



Silk Performance Manager 20.0

API Help

Micro Focus
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
UK
<http://www.microfocus.com>

© Copyright 2001-2019 Micro Focus or one of its affiliates.

MICRO FOCUS, the Micro Focus logo and Silk Performance Manager are trademarks or registered trademarks of Micro Focus or one of its affiliates.

All other marks are the property of their respective owners.

2019-05-02

Contents

Introduction	7
Setting Up Web Services	8
Testing Your Configuration	8
Web Service Session Handling	9
Web Services	10
sccsystem	10
AppModule Class	10
Entity Class	11
NamedEntity Class	11
convertFromReadableTime	12
convertToReadableTime Method (sccsystem)	13
encryptPassword Method (sccsystem)	13
getApplicationModules Method (AppModule)	14
logonUser Method (sccsystem)	14
scentities	15
Project Class	17
Location Class	18
LocationDetails Class	19
EssentialDescription Class	21
EssentialGroup Class	22
EssentialSubGroup Class	22
ExecServer Class	22
FilePoolEntry Class	26
User Class	28
UserDetails Class	29
UserGroup Class	33
UserGroupMembership Class	35
Role Class	37
getAllRoles Method (scentities)	38
getAllUserGroups Method (scentities)	38
getProjects Method (scentities)	39
getProjectsForUser Method (scentities)	39
getLocations Method (scentities)	39
getLocationsForProject Method (scentities)	40
activateExecServer Method (scentities)	40
activateProjects Method (scentities)	41
addLocationToProject Method (scentities)	41
addResourceTag Method (scentities)	41
createExecServer Method (scentities)	42
createLocation Method (scentities)	42
createProject Method (scentities)	42
createUser Method (scentities)	43
createUsergroup Method	43
deactivateExecServer Method (scentities)	44
deactivateProjects Method (scentities)	44
deleteLocations Method (scentities)	44
deleteProjects Method (scentities)	45
deleteUsers Method (scentities)	45
deleteUserGroup Method (scentities)	45
encryptAndSetPassword Method (scentities)	46

getAllEssentials Method (scentities)	46
getAllMemberships Method (scentities)	46
getAllProjects Method (scentities)	47
getAllResourceTags Method (scentities)	47
getEssentialGroups Method (scentities)	48
getEssentials Method (scentities)	48
getEssentialSubGroups Method (scentities)	48
getExecServerById Method (scentities)	49
getExecServersOfLocation Method (scentities)	49
getFilePoolEntries Methods (scentities)	50
getGroupById Method (scentities)	50
getGroupByName Method (scentities)	50
getLocationDetails Method (scentities)	51
getMembershipsOfGroup Method (scentities)	51
getMembershipsOfUser Method (scentities)	51
getProjectById Method (scentities)	52
getRoleById Method (scentities)	52
getUserDetails Method (scentities)	53
getUsers Method (scentities)	53
isMixedModeAuthentication Method (scentities)	53
removeLocationFromProject Method (scentities)	54
removeResourceTag method (scentities)	54
setMixedModeAuthentication Method (scentities)	55
setPassword Method (scentities)	55
setProxyUsernameAndPasswordForLocation Method (scentities)	55
updateExecServer Method (scentities)	56
updateLocation Method (scentities)	56
updateMembershipsOfGroup Method (scentities)	57
updateMembershipsOfUser Method (scentities)	57
updateProject Method (scentities)	57
updateUser Method (scentities)	58
updateUserGroup Method (scentities)	58
sccadminctrl	58
setProjectState Method (sccadminctrl)	59
sventities	59
ActionSetting Class	61
ClientSideExecutionSet Class	62
Condition Class	63
CustomIncident Class	65
ExecutionLogEntry Class	66
ExecutionSet Class	69
Incident Class	71
Measure Class	72
Monitor Class	73
Rule Class	75
RuleExpressionPart Class	77
RuleIncident Class	78
Server Class	79
ServerSideExecutionSet Class	80
Transaction Class	80
addRuleExpressionPart Method (sventities)	80
createCondition Method (sventities)	81
createCustomIncident Method (sventities)	81
createDefaultRule Method (sventities)	82
createRule Method (sventities)	82
createRuleWithEssentialAction Method (sventities)	83

deleteCondition Method (sventities)	83
deleteCustomIncident Method (sventities)	84
deleteRule Method (sventities)	84
deleteRuleExpressionPart Method (sventities)	85
getActionEssentials Method (sventities)	85
getActionSettings Method (sventities)	85
getActionTypeIds Method (sventities)	86
getAlertingLocations Method (sventities)	86
getClientSideExecutionSets Method (sventities)	86
getCondition Method (sventities)	87
getConditions Method (sventities)	87
getCustomIncident Method (sventities)	88
getCustomIncidents Method (sventities)	88
getCustomIncidents Method (sventities)	88
getExecutionLog Method (sventities)	89
getMeasureNamesForProject Method (sventities)	90
getMeasures Method (sventities)	90
getMonitors Method (sventities)	91
getPageURL Method (sventities)	91
getPageURL Method (sventities)	92
getProfiles Method (sventities)	95
getResultLocations Method (sventities)	95
getRule Method (sventities)	96
getRuleExpressionParts Method (sventities)	96
getRules Method (sventities)	96
getRuleIncidents Method (sventities)	97
getRuleIncidents Method (sventities)	97
getServerSideExecutionSets Method (sventities)	98
getServersForProject Method (sventities)	98
getTransactionsForProject Method (sventities)	98
getTransactionsForMonitor Method (sventities)	99
moveRuleExpressionPart Method (sventities)	99
updateActionSettings Method (sventities)	100
updateCondition (sventities)	100
updateCustomIncident Method (sventities)	100
updateEssentialActionSettings Method (sventities)	101
updateRule Method (sventities)	101
updateRuleExpressionPart	102
svdata	102
DataPoint Class	103
getClientMeasureData Method (svdata)	106
getClientMeasureDataAcrossProjects Method (svdata)	107
getLastClientData Method (svdata)	108
getLastData Method (svdata)	109
getLastServerData Method (svdata)	109
getMeasureData Method (svdata)	110
getServerMeasureData Method (svdata)	111
getServerMeasureDataAcrossProjects Method (svdata)	112
mergeSeries Method (svdata)	113
svmonconfctrl	113
MonitorSetting Class	114
MonitorSettingSpec Class	115
RunExclusion Class	118
Schedule Class	120
ScheduleSettings Class	123
addScheduleExclusion Method (svmonconfctrl)	123

createMonitor Method (svmonconfctrl)	124
createMonitor Method (svmonconfctrl)	124
createMonitor Method (svmonconfctrl)	125
deleteMonitor Method (svmonconfctrl)	126
getLocationsOfSchedule Method (svmonconfctrl)	126
getMonitorSchedule Method (svmonconfctrl)	126
getMonitorSettings Method (svmonconfctrl)	127
getMonitorSettingSpec Method (svmonconfctrl)	127
getProjectSchedule Method (svmonconfctrl)	128
getScheduleExclusions Method (svmonconfctrl)	128
getTestScripts Method (svmonconfctrl)	128
getUsergroupNames Method (svmonconfctrl)	129
makeScheduleLocal Method (svmonconfctrl)	129
redefineSchedule Method (svmonconfctrl)	130
removeSchedule Method (svmonconfctrl)	130
runMonitor Method (svmonconfctrl)	130
runMonitorsInProject Method (svmonconfctrl)	131
setLocationsOfSchedule Method (svmonconfctrl)	131
setMonitorState Method (svmonconfctrl)	132
updateMonitorSettings Method (svmonconfctrl)	132
Accessing Web Services through Java	133
Generating Stub Classes	133
Instantiating Stub Classes and Invoking Methods	133
Sample Transaction	134

Introduction

In addition to offering access through HTML Web pages, Silk Performance Manager provides web services as an additional option for querying data and applying configuration changes.

Web services are available through SOAP calls, which are standardized XML messages that are sent and received through a front-end server using the HTTP protocol, allowing SOAP-enabled clients to get fast, direct access to Silk Performance Manager data and configuration settings.

Setting Up Web Services

Web services do not require setup. They are enabled by default on each front-end server. For example, if `http://www.yourFrontend.com/login` is the URL that you use to access Silk Performance Manager, then `http://www.yourFrontend.com/services` is the base URL you use to access available web services.

When you access the base URL using your browser, you are presented with a simple HTML list of all available web services. This list is provided by Apache Axis, which is the SOAP stack that Silk Performance Manager uses. For additional information, see <http://ws.apache.org/axis/>. This web page provides links to Web Service Description Language (WSDL) standardized XML files, where each file describes the interface of a single web service. These files are not human readable. For this reason, SOAP-enabled clients read WSDL files and thereby retrieve information required for invoking methods on corresponding web services.

Testing Your Configuration

In general, SOAP clients use HTTP POST requests for method invocation of web services. However, when method calls are not overly complex, you can also use GET requests for invocation.

You can test your Silk Performance Manager web service configuration directly from your browser. Browsers generally use HTTP GET requests.

The simple list of web services contains the service `Version`, which is an Axis service that provides the single method `getVersion()`. You can invoke this method directly from your browser by using the following URL format:

```
http://www.yourFrontend.com/services/Version?method=getVersion
```

The response that you will receive consists of an XML file, which is not human readable and which contains the return value of the method as an Axis version information string similar to the following:
Apache Axis version: 1.1 Built on Jun 13, 2003 (09:19:43 EDT)

You can also call web service methods that take parameters from your browser. If your installation starts and is properly connected to a database, you can login to Silk Performance Manager through a web service call.

If you have not changed the default login credentials of your Silk Performance Manager installation, you can use the username "admin" and the password "admin". To login, use the method `logonUser` of the web service `sccsystem` as follows:

```
http://www.yourFrontend.com/services/sccsystem?  
method=logonUser&username=admin&plainPwd=admin
```

If the credentials that you have supplied are correct, the response contains a session identifier that can be used in subsequent calls. The response should contain a line that resembles the following:
<logonUserReturn xsi:type="xsd:long">4756169926993183070</logonUserReturn>

Web Service Session Handling

Silk Performance Manager data is protected against unauthorized access. You must provide your login credentials before data access is granted. This is true not only when working with the HTML front-end, but also when communicating with Silk Performance Manager through SOAP calls.

Therefore, the first step in querying data or applying configuration changes for Silk Performance Manager is authentication. If the authentication is successful, a user session is created that allows execution of subsequent operations in the context of that user login.

When accessing Silk Performance Manager through a web browser, the session information is not visible to the user. The browser uses cookies to handle the session information. In contrast to using Silk Performance Manager through HTML, SOAP calls must handle information manually.

Authentication through web services is done through the SOAP call `logonUser()` of the web service `sccsystem`. See [Testing Your Configuration](#) for an example of how to invoke this call directly from your browser. The method call returns a session identifier that references the session created on the server and at the same time is used as a key to access Silk Performance Manager in the context of this session.

Each subsequent SOAP call that requires a session for execution takes the session identifier as one of its parameters, checks its validity, and executes in the context of the corresponding session.

The following Java code sample shows simple access to Silk Performance Manager through web services and demonstrates use of the session identifier:

```
long sessionId = sccsystem.logonUser("admin", "admin");
Project[] projects = sccentities.getProjects(sessionId);
```

A Silk Performance Manager session that has been created through web services cannot be ended explicitly. Instead, sessions end automatically when they are no longer used. As soon as a session times out on a server, subsequent SOAP calls that attempt to use the session throw exceptions.

Web Services

Silk Performance Manager offers the following web services:

- [sccsystem](#)
- [sccentities](#)
- [sccadminctrl](#)
- [sventities](#)
- [svdata](#)
- [svmonconfctrl](#)

This section offers a description of each of these services and of the methods that are available with each service. It also lists the location of the WSDL document that corresponds to each service.

The descriptions use a pseudo-code syntax that is closely related to the Java syntax.

sccsystem

Description

This is the root service. It provides an authentication method and simple utility methods.

The WSDL file of the service is available at `/services/sccsystem?wsdl`.

Methods

Name	Description
logonUser	Logs on with given username and password. <code>long</code> .
encryptPassword	Takes a clear text password and returns the encrypted version. <code>String</code> .
convertToReadableTime	Converts the given time, which is the number of milliseconds since midnight, January 1, 1970 UTC, into a human readable string representation. <code>String</code> .
convertFromReadableTime	Converts the given date/time to the number of milliseconds since midnight, January 1, 1970 UTC. <code>long</code> .
getApplicationModules	Retrieves the list of installed application modules. AppModule .

AppModule Class

Description

The `AppModule` class represents an application module.

Inheritance Hierarchy

- [Entity](#)
 - [NamedEntity](#)
 - `AppModule`

Syntax

```
public class AppModule extends NamedEntity
```

Entity Class

Description

The `Entity` class represents an entity.

Syntax

```
public class Entity implements Serializable
```

Properties

Name	Description
<code>id</code>	Gets or sets the identifier of the entity. <code>int</code> .

id Property (Entity)

Class

`Entity`.

Action

Returns the unique identifier.

Syntax

```
public int id;
```

Access

Read and write.

NamedEntity Class

Description

The `NamedEntity` class represents a named entity.

Inheritance Hierarchy

- `Entity`
 - `NamedEntity`

Syntax

```
public class NamedEntity extends Entity
```

Properties

Name	Description
<code>name</code>	Gets or sets the name of the entity. <code>String</code> .

Name	Description
description	Gets or sets the description of the entity. <code>String</code> .

name Property

Class

NamedEntity.

Action

Gets or sets the name.

Syntax

```
public String name;
```

Access

Read and write.

description Property

Class

NamedEntity.

Action

Returns the description.

Syntax

```
public String description;
```

Access

Read and write.

convertFromReadableTime

Class

[scssystem](#).

Action

Converts a given time string into the number of milliseconds since midnight, January 1, 1970 UTC.

Syntax

```
unreadableTime = scssystem.convertFromReadableTime(readableTime, plainPwd)
```

Variable	Description
<code>unreadableTime</code>	The time as the number of milliseconds since midnight, January 1, 1970 UTC. <code>long</code> .

Variable	Description
<i>readableTime</i>	The time in the readable format "YYYY-MM-DD HH:MM:SS.MMM". Based on UTC. <i>String</i> .

convertToReadableTime Method (sccsystem)

Class

[sccsystem](#).

Action

Converts a time given as the number of milliseconds since midnight, January 1, 1970 UTC, into a human-readable string representation. The returned format is "YYYY-MM-DD HH:MM:SS.MMM".

Syntax

```
readableTime = sccsystem.convertToReadableTime(unreadableTime, plainPwd)
```

Variable	Description
<i>readableTime</i>	The time in the readable format "YYYY-MM-DD HH:MM:SS.MMM". Based on UTC. <i>String</i> .
<i>unreadableTime</i>	The time as the number of milliseconds since midnight, January 1, 1970 UTC. <i>long</i> .

encryptPassword Method (sccsystem)

Class

[sccsystem](#).

Action

Returns the encrypted version of a password.

Syntax

```
password = sccsystem.encryptPassword(username, plainPwd)
```

Variable	Description
<i>password</i>	The encrypted password. <i>String</i> .
<i>sessionId</i>	The unique identifier of the actual session. <i>long</i> .
<i>plainPwd</i>	The plain text password. <i>String</i> .

Example

To encrypt the password *admin*, first retrieve the session ID, for example by using the [logonUser](#) method, and then call the `encryptPassword` method:

```
long sessionId = sccsystem.logonUser("admin", "admin");
string password = sccsystem.encryptPassword(sessionId, "admin");
```

getApplicationModules Method (AppModule)

Class

[AppModule](#).

Action

Retrieves the list of the application modules that are installed, for example Silk Performance Manager.

Syntax

```
applicationModules = getApplicationModules(sessionId)
```

Variable	Description
<i>applicationModules</i>	The list of the installed application modules. AppModule[].
<i>sessionId</i>	The unique ID of the actual session. long.

logonUser Method (scssystem)

Class

[scssystem](#).

Action

Adds a column to the underlying *ActiveData* object and returns a value that indicates success or failure. Logs on with a given username and password. The returned session identifier can be used in subsequent calls. It acts as a unique key. The session created with the method expires when it is not used for a period of time.

Syntax

```
sessionId = scssystem.logonUser(username, plainPwd)
```

Variable	Description
<i>sessionId</i>	A unique identifier that is used as a key to identify the session that is created with the method. long.
<i>username</i>	The username. String.
<i>plainPwd</i>	The password of the specified user. String.

Example

To create a new session for the user *admin* with the password *admin*, type:

```
long sessionId = scssystem.logonUser("admin", "admin");
```

scentities

Description

This service provides read access to the two main entities that exist on the SCA level—Project and Location.

The WSDL file of the service is available at `/services/scentities?wsdl`.

Methods

Name	Description
<i>getProjects</i>	Retrieves all defined projects. <i>Project[]</i> .
<i>getProjectsForUser</i>	Retrieves all defined projects that meet the specified filter criteria. <i>Project[]</i> .
<i>getLocations</i>	Retrieves all defined locations. <i>Location[]</i> .
<i>getLocationsForProject</i>	Retrieves all locations associated with the given project. <i>Location[]</i> .
<i>activateExecServer</i>	Activates an existing execution server if it is not already active.
<i>activateProjects</i>	Activates the projects with the given Ids.
<i>addLocationToProject</i>	Adds an association from a location to a project. <code>boolean</code> .
<i>addResourceTag</i>	Adds resource tags to an execution server.
<i>createExecServer</i>	Creates a new execution server for the given properties. <code>int</code> .
<i>createLocation</i>	Creates a new location with the data provided by the location details. <code>int</code> .
<i>createProject</i>	Creates a new project with the given properties. <code>int</code> .
<i>createUser</i>	Creates a new user with the data of the given user details - the given session ID is ignored and the login has to be unique. <code>int</code> .
<i>createUsergroup</i>	Creates a new user group. <code>int</code> .
<i>deactivateExecServer</i>	Deactivates an existing execution server if it is active.
<i>deactivateProjects</i>	Deactivates the projects with the given identifiers.
<i>deleteLocations</i>	Deletes the locations with the specified location identifiers, if no execution servers exist for the locations.
<i>deleteProjects</i>	Deletes the projects with the specified identifiers if the logged in user has the right to delete projects.
<i>deleteUsers</i>	Deletes the users with the specified identifiers.
<i>deleteUserGroup</i>	Deletes the user group identified by the ID.
<i>encryptAndSetPassword</i>	Sets the password for the user. The password is encrypted on the server and therefore is transmitted in plain text.
<i>getAllEssentials</i>	Queries all essentials. <i>EssentialDescription[]</i> .
<i>getAllMemberships</i>	Retrieves all memberships. <i>UserGroupMembership[]</i> .

Name	Description
<i>getAllProjects</i>	Retrieves all projects with the given name that are assigned to the logged in user. Alternatively, retrieves all projects with the given name if the user is an administrator. Supports the wildcard characters "*" and "?". <i>Project[]</i> .
<i>getAllResourceTags</i>	Retrieves all resource tags of a location. <i>String[]</i> .
<i>getAllRoles</i>	Retrieves all roles. <i>Role[]</i> .
<i>getAllUserGroups</i>	Retrieves all user groups. <i>UserGroup[]</i> .
<i>getEssentialGroups</i>	Queries all essential groups. <i>EssentialGroup[]</i> .
<i>getEssentials</i>	Queries all essentials which are children of the given subgroup. <i>EssentialDescription[]</i> .
<i>getEssentialSubGroups</i>	Queries all Essential Sub Groups. <i>EssentialSubGroup[]</i> .
<i>getExecServerById</i>	Retrieves the execution server that is identified by the given ID. <i>ExecServer</i> .
<i>getExecServersOfLocation</i>	Retrieves the execution servers of a location. <i>ExecServer[]</i> .
<i>getFilePoolEntries</i>	Retrieves information about the file pool entries. <i>FilePoolEntry[]</i> .
<i>getGroupById</i>	Retrieves a user group by ID. <i>UserGroup</i> .
<i>getGroupByName</i>	Retrieves a user group by name. <i>UserGroup</i> .
<i>getLocationDetails</i>	Retrieves detailed information about the location specified by the ID. <i>LocationDetails</i> .
<i>getMembershipsOfGroup</i>	Retrieves all memberships of the specified user group. <i>UserGroupMembership[]</i> .
<i>getMembershipsOfUser</i>	Retrieves all user group memberships of the specified user. <i>UserGroupMembership[]</i> .
<i>getProjectById</i>	Retrieves the project with the given ID. <i>Project</i> .
<i>getRoleById</i>	Retrieves the specified role. <i>Role</i> .
<i>getUserDetails</i>	Retrieves details of the user with the given ID. <i>UserDetails</i> .
<i>getUsers</i>	Retrieves all users matching the given login name. <i>User[]</i> .
<i>isMixedModeAuthentication</i>	Retrieves whether mixed mode authentication is active for a user. <i>boolean</i> .
<i>removeLocationFromProject</i>	Removes an association from a location to a project. <i>boolean</i> .
<i>removeResourceTag</i>	Removes resource tags from an execution server.
<i>setMixedModeAuthentication</i>	Sets the authentication mode for a user.
<i>setPassword</i>	Sets the password for the user. The password has to be encrypted on the client. This method is for internal use only.
<i>setProxyUsernameAndPasswordForLocation</i>	Sets the proxy username and the password for a location identified by the <i>locationId</i> .
<i>updateExecServer</i>	Updates the properties of an existing execution server.

