

# 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>3</b>
2.1 Class List	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 <code>rstudio::launcher_plugins::AbstractMain</code> Class Reference	5
3.1.1 Detailed Description	5
3.1.2 Member Function Documentation	5
3.1.2.1 <code>run()</code>	5
3.2 <code>rstudio::launcher_plugins::AbstractPluginApi</code> Class Reference	6
3.2.1 Detailed Description	6
3.2.2 Member Function Documentation	6
3.2.2.1 <code>initialize()</code>	7
3.3 <code>rstudio::launcher_plugins::json::Array</code> Class Reference	7
3.3.1 Detailed Description	9
3.3.2 Constructor & Destructor Documentation	9
3.3.2.1 <code>Array()</code> [1/3]	9
3.3.2.2 <code>Array()</code> [2/3]	9
3.3.2.3 <code>Array()</code> [3/3]	9
3.3.3 Member Function Documentation	10
3.3.3.1 <code>begin()</code>	10
3.3.3.2 <code>end()</code>	10
3.3.3.3 <code>erase()</code> [1/2]	10
3.3.3.4 <code>erase()</code> [2/2]	11
3.3.3.5 <code>getBack()</code>	11
3.3.3.6 <code>getFront()</code>	11
3.3.3.7 <code>getSize()</code>	12
3.3.3.8 <code>getValueAt()</code>	12
3.3.3.9 <code>isEmpty()</code>	12
3.3.3.10 <code>operator=()</code> [1/2]	12
3.3.3.11 <code>operator=()</code> [2/2]	13
3.3.3.12 <code>operator[]()</code>	13
3.3.3.13 <code>push_back()</code>	14
3.3.3.14 <code>rbegin()</code>	14
3.3.3.15 <code>rend()</code>	14
3.3.3.16 <code>toSetString()</code>	14
3.3.3.17 <code>toStringPairList()</code>	15
3.3.3.18 <code>toVectorInt()</code>	15
3.3.3.19 <code>toVectorString()</code>	15
3.4 <code>rstudio::launcher_plugins::Error</code> Class Reference	16
3.4.1 Detailed Description	18

3.4.2 Constructor & Destructor Documentation	18
3.4.2.1 Error() [1/9]	18
3.4.2.2 Error() [2/9]	18
3.4.2.3 Error() [3/9]	18
3.4.2.4 Error() [4/9]	19
3.4.2.5 Error() [5/9]	19
3.4.2.6 Error() [6/9]	19
3.4.2.7 Error() [7/9]	20
3.4.2.8 Error() [8/9]	20
3.4.2.9 Error() [9/9]	21
3.4.3 Member Function Documentation	21
3.4.3.1 addOrUpdateProperty() [1/3]	21
3.4.3.2 addOrUpdateProperty() [2/3]	21
3.4.3.3 addOrUpdateProperty() [3/3]	22
3.4.3.4 addProperty() [1/3]	22
3.4.3.5 addProperty() [2/3]	22
3.4.3.6 addProperty() [3/3]	23
3.4.3.7 asString()	23
3.4.3.8 getCause()	23
3.4.3.9 getCode()	23
3.4.3.10 getLocation()	24
3.4.3.11 getMessage()	24
3.4.3.12 getName()	24
3.4.3.13 getProperties()	24
3.4.3.14 getProperty()	24
3.4.3.15 getSummary()	25
3.4.3.16 isExpected()	25
3.4.3.17 operator bool()	25
3.4.3.18 operator"!"()	26
3.4.3.19 operator"!="() [1/2]	26
3.4.3.20 operator"!="() [2/2]	27
3.4.3.21 operator==( ) [1/2]	27
3.4.3.22 operator==( ) [2/2]	27
3.5 rstudio::launcher_plugins::ErrorLocation Class Reference	29
3.5.1 Detailed Description	29
3.5.2 Constructor & Destructor Documentation	30
3.5.2.1 ErrorLocation() [1/3]	30
3.5.2.2 ErrorLocation() [2/3]	30
3.5.2.3 ErrorLocation() [3/3]	30
3.5.3 Member Function Documentation	31
3.5.3.1 asString()	31
3.5.3.2 getFile()	31

3.5.3.3	<a href="#">getFunction()</a>	31
3.5.3.4	<a href="#">getLine()</a>	31
3.5.3.5	<a href="#">hasLocation()</a>	32
3.5.3.6	<a href="#">operator=()</a>	32
3.5.3.7	<a href="#">operator==()</a>	32
3.6	<a href="#">rstudio::launcher_plugins::ErrorLock Class Reference</a>	33
3.6.1	<a href="#">Detailed Description</a>	33
3.7	<a href="#">rstudio::launcher_plugins::logging::FileLogDestination Class Reference</a>	33
3.7.1	<a href="#">Detailed Description</a>	34
3.7.2	<a href="#">Constructor &amp; Destructor Documentation</a>	34
3.7.2.1	<a href="#">FileLogDestination()</a>	34
3.7.3	<a href="#">Member Function Documentation</a>	34
3.7.3.1	<a href="#">getId()</a>	35
3.7.3.2	<a href="#">writeLog()</a>	35
3.8	<a href="#">rstudio::launcher_plugins::logging::FileLogOptions Class Reference</a>	35
3.8.1	<a href="#">Detailed Description</a>	36
3.8.2	<a href="#">Constructor &amp; Destructor Documentation</a>	36
3.8.2.1	<a href="#">FileLogOptions()</a> [1/2]	36
3.8.2.2	<a href="#">FileLogOptions()</a> [2/2]	36
3.8.3	<a href="#">Member Function Documentation</a>	37
3.8.3.1	<a href="#">doRotation()</a>	37
3.8.3.2	<a href="#">getDirectory()</a>	37
3.8.3.3	<a href="#">getFileMode()</a>	37
3.8.3.4	<a href="#">getMaxSizeMb()</a>	38
3.8.3.5	<a href="#">includePid()</a>	38
3.9	<a href="#">rstudio::launcher_plugins::system::FilePath Class Reference</a>	38
3.9.1	<a href="#">Detailed Description</a>	41
3.9.2	<a href="#">Member Typedef Documentation</a>	41
3.9.2.1	<a href="#">RecursiveIterationFunction</a>	41
3.9.3	<a href="#">Member Enumeration Documentation</a>	42
3.9.3.1	<a href="#">MoveType</a>	42
3.9.4	<a href="#">Constructor &amp; Destructor Documentation</a>	42
3.9.4.1	<a href="#">FilePath()</a>	42
3.9.5	<a href="#">Member Function Documentation</a>	42
3.9.5.1	<a href="#">completeChildPath()</a> [1/2]	43
3.9.5.2	<a href="#">completeChildPath()</a> [2/2]	43
3.9.5.3	<a href="#">completePath()</a>	43
3.9.5.4	<a href="#">copy()</a>	44
3.9.5.5	<a href="#">copyDirectoryRecursive()</a>	44
3.9.5.6	<a href="#">createAliasedPath()</a>	44
3.9.5.7	<a href="#">createDirectory()</a>	45
3.9.5.8	<a href="#">ensureDirectory()</a>	45

3.9.5.9 ensureFile()	45
3.9.5.10 exists() [1/2]	46
3.9.5.11 exists() [2/2]	46
3.9.5.12 getAbsolutePath()	46
3.9.5.13 getAbsolutePathNative()	46
3.9.5.14 getCanonicalPath()	47
3.9.5.15 getChildren()	47
3.9.5.16 getChildrenRecursive()	47
3.9.5.17 getExtension()	48
3.9.5.18 getExtensionLowerCase()	48
3.9.5.19 getFilename()	48
3.9.5.20 getLastWriteTime()	48
3.9.5.21 getLexicallyNormalPath()	49
3.9.5.22 getMimeType()	49
3.9.5.23 getParent()	49
3.9.5.24 getRelativePath()	49
3.9.5.25 getSize()	50
3.9.5.26 getSizeRecursive()	50
3.9.5.27 getStem()	50
3.9.5.28 hasExtension()	50
3.9.5.29 hasExtensionLowerCase()	51
3.9.5.30 hasTextMimeType()	51
3.9.5.31 isDirectory()	51
3.9.5.32 isEmpty()	52
3.9.5.33 isEqualCaseInsensitive()	52
3.9.5.34 isEquivalentTo()	52
3.9.5.35 isHidden()	53
3.9.5.36 isJunction()	53
3.9.5.37 isRegularFile()	53
3.9.5.38 isRootPath()	53
3.9.5.39 isSymlink()	54
3.9.5.40 isWithin()	54
3.9.5.41 makeCurrent()	54
3.9.5.42 makeCurrentPath()	55
3.9.5.43 move()	55
3.9.5.44 moveIndirect()	55
3.9.5.45 openForRead()	56
3.9.5.46 openForWrite()	56
3.9.5.47 operator!=(())	57
3.9.5.48 operator<()	57
3.9.5.49 operator==(())	57
3.9.5.50 remove()	58

3.9.5.51 removeIfExists()	58
3.9.5.52 resetDirectory()	58
3.9.5.53 resolveAliasedPath()	58
3.9.5.54 resolveSymlink()	59
3.9.5.55 safeCurrentPath()	59
3.9.5.56 setLastWriteTime()	59
3.9.5.57 tempFilePath() [1/2]	60
3.9.5.58 tempFilePath() [2/2]	60
3.9.5.59 uniqueFilePath() [1/2]	60
3.9.5.60 uniqueFilePath() [2/2]	61
3.10 rstudio::launcher_plugins::logging::ILogDestination Class Reference	61
3.10.1 Detailed Description	62
3.10.2 Constructor & Destructor Documentation	62
3.10.2.1 ILogDestination()	62
3.10.3 Member Function Documentation	62
3.10.3.1 getId()	63
3.10.3.2 getLogLevel()	63
3.10.3.3 writeLog()	63
3.11 rstudio::launcher_plugins::options::Options::Init Class Reference	63
3.11.1 Detailed Description	64
3.11.2 Constructor & Destructor Documentation	64
3.11.2.1 Init()	64
3.11.3 Member Function Documentation	64
3.11.3.1 operator>() [1/2]	64
3.11.3.2 operator>() [2/2]	65
3.12 rstudio::launcher_plugins::json::detail::is_json_type< T > Struct Template Reference	66
3.13 rstudio::launcher_plugins::json::Object::Iterator Class Reference	66
3.13.1 Detailed Description	67
3.13.2 Constructor & Destructor Documentation	67
3.13.2.1 Iterator() [1/2]	67
3.13.2.2 Iterator() [2/2]	67
3.13.3 Member Function Documentation	67
3.13.3.1 operator"!=""()	67
3.13.3.2 operator*()	68
3.13.3.3 operator++() [1/2]	68
3.13.3.4 operator++() [2/2]	68
3.13.3.5 operator--() [1/2]	69
3.13.3.6 operator--() [2/2]	69
3.13.3.7 operator=()	69
3.13.3.8 operator==()	69
3.14 rstudio::launcher_plugins::json::Array::Iterator Class Reference	70
3.14.1 Detailed Description	71

