MICRO FOCUS

Silk Performance Manager 21.0

API Help

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

2020-11-11

Contents

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. `long`.
encryptPassword	Takes a clear text password and returns the encrypted version. `String`.
convertToReadableTime	Converts the given time, which is the number of milliseconds since midnight, January 1, 1970 UTC, into a human readable string representation. `String`.
convertFromReadableTime	Converts the given date/time to the number of milliseconds since midnight, January 1, 1970 UTC. `long`.
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
id	Gets or sets the identifier of the entity. `int`.

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
name	Gets or sets the name of the entity. `String`.

Name	Description
description	Gets or sets the description of the entity. `String`.

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

sccsystem.

Action

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

Syntax

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

Variable	Description
unreadableTime	The time as the number of milliseconds since midnight, January 1, 1970 UTC. `long`.

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

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

encryptPassword Method (sccsystem)

Class

sccsystem.

Action

Returns the encrypted version of a password.

Syntax

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

Variable	Description
password	The encrypted password. `String`.
sessionId	The unique identifier of the actual session. `long`.
plainPwd	The plain text password. `String`.

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
applicationModules	The list of the installed application modules. `Appmodule[]`.
sessionId	The unique ID of the actual session. `long`.

logonUser Method (sccsystem)

Class

sccsystem.

