 187
- rstudio::launcher\_plugins::api::JobStatusRequest, 190
  - isCancelRequest, 191
- rstudio::launcher\_plugins::api::JobStatusResponse, 192
  - JobStatusResponse, 192
  - toJson, 193
- rstudio::launcher\_plugins::api::Mount, 210
  - Destination, 211
  - fromJson, 210
  - IsReadOnly, 211
  - Source, 211
  - toJson, 211
- rstudio::launcher\_plugins::api::MountSource, 212
  - asAzureFileMountSource, 214
  - asCephFsMountSource, 214, 215
  - asGlusterFsMountSource, 215
  - asHostMountSource, 216
  - asNfsMountSource, 216, 217
  - AZURE\_FILE, 213
  - CEPH\_FS, 213
  - CustomType, 219
  - fromJson, 217
  - GLUSTER\_FS, 213
  - HOST, 213
  - isAzureFileMountSource, 217
  - isCephFsMountSource, 218
  - isGlusterFsMountSource, 218
  - isHostMountSource, 218
  - isNfsMountSource, 218
  - isPassthroughMountSource, 218
  - NFS, 213
  - PASSTHROUGH, 213
  - SourceObject, 219
  - SourceType, 219
  - toJson, 219
  - Type, 213

- rstudio::launcher\_plugins::api::MultiStreamResponse, 220
  - MultiStreamResponse, 220
  - toJson, 221
- rstudio::launcher\_plugins::api::NetworkInfo, 221
  - Hostname, 222
  - IpAddresses, 222
- rstudio::launcher\_plugins::api::NetworkRequest, 222
- rstudio::launcher\_plugins::api::NetworkResponse, 223
  - NetworkResponse, 223
  - toJson, 224
- rstudio::launcher\_plugins::api::NfsMountSource, 224
  - fromJson, 225
  - getHost, 225
  - getPath, 226
- rstudio::launcher\_plugins::api::OutputStreamRequest, 255
  - getStreamType, 256
  - isCancelRequest, 256
- rstudio::launcher\_plugins::api::OutputStreamResponse, 257
  - OutputStreamResponse, 257, 258
  - toJson, 258
- rstudio::launcher\_plugins::api::PlacementConstraint, 259
  - fromJson, 261
  - Name, 261
  - PlacementConstraint, 260
  - toJson, 261
  - Value, 261
- rstudio::launcher\_plugins::api::Request, 282
  - BOOTSTRAP, 284
  - CONTROL\_JOB, 284
  - fromJson, 284
  - GET\_CLUSTER\_INFO, 284
  - GET\_JOB, 284
  - GET\_JOB\_NETWORK, 284
  - GET\_JOB\_OUTPUT, 284
  - GET\_JOB\_RESOURCE\_UTIL, 284
  - GET\_JOB\_STATUS, 284
  - getId, 285
  - getType, 285
  - HEARTBEAT, 284
  - INVALID, 284
  - Request, 284
  - SUBMIT\_JOB, 284
  - Type, 283
- rstudio::launcher\_plugins::api::ResourceLimit, 285
  - DefaultValue, 287
  - fromJson, 287
  - MaxValue, 287
  - ResourceLimit, 286
  - ResourceType, 287
  - toJson, 287
  - Value, 288
- rstudio::launcher\_plugins::api::ResourceLimit::Type, 310
  - CPU\_COUNT, 310
  - CPU\_TIME, 310
  - MEMORY, 310
  - MEMORY\_SWAP, 310
- rstudio::launcher\_plugins::api::ResourceUtilData, 288
  - CpuPercent, 288
  - CpuSeconds, 288
  - ResidentMem, 289
  - VirtualMem, 289
- rstudio::launcher\_plugins::api::ResourceUtilStreamRequest, 289
  - isCancelRequest, 290
- rstudio::launcher\_plugins::api::ResourceUtilStreamResponse, 290
  - ResourceUtilStreamResponse, 291
  - toJson, 291
- rstudio::launcher\_plugins::api::Response, 292
  - BOOTSTRAP, 293
  - CLUSTER\_INFO, 293
  - CONTROL\_JOB, 293
  - ERROR, 293
  - HEARTBEAT, 293
  - JOB\_NETWORK, 293
  - JOB\_OUTPUT, 293
  - JOB\_RESOURCE\_UTIL, 293
  - JOB\_STATE, 293
  - JOB\_STATUS, 293
  - Response, 293
  - toJson, 294
  - Type, 293
- rstudio::launcher\_plugins::api::StreamSequenceId, 295
  - operator=, 297
  - StreamSequenceId, 296
  - toJson, 297
- rstudio::launcher\_plugins::api::SubmitJobRequest, 298
  - getJob, 299
- rstudio::launcher\_plugins::api::UserRequest, 318
  - getRequestUsername, 319
  - getUser, 319