3.14.2 Constructor & Destructor Documentation	71
3.14.2.1 Iterator() [1/2]	71
3.14.2.2 Iterator() [2/2]	71
3.14.3 Member Function Documentation	71
3.14.3.1 operator!=(())	71
3.14.3.2 operator*()	72
3.14.3.3 operator++() [1/2]	72
3.14.3.4 operator++() [2/2]	72
3.14.3.5 operator--() [1/2]	72
3.14.3.6 operator--() [2/2]	73
3.14.3.7 operator=()	73
3.14.3.8 operator==(())	73
3.15 rstudio::launcher_plugins::json::Object::Member Class Reference	73
3.15.1 Detailed Description	74
3.15.2 Constructor & Destructor Documentation	74
3.15.2.1 Member()	74
3.15.3 Member Function Documentation	74
3.15.3.1 getName()	74
3.15.3.2 getValue()	75
3.16 rstudio::launcher_plugins::json::Object Class Reference	75
3.16.1 Detailed Description	77
3.16.2 Constructor & Destructor Documentation	77
3.16.2.1 Object() [1/3]	77
3.16.2.2 Object() [2/3]	78
3.16.2.3 Object() [3/3]	78
3.16.3 Member Function Documentation	78
3.16.3.1 begin()	78
3.16.3.2 createMember()	78
3.16.3.3 end()	79
3.16.3.4 erase() [1/3]	79
3.16.3.5 erase() [2/3]	79
3.16.3.6 erase() [3/3]	80
3.16.3.7 find() [1/2]	80
3.16.3.8 find() [2/2]	80
3.16.3.9 getSchemaDefaults()	81
3.16.3.10 getSize()	81
3.16.3.11 hasMember() [1/2]	81
3.16.3.12 hasMember() [2/2]	82
3.16.3.13 insert() [1/2]	82
3.16.3.14 insert() [2/2]	82
3.16.3.15 isEmpty()	83
3.16.3.16 mergeObjects()	83



3.16.3.17 operator=() [1/2]	83
3.16.3.18 operator=() [2/2]	84
3.16.3.19 operator[]() [1/2]	84
3.16.3.20 operator[]() [2/2]	84
3.16.3.21 rbegin()	85
3.16.3.22 rend()	85
3.16.3.23 toStringMap()	85
3.16.3.24 toStringPairList()	86
3.17 rstudio::launcher_plugins::options::Options Class Reference	86
3.17.1 Detailed Description	87
3.17.2 Member Function Documentation	87
3.17.2.1 getHeartbeatIntervalSeconds()	87
3.17.2.2 getInstance()	88
3.17.2.3 getJobExpiryHours()	88
3.17.2.4 getLogLevel()	88
3.17.2.5 getScratchPath()	88
3.17.2.6 getServerUser()	88
3.17.2.7 getThreadPoolSize()	89
3.17.2.8 readOptions()	89
3.17.2.9 registerOptions()	89
3.18 rstudio::launcher_plugins::system::PathScopeImplDeleter Struct Reference	90
3.18.1 Detailed Description	90
3.19 rstudio::launcher_plugins::quickstart::QuickStartPluginApi Class Reference	90
3.19.1 Detailed Description	91
3.19.2 Member Function Documentation	91
3.19.2.1 initialize()	91
3.20 rstudio::launcher_plugins::system::RemoveOnExitScope Class Reference	91
3.20.1 Detailed Description	92
3.20.2 Constructor & Destructor Documentation	92
3.20.2.1 RemoveOnExitScope()	92
3.21 rstudio::launcher_plugins::system::RestoreCurrentPathScope Class Reference	92
3.21.1 Detailed Description	93
3.21.2 Constructor & Destructor Documentation	93
3.21.2.1 RestoreCurrentPathScope()	93
3.22 rstudio::launcher_plugins::singularity::SingularityOptions Class Reference	93
3.22.1 Detailed Description	94
3.22.2 Member Function Documentation	94
3.22.2.1 getInstance()	94
3.22.2.2 getRContainer()	94
3.22.2.3 getRSessionContainer()	95
3.22.2.4 initialize()	95
3.23 rstudio::launcher_plugins::singularity::SingularityPluginApi Class Reference	95

3.23.1 Detailed Description	96
3.23.2 Member Function Documentation	96
3.23.2.1 initialize()	96
3.24 rstudio::launcher_plugins::Success Class Reference	96
3.24.1 Detailed Description	97
3.25 rstudio::launcher_plugins::system::User Class Reference	97
3.25.1 Detailed Description	98
3.25.2 Constructor & Destructor Documentation	98
3.25.2.1 User() [1/2]	98
3.25.2.2 User() [2/2]	98
3.25.3 Member Function Documentation	98
3.25.3.1 exists()	98
3.25.3.2 getCurrentUser()	99
3.25.3.3 getGroupId()	100
3.25.3.4 getHomePath()	100
3.25.3.5 getUserFromIdentifier() [1/2]	100
3.25.3.6 getUserFromIdentifier() [2/2]	101
3.25.3.7 getUserHomePath()	101
3.25.3.8 getUserId()	101
3.25.3.9 getUsername()	102
3.25.3.10 isAllUsers()	102
3.25.3.11 isEmpty()	102
3.25.3.12 operator=()	102
3.26 rstudio::launcher_plugins::options::Value< T > Class Template Reference	103
3.26.1 Detailed Description	103
3.26.2 Constructor & Destructor Documentation	104
3.26.2.1 Value()	104
3.26.3 Member Function Documentation	105
3.26.3.1 setDefaultValue()	105
3.27 rstudio::launcher_plugins::json::Value Class Reference	105
3.27.1 Detailed Description	108
3.27.2 Constructor & Destructor Documentation	108
3.27.2.1 Value() [1/12]	108
3.27.2.2 Value() [2/12]	109
3.27.2.3 Value() [3/12]	109
3.27.2.4 Value() [4/12]	109
3.27.2.5 Value() [5/12]	109
3.27.2.6 Value() [6/12]	110
3.27.2.7 Value() [7/12]	110
3.27.2.8 Value() [8/12]	110
3.27.2.9 Value() [9/12]	111
3.27.2.10 Value() [10/12]	111

3.27.2.11 Value() [11/12]	111
3.27.2.12 Value() [12/12]	111
3.27.3 Member Function Documentation	112
3.27.3.1 clone()	112
3.27.3.2 coerce()	112
3.27.3.3 getArray()	112
3.27.3.4 getBool()	113
3.27.3.5 getDouble()	113
3.27.3.6 getFloat()	113
3.27.3.7 getInt()	113
3.27.3.8 getInt64()	114
3.27.3.9 getObject()	114
3.27.3.10 getString()	114
3.27.3.11 getType()	114
3.27.3.12 getUInt()	115
3.27.3.13 getUInt64()	115
3.27.3.14 getValue()	115
3.27.3.15 isArray()	115
3.27.3.16 isBool()	116
3.27.3.17 isDouble()	116
3.27.3.18 isFloat()	116
3.27.3.19 isInt()	116
3.27.3.20 isInt64()	117
3.27.3.21 isNull()	117
3.27.3.22 isObject()	117
3.27.3.23 isString()	117
3.27.3.24 isUInt()	118
3.27.3.25 isUInt64()	118
3.27.3.26 operator=() [1/11]	118
3.27.3.27 operator=() [2/11]	118
3.27.3.28 operator=() [3/11]	119
3.27.3.29 operator=() [4/11]	119
3.27.3.30 operator=() [5/11]	119
3.27.3.31 operator=() [6/11]	121
3.27.3.32 operator=() [7/11]	121
3.27.3.33 operator=() [8/11]	121
3.27.3.34 operator=() [9/11]	123
3.27.3.35 operator=() [10/11]	123
3.27.3.36 operator=() [11/11]	123
3.27.3.37 operator==()	125
3.27.3.38 parse() [1/2]	125
3.27.3.39 parse() [2/2]	125

3.27.3.40 <a href="#">parseAndValidate()</a> . . . . .	127
3.27.3.41 <a href="#">validate()</a> . . . . .	127
3.27.3.42 <a href="#">write()</a> [1/2] . . . . .	128
3.27.3.43 <a href="#">write()</a> [2/2] . . . . .	128
3.27.3.44 <a href="#">writeFormatted()</a> [1/2] . . . . .	128
3.27.3.45 <a href="#">writeFormatted()</a> [2/2] . . . . .	128

<b>Index</b>	<b>131</b>
--------------	------------

# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

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

rstudio::launcher_plugins::ErrorLocation . . . . .	29
rstudio::launcher_plugins::ErrorLock . . . . .	33
rstudio::launcher_plugins::Error . . . . .	16
rstudio::launcher_plugins::Success . . . . .	96
rstudio::launcher_plugins::logging::FileLogOptions . . . . .	35
rstudio::launcher_plugins::system::FilePath . . . . .	38
rstudio::launcher_plugins::options::Options::Init . . . . .	63
is_base_of	
rstudio::launcher_plugins::json::detail::is_json_type< T > . . . . .	66
iterator	
rstudio::launcher_plugins::json::Array::Iterator . . . . .	70
rstudio::launcher_plugins::json::Object::Iterator . . . . .	66
rstudio::launcher_plugins::json::Object::Member . . . . .	73
noncopyable	
rstudio::launcher_plugins::AbstractMain . . . . .	5
rstudio::launcher_plugins::AbstractPluginApi . . . . .	6
rstudio::launcher_plugins::quickstart::QuickStartPluginApi . . . . .	90
rstudio::launcher_plugins::singularity::SingularityPluginApi . . . . .	95
rstudio::launcher_plugins::logging::ILogDestination . . . . .	61
rstudio::launcher_plugins::logging::FileLogDestination . . . . .	33
rstudio::launcher_plugins::options::Options . . . . .	86
rstudio::launcher_plugins::singularity::SingularityOptions . . . . .	93
rstudio::launcher_plugins::system::RemoveOnExitScope . . . . .	91
rstudio::launcher_plugins::system::RestoreCurrentPathScope . . . . .	92
rstudio::launcher_plugins::system::PathScopeImplDeleter . . . . .	90
rstudio::launcher_plugins::system::User . . . . .	97
rstudio::launcher_plugins::options::Value< T > . . . . .	103
rstudio::launcher_plugins::json::Value . . . . .	105
rstudio::launcher_plugins::json::Array . . . . .	7
rstudio::launcher_plugins::json::Object . . . . .	75



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">rstudio::launcher_plugins::AbstractMain</a>	5
Base class for the Plugin Main class, which runs the plugin . . . . .	
<a href="#">rstudio::launcher_plugins::AbstractPluginApi</a>	6
Base class for the Launcher Plugin API . . . . .	
<a href="#">rstudio::launcher_plugins::json::Array</a>	7
Class which represents a JSON array . . . . .	
<a href="#">rstudio::launcher_plugins::Error</a>	16
Class which represents an error . . . . .	
<a href="#">rstudio::launcher_plugins::ErrorLocation</a>	29
Class which represents the location of an error . . . . .	
<a href="#">rstudio::launcher_plugins::ErrorLock</a>	33
A class which can be derived from in order to prevent child classes from being derived from further . . . . .	
<a href="#">rstudio::launcher_plugins::logging::FileLogDestination</a>	33
Class which allows sending log messages to a file . . . . .	
<a href="#">rstudio::launcher_plugins::logging::FileLogOptions</a>	35
Class which represents the options for a file logger . . . . .	
<a href="#">rstudio::launcher_plugins::system::FilePath</a>	38
Class which represents a path on the system. May be any type of file (e.g. directory, symlink, regular file, etc.) . . . . .	
<a href="#">rstudio::launcher_plugins::logging::ILogDestination</a>	61
Interface which allows a logger to write a log message to a destination . . . . .	
<a href="#">rstudio::launcher_plugins::options::Options::Init</a>	63
Class for initializing <a href="#">Options</a> . . . . .	
<a href="#">rstudio::launcher_plugins::json::detail::is_json_type&lt; T &gt;</a>	66
<a href="#">rstudio::launcher_plugins::json::Object::Iterator</a>	66
Class which allows iterating over the members of a JSON object . . . . .	
<a href="#">rstudio::launcher_plugins::json::Array::Iterator</a>	70
Class which allows iterating over the elements of a JSON array . . . . .	
<a href="#">rstudio::launcher_plugins::json::Object::Member</a>	73
Class which represents a single member of a JSON object . . . . .	
<a href="#">rstudio::launcher_plugins::json::Object</a>	75
Class which represents a specific type of JSON <a href="#">Value</a> : a JSON object . . . . .	
<a href="#">rstudio::launcher_plugins::options::Options</a>	86
<a href="#">Options</a> for the plugin . . . . .	