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 = sccsystem.logonUser(username, plainPwd)
```

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

Example

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

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

sccentities

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/sccentities?wsdl`.

Methods

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

Name	Description
getAllProjects	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 "?". *Project[]*.
getAllResourceTags	Retrieves all resource tags of a location. `String[]`.
getAllRoles	Retrieves all roles. *Role[]*.
getAllUserGroups	Retrieves all user groups. *UserGroup[]*.
getEssentialGroups	Queries all essential groups. *EssentialGroup[]*.
getEssentials	Queries all essentials which are children of the given subgroup. *EssentialDescription[]*.
getEssentialSubGroups	Queries all Essential Sub Groups. *EssentialSubGroup[]*.
getExecServerById	Retrieves the execution server that is identified by the given ID. *ExecServer*.
getExecServersOfLocation	Retrieves the execution servers of a location. *ExecServer[]*.
getFilePoolEntries	Retrieves information about the file pool entries. *FilePoolEntry[]*.
getGroupById	Retrieves a user group by ID. *UserGroup*.
getGroupByName	Retrieves a user group by name. *UserGroup*.
getLocationDetails	Retrieves detailed information about the location specified by the ID. *LocationDetails*.
getMembershipsOfGroup	Retrieves all memberships of the specified user group. *UserGroupMembership[]*.
getMembershipsOfUser	Retrieves all user group memberships of the specified user. *UserGroupMembership[]*.
getProjectById	Retrieves the project with the given ID. *Project*.
getRoleById	Retrieves the specified role. *Role*.
getUserDetails	Retrieves details of the user with the given ID. *UserDetails*.
getUsers	Retrieves all users matching the given login name. *User[]*.
isMixedModeAuthentication	Retrieves whether mixed mode authentication is active for a user. `boolean`.
removeLocationFromProject	Removes an association from a location to a project. `boolean`.
removeResourceTag	Removes resource tags from an execution server.
setMixedModeAuthentication	Sets the authentication mode for a user.
setPassword	Sets the password for the user. The password has to be encrypted on the client. This method is for internal use only.
setProxyUsernameAndPasswordForLocation	Sets the proxy username and the password for a location identified by the *locationId*.
updateExecServer	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. `int`.
name	The name of the project. `String`.
description	The description of the project. `String`.
active	Indicates whether the project is active or not. `boolean`.
appModuleIds	The application module IDs. `int[]`.

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. `int`.
name	The name of the location. `String`.
isActive	Indicates whether at least one execution server of this location is active. `boolean`.

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
proxyHostName	Gets or sets the hostname of the proxy server. `String`.
proxyPort	Gets or sets the port of the proxy server. `int`.
proxyType	The type of the proxy server: • 0: No proxy. • 1: Http proxy. • 2: Socks proxy. `int`.
useCentralProxy	Indicates whether the central proxy shall be used or not. `boolean`.
isReal	*Deprecated:* 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. `boolean`.
appModuleID	Gets or sets the application server module ID. `int`.
execServerHostName	The hostname of this execution server. `String`.
locationId	The ID of the location to which the execution server is assigned. `int`.
maxBandwidth	The maximum bandwidth. `int`.
maxConcurrentExecutions	The maximal number of concurrent executions that are possible on this execution server. `int`.
problemAlertTimeout	Responsiveness timeout. `int`.
rmiProxyPort	Java RMI port of this execution server. `int`.
rmiProxyPortSSLJava	RMI port of this execution server for SSL. `int`.
state	Current state of this execution server. The following state are allowed: • 0: inactive. • 1: active. • 2: failure.
useSSL	Indicates whether SSL is used for connecting this execution server or not. `boolean`.

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
name	Gets or sets the name of the file pool entry. `String`.
description	Gets or sets the description of the file pool entry. `String`.
id	Gets or sets the unique identifier of the file pool entry. `int`.
fullpathname	Gets or sets the full path to the file pool entry. `String`.
isSilkTestPackage	Gets whether the file pool entry belongs to the Silk Test package. `boolean`.

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
`ID`	The unique identifier of the user. `int`.
`name`	The last name of the user. `String`.
`description`	The description of the user. `String`.
`firstName`	The first name of the user. `String`.
`isLocked`	Indicates whether the user is locked or not. `boolean`.
`login`	The login name of the user. `String`.

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. `String`.
eMail	Gets or sets the email address for notifications. `String`.
firstDayOfWeek	Gets or sets the first day of week. `int`.
pageRefreshTime	Gets or sets the delay between automated page refreshes. `int`.

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

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

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. `int`.
groupId	The ID of the user group. `int`.
roleId	The ID of the role. `int`.
groupName	The name of the user group. `String`.
roleName	The name of the role. `String`.

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. `int`.
name	The name of the role. `String`.

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 (sccentities)

Class

sccentities.

Action

Retrieves all roles.

Syntax

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

Variable	Description
roles	The roles. *Role[]*.
sessionId	The session identifier. `long`.

getAllUserGroups Method (sccentities)

Class

sccentities.

Action

Retrieves all user groups.

Syntax

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

Variable	Description
userGroups	The user groups. *`UserGroup[]`*.
sessionId	The session identifier. `long`.

getProjects Method (sccentities)

Class

`sccentities`.

Action

Retrieves all projects that are defined for a session.

Syntax

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

Variable	Description
projectList	A list of the projects that are defined for the given session. *`Project[]`*.
sessionId	The session identifier. `long`.

getProjectsForUser Method (sccentities)

Class

`sccentities`.

Action

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

Syntax

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

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

getLocations Method (sccentities)

Class

`sccentities`.

Action

Retrieves all locations that are defined for a session.

Syntax

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

Variable	Description
locationList	A list of the locations that are defined for the given session. *Location[]*.
sessionId	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
locationList	A list of the locations that are associated with the given project. *Location[]*.
sessionId	The session identifier. `long`.
projectId	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
sessionId	The session identifier. `long`.
execServerId	The execution server identifier. `int`.
locationId	The location identifier. `int`.

activateProjects Method (sccentities)

Class

sccentities.

Action

Activates the specified projects.

Syntax

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

Variable	Description
sessionId	The session identifier. `long`.
projectIds	The project identifiers. `int[]`.

addLocationToProject Method (sccentities)

Class

sccentities.

Action

Adds an association from a location to a project.

Syntax

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

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

addResourceTag Method (sccentities)

Class

sccentities.

Action

Adds a resource tag to an execution server.

Syntax

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

Variable	Description
sessionId	The session identifier. `long`.
locationId	The location identifier. `int`.
execServerId	The execution server identifier. `int`.
resourceTag	The resource tag to add to the execution server. `String`.

createExecServer Method (sccentities)

Class

sccentities.

Action

Creates a new execution server with the given properties.

Syntax

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

Variable	Description
execServerId	The identifier for the new execution server. `int`.
sessionId	The session identifier. `long`.
execServer	The execution server. *ExecServer*.

createLocation Method (sccentities)

Class

sccentities.

Action

Retrieves all locations associated with the given project.

Syntax

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

Variable	Description
locationId	The identifier for the new location. `int`.
sessionId	The session identifier. `long`.
locationDetails	The details for the new location. *LocationDetails*.

createProject Method (sccentities)

Class

sccentities.

Action

Creates a new project with the given properties.

Syntax

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

Variable	Description
projectId	The identifier for the new project. `int`.
sessionId	The session identifier. `long`.
project	The project. *`Project`*.

createUser Method (sccentities)

Class

`sccentities`.

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 = sccentities.createUser(sessionId, userDetails)
```

Variable	Description
userId	The identifier for the new user. `int`.
sessionId	The session identifier. `long`.
userDetails	The detailed information for the new user. *`UserDetails`*.

createUsergroup Method

Class

`sccentities`.

Action

Creates a new user group.

Syntax

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

Variable	Description
userGroupId	The user group identifier. `int`.
sessionId	The session identifier. `long`.

Variable	Description
userGroup	The new user group. *UserGroup*.

deactivateExecServer Method (sccentities)

Class

sccentities.

Action

Deactivates an existing execution server if it is active.

Syntax

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

Variable	Description
sessionId	The session identifier. `long`.
execServer	The identifier of the execution server. `int`.

deactivateProjects Method (sccentities)

Class

sccentities.

Action

Deactivates the projects with the given IDs.