  - UserRequest, 318
- rstudio::launcher\_plugins::comms::AbstractLauncherCommunicator, 20
  - AbstractLauncherCommunicator, 21
  - onDataReceived, 21
  - registerRequestHandler, 22
  - reportError, 22
  - sendResponse, 22
  - shared\_from\_derived, 22
  - start, 23
  - stop, 23
  - waitForExit, 23
- rstudio::launcher\_plugins::Error, 90
  - addOrUpdateProperty, 94
  - addProperty, 95
  - asString, 96
  - Error, 92, 93
  - getCause, 96
  - getCode, 96
  - getLocation, 96

- getMessage, 97
- getName, 97
- getProperties, 97
- getProperty, 97
- getSummary, 98
- isExpected, 98
- operator bool, 98
- operator!, 98
- operator!=, 99
- operator==, 100
- rstudio::launcher\_plugins::ErrorLocation, 100
  - asString, 102
  - ErrorLocation, 101, 102
  - getFile, 102
  - getFunction, 102
  - getLine, 103
  - hasLocation, 103
  - operator=, 103
  - operator==, 104
- rstudio::launcher\_plugins::ErrorLock, 104
- rstudio::launcher\_plugins::jobs::AbstractJobRepository, 15
  - AbstractJobRepository, 16
  - addJob, 16
  - getJob, 17
  - getJobs, 17
  - initialize, 17
  - removeJob, 18
- rstudio::launcher\_plugins::jobs::AbstractJobStatusWatcher, 18
  - AbstractJobStatusWatcher, 19
  - updateJobStatus, 19
- rstudio::launcher\_plugins::jobs::AbstractTimedJobStatusWatcher, 39
  - AbstractTimedJobStatusWatcher, 40
- rstudio::launcher\_plugins::jobs::JobStatusNotifier, 187
  - subscribe, 188
  - updateJob, 190
- rstudio::launcher\_plugins::json::Array, 46
  - Array, 48
  - begin, 49
  - end, 49
  - erase, 49, 50
  - getBack, 50
  - getFront, 50
  - getSize, 50
  - getValueAt, 51
  - isEmpty, 51
  - operator=, 51, 52
  - operator[], 52
  - parse, 52, 53
  - push\_back, 53–57
  - rbegin, 57
  - rend, 57
  - toSetString, 58
  - toStringPairList, 58
  - toVectorInt, 58
  - toVectorString, 59
- rstudio::launcher\_plugins::json::Array::Iterator, 162
  - Iterator, 163
  - operator!=, 163
  - operator\*, 164
  - operator++, 164
  - operator--, 164
  - operator=, 165
  - operator==, 165
- rstudio::launcher\_plugins::json::detail::is\_json\_type< T >, 157
- rstudio::launcher\_plugins::json::Object, 227
  - begin, 231
  - createMember, 231
  - end, 232
  - erase, 232, 233
  - find, 233
  - getSchemaDefaults, 234
  - getSize, 234
  - hasMember, 234, 235
  - insert, 235–239
  - isEmpty, 239
  - mergeObjects, 239
  - Object, 230, 231
  - operator=, 240
  - operator[], 241
  - parse, 241, 242
  - rbegin, 242
  - rend, 242
  - toStringMap, 243
  - toStringPairList, 243
- rstudio::launcher\_plugins::json::Object::Iterator, 158
  - Iterator, 159
  - operator!=, 159
  - operator\*, 160
  - operator++, 160
  - operator--, 160, 161
  - operator=, 161
  - operator==, 161
- rstudio::launcher\_plugins::json::Object::Member, 208
  - getName, 209
  - getValue, 209
  - Member, 209
- rstudio::launcher\_plugins::json::Value, 322
  - clone, 329
  - coerce, 329
  - getArray, 329
  - getBool, 329
  - getDouble, 330
  - getFloat, 330
  - getInt, 330
  - getInt64, 330
  - getObject, 331
  - getString, 331
  - getType, 331
  - getUInt, 331
  - getUInt64, 332
  - getValue, 332
  - isArray, 332

- isBool, 333
- isDouble, 333
- isFloat, 333
- isInt, 333
- isInt64, 333
- isNull, 334
- isObject, 334
- isString, 334
- isUInt, 334
- isUInt64, 335
- operator!=, 335
- operator=, 335, 336, 338, 340, 342
- operator==, 342
- parse, 342, 344
- parseAndValidate, 344
- setValueAtPath, 344–349
- validate, 349