<a href="#">rstudio::launcher_plugins::system::PathScopeImplDeleter</a>	
Struct which implements the deleter for PathScopeImpl	90
<a href="#">rstudio::launcher_plugins::quickstart::QuickStartPluginApi</a>	
Launcher Plugin API for the QuickStart Plugin	90
<a href="#">rstudio::launcher_plugins::system::RemoveOnExitScope</a>	
RAII class for restoring the current working directory	91
<a href="#">rstudio::launcher_plugins::system::RestoreCurrentPathScope</a>	
RAII class for restoring the current working directory	92
<a href="#">rstudio::launcher_plugins::singularity::SingularityOptions</a>	
Class which stores options specific to the Singularity Container system	93
<a href="#">rstudio::launcher_plugins::singularity::SingularityPluginApi</a>	
Launcher Plugin API for the Singularity Plugin	95
<a href="#">rstudio::launcher_plugins::Success</a>	
Class which represents a successful operation (i.e. no error)	96
<a href="#">rstudio::launcher_plugins::system::User</a>	
Class which represents a system user	97
<a href="#">rstudio::launcher_plugins::options::Value&lt; T &gt;</a>	
Concrete class which represents an option <a href="#">Value</a>	103
<a href="#">rstudio::launcher_plugins::json::Value</a>	
Class which represents a json value	105



## Chapter 3

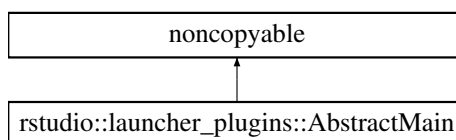
# Class Documentation

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

Base class for the Plugin Main class, which runs the plugin.

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

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



#### Public Member Functions

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

#### 3.1.1 Detailed Description

Base class for the Plugin Main class, which runs the plugin.

#### 3.1.2 Member Function Documentation

##### 3.1.2.1 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 <i>in_argList</i> .
<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:

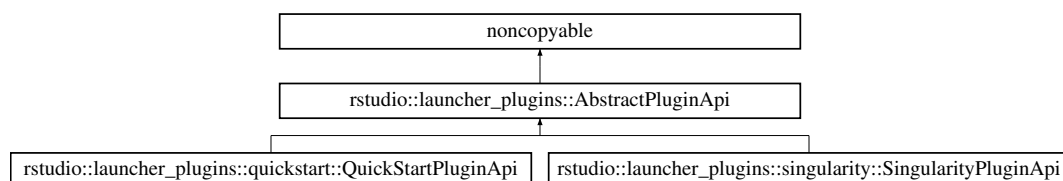
- /home/maria/GitRepos/rstudio-launcher-plugin-sdk/sdk/include/AbstractMain.hpp

## 3.2 rstudio::launcher\_plugins::AbstractPluginApi Class Reference

Base class for the Launcher Plugin API.

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

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



### Public Member Functions

- virtual `~AbstractPluginApi()`=default

*Virtual destructor.*

- virtual `Error initialize()`=0

*This method should initialize any components needed to communicate with the job scheduling tool, including custom options (TODO: other examples).*

### 3.2.1 Detailed Description

Base class for the Launcher Plugin API.

### 3.2.2 Member Function Documentation

### 3.2.2.1 initialize()

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

This method should initialize any components needed to communicate with the job scheduling tool, including custom options (TODO: other examples).

#### Returns

[Success](#) if all components needed by this Plugin were successfully initialized; [Error](#) otherwise.

Implemented in [rstudio::launcher\\_plugins::singularity::SingularityPluginApi](#), and [rstudio::launcher\\_plugins::quickstart::QuickStartPluginApi](#).

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

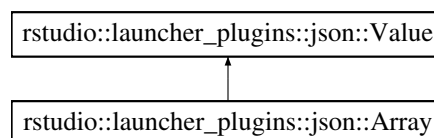
- `/home/maria/GitRepos/rstudio-launcher-plugin-sdk/sdk/include/AbstractPluginApi.hpp`

## 3.3 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 >` [ReverseIterator](#)  
*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)  
*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.*
- void [push\\_back](#) (const [Value](#) &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

### 3.3.1 Detailed Description

Class which represents a JSON array.

### 3.3.2 Constructor & Destructor Documentation

#### 3.3.2.1 `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.
--------------------	--

#### 3.3.2.2 `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.
-----------------	------------------------------

#### 3.3.2.3 `Array()` [3/3]

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

Move constructor.

## Parameters

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

### 3.3.3 Member Function Documentation

#### 3.3.3.1 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.

#### 3.3.3.2 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.

#### 3.3.3.3 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.

**3.3.3.4 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.

**3.3.3.5 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.

**3.3.3.6 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.

### 3.3.3.7 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.

### 3.3.3.8 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.

### 3.3.3.9 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.

### 3.3.3.10 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](#).

**3.3.3.11 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.

**3.3.3.12 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.

### 3.3.3.13 push\_back()

```
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

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

### 3.3.3.14 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.

### 3.3.3.15 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 `ReverseIterator` to determine when reverse iteration has ended.

#### Returns

An iterator before the first member of this array.

### 3.3.3.16 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**

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

**Returns**

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

**3.3.3.17 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.

**3.3.3.18 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.

**3.3.3.19 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

<code>out_set</code>	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:

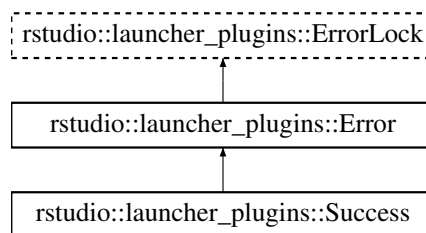
- /home/maria/GitRepos/rstudio-launcher-plugin-sdk/sdk/include/json/Json.hpp

### 3.4 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(const boost::system::error_code &in_ec, const ErrorLocation &in_location)`  
*Constructor.*
- `Error(const boost::system::error_code &in_ec, const Error &in_cause, const ErrorLocation &in_location)`  
*Constructor.*
- `Error(const boost::system::error_code &in_ec, std::string in_message, const ErrorLocation &in_location)`  
*Constructor.*
- `Error(const boost::system::error_code &in_ec, std::string in_message, const Error &in_cause, const ErrorLocation &in_location)`  
*Constructor.*
- `Error(int in_code, std::string in_name, const ErrorLocation &in_location)`  
*Constructor.*
- `Error(int in_code, std::string in_name, const Error &in_cause, const ErrorLocation &in_location)`  
*Constructor.*
- `Error(int in_code, std::string in_name, std::string in_message, const ErrorLocation &in_location)`

Constructor.

- `Error` (int in\_code, std::string in\_name, 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.

- `bool operator!` () const

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

- `bool operator==` (const `Error` &in\_other) const

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

- `bool operator==` (const boost::system::error\_code &in\_ec) const

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

- `bool operator!=` (const `Error` &in\_other) const

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

- `bool operator!=` (const boost::system::error\_code &in\_ec) 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.

### 3.4.1 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.

### 3.4.2 Constructor & Destructor Documentation

#### 3.4.2.1 Error() [1/9]

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

Copy constructor.

Parameters

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

#### 3.4.2.2 Error() [2/9]

```
rstudio::launcher_plugins::Error::Error (
    const boost::system::error_code & in_ec,
    const ErrorLocation & in_location )
```

Constructor.

Parameters

<i>in_ec</i>	The boost error code to convert from.
<i>in_location</i>	The location of the error.

#### 3.4.2.3 Error() [3/9]

```
rstudio::launcher_plugins::Error::Error (
    const boost::system::error_code & in_ec,
    const Error & in_cause,
    const ErrorLocation & in_location )
```

Constructor.

## Parameters

<i>in_ec</i>	The boost error code to convert from.
<i>in_cause</i>	The error which caused this error.
<i>in_location</i>	The location of the error.

**3.4.2.4 Error()** [4/9]

```
rstudio::launcher_plugins::Error::Error (
    const boost::system::error_code & in_ec,
    std::string in_message,
    const ErrorLocation & in_location )
```

Constructor.

## Parameters

<i>in_ec</i>	The boost error code to convert from.
<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.

**3.4.2.5 Error()** [5/9]

```
rstudio::launcher_plugins::Error::Error (
    const boost::system::error_code & in_ec,
    std::string in_message,
    const Error & in_cause,
    const ErrorLocation & in_location )
```

Constructor.

## Parameters

<i>in_ec</i>	The boost error code to convert from.
<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.

**3.4.2.6 Error()** [6/9]

```
rstudio::launcher_plugins::Error::Error (
    int in_code,
```

```
std::string in_name,
const ErrorLocation & in_location )
```

Constructor.

#### Parameters

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

### 3.4.2.7 Error() [7/9]

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

Constructor.

#### Parameters

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

### 3.4.2.8 Error() [8/9]

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

Constructor.

#### Parameters

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



### 3.4.2.9 Error() [9/9]

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

Constructor.

#### Parameters

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

## 3.4.3 Member Function Documentation

### 3.4.3.1 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.

### 3.4.3.2 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.

**3.4.3.3 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.

**3.4.3.4 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.

**3.4.3.5 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.

### 3.4.3.6 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.

### 3.4.3.7 asString()

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

Formats the error as a string.

#### Returns

The error formatted as a string.

### 3.4.3.8 getCause()

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

Gets the error which caused this error.

#### Returns

The error which caused this error.

### 3.4.3.9 getCode()

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

Gets the error code.

#### Returns

The error code.

#### 3.4.3.10 getLocation()

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

Gets the location where the error occurred.

##### Returns

The location where the error occurred.

#### 3.4.3.11 getMessage()

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

Gets the error message.

##### Returns

The error message.

#### 3.4.3.12 getName()

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

Gets the name of the error.

##### Returns

The name of the error.

#### 3.4.3.13 getProperties()

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

Gets the custom properties of the error.

##### Returns

The custom properties of this error.

#### 3.4.3.14 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.

**3.4.3.15 getSummary()**

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

Gets the cause of the error.

**Returns**

The cause of the error.

**3.4.3.16 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.

**3.4.3.17 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.

#### 3.4.3.18 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.

#### 3.4.3.19 operator"!="( [1/2]

```
bool rstudio::launcher_plugins::Error::operator!= (
    const boost::system::error_code & in_ec ) const
```

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

**Parameters**

<i>in_ec</i>	The boost error code to compare with this error.
--------------	--

**Returns**

True if `!(this == in_ec)` would return true; false otherwise.

**3.4.3.20 operator!=( ) [2/2]**

```
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.

**3.4.3.21 operator==( ) [1/2]**

```
bool rstudio::launcher_plugins::Error::operator== (
    const boost::system::error_code & in_ec ) const
```

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

**Parameters**

<i>in_ec</i>	The boost error code to compare with this error.
--------------	--

**Returns**

True if `in_ec` has the same error code and category name as this error; false otherwise.