Syntax

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

Variable	Description
sessionId	The session identifier. `long`.
projectIds	The identifiers of the projects to be deactivated. `int[]`.

deleteLocations Method (sccentities)

Class

sccentities.

Action

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

Syntax

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

Variable	Description
sessionId	The session identifier. `long`.
locationIds	The location identifiers. `int[]`.

deleteProjects Method (sccentities)

Class

sccentities.

Action

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

Syntax

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

Variable	Description
sessionId	The session identifier. `long`.
projectIds	The project identifiers. `int[]`.

deleteUsers Method (sccentities)

Class

sccentities.

Action

Deletes the specified users.

Syntax

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

Variable	Description
sessionId	The session identifier. `long`.
userIds	The identifiers of the users to be deleted. `int[]`.

deleteUserGroup Method (sccentities)

Class

sccentities.

Action

Deletes the user group identified by the identifier.

Syntax

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

Variable	Description
sessionId	The session identifier. `long`.
groupId	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
sessionId	The session identifier. `long`.
userId	The identifiers of the user. `int`.
password	The password in plain text. `String`.

getAllEssentials Method (sccentities)

Class

sccentities.

Action

Queries all essentials.

Syntax

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

Variable	Description
essentials	The essentials. *EssentialDescription[]*.
sessionId	The session identifier. `long`.

getAllMemberships Method (sccentities)

Class

sccentities.

Action

Retrieves all memberships.

Syntax

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

Variable	Description
memberships	All memberships. *UserGroupMembership*.
sessionId	The session identifier. `long`.

getAllProjects Method (sccentities)

Class

sccentities.