- Value, 325–328
- write, 349
- writeFormatted, 350
- rstudio::launcher\_plugins::local::LocalJobRepository, 193
  - LocalJobRepository, 194
  - saveJob, 194
  - setJobOutputPaths, 194
- rstudio::launcher\_plugins::local::LocalJobRunner, 195
  - initialize, 196
  - LocalJobRunner, 195
  - runJob, 196
- rstudio::launcher\_plugins::local::LocalJobSource, 197
  - cancelJob, 198
  - createOutputStream, 198
  - createResourceStream, 199
  - getConfiguration, 199
  - getNetworkInfo, 200
  - initialize, 200
  - killJob, 201
  - LocalJobSource, 198
  - resumeJob, 201
  - stopJob, 202
  - submitJob, 202
  - suspendJob, 203
- rstudio::launcher\_plugins::local::LocalOptions, 203
  - getInstance, 204
  - getNodeConnectionTimeoutSeconds, 204
  - getSecureCookieKeyFile, 204
  - initialize, 205
  - shouldSaveUnspecifiedOutput, 205
- rstudio::launcher\_plugins::local::LocalPluginApi, 205
  - LocalPluginApi, 206
- rstudio::launcher\_plugins::local::LocalResourceStream, 206
  - LocalResourceStream, 207
- rstudio::launcher\_plugins::local::LocalSecureCookie, 207
  - getKey, 208
  - initialize, 208
- rstudio::launcher\_plugins::logging::FileLogDestination, 108
  - FileLogDestination, 109
  - getId, 109
  - writeLog, 110
- rstudio::launcher\_plugins::logging::FileLogOptions, 110
  - doRotation, 112
  - FileLogOptions, 111
  - getDirectory, 112
  - getFileMode, 112
  - getMaxSizeMb, 112
  - includePid, 113
- rstudio::launcher\_plugins::logging::ILogDestination, 152
  - getId, 153
  - getLogLevel, 154
  - ILogDestination, 153
  - writeLog, 154
- rstudio::launcher\_plugins::Noncopyable, 226
- rstudio::launcher\_plugins::Optional< T >, 244
  - getValueOr, 246
  - hasValue, 247
  - operator bool, 247
  - operator!, 247
  - operator=, 247, 248
  - Optional, 245
- rstudio::launcher\_plugins::options::AbstractUserProfiles, 42
  - AbstractUserProfiles, 43
  - getValueForUser, 43
  - initialize, 44
  - isValueNotFoundError, 44
  - validateValue, 44, 45
- rstudio::launcher\_plugins::options::Options, 249
  - enableDebugLogging, 250
  - getHeartbeatIntervalSeconds, 251
  - getInstance, 251
  - getJobExpiryHours, 251
  - getLauncherConfigFile, 251
  - getLoggingDir, 252
  - getLogLevel, 252
  - getMaxMessageSize, 252
  - getPluginName, 252
  - getRSandboxPath, 253
  - getScratchPath, 253
  - getServerUser, 253
  - getThreadPoolSize, 254
  - readOptions, 254
  - registerOptions, 254
  - useUnprivilegedMode, 255
- rstudio::launcher\_plugins::options::Options::Init, 154
  - Init, 155
  - operator(), 155, 156
- rstudio::launcher\_plugins::options::Value< T >, 319
  - setDefaultValue, 321
  - Value, 321
- rstudio::launcher\_plugins::quickstart::QuickStartJobRepository, 269
  - QuickStartJobRepository, 269

- rstudio::launcher\_plugins::quickstart::QuickStartJobSource, 270
  - cancelJob, 271
  - createOutputStream, 272
  - createResourceStream, 272
  - getConfiguration, 273
  - getNetworkInfo, 273
  - initialize, 274
  - killJob, 274
  - QuickStartJobSource, 271
  - resumeJob, 274
  - stopJob, 275
  - submitJob, 275
  - suspendJob, 276
- rstudio::launcher\_plugins::quickstart::QuickStartJobStatusWatcher, 277
  - QuickStartJobStatusWatcher, 277
- rstudio::launcher\_plugins::quickstart::QuickStartOptions, 278
  - getInstance, 278
  - getSampleOption, 278
  - initialize, 279
- rstudio::launcher\_plugins::quickstart::QuickStartPluginApi, 279
  - QuickStartPluginApi, 280