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

```
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

<code>in_other</code>	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:

- `/home/maria/GitRepos/rstudio-launcher-plugin-sdk/sdk/include/Error.hpp`

## 3.5 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.*

### 3.5.1 Detailed Description

Class which represents the location of an error.

## 3.5.2 Constructor & Destructor Documentation

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

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

Copy constructor.

#### Parameters

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

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

```
rstudio::launcher_plugins::ErrorLocation::ErrorLocation (
    ErrorLocation && in_other ) [noexcept]
```

Move constructor.

#### Parameters

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

### 3.5.2.3 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.

### 3.5.3 Member Function Documentation

#### 3.5.3.1 asString()

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

Formats the error location as a string.

##### Returns

The error location formatted as a string.

#### 3.5.3.2 getFile()

```
const std::string& rstudio::launcher_plugins::ErrorLocation::getFile ( ) const
```

Gets the file where the error occurred.

##### Returns

The file where the error occurred.

#### 3.5.3.3 getFunction()

```
const std::string& rstudio::launcher_plugins::ErrorLocation::getFunction ( ) const
```

Gets the function where the error occurred.

##### Returns

The function where the error occurred.

#### 3.5.3.4 getLine()

```
long rstudio::launcher_plugins::ErrorLocation::getLine ( ) const
```

Gets the line where the error occurred.

##### Returns

The line where the error occurred.

### 3.5.3.5 hasLocation()

```
bool rstudio::launcher_plugins::ErrorLocation::hasLocation ( ) const
```

Checks whether the location is set.

#### Returns

True if a location has been set; false otherwise.

### 3.5.3.6 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.

### 3.5.3.7 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:

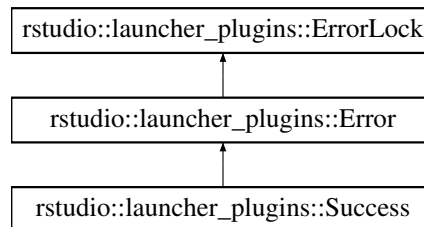
- /home/maria/GitRepos/rstudio-launcher-plugin-sdk/sdk/include/Error.hpp

## 3.6 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**

### 3.6.1 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:

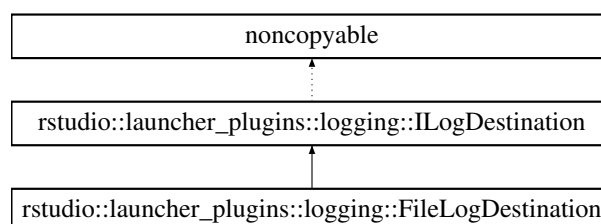
- /home/maria/GitRepos/rstudio-launcher-plugin-sdk/sdk/include/Error.hpp

## 3.7 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 [writeLog](#) (LogLevel in\_logLevel, const std::string &in\_message) override  
*Writes a message to the log file.*

## Additional Inherited Members

### 3.7.1 Detailed Description

Class which allows sending log messages to a file.

### 3.7.2 Constructor & Destructor Documentation

#### 3.7.2.1 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.

### 3.7.3 Member Function Documentation

### 3.7.3.1 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](#).

### 3.7.3.2 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:

- /home/maria/GitRepos/rstudio-launcher-plugin-sdk/sdk/include/logging/FileLogDestination.hpp

## 3.8 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\_do↵  
Rotation, 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.*

### 3.8.1 Detailed Description

Class which represents the options for a file logger.

### 3.8.2 Constructor & Destructor Documentation

#### 3.8.2.1 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.
---------------------	---

#### 3.8.2.2 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.
---------------------	---



## Parameters

<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.

### 3.8.3 Member Function Documentation

#### 3.8.3.1 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.

#### 3.8.3.2 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.

#### 3.8.3.3 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.

#### 3.8.3.4 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.

#### 3.8.3.5 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:

- /home/maria/GitRepos/rstudio-launcher-plugin-sdk/sdk/include/logging/FileLogDestination.hpp

## 3.9 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.*
- [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) const  
*Copies this file path to the specified location.*
- [Error copyDirectoryRecursive](#) (const [FilePath](#) &in\_targetPath) 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 '.'.*
- 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 getMimeTypeType (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.*
- `bool isJunction () const`  
*Checks whether this file path is a Windows junction.*
- `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 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) const`  
*Moves the current directory to the specified directory.*
- `Error moveIndirect (const FilePath &in_targetPath) 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.*

## 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.*

### 3.9.1 Detailed Description

Class which represents a path on the system. May be any type of file (e.g. directory, symlink, regular file, etc.)

### 3.9.2 Member Typedef Documentation

#### 3.9.2.1 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.
<i>FilePath</i>	The current <a href="#">FilePath</a> object in the recursive iteration.

## Returns

True if the computation can continue; false otherwise.

### 3.9.3 Member Enumeration Documentation

#### 3.9.3.1 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

### 3.9.4 Constructor & Destructor Documentation

#### 3.9.4.1 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.
------------------------	--

### 3.9.5 Member Function Documentation

**3.9.5.1 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.

**3.9.5.2 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.

**3.9.5.3 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.

#### 3.9.5.4 copy()

```
Error rstudio::launcher_plugins::system::FilePath::copy (  
    const FilePath & in_targetPath ) const
```

Copies this file path to the specified location.

#### Parameters

<i>in_targetPath</i>	The location to copy this file path to.
----------------------	---

#### Returns

**Success** if the copy could be completed; **Error** otherwise.

#### 3.9.5.5 copyDirectoryRecursive()

```
Error rstudio::launcher_plugins::system::FilePath::copyDirectoryRecursive (  
    const FilePath & in_targetPath ) 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.
----------------------	--

#### Returns

**Success** if the copy could be completed; **Error** otherwise.

#### 3.9.5.6 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.

**3.9.5.7 createDirectory()**

```
Error rstudio::launcher_plugins::system::FilePath::createDirectory (
    const std::string & in_filePath ) const
```

Creates the specified directory.

## Parameters

<i>in_filePath</i>	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.

**3.9.5.8 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.

**3.9.5.9 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.

### 3.9.5.10 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.

### 3.9.5.11 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.

### 3.9.5.12 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.

### 3.9.5.13 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.

#### 3.9.5.14 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.

#### 3.9.5.15 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

<i>out_filePaths</i>	The children of this directory.
----------------------	---------------------------------

##### Returns

**Success** if the children could be retrieved; **Error** otherwise (e.g. if this path does not exist).

#### 3.9.5.16 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

<i>in_iterationFunction</i>	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).

#### 3.9.5.17 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.

#### 3.9.5.18 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.

#### 3.9.5.19 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.

#### 3.9.5.20 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.

### 3.9.5.21 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.

### 3.9.5.22 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.

### 3.9.5.23 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.

### 3.9.5.24 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.

**3.9.5.25 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.

**3.9.5.26 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.

**3.9.5.27 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.

**3.9.5.28 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.

**3.9.5.29 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.

**3.9.5.30 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.

**3.9.5.31 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.

### 3.9.5.32 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.

### 3.9.5.33 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.

### 3.9.5.34 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.



#### 3.9.5.35 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.

#### 3.9.5.36 isJunction()

```
bool rstudio::launcher_plugins::system::FilePath::isJunction ( ) const
```

Checks whether this file path is a Windows junction.

##### Returns

True if this file path is a Windows junction; false otherwise.

#### 3.9.5.37 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.

#### 3.9.5.38 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.

**3.9.5.39 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.

**3.9.5.40 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.

**3.9.5.41 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.

**3.9.5.42 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: false.
----------------------	---

**Returns**

**Success** if the working directory was changed; **Error** otherwise.

**3.9.5.43 move()**

```
Error rstudio::launcher_plugins::system::FilePath::move (
    const FilePath & in_targetPath,
    MoveType in_type = MoveCrossDevice ) 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 <code>MoveType</code> for more details. Default: <code>MoveCrossDevice</code> .

**Returns**

**Success** if this directory could be moved to the target; **Error** otherwise.

**3.9.5.44 moveIndirect()**

```
Error rstudio::launcher_plugins::system::FilePath::moveIndirect (
    const FilePath & in_targetPath ) 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.
----------------------	---

## Returns

[Success](#) if this directory could be moved to the target; [Error](#) otherwise.

**3.9.5.45 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.)

**3.9.5.46 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.)

### 3.9.5.47 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.

### 3.9.5.48 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.

### 3.9.5.49 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.

**3.9.5.50 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.

**3.9.5.51 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.

**3.9.5.52 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.

**3.9.5.53 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.

**3.9.5.54 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.

**3.9.5.55 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.

**3.9.5.56 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.
----------------	--

**3.9.5.57 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.

**3.9.5.58 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.

**3.9.5.59 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.

## 3.9.5.60 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:

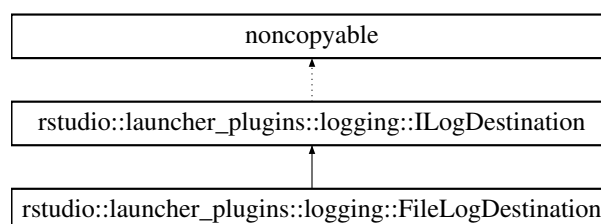
- /home/maria/GitRepos/rstudio-launcher-plugin-sdk/sdk/include/system/FilePath.hpp

## 3.10 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.*
- 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.*

### 3.10.1 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.

### 3.10.2 Constructor & Destructor Documentation

#### 3.10.2.1 ILogDestination()

```
rstudio::launcher_plugins::logging::ILogDestination::ILogDestination (
    LogLevel in_logLevel ) [inline], [explicit]
```

Constructor.

#### Parameters

<i>in_logLevel</i>	The most detailed level of log to be written to this log destination.
--------------------	---

### 3.10.3 Member Function Documentation

### 3.10.3.1 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](#).

### 3.10.3.2 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.

### 3.10.3.3 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:

- /home/maria/GitRepos/rstudio-launcher-plugin-sdk/sdk/include/logging/ILogDestination.hpp

## 3.11 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.*

### 3.11.1 Detailed Description

Class for initializing [Options](#).

### 3.11.2 Constructor & Destructor Documentation

#### 3.11.2.1 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.
-----------------	---

### 3.11.3 Member Function Documentation

#### 3.11.3.1 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.

## 3.11.3.2 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

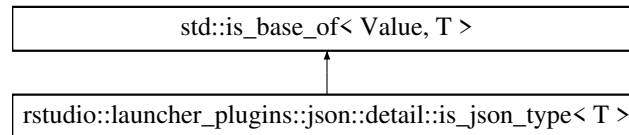
A reference to this [Init](#) object.

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

- /home/maria/GitRepos/rstudio-launcher-plugin-sdk/sdk/include/options/Options.hpp

### 3.12 rstudio::launcher\_plugins::json::detail::is\_json\_type< T > Struct Template Reference

Inheritance diagram for rstudio::launcher\_plugins::json::detail::is\_json\_type< T >:



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

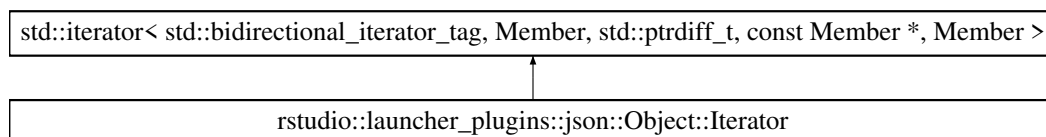
- /home/maria/GitRepos/rstudio-launcher-plugin-sdk/sdk/include/json/Json.hpp

### 3.13 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**

### 3.13.1 Detailed Description

Class which allows iterating over the members of a JSON object.

### 3.13.2 Constructor & Destructor Documentation

#### 3.13.2.1 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.

#### 3.13.2.2 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.
-----------------	-----------------------

### 3.13.3 Member Function Documentation

#### 3.13.3.1 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.

### 3.13.3.2 `operator*()`

```
reference rstudio::launcher_plugins::json::Object::Iterator::operator* ( ) const
```

Dereference operator.