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=sccentities.getAllProjects(sessionId, ProjectName)
```

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

getAllResourceTags Method (sccentities)

Class

sccentities.

Action

Retrieves all resource tags of a location.

Syntax

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

Variable	Description
resourceTags	The resource tags. `String[]`.
sessionId	The session identifier. `long`.
locationId	The location identifier. `int`.

Variable	Description
execServerId	The execution server identifier. `int`.

getEssentialGroups Method (sccentities)

Class

sccentities.

Action

Queries all essential groups.

Syntax

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

Variable	Description
essentialGroups	The essential group identifiers. *EssentialGroup[]*.
sessionId	The session identifier. `long`.

getEssentials Method (sccentities)

Class

sccentities.

Action

Queries all essentials that are children of the specified subgroup.

Syntax

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

Variable	Description
essentials	The essentials that are children of the given essential. *EssentialDescription[]*.
sessionId	The session identifier. `long`.
parentId	The identifier of the parent essential. `int`.

getEssentialSubGroups Method (sccentities)

Class

sccentities.

Action

Queries all sub-groups of the specified essential group.

Syntax

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

Variable	Description
essentialSubGroups	The essential sub-group identifiers. *EssentialSubGroup[]*.
sessionId	The session identifier. `long`.
essentialGroupId	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
execServer	The execution server. *ExecServer*.
sessionId	The session identifier. `long`.
execServerId	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
execServers	The execution servers of the specified location. *ExecServer[]*.
sessionId	The session identifier. `long`.
locationId	The identifier of the location. `int`.

getFilePoolEntries Methods (sccentities)

Class

sccentities.

Action

Retrieves information about the file pool entries.

Syntax

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

Variable	Description
filePoolEntries	The file pool entries. *FilePoolEntry[]*.
sessionId	The session identifier. `long`.
projectId	The project identifier. `int`.

getGroupById Method (sccentities)

Class

sccentities.

Action

Retrieves a user group by ID.

Syntax

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

Variable	Description
userGroup	The user group with the specified identifier. *UserGroup*.
sessionId	The session identifier. `long`.
groupId	The group identifier. `int`.

getGroupByName Method (sccentities)

Class

sccentities.

Action

Retrieves a user group by name.

Syntax

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

Variable	Description
userGroup	The user group with the specified name. *UserGroup*.
sessionId	The session identifier. `long`.
groupName	The group name. `String`.

getLocationDetails Method (sccentities)

Class

sccentities.

Action

Retrieves detailed information about the location specified by the ID.

Syntax

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

Variable	Description
locationDetails	The details of the specified location. *LocationDetails*.
sessionId	The session identifier. `long`.
locationd	The location. `int`.

getMembershipsOfGroup Method (sccentities)

Class

sccentities.

Action

Retrieves all memberships of the specified user group.

Syntax

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

Variable	Description
userGroupMemberships	The memberships of the user group. *UserGroupMembership[]*.
sessionId	The session identifier. `long`.
groupId	The group identifier. `int`.

getMembershipsOfUser Method (sccentities)

Class

sccentities.

Action

Retrieves all user group memberships of the specified user.

Syntax

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

Variable	Description
userGroupMemberships	The user group memberships of the specified user. *UserGroupMemberships[]*.
sessionId	The session identifier. `long`.
userId	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
project	The project. *Project*.
sessionId	The session identifier. `long`.
projectId	The project identifier. `int`.

getRoleById Method (sccentities)

Class

sccentities.

Action

Retrieves the specified role.

Syntax

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

Variable	Description
role	The role with the specified identifier. *Role*.
sessionId	The session identifier. `long`.
roleId	The identifier of the role. `int`.

getUserDetails Method (sccentities)

Class

sccentities.

Action

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

Syntax

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

Variable	Description
userDetails	The user details of the specified user. *UserDetails*.
sessionId	The session identifier. `long`.
userId	The user identifier. `int`.

getUsers Method (sccentities)

Class

sccentities.

Action

Retrieves all users matching the given login name.

Syntax

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

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

isMixedModeAuthentication Method (sccentities)

Class

sccentities.

Action

Retrieves whether mixed mode authentication is active for a user.

Syntax

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

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

removeLocationFromProject Method (sccentities)

Class

sccentities.

Action

Removes an association from a location to a project.

Syntax

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

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

removeResourceTag method (sccentities)

Class

sccentities.

Action

Removes resource tags from an execution server.

Syntax

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

Variable	Description
sessionId	The session identifier. `long`.
locationId	The location identifier. `int`.
execServerId	The execution server identifier. `int`.

Variable	Description
resourceTag	The resource tag to be removed from the server. `String`.

setMixedModeAuthentication Method (sccentities)

Class

sccentities.

Action

Sets the authentication mode for a user.

Syntax

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

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

setPassword Method (sccentities)

Class

sccentities.

Action

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

Syntax

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

Variable	Description
sessionId	The session identifier. `long`.
userId	The user identifier. `int`.
password	The new password for the user. `String`.