- rstudio::launcher\_plugins::quickstart::QuickStartResourceStream, 280
  - initialize, 281
  - QuickStartResourceStream, 281
- rstudio::launcher\_plugins::Success, 299
- rstudio::launcher\_plugins::system::AsioService, 59
  - post, 60
  - setSignalHandler, 60
  - startThreads, 61
  - stop, 61
- rstudio::launcher\_plugins::system::AsioStream, 61
  - AsioStream, 62
  - readBytes, 62
  - writeBytes, 62
- rstudio::launcher\_plugins::system::AsyncDeadlineEvent, 63
  - AsyncDeadlineEvent, 63, 64
- rstudio::launcher\_plugins::system::AsyncTimedEvent, 65
  - reportError, 65
  - start, 66
- rstudio::launcher\_plugins::system::DateTime, 82
  - DateTime, 83, 84
  - fromString, 84
  - operator!=, 84
  - operator<, 87
  - operator<=, 87
  - operator>, 88
  - operator>=, 89
  - operator+, 85
  - operator+=", 85
  - operator-, 85, 86
  - operator-=, 86
  - operator=, 87, 88
  - operator==, 88
  - toString, 89, 90
- rstudio::launcher\_plugins::system::FilePath, 115
  - changeFileMode, 119
  - changeOwnership, 120
  - completeChildPath, 120, 121
  - completePath, 121
  - copy, 121
  - copyDirectoryRecursive, 122
  - createAliasedPath, 122
  - createDirectory, 122
  - ensureDirectory, 123
  - ensureFile, 123
  - exists, 123
  - FilePath, 119
  - getAbsolutePath, 124
  - getAbsolutePathNative, 124
  - getCanonicalPath, 124
  - getChildren, 124
  - getChildrenRecursive, 125
  - getExtension, 125
  - getExtensionLowerCase, 125
  - getFileMode, 125
  - getFilename, 127
  - getLastWriteTime, 127
  - getLexicallyNormalPath, 127
  - getMimeType, 127
  - getParent, 128
  - getRelativePath, 128
  - getSize, 128
  - getSizeRecursive, 128
  - getStem, 129
  - hasExtension, 129
  - hasExtensionLowerCase, 129
  - hasTextMimeType, 130
  - isDirectory, 130
  - isEmpty, 130
  - isEqualCaseInsensitive, 130
  - isEquivalentTo, 131
  - isHidden, 131
  - isReadable, 131
  - isRegularFile, 132
  - isRootPath, 132
  - isSymlink, 132
  - isWithin, 132
  - isWriteable, 133
  - makeCurrent, 133
  - makeCurrentPath, 134
  - move, 134
  - MoveCrossDevice, 119
  - MoveDirect, 119
  - moveIndirect, 134
  - MoveType, 118
  - openForRead, 135
  - openForWrite, 135
  - operator!=, 135
  - operator<, 136



- operator==, 136
- RecursiveIterationFunction, 118
- remove, 137
- removeIfExists, 137
- resetDirectory, 137
- resolveAliasedPath, 137
- resolveSymlink, 138
- safeCurrentPath, 138
- setLastWriteTime, 138
- tempFilePath, 139
- testWritePermissions, 139
- uniqueFilePath, 140
- rstudio::launcher\_plugins::system::PathScopeImplDeleter, 259
- rstudio::launcher\_plugins::system::posix::IpAddress, 157
  - Address, 157
  - Name, 157
- rstudio::launcher\_plugins::system::process::AbstractChildProcess, 13
  - AbstractChildProcess, 14
  - getPid, 14
  - run, 14
  - terminate, 14
  - writeToStdin, 15
- rstudio::launcher\_plugins::system::process::AsyncProcessCallbacks, 64
- rstudio::launcher\_plugins::system::process::ProcessInfo, 262
  - Arguments, 263
  - Executable, 263
  - getProcessInfo, 262
  - Owner, 263
  - PGrp, 263
  - Pid, 263
  - PPid, 263
  - State, 264
- rstudio::launcher\_plugins::system::process::ProcessOptions, 264
  - CloseStdIn, 265
  - Mounts, 265
  - PamProfile, 265
  - Password, 265
  - RunAsUser, 266
  - UseSandbox, 266
- rstudio::launcher\_plugins::system::process::ProcessResult, 266
  - ExitCode, 267
  - StdError, 267
  - StdOut, 267
- rstudio::launcher\_plugins::system::process::ProcessSupervisor, 267