#### Returns

A reference to the value this iterator is currently pointing at.

### 3.13.3.3 `operator++()` [1/2]

```
Iterator& rstudio::launcher_plugins::json::Object::Iterator::operator++ ( )
```

Pre-increment operator.

#### Returns

A reference to this operator, incremented by one position.

### 3.13.3.4 `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.



### 3.13.3.5 operator--() [1/2]

```
Iterator& rstudio::launcher_plugins::json::Object::Iterator::operator-- ( )
```

Pre-decrement operator.

#### Returns

A reference to this operator, decremented by one position.

### 3.13.3.6 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.

### 3.13.3.7 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.

### 3.13.3.8 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:

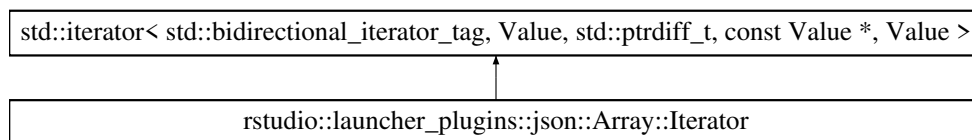
- `/home/maria/GitRepos/rstudio-launcher-plugin-sdk/sdk/include/json/Json.hpp`

### 3.14 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`

### 3.14.1 Detailed Description

Class which allows iterating over the elements of a JSON array.

### 3.14.2 Constructor & Destructor Documentation

#### 3.14.2.1 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.

#### 3.14.2.2 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.
-----------------	-----------------------

### 3.14.3 Member Function Documentation

#### 3.14.3.1 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.

**3.14.3.2 operator\*()**

```
reference rstudio::launcher_plugins::json::Array::Iterator::operator* ( ) const
```

Dereference operator.

**Returns**

A reference to the value this iterator is currently pointing at.

**3.14.3.3 operator++() [1/2]**

```
Iterator& rstudio::launcher_plugins::json::Array::Iterator::operator++ ( )
```

Pre-increment operator.

**Returns**

A reference to this operator, incremented by one position.

**3.14.3.4 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.

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

```
Iterator& rstudio::launcher_plugins::json::Array::Iterator::operator-- ( )
```

Pre-decrement operator.

**Returns**

A reference to this operator, decremented by one position.

### 3.14.3.6 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.

### 3.14.3.7 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.

### 3.14.3.8 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:

- /home/maria/GitRepos/rstudio-launcher-plugin-sdk/sdk/include/json/Json.hpp

## 3.15 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](#)

### 3.15.1 Detailed Description

Class which represents a single member of a JSON object.

### 3.15.2 Constructor & Destructor Documentation

#### 3.15.2.1 [Member](#)()

```
rstudio::launcher_plugins::json::Object::Member::Member (
    const std::shared_ptr< Impl > & in_impl )
```

Creates a [Member](#) object via its private implementation.

#### Parameters

<i>in_impl</i>	The private implementation of the member.
----------------	---

### 3.15.3 Member Function Documentation

#### 3.15.3.1 [getName](#)()

```
const std::string& rstudio::launcher_plugins::json::Object::Member::getName ( ) const
```

Gets the name of the member.

#### Returns

The name of the member.

### 3.15.3.2 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:

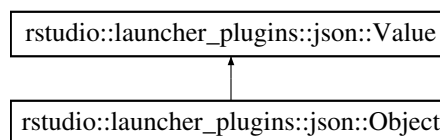
- /home/maria/GitRepos/rstudio-launcher-plugin-sdk/sdk/include/json/Json.hpp

## 3.16 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)  
*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 [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.*
- 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

### 3.16.1 Detailed Description

Class which represents a specific type of JSON `Value`: a JSON object.

### 3.16.2 Constructor & Destructor Documentation

#### 3.16.2.1 `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

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

**3.16.2.2 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.
-----------------	-------------------------------

**3.16.2.3 Object()** [3/3]

```
rstudio::launcher_plugins::json::Object::Object (
    Object && in_other )
```

Move constructor.

**Parameters**

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

**3.16.3 Member Function Documentation****3.16.3.1 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.

**3.16.3.2 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.

**3.16.3.3 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.

**3.16.3.4 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.

**3.16.3.5 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> ↔ <i>_itr</i>	The iterator pointing to the member to erase.
----------------------------	---

**Returns**

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

**3.16.3.6 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.

**3.16.3.7 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.

**3.16.3.8 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.

### 3.16.3.9 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.

### 3.16.3.10 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.

### 3.16.3.11 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.

**3.16.3.12 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.

**3.16.3.13 insert() [1/2]**

```
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.
------------------	-----------------------

**3.16.3.14 insert() [2/2]**

```
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.

**3.16.3.15 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.

**3.16.3.16 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.

**3.16.3.17 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.

**3.16.3.18 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.

**3.16.3.19 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.

**3.16.3.20 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

<code>in_name</code>	The name of the member to access.
----------------------	-----------------------------------

#### Returns

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

### 3.16.3.21 `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.

### 3.16.3.22 `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.

### 3.16.3.23 `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.

**3.16.3.24 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:

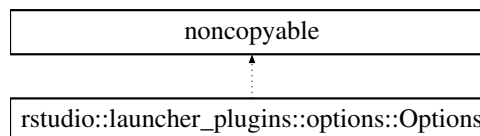
- /home/maria/GitRepos/rstudio-launcher-plugin-sdk/sdk/include/json/Json.hpp

**3.17 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.*
- unsigned int [getJobExpiryHours](#) () const  
*Gets the number of hours after which finished jobs expire and should be pruned from the plugin.*
- unsigned int [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](#) & [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.*
- unsigned int [getThreadPoolSize](#) () const  
*Gets the size of the thread pool.*

## Static Public Member Functions

- static [Options](#) & [getInstance](#) ()  
*Gets the single instance of [Options](#) for the plugin.*

### 3.17.1 Detailed Description

[Options](#) for the plugin.

### 3.17.2 Member Function Documentation

#### 3.17.2.1 [getHeartbeatIntervalSeconds\(\)](#)

```
unsigned int rstudio::launcher_plugins::options::Options::getHeartbeatIntervalSeconds ( )
const
```

Gets the number of seconds between heartbeats.

#### Returns

The number of seconds between heartbeats.

### 3.17.2.2 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.

### 3.17.2.3 getJobExpiryHours()

```
unsigned int 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.

### 3.17.2.4 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.

### 3.17.2.5 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.

### 3.17.2.6 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.

### 3.17.2.7 getThreadPoolSize()

```
unsigned int rstudio::launcher_plugins::options::Options::getThreadPoolSize ( ) const
```

Gets the size of the thread pool.

## Returns

The size of the thread pool.

### 3.17.2.8 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.

### 3.17.2.9 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.

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

- `/home/maria/GitRepos/rstudio-launcher-plugin-sdk/sdk/include/options/Options.hpp`

### 3.18 `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.*

#### 3.18.1 Detailed Description

Struct which implements the deleter for `PathScopImpl`.

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

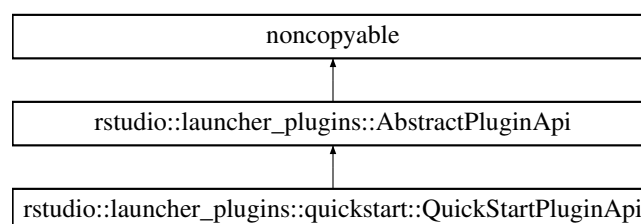
- `/home/maria/GitRepos/rstudio-launcher-plugin-sdk/sdk/include/system/FilePath.hpp`

### 3.19 `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

- [Error initialize](#) () override

*This method should initialize any components needed to communicate with the job scheduling tool, including custom options (TODO: other examples).*

### 3.19.1 Detailed Description

Launcher Plugin API for the QuickStart Plugin.

### 3.19.2 Member Function Documentation

#### 3.19.2.1 initialize()

```
Error rstudio::launcher_plugins::quickstart::QuickStartPluginApi::initialize ( ) [override],  
[virtual]
```

This method should initialize any components needed to communicate with the job scheduling tool, including custom options (TODO: other examples).

#### Returns

[Success](#) if all components needed by this Plugin were successfully initialized; [Error](#) otherwise.

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

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

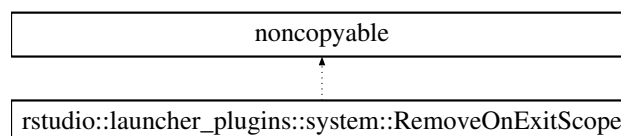
- /home/maria/GitRepos/rstudio-launcher-plugin-sdk/plugins/QuickStart/include/QuickStartPluginApi.hpp

## 3.20 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.*

### 3.20.1 Detailed Description

RAII class for restoring the current working directory.

### 3.20.2 Constructor & Destructor Documentation

#### 3.20.2.1 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:

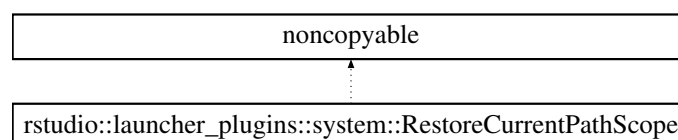
- /home/maria/GitRepos/rstudio-launcher-plugin-sdk/sdk/include/system/FilePath.hpp

## 3.21 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.*

### 3.21.1 Detailed Description

RAII class for restoring the current working directory.

### 3.21.2 Constructor & Destructor Documentation

#### 3.21.2.1 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:

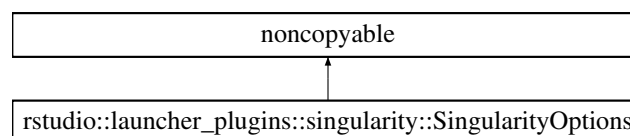
- /home/maria/GitRepos/rstudio-launcher-plugin-sdk/sdk/include/system/FilePath.hpp

## 3.22 rstudio::launcher\_plugins::singularity::SingularityOptions Class Reference

Class which stores options specific to the Singularity Container system.

```
#include <SingularityOptions.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::singularity::SingularityOptions:



## Public Member Functions

- const [system::FilePath](#) & [getRContainer](#) () const  
*Gets the Singularity container to use for R.*
- const [system::FilePath](#) & [getRSessionContainer](#) () const  
*Gets the Singularity container to use for R Sessions.*
- void [initialize](#) ()  
*Method which initializes [SingularityOptions](#). This method should be called exactly once, before the options file is read.*

## Static Public Member Functions

- static [SingularityOptions](#) & [getInstance](#) ()  
*Gets the single instance of [SingularityOptions](#) for the plugin.*

### 3.22.1 Detailed Description

Class which stores options specific to the Singularity Container system.

### 3.22.2 Member Function Documentation

#### 3.22.2.1 [getInstance\(\)](#)

```
static SingularityOptions& rstudio::launcher_plugins::singularity::SingularityOptions::get↵
Instance ( ) [static]
```

Gets the single instance of [SingularityOptions](#) for the plugin.

#### Returns

The single instance of [SingularityOptions](#) for the plugin.

#### 3.22.2.2 [getRContainer\(\)](#)

```
const system::FilePath& rstudio::launcher_plugins::singularity::SingularityOptions::getR↵
Container ( ) const
```

Gets the Singularity container to use for R.

NOTE: This option is a placeholder until I know what options I need. It will not end up in the release version of this plugin.

#### Returns

The Singularity container to use for R.