Name	Description
updateLocation	Updates a location with the data that is provided by the location details.
updateMembershipsOfGroup	Updates the memberships of a group.
updateMembershipsOfUser	Updates the user group memberships of a user.
updateProject	Updates an existing project with the given properties.
updateUser	Updates the properties of an existing user.
updateUserGroup	Updates an existing user group.

Project Class

Description

The `Project` class represents a project.

Inheritance Hierarchy

- [Entity](#)
 - [NamedEntity](#)
 - Project

Syntax

```
public class Project extends NamedEntity
```

Properties

Name	Description
ID	The unique identifier of the project. <code>int</code> .
name	The name of the project. <code>String</code> .
description	The description of the project. <code>String</code> .
active	Indicates whether the project is active or not. <code>boolean</code> .
appModuleIds	The application module IDs. <code>int[]</code> .

active Property (Project)

Class

[Project](#).

Action

Gets or sets whether the project is active or inactive.

Syntax

```
public boolean active;
```

Access

Read and write.

appModuleIds Property (Project)

Class

[Project](#).

Action

Gets or sets the application module IDs.

Syntax

```
protected int[] appModuleIds;
```

Access

Read and write.

Location Class

Description

The `Location` class represents a location.

Inheritance Hierarchy

- [Entity](#)
 - [NamedEntity](#)
 - `Location`

Syntax

```
public class Location extends NamedEntity
```

Properties

Name	Description
ID	The unique identifier of the location. <code>int</code> .
name	The name of the location. <code>String</code> .
isActive	Indicates whether at least one execution server of this location is active. <code>boolean</code> .

isActive property (Location)

Class

[Location](#).

Action

Indicates whether at least one execution server of this location is active.

Syntax

```
public boolean isActive;
```

Access

Read and write.

LocationDetails Class

Description

The `Location` class represents a location.

Inheritance Hierarchy

- *Entity*
 - *NamedEntity*
 - *Location*
 - `LocationDetails`

Syntax

```
public class LocationDetails extends Location
```

Properties

Name	Description
<i>proxyHostName</i>	Gets or sets the hostname of the proxy server. <code>String</code> .
<i>proxyPort</i>	Gets or sets the port of the proxy server. <code>int</code> .
<i>proxyType</i>	The type of the proxy server: <ul style="list-style-type: none">• 0: No proxy.• 1: Http proxy.• 2: Socks proxy. <code>int</code> .
<i>useCentralProxy</i>	Indicates whether the central proxy shall be used or not. <code>boolean</code> .
<i>isReal</i>	<i>Deprecated</i> : Should always be false. Used to indicate that this location is a real location that was created by a third party tool.

proxyHostName Property (LocationDetails)

Class

LocationDetails.

Action

Gets or sets the hostname of the proxy server.

Syntax

```
public String proxyHostName;
```

Access

Read and write.

proxyPort Property (LocationDetails)

Class

[LocationDetails](#).

Action

Gets or sets the port of the proxy server.

Syntax

```
public int proxyPort;
```

Access

Read and write.

proxyType Property (LocationDetails)

Class

[LocationDetails](#).

Action

The type of the proxy server:

- 0: No proxy.
- 1: Http proxy.
- 2: Socks proxy.

Syntax

```
public int proxyType;
```

Access

Read and write.

useCentralProxy Property (LocationDetails)

Class

[LocationDetails](#).

Action

Indicates whether the central proxy shall be used or not.

Syntax

```
public boolean useCentralProxy;
```

Access

Read and write.

isReal Property (LocationDetails)

Class

[LocationDetails](#).

Action

Deprecated: Should always be false. Used to indicate that this location is a real location that was created by a third party tool.

Syntax

```
public boolean isReal;
```

Access

Read and write.

EssentialDescription Class

Description

The `EssentialDescription` class provides information about an essential.

Inheritance Hierarchy

- [Entity](#)
 - [NamedEntity](#)
 - `EssentialDescription`

Syntax

```
public class EssentialDescription extends NamedEntity
```

Properties

Name	Description
fullPathName	The name of the essential. Includes the full path to the essential. String.

fullPathName Property (EssentialDescription)

Class

[EssentialDescription](#).

Action

The name of and the full path to the essential.

Syntax

```
public String fullPathName;
```

Access

Read and write.

EssentialGroup Class

Description

The `EssentialGroup` class represents an essential group.

Inheritance Hierarchy

- [Entity](#)
 - [NamedEntity](#)
 - `EssentialGroup`

Syntax

```
public class EssentialGroup extends NamedEntity
```

EssentialSubGroup Class

Description

The `EssentialSubGroup` class represents a sub-group of an essentials group..

Inheritance Hierarchy

- [Entity](#)
 - [NamedEntity](#)
 - `EssentialSubGroup`

Syntax

```
public class EssentialSubGroup extends NamedEntity
```

ExecServer Class

Description

The `ExecServer` class represents an execution server.

Inheritance Hierarchy

- [Entity](#)
 - [NamedEntity](#)
 - `ExecServer`

Syntax

```
public class ExecServer extends NamedEntity
```

Properties

Name	Description
active	Indicates whether this execution server is activated or not. <code>boolean</code> .
appModuleID	Gets or sets the application server module ID. <code>int</code> .
execServerHostName	The hostname of this execution server. <code>String</code> .
locationId	The ID of the location to which the execution server is assigned. <code>int</code> .
maxBandwidth	The maximum bandwidth. <code>int</code> .
maxConcurrentExecutions	The maximal number of concurrent executions that are possible on this execution server. <code>int</code> .
problemAlertTimeout	Responsiveness timeout. <code>int</code> .
rmiProxyPort	Java RMI port of this execution server. <code>int</code> .
rmiProxyPortSSLJava	RMI port of this execution server for SSL. <code>int</code> .
state	Current state of this execution server. The following state are allowed: <ul style="list-style-type: none">• 0: inactive.• 1: active.• 2: failure.
useSSL	Indicates whether SSL is used for connecting this execution server or not. <code>boolean</code> .

active Property (ExecServer)

Class

[ExecServer](#).

Action

Indicates whether this execution server is activated or not.

Syntax

```
public boolean active;
```

Access

Read and write.

appModuleID Property (ExecServer)

Class

[ExecServer](#).

Action

Gets or sets the application server module ID. The value is 1 for Silk Performance Manager.

Syntax

```
public int appModuleID;
```

Access

Read and write.

execServerHostName Property (ExecServer)**Class**

[ExecServer](#).

Action

Gets or sets the hostname of this execution server.

Syntax

```
public String execServerHostName;
```

Access

Read and write.

locationId Property (ExecServer)**Class**

[ExecServer](#).

Action

Gets or sets the ID of the location the execution server is assigned to.

Syntax

```
public int locationId;
```

Access

Read and write.

maxBandwidth Property (ExecServer)**Class**

[ExecServer](#).

Action

Gets or sets the maximum bandwidth of the execution server.

Syntax

```
public int maxBandwidth;
```

Access

Read and write.

maxConcurrentExecutions Property (ExecServer)

Class

[ExecServer](#).

Action

Gets or sets the maximum number of concurrent executions that are allowed on this execution server.

Syntax

```
public int maxConcurrentExecutions;
```

Access

Read and write.

problemAlertTimeout Property (ExecServer)

Class

[ExecServer](#).

Action

Gets or sets the responsiveness timeout.

Syntax

```
public int problemAlertTimeout;
```

Access

Read and write.

rmiProxyPort Property (ExecServer)

Class

[ExecServer](#).

Action

Gets or sets the Java RMI port of an execution server.

Syntax

```
public int rmiProxyPort;
```

Access

Read and write.

rmiProxyPortSSL Property (ExecServer)

Class

[ExecServer](#).

Action

Gets or sets the Java RMI port of an execution server for SSL.

Syntax

```
public int rmiProxyPortSSL;
```

Access

Read and write.

state Property (ExecServer)

Class

[ExecServer](#).

Action

Gets or sets the current state of this execution server. The following states are available:

- 0: inactive
- 1: active
- 2: failure

Syntax

```
public int state;
```

Access

Read and write.

useSSL Property (ExecServer)

Class

[ExecServer](#).

Action

Gets or sets whether SSL is used for connecting this execution server or not.

Syntax

```
public boolean useSSL;
```

Access

Read and write.

FilePoolEntry Class

Description

The `FilePoolEntry` class represents a file pool entry.

Inheritance Hierarchy

- *Entity*
 - *NamedEntity*
 - FilePoolEntry

Syntax

```
public class FilePoolEntry extends NamedEntity
```

Properties

Name	Description
<i>name</i>	Gets or sets the name of the file pool entry. <i>String</i> .
<i>description</i>	Gets or sets the description of the file pool entry. <i>String</i> .
<i>id</i>	Gets or sets the unique identifier of the file pool entry. <i>int</i> .
<i>fullpathname</i>	Gets or sets the full path to the file pool entry. <i>String</i> .
<i>isSilkTestPackage</i>	Gets whether the file pool entry belongs to the Silk Test package. <i>boolean</i> .

fullPathName Property (FilePoolEntry)

Class

FilePoolEntry.

Action

Gets or sets the full path to the file pool entry.

Syntax

```
public String fullPathName;
```

Access

Read and write.

isSilkTestPackage Property (FilePoolEntry)

Class

FilePoolEntry.

Action

Gets whether the file pool entry belongs to the Silk Test package.

Syntax

```
private String isSilkTestPackage;
```

Access

Read only.

User Class

Description

The `User` class represents a user.

Inheritance Hierarchy

- `Entity`
 - `NamedEntity`
 - `User`

Syntax

```
public class User extends NamedEntity implements Serializable
```

Properties

Name	Description
<code>ID</code>	The unique identifier of the user. <code>int</code> .
<code>name</code>	The last name of the user. <code>String</code> .
<code>description</code>	The description of the user. <code>String</code> .
<code>firstName</code>	The first name of the user. <code>String</code> .
<code>isLocked</code>	Indicates whether the user is locked or not. <code>boolean</code> .
<code>login</code>	The login name of the user. <code>String</code> .

firstName Property (User)

Class

`User`.

Action

Gets or sets the first name of the user.

Syntax

```
public String firstName;
```

Access

Read and write.

isLocked Property (User)

Class

`User`.

Action

Indicates whether the user is locked or not.

Syntax

```
public boolean isLocked;
```

Access

Read and write.

login Property (User)

Class

[User](#).

Action

Gets or sets the login name of the user.

Syntax

```
public String login;
```

Access

Read and write.

UserDetails Class

Description

The `UserDetails` class provides detailed information about a user.

Inheritance Hierarchy

- [Entity](#)
 - [NamedEntity](#)
 - [User](#)
 - `UserDetails`

Syntax

```
public class UserDetails extends User
```

Properties

Name	Description
dateFormat	Gets or sets the date format. <code>String</code> .
email	Gets or sets the email address for notifications. <code>String</code> .
firstDayOfWeek	Gets or sets the first day of week. <code>int</code> .
pageRefreshTime	Gets or sets the delay between automated page refreshes. <code>int</code> .

Name	Description
roleId	Gets or sets the role ID of the user. <code>int</code> .
separatorString	Gets or sets the preferred separator string of the user. <code>String</code> .
shortDateFormat	Gets or sets the short date format. <code>String</code> .
timeZoneId	Gets or sets the time zone ID. <code>String</code> .
assignedProjects	Gets the ids of the projects that are assigned to the user. <code>int[]</code> .

dateFormat Property (UserDetails)

Class

[UserDetails](#).

Action

Gets or sets the date format for the user. For example, "EEE, MMM dd, yyyy - h:mm:ss a".

Syntax

```
public String dateFormat;
```

Access

Read and write.

eMail Property (UserDetails)

Class

[UserDetails](#).

Action

Gets or sets the email address for notifications.

Syntax

```
public String eMail;
```

Access

Read and write.

firstDayOfWeek Property (UserDetails)

Class

[UserDetails](#).

Action

Gets or sets first day of week for a user.

Syntax

```
public int firstDayOfWeek;
```

Access

Read and write.

pageRefreshTime Property (UserDetails)

Class

[UserDetails](#).

Action

Gets or sets the delay between automated page refreshes.

Syntax

```
public int pageRefreshTime;
```

Access

Read and write.

roleId Property (UserDetails)

Class

[UserDetails](#).

Action

Gets or sets the role ID of the user. The following values are allowed:

ID	Role
0	SuperUser
1	Administrator
2	Project Manager
3	Analyst
6	Reporter

Syntax

```
public int roleId;
```

Access

Read and write.

separatorString Property (UserDetails)

Class

[UserDetails](#).

Action

Gets or sets the preferred separator string of the user. For example, "," or ";".

Syntax

```
public String separatorString;
```

Access

Read and write.

shortDateFormat Property (UserDetails)**Class**

[UserDetails](#).

Action

Gets or sets the short date format. For example, "M/d/yyyy, h:mm a".

Syntax

```
public String shortDateFormat;
```

Access

Read and write.

timeZoneId Property (UserDetails)**Class**

[UserDetails](#).

Action

Gets or sets the time zone ID for the user in the format <continent> + "/" + <city>. For example

Syntax

```
public String timeZoneId;
```

Access

Read and write.

assignedProjects Property (UserDetails)**Class**

[UserDetails](#).

Action

Gets the IDs of the projects that are assigned to the user.

Syntax

```
protected int[] assignedProjects;
```

Access

Read only.

UserGroup Class

Description

The `UserGroup` class represents a user group.

Syntax

```
public class UserGroup
```

Properties

Name	Description
<i>id</i>	The ID of the user group. <code>int</code> .
<i>name</i>	The name of the user group. <code>String</code> .
<i>isLocked</i>	Indicates whether the user group is locked or not. <code>boolean</code> .
<i>description</i>	The description of the user group. <code>String</code> .
<i>createdAt</i>	The timestamp at which the user group was created. <code>long</code> .
<i>createdBy</i>	The name of the user who created the user group. <code>String</code> .
<i>projectAssignments</i>	The IDs of the projects that are assigned to the user group. <code>int[]</code> .

id Property (UserGroup)

Class

UserGroup.

Action

Gets the ID of the user group.

Syntax

```
protected int id;
```

Access

Read and write.

name Property (UserGroup)

Class

UserGroup.

Action

The name of the user group.

Syntax

```
protected String name;
```

Access

Read and write.

isLocked Property (UserGroup)**Class**

UserGroup.

Action

Indicates whether the user group is locked or not.

Syntax

```
protected boolean isLocked;
```

Access

Read and write.

description Property (UserGroup)**Class**

UserGroup.

Action

The description of the user group.

Syntax

```
protected String description;
```

Access

Read and write.

createdAt Property (UserGroup)**Class**

UserGroup.

Action

The timestamp at which the user group was created.

Syntax

```
protected long createdAt;
```

Access

Read only.

createdBy Property (UserGroup)

Class

[UserGroup](#).

Action

The name of the user who created the user group.

Syntax

```
protected String createdBy;
```

Access

Read only.

projectAssignments Property (UserGroup)

Class

[UserGroup](#).

Action

The IDs of the projects that are assigned to the user group.

Syntax

```
protected int[] projectAssignments;
```

Access

Read and write.

UserGroupMembership Class

Description

The `UserGroup` class represents a membership in a user group.

Syntax

```
public class UserGroupMembership
```

Properties

Name	Description
userId	The ID of the user group membership. <code>int</code> .
groupId	The ID of the user group. <code>int</code> .
roleId	The ID of the role. <code>int</code> .
groupName	The name of the user group. <code>String</code> .
roleName	The name of the role. <code>String</code> .

userId Property (UserGroupMembership)

Class

[UserGroupMembership](#).

Action

Gets or sets the ID of the user group member.

Syntax

```
protected int userId;
```

Access

Read and write.

groupId Property (UserGroupMembership)

Class

[UserGroupMembership](#).

Action

Gets or sets the ID of the user group.

Syntax

```
protected int groupId;
```

Access

Read and write.

roleId Property (UserGroupMembership)

Class

[UserGroupMembership](#).

Action

Gets or sets the ID of the role.

Syntax

```
protected int roleId;
```

Access

Read and write.

groupName Property (UserGroupMembership)

Class

[UserGroupMembership](#).

Action

The name of the group.

Syntax

```
protected String groupName;
```

Access

Read only.

roleName Property (UserGroupMembership)

Class

[UserGroupMembership](#).

Action

The name of the role.

Syntax

```
protected String roleName;
```

Access

Read only.

Role Class

Description

The `Role` class represents a user role.

Syntax

```
public class Role
```

Properties

Name	Description
id	The unique identifier of the role. <code>int</code> .
name	The name of the role. <code>String</code> .

id Property (Role)

Class

[Role](#).

Action

Gets or sets the unique identifier of the role.

Syntax

```
protected int id;
```

Access

Read and write.

name Property (Role)

Class

[Role](#).

Action

Gets or sets the name of the role.

Syntax

```
protected String name;
```

Access

Read and write.

getAllRoles Method (scentities)

Class

[scentities](#).

Action

Retrieves all roles.

Syntax

```
roles = scentities.getAllRoles(sessionId)
```

Variable	Description
<i>roles</i>	The roles. Role[] .
<i>sessionId</i>	The session identifier. long.

getAllUserGroups Method (scentities)

Class

[scentities](#).

Action

Retrieves all user groups.

Syntax

```
userGroups = scentities.getAllUserGroups(sessionId)
```

Variable	Description
<i>userGroups</i>	The user groups. UserGroup[] .
<i>sessionId</i>	The session identifier. long.

getProjects Method (scentities)

Class

[scentities](#).

Action

Retrieves all projects that are defined for a session.

Syntax

```
projectList = scentities.getProjects(sessionId)
```

Variable	Description
<i>projectList</i>	A list of the projects that are defined for the given session. Project[] .
<i>sessionId</i>	The session identifier. long.

getProjectsForUser Method (scentities)

Class

[scentities](#).

Action

Retrieves all projects that are defined for a user in a session.

Syntax

```
projectList = scentities.getProjectsForUser(sessionId, UserId, appModuleId)
```

Variable	Description
<i>projectList</i>	A list of the projects that are defined for the given session and user. Project[] .
<i>sessionId</i>	The session identifier. long.
<i>UserId</i>	The user identifier. int.
<i>appModuleId</i>	The identifier of the application module. int.

getLocations Method (scentities)

Class

[scentities](#).

Action

Retrieves all locations that are defined for a session.

Syntax

```
locationList = sccentities.getLocations(sessionId)
```

Variable	Description
<i>locationList</i>	A list of the locations that are defined for the given session. <i>Location[]</i> .
<i>sessionId</i>	The session identifier. long.

getLocationsForProject Method (sccentities)

Class

sccentities.

Action

Retrieves all locations associated with the given project.

Syntax

```
locationList = sccentities.getLocationsForProject(sessionId, projectId)
```

Variable	Description
<i>locationList</i>	A list of the locations that are associated with the given project. <i>Location[]</i> .
<i>sessionId</i>	The session identifier. long.
<i>projectId</i>	The project identifier. int.

activateExecServer Method (sccentities)

Class

sccentities.

Action

Activates an existing execution server if it is not already active.

Syntax

```
sccentities.activateExecServer(sessionId, execServerId, locationId)
```

Variable	Description
<i>sessionId</i>	The session identifier. long.
<i>execServerId</i>	The execution server identifier. int.
<i>locationId</i>	The location identifier. int.

activateProjects Method (scentities)

Class

scentities.

Action

Activates the specified projects.

Syntax

```
scentities.activateProjects(sessionId, projectIds)
```

Variable	Description
<i>sessionId</i>	The session identifier. long.
<i>projectIds</i>	The project identifiers. int[].

addLocationToProject Method (scentities)

Class

scentities.

Action

Adds an association from a location to a project.

Syntax

```
success = scentities.addLocationToProject(sessionId, projectId, locationId)
```

Variable	Description
<i>success</i>	Whether the association was added or not. boolean.
<i>sessionId</i>	The session identifier. long.
<i>projectId</i>	The project identifier. int.
<i>locationId</i>	The location identifier. int.

addResourceTag Method (scentities)

Class

scentities.