setProxyUsernameAndPasswordForLocation Method (sccentities)

Class

sccentities.

Action

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

Syntax

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

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

updateExecServer Method (sccentities)

Class

sccentities.

Action

Updates the properties of an existing execution server.

Syntax

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

Variable	Description
sessionId	The session identifier. `long`.
execServer	The execution server which should be updated. *ExecServer*.

updateLocation Method (sccentities)

Class

sccentities.

Action

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

Syntax

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

Variable	Description
sessionId	The session identifier. `long`.
locationDetails	The location details. *LocationDetails*.

updateMembershipsOfGroup Method (sccentities)

Class

sccentities.

Action

Updates the memberships of a group.

Syntax

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

Variable	Description
sessionId	The session identifier. `long`.
groupId	The group identifier. `int`.
memberships	The memberships of the specified user group. *UserGroupMemberships[]*.

updateMembershipsOfUser Method (sccentities)

Class

sccentities.

Action

Updates the user group memberships of a user.

Syntax

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

Variable	Description
sessionId	The session identifier. `long`.
userId	The group identifier. `int`.
memberships	The memberships of the specified user group. *UserGroupMemberships[]*.

updateProject Method (sccentities)

Class

sccentities.

Action

Updates an existing project with the given properties.

Syntax

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

Variable	Description
sessionId	The session identifier. `long`.
project	The project to be updated. *`Project`*.

updateUser Method (sccentities)

Class

`sccentities`.

Action

Updates the properties of an existing user.

Syntax

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

Variable	Description
sessionId	The session identifier. `long`.
userDetails	The location details. *`UserDetails`*.

updateUserGroup Method (sccentities)

Class

`sccentities`.

Action

Updates an existing user group.

Syntax

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

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

Name	Description
getRuleIncidents	Returns all rule incidents for the specified project. *RuleIncident[]*.
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. `boolean`.
updateActionSettings	Updates the action settings for a non-essential action.
updateCondition	Updates a condition with the specified values. `boolean`.
updateCustomIncident	Updates a custom incident with the given values. `boolean`.
updateEssentialActionSettings	Updates the action settings for an essential action.
updateRule	Updates a rule with the specified values. `boolean`.
updateRuleExpressionPart	Update a rule expression part with the specified values. `boolean`.

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. `String`.
value	The value of the action setting. `String`.

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

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
projectId	Gets or sets the project. `int`.
invalidateInReports	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
time	Gets or sets the time that the corresponding monitor ran. Specified in milliseconds since midnight, January 1, 1970 UTC. `long`.
message	Gets or sets the readable message of the execution log entry. `String`.
severity	Gets or sets the severity of the execution log entry. The value is equal to one of the following: • SEVERITY_INFO= 0 • SEVERITY_SUCCESS= 1 • SEVERITY_WARNING= 2 • SEVERITY_ERROR= 3 `int`.
transaction	Gets or sets the name of the corresponding transaction. `String`.
location	Gets or sets the name of the location the monitor runs from. `String`.
transactionId	Gets or sets the identifier of the transaction. `String`.
locationId	Gets or sets the identifier of the location. `int`.
projectId	Gets or sets the identifier of the corresponding project. `int`.

Name	Description
hasResultFiles	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. `boolean`.
hasResults	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. `boolean`.

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
projectId	Gets or sets the project. `int`.
transactionId	Gets or sets the transaction. `int`.
locationId	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
state	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`.
nextRun	Gets or sets the next time at which the monitor should run. Specified in milliseconds since midnight, January 1, 1970 UTC. `long`.
hasOwnSchedule	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. `boolean`.
monitoredServerId	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. `int`.

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. `String`.
projectId	Gets or sets the project. `int`.
active	Gets or sets whether the rule is active. `boolean`.
recurThreshold	Gets or sets how often the expression must be matched. `short`.
recurTimespan	Gets or sets the timespan within which the expression must be matched `recurThreshold` times. `long`.
severity	Gets or sets the severity of the rule. `String`.

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. `int`.
conditionId	Gets or sets the condition. `int`.
conditionName	Gets or sets the name of the condition. `String`.
operator	Gets or sets the operator of the condition. `String`.

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
ruleId	Gets or sets the rule. `int`.
notifyFailMessage	Gets or sets the notification message. `String`.
notifyFailState	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
position	The position at which the rule expression part was added to the rule. `int`.

Variable	Description
sessionId	The session identifier. `long`.
ruleId	The rule identifier. `int`.
operator	The operator. `String`.
conditionId	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
conditionId	The position at which the rule expression part was added to the rule. `int`.
sessionId	The session identifier. `long`.
name	The name of the new condition. `String`.
transactionId	The transaction identifier. `int`.
measurement	The measurement. `String`.
operator	The operator. `String`.
thresholdValue	The value of the threshold. `double`.
measureType	The type of the measure. `String`.
locationRestriction	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
incidentId	The identifier of the new custom incident. `int`.
sessionId	The session identifier. `long`.
name	The name of the new custom incident. `String`.