### 3.22.2.3 getRSessionContainer()

```
const system::FilePath& rstudio::launcher_plugins::singularity::SingularityOptions::getRSessionContainer ( ) const
```

Gets the Singularity container to use for R Sessions.

NOTE: This option is a placeholder until I know what options I need. It will not end up in the release version of this plugin.

#### Returns

The Singularity container to use for R Sessions.

### 3.22.2.4 initialize()

```
void rstudio::launcher_plugins::singularity::SingularityOptions::initialize ( )
```

Method which initializes [SingularityOptions](#). This method should be called exactly once, before the options file is read.

This is where [SingularityOptions](#) are registered with the Options object.

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

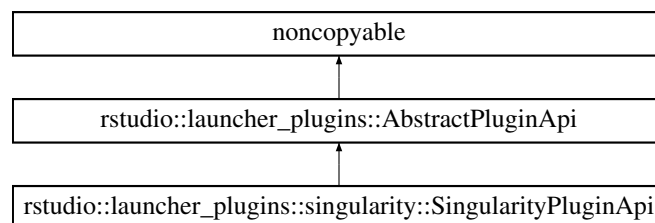
- /home/maria/GitRepos/rstudio-launcher-plugin-sdk/plugins/Singularity/include/SingularityOptions.hpp

## 3.23 rstudio::launcher\_plugins::singularity::SingularityPluginApi Class Reference

Launcher Plugin API for the Singularity Plugin.

```
#include <SingularityPluginApi.hpp>
```

Inheritance diagram for rstudio::launcher\_plugins::singularity::SingularityPluginApi:



### Public Member Functions

- [Error initialize](#) () override

*This method should initialize any components needed to communicate with the job scheduling tool, including custom options (TODO: other examples).*

### 3.23.1 Detailed Description

Launcher Plugin API for the Singularity Plugin.

### 3.23.2 Member Function Documentation

#### 3.23.2.1 initialize()

```
Error rstudio::launcher_plugins::singularity::SingularityPluginApi::initialize ( ) [override],  
[virtual]
```

This method should initialize any components needed to communicate with the job scheduling tool, including custom options (TODO: other examples).

#### Returns

[Success](#) if all components needed by this Plugin were successfully initialized; [Error](#) otherwise.

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

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

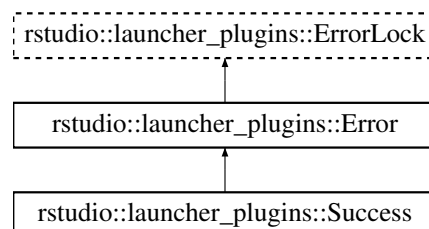
- /home/maria/GitRepos/rstudio-launcher-plugin-sdk/plugins/Singularity/include/SingularityPluginApi.hpp

## 3.24 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.*

### 3.24.1 Detailed Description

Class which represents a successful operation (i.e. no error).

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

- /home/maria/GitRepos/rstudio-launcher-plugin-sdk/sdk/include/Error.hpp

## 3.25 rstudio::launcher\_plugins::system::User Class Reference

Class which represents a system user.

```
#include <User.hpp>
```

### Public Member Functions

- [User](#) ()  
*Default constructor.*
- [User](#) (const [User](#) &in\_other)  
*Copy constructor.*
- 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.*
- [UidType](#) [getUserId](#) () const  
*Gets the ID of this user.*
- const std::string & [getUsername](#) () const  
*Returns the name of this user.*
- [User](#) & [operator=](#) (const [User](#) &in\_other)  
*Overloaded assignment operator.*

### 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.*

### 3.25.1 Detailed Description

Class which represents a system user.

### 3.25.2 Constructor & Destructor Documentation

#### 3.25.2.1 User() [1/2]

```
rstudio::launcher_plugins::system::User::User ( )
```

Default constructor.

Creates a user object which represents all users.

#### 3.25.2.2 User() [2/2]

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

Copy constructor.

##### Parameters

<i>in_other</i>	The user to copy.
-----------------	-------------------

### 3.25.3 Member Function Documentation

#### 3.25.3.1 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.

### 3.25.3.2 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.

**3.25.3.3 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.

**3.25.3.4 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.

**3.25.3.5 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.

**3.25.3.6** `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.

**3.25.3.7** `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.

**3.25.3.8** `getUserId()`

```
UidType rstudio::launcher_plugins::system::User::getUserId ( ) const
```

Gets the ID of this user.

**Returns**

The ID of this user.

**3.25.3.9 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).

**3.25.3.10 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.

**3.25.3.11 isEmpty()**

```
bool rstudio::launcher_plugins::system::User::isEmpty ( ) const
```

Checks whether this user is empty or not.

**Returns****3.25.3.12 operator=()**

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

Overloaded assignment operator.

## Parameters

<code>in_other</code>	The user to copy to this one.
-----------------------	-------------------------------

## Returns

This user.

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

- `/home/maria/GitRepos/rstudio-launcher-plugin-sdk/sdk/include/system/User.hpp`

## 3.26 `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](#)

#### 3.26.1 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.
----------	-------------------------------

## 3.26.2 Constructor & Destructor Documentation

### 3.26.2.1 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.
-------------------	---

### 3.26.3 Member Function Documentation

#### 3.26.3.1 setDefaultValue()

```
template<class T>
Value& rstudio::launcher_plugins::options::Value< T >::setDefaultValue (
    const T & in_defaultValue )
```

Sets the default value of the option.

##### Parameters

<code>in_defaultValue</code>	The default value of the option.
------------------------------	----------------------------------

##### Returns

A reference to this value.

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

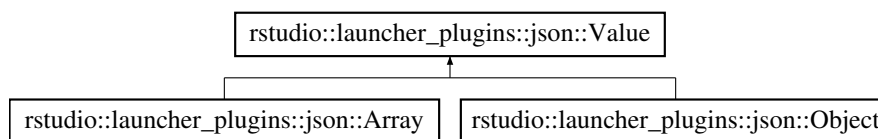
- /home/maria/GitRepos/rstudio-launcher-plugin-sdk/sdk/include/options/Options.hpp

## 3.27 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.*
- [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.*
- `Error parse (const char *in_jsonStr)`  
  *Parses the JSON string into this value.*
- `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 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](#)

### 3.27.1 Detailed Description

Class which represents a json value.

### 3.27.2 Constructor & Destructor Documentation

#### 3.27.2.1 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.
---------------------	--



### 3.27.2.2 Value() [2/12]

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

Copy constructor.

#### Parameters

<i>in_other</i>	The value to copy.
-----------------	--------------------

### 3.27.2.3 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.
-----------------	-------------------------

### 3.27.2.4 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.
-----------------	--

### 3.27.2.5 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.
-----------------	--

**3.27.2.6 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.
-----------------	--

**3.27.2.7 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.
-----------------	--

**3.27.2.8 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.
-----------------	--

### 3.27.2.9 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.
-----------------	--

### 3.27.2.10 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.
-----------------	--

### 3.27.2.11 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.
-----------------	--

### 3.27.2.12 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.
-----------------	--

### 3.27.3 Member Function Documentation

#### 3.27.3.1 clone()

```
Value rstudio::launcher_plugins::json::Value::clone ( ) const
```

Makes a copy of this JSON value.

##### Returns

A copy of this JSON value.

#### 3.27.3.2 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.

#### 3.27.3.3 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.

#### 3.27.3.4 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.

#### 3.27.3.5 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.

#### 3.27.3.6 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.

#### 3.27.3.7 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.

### 3.27.3.8 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.

### 3.27.3.9 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.

### 3.27.3.10 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.

### 3.27.3.11 getType()

```
Type rstudio::launcher_plugins::json::Value::getType ( ) const
```

Gets the type of this value.

#### Returns

The type of this value.

### 3.27.3.12 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.

### 3.27.3.13 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.

### 3.27.3.14 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`.

### 3.27.3.15 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.

**3.27.3.16 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.

**3.27.3.17 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.

**3.27.3.18 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.

**3.27.3.19 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.



### 3.27.3.20 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.

### 3.27.3.21 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.

### 3.27.3.22 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.

### 3.27.3.23 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.

### 3.27.3.24 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.

### 3.27.3.25 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.

### 3.27.3.26 operator=() [1/11]

```
Value& rstudio::launcher_plugins::json::Value::operator= (
    bool in_value )
```

Assignment operator.

#### Parameters

<code>in_value</code>	The literal value to set this JSON <a href="#">Value</a> to.
-----------------------	--

#### Returns

A reference to this value.

### 3.27.3.27 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.

**3.27.3.28 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.

**3.27.3.29 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.

**3.27.3.30 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.

**3.27.3.31 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.

**3.27.3.32 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.

**3.27.3.33 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.

**3.27.3.34 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.

**3.27.3.35 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.

**3.27.3.36 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.

**3.27.3.37 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.

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

```
Error rstudio::launcher_plugins::json::Value::parse (
    const char * in_jsonStr )
```

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](#))

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

```
Error rstudio::launcher_plugins::json::Value::parse (
    const std::string & in_jsonStr )
```

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](#))

**3.27.3.40 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.

**3.27.3.41 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.

**3.27.3.42 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.

**3.27.3.43 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.
-------------------	---

**3.27.3.44 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.

**3.27.3.45 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:

- `/home/maria/GitRepos/rstudio-launcher-plugin-sdk/sdk/include/json/Json.hpp`



# Index

- addOrUpdateProperty
  - rstudio::launcher\_plugins::Error, [21](#), [22](#)
- addProperty
  - rstudio::launcher\_plugins::Error, [22](#), [23](#)
- Array
  - rstudio::launcher\_plugins::json::Array, [9](#)
- asString
  - rstudio::launcher\_plugins::Error, [23](#)
  - rstudio::launcher\_plugins::ErrorLocation, [31](#)
- begin
  - rstudio::launcher\_plugins::json::Array, [10](#)
  - rstudio::launcher\_plugins::json::Object, [78](#)
- clone
  - rstudio::launcher\_plugins::json::Value, [112](#)
- coerce
  - rstudio::launcher\_plugins::json::Value, [112](#)
- completeChildPath
  - rstudio::launcher\_plugins::system::FilePath, [42](#), [43](#)
- completePath
  - rstudio::launcher\_plugins::system::FilePath, [43](#)
- copy
  - rstudio::launcher\_plugins::system::FilePath, [44](#)
- copyDirectoryRecursive
  - rstudio::launcher\_plugins::system::FilePath, [44](#)
- createAliasedPath
  - rstudio::launcher\_plugins::system::FilePath, [44](#)
- createDirectory
  - rstudio::launcher\_plugins::system::FilePath, [45](#)
- createMember
  - rstudio::launcher\_plugins::json::Object, [78](#)
- doRotation
  - rstudio::launcher\_plugins::logging::FileLogOptions, [37](#)
- end
  - rstudio::launcher\_plugins::json::Array, [10](#)
  - rstudio::launcher\_plugins::json::Object, [79](#)
- ensureDirectory
  - rstudio::launcher\_plugins::system::FilePath, [45](#)
- ensureFile
  - rstudio::launcher\_plugins::system::FilePath, [45](#)
- erase
  - rstudio::launcher\_plugins::json::Array, [10](#), [11](#)
  - rstudio::launcher\_plugins::json::Object, [79](#), [80](#)
- Error
  - rstudio::launcher\_plugins::Error, [18–20](#)
- ErrorLocation
  - rstudio::launcher\_plugins::ErrorLocation, [30](#)
- exists
  - rstudio::launcher\_plugins::system::FilePath, [45](#), [46](#)
  - rstudio::launcher\_plugins::system::User, [98](#)
- FileLogDestination
  - rstudio::launcher\_plugins::logging::FileLogDestination, [34](#)
- FileLogOptions
  - rstudio::launcher\_plugins::logging::FileLogOptions, [36](#)
- FilePath
  - rstudio::launcher\_plugins::system::FilePath, [42](#)
- find
  - rstudio::launcher\_plugins::json::Object, [80](#)
- getAbsolutePath
  - rstudio::launcher\_plugins::system::FilePath, [46](#)
- getAbsolutePathNative
  - rstudio::launcher\_plugins::system::FilePath, [46](#)
- getArray
  - rstudio::launcher\_plugins::json::Value, [112](#)
- getBack
  - rstudio::launcher\_plugins::json::Array, [11](#)
- getBool
  - rstudio::launcher\_plugins::json::Value, [112](#)
- getCanonicalPath
  - rstudio::launcher\_plugins::system::FilePath, [46](#)
- getCause
  - rstudio::launcher\_plugins::Error, [23](#)
- getChildren
  - rstudio::launcher\_plugins::system::FilePath, [47](#)
- getChildrenRecursive
  - rstudio::launcher\_plugins::system::FilePath, [47](#)
- getCode
  - rstudio::launcher\_plugins::Error, [23](#)
- getCurrentUser
  - rstudio::launcher\_plugins::system::User, [98](#)
- getDirectory
  - rstudio::launcher\_plugins::logging::FileLogOptions, [37](#)
- getDouble
  - rstudio::launcher\_plugins::json::Value, [113](#)
- getExtension
  - rstudio::launcher\_plugins::system::FilePath, [47](#)
- getExtensionLowerCase
  - rstudio::launcher\_plugins::system::FilePath, [48](#)
- getFile
  - rstudio::launcher\_plugins::ErrorLocation, [31](#)
- getFileMode