Action

Adds a resource tag to an execution server.

Syntax

```
scentities.addResourceTag(sessionId, locationId, execServerId, resourceTag)
```

Variable	Description
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>locationId</i>	The location identifier. <code>int</code> .
<i>execServerId</i>	The execution server identifier. <code>int</code> .
<i>resourceTag</i>	The resource tag to add to the execution server. <code>String</code> .

createExecServer Method (scentities)

Class

[scentities](#).

Action

Creates a new execution server with the given properties.

Syntax

```
execServerId = scentities.createExecServer(sessionId, execServer)
```

Variable	Description
<i>execServerId</i>	The identifier for the new execution server. <code>int</code> .
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>execServer</i>	The execution server. ExecServer .

createLocation Method (scentities)

Class

[scentities](#).

Action

Retrieves all locations associated with the given project.

Syntax

```
locationId = scentities.createLocation(sessionId, locationDetails)
```

Variable	Description
<i>locationId</i>	The identifier for the new location. <code>int</code> .
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>locationDetails</i>	The details for the new location. LocationDetails .

createProject Method (scentities)

Class

[scentities](#).

Action

Creates a new project with the given properties.

Syntax

```
projectId = scentities.createProject(sessionId, project)
```

Variable	Description
<i>projectId</i>	The identifier for the new project. int.
<i>sessionId</i>	The session identifier. long.
<i>project</i>	The project. Project .

createUser Method (scentities)

Class

[scentities](#).

Action

Creates a new user with the given user details. The given session identifier is ignored and the login has to be unique.

Syntax

```
userId = scentities.createUser(sessionId, userDetails)
```

Variable	Description
<i>userId</i>	The identifier for the new user. int.
<i>sessionId</i>	The session identifier. long.
<i>userDetails</i>	The detailed information for the new user. UserDetails .

createUsergroup Method

Class

[scentities](#).

Action

Creates a new user group.

Syntax

```
userGroupId = scentities.createUsergroup(sessionId, userGroup)
```

Variable	Description
<i>userGroupId</i>	The user group identifier. int.
<i>sessionId</i>	The session identifier. long.

Variable	Description
<i>userGroup</i>	The new user group. UserGroup .

deactivateExecServer Method (scentities)

Class

[scentities](#).

Action

Deactivates an existing execution server if it is active.

Syntax

```
scentities.deactivateExecServer(sessionId, execServerId)
```

Variable	Description
<i>sessionId</i>	The session identifier. long.
<i>execServer</i>	The identifier of the execution server. int.

deactivateProjects Method (scentities)

Class

[scentities](#).

Action

Deactivates the projects with the given IDs.

Syntax

```
scentities.deactivateProjects(sessionId, projectIds)
```

Variable	Description
<i>sessionId</i>	The session identifier. long.
<i>projectIds</i>	The identifiers of the projects to be deactivated. int[].

deleteLocations Method (scentities)

Class

[scentities](#).

Action

Deletes the locations specified by the location identifiers, if no execution servers exist for the locations.

Syntax

```
scentities.deleteLocations(sessionId, locationIds, locationId)
```

Variable	Description
<i>sessionId</i>	The session identifier. long.
<i>locationIds</i>	The location identifiers. int[].

deleteProjects Method (scentities)

Class

scentities.

Action

Deletes the projects that are specified with the given identifiers, if the logged in user has the right to delete projects.

Syntax

```
scentities.deleteProjects(sessionId, projectIds, locationId)
```

Variable	Description
<i>sessionId</i>	The session identifier. long.
<i>projectIds</i>	The project identifiers. int[].

deleteUsers Method (scentities)

Class

scentities.

Action

Deletes the specified users.

Syntax

```
scentities.deleteUsers(sessionId, userIds)
```

Variable	Description
<i>sessionId</i>	The session identifier. long.
<i>userIds</i>	The identifiers of the users to be deleted. int[].

deleteUserGroup Method (scentities)

Class

scentities.

Action

Deletes the user group identified by the identifier.

Syntax

```
sccentities.deleteUserGroup(sessionId, groupId)
```

Variable	Description
<i>sessionId</i>	The session identifier. long.
<i>groupId</i>	The identifier of the user group that should be deleted. int.

encryptAndSetPassword Method (sccentities)

Class

[sccentities](#).

Action

Sets the password for the user. The password is encrypted on the server and therefore is transmitted in plain text.

Syntax

```
sccentities.encryptAndSetPassword(sessionId, userId, password)
```

Variable	Description
<i>sessionId</i>	The session identifier. long.
<i>userId</i>	The identifiers of the user. int.
<i>password</i>	The password in plain text. String.

getAllEssentials Method (sccentities)

Class

[sccentities](#).

Action

Queries all essentials.

Syntax

```
essentials=sccentities.getAllEssentials(sessionId)
```

Variable	Description
<i>essentials</i>	The essentials. EssentialDescription[] .
<i>sessionId</i>	The session identifier. long.

getAllMemberships Method (sccentities)

Class

[sccentities](#).

Action

Retrieves all memberships.

Syntax

```
memberships = scentities.getAllMemberships(sessionId)
```

Variable	Description
<i>memberships</i>	All memberships. UserGroupMembership .
<i>sessionId</i>	The session identifier. long.

getAllProjects Method (scentities)

Class

[scentities](#).

Action

Retrieves all projects with the specified name that are assigned to the logged in user, or all projects if the user is an administrator. Supports the wildcard characters * and ?.

Syntax

```
projects=scentities.getAllProjects(sessionId, projectName)
```

Variable	Description
<i>projects</i>	The projects that are assigned to the logged in user. Project[] .
<i>sessionId</i>	The session identifier. long.
<i>projectName</i>	The name of the project to retrieve. String.

getAllResourceTags Method (scentities)

Class

[scentities](#).

Action

Retrieves all resource tags of a location.

Syntax

```
resourceTags = scentities.getAllResourceTags(sessionId, locationId, execServerId)
```

Variable	Description
<i>resourceTags</i>	The resource tags. String[] .
<i>sessionId</i>	The session identifier. long.
<i>locationId</i>	The location identifier. int.

Variable	Description
<i>execServerId</i>	The execution server identifier. int.

getEssentialGroups Method (scentities)

Class

scentities.

Action

Queries all essential groups.

Syntax

```
essentialGroups = scentities.getEssentialGroups(sessionId, execServerId, locationId)
```

Variable	Description
<i>essentialGroups</i>	The essential group identifiers. <i>EssentialGroup[]</i> .
<i>sessionId</i>	The session identifier. long.

getEssentials Method (scentities)

Class

scentities.

Action

Queries all essentials that are children of the specified subgroup.

Syntax

```
essentials = scentities.getEssentials(sessionId, parentId)
```

Variable	Description
<i>essentials</i>	The essentials that are children of the given essential. <i>EssentialDescription[]</i> .
<i>sessionId</i>	The session identifier. long.
<i>parentId</i>	The identifier of the parent essential. int.

getEssentialSubGroups Method (scentities)

Class

scentities.

Action

Queries all sub-groups of the specified essential group.

Syntax

```
essentialSubGroups = sccentities.getEssentialSubGroups(sessionId, essentialGroupId)
```

Variable	Description
<i>essentialSubGroups</i>	The essential sub-group identifiers. EssentialSubGroup[] .
<i>sessionId</i>	The session identifier. long.
<i>essentialGroupId</i>	The essential group identifier. EssentialGroup[]

getExecServerById Method (sccentities)

Class

[sccentities](#).

Action

Retrieves the execution server that is identified by the given ID.

Syntax

```
execServer = sccentities.getExecServerById(sessionId, execServerId)
```

Variable	Description
<i>execServer</i>	The execution server. ExecServer .
<i>sessionId</i>	The session identifier. long.
<i>execServerId</i>	The execution server identifier. int.

getExecServersOfLocation Method (sccentities)

Class

[sccentities](#).

Action

Retrieves the execution servers of a location.

Syntax

```
execServers = sccentities.getExecServersOfLocation(sessionId, locationId)
```

Variable	Description
<i>execServers</i>	The execution servers of the specified location. ExecServer[] .
<i>sessionId</i>	The session identifier. long.
<i>locationId</i>	The identifier of the location. int.

getFilePoolEntries Methods (scentities)

Class

scentities.

Action

Retrieves information about the file pool entries.

Syntax

```
filePoolEntries = scentities.getFilePoolEntries(sessionId, execServerId)
```

Variable	Description
<i>filePoolEntries</i>	The file pool entries. <i>FilePoolEntry[]</i> .
<i>sessionId</i>	The session identifier. long.
<i>projectId</i>	The project identifier. int.

getGroupById Method (scentities)

Class

scentities.

Action

Retrieves a user group by ID.

Syntax

```
userGroup = scentities.getGroupById(sessionId, groupId)
```

Variable	Description
<i>userGroup</i>	The user group with the specified identifier. <i>UserGroup</i> .
<i>sessionId</i>	The session identifier. long.
<i>groupId</i>	The group identifier. int.

getGroupByName Method (scentities)

Class

scentities.

Action

Retrieves a user group by name.

Syntax

```
userGroup = scentities.getGroupByName(sessionId, groupName)
```

Variable	Description
<i>userGroup</i>	The user group with the specified name. UserGroup .
<i>sessionId</i>	The session identifier. long.
<i>groupName</i>	The group name. String.

getLocationDetails Method (scentities)

Class

[scentities](#).

Action

Retrieves detailed information about the location specified by the ID.

Syntax

```
locationDetails = scentities.getLocationDetails(sessionId, execServerId)
```

Variable	Description
<i>locationDetails</i>	The details of the specified location. LocationDetails .
<i>sessionId</i>	The session identifier. long.
<i>locationId</i>	The location. int.

getMembershipsOfGroup Method (scentities)

Class

[scentities](#).

Action

Retrieves all memberships of the specified user group.

Syntax

```
userGroupMemberships = scentities.getMembershipsOfGroup(sessionId, groupId)
```

Variable	Description
<i>userGroupMemberships</i>	The memberships of the user group. UserGroupMembership[] .
<i>sessionId</i>	The session identifier. long.
<i>groupId</i>	The group identifier. int.

getMembershipsOfUser Method (scentities)

Class

[scentities](#).

Action

Retrieves all user group memberships of the specified user.

Syntax

```
userGroupMemberships = sccentities.getMembershipsOfUser(sessionId, userId)
```

Variable	Description
<code>userGroupMemberships</code>	The user group memberships of the specified user. UserGroupMemberships[] .
<code>sessionId</code>	The session identifier. long.
<code>userId</code>	The user identifier. int.

getProjectById Method (sccentities)

Class

[sccentities](#).

Action

Retrieves the project with the given ID.

Syntax

```
project = sccentities.getProjectById(sessionId, execServerId)
```

Variable	Description
<code>project</code>	The project. Project .
<code>sessionId</code>	The session identifier. long.
<code>projectId</code>	The project identifier. int.

getRoleById Method (sccentities)

Class

[sccentities](#).

Action

Retrieves the specified role.

Syntax

```
role = sccentities.getRoleById(sessionId, roleId)
```

Variable	Description
<code>role</code>	The role with the specified identifier. Role .
<code>sessionId</code>	The session identifier. long.
<code>roleId</code>	The identifier of the role. int.

getUserDetails Method (scentities)

Class

scentities.

Action

Activates an existing execution server if it is not already active.

Syntax

```
userDetails = scentities.getUserDetails(sessionId, userId)
```

Variable	Description
<i>userDetails</i>	The user details of the specified user. <i>UserDetails</i> .
<i>sessionId</i>	The session identifier. long.
<i>userId</i>	The user identifier. int.

getUsers Method (scentities)

Class

scentities.

Action

Retrieves all users matching the given login name.

Syntax

```
users = scentities.getUsers(sessionId, login)
```

Variable	Description
<i>users</i>	All users with the specified login name. <i>User[]</i> .
<i>sessionId</i>	The session identifier. long.
<i>login</i>	The login name of the users to return. Allows the wildcard characters "?" and "**". String.

isMixedModeAuthentication Method (scentities)

Class

scentities.

Action

Retrieves whether mixed mode authentication is active for a user.

Syntax

```
isMixedModeAuthentication = scentities.isMixedModeAuthentication(sessionId,
userId)
```

Variable	Description
<i>isMixedModeAuthentication</i>	Whether mixed mode authentication is active for the specified user. boolean.
<i>sessionId</i>	The session identifier. long.
<i>userId</i>	The user identifier. int.

removeLocationFromProject Method (scentities)

Class

[scentities](#).

Action

Removes an association from a location to a project.

Syntax

```
isRemoved = scentities.removeLocationFromProject(sessionId, execServerId,
locationId)
```

Variable	Description
<i>isRemoved</i>	Whether the association was successfully removed or not. boolean.
<i>sessionId</i>	The session identifier. long.
<i>projectId</i>	The project identifier. int.
<i>locationId</i>	The location identifier. int.

removeResourceTag method (scentities)

Class

[scentities](#).

Action

Removes resource tags from an execution server.

Syntax

```
scentities.activateExecServer(sessionId, locationId, execServerId,
resourceTag)
```

Variable	Description
<i>sessionId</i>	The session identifier. long.
<i>locationId</i>	The location identifier. int.
<i>execServerId</i>	The execution server identifier. int.

Variable	Description
<i>resourceTag</i>	The resource tag to be removed from the server. String.

setMixedModeAuthentication Method (scentities)

Class

scentities.

Action

Sets the authentication mode for a user.

Syntax

```
scentities.setMixedModeAuthentication(sessionId, userId, useMixedMode)
```

Variable	Description
<i>sessionId</i>	The session identifier. long.
<i>userId</i>	The user identifier. int.
<i>useMixedMode</i>	Whether to use mixed mode to authenticate the specified user or not. boolean.

setPassword Method (scentities)

Class

scentities.

Action

Sets the password for the user. The password has to be encrypted on the client. This method is for internal use only.

Syntax

```
scentities.setPassword(sessionId, userId, password)
```

Variable	Description
<i>sessionId</i>	The session identifier. long.
<i>userId</i>	The user identifier. int.
<i>password</i>	The new password for the user. String.

setProxyUsernameAndPasswordForLocation Method (scentities)

Class

scentities.

Action

Sets the proxy user name and the password for the specified location.

Syntax

```
scentities.setProxyUsernameAndPasswordForLocation(sessionId, locationId,  
proxyUsername, proxyPassword)
```

Variable	Description
<i>sessionId</i>	The session identifier. long.
<i>locationId</i>	The location identifier. int.
<i>proxyUsername</i>	The proxy user name. String.
<i>proxyPassword</i>	The proxy password for the user. String.

updateExecServer Method (scentities)

Class

[scentities](#).

Action

Updates the properties of an existing execution server.

Syntax

```
scentities.updateExecServer(sessionId, execServer)
```

Variable	Description
<i>sessionId</i>	The session identifier. long.
<i>execServer</i>	The execution server which should be updated. ExecServer .

updateLocation Method (scentities)

Class

[scentities](#).

Action

Updates a location with the data that is provided by the location details.

Syntax

```
scentities.updateLocation(sessionId, locationDetails)
```

Variable	Description
<i>sessionId</i>	The session identifier. long.
<i>locationDetails</i>	The location details. LocationDetails .

updateMembershipsOfGroup Method (scentities)

Class

[scentities](#).

Action

Updates the memberships of a group.

Syntax

```
scentities.updateMembershipsOfGroup(sessionId, groupId, memberships)
```

Variable	Description
<i>sessionId</i>	The session identifier. long.
<i>groupId</i>	The group identifier. int.
<i>memberships</i>	The memberships of the specified user group. UserGroupMemberships[] .

updateMembershipsOfUser Method (scentities)

Class

[scentities](#).

Action

Updates the user group memberships of a user.

Syntax

```
scentities.updateMembershipsOfUser(sessionId, userId, memberships)
```

Variable	Description
<i>sessionId</i>	The session identifier. long.
<i>userId</i>	The group identifier. int.
<i>memberships</i>	The memberships of the specified user group. UserGroupMemberships[] .

updateProject Method (scentities)

Class

[scentities](#).

Action

Updates an existing project with the given properties.

Syntax

```
scentities.updateProject(sessionId, project)
```

Variable	Description
<i>sessionId</i>	The session identifier. long.
<i>project</i>	The project to be updated. Project .

updateUser Method (scentities)

Class

[scentities](#).

Action

Updates the properties of an existing user.

Syntax

```
scentities.updateUser(sessionId, userDetails)
```

Variable	Description
<i>sessionId</i>	The session identifier. long.
<i>userDetails</i>	The location details. UserDetails .

updateUserGroup Method (scentities)

Class

[scentities](#).

Action

Updates an existing user group.

Syntax

```
scentities.updateUserGroup(sessionId, userGroup)
```

Variable	Description
<i>sessionId</i>	The session identifier. long.
<i>userGroup</i>	The user group. UserGroup .

sccadminctrl

Description

This service provides methods for configuring entities on the SCA level.

The WSDL file of the service is available at `/services/sccadminctrl?wsdl`.

Methods

Name	Description
<code>setProjectState</code>	Activates or deactivates the given project.

setProjectState Method (sccadminctrl)

Class

[sccadminctrl](#).

Action

Activates or deactivates the given project.

Syntax

```
sccadminctrl.setProjectState(sessionId, projectId, active)
```

Variable	Description
<i>sessionId</i>	The session identifier. long.
<i>projectId</i>	The project identifier. int.
<i>active</i>	Whether the project is active or not. boolean.

sventities

Description

This service provides read access to several Silk Performance Manager entities.

The WSDL file of the service is available at `/services/sventities?wsdl`.

Methods

Name	Description
addRuleExpressionPart	Adds a rule expression part as the last expression part to the specified rule. int.
createCondition	Create a condition with the specified parameters. int.
createCustomIncident	Create a custom incident with the specified parameters. int.
createDefaultRule	Creates a default rule with the specified default conditions. int.
createRule	Creates a rule with the specified parameters. int.
createRuleWithEssentialAction	Creates a rule with the specified parameters. int.
deleteCondition	Deletes the specified condition. boolean.
deleteCustomIncident	Deletes the specified custom incident. boolean.
deleteRule	Deletes the specified rule. boolean.

Name	Description
<i>deleteRuleExpressionPart</i>	Deletes the rule expression part with the specified position. boolean.
<i>getActionEssentials</i>	Retrieves all action essentials. <i>EssentialDescription[]</i> .
<i>getActionSettings</i>	Retrieves all action settings for the specified rule. <i>ActionSetting[]</i> .
<i>getActionTypeIds</i>	Retrieves the identifiers of the action types. <i>String[]</i> .
<i>getAlertingLocations</i>	Retrieves all locations that have an alerting execution server. <i>int[]</i> .
<i>getClientSideExecutionSets</i>	Retrieves all execution sets of client side monitoring transactions. <i>ClientSideExecutionSet[]</i> .
<i>getCondition</i>	Retrieves the condition with the specified identifier. <i>Condition</i> .
<i>getConditions</i>	Retrieves all conditions for the specified project. <i>Condition[]</i> .
<i>getCustomIncident</i>	Retrieves the custom incident with the specified identifier. <i>CustomIncident</i> .
<i>getCustomIncidents</i>	Retrieves all custom incidents for the specified project. <i>CustomIncident[]</i> .
<i>getCustomIncidents</i>	Retrieves all custom incidents for the specified project within the specified timespan. <i>CustomIncident[]</i> .
<i>getExecutionLog</i>	Retrieves all execution log entries that meet the specified filter criteria. <i>ExecutionLogEntry[]</i> .
<i>getMeasureNamesForProject</i>	Retrieves the names of all measures that are available for the specified project. <i>String[]</i> .
<i>getMeasures</i>	Retrieves all measures for a given execution set. <i>Measure[]</i> .
<i>getMonitors</i>	Retrieves all monitors for the specified project. <i>Monitor[]</i> .
<i>getPageUrl</i>	Returns a URL, except <i>host:port</i> , which can be used as a bookmark. <i>String</i> .
<i>getPageUrl</i>	Returns a URL, except <i>host:port</i> , which can be used as a bookmark. <i>String</i> .
<i>getProfiles</i>	Retrieves the profiles that are available for the specified essential. <i>String[]</i> .
<i>getResultLocations</i>	Retrieves all locations where monitoring data exist for a given user. <i>Location[]</i> .
<i>getRule</i>	Retrieves the rule with the specified identifier. <i>Rule</i> .
<i>getRules</i>	Retrieves all rules for the specified project. <i>Rule[]</i> .
<i>getRuleExpressionParts</i>	Returns all rule expression parts for the specified rule. <i>RuleExpressionPart[]</i> .
<i>getRuleIncidents</i>	Returns all rule incidents for the specified project. <i>RuleIncident[]</i> .