description	The description of the new custom incident. `String`.
projectId	The project identifier. `int`.
invalidateInReports	Whether to invalidate the custom incident in reports or not. `boolean`.
severity	The severity of the new custom incident. `String`.
beginsAt	Where the custom incident starts. `long`.
endsAt	Where the custom incident ends. `long`.

createDefaultRule Method (sventities)

Class

sccentities.

Action

Creates a default rule with the specified default conditions.

Syntax

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

Variable	Description
ruleID	The identifier of the new rule. `int`.
sessionId	The session identifier. `long`.
monitorId	The identifier of the monitor. `int`.
measures	The measures for the new rule. `String[]`.

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

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

deleteCondition Method (sventities)

Class

sventities.

Action

Deletes the specified condition.

Syntax

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

Variable	Description
success	Whether the deletion was successful or not. `boolean`.
sessionId	The session identifier. `long`.
conditionId	The identifier of the condition. `int`.

deleteCustomIncident Method (sventities)

Class

sventities.

Action

Deletes the specified custom incident.

Syntax

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

Variable	Description
success	Whether the deletion was successful or not. `boolean`.
sessionId	The session identifier. `long`.
customIncidentId	The identifier of the custom incident. `int`.

deleteRule Method (sventities)

Class

sventities.

Action

Deletes the specified rule.

Syntax

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

Variable	Description
success	Whether the deletion was successful or not. `boolean`.
sessionId	The session identifier. `long`.
ruleId	The identifier of the rule. `int`.

deleteRuleExpressionPart Method (sventities)

Class

sventities.

Action

Deletes the rule expression part with the specified position.

Syntax

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

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

getActionEssentials Method (sventities)

Class

sventities.

Action

Retrieves all action essentials.

Syntax

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

Variable	Description
description	The descriptions of the essentials. *EssentialDescription[]*.
sessionId	The session identifier. `long`.
projectId	The project identifier. `int`.

getActionSettings Method (sventities)

Class

sventities.

Action

Retrieves all action settings for the specified rule.

Syntax

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

Variable	Description
settings	The settings of the rule. *ActionSetting[]*.
sessionId	The session identifier. `long`.
ruleId	The rule identifier. `int`.

getActionTypeIds Method (sventities)

Class

sventities.

Action

Retrieves the identifiers of the action types.

Syntax

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

Variable	Description
actionTypeIds	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
locations	The identifiers of all locations that have an alerting execution server. `int[]`.
sessionId	The session identifier. `long`.
projectId	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
executionSets	The client side execution sets. *ClientSideExecutionSet[]*.
sessionId	The session identifier. `long`.
transactionId	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
condition	The condition. *Condition*.
sessionId	The session identifier. `long`.
conditionId	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
conditions	The conditions. *Condition[]*.
sessionId	The session identifier. `long`.
projectId	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
customIncident	The custom incident. *CustomIncident*.
sessionId	The session identifier. `long`.
customIncidentId	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
customIncidents	The custom incidents in the specified project. *CustomIncident[]*.
sessionId	The session identifier. `long`.
projectId	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
customIncidents	The custom incidents in the specified project. *`CustomIncident[]`*.
sessionId	The session identifier. `long`.
projectId	The identifier of the project. `int`.
startTimeUTC	The start time for the timespan in which to retrieve all custom incidents. `long`.
endTimeUTC	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
executionLogEntries	An array of all execution log entries that suit the specified filter criteria. *`ExecutionLogEntry[]`*.
sessionId	The session identifier. `long`.
projectId	The project identifier. `int`.
from	The earliest timestamp that the method should return execution log entries for. Specified in milliseconds since midnight, January 1, 1970 UTC. `long`.
to	The latest timestamp that the method should return execution log entries for. Specified in milliseconds since midnight, January 1, 1970 UTC. `long`.
sortCol	The column by which the returned array is sorted. The numerical value must match one of the following: • 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
isAscending	Set to true to use ascending order for the returned array or set to false to use descending order. `boolean`.
filterColumns	This parameter works in combination with the *filterIds* parameter. If neither array is null, then they must be of equal length. Each element in the array *filterColumns* specifies a column that should be filtered. The corresponding element, which is the element with the same index, in the array *filterIds* specifies the value that must be

Variable	Description
	matched by all execution log entries. Each element in the array *filterColumns* must match one of the following values: • COL_STATUS = 2 • COL_TRANS_ID= 4 • COL_LOCATION_ID= 6
filterIds	This parameter works in combination with the *filterColumns* parameter. If neither array is null, then they must be of equal length. Each element in the array *filterColumns* specifies a column that should be filtered. The corresponding element, which is the element with the same index, in the array *filterIds* specifies the value that must be matched by all execution log entries. Each element in the array *filterColumns* must match one of the following values: • COL_STATUS = 2 • COL_TRANS_ID= 4 • COL_LOCATION_ID= 6
maxListSize	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 *maxListSize*. See also the description of the parameter *actualPage*.
actualPage	Selects the result page that is to be returned. See also the description of the parameter *maxListSize*.

getExecutionSets Method (sventities)

Class

sventities.

Action

Retrieves all execution sets for a given project.

Syntax