  - hasRunningChildren, 268
  - runAsyncProcess, 268
  - waitForExit, 269
- rstudio::launcher\_plugins::system::process::SyncChildProcess, 300
  - run, 301
- SyncChildProcess, 300
  - writeToStdin, 301
- rstudio::launcher\_plugins::system::RemoveOnExitScope, 281
  - RemoveOnExitScope, 282
- rstudio::launcher\_plugins::system::RestoreCurrentPathScope, 294
  - RestoreCurrentPathScope, 295
- rstudio::launcher\_plugins::system::TimeDuration, 301
  - getHours, 304
  - getMicroseconds, 304
  - getMinutes, 304
  - getSeconds, 304
  - Hours, 305
  - Infinity, 305
  - isInfinity, 305
  - Microseconds, 305
  - Minutes, 306
  - operator!=, 306
  - operator<, 307
  - operator<=, 307
  - operator>, 308
  - operator>=, 309
  - operator=, 307, 308
  - operator==, 308
  - Seconds, 309
  - TimeDuration, 303
- rstudio::launcher\_plugins::system::User, 310
  - exists, 312
  - getCurrentUser, 313
  - getGroupId, 313
  - getHomePath, 313
  - getShell, 314
  - getUserFromIdentifier, 314
  - getUserHomePath, 315
  - getUserId, 315
  - getUsername, 315
  - isAllUsers, 315
  - isEmpty, 316
  - operator!=, 316
  - operator=, 316, 317
  - operator==, 317
  - User, 312
- run
  - rstudio::launcher\_plugins::AbstractMain, 24
  - rstudio::launcher\_plugins::system::process::AbstractChildProcess, 14
  - rstudio::launcher\_plugins::system::process::SyncChildProcess, 301
  - RunAsGroupId
    - rstudio::launcher\_plugins::api::Container, 77
  - RunAsUser
    - rstudio::launcher\_plugins::system::process::ProcessOptions, 266
  - RunAsUserId
    - rstudio::launcher\_plugins::api::Container, 77
  - runAsyncProcess



- rstudio::launcher\_plugins::system::process::ProcessSupervisor, 268
- runJob
  - rstudio::launcher\_plugins::local::LocalJobRunner, 196
- RUNNING
  - rstudio::launcher\_plugins::api::Job, 167
- safeCurrentPath
  - rstudio::launcher\_plugins::system::FilePath, 138
- saveJob
  - rstudio::launcher\_plugins::local::LocalJobRepository, 194
- Seconds
  - rstudio::launcher\_plugins::system::TimeDuration, 309
- sendResponse
  - rstudio::launcher\_plugins::api::AbstractMultiStream< R, Args >, 29, 30
  - rstudio::launcher\_plugins::comms::AbstractLauncherCommunicator, 22
- setDefaultValue
  - rstudio::launcher\_plugins::options::Value< T >, 321
- setJobOutputPaths
  - rstudio::launcher\_plugins::local::LocalJobRepository, 194
- setLastWriteTime
  - rstudio::launcher\_plugins::system::FilePath, 138
- setProgramId
  - Logger.hpp, 377
- setSignalHandler
  - rstudio::launcher\_plugins::system::AsioService, 60
- setStreamComplete
  - rstudio::launcher\_plugins::api::AbstractResourceStream, 38
- setValueAtPointerPath
  - rstudio::launcher\_plugins::json::Value, 344–349
- shared\_from\_derived
  - rstudio::launcher\_plugins::comms::AbstractLauncherCommunicator, 22
- shouldSaveUnspecifiedOutput
  - rstudio::launcher\_plugins::local::LocalOptions, 205
- Source
  - rstudio::launcher\_plugins::api::Mount, 211
- SourceObject
  - rstudio::launcher\_plugins::api::MountSource, 219
- SourceType
  - rstudio::launcher\_plugins::api::MountSource, 219
- StandardErrFile
  - rstudio::launcher\_plugins::api::Job, 174
- StandardIn
  - rstudio::launcher\_plugins::api::Job, 174
- StandardOutFile
  - rstudio::launcher\_plugins::api::Job, 174
- start
  - rstudio::launcher\_plugins::api::AbstractOutputStream, 34
- rstudio::launcher\_plugins::api::FileOutputStream, 114
  - rstudio::launcher\_plugins::comms::AbstractLauncherCommunicator, 23