Name	Description
getRuleIncidents	Returns all rule incidents within the given timespan for the specified project. RuleIncident[] .
getServerSideExecutionSets	Retrieves all execution sets of server side monitoring transactions. ServerSideExecutionSet[] .
getServersForProject	Retrieves all infrastructure servers where monitoring data exists for a given project. Server[] .
getTransactionsForProject	Retrieves all transactions for the specified project. Transaction[] .
getTransactionsForMonitor	Retrieves all transactions for the specified monitor. Transaction[] .
moveRuleExpressionPart	Move rule expression part up or down. <code>boolean</code> .
updateActionSettings	Updates the action settings for a non-essential action.
updateCondition	Updates a condition with the specified values. <code>boolean</code> .
updateCustomIncident	Updates a custom incident with the given values. <code>boolean</code> .
updateEssentialActionSettings	Updates the action settings for an essential action.
updateRule	Updates a rule with the specified values. <code>boolean</code> .
updateRuleExpressionPart	Update a rule expression part with the specified values. <code>boolean</code> .

ActionSetting Class

Description

The ActionSetting class represents a setting of an action.

Syntax

```
public class ActionSetting
```

Properties

Name	Description
name	The name of the action setting. <code>String</code> .
value	The value of the action setting. <code>String</code> .

name Property (ActionSetting)

Class

[ActionSetting](#).

Action

Gets or sets the name.

Syntax

```
public String name;
```

Access

Read and write.

value Property (ActionSetting)

Class

[ActionSetting](#).

Action

Gets or sets the value.

Syntax

```
public String value;
```

Access

Read and write.

ClientSideExecutionSet Class

Description

The `ClientSideExecutionSet` class represents a client-side execution set.

Inheritance Hierarchy

- [Entity](#)
 - [ExecutionSet](#)
 - `ClientSideExecutionSet`

Syntax

```
public class ClientSideExecutionSet extends ExecutionSet
```

Properties

Name	Description
browserId	Gets or sets the browser. int.
speedId	Gets or sets the speed. int.

browserId Property (ClientSideExecutionSet)

Class

[ClientSideExecutionSet](#).

Action

Gets or sets the browser.

Syntax

```
public int browserId;
```

Access

Read and write.

speedId Property (ClientSideExecutionSet)

Class

ClientSideExecutionSet.

Action

Gets or sets the speed.

Syntax

```
public int speedId;
```

Access

Read and write.

Condition Class

Description

The `Condition` class represents a condition.

Inheritance Hierarchy

- *Entity*
 - *NamedEntity*
 - `Condition`

Syntax

```
public class Condition extends NamedEntity
```

Properties

Name	Description
<i>projectId</i>	Gets or sets the project. <code>int</code> .
<i>transactionId</i>	Gets or sets the transaction. <code>int</code> .
<i>measurement</i>	Gets or sets the name of the measurement. <code>String</code> .
<i>measureType</i>	Gets or sets the type of the measurement. <code>String</code> .
<i>operator</i>	Gets or sets the operator of the condition. <code>String</code> .
<i>thresholdValue</i>	Gets or sets the value to be compared against. <code>double</code> .
<i>locationRestriction</i>	Gets or sets the locations which have to be matched for the condition to be true. <code>short</code> .

measurement Property (Condition)

Class

Condition.

Action

Gets or sets the name of the measurement.

Syntax

```
public String measurement;
```

Access

Read and write.

measureType Property (Condition)

Class

Condition.

Action

Gets or sets the type of the measurement.

Syntax

```
public String measureType;
```

Access

Read and write.

operator Property (Condition)

Class

Condition.

Action

Gets or sets the operator of the condition.

Syntax

```
public String operator;
```

Access

Read and write.

thresholdValue Property (Condition)

Class

Condition.

Action

Gets or sets the value to be compared against.

Syntax

```
public double thresholdValue;
```

Access

Read and write.

locationRestriction Property (Condition)

Class

Condition.

Action

Gets or sets the locations which have to be matched for the condition to be true.

Syntax

```
public short locationRestriction;
```

Access

Read and write.

CustomIncident Class

Description

The CustomIncident class represents a custom incident that has been entered by a user.

Inheritance Hierarchy

- *Entity*
 - *NamedEntity*
 - *Incident*
 - CustomIncident

Syntax

```
public class CustomIncident extends Incident
```

Properties

Name	Description
<i>projectId</i>	Gets or sets the project. int.
<i>invalidateInReports</i>	Gets or sets whether to override system incidents or not. boolean.

invalidateInReports Property (CustomIncident)

Class

CustomIncident.

Action

Gets or sets whether to override system incidents or not.

Syntax

```
public boolean invalidateInReports;
```

Access

Read and write.

ExecutionLogEntry Class

Description

The `ExecutionLogEntry` class represents an entry in the execution log.

Syntax

```
public class ExecutionLogEntry implements Serializable
```

Properties

Name	Description
<i>time</i>	Gets or sets the time that the corresponding monitor ran. Specified in milliseconds since midnight, January 1, 1970 UTC. <code>long</code> .
<i>message</i>	Gets or sets the readable message of the execution log entry. <code>String</code> .
<i>severity</i>	Gets or sets the severity of the execution log entry. The value is equal to one of the following: <ul style="list-style-type: none">• SEVERITY_INFO= 0• SEVERITY_SUCCESS= 1• SEVERITY_WARNING= 2• SEVERITY_ERROR= 3 <code>int</code> .
<i>transaction</i>	Gets or sets the name of the corresponding transaction. <code>String</code> .
<i>location</i>	Gets or sets the name of the location the monitor runs from. <code>String</code> .
<i>transactionId</i>	Gets or sets the identifier of the transaction. <code>String</code> .
<i>locationId</i>	Gets or sets the identifier of the location. <code>int</code> .
<i>projectId</i>	Gets or sets the identifier of the corresponding project. <code>int</code> .

Name	Description
<i>hasResultFiles</i>	Gets or sets whether result files are available for the corresponding run. Set to true if result files should be available, and set to false if not. <code>boolean</code> .
<i>hasResults</i>	Gets or sets whether results, for example monitored values, are available for the corresponding run. Set to true if results should be available, and set to false if not. <code>boolean</code> .

location Property (ExecutionLogEntry)

Class

[*ExecutionLogEntry*](#).

Action

Gets or sets the name of the location the monitor runs from.

Syntax

```
public String location;
```

Access

Read and write.

message Property (ExecutionLogEntry)

Class

[*ExecutionLogEntry*](#).

Action

Gets or sets the readable message of the execution log entry.

Syntax

```
public String message;
```

Access

Read and write.

severity Property (ExecutionLogEntry)

Class

[*ExecutionLogEntry*](#).

Action

Gets or sets the severity of the entry. The value is equal to one of the following:

- SEVERITY_INFO= 0
- SEVERITY_SUCCESS= 1

- SEVERITY_WARNING= 2
- SEVERITY_ERROR= 3

Syntax

```
public int severity;
```

Access

Read and write.

time Property

Class

- [DataPoint](#)
- [ExecutionLogEntry](#)

Action

Gets or sets the time that the corresponding monitor ran. Specified in milliseconds since midnight, January 1, 1970 UTC.

Syntax

```
public long time;
```

Access

Read and write.

transaction Property (ExecutionLogEntry)

Class

[ExecutionLogEntry](#).

Action

Gets or sets the name of the corresponding transaction.

Syntax

```
public String transaction;
```

Access

Read and write.

hasResultFiles Property (ExecutionLogEntry)

Class

[ExecutionLogEntry](#).

Action

Gets or sets whether result files are available for the corresponding run. Set to true if result files should be available, and set to false if not.

Syntax

```
public String hasResultFiles;
```

Access

Read and write.

hasResults Property (ExecutionLogEntry)

Class

ExecutionLogEntry.

Action

Gets or sets whether results, for example monitored values, are available for the corresponding run. Set to true if results should be available, and set to false if not.

Syntax

```
public String hasResults;
```

Access

Read and write.

ExecutionSet Class

Description

The `ExecutionSet` class represents an execution set.

Inheritance Hierarchy

- *Entity*
 - `ExecutionSet`

Syntax

```
public class ExecutionSet extends Entity
```

Properties

Name	Description
<i>projectId</i>	Gets or sets the project. int.
<i>transactionId</i>	Gets or sets the transaction. int.
<i>locationId</i>	Gets or sets the location. int.

projectId Property

Class

- *Condition*.
- *CustomIncident*.

- [ExecutionSet](#).
- [ExecutionLogEntry](#).
- [Rule](#).

Action

Gets or sets the project.

Syntax

```
public int projectId;
```

Access

Read and write.

browserId Property (ClientSideExecutionSet)

Class

[ClientSideExecutionSet](#).

Action

Gets or sets the browser.

Syntax

```
public int browserId;
```

Access

Read and write.

transactionId Property

Class

- [Condition](#).
- [ExecutionLogEntry](#)
- [ExecutionSet](#)

Action

Gets or sets the identifier of the transaction.

Syntax

```
public int transactionId;
```

Access

Read and write.

locationId Property

Class

- [ExecutionLogEntry](#)
- [ExecutionSet](#)

Action

Gets or sets the identifier of the location.

Syntax

```
public int locationId;
```

Access

Read and write.

Incident Class

Description

The Incident class represents an incident.

Inheritance Hierarchy

- [Entity](#)
 - [NamedEntity](#)
 - Incident

Syntax

```
public class Incident extends NamedEntity
```

Properties

Name	Description
beginsAt	Gets or sets the start time of the incident. long.
endsAt	Gets or sets the start time of the incident. long.
severity	Gets or sets how severe this incident is. String.

beginsAt Property (Incident)

Class

[Incident](#).

Action

Gets or sets the start time of the incident.

Syntax

```
public long beginsAt;
```

Access

Read and write.

endsAt property (Incident)

Class

[Incident](#).

Action

Gets or sets the end time of the incident.

Syntax

```
public long endsAt;
```

Access

Read and write.

severity Property (Incident)

Class

[Incident](#).

Action

Gets or sets how severe this incident is.

Syntax

```
public long severity;
```

Access

Read and write.

Measure Class

Description

The `Measure` class represents a measure.

Inheritance Hierarchy

- [Entity](#)
 - [NamedEntity](#)
 - Measure

Syntax

```
public class Measure extends NamedEntity
```


Monitor Class

Description

The `Monitor` class represents a monitor.

Inheritance Hierarchy

- `Entity`
 - `NamedEntity`
 - `Monitor`

Syntax

```
public class Monitor extends NamedEntity
```

Properties

Name	Description
<code>state</code>	<p>Gets or sets the current state of the monitor. The following states are available:</p> <ul style="list-style-type: none">• <code>STATE_ACTIVE = 1</code>• <code>STATE_INACTIVE = 0</code>• <code>STATE_INACTIVE_PROJECT = -1</code>• <code>STATE_INACTIVE_INFRASTRUCTURE = -2</code>• <code>STATE_SCHEDULE_FINISHED = -3</code>• <code>STATE_NO_EXEC_SERVERS = -4</code>• <code>STATE_NO_LICENSE = -5</code> <p>The default value is 1. <code>int</code>.</p>
<code>nextRun</code>	<p>Gets or sets the next time at which the monitor should run. Specified in milliseconds since midnight, January 1, 1970 UTC. <code>long</code>.</p>
<code>hasOwnSchedule</code>	<p>Gets or sets whether the monitor has an individual schedule or is using the schedule of the project. True if the monitor has an individual schedule, false if it has not. <code>boolean</code>.</p>
<code>monitoredServerId</code>	<p>Gets or sets the identifier of the server that the monitor is monitoring. Zero if the monitor is a client-side monitor that is not associated with a specific server. <code>int</code>.</p>

state Property (Monitor)

Class

`Monitor`.

Action

Gets or sets the current state of the monitor. The following states are available:

- `STATE_ACTIVE = 1`
- `STATE_INACTIVE = 0`

- STATE_INACTIVE_PROJECT = -1
- STATE_INACTIVE_INFRASTRUCTURE = -2
- STATE_SCHEDULE_FINISHED = -3
- STATE_NO_EXEC_SERVERS = -4
- STATE_NO_LICENSE = -5

The default value is 1. int.

Syntax

```
public int state = 1;
```

Access

Read and write.

nextRun Property (Monitor)

Class

Monitor.

Action

Gets or sets the next time at which the monitor should run. Specified in milliseconds since midnight, January 1, 1970 UTC.

Syntax

```
public long nextRun;
```

Access

Read and write.

hasOwnSchedule Property (Monitor)

Class

Monitor.

Action

Gets or sets whether the monitor has an individual schedule or is using the schedule of the project. True if the monitor has an individual schedule, false if it has not.

Syntax

```
public boolean hasOwnSchedule;
```

Access

Read and write.

monitoredServerId Property (Monitor)

Class

Monitor.

Action

Gets or sets the identifier of the server that the monitor is monitoring. Zero if the monitor is a client-side monitor that is not associated with a specific server.

Syntax

```
public int monitoredServerId;
```

Access

Read and write.

Rule Class

Description

The `Rule` class represents a rule.

Inheritance Hierarchy

- [Entity](#)
 - [NamedEntity](#)
 - Rule

Syntax

```
public class Rule extends NamedEntity
```

Properties

Name	Description
actionType	Gets or sets the type of action which should be taken if the rule matches. <code>String</code> .
projectId	Gets or sets the project. <code>int</code> .
active	Gets or sets whether the rule is active. <code>boolean</code> .
recurThreshold	Gets or sets how often the expression must be matched. <code>short</code> .
recurTimespan	Gets or sets the timespan within which the expression must be matched <code>recurThreshold</code> times. <code>long</code> .
severity	Gets or sets the severity of the rule. <code>String</code> .

actionType Property (Rule)

Class

[Rule](#).

Action

Gets or sets the type of action which should be taken if the rule matches.

Syntax

```
public String actionType;
```

Access

Read and write.

active Property (Rule)**Class**

[Rule](#).

Action

Indicates whether the rule is active or not.

Syntax

```
public boolean active;
```

Access

Read and write.

recurThreshold Property (Rule)**Class**

[Rule](#).

Action

Gets or sets how often the expression must be matched.

Syntax

```
public short recurThreshold;
```

Access

Read and write.

recurTimespan Property (Rule)**Class**

[Rule](#).

Action

Gets or sets the timespan within which the expression must be matched `recurThreshold` times.

Syntax

```
public long recurTimespan;
```

Access

Read and write.

severity Property (Rule)

Class

[Rule](#).

Action

Gets or sets the severity of the rule.

Syntax

```
public String severity;
```

Access

Read and write.

RuleExpressionPart Class

Description

The `RuleExpressionPart` class represents a part of a rule expression. One or several expressions are used to form a rule.

Inheritance Hierarchy

- [Entity](#)
 - `RuleExpressionPart`

Syntax

```
public class RuleExpressionPart extends Entity
```

Properties

Name	Description
ruleId	Gets or sets the rule. <code>int</code> .
conditionId	Gets or sets the condition. <code>int</code> .
conditionName	Gets or sets the name of the condition. <code>String</code> .
operator	Gets or sets the operator of the condition. <code>String</code> .

ruleId Property (RuleExpressionPart)

Class

[RuleExpressionPart](#).

Action

Gets or sets the identifier of the rule.

Syntax

```
public int ruleId;
```

Access

Read and write.

conditionId Property (RuleExpressionPart)

Class

RuleExpressionPart.

Action

Gets or sets the identifier of the condition.

Syntax

```
public int conditionId;
```

Access

Read and write.

conditionName Property (RuleExpressionPart)

Class

RuleExpressionPart.

Action

Gets or sets the name of the condition.

Syntax

```
public String conditionName;
```

Access

Read and write.

RuleIncident Class

Description

The `RuleIncident` class holds information for an incident that is caused by a rule.

Inheritance Hierarchy

- *Entity*
 - *NamedEntity*
 - *Incident*
 - `RuleIncident`

Syntax

```
public class RuleIncident extends Incident
```

Properties

Name	Description
<code>ruleId</code>	Gets or sets the rule. int.
<code>notifyFailMessage</code>	Gets or sets the notification message. String.
<code>notifyFailState</code>	Gets or sets the notification state. byte.

notifyFailMessage Property (RuleIncident)

Class

`RuleIncident`.

Action

Gets or sets the notification message.

Syntax

```
public String notifyFailMessage;
```

Access

Read and write.

notifyFailState Property (RuleIncident)

Class

`RuleIncident`.

Action

Gets or sets the notification state.

Syntax

```
public String notifyFailMessage;
```

Access

Read and write.

Server Class

Description

The `Server` class represents a server.

Inheritance Hierarchy

- `Entity`
 - `NamedEntity`
 - `Server`

Syntax

```
public class Server extends NamedEntity
```

ServerSideExecutionSet Class

Description

The `ServerSideExecutionSet` class represents a server side execution set.

Inheritance Hierarchy

- `Entity`
 - `ExecutionSet`
 - `ServerSideExecutionSet`

Syntax

```
public class ServerSideExecutionSet extends ExecutionSet
```

Transaction Class

Description

The `Transaction` class represents a transaction.

Inheritance Hierarchy

- `Entity`
 - `NamedEntity`
 - `Transaction`

Syntax

```
public class Transaction extends NamedEntity
```

addRuleExpressionPart Method (sventities)

Class

`sventities.`

Action

Adds a rule expression part as the last expression part to the specified rule.

Syntax

```
position = sventities.addRuleExpressionPart(sessionId, ruleId, operator, conditionId)
```

Variable	Description
<code>position</code>	The position at which the rule expression part was added to the rule. <code>int</code> .

Variable	Description
<i>sessionId</i>	The session identifier. long.
<i>ruleId</i>	The rule identifier. int.
<i>operator</i>	The operator. String.
<i>conditionId</i>	The condition identifier. int.

createCondition Method (sventities)

Class

[sventities](#).

Action

Creates a condition with the specified parameters.

Syntax

```
conditionId = sventities.createCondition(sessionId, name, transactionId, measurement, operator, thresholdValue, measureType, locationRestriction)
```

Variable	Description
<i>conditionId</i>	The position at which the rule expression part was added to the rule. int.
<i>sessionId</i>	The session identifier. long.
<i>name</i>	The name of the new condition. String.
<i>transactionId</i>	The transaction identifier. int.
<i>measurement</i>	The measurement. String.
<i>operator</i>	The operator. String.
<i>thresholdValue</i>	The value of the threshold. double.
<i>measureType</i>	The type of the measure. String.
<i>locationRestriction</i>	The restriction for the location. short.

createCustomIncident Method (sventities)

Class

[sventities](#).

Action

Creates a custom incident with the specified parameters.

Syntax

```
incidentId = sventities.createCustomIncident(sessionId, name, description, projectId, invalidateInReports, severity, beginsAt, endsAt)
```

Variable	Description
<i>incidentId</i>	The identifier of the new custom incident. <code>int</code> .
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>name</i>	The name of the new custom incident. <code>String</code> .
<i>description</i>	The description of the new custom incident. <code>String</code> .
<i>projectId</i>	The project identifier. <code>int</code> .
<i>invalidateInReports</i>	Whether to invalidate the custom incident in reports or not. <code>boolean</code> .
<i>severity</i>	The severity of the new custom incident. <code>String</code> .
<i>beginsAt</i>	Where the custom incident starts. <code>long</code> .
<i>endsAt</i>	Where the custom incident ends. <code>long</code> .

createDefaultRule Method (sventities)

Class

`scentities`.

Action

Creates a default rule with the specified default conditions.

Syntax

```
ruleID = scentities.createDefaultRule(sessionId, monitorId, measures)
```

Variable	Description
<i>ruleID</i>	The identifier of the new rule. <code>int</code> .
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>monitorId</i>	The identifier of the monitor. <code>int</code> .
<i>measures</i>	The measures for the new rule. <code>String[]</code> .

createRule Method (sventities)

Class

`sventities`.

Action

Creates a new rule with the specified parameters.

Syntax

```
ruleID = sventities.createRule(sessionId, name, projectId, actionType, active, recurThreshold, recurTimespan, severity, conditionId)
```

Variable	Description
<i>ruleID</i>	The identifier of the new rule. <code>int</code> .
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>name</i>	The name of the new rule. <code>String</code> .
<i>projectId</i>	The identifier of the project. <code>int</code> .
<i>actionType</i>	The action type of the new rule. <code>String</code> .
<i>active</i>	Whether the new rule is active or not. <code>boolean</code> .
<i>recurThreshold</i>	The threshold for the recurrence of the new rule. <code>short</code> .
<i>recurTimespan</i>	The timespan for the recurrence of the new rule. <code>long</code> .
<i>severity</i>	The severity of the new rule. <code>String</code> .
<i>conditionId</i>	The identifier of the condition. <code>int</code> .

createRuleWithEssentialAction Method (sventities)

Class

[sventities](#).