```
execSets = sventities.getExecutionSets(sessionId, projectId)
```

Variable	Description
execSets	The array of all execution sets that are associated with the given project. *ExecutionSet[]*.
sessionId	The session identifier. `long`.
projectId	The project identifier. `int`.

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
measureNames	An array of all measure names where monitoring data exists for the given project. `String[]`.
sessionId	The session identifier. `long`.
projectId	The project identifier. `int`.

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
measures	An array of all measures that are associated with the given execution set. *Measure[]*.
sessionId	The session identifier. `long`.
execSetId	The execution set identifier. `int`.

getMonitors Method (sventities)

Class

sventities.

Action

Retrieves all monitors for the given project.

Note: The binary monitor package is not included!

The binary package can be downloaded through the following URL: `http://<yourserver>/servicesExchange?<authentication-info>&hid=pmMonitorPackage&monitorId=<monitorId>` where `<authentication-info>` is one of the following two options:

- `userName=<plain-text username>&passWord=<plain-text password>` where `<plain-text username>` and `<plain-text password>` need to be replaced by valid credentials for your installation
- `sid=<session id>` where `<session id>` needs to be replaced by a valid session id retrieved via `logonUser()` and where `<monitorId>` is a valid id of a monitor in the SPM database.

> **Example**
>
> The request below will authenticate the user `admin` with the password `myAdminPwd` and download the binary monitor package of the monitor with the ID 23, if it exists.
>
> ```
> http://<yourserver>/servicesExchange?
> userName=admin&passWord=myAdminPwd&hid=pmMonitorPackage&monitorI
> d=23
> ```

Syntax

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

Variable	Description
monitors	An array of all monitors that are associated with the given project. *Monitor[]*.
sessionId	The session identifier. `long`.
projectId	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
pageURL	A URL for a certain report, for example */DEF/Monitoring/Reports?rAc&pId=2&mainTab=0*. `String`.
sessionId	The session identifier. `long`.
projectId	The project identifier. `int`.
report	The earliest timestamp that the method should return execution log entries for. Specified in milliseconds since midnight, January 1, 1970 UTC. `long`.
params	The latest timestamp that the method should return execution log entries for. Specified in milliseconds since midnight, January 1, 1970 UTC. `long`.

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

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
High Level SLA Status	SLA status for all projects as a list.	hlSLAStatus
High Level Snapshot	Snapshot for all projects as list.	hlSnapShot

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

Report	Description	Report Parameter
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
Administration Reports ExecServer Log	The page that displays the execution server logs.	adminReportsFrontendServerLog
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
profiles	The array with the profile names. `String[]`.
sessionId	The session identifier. `long`.
ProjectId	The project identifier. `int`.
essentialName	The name of the essential. `String`.

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
locations	The array of all locations where monitoring data exists for the given user. *Location[]*.
sessionId	The session identifier. `long`.
userId	The user identifier. `int`.

getRule Method (sventities)

Class

sventities.

Action

Retrieves the rule with the specified identifier.

Syntax

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

Variable	Description
rule	The rule with the specified identifier. null if no matching rule is found or no sufficient access rights exist. *Rule*.

Variable	Description
sessionId	The session identifier. `long`.
ruleId	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
ruleExpressionParts	The array of all rule expression parts that are associated to the specified rule. *RuleExpressionPart[]*.
sessionId	The session identifier. `long`.
ruleId	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
rules	The array of all rules within the specified project. null if no rules are available or no sufficient access rights exist. *Rule[]*.
sessionId	The session identifier. `long`.
projectId	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
ruleIncidents	The array of all incidents within the specified project. null if no incidents are available or no sufficient access rights exist. *RuleIncident[]*.
sessionId	The session identifier. `long`.
projectId	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
ruleIncidents	The array of all incidents within the specified project. null if no incidents are available or no sufficient access rights exist. *RuleIncident[]*.
sessionId	The session identifier. `long`.
projectId	The project identifier. `int`.
startTimeUTC	Start time in milliseconds from 1.1.1970. `long`.
endTimeUTC	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
execSets	The array of all execution sets that are associated with the given transaction. *ServerSideExecutionSet[]*.
sessionId	The session identifier. `long`.
transactionId	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
servers	The array of all infrastructure servers that are associated with the given project. *Server[]*.
sessionId	The session identifier. `long`.
projectId	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
transactions	The array of all transactions that are associated with the specified project. *Transaction[]*.
sessionId	The session identifier. `long`.
projectId	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
transactions	The array of all transactions that are associated with the specified monitor. *Transaction[]*.
sessionId	The session identifier. `long`.
monitorId	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
success	Whether the move was successful or not. `boolean`.
sessionId	The session identifier. `long`.
ruleId	The identifier of the rule to which the rule expression part belongs. `int`.
position	The current position of the rule expression part that you want to move. `int`.
destination	The position in the rule to which you want to move the rule expression part. `int`.

updateActionSettings Method (sventities)

Class

sventities.

Action

Updates the action settings for a non-essential action.

Syntax

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

Variable	Description
sessionId	The session identifier. `long`.
ruleId	The rule identifier. `int`.
actionType	The new action type. `String`.
settings	The new settings. *`ActionSetting[]`*.

updateCondition (sventities)

Class

`sventities`.

Action

Updates a condition with the specified values.

Syntax

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

Variable	Description
success	Whether the update was successful or not. `boolean`.
sessionId	The session identifier. `long`.
condition	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
success	Whether the update was successful or not. `boolean`.
sessionId	The session identifier. `long`.
customIncident	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
sessionId	The session identifier. `long`.
ruleId	The rule identifier. `int`.
essential	The name of the essential action. `String`.
settings	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
success	Whether the update was successful or not. `boolean`.
sessionId	The session identifier. `long`.
ruleId	The rule identifier. `int`.
name	The name of the rule. `String`.
active	Whether the rule is active or not. `boolean`.
recurThreshold	Specifies how many times the expression must be true to raise an incident. `short`.
recurTimespan	Specifies in which timespan the expression must be true *recurThreshold* times to raise an incident. `long`.
severity	The severity of the rule. The following values are available for this parameter: • Error

Variable	Description
	• Warning • Informational • Service Target Violation `String`.

updateRuleExpressionPart

Class

sventities.

Action

Update a rule expression part with the given values.

Syntax

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

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

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*.

Name	Description
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[]*.
getMessages	Retrieves monitor execution messages for the specified execution set within the time frame specified. *Message[]*
getResultFiles	Retrieves information of result files for the execution set and the time frame specified. *ResultFile[]*
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. `long`.
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". `String`.
count	Gets or sets the number of values contained in this data point. `double`.
sum	Gets or sets the sum of all values contained in this data point. `double`.
sumSquare	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. `double`.
min	Gets or sets the lowest of all values that are contained in this data point. `double`.

Name	Description
max	Gets or sets the highest of all values that are contained in this data point. `double`.
boundCount1	Gets or sets the number of values contained in this data point that lie below their corresponding lower boundaries. `double`.
boundCount2	Gets or sets the number of values contained in this data point that lie above their corresponding upper boundaries. `double`.

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.

Message Class

Description

The `Message` class represents a transaction message that occurred during a single run of a monitor.

Inheritance Hierarchy

- `Entity`
  - Message

Syntax