- rstudio::launcher\_plugins::logging::FileLogOptions, 37
- getFilename
  - rstudio::launcher\_plugins::system::FilePath, 48
- getFloat
  - rstudio::launcher\_plugins::json::Value, 113
- getFront
  - rstudio::launcher\_plugins::json::Array, 11
- getFunction
  - rstudio::launcher\_plugins::ErrorLocation, 31
- getGroupId
  - rstudio::launcher\_plugins::system::User, 100
- getHeartbeatIntervalSeconds
  - rstudio::launcher\_plugins::options::Options, 87
- getHomePath
  - rstudio::launcher\_plugins::system::User, 100
- getId
  - rstudio::launcher\_plugins::logging::FileLogDestination, 34
  - rstudio::launcher\_plugins::logging::ILogDestination, 62
- getInstance
  - rstudio::launcher\_plugins::options::Options, 87
  - rstudio::launcher\_plugins::singularity::SingularityOptions, 94
- getInt
  - rstudio::launcher\_plugins::json::Value, 113
- getInt64
  - rstudio::launcher\_plugins::json::Value, 113
- getJobExpiryHours
  - rstudio::launcher\_plugins::options::Options, 88
- getLastWriteTime
  - rstudio::launcher\_plugins::system::FilePath, 48
- getLexicallyNormalPath
  - rstudio::launcher\_plugins::system::FilePath, 48
- getLine
  - rstudio::launcher\_plugins::ErrorLocation, 31
- getLocation
  - rstudio::launcher\_plugins::Error, 23
- getLogLevel
  - rstudio::launcher\_plugins::logging::ILogDestination, 63
  - rstudio::launcher\_plugins::options::Options, 88
- getMaxSizeMb
  - rstudio::launcher\_plugins::logging::FileLogOptions, 37
- getMessage
  - rstudio::launcher\_plugins::Error, 24
- getMimeType
  - rstudio::launcher\_plugins::system::FilePath, 49
- getName
  - rstudio::launcher\_plugins::Error, 24
  - rstudio::launcher\_plugins::json::Object::Member, 74
- getObject
  - rstudio::launcher\_plugins::json::Value, 114
- getParent
  - rstudio::launcher\_plugins::system::FilePath, 49
- getProperties
  - rstudio::launcher\_plugins::Error, 24
- getProperty
  - rstudio::launcher\_plugins::Error, 24
- getRContainer
  - rstudio::launcher\_plugins::singularity::SingularityOptions, 94
- getRelativePath
  - rstudio::launcher\_plugins::system::FilePath, 49
- getRSessionContainer
  - rstudio::launcher\_plugins::singularity::SingularityOptions, 94
- getSchemaDefaults
  - rstudio::launcher\_plugins::json::Object, 81
- getScratchPath
  - rstudio::launcher\_plugins::options::Options, 88
- getServerUser
  - rstudio::launcher\_plugins::options::Options, 88
- getSize
  - rstudio::launcher\_plugins::json::Array, 11
  - rstudio::launcher\_plugins::json::Object, 81
  - rstudio::launcher\_plugins::system::FilePath, 50
- getSizeRecursive
  - rstudio::launcher\_plugins::system::FilePath, 50
- getStem
  - rstudio::launcher\_plugins::system::FilePath, 50
- getString
  - rstudio::launcher\_plugins::json::Value, 114
- getSummary
  - rstudio::launcher\_plugins::Error, 25
- getThreadPoolSize
  - rstudio::launcher\_plugins::options::Options, 89
- getType
  - rstudio::launcher\_plugins::json::Value, 114
- getUInt
  - rstudio::launcher\_plugins::json::Value, 114
- getUInt64
  - rstudio::launcher\_plugins::json::Value, 115
- getUserFromIdentifier
  - rstudio::launcher\_plugins::system::User, 100, 101
- getUserHomePath
  - rstudio::launcher\_plugins::system::User, 101
- getUserId
  - rstudio::launcher\_plugins::system::User, 101
- getUsername
  - rstudio::launcher\_plugins::system::User, 102
- getValue
  - rstudio::launcher\_plugins::json::Object::Member, 74
  - rstudio::launcher\_plugins::json::Value, 115
- getValueAt
  - rstudio::launcher\_plugins::json::Array, 12
- hasExtension
  - rstudio::launcher\_plugins::system::FilePath, 50
- hasExtensionLowerCase
  - rstudio::launcher\_plugins::system::FilePath, 51
- hasLocation
  - rstudio::launcher\_plugins::ErrorLocation, 31



- hasMember
  - rstudio::launcher\_plugins::json::Object, [81](#), [82](#)
- hasTextMimeType
  - rstudio::launcher\_plugins::system::FilePath, [51](#)
- ILogDestination
  - rstudio::launcher\_plugins::logging::ILogDestination, [62](#)
- includePid
  - rstudio::launcher\_plugins::logging::FileLogOptions, [38](#)
- Init
  - rstudio::launcher\_plugins::options::Options::Init, [64](#)
- initialize
  - rstudio::launcher\_plugins::AbstractPluginApi, [6](#)
  - rstudio::launcher\_plugins::quickstart::QuickStartPluginApi, [91](#)
  - rstudio::launcher\_plugins::singularity::SingularityOptions, [95](#)
  - rstudio::launcher\_plugins::singularity::SingularityPluginApi, [96](#)
- insert
  - rstudio::launcher\_plugins::json::Object, [82](#)
- isAllUsers
  - rstudio::launcher\_plugins::system::User, [102](#)
- isArray
  - rstudio::launcher\_plugins::json::Value, [115](#)
- isBool
  - rstudio::launcher\_plugins::json::Value, [116](#)
- isDirectory
  - rstudio::launcher\_plugins::system::FilePath, [51](#)
- isDouble
  - rstudio::launcher\_plugins::json::Value, [116](#)
- isEmpty
  - rstudio::launcher\_plugins::json::Array, [12](#)
  - rstudio::launcher\_plugins::json::Object, [83](#)
  - rstudio::launcher\_plugins::system::FilePath, [51](#)
  - rstudio::launcher\_plugins::system::User, [102](#)
- isEqualCaseInsensitive
  - rstudio::launcher\_plugins::system::FilePath, [52](#)
- isEquivalentTo
  - rstudio::launcher\_plugins::system::FilePath, [52](#)
- isExpected
  - rstudio::launcher\_plugins::Error, [25](#)
- isFloat
  - rstudio::launcher\_plugins::json::Value, [116](#)
- isHidden
  - rstudio::launcher\_plugins::system::FilePath, [52](#)
- isInt
  - rstudio::launcher\_plugins::json::Value, [116](#)
- isInt64
  - rstudio::launcher\_plugins::json::Value, [116](#)
- isJunction
  - rstudio::launcher\_plugins::system::FilePath, [53](#)
- isNull
  - rstudio::launcher\_plugins::json::Value, [117](#)
- isObject
  - rstudio::launcher\_plugins::json::Value, [117](#)
- isRegularFile
  - rstudio::launcher\_plugins::system::FilePath, [53](#)
- isRootPath
  - rstudio::launcher\_plugins::system::FilePath, [53](#)
- isString
  - rstudio::launcher\_plugins::json::Value, [117](#)
- isSymlink
  - rstudio::launcher\_plugins::system::FilePath, [54](#)
- isUInt
  - rstudio::launcher\_plugins::json::Value, [117](#)
- isUInt64
  - rstudio::launcher\_plugins::json::Value, [118](#)
- isWithin
  - rstudio::launcher\_plugins::system::FilePath, [54](#)
- Iterator
  - rstudio::launcher\_plugins::json::Array::Iterator, [71](#)
  - rstudio::launcher\_plugins::json::Object::Iterator, [67](#)
- makeCurrent
  - rstudio::launcher\_plugins::system::FilePath, [54](#)
- makeCurrentPath
  - rstudio::launcher\_plugins::system::FilePath, [55](#)
- Member
  - rstudio::launcher\_plugins::json::Object::Member, [74](#)
- mergeObjects
  - rstudio::launcher\_plugins::json::Object, [83](#)
- move
  - rstudio::launcher\_plugins::system::FilePath, [55](#)
- MoveCrossDevice
  - rstudio::launcher\_plugins::system::FilePath, [42](#)
- MoveDirect
  - rstudio::launcher\_plugins::system::FilePath, [42](#)
- moveIndirect
  - rstudio::launcher\_plugins::system::FilePath, [55](#)
- MoveType
  - rstudio::launcher\_plugins::system::FilePath, [42](#)
- Object
  - rstudio::launcher\_plugins::json::Object, [77](#), [78](#)
- openForRead
  - rstudio::launcher\_plugins::system::FilePath, [56](#)
- openForWrite
  - rstudio::launcher\_plugins::system::FilePath, [56](#)
- operator bool
  - rstudio::launcher\_plugins::Error, [25](#)
- operator!
  - rstudio::launcher\_plugins::Error, [25](#)
- operator!=
  - rstudio::launcher\_plugins::Error, [26](#), [27](#)
  - rstudio::launcher\_plugins::json::Array::Iterator, [71](#)
  - rstudio::launcher\_plugins::json::Object::Iterator, [67](#)
  - rstudio::launcher\_plugins::system::FilePath, [56](#)
- operator<
  - rstudio::launcher\_plugins::system::FilePath, [57](#)
- operator\*
  - rstudio::launcher\_plugins::json::Array::Iterator, [72](#)
  - rstudio::launcher\_plugins::json::Object::Iterator, [68](#)
- operator()
  -