  - rstudio::launcher\_plugins::system::AsyncTimedEvent, 66
  - startThreads
    - rstudio::launcher\_plugins::system::AsioService, 61
  - State
    - rstudio::launcher\_plugins::api::Job, 167
    - rstudio::launcher\_plugins::system::process::ProcessInfo, 264
  - stateFromString
    - rstudio::launcher\_plugins::api::Job, 170
  - stateToString
    - rstudio::launcher\_plugins::api::Job, 171
  - Status
    - rstudio::launcher\_plugins::api::Job, 174
  - StatusMessage
    - rstudio::launcher\_plugins::api::Job, 174
  - stderr
    - rstudio::launcher\_plugins::system::process::ProcessResult, 267
  - StdOut
    - rstudio::launcher\_plugins::system::process::ProcessResult, 267
  - STOP
    - rstudio::launcher\_plugins::api::ControlJobRequest, 80
  - stop
    - rstudio::launcher\_plugins::comms::AbstractLauncherCommunicator, 23
    - rstudio::launcher\_plugins::system::AsioService, 61
  - stopJob
    - rstudio::launcher\_plugins::api::IJobSource, 150
    - rstudio::launcher\_plugins::local::LocalJobSource, 202
    - rstudio::launcher\_plugins::quickstart::QuickStartJobSource, 275
  - StreamSequenceId
    - rstudio::launcher\_plugins::api::StreamSequenceId, 296
  - STRING
    - rstudio::launcher\_plugins::api::JobConfig, 176
  - SubmissionTime
    - rstudio::launcher\_plugins::api::Job, 174
  - SUBMIT\_JOB
    - rstudio::launcher\_plugins::api::Request, 284
  - submitJob
    - rstudio::launcher\_plugins::api::IJobSource, 151
    - rstudio::launcher\_plugins::local::LocalJobSource, 202
    - rstudio::launcher\_plugins::quickstart::QuickStartJobSource, 275
  - subscribe
    - rstudio::launcher\_plugins::jobs::JobStatusNotifier, 188
  - SupplementalGroupIds

- rstudio::launcher\_plugins::api::Container, 77
- SupportsContainers
  - rstudio::launcher\_plugins::api::ContainerConfiguration, 78
- SUSPENDED
  - rstudio::launcher\_plugins::api::Job, 167
- suspendJob
  - rstudio::launcher\_plugins::api::JobSource, 151
  - rstudio::launcher\_plugins::local::LocalJobSource, 203
  - rstudio::launcher\_plugins::quickstart::QuickStartJobSource, 276
- SyncChildProcess
  - rstudio::launcher\_plugins::system::process::SyncChildProcess, 300
- Tags
  - rstudio::launcher\_plugins::api::Job, 175
- TargetPort
  - rstudio::launcher\_plugins::api::ExposedPort, 108
- tempFilePath
  - rstudio::launcher\_plugins::system::FilePath, 139
- temporarilyDropPrivileges
  - PosixSystem.hpp, 388
- terminate
  - rstudio::launcher\_plugins::system::process::AbstractChildProcess, 14
- testWritePermissions
  - rstudio::launcher\_plugins::system::FilePath, 139
- TimeDuration
  - rstudio::launcher\_plugins::system::TimeDuration, 303
- toJson
  - rstudio::launcher\_plugins::api::BootstrapResponse, 71
  - rstudio::launcher\_plugins::api::ClusterInfoResponse, 75
  - rstudio::launcher\_plugins::api::Container, 77
  - rstudio::launcher\_plugins::api::ControlJobResponse, 82
  - rstudio::launcher\_plugins::api::ErrorResponse, 106
  - rstudio::launcher\_plugins::api::ExposedPort, 107
  - rstudio::launcher\_plugins::api::Job, 171
  - rstudio::launcher\_plugins::api::JobConfig, 177
  - rstudio::launcher\_plugins::api::JobStateResponse, 187
  - rstudio::launcher\_plugins::api::JobStatusResponse, 193
  - rstudio::launcher\_plugins::api::Mount, 211
  - rstudio::launcher\_plugins::api::MountSource, 219
  - rstudio::launcher\_plugins::api::MultiStreamResponse, 221