Action

Creates a new rule with the specified parameters.

Syntax

```
ruleID = sventities.createRuleWithEssentialAction(sessionId, name, projectId, essential, active, recurThreshold, recurTimespan, severity, conditionId)
```

Variable	Description
<i>ruleID</i>	The identifier of the new rule. <code>int</code> .
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>name</i>	The name of the new rule. <code>String</code> .
<i>projectId</i>	The identifier of the project. <code>int</code> .
<i>essential</i>	The action type of the new rule. <code>String</code> .
<i>active</i>	Whether the new rule is active or not. <code>boolean</code> .
<i>recurThreshold</i>	The threshold for the recurrence of the new rule. <code>short</code> .
<i>recurTimespan</i>	The timespan for the recurrence of the new rule. <code>long</code> .
<i>severity</i>	The severity of the new rule. <code>String</code> .
<i>conditionId</i>	The identifier of the condition. <code>int</code> .

deleteCondition Method (sventities)

Class

[sventities](#).

Action

Deletes the specified condition.

Syntax

```
success = sventities.deleteCondition(sessionId, conditionId)
```

Variable	Description
<i>success</i>	Whether the deletion was successful or not. <code>boolean</code> .
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>conditionId</i>	The identifier of the condition. <code>int</code> .

deleteCustomIncident Method (sventities)

Class

`sventities`.

Action

Deletes the specified custom incident.

Syntax

```
success = sventities.deleteCustomIncident(sessionId, customIncidentId)
```

Variable	Description
<i>success</i>	Whether the deletion was successful or not. <code>boolean</code> .
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>customIncidentId</i>	The identifier of the custom incident. <code>int</code> .

deleteRule Method (sventities)

Class

`sventities`.

Action

Deletes the specified rule.

Syntax

```
success = sventities.deleteRule(sessionId, ruleId)
```

Variable	Description
<i>success</i>	Whether the deletion was successful or not. <code>boolean</code> .
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>ruleId</i>	The identifier of the rule. <code>int</code> .

deleteRuleExpressionPart Method (sventities)

Class

[sventities](#).

Action

Deletes the rule expression part with the specified position.

Syntax

```
success = sventities.deleteRuleExpressionPart(sessionId, ruleId, position)
```

Variable	Description
<i>success</i>	Whether the deletion was successful or not. <code>boolean</code> .
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>ruleId</i>	The rule identifier. <code>int</code> .
<i>position</i>	The position of the expression part in the rule. <code>int</code> .

getActionEssentials Method (sventities)

Class

[sventities](#).

Action

Retrieves all action essentials.

Syntax

```
description = sventities.getActionEssentials(sessionId, projectId)
```

Variable	Description
<i>description</i>	The descriptions of the essentials. EssentialDescription[] .
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>projectId</i>	The project identifier. <code>int</code> .

getActionSettings Method (sventities)

Class

[sventities](#).

Action

Retrieves all action settings for the specified rule.

Syntax

```
settings = sventities.getActionSettings(sessionId, ruleId)
```

Variable	Description
<i>settings</i>	The settings of the rule. ActionSetting[] .
<i>sessionId</i>	The session identifier. long.
<i>ruleId</i>	The rule identifier. int.

getActionTypeIds Method (sventities)

Class

[sventities](#).

Action

Retrieves the identifiers of the action types.

Syntax

```
actionTypeIds = sventities.getActionTypeIds()
```

Variable	Description
<i>actionTypeIds</i>	the identifiers of the action types. String[] .

getAlertingLocations Method (sventities)

Class

[sventities](#).

Action

Retrieves all locations that have an alerting execution server.

Syntax

```
locations = sventities.getAlertingLocations(sessionId, projectId)
```

Variable	Description
<i>locations</i>	The identifiers of all locations that have an alerting execution server. int[] .
<i>sessionId</i>	The session identifier. long.
<i>projectId</i>	The project identifier. int.

getClientSideExecutionSets Method (sventities)

Class

[sventities](#).

Action

Retrieves all execution sets of client side monitoring transactions.

Syntax

```
executionSets = sventities.getClientSideExecutionSets(sessionId, transactionId)
```

Variable	Description
<i>executionSets</i>	The client side execution sets. ClientSideExecutionSet[] .
<i>sessionId</i>	The session identifier. long.
<i>transactionId</i>	The transaction identifier. int.

getCondition Method (sventities)

Class

[sventities](#).

Action

Retrieves the condition with the specified identifier.

Syntax

```
condition = sventities.getCondition(sessionId, conditionId)
```

Variable	Description
<i>condition</i>	The condition. Condition .
<i>sessionId</i>	The session identifier. long.
<i>conditionId</i>	The condition identifier. int.

getConditions Method (sventities)

Class

[sventities](#).

Action

Retrieves all conditions for the specified project.

Syntax

```
conditions = sventities.getConditions(sessionId, projectId)
```

Variable	Description
<i>conditions</i>	The conditions. Condition[] .
<i>sessionId</i>	The session identifier. long.
<i>projectId</i>	The project identifier. int.

getCustomIncident Method (sventities)

Class

sventities.

Action

Retrieves the custom incident with the specified identifier.

Syntax

```
customIncident = sventities.getCustomIncident(sessionId, customIncidentId)
```

Variable	Description
<i>customIncident</i>	The custom incident. <i>CustomIncident</i> .
<i>sessionId</i>	The session identifier. long.
<i>customIncidentId</i>	The identifier of the custom incident. int.

getCustomIncidents Method (sventities)

Class

sventities.

Action

Retrieves all custom incidents for the specified project.

Syntax

```
customIncidents = sventities.getCustomIncidents(sessionId, projectId)
```

Variable	Description
<i>customIncidents</i>	The custom incidents in the specified project. <i>CustomIncident[]</i> .
<i>sessionId</i>	The session identifier. long.
<i>projectId</i>	The identifier of the project. int.

getCustomIncidents Method (sventities)

Class

sventities.

Action

Retrieves all custom incidents for the specified project within the specified timespan.

Syntax

```
customIncidents = sventities.getCustomIncidents(sessionId, projectId, startTimeUTC, endTimeUTC)
```


Variable	Description
<i>customIncidents</i>	The custom incidents in the specified project. CustomIncident[] .
<i>sessionId</i>	The session identifier. long.
<i>projectId</i>	The identifier of the project. int.
<i>startTimeUTC</i>	The start time for the timespan in which to retrieve all custom incidents. long.
<i>endTimeUTC</i>	The end time for the timespan in which to retrieve all custom incidents. long.

getExecutionLog Method (sventities)

Class

[sventities](#).

Action

Retrieves all execution log entries that meet the specified filter criteria.

Syntax

```
executionLogEntries = sventities.getExecutionLog(sessionId, projectId, from, to, sortCol, isAscending, filterColumns, filterIds, maxListSize, actualPage)
```

Variable	Description
<i>executionLogEntries</i>	An array of all execution log entries that suit the specified filter criteria. ExecutionLogEntry[] .
<i>sessionId</i>	The session identifier. long.
<i>projectId</i>	The project identifier. int.
<i>from</i>	The earliest timestamp that the method should return execution log entries for. Specified in milliseconds since midnight, January 1, 1970 UTC. long.
<i>to</i>	The latest timestamp that the method should return execution log entries for. Specified in milliseconds since midnight, January 1, 1970 UTC. long.
<i>sortCol</i>	The column by which the returned array is sorted. The numerical value must match one of the following: <ul style="list-style-type: none"> • COL_TIME = 0 • COL_MESSAGE= 1 • COL_STATUS = 2 • COL_TRANS_NAME= 3 • COL_TRANS_ID= 4 • COL_LOCATION_NAME= 5 • COL_LOCATION_ID= 6
<i>isAscending</i>	Set to true to use ascending order for the returned array or set to false to use descending order. boolean.
<i>filterColumns</i>	This parameter works in combination with the <i>filterIds</i> parameter. If neither array is null, then they must be of equal length. Each element in the array <i>filterColumns</i> specifies a column that should be filtered. The corresponding element, which is the element with the same index, in the array <i>filterIds</i> specifies the value that must be

Variable	Description
	<p>matched by all execution log entries. Each element in the array <i>filterColumns</i> must match one of the following values:</p> <ul style="list-style-type: none"> • COL_STATUS = 2 • COL_TRANS_ID= 4 • COL_LOCATION_ID= 6
<i>filterIds</i>	<p>This parameter works in combination with the <i>filterColumns</i> parameter. If neither array is null, then they must be of equal length. Each element in the array <i>filterColumns</i> specifies a column that should be filtered. The corresponding element, which is the element with the same index, in the array <i>filterIds</i> specifies the value that must be matched by all execution log entries. Each element in the array <i>filterColumns</i> must match one of the following values:</p> <ul style="list-style-type: none"> • COL_STATUS = 2 • COL_TRANS_ID= 4 • COL_LOCATION_ID= 6
<i>maxListSize</i>	<p>Queried entries are divided into pages. This parameter specifies the length of these pages. The method call always returns one of these pages, which means that the size of the returned array is always less or equal to the value of parameter <i>maxListSize</i>. See also the description of the parameter <i>actualPage</i>.</p>
<i>actualPage</i>	<p>Selects the result page that is to be returned. See also the description of the parameter <i>maxListSize</i>.</p>

getMeasureNamesForProject Method (sventities)

Class

sventities.

Action

Retrieves the names of all measures that are available for the specified project.

Syntax

```
measureNames = sventities.getMeasureNamesForProject(sessionId, projectId)
```

Variable	Description
<i>measureNames</i>	An array of all measure names where monitoring data exists for the given project. <code>String[]</code> .
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>projectId</i>	The project identifier. <code>int</code> .

getMeasures Method (sventities)

Class

sventities.

Action

Retrieves all measures for a given execution set. The method can be used for both client-side and infrastructure execution sets.

Syntax

```
measures = sventities.getMeasures(sessionId, execSetId)
```

Variable	Description
<i>measures</i>	An array of all measures that are associated with the given execution set. Measure[] .
<i>sessionId</i>	The session identifier. long.
<i>execSetId</i>	The execution set identifier. int.

getMonitors Method (sventities)

Class

[sventities](#).

Action

Retrieves all monitors for the specified project.

Syntax

```
monitors = sventities.getMonitors(sessionId, projectId)
```

Variable	Description
<i>monitors</i>	An array of all monitors that are associated with the given project. Monitor[] .
<i>sessionId</i>	The session identifier. long.
<i>projectId</i>	The project identifier. int.

getPageURL Method (sventities)

Class

[sventities](#).

Action

Returns a URL, except *host:port*, which can be used as a bookmark.

Syntax

```
pageURL = sventities.getPageURL(sessionId, projectId, report, params)
```

Variable	Description
<i>pageURL</i>	A URL for a certain report, for example <code>/DEF/Monitoring/Reports?rAc&pId=2&mainTab=0</code> . String.
<i>sessionId</i>	The session identifier. long.

Variable	Description
<i>projectId</i>	The project identifier. <code>int</code> .
<i>report</i>	The earliest timestamp that the method should return execution log entries for. Specified in milliseconds since midnight, January 1, 1970 UTC. <code>long</code> .
<i>params</i>	The latest timestamp that the method should return execution log entries for. Specified in milliseconds since midnight, January 1, 1970 UTC. <code>long</code> .

getPageURL Method (sventities)

Class

[sventities](#).

Action

Returns a URL, except *host:port*, which can be used as a bookmark.

Syntax

```
pageURL = sventities.getPageURL(sessionId, projectId, beginTime, endTime, transactions, locations, report, params)
```

Variable	Description
<i>pageURL</i>	A URL for a certain report, for example <code>/DEF/Monitoring/Reports?rAc&pId=2&mainTab=0</code> . <code>String</code> .
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>projectId</i>	The project identifier. <code>int</code> .
<i>beginTime</i>	The begin time in milliseconds from 1.1.1970. 0 if the last 24 hours should be returned. <code>long</code> .
<i>endTime</i>	The end time in milliseconds from 1.1.1970. 0 if the last 24 hours should be returned. <code>long</code> .
<i>transactions</i>	An array with the identifiers of the allowed transactions. null or an empty array if all transactions are allowed. <code>int[]</code> .
<i>locations</i>	An array with the identifiers of the allowed locations. null or an empty array if all locations are allowed. <code>int[]</code> .
<i>report</i>	The requested report. For information on the supported reports, see Supported Reports for the getPageURL Method . <code>String</code> .
<i>params</i>	The latest timestamp that the method should return execution log entries for. Specified in milliseconds since midnight, January 1, 1970 UTC. <code>long</code> .

Examples

```
String url = ws.getPageUrl(sessionId, projectId, "incidentsLog", null);
```

```
String url = ws.getPageUrl(sessionId, projectId, "cCustomTimerDetailsRep",  
new String[] {"Sv_Ping@MyServer"});
```

```
String url = ws.getPageUrl(sessionId, projectId, 1136293633214L,  
1137503236714L, new int[] {0},  
new int[] {2}, "cCustomCounterDetailsRep", new String[] {"Sv_Ping@MyServer"});
```

```
String url = ws.getPageUrl(sessionId, -1, "adminReportsAuditLog", null);
```

Supported Reports for the getPageURL Method

The following reports are supported for the `getPageURL` method of the `sventities` webservice:

Project-level reports

Report	Description	Report Parameter
Incidents Log	This log lists all incidents that occurred during a specified time period.	incidentsLog
Service Level Log	Offers information about service level violations.	serviceLevelLog
Execution Log	Lists all transaction executions within a specified time period.	executionLog
Custom Reports	Defines custom reports that transform raw data into charts and graphs.	customReports
Health Overview Report	Overview report of the project health.	cHealthOverRep
Availability Overview Report	Overview report of the project availability.	cAvailOverRep
Accuracy Overview Report	Overview report of the project accuracy.	cAccurOverRep
Performance Overview Report	Overview report of the project performance.	cPerfOverRep
Availability Detail Report	Detail report of availability error count.	cAvailDetailRep
Accuracy Detail Report	Detail report of accuracy error count.	cAccurDetailRep
Transaction Response Time Detail Report	Detail report of transaction response time.	cTransactionResponseTimeDetailRep
Page Timer Detail Report	Detail report of a given page timer. Needs page timer name in params.	cPageTimerDetailsRep
Custom Timer Detail Report	Detail report of a given custom timer. Needs custom timer name in params.	cCustomTimerDetailsRep
Custom Counter Detail Report	Detail report of a given custom counter. Needs custom counter name in params.	cCustomCounterDetailsRep

High-level reports

High level reports are not related to a specific project, they show an overview over all projects.

Report	Description	Report Parameter
High Level Health Status	Health status for all projects as a list.	hlHealthStatus
High Level Health Drilldown	Health drill-down for all projects as a list.	hlHealthDrilldown
High Level Health History	Health history for all projects as a list.	hlHealthHistory

Report	Description	Report Parameter
High Level SLA Status	SLA status for all projects as a list.	hiSLAStatus
High Level Snapshot	Snapshot for all projects as list.	hiSnapShot

Configuration reports

The returned URLs link to **Silk Performance Manager > Configuration** pages.

Report	Description	Report Parameter
Configuration Monitors	Configures monitors for the given project.	confMonitors
Configuration Rules	Configures rules for the given project.	confRules
Configuration Conditions	Configures conditions for the given project.	confConditions
Configuration Custom Incidents	Configures custom incidents for the given project.	confCustomIncidents

Administration reports

The returned URLs link to **Silk Performance Manager > Administration** pages.

Report	Description	Report Parameter
Administration System Database	The database administration page.	adminSystemDatabase
Administration System Chart Engines	The page for administrating chart engines.	adminSystemChartEngines
Administration System Notification	The page for administrating notification types.	adminSystemNotification
Administration System Notification Mail	The page for administrating email notifications.	adminSystemNotificationMail
Administration System Notification SMS	The page for administrating SMS notifications.	adminSystemNotificationSMS
Administration System Notification PageGate	The page for administrating PageGate pager notifications.	adminSystemNotificationPage
Administration System Notification SNMP	The page for administrating SNMP trap notifications.	adminSystemNotificationSNMP
Administration System Proxy	The page for administrating the system proxy settings.	adminSystemProxy
Administration Users	The page for administrating users.	adminUsers
Administration Projects	The page for administrating projects.	adminProjects
Administration Locations	The page for administrating locations.	adminLocations
Administration Reports Custom Reports	The page for administrating custom reports.	adminReportsCustomReports
Administration Reports Audit Log	The page that displays the audit log.	adminReportsAuditLog
Administration Reports FrontendServer Log	The page that displays the front-end server logs.	adminReportsFrontendServerLog
Administration Reports AppServer Log	The page that displays the application server logs.	adminReportsAppServerLog

Report	Description	Report Parameter
Administration Reports ExecServer Log	The page that displays the execution server logs.	adminReportsFrontend ServerLog
Administration Files Uploads	The Files > Uploads page.	adminFilesUploads
Administration Files Essentials	The Files > Essentials page.	adminFilesEssentials

getProfiles Method (sventities)

Class

[sventities](#).

Action

Retrieves the profiles that are available for the specified essential.

Syntax

```
profiles = sventities.getProfiles(sessionId, projectId, essentialName)
```

Variable	Description
<i>profiles</i>	The array with the profile names. <code>String[]</code> .
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>projectId</i>	The project identifier. <code>int</code> .
<i>essentialName</i>	The name of the essential. <code>String</code> .

getResultLocations Method (sventities)

Class

[sventities](#).

Action

Retrieves all locations where monitoring data exist for a specified user.

Syntax

```
locations = sventities.getResultLocations(sessionId, projectId, essentialName)
```

Variable	Description
<i>locations</i>	The array of all locations where monitoring data exists for the given user. Location[] .
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>userId</i>	The user identifier. <code>int</code> .

getRule Method (sventities)

Class

[sventities](#).

Action

Retrieves the rule with the specified identifier.

Syntax

```
rule = sventities.getRule(sessionId, ruleId)
```

Variable	Description
<i>rule</i>	The rule with the specified identifier. null if no matching rule is found or no sufficient access rights exist. Rule .
<i>sessionId</i>	The session identifier. long.
<i>ruleId</i>	The rule identifier. int.

getRuleExpressionParts Method (sventities)

Class

[sventities](#).

Action

Returns all rule expression parts for the specified rule.

Syntax

```
ruleExpressionParts = sventities.getRuleExpressionParts(sessionId, ruleId)
```

Variable	Description
<i>ruleExpressionParts</i>	The array of all rule expression parts that are associated to the specified rule. RuleExpressionPart[] .
<i>sessionId</i>	The session identifier. long.
<i>ruleId</i>	The rule identifier. int.

getRules Method (sventities)

Class

[sventities](#).

Action

Retrieves all rules for the specified project.

Syntax

```
rules = sventities.getRules(sessionId, projectId)
```

Variable	Description
<i>rules</i>	The array of all rules within the specified project. null if no rules are available or no sufficient access rights exist. Rule[] .
<i>sessionId</i>	The session identifier. long.
<i>projectId</i>	The project identifier. int.

getRuleIncidents Method (sventities)

Class

[sventities](#).

Action

Returns all rule incidents for the specified project.

Syntax

```
ruleIncidents = sventities.getRuleIncidents(sessionId, projectId)
```

Variable	Description
<i>ruleIncidents</i>	The array of all incidents within the specified project. null if no incidents are available or no sufficient access rights exist. RuleIncident[] .
<i>sessionId</i>	The session identifier. long.
<i>projectId</i>	The project identifier. int.

getRuleIncidents Method (sventities)

Class

[sventities](#).

Action

Returns all rule incidents within the given timespan for the specified project.

Syntax

```
ruleIncidents = sventities.getRuleIncidents(sessionId, projectId, startTimeUTC, endTimeUTC)
```

Variable	Description
<i>ruleIncidents</i>	The array of all incidents within the specified project. null if no incidents are available or no sufficient access rights exist. RuleIncident[] .
<i>sessionId</i>	The session identifier. long.
<i>projectId</i>	The project identifier. int.

Variable	Description
<i>startTimeUTC</i>	Start time in milliseconds from 1.1.1970. long.
<i>endTimeUTC</i>	End time in milliseconds from 1.1.1970. long.

getServerSideExecutionSets Method (sventities)

Class

sventities.

Action

Retrieves all execution sets of server side monitoring transactions.