- rstudio::launcher\_plugins::options::Options::Init, 64, 65
- operator++
  - rstudio::launcher\_plugins::json::Array::Iterator, 72
  - rstudio::launcher\_plugins::json::Object::Iterator, 68
- operator--
  - rstudio::launcher\_plugins::json::Array::Iterator, 72
  - rstudio::launcher\_plugins::json::Object::Iterator, 68, 69
- operator=
  - rstudio::launcher\_plugins::ErrorLocation, 32
  - rstudio::launcher\_plugins::json::Array, 12, 13
  - rstudio::launcher\_plugins::json::Array::Iterator, 73
  - rstudio::launcher\_plugins::json::Object, 83, 84
  - rstudio::launcher\_plugins::json::Object::Iterator, 69
  - rstudio::launcher\_plugins::json::Value, 118, 119, 121, 123
  - rstudio::launcher\_plugins::system::User, 102
- operator==
  - rstudio::launcher\_plugins::Error, 27
  - rstudio::launcher\_plugins::ErrorLocation, 32
  - rstudio::launcher\_plugins::json::Array::Iterator, 73
  - rstudio::launcher\_plugins::json::Object::Iterator, 69
  - rstudio::launcher\_plugins::json::Value, 125
  - rstudio::launcher\_plugins::system::FilePath, 57
- operator[]
  - rstudio::launcher\_plugins::json::Array, 13
  - rstudio::launcher\_plugins::json::Object, 84
- parse
  - rstudio::launcher\_plugins::json::Value, 125
- parseAndValidate
  - rstudio::launcher\_plugins::json::Value, 127
- push\_back
  - rstudio::launcher\_plugins::json::Array, 13
- rbegin
  - rstudio::launcher\_plugins::json::Array, 14
  - rstudio::launcher\_plugins::json::Object, 85
- readOptions
  - rstudio::launcher\_plugins::options::Options, 89
- RecursiveIterationFunction
  - rstudio::launcher\_plugins::system::FilePath, 41
- registerOptions
  - rstudio::launcher\_plugins::options::Options, 89
- remove
  - rstudio::launcher\_plugins::system::FilePath, 58
- removeIfExists
  - rstudio::launcher\_plugins::system::FilePath, 58
- RemoveOnExitScope
  - rstudio::launcher\_plugins::system::RemoveOnExitScope, 92
- rend
  - rstudio::launcher\_plugins::json::Array, 14
  - rstudio::launcher\_plugins::json::Object, 85
- resetDirectory
  - rstudio::launcher\_plugins::system::FilePath, 58
- resolveAliasedPath
  - rstudio::launcher\_plugins::system::FilePath, 58
- resolveSymlink
  - rstudio::launcher\_plugins::system::FilePath, 59
- RestoreCurrentPathScope
  - rstudio::launcher\_plugins::system::RestoreCurrentPathScope, 93
- rstudio::launcher\_plugins::AbstractMain, 5
  - run, 5
- rstudio::launcher\_plugins::AbstractPluginApi, 6
  - initialize, 6
- rstudio::launcher\_plugins::Error, 16
  - addOrUpdateProperty, 21, 22
  - addProperty, 22, 23
  - asString, 23
  - Error, 18–20
  - getCause, 23
  - getCode, 23
  - getLocation, 23
  - getMessage, 24
  - getName, 24
  - getProperties, 24
  - getProperty, 24
  - getSummary, 25
  - isExpected, 25
  - operator bool, 25
  - operator!, 25
  - operator!=, 26, 27
  - operator==, 27
- rstudio::launcher\_plugins::ErrorLocation, 29
  - asString, 31
  - ErrorLocation, 30
  - getFile, 31
  - getFunction, 31
  - getLine, 31
  - hasLocation, 31
  - operator=, 32
  - operator==, 32
- rstudio::launcher\_plugins::ErrorLock, 33
- rstudio::launcher\_plugins::json::Array, 7
  - Array, 9
  - begin, 10
  - end, 10
  - erase, 10, 11
  - getBack, 11
  - getFront, 11
  - getSize, 11
  - getValueAt, 12
  - isEmpty, 12
  - operator=, 12, 13
  - operator[], 13
  - push\_back, 13
  - rbegin, 14
  - rend, 14
  - toSetString, 14
  - toStringPairList, 15
  - toVectorInt, 15
  - toVectorString, 15
- rstudio::launcher\_plugins::json::Array::Iterator, 70
  - Iterator, 71

- operator!=, 71
- operator\*, 72
- operator++, 72
- operator--, 72
- operator=, 73
- operator==, 73
- rstudio::launcher\_plugins::json::detail::is\_json\_type< T  
>, 66
- rstudio::launcher\_plugins::json::Object, 75
  - begin, 78
  - createMember, 78
  - end, 79
  - erase, 79, 80
  - find, 80
  - getSchemaDefaults, 81
  - getSize, 81
  - hasMember, 81, 82
  - insert, 82
  - isEmpty, 83
  - mergeObjects, 83
  - Object, 77, 78
  - operator=, 83, 84
  - operator[], 84
  - rbegin, 85
  - rend, 85
  - toStringMap, 85
  - toStringPairList, 86
- rstudio::launcher\_plugins::json::Object::Iterator, 66
  - Iterator, 67
  - operator!=, 67
  - operator\*, 68
  - operator++, 68
  - operator--, 68, 69
  - operator=, 69
  - operator==, 69
- rstudio::launcher\_plugins::json::Object::Member, 73
  - getName, 74
  - getValue, 74
  - Member, 74
- rstudio::launcher\_plugins::json::Value, 105
  - clone, 112
  - coerce, 112
  - getArray, 112
  - getBool, 112
  - getDouble, 113
  - getFloat, 113
  - getInt, 113
  - getInt64, 113
  - getObject, 114
  - getString, 114
  - getType, 114
  - getUInt, 114
  - getUInt64, 115
  - getValue, 115
  - isArray, 115
  - isBool, 116
  - isDouble, 116
  - isFloat, 116
  - isInt, 116
  - isInt64, 116
  - isNull, 117
  - isObject, 117
  - isString, 117
  - isUInt, 117
  - isUInt64, 118
  - operator=, 118, 119, 121, 123
  - operator==, 125
  - parse, 125
  - parseAndValidate, 127
  - validate, 127
  - Value, 108–111
  - write, 127, 128
  - writeFormatted, 128
- rstudio::launcher\_plugins::logging::FileLogDestination, 33
  - FileLogDestination, 34
  - getId, 34
  - writeLog, 35
- rstudio::launcher\_plugins::logging::FileLogOptions, 35
  - doRotation, 37
  - FileLogOptions, 36
  - getDirectory, 37
  - getFileMode, 37
  - getMaxSizeMb, 37
  - includePid, 38
- rstudio::launcher\_plugins::logging::ILogDestination, 61
  - getId, 62
  - getLogLevel, 63
  - ILogDestination, 62
  - writeLog, 63
- rstudio::launcher\_plugins::options::Options, 86
  - getHeartbeatIntervalSeconds, 87
  - getInstance, 87
  - getJobExpiryHours, 88
  - getLogLevel, 88
  - getScratchPath, 88
  - getServerUser, 88
  - getThreadPoolSize, 89
  - readOptions, 89
  - registerOptions, 89
- rstudio::launcher\_plugins::options::Options::Init, 63
  - Init, 64
  - operator(), 64, 65
- rstudio::launcher\_plugins::options::Value< T >, 103
  - setDefaultValue, 105
  - Value, 104
- rstudio::launcher\_plugins::quickstart::QuickStartPluginApi, 90
  - initialize, 91
- rstudio::launcher\_plugins::singularity::SingularityOptions, 93
  - getInstance, 94
  - getRContainer, 94
  - getRSessionContainer, 94
  - initialize, 95

- rstudio::launcher\_plugins::singularity::SingularityPluginApi, 95
  - initialize, 96
- rstudio::launcher\_plugins::Success, 96
- rstudio::launcher\_plugins::system::FilePath, 38
  - completeChildPath, 42, 43
  - completePath, 43
  - copy, 44
  - copyDirectoryRecursive, 44
  - createAliasedPath, 44
  - createDirectory, 45
  - ensureDirectory, 45
  - ensureFile, 45
  - exists, 45, 46
  - FilePath, 42
  - getAbsolutePath, 46
  - getAbsolutePathNative, 46
  - getCanonicalPath, 46
  - getChildren, 47
  - getChildrenRecursive, 47
  - getExtension, 47
  - getExtensionLowerCase, 48
  - getFilename, 48
  - getLastWriteTime, 48
  - getLexicallyNormalPath, 48
  - getMimeType, 49
  - getParent, 49
  - getRelativePath, 49
  - getSize, 50
  - getSizeRecursive, 50
  - getStem, 50
  - hasExtension, 50
  - hasExtensionLowerCase, 51
  - hasTextMimeType, 51
  - isDirectory, 51
  - isEmpty, 51
  - isEqualCaseInsensitive, 52
  - isEquivalentTo, 52
  - isHidden, 52
  - isJunction, 53
  - isRegularFile, 53
  - isRootPath, 53
  - isSymlink, 54
  - isWithin, 54
  - makeCurrent, 54
  - makeCurrentPath, 55
  - move, 55
  - MoveCrossDevice, 42
  - MoveDirect, 42
  - moveIndirect, 55
  - MoveType, 42
  - openForRead, 56
  - openForWrite, 56
  - operator!=, 56
  - operator<, 57
  - operator==, 57
  - RecursiveIterationFunction, 41
  - remove, 58
  - removeIfExists, 58
  - resetDirectory, 58
  - resolveAliasedPath, 58
  - resolveSymlink, 59
  - safeCurrentPath, 59
  - setLastWriteTime, 59
  - tempFilePath, 60
  - uniqueFilePath, 60, 61
- rstudio::launcher\_plugins::system::PathScopeImplDeleter, 90
- rstudio::launcher\_plugins::system::RemoveOnExitScope, 91
  - RemoveOnExitScope, 92
- rstudio::launcher\_plugins::system::RestoreCurrentPathScope, 92
  - RestoreCurrentPathScope, 93
- rstudio::launcher\_plugins::system::User, 97
  - exists, 98
  - getCurrentUser, 98
  - getGroupId, 100
  - getHomePath, 100
  - getUserFromIdentifier, 100, 101
  - getUserHomePath, 101
  - getUserId, 101
  - getUsername, 102
  - isAllUsers, 102
  - isEmpty, 102
  - operator=, 102
  - User, 98
- run
  - rstudio::launcher\_plugins::AbstractMain, 5
- safeCurrentPath
  - rstudio::launcher\_plugins::system::FilePath, 59
- setDefaultValue
  - rstudio::launcher\_plugins::options::Value< T >, 105
- setLastWriteTime
  - rstudio::launcher\_plugins::system::FilePath, 59
- tempFilePath
  - rstudio::launcher\_plugins::system::FilePath, 60
- toSetString
  - rstudio::launcher\_plugins::json::Array, 14
- toStringMap
  - rstudio::launcher\_plugins::json::Object, 85
- toStringPairList
  - rstudio::launcher\_plugins::json::Array, 15
  - rstudio::launcher\_plugins::json::Object, 86
- toVectorInt
  - rstudio::launcher\_plugins::json::Array, 15
- toVectorString
  - rstudio::launcher\_plugins::json::Array, 15
- uniqueFilePath
  - rstudio::launcher\_plugins::system::FilePath, 60, 61
- User
  - rstudio::launcher\_plugins::system::User, 98

validate

    rstudio::launcher\_plugins::json::Value, [127](#)

Value

    rstudio::launcher\_plugins::json::Value, [108–111](#)

    rstudio::launcher\_plugins::options::Value< T >, [104](#)

write

    rstudio::launcher\_plugins::json::Value, [127](#), [128](#)

writeFormatted

    rstudio::launcher\_plugins::json::Value, [128](#)

writeLog

    rstudio::launcher\_plugins::logging::FileLogDestination, [35](#)

    rstudio::launcher\_plugins::logging::ILogDestination, [63](#)