  - rstudio::launcher\_plugins::api::NetworkResponse, 224
  - rstudio::launcher\_plugins::api::OutputStreamResponse, 258
  - rstudio::launcher\_plugins::api::PlacementConstraint, 261
  - rstudio::launcher\_plugins::api::ResourceLimit, 287
  - rstudio::launcher\_plugins::api::ResourceUtilStreamResponse, 291
  - rstudio::launcher\_plugins::api::Response, 294
  - rstudio::launcher\_plugins::api::StreamSequenceId, 297
  - toJsonArray
    - Json.hpp, 364
  - toJsonValue
    - Json.hpp, 365–367
  - toSetString
    - rstudio::launcher\_plugins::json::Array, 58
  - toString
    - rstudio::launcher\_plugins::system::DateTime, 89
  - toStringMap
    - rstudio::launcher\_plugins::json::Object, 243
  - toStringPairList
    - rstudio::launcher\_plugins::json::Array, 58
    - rstudio::launcher\_plugins::json::Object, 243
  - toVectorInt
    - rstudio::launcher\_plugins::json::Array, 58
  - toVectorString
    - rstudio::launcher\_plugins::json::Array, 59
  - Type
    - Json.hpp, 354
    - rstudio::launcher\_plugins::api::JobConfig, 176
    - rstudio::launcher\_plugins::api::MountSource, 213
    - rstudio::launcher\_plugins::api::Request, 283
    - rstudio::launcher\_plugins::api::Response, 293
  - typeAsString
    - Json.hpp, 367
  - uniqueFilePath
    - rstudio::launcher\_plugins::system::FilePath, 140
  - UNKNOWN
    - rstudio::launcher\_plugins::api::Job, 167
  - updateJob
    - rstudio::launcher\_plugins::jobs::JobStatusNotifier, 190
  - updateJobStatus
    - rstudio::launcher\_plugins::jobs::AbstractJobStatusWatcher, 19
  - User
    - rstudio::launcher\_plugins::api::Job, 175
    - rstudio::launcher\_plugins::system::User, 312
  - UserRequest
    - rstudio::launcher\_plugins::api::UserRequest, 318
  - UseSandbox
    - rstudio::launcher\_plugins::system::process::ProcessOptions, 266
  - useUnprivilegedMode
    - rstudio::launcher\_plugins::options::Options, 255
  - validate
    - rstudio::launcher\_plugins::json::Value, 349
  - validateValue
    - rstudio::launcher\_plugins::options::AbstractUserProfiles, 44, 45
  - Value

- [rstudio::launcher\\_plugins::api::JobConfig](#), [177](#)
  - [rstudio::launcher\\_plugins::api::PlacementConstraint](#),  
[261](#)
  - [rstudio::launcher\\_plugins::api::ResourceLimit](#), [288](#)
  - [rstudio::launcher\\_plugins::json::Value](#), [325–328](#)
  - [rstudio::launcher\\_plugins::options::Value< T >](#),  
[321](#)
- ValueType
  - [rstudio::launcher\\_plugins::api::JobConfig](#), [178](#)
- VirtualMem
  - [rstudio::launcher\\_plugins::api::ResourceUtilData](#),  
[289](#)
- waitForExit
  - [rstudio::launcher\\_plugins::comms::AbstractLauncherCommunicator](#),  
[23](#)
  - [rstudio::launcher\\_plugins::system::process::ProcessSupervisor](#),  
[269](#)
- WorkingDirectory
  - [rstudio::launcher\\_plugins::api::Job](#), [175](#)
- write
  - [rstudio::launcher\\_plugins::json::Value](#), [349](#)
- writeBytes
  - [rstudio::launcher\\_plugins::system::AsioStream](#), [62](#)
- writeError
  - [Logger.hpp](#), [378](#)
- writeFormatted
  - [rstudio::launcher\\_plugins::json::Value](#), [350](#)
- writeLog
  - [rstudio::launcher\\_plugins::logging::FileLogDestination](#),  
[110](#)
  - [rstudio::launcher\\_plugins::logging::ILogDestination](#),  
[154](#)
- writeToStdin
  - [rstudio::launcher\\_plugins::system::process::AbstractChildProcess](#),  
[15](#)
  - [rstudio::launcher\\_plugins::system::process::SyncChildProcess](#),  
[301](#)