Syntax

```
execSets = sventities.getServerSideExecutionSets(sessionId, transactionId)
```

Variable	Description
<i>execSets</i>	The array of all execution sets that are associated with the given transaction. ServerSideExecutionSet[] .
<i>sessionId</i>	The session identifier. long.
<i>transactionId</i>	The project identifier. int.

getServersForProject Method (sventities)

Class

sventities.

Action

Retrieves all infrastructure servers where monitoring data exists for a given project.

Syntax

```
servers = sventities.getServersForProject(sessionId, projectId)
```

Variable	Description
<i>servers</i>	The array of all infrastructure servers that are associated with the given project. Server[] .
<i>sessionId</i>	The session identifier. long.
<i>projectId</i>	The project identifier. int.

getTransactionsForProject Method (sventities)

Class

sventities.

Action

Retrieves all transactions for the specified project.

Syntax

```
transactions = sventities.getTransactionsForProject(sessionId, projectId)
```

Variable	Description
<i>transactions</i>	The array of all transactions that are associated with the specified project. Transaction[] .
<i>sessionId</i>	The session identifier. long.
<i>projectId</i>	The project identifier. int.

getTransactionsForMonitor Method (sventities)

Class

[sventities](#).

Action

Retrieves all transactions for the specified monitor.

Syntax

```
transactions = sventities.getTransactionsForMonitor(sessionId, monitorId)
```

Variable	Description
<i>transactions</i>	The array of all transactions that are associated with the specified monitor. Transaction[] .
<i>sessionId</i>	The session identifier. long.
<i>monitorId</i>	The monitor identifier. int.

moveRuleExpressionPart Method (sventities)

Class

[sventities](#).

Action

Move rule expression part up or down.

Syntax

```
success = sventities.moveRuleExpressionPart(sessionId, ruleId, position, destination)
```

Variable	Description
<i>success</i>	Whether the move was successful or not. boolean.
<i>sessionId</i>	The session identifier. long.

Variable	Description
<i>ruleId</i>	The identifier of the rule to which the rule expression part belongs. <code>int</code> .
<i>position</i>	The current position of the rule expression part that you want to move. <code>int</code> .
<i>destination</i>	The position in the rule to which you want to move the rule expression part. <code>int</code> .

updateActionSettings Method (sventities)

Class

[sventities](#).

Action

Updates the action settings for a non-essential action.

Syntax

```
sventities.updateActionSettings(sessionId, ruleId, actionType, settings)
```

Variable	Description
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>ruleId</i>	The rule identifier. <code>int</code> .
<i>actionType</i>	The new action type. <code>String</code> .
<i>settings</i>	The new settings. ActionSetting[] .

updateCondition (sventities)

Class

[sventities](#).

Action

Updates a condition with the specified values.

Syntax

```
success = sventities.updateCondition(sessionId, condition)
```

Variable	Description
<i>success</i>	Whether the update was successful or not. <code>boolean</code> .
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>condition</i>	The condition with the new values. Condition .

updateCustomIncident Method (sventities)

Class

[sventities](#).

Action

Updates a custom incident with the specified values.

Syntax

```
success = sventities.updateCustomIncident(sessionId, customIncident)
```

Variable	Description
<i>success</i>	Whether the update was successful or not. <code>boolean</code> .
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>customIncident</i>	The custom incident with the values to be updated. CustomIncident .

updateEssentialActionSettings Method (sventities)

Class

[sventities](#).

Action

Updates the action settings for an essential action.

Syntax

```
sventities.updateEssentialActionSettings(sessionId, ruleId, essential, settings)
```

Variable	Description
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>ruleId</i>	The rule identifier. <code>int</code> .
<i>essential</i>	The name of the essential action. <code>String</code> .
<i>settings</i>	The settings for the essential action. ActionSetting[] .

updateRule Method (sventities)

Class

[sventities](#).

Action

Updates a rule with the specified values.

Syntax

```
success = sventities.updateRule(sessionId, ruleId, name, active, recurThreshold, recurTimespan, severity)
```

Variable	Description
<i>success</i>	Whether the update was successful or not. <code>boolean</code> .

Variable	Description
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>ruleId</i>	The rule identifier. <code>int</code> .
<i>name</i>	The name of the rule. <code>String</code> .
<i>active</i>	Whether the rule is active or not. <code>boolean</code> .
<i>recurThreshold</i>	Specifies how many times the expression must be true to raise an incident. <code>short</code> .
<i>recurTimespan</i>	Specifies in which timespan the expression must be true <i>recurThreshold</i> times to raise an incident. <code>long</code> .
<i>severity</i>	The severity of the rule. The following values are available for this parameter: <ul style="list-style-type: none"> • Error • Warning • Informational • Service Target Violation <code>String</code> .

updateRuleExpressionPart

Class

[sventities](#).

Action

Update a rule expression part with the given values.

Syntax

```
success = sventities.updateRuleExpressionPart(sessionId, ruleId, position,
conditionId, operator)
```

Variable	Description
<i>success</i>	Whether the update was successful or not. <code>boolean</code> .
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>ruleId</i>	The rule identifier. <code>int</code> .
<i>position</i>	The position of the expression part that should be updated in the specified rule. <code>int</code> .
<i>conditionId</i>	The identifier of the condition that should be updated. <code>int</code> .
<i>operator</i>	The updated operator. <code>String</code> .

svdata

Description

This service provides functionality for querying monitor data.

The WSDL file of the service is available at `/services/svdata?wsdl`.

Methods

Name	Description
getClientMeasureData	Queries monitoring data of client side monitoring transactions from the database. DataPoint[] .
getClientMeasureDataAcrossProjects	Queries monitoring data of client side monitoring transactions across multiple projects from the database. DataPoint[] .
getLastClientData	Queries the most recent monitoring data of client side monitoring transactions from the database. DataPoint .
getLastData	Queries the most recent monitoring data from the database, including client and server side transactions. DataPoint .
getLastServerData	Queries the most recent monitoring data of server side monitoring transactions from the database. DataPoint .
getMeasureData	Queries monitoring data from the database. Both client and server side monitoring data is included. DataPoint[] .
getServerMeasureData	Queries monitoring data of server side monitoring transactions from the database. DataPoint[] .
getServerMeasureDataAcrossProjects	Queries monitoring data of server side monitoring transactions across multiple projects from the database. DataPoint[] .
mergeSeries	Merges a specified series to a single DataPoint. DataPoint .

DataPoint Class

Description

The `DataPoint` class represents a data point.

Inheritance Hierarchy

- `DataPoint`

Syntax

```
public class DataPoint implements Serializable
```

Properties

Name	Description
time	Gets or sets the timestamp of the monitored data. Specified in milliseconds since midnight, January 1, 1970 UTC. <code>long</code> .
readableTime	Gets or sets the human-readable timestamp of the monitored data. Specified in UTC. The time format used is "YYYY-MM-DD HH:MM:SS:MMM". <code>String</code> .
count	Gets or sets the number of values contained in this data point. <code>double</code> .
sum	Gets or sets the sum of all values contained in this data point. <code>double</code> .

Name	Description
<i>sumSquare</i>	Gets or sets the sum of all squared values contained in this data point. This parameter is useful to calculate the standard deviation and the variance. <code>double</code> .
<i>min</i>	Gets or sets the lowest of all values that are contained in this data point. <code>double</code> .
<i>max</i>	Gets or sets the highest of all values that are contained in this data point. <code>double</code> .
<i>boundCount1</i>	Gets or sets the number of values contained in this data point that lie below their corresponding lower boundaries. <code>double</code> .
<i>boundCount2</i>	Gets or sets the number of values contained in this data point that lie above their corresponding upper boundaries. <code>double</code> .

readableTime Property (DataPoint)

Class

[*DataPoint*](#).

Action

Gets or sets the human-readable timestamp of the monitored data. Specified in milliseconds since midnight, January 1, 1970 UTC.

Syntax

```
public String readableTime;
```

Access

Read and write.

count Property (DataPoint)

Class

[*DataPoint*](#).

Action

Gets or sets the number of values contained in this data point.

Syntax

```
public double count;
```

Access

Read and write.

sum Property (DataPoint)

Class

[*DataPoint*](#).

Action

Gets or sets the sum of all values contained in this data point.

Syntax

```
public double sum;
```

Access

Read and write.

sumSquare Property (DataPoint)**Class**

[DataPoint](#).

Action

Gets or sets the sum of all squared values contained in this data point. This parameter is useful to calculate the standard deviation and the variance.

Syntax

```
public double sumSquare;
```

Access

Read and write.

min Property (DataPoint)**Class**

[DataPoint](#).

Action

Gets or sets the lowest of all values that are contained in this data point.

Syntax

```
public double min;
```

Access

Read and write.

max Property (DataPoint)**Class**

[DataPoint](#).

Action

Gets or sets the highest of all values that are contained in this data point.

Syntax

```
public double max;
```

Access

Read and write.

boundCount1 Property (DataPoint)

Class

[DataPoint](#).

Action

Gets or sets the number of values contained in this data point that lie below their corresponding lower boundaries.

Syntax

```
public double boundCount1;
```

Access

Read and write.

boundCount2 Property (DataPoint)

Class

[DataPoint](#).

Action

Gets or sets the number of values contained in this data point that lie above their corresponding upper boundaries.

Syntax

```
public double boundCount2;
```

Access

Read and write.

getClientMeasureData Method (svdata)

Class

[svdata](#).

Action

Queries monitoring data of client side monitoring transactions from the database. The method's parameters are used to filter the data. All data that pass the filter are merged. For each available timestamp, one `DataPoint` is returned.

Syntax

```
dataPoints = svdata.getClientMeasureData(sessionId, measureName, projectId, transactionIds, locationIds, startTime, endTime, mergeSpan)
```

Variable	Description
<i>dataPoints</i>	A merged time series that includes all the data that pass the specified filter conditions. <i>DataPoint[]</i> .
<i>sessionId</i>	The session identifier. <i>long</i> .
<i>measureName</i>	The name of the measure to be queried. <i>String</i> .
<i>projectId</i>	The project identifier. <i>int</i> .
<i>transactionIds</i>	Identifiers of all transactions that are to be included. When this parameter is null or empty, which means that the length of the parameter is zero, all transactions are included. <i>int[]</i> .
<i>locationIds</i>	Identifiers of all locations that are to be included. When this parameter is null or empty, which means that the length of the parameter is zero, all locations are included. <i>int[]</i> .
<i>startTime</i>	The earliest timestamp the method should return data for. Specified in UTC. The time format used must be "YYYY-MMDD HH:MM:SS.MMM". <i>String</i> .
<i>endTime</i>	The latest timestamp the method should return data for. Specified in UTC. The time format used must be "YYYY-MMDD HH:MM:SS.MMM". <i>String</i> .
<i>mergeSpan</i>	The amount of time, in minutes, that data should be merged into a single timestamp. The following values are supported: <ul style="list-style-type: none">• Unmerged (real) values= 0• Quarter of an hour= 15• Hourly = 60• Daily = 1440• Weekly = 10080

getClientMeasureDataAcrossProjects Method (svdata)

Class

[svdata](#).

Action

Queries monitoring data of client side monitoring transactions across multiple projects from the database. The method's parameters are used to filter the data. All data that pass the filter are merged. For each available timestamp, one *DataPoint* is returned.

Syntax

```
dataPoints = svdata.getClientMeasureDataAcrossProjects(sessionId, measureName, projectIds, transactionIds, locationIds, startTime, endTime, mergeSpan)
```

Variable	Description
<i>dataPoints</i>	A merged time series that includes all the data that pass the specified filter conditions. <i>DataPoint[]</i> .
<i>sessionId</i>	The session identifier. <i>long</i> .
<i>measureName</i>	The name of the measure to be queried. <i>String</i> .
<i>projectIds</i>	All identifiers of the projects that should be included. When this parameter is null or empty, which means that the length of the parameter is zero, all projects are included. <i>int[]</i> .
<i>transactionIds</i>	Identifiers of all transactions that are to be included. When this parameter is null or empty, which means that the length of the parameter is zero, all transactions are included. <i>int[]</i> .
<i>locationIds</i>	Identifiers of all locations that are to be included. When this parameter is null or empty, which means that the length of the parameter is zero, all locations are included. <i>int[]</i> .
<i>startTime</i>	The earliest timestamp the method should return data for. Specified in UTC. The time format used must be "YYYY-MMDD HH:MM:SS.MMM". <i>String</i> .
<i>endTime</i>	The latest timestamp the method should return data for. Specified in UTC. The time format used must be "YYYY-MMDD HH:MM:SS.MMM". <i>String</i> .
<i>mergeSpan</i>	The amount of time, in minutes, that data should be merged into a single timestamp. The following values are supported: <ul style="list-style-type: none"> • Unmerged (real) values= 0 • Quarter of an hour= 15 • Hourly = 60 • Daily = 1440 • Weekly = 10080

getLastClientData Method (svdata)

Class

svdata.

Action

Queries the most recent monitoring data of client side monitoring transactions from the database. The method's parameters are used to filter the data. All data that pass the filter are merged. For each available timestamp, one *DataPoint* is returned.

Syntax

```
dataPoints = svdata.getLastClientData(sessionId, measureName, projectId, transactionIds, locationIds)
```

Variable	Description
<i>dataPoints</i>	A merged time series that includes all the data that pass the specified filter conditions. <i>DataPoint[]</i> .
<i>sessionId</i>	The session identifier. <i>long</i> .

Variable	Description
<i>measureName</i>	The name of the measure to be queried. <i>String</i> .
<i>projectId</i>	The project identifier. <i>int</i> .
<i>transactionIds</i>	Identifiers of all transactions that are to be included. When this parameter is null or empty, which means that the length of the parameter is zero, all transactions are included. <i>int[]</i> .
<i>locationIds</i>	Identifiers of all locations that are to be included. When this parameter is null or empty, which means that the length of the parameter is zero, all locations are included. <i>int[]</i> .

getLastData Method (svdata)

Class

[svdata](#).

Action

Queries the most recent monitoring data from the database, for both client-side and server-side transactions. The method's parameters are used to filter the data. All data that pass the filter are merged. For each available timestamp, one *DataPoint* is returned.

Syntax

```
dataPoints = svdata.getServerMeasureData(sessionId, measureName, projectId, transactionIds)
```

Variable	Description
<i>dataPoints</i>	A merged time series that includes all the data that pass the specified filter conditions. <i>DataPoint[]</i> .
<i>sessionId</i>	The session identifier. <i>long</i> .
<i>measureName</i>	The name of the measure to be queried. <i>String</i> .
<i>projectId</i>	The project identifier. <i>int</i> .
<i>transactionIds</i>	Identifiers of all transactions that are to be included. When this parameter is null or empty, which means that the length of the parameter is zero, all transactions are included. <i>int[]</i> .

getLastServerData Method (svdata)

Class

[svdata](#).

Action

Queries the most recent monitoring data of server side monitoring transactions from the database. The method's parameters are used to filter the data. All data that pass the filter are merged. For each available timestamp, one *DataPoint* is returned.

Syntax

```
dataPoints = svdata.getLastServerData(sessionId, measureName, projectId, transactionIds, , serverIds)
```

Variable	Description
<i>dataPoints</i>	A merged time series that includes all the data that pass the specified filter conditions. DataPoint[] .
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>measureName</i>	The name of the measure to be queried. <code>String</code> .
<i>projectId</i>	The project identifier. <code>int</code> .
<i>transactionIds</i>	Identifiers of all transactions that are to be included. When this parameter is null or empty, which means that the length of the parameter is zero, all transactions are included. <code>int[]</code> .
<i>serverIds</i>	Identifiers of all servers that should be included. When this parameter is null or empty, which means that the length of the parameter is zero, all servers are included.

getMeasureData Method (svdata)

Class

[svdata](#).

Action

Queries monitoring data from the database. Both client and server side monitoring data is included. The method's parameters are used to filter the data. All data that pass the filter are merged. For each available timestamp, one `DataPoint` is returned.

Syntax

```
dataPoints = svdata.getMeasureData(sessionId, measureName, projectId, startTime, endTime, mergeSpan)
```

Variable	Description
<i>dataPoints</i>	A merged time series that includes all the data that pass the specified filter conditions. DataPoint[] .
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>measureName</i>	The name of the measure to be queried. <code>String</code> .
<i>projectId</i>	The project identifier. <code>int</code> .
<i>startTime</i>	The earliest timestamp the method should return data for. Specified in UTC. The time format used must be "YYYY-MMDD HH:MM:SS.MMM". <code>String</code> .
<i>endTime</i>	The latest timestamp the method should return data for. Specified in UTC. The time format used must be "YYYY-MMDD HH:MM:SS.MMM". <code>String</code> .
<i>mergeSpan</i>	The amount of time, in minutes, that data should be merged into a single timestamp. The following values are supported: <ul style="list-style-type: none">• Unmerged (real) values= 0• Quarter of an hour= 15

Variable	Description
	<ul style="list-style-type: none"> Hourly = 60 Daily = 1440 Weekly = 10080

getServerMeasureData Method (svdata)

Class

[svdata](#).

Action

Queries monitoring data of server side monitoring transactions from the database. The method's parameters are used to filter the data. All data that pass the filter are merged. For each available timestamp, one `DataPoint` is returned.

Syntax

```
dataPoints = svdata.getServerMeasureData(sessionId, measureName, projectId, transactionIds, locationIds, serverIds, startTime, endTime, mergeSpan)
```

Variable	Description
<i>dataPoints</i>	A merged time series that includes all the data that pass the specified filter conditions. DataPoint[] .
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>measureName</i>	The name of the measure to be queried. <code>String</code> .
<i>projectId</i>	The project identifier. <code>int</code> .
<i>transactionIds</i>	Identifiers of all transactions that are to be included. When this parameter is null or empty, which means that the length of the parameter is zero, all transactions are included. <code>int[]</code> .
<i>locationIds</i>	Identifiers of all locations that are to be included. When this parameter is null or empty, which means that the length of the parameter is zero, all locations are included. <code>int[]</code> .
<i>serverIds</i>	Identifiers of all servers that should be included. When this parameter is null or empty, which means that the length of the parameter is zero, all servers are included.
<i>startTime</i>	The earliest timestamp the method should return data for. Specified in UTC. The time format used must be "YYYY-MMDD HH:MM:SS.MMM". <code>String</code> .
<i>endTime</i>	The latest timestamp the method should return data for. Specified in UTC. The time format used must be "YYYY-MMDD HH:MM:SS.MMM". <code>String</code> .
<i>mergeSpan</i>	The amount of time, in minutes, that data should be merged into a single timestamp. The following values are supported: <ul style="list-style-type: none"> Unmerged (real) values= 0 Quarter of an hour= 15 Hourly = 60 Daily = 1440

Variable	Description
	<ul style="list-style-type: none"> Weekly = 10080

getServerMeasureDataAcrossProjects Method (svdata)

Class

svdata.

Action

Queries monitoring data of server side monitoring transactions across multiple projects from the database. Both client and server-side monitoring data is included. The method's parameters are used to filter the data. All data that pass the filter are merged. For each available timestamp, one *DataPoint* is returned.

Syntax

```
dataPoints = svdata.getServerMeasureDataAcrossProjects(sessionId,  
measureName, projectIds, transactionIds, locationIds, serverIds, startTime,  
endTime, mergeSpan)
```

Variable	Description
<i>dataPoints</i>	A merged time series that includes all the data that pass the specified filter conditions. <i>DataPoint</i> [].
<i>sessionId</i>	The session identifier. long.
<i>measureName</i>	The name of the measure to be queried. String.
<i>projectIds</i>	The project identifiers. When this parameter is null or empty, which means that the length of the parameter is zero, all projects are included. int [].
<i>transactionIds</i>	Identifiers of all transactions that are to be included. When this parameter is null or empty, which means that the length of the parameter is zero, all transactions are included. int [].
<i>locationIds</i>	Identifiers of all locations that are to be included. When this parameter is null or empty, which means that the length of the parameter is zero, all locations are included. int [].
<i>serverIds</i>	Identifiers of all servers that should be included. When this parameter is null or empty, which means that the length of the parameter is zero, all servers are included.
<i>startTime</i>	The earliest timestamp the method should return data for. Specified in UTC. The time format used must be "YYYY-MMDD HH:MM:SS.MMM". String.
<i>endTime</i>	The latest timestamp the method should return data for. Specified in UTC. The time format used must be "YYYY-MMDD HH:MM:SS.MMM". String.
<i>mergeSpan</i>	The amount of time, in minutes, that data should be merged into a single timestamp. The following values are supported: <ul style="list-style-type: none"> Unmerged (real) values= 0 Quarter of an hour= 15 Hourly = 60 Daily = 1440

Variable	Description
	<ul style="list-style-type: none"> Weekly = 10080

mergeSeries Method (svdata)

Class

svdata.

Action

Merges a specified series to a single `DataPoint`.

Syntax

```
dataPoint = svdata.mergeSeries(series)
```

Variable	Description
<i>dataPoint</i>	A merged time series that includes all the data as a single <code>DataPoint</code> . The returned data point uses <code>series[0].getTime()</code> as its time. DataPoint .
<i>series</i>	The data series to merge. DataPoint[] .

svmonconfctrl

Description

This service enables the caller to query and apply changes to the active configuration.

The WSDL file of the service is available at `/services/svmonconfctrl?wsdl`.

Methods

Name	Description
addScheduleExclusion	Adds an exclusion to the specified schedule. <code>String</code> .
createMonitor	Creates a new monitor with the default monitor settings and the project schedule. <code>int</code> .
createMonitor	Creates a new monitor with the specified test script. <code>int</code> .
createMonitor	Creates a new monitor with the specified test script and user group. <code>int</code> .
deleteMonitor	Deletes the monitor with the specified identifier.
getLocationsOfSchedule	Retrieves the identifiers of the locations of a schedule. <code>int[]</code> .
getMonitorSchedule	Retrieves the schedule settings of the specified monitor. ScheduleSettings .
getMonitorSettings	Retrieves the settings of the specified monitor, for example the browser type, the name, and so on. MonitorSetting[] .

Name	Description
getMonitorSettingSpec	Retrieves detailed information about a monitor setting, for example the name, the description, the data type, and so on. MonitorSettingSpec .
getProjectSchedule	Retrieves the settings of the client-side or server-side project schedule. Schedule .
getScheduleExclusions	Retrieves all exclusions of the specified schedule. RunExclusion[] .
getTestScripts	Retrieves the names of the available test scripts for the specified essential or file pool entry. String[] .
getUsergroupNames	Retrieves the available user groups for the specified essential or file pool entry and the specified test script. String[] .
makeScheduleLocal	Duplicates the project schedule for a monitor without a monitor schedule and sets the duplicate as the monitor schedule.
redefineSchedule	Redefines the settings of the specified schedule. String .
removeSchedule	Removes the monitor schedule from the specified monitor and uses the project schedule instead.
runMonitor	Runs the specified monitor. String .
runMonitorsInProject	Runs the monitors that belong to the specified project. String .
setLocationsOfSchedule	Sets the locations of a schedule.
setMonitorState	Activates or deactivates the specified monitor. String .
updateMonitorSettings	Updates the settings of the specified monitor, for example the browser type, the name, and so on.

MonitorSetting Class

Description

The `MonitorSetting` class represents a monitor setting.

Syntax

```
public class MonitorSetting
```

Properties

Name	Description
name	Gets or sets the name of the monitor setting. String .
value	Gets or sets the value of the monitor setting. String .

name Property

Class

- [MonitorSetting](#)
- [MonitorSettingSpec](#)

Action

Gets or sets the name of the monitor setting. Can have one of the following prefixes:

- *Setting* if it is a setting, for example Name.
- *Attribute* if it is an attribute, for example defined in the bdl script.

Syntax

```
public String name;
```

Access

Read and write.

value Property

Class

- [MonitorSetting](#)
- [MonitorSettingSpec](#)

Action

Gets or sets the value of the monitor setting.

Syntax

```
public String value;
```

Access

Read and write.

MonitorSettingSpec Class

Description

The `MonitorSettingSpec` class represents a detailed monitor setting specification.

Syntax

```
public class MonitorSettingSpec
```

Properties

Name	Description
name	Gets or sets the name of the monitor setting. <code>String</code> .
description	Gets or sets the description of the monitor setting. <code>String</code> .
value	Gets or sets the value of the monitor setting. <code>String</code> .
defaultValue	<i>Optional:</i> Gets or sets the default value of the monitor setting. <code>String</code> .
dataType	Gets or sets the data type of the monitor setting. <code>String</code> .
allowedValues	<i>Optional:</i> Gets the allowed values of the monitor setting. <code>String[]</code> .

name Property

Class

- [MonitorSetting](#)
- [MonitorSettingSpec](#)

Action

Gets or sets the name of the monitor setting. Can have one of the following prefixes:

- *Setting* if it is a setting, for example Name.
- *Attribute* if it is an attribute, for example defined in the bdl script.

Syntax

```
public String name;
```

Access

Read and write.

description Property (MonitorSettingSpec)

Class

[MonitorSettingSpec](#).

Action

Gets or sets the description of the monitor setting.

Syntax

```
public String description;
```

Access

Read and write.

value Property

Class

- [MonitorSetting](#)
- [MonitorSettingSpec](#)

Action

Gets or sets the value of the monitor setting.

Syntax

```
public String value;
```

Access

Read and write.

defaultValue Property (MonitorSettingSpec)

Class

MonitorSettingSpec.

Action

Gets or sets the default value of the monitor setting.

Syntax

```
public String defaultValue;
```

Access

Read and write.

dataType Property (MonitorSettingSpec)

Class

MonitorSettingSpec.

Action

Gets or sets the data type of the monitor setting. Can also include the allowed value, for example for type `select`. The following values are allowed:

- `boolean`
- `float`
- `string`
- `password`
- `select`

Syntax

```
public String dataType;
```

Access

Read and write.

allowedValues Property (MonitorSettingSpec)

Class

MonitorSettingSpec.

Action

Gets the allowed values for the monitor setting.

Syntax

```
protected String[] allowedValues;
```

Access

Read and write.

RunExclusion Class

Description

The `RunExclusion` class represents an exclusion.

Inheritance Hierarchy

- *Entity*
 - `RunExclusion`

Syntax

```
public class RunExclusion extends Entity
```

Properties

Name	Description
<i>excludedDays</i>	Gets or sets the days of the week that should be excluded from the corresponding schedule. <code>byte</code> .
<i>dontRunFromHour</i>	Gets or sets the earliest hour of the specified day that should be excluded from the corresponding schedule. <code>int</code> .
<i>dontRunFromMinute</i>	Gets or sets the earliest minute of the specified hour that should be excluded from the corresponding schedule. <code>int</code> .
<i>dontRunHowLong</i>	Gets or sets the time in milliseconds that should be excluded from the corresponding schedule. <code>long</code> .
<i>timeZone</i>	Gets or sets the time zone. <code>String</code> .

excludedDays Property (RunExclusion)

Class

RunExclusion.

Action

Gets or sets the days of the week that should be excluded from the corresponding schedule. Each of the lower seven bits of the parameter are used to indicate that the exclusion is defined for the corresponding day of the week. The relation between bits and days of the week is as follows:

- `BIT_MONDAY = 0x01`
- `BIT_TUESDAY = 0x02`
- `BIT_WEDNESDAY = 0x04`
- `BIT_THURSDAY = 0x08`
- `BIT_FRIDAY = 0x10`
- `BIT_SATURDAY = 0x20`
- `BIT_SUNDAY = 0x40`

Syntax

```
public byte excludedDays;
```

Access

Read and write.

dontRunFromHour Property (RunExclusion)**Class**

[RunExclusion](#).

Action

Gets or sets the earliest hour of the specified day that should be excluded from the corresponding schedule.

Syntax

```
public int dontRunFromHour;
```

Access

Read and write.

dontRunFromMinute Property (RunExclusion)**Class**

[RunExclusion](#).

Action

Gets or sets the earliest minute of the specified hour that should be excluded from the corresponding schedule.

Syntax

```
public int dontRunFromMinute;
```

Access

Read and write.

dontRunHowLong Property (RunExclusion)**Class**

[RunExclusion](#).

Action

Gets or sets the time in milliseconds that should be excluded from the corresponding schedule.

Syntax

```
public long dontRunHowLong;
```

Access

Read and write.

timeZone Property (RunExclusion)

Class

[RunExclusion](#).

Action

Gets or sets the time zone.

Syntax

```
public String timeZone;
```

Access

Read and write.

Schedule Class

Description

The `Schedule` class represents a schedule.

Inheritance Hierarchy

- [Entity](#)
 - `Schedule`

Syntax

```
public class Schedule extends Entity
```

Properties

Name	Description
endTime	Gets or sets how long the monitor is to run. <code>int</code> .
startTime	Gets or sets the time in milliseconds at which the monitor is to run for the first time. <code>long</code> .
endTime	Gets or sets the time in milliseconds at which the monitor is to run for the last time. <code>long</code> .
interval	Gets or sets the time in milliseconds between two successive runs of the monitor. <code>long</code> .
totalRuns	Gets or sets the total number of runs that the monitor should execute. This is only used if the value of the <code>endTime</code> is <code>ENDTYPE_RUNCOUNT</code> . <code>int</code> .
nextTimestamp	Gets the start time of the next scheduled run. <code>long</code> .

endType Property (ScheduleSettings)

Class

- [Schedule](#)
- [ScheduleSettings](#)

Action

Gets or sets how long the monitor is to run. The following values are available:

endType	Value	Description
ENDTYPE_RUNFOREVER	0	Executes the monitor with the specified interval forever.
ENDTYPE_TIME	1	Executes the monitor with the specified interval until the specified end time is reached.
ENDTYPE_RUNCOUNT	2	Executes the monitor with the specified interval until the specified number of runs is reached.

Syntax

```
public int endType;
```

Access

Read and write.

startTime Property

Class

- [Schedule](#)
- [ScheduleSettings](#)

Action

Gets or sets the time in milliseconds at which the monitor is to run for the first time. The time is specified in milliseconds since midnight, January 1, 1970 UTC.

Syntax

```
public long startTime;
```

Access

Read and write.

endTime Property (ScheduleSettings)

Class

- [Schedule](#)
- [ScheduleSettings](#)

Action

Gets or sets the time in milliseconds at which the monitor is to run for the last time. The time is specified in milliseconds since midnight, January 1, 1970 UTC.

Syntax

```
public long endTime;
```

Access

Read and write.

interval Property (ScheduleSettings)

Class

- [Schedule](#)
- [ScheduleSettings](#)

Action

Gets or sets the time in milliseconds between two successive runs of the monitor.

Syntax

```
public long interval;
```

Access

Read and write.

totalRuns Property (ScheduleSettings)

Class

- [Schedule](#)
- [ScheduleSettings](#)

Action

Gets or sets the total number of runs that the monitor should execute. This is only used if the value of the [endType](#) is `ENDTYPE_RUNCOUNT`.

Syntax

```
public int totalRuns;
```

Access

Read and write.

nextTimestamp Property (ScheduleSettings)

Class

- [Schedule](#)
- [ScheduleSettings](#)

Action

Gets the start time of the next scheduled run. The time is specified in milliseconds since midnight, January 1, 1970 UTC.

Syntax

```
public long nextTimestamp;
```

Access

Read only.

ScheduleSettings Class

Description

The ScheduleSettings class represents the settings of a schedule.

Inheritance Hierarchy

- *Entity*
 - ScheduleSettings

Syntax

```
public class ScheduleSettings extends Entity
```

Properties

Name	Description
<i>endType</i>	Gets or sets how long the monitor is to run. <i>int</i> .
<i>startTime</i>	Gets or sets the time in milliseconds at which the monitor is to run for the first time. <i>long</i> .
<i>endTime</i>	Gets or sets the time in milliseconds at which the monitor is to run for the last time. <i>long</i> .
<i>interval</i>	Gets or sets the time in milliseconds between two successive runs of the monitor. <i>long</i> .
<i>totalRuns</i>	Gets or sets the total number of runs that the monitor should execute. This is only used if the value of the <i>endType</i> is <i>ENDTYPE_RUNCOUNT</i> . <i>int</i> .
<i>nextTimestamp</i>	Gets the start time of the next scheduled run. <i>long</i> .

addScheduleExclusion Method (svmonconfctrl)

Class

svmonconfctrl.

Action

Adds an exclusion to the specified schedule.

Syntax

```
status = sccentities.addScheduleExclusion(sessionId, scheduleId, exclusion)
```

Variable	Description
<i>status</i>	A status message that provides information regarding whether the requested process is delayed due to an unavailable application server. <i>String</i> .
<i>sessionId</i>	The session identifier. <i>long</i> .
<i>scheduleId</i>	The identifier of the schedule to which the exclusion is to be added. <i>int</i> .
<i>exclusion</i>	The exclusion to be added. RunExclusion .

createMonitor Method (svmonconfctrl)

Class

[svmonconfctrl](#).

Action

Creates a new monitor with the default monitor settings and the project schedule. The first test script and the first user group for that script are used for monitor creation.

Syntax

```
monitorId = sccentities.createMonitor(sessionId, projectId, essentialName, name, activate)
```

Variable	Description
<i>monitorId</i>	The identifier of the created monitor. <i>int</i> .
<i>sessionId</i>	The session identifier. <i>long</i> .
<i>projectId</i>	The identifier of the project for which the monitor should be created. <i>int</i> .
<i>essentialName</i>	Hierarchy path and name of the essential or file pool entry. For example <code>CustomMonitors/performerProject.ltz</code> for a file pool entry or <code>Standard/Pinger/FtpPinger</code> for the ftp pinger essential. <i>String</i> .
<i>name</i>	The name of the new monitor. <i>String</i> .
<i>activate</i>	Whether the monitor is activated after creation or not. <i>boolean</i> .

createMonitor Method (svmonconfctrl)

Class

[svmonconfctrl](#).

Action

Creates a new monitor with the specified test script. A project may contain more than one script. Monitors created with this method have the default monitor settings and the project schedule. The specified test script and the first user group for that script are used for monitor creation.

Syntax

```
monitorId = sccentities.createMonitor(sessionId, projectId, essentialName, name, testScriptName, activate)
```

Variable	Description
<i>monitorId</i>	The identifier of the created monitor. <code>int</code> .
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>projectId</i>	The identifier of the project for which the monitor should be created. <code>int</code> .
<i>essentialName</i>	Hierarchy path and name of the essential or file pool entry. For example <code>CustomMonitors/performerProject.ltz</code> for a file pool entry or <code>Standard/Pinger/FtpPinger</code> for the ftp pinger essential. <code>String</code> .
<i>name</i>	The name of the new monitor. <code>String</code> .
<i>testScriptName</i>	The name of the test script. <code>String</code> .
<i>activate</i>	Whether the monitor is activated after creation or not. <code>boolean</code> .

createMonitor Method (svmonconfctrl)

Class

[svmonconfctrl](#).

Action

Creates a new monitor with the specified test script and user group. A project may contain more than one script and at least one user group exists for each script. Monitors created with this method have the default monitor settings and the project schedule. The specified test script and the specified user group are used for monitor creation.

Syntax

```
monitorId = sccentities.createMonitor(sessionId, projectId, essentialName, name, testScriptName, userGroupName, activate)
```

Variable	Description
<i>monitorId</i>	The identifier of the created monitor. <code>int</code> .
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>projectId</i>	The identifier of the project for which the monitor should be created. <code>int</code> .
<i>essentialName</i>	Hierarchy path and name of the essential or file pool entry. For example <code>CustomMonitors/performerProject.ltz</code> for a file pool entry or <code>Standard/Pinger/FtpPinger</code> for the ftp pinger essential. <code>String</code> .
<i>name</i>	The name of the new monitor. <code>String</code> .
<i>testScriptName</i>	The name of the test script. <code>String</code> .
<i>userGroupName</i>	The name of the user group. <code>String</code> .
<i>activate</i>	Whether the monitor is activated after creation or not. <code>boolean</code> .

deleteMonitor Method (svmonconfctrl)

Class

[svmonconfctrl](#).

Action

Deletes the monitor with the specified identifier.

Syntax

```
scentities.deleteMonitor(sessionId, monitorId)
```

Variable	Description
<i>sessionId</i>	The session identifier. long.
<i>monitorId</i>	The identifier of the monitor that should be deleted. int.

getLocationsOfSchedule Method (svmonconfctrl)

Class

[svmonconfctrl](#).

Action

Retrieves the identifiers of the locations of a schedule.

Syntax

```
locationIds = scentities.getLocationsOfSchedule(sessionId, scheduleId)
```

Variable	Description
<i>locationIds</i>	The identifiers of the locations. int[].
<i>sessionId</i>	The session identifier. long.
<i>scheduleId</i>	The identifier of the schedule. int.

getMonitorSchedule Method (svmonconfctrl)

Class

[svmonconfctrl](#).

Action

Retrieves the schedule settings of the specified monitor.

Syntax

```
scheduleSettings = scentities.getMonitorSchedule(sessionId, monitorId)
```

Variable	Description
<i>scheduleSettings</i>	The schedule of the given monitor, or null if the monitor uses the project schedule. ScheduleSettings .
<i>sessionId</i>	The session identifier. long.
<i>monitorId</i>	The identifier of the monitor. int.

getMonitorSettings Method (svmonconfctrl)

Class

[svmonconfctrl](#).

Action

Retrieves the settings of the specified monitor, for example the browser type, the name, and so on.

Syntax

```
monitorSettings = scentities.getMonitorSettings(sessionId, monitorId)
```

Variable	Description
<i>monitorSettings</i>	The settings of the specified monitor. MonitorSetting[] .
<i>sessionId</i>	The session identifier. long.
<i>monitorId</i>	The identifier of the monitor. int.

getMonitorSettingSpec Method (svmonconfctrl)

Class

[svmonconfctrl](#).

Action

Retrieves detailed information about a monitor setting, for example the name, the description, the data type, and so on.

Syntax

```
monitorSettingSpec = scentities.getLocationsOfSchedule(sessionId, monitorId, settingName)
```

Variable	Description
<i>monitorSettingSpec</i>	The detailed specification of the specified monitor setting. MonitorSettingSpec .
<i>sessionId</i>	The session identifier. long.
<i>monitorId</i>	The identifier of the monitor. int.
<i>settingName</i>	The name of the setting. String.

getProjectSchedule Method (svmonconfctrl)

Class

[svmonconfctrl](#).

Action

Retrieves the settings of the client-side or server-side project schedule.

Syntax

```
schedule = sccentities.getProjectSchedule(sessionId, projectId, serverSide)
```

Variable	Description
<code>schedule</code>	The project schedule. Schedule .
<code>sessionId</code>	The session identifier. <code>long</code> .
<code>projectId</code>	The identifier of the project. <code>int</code> .
<code>serverSide</code>	When true, the project schedule for server-side monitoring is returned. When false, the project schedule for client-side monitoring is returned. <code>boolean</code> .

getScheduleExclusions Method (svmonconfctrl)

Class

[svmonconfctrl](#).

Action

Retrieves all exclusions of the specified schedule.

Syntax

```
runExclusions = sccentities.getScheduleExclusions(sessionId, scheduleId)
```

Variable	Description
<code>runExclusions</code>	The schedule exclusions. RunExclusion[] .
<code>sessionId</code>	The session identifier. <code>long</code> .
<code>scheduleId</code>	The identifier of the schedule. <code>int</code> .

getTestScripts Method (svmonconfctrl)

Class

[svmonconfctrl](#).

Action

Retrieves the names of the available test scripts for the specified essential or file pool entry.

Syntax

```
testScripts = scentities.getTestScripts(sessionId, projectId, essentialName)
```

Variable	Description
<i>testScripts</i>	The names of the test scripts. <code>String[]</code> .
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>projectId</i>	The project identifier. <code>int</code> .
<i>essentialName</i>	The hierarchy path and the name of the of the essential or the file pool entry. <code>String</code> .

getUsergroupNames Method (svmonconfctrl)

Class

[svmonconfctrl](#).

Action

Retrieves the available user groups for the specified essential or file pool entry and the specified test script.

Syntax

```
usergroupNames = scentities.getUsergroupNames(sessionId, projectId, essentialName, scriptName)
```

Variable	Description
<i>usergroupNames</i>	The names of the user groups. <code>String[]</code> .
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>projectId</i>	The project identifier. <code>int</code> .
<i>essentialName</i>	The hierarchy path and the name of the of the essential or the file pool entry. <code>String</code> .
<i>scriptName</i>	The name of the test script. <code>String</code> .

makeScheduleLocal Method (svmonconfctrl)

Class

[svmonconfctrl](#).

Action

Duplicates the project schedule for a monitor without a monitor schedule and sets the duplicate as the monitor schedule. The new monitor schedule can then be modified without influencing the project schedule. If the monitor already has a monitor schedule nothing is done.

Syntax

```
scentities.makeScheduleLocal(sessionId, monitorId)
```

Variable	Description
<i>sessionId</i>	The session identifier. <code>long</code> .

Variable	Description
<i>monitorId</i>	The identifier of the monitor. <i>int</i> .

redefineSchedule Method (svmonconfctrl)

Class

[svmonconfctrl](#).

Action

Redefines the settings of the specified project schedule or monitor schedule.

Syntax

```
status = sccentities.redefineSchedule(sessionId, schedule)
```

Variable	Description
<i>status</i>	A status message that provides information regarding whether the requested process is delayed due to an unavailable application server. <i>String</i> .
<i>sessionId</i>	The session identifier. <i>long</i> .
<i>schedule</i>	The schedule with the new settings. The identifier of the schedule must match the identifier of the schedule that you want to update. Schedule .

removeSchedule Method (svmonconfctrl)

Class

[svmonconfctrl](#).

Action

Removes the monitor schedule from the specified monitor and uses the project schedule instead. If there is no monitor schedule, nothing is done.

Syntax

```
sccentities.removeSchedule(sessionId, monitorId)
```

Variable	Description
<i>sessionId</i>	The session identifier. <i>long</i> .
<i>monitorId</i>	The identifier of the monitor. <i>int</i> .

runMonitor Method (svmonconfctrl)

Class

[svmonconfctrl](#).

Action

Runs the specified monitor.

Syntax

```
status = sccentities.runMonitor(sessionId, monitorId)
```

Variable	Description
<i>status</i>	A status message that provides information regarding whether the requested process is delayed due to an unavailable application server. <i>String</i> .
<i>sessionId</i>	The session identifier. <i>long</i> .
<i>monitorId</i>	The identifier of the monitor. <i>int</i> .

runMonitorsInProject Method (svmonconfctrl)

Class

[svmonconfctrl](#).

Action

Runs the monitors that belong to the specified project.

Syntax

```
status = sccentities.runMonitorsInProject(sessionId, monitorId)
```

Variable	Description
<i>status</i>	A status message that provides information regarding whether the requested process is delayed due to an unavailable application server. <i>String</i> .
<i>sessionId</i>	The session identifier. <i>long</i> .
<i>projectId</i>	The identifier of the project. <i>int</i> .
<i>onlyActive</i>	If true, only active monitors are scheduled to run. If false, all monitors are scheduled. <i>boolean</i> .

setLocationsOfSchedule Method (svmonconfctrl)

Class

[svmonconfctrl](#).

Action

Sets the locations of a schedule. Only locations assigned to the associated project can be assigned to the schedule.

Syntax

```
runExclusions = sccentities.setLocationsOfSchedule(sessionId, scheduleId)
```

Variable	Description
<i>sessionId</i>	The session identifier. <i>long</i> .
<i>scheduleId</i>	The identifier of the schedule. <i>int</i> .

Variable	Description
<i>locationIds</i>	The identifiers of the locations to set in the schedule. <code>int[]</code> .

setMonitorState Method (svmonconfctrl)

Class

[svmonconfctrl](#).

Action

Activates or deactivates the specified monitor.

Syntax

```
status = sccentities.setMonitorState(sessionId, monitorId)
```

Variable	Description
<i>status</i>	A status message that provides information regarding whether the requested process is delayed due to an unavailable application server. <code>String</code> .
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>monitorId</i>	The identifier of the monitor. <code>int</code> .
<i>active</i>	When true, the monitor is activated. Otherwise, the monitor is deactivated. <code>boolean</code> .

updateMonitorSettings Method (svmonconfctrl)

Class

[svmonconfctrl](#).

Action

Updates the settings of the specified monitor, for example the browser type, the name, and so on.

Syntax

```
sccentities.updateMonitorSettings(sessionId, monitorId)
```

Variable	Description
<i>sessionId</i>	The session identifier. <code>long</code> .
<i>monitorId</i>	The identifier of the monitor. <code>int</code> .
<i>monitorSettings</i>	The values of the settings to be updated. <code>MonitorSetting[]</code> .

Accessing Web Services through Java

A powerful option for accessing web services is to create SOAP calls directly from a Java application. The interface of a web service can be directly mapped to an interface in the Java programming language. Methods can then be called similarly to calling the methods of a local object.

Accessing a web service with Java involves the following steps:

1. Generating stub-classes out of the WSDL files that describe the web service interface.
2. Instantiating the stub classes in your application and invoking method calls.

These steps depend on the type of SOAP stack you are using. The SOAP stack manages the transformation of Java method calls (and its parameters) into SOAP packages, the transmission to a web server using HTTP, and the retransformation of the web service's SOAP response into Java objects.

The explanations and samples in this chapter are based on the Axis 1.1 SOAP stack from Apache.

To successfully use the Axis SOAP stack, ensure that the following JAR files are included in your CLASSPATH. These files are located in the `lib` subfolder of your Axis 1.1 installation folder:

- `axis.jar`
- `commons-discovery.jar`
- `commons-logging.jar`
- `jaxrpc.jar`
- `saaj.jar`
- `wsdl4j.jar`

The CLASSPATH must contain these files for all samples and explanations contained in this chapter.

Generating Stub Classes

Stub classes are directly generated out of the WSDL file of a web service. Axis provides the `org.apache.axis.wsdl.WSDL2Java` Java class, which handles this task.

The following program execution generates all the Java source files that are required to call methods of the web service `sccsystem`. The option `-o` specifies the destination directory of the generated Java files.

```
java org.apache.axis.wsdl.WSDL2Java -o <destination folder> http://  
www.yourFrontend.com/services/sccsystem?wsdl
```

After building the stub classes of the web service `sccentities` by using `WSDL2Java`, the `com.segue.scc` package is available in the destination folder specified using the option `-o`. This package contains the bean classes that are returned by the different method calls of the web service `sccentities` and the stub classes and the interfaces that are required to call methods of the web service `sccentities`.

Instantiating Stub Classes and Invoking Methods

Of the created stub classes, the following two are particularly useful for directly calling the corresponding web service:

- The interface that corresponds to the actual web service.
- The `ServiceLocator` class that provides methods for creating an instance of such an interface.

For the `sccentities` service, the created interface is the `MainEntities` class. The service locator class, called `MainEntitiesServiceLocator`, is located in the same package. The following sample

code shows how to acquire an instance of `MainEntities` from the locator class and how to call the `getProjects()` method:

```
MainEntities sccEntities = new MainEntitiesServiceLocator().getsccentities();
Project[] projects = sccEntities.getProjects(sessionId);
```

Sample Transaction

The following sample code represents a complete transaction using the web services of Silk Performance Manager. The sample code prints out the intervals of the schedules of all monitors, grouped by the project that each monitor is associated with.

```
public static void main(String[] args)
throws ServiceException,
RemoteException
{
    SystemService system = new SystemServiceServiceLocator().getscssystem();
    MainEntities entities = new MainEntitiesServiceLocator().getsccentities();
    MeasureEntities svEntities = new
MeasureEntitiesServiceLocator().getsventities();
    MonitorConfigControl config = new
MonitorConfigControlServiceLocator().getsvmonconfctrl();

    long sessionId = system.logonUser("admin","admin");
    Project[] projects = entities.getProjects(sessionId);

    for(int i = 0; i < projects.length; i++)
    {
        Monitor[] monitors = svEntities.getMonitors(sessionId,
projects[i].getId());

        System.out.println("Project '" + projects[i].getName() + "'");
        for(int j = 0; j < monitors.length; j++)
        {
            Schedule schedule = config.getMonitorSchedule(sessionId,
monitors[j].getId());

            System.out.print("Monitor '" + monitors[j].getName() + "': ");

            if(schedule == null)
                System.out.println("no schedule defined");
            else
                System.out.println(schedule.getInterval() + "ms");
        }
        System.out.println();
    }
}
```

Index

A

- ActionSetting class
 - sventities 61
- actionType property
 - Rule 75
- activateExecServer method
 - scentities 40
- activateProjects method
 - scentities 41
- active property
 - Entity 17
 - ExecServer 23
 - Rule 76
- addLocationToProject method
 - scentities 41
- addResourceTag method
 - scentities 41
- addRuleExpressionPart method
 - sventities 80
- addScheduleExclusion method
 - svmonconfctrl 123
- allowedValues property
 - MonitorSettingSpec 117
- AppModule class
 - sccsystem 10
- appModuleID property
 - ExecServer 23
- assignedProjects property
 - UserDetails 32

B

- beginsAt property
 - Incident 71
- boundCount1 property
 - DataPoint 106
- boundCount2 property
 - DataPoint 106
- browserId property
 - ClientSideExecutionSet 62, 70

C

- ClientSideExecutionSet class
 - sventities 62
- Condition class
 - sventities 63
- conditionId property
 - RuleExpressionPart 78
- conditionName property
 - RuleExpressionPart 78
- configuration
 - web services 8
- convertToReadableTime method
 - sccsystem 12, 13
- count property
 - DataPoint 104
- createCondition method
 - sventities 81

- createCustomIncident method
 - sventities 81
- createdAt property
 - UserGroup 34
- createdBy property
 - UserGroup 35
- createDefaultRule method
 - scentities 82
- createExecServer method
 - scentities 42
- createLocation method
 - scentities 42
- createMonitor method
 - svmonconfctrl, default settings 124
 - svmonconfctrl, test script 124
 - svmonconfctrl, user group 125
- createProject method
 - scentities 42
- createRule method
 - sventities 82
- createRuleWithEssentialAction method
 - sventities 83
- createUser method
 - scentities 43
- createUsergroup method
 - scentities 43
- CustomIncident class
 - sventities 65

D

- DataPoint class
 - svdata 103
- dataType property
 - MonitorSettingSpec 117
- dateFormat property
 - UserDetails 30
- deactivateExecServer method
 - scentities 44
- deactivateProjects method
 - scentities 44
- defaultValue property
 - MonitorSettingSpec 117
- deleteCondition method
 - sventities 83
- deleteCustomIncident method
 - sventities 84
- deleteLocations method
 - scentities 44
- deleteMonitor method
 - svmonconfctrl 126
- deleteProjects method
 - scentities 45
- deleteRule method
 - sventities 84
- deleteRuleExpressionPart method
 - sventities 85
- deleteUserGroup method

- sccentities 45
- deleteUsers method
 - sccentities 45
- description property
 - MonitorSettingSpec 116
 - NamedEntity 12
 - UserGroup 34
- dontRunFromHour property
 - RunExclusion 119
- dontRunFromMinute property
 - RunExclusion 119
- dontRunHowLong property
 - RunExclusion 119

E

- eMail property
 - UserDetails 30
- encryptAndSetPassword method
 - sccentities 46
- encryptPassword method
 - sccsystem 13
- endsAt property
 - Incident 72
- endTime property
 - Schedule 121
 - ScheduleSettings 121
- endType property
 - Schedule 121
 - ScheduleSettings 121
- Entity class
 - webservices 11
- EssentialDescription class
 - sccentities 21
- EssentialGroup class
 - sccentities 22
- EssentialSubGroup class
 - sccentities 22
- excludedDays property
 - RunExclusion 118
- execServerHostName property
 - ExecServer 24
- ExecutionLogEntry class
 - sventities 66
- ExecutionSet class
 - sventities 69

F

- firstDayOfWeek property
 - UserDetails 30
- firstName property
 - User 28
- fullPathName property
 - EssentialDescription 21
 - FilePoolEntry 27

G

- getActionEssentials method
 - sventities 85
- getActionSettings method
 - sventities 85

- getActionTypeIds method
 - sventities 86
- getAlertingLocations method
 - sventities 86
- getAllEssentials method
 - sccentities 46
- getAllMemberships method
 - sccentities 46
- getAllProjects method
 - sccentities 47
- getAllResourceTags method
 - sccentities 47
- getAllRoles method
 - sccentities 38
- getAllUserGroups method
 - sccentities 38
- getApplicationModules method
 - AppModule 14
- getClientMeasureData method
 - svdata 106
- getClientMeasureDataAcrossProjects method
 - svdata 107
- getClientSideExecutionSets method
 - sventities 86
- getCondition method
 - sventities 87
- getConditions method
 - sventities 87
- getCustomIncident method
 - sventities 88
- getCustomIncidents method
 - sventities 88
- getEssentialGroups method
 - sccentities 48
- getEssentials method
 - sccentities 48
- getEssentialSubGroups method
 - sccentities 48
- getExecServerById method
 - sccentities 49
- getExecServersOfLocation method
 - sccentities 49
- getExecutionLog method
 - sventities 89
- getFilePoolEntries method
 - sccentities 50
- getGroupById method
 - sccentities 50
- getGroupName method
 - sccentities 50
- getLastClientData method
 - svdata 108
- getLastData method
 - svdata 109
- getLastServerData method
 - svdata 109
- getLocationDetails method
 - sccentities 51
- getLocations method
 - sccentities 39
- getLocationsForProject method
 - sccentities 40

- getLocationsOfSchedule method
 - svmonconfctrl 126, 127
- getMeasureData method
 - svdata 110
- getMeasureNamesForProject method
 - sventities 90
- getMeasures method
 - sventities 90
- getMembershipsOfGroup method
 - sccentities 51
- getMembershipsOfUser method
 - sccentities 51
- getMonitors method
 - sventities 91
- getMonitorSchedule method
 - svmonconfctrl 126
- getMonitorSettings method
 - svmonconfctrl 127
- getPageURL
 - supported reports 93
- getPageURL method
 - sventities 91
 - sventities, detailed 92
- getProfiles method
 - sventities 95
- getProjectById method
 - sccentities 52
- getProjects method
 - sccentities 39
- getProjectSchedule method
 - svmonconfctrl 128
- getProjectsForUser method
 - sccentities 39
- getResultLocations method
 - sventities 95
- getRoleById method
 - sccentities 52
- getRule method
 - sventities 96
- getRuleIncidents method
 - sventities 97
 - sventities, timespan 97
- getRules method
 - sventities 96
- getRulesExpressionParts method
 - sventities 96
- getScheduleExclusions method
 - svmonconfctrl 128
- getServerMeasureData method
 - svdata 111
- getServerMeasureDataAcrossProjects method
 - svdata 112
- getServersForProject method
 - sventities 98
- getServerSideExecutionSets method
 - sventities 98
- getTestScripts method
 - svmonconfctrl 128
- getTransactionsForMonitor method
 - sventities 99
- getTransactionsForProject method
 - sventities 98

- getUserDetails method
 - sccentities 53
- getUsergroupNames method
 - svmonconfctrl 129
- getUsers method
 - sccentities 53
- groupId property
 - UserGroupMembership 36
- groupName property
 - UserGroupMembership 36

H

- hasOwnSchedule property
 - Monitor 74
- hasResultFiles property
 - ExecutionLogEntry 68
- hasResults property
 - ExecutionLogEntry 69

I

- id property
 - Entity 11
 - Role 37
 - UserGroup 33
- ID property
 - Project 18
- Incident class
 - sventities 71
- interval property
 - Schedule 122
 - ScheduleSettings 122
- invalidateInReports property
 - CustomIncident 66
- isActive property
 - Location 18
- isLocked property
 - User 28
 - UserGroup 34
- isMixedModeAuthentication method
 - sccentities 53
- isReal property
 - LocationDetails 21

J

- Java
 - web services, accessing 133

L

- Location class
 - sccentities 18
- location property
 - ExecutionLogEntry 67
- LocationDetails class
 - sccentities 19
- locationId property
 - ExecServer 24
 - ExecutionSet 71

- locationRestriction property
 - Condition 65
- login property
 - User 29
- logonUser method
 - scssystem 14

M

- makeScheduleLocal method
 - svmonconfctrl 129
- max property
 - DataPoint 105
- maxBandwidth property
 - ExecServer 24
- maxConcurrentExecutions property
 - ExecServer 25
- Measure class
 - sventities 72
- measurement property
 - Condition 64
- measureType property
 - Condition 64
- mergeSeries method
 - svdata 113
- message property
 - ExecutionLogEntry 67
- method invocation
 - web services 8
- methods
 - invoking 133
- min property
 - DataPoint 105
- Monitor class
 - sventities 73
- monitoredServerId property
 - Monitor 74
- MonitorSetting class
 - svmonconfctrl 114
- MonitorSettingSpec class
 - svmonconfctrl 115
- moveRuleExpressionPart method
 - sventities 99

N

- name property
 - ActionSetting 61
 - MonitorSetting 114, 116
 - MonitorSettingSpec 114, 116
 - NamedEntity 12
 - Role 38
 - UserGroup 33
- NamedEntity class
 - webservices 11
- nextRun property
 - Monitor 74
- nextTimestamp property
 - Schedule 122
- notifyFailMessage property
 - RuleIncident 79
- notifyFailState property

- RuleIncident 79

O

- operator property
 - Condition 64

P

- pageRefreshTime property
 - UserDetails 31
- problemAlertTimeout property
 - ExecServer 25
- projectAssignments property
 - UserGroup 35
- projectId property
 - Condition 69
 - CustomIncident 69
 - ExecutionLogEntry 69
 - ExecutionSet 69
 - Rule 69
- proxyHostName property
 - LocationDetails 19
- proxyPort property
 - LocationDetails 20
- proxyType property
 - LocationDetails 20

R

- readableTime property
 - DataPoint 104
- recurThreshold property
 - Rule 76
- recurTimespan property
 - Rule 76
- redefineSchedule method
 - svmonconfctrl 130
- removeLocationFromProject method
 - scentities 54
- removeResourceTag method
 - scentities 54
- removeSchedule method
 - svmonconfctrl 130
- rmiProxyPort property
 - ExecServer 25
- rmiProxyPortSSL property
 - ExecServer 25
- Role class
 - scentities 37
- roleId property
 - UserDetails 31
 - UserGroupMembership 36
- roleName property
 - UserGroupMembership 37
- Rule class
 - sventities 75
- RuleExpressionPart class
 - sventities 77
- roleId property
 - RuleExpressionPart 77

- RuleIncident class
 - sventities 78
- RunExclusion class
 - svmonconfctrl 118
- runMonitor method
 - svmonconfctrl 130
- runMonitorsInProject method
 - svmonconfctrl 131

S

- sccadminctrl
 - methods 58
- sccentities
 - methods 15
- sccsystem
 - methods 10
- Schedule class
 - svmonconfctrl 120
- ScheduleSettings class
 - svmonconfctrl 123
- separatorString property
 - UserDetails 31
- Server class
 - sventities 79
- ServerSideExecutionSet class
 - sventities 80
- sessions
 - creating 9
 - ending 9
 - handling 9
- setLocationsOfSchedule method
 - svmonconfctrl 131
- setMixedModeAuthentication method
 - sccentities 55
- setMonitorState method
 - svmonconfctrl 132
- setPassword method
 - sccentities 55
- setProjectState method
 - sccadminctrl 59
- setProxyUsernameAndPasswordForLocation method
 - sccentities 55
- severity property
 - Incident 72
 - Rule 77
- shortDateFormat property
 - UserDetails 32
- Silk Performance Manager
 - API 7
- SOAP
 - sessions 9
- speedId property
 - ClientSideExecutionSet 63
- startTime property
 - Schedule 121
 - ScheduleSettings 121, 122
- state property
 - ExecServer 26
 - Monitor 73
- stub classes
 - generating 133

- instantiating 133
- sum property
 - DataPoint 104
- sumSquare property
 - DataPoint 105
- svdata
 - methods 102
- sventities
 - methods 59
- svmonconfctrl
 - methods 113

T

- thresholdValue property
 - Condition 64
- time property
 - DataPoint 68
 - ExecutionLogEntry 68
- timeZone property
 - RunExclusion 120
- timeZoneId property
 - UserDetails 32
- totalRuns property
 - Schedule 122
 - ScheduleSettings 122
- transaction
 - sample 134
- Transaction class
 - sventities 80
- transaction property
 - ExecutionLogEntry 68
- transactionId property
 - Condition 70
 - ExecutionLogEntry 70
 - ExecutionSet 70

U

- updateActionSettings method
 - sventities 100
- updateCondition method
 - sventities 100
- updateCustomIncident method
 - sventities 100
- updateEssentialActionSettings method
 - sventities 101
- updateExecServer method
 - sccentities 56
- updateLocation method
 - sccentities 56
- updateMembershipsOfGroup method
 - sccentities 57
- updateMembershipsOfUser method
 - sccentities 57
- updateMonitorSettings method
 - svmonconfctrl 132
- updateProject method
 - sccentities 57
- updateRule method
 - sventities 101
- updateRuleExpressionPart method

- sventities 102
- updateUser method
 - scentities 58
- updateUserGroup method
 - scentities 58
- useCentralProxy property
 - LocationDetails 20
- User class
 - scentities 28
- UserDetails class
 - scentities 29
- UserGroup class
 - scentities 33
- UserGroupMembership class
 - scentities 35
- userId property
 - UserGroupMembership 36
- useSSL property
 - ExecServer 26

V

- value property
 - ActionSetting 62
 - MonitorSetting 115, 116
 - MonitorSettingSpec 115, 116

W

- web service sessions
 - creating 9
 - ending 9
 - handling 9
- web services
 - about 7
 - accessing, Java 133
 - overview 10
 - sessions 9
 - setup 8