```
public class Message extends Entity
```

Properties

Name	Description
overviewMsg	The overview portion of the message. String.
detailMsg	The full message text. String.
errorClass	The error class of the message • ERROR_CLASS_PERFORMANCE = 0 • ERROR_CLASS_ACCURACY = 1 • ERROR_CLASS_AVAILABILITY = 2 • ERROR_CLASS_SYSTEM = 4 int.
count	The number of times the message has been observed. int.
time	The time at which the corresponding transaction was executed. Specified in UTC. The time format used is YYYY-MM-DD HH:MM:SS.MMM. String.
exactTime	The exact time at which the message has been observed during the run of the transaction. Specified in UTC. The time format used is YYYY-MM-DD HH:MM:SS.MMM. String.

overviewMsg Property (Message)

Class

Message.

Action

The overview portion of the message. String.

Syntax

```
public String overviewMsg;
```

Access

Read and write.

detailMsg Property (Message)

Class

Message.

Action

The full message text. String.

Syntax

```
public String detailMsg;
```

Access

Read and write.

errorClass Property (Message)

Class

Message.

Action

The error class of the message.

- ERROR_CLASS_PERFORMANCE = 0
- ERROR_CLASS_ACCURACY = 1
- ERROR_CLASS_AVAILABILITY = 2
- ERROR_CLASS_SYSTEM = 4

int.

Syntax

```
public int errorClass;
```

Access

Read and write.

count Property (Message)

Class

Message.

Action

The number of times the message has been observed. int.

Syntax

```
public int count;
```

Access

Read and write.

time Property (Message)

Class

Message.

Action

The time at which the corresponding transaction was executed. Specified in UTC. The time format used is `YYYY-MM-DD HH:MM:SS.MMM`. String.

Syntax

```
public String time;
```

Access

Read and write.

exactTime Property (Message)

Class

Message.

Action

The exact time at which the message has been observed during the run of the transaction. Specified in UTC. The time format used is `YYYY-MM-DD HH:MM:SS.MMM`. String.

Syntax

```
public String exactTime;
```

Access

Read and write.

ResultFile Class

Description

The `ResultFile` class represents the information of a result file that was created during a run of a monitor.

Inheritance Hierarchy

- `Entity`
  - NamedEntity
    - ResultFile

Syntax

```
public class ResultFile extends NamedEntity
```

Properties

Name	Description
execSetId	The identifier of the corresponding execution set. int.
time	The time at which the corresponding transaction was executed. Specified in UTC. The time format used is `YYYY-MM-DD HH:MM:SS.MMM`. String.
zippedSize	The size of the compressed result file in bytes. long.

execSetId Property (ResultFile)

Class

ResultFile.

Action

The identifier of the corresponding execution set. int.

Syntax

```
public int execSetId;
```

Access

Read and write.

time Property (ResultFile)

Class

ResultFile.

Action

The time at which the corresponding transaction was executed. Specified in UTC. The time format used is `YYYY-MM-DD HH:MM:SS.MMM`. String.

Syntax

```
public String time;
```

Access

Read and write.

zippedSize Property (ResultFile)

Class

ResultFile.

Action

The size of the compressed result file in bytes. long.

Syntax

```
public long zippedSize;
```

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
dataPoints	A merged time series that includes all the data that pass the specified filter conditions. *DataPoint[]*.
sessionId	The session identifier. `long`.
measureName	The name of the measure to be queried. `String`.
projectId	The project identifier. `int`.
transactionIds	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[]`.
locationIds	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[]`.
startTime	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`.
endTime	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`.
mergeSpan	The aggregation level to be queried from the database. The value reflects the amount of minutes that are aggregated into a single value in the time series. The method returns the actual values stored in the database for the aggregation level specified and therefore also depends on the settings of your installation for deleting old results. The following values are supported: • 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
dataPoints	A merged time series that includes all the data that pass the specified filter conditions. *DataPoint[]*.
sessionId	The session identifier. `long`.
measureName	The name of the measure to be queried. `String`.
projectIds	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. `int[]`.
transactionIds	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[]`.
locationIds	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[]`.
startTime	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`.
endTime	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`.
mergeSpan	The aggregation level to be queried from the database. The value reflects the amount of minutes that are aggregated into a single value in the time series. The method returns the actual values stored in the database for the aggregation level specified and therefore also depends on the settings of your installation for deleting old results. The following values are supported: • 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
dataPoints	A merged time series that includes all the data that pass the specified filter conditions. *DataPoint[]*.
sessionId	The session identifier. `long`.
measureName	The name of the measure to be queried. `String`.
projectId	The project identifier. `int`.
transactionIds	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[]`.
locationIds	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[]`.

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
dataPoints	A merged time series that includes all the data that pass the specified filter conditions. *DataPoint[]*.
sessionId	The session identifier. `long`.
measureName	The name of the measure to be queried. `String`.
projectId	The project identifier. `int`.
transactionIds	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[]`.

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
dataPoints	A merged time series that includes all the data that pass the specified filter conditions. *`DataPoint[]`*.
sessionId	The session identifier. `long`.
measureName	The name of the measure to be queried. `String`.
projectId	The project identifier. `int`.
transactionIds	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[]`.
serverIds	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
dataPoints	A merged time series that includes all the data that pass the specified filter conditions. *`DataPoint[]`*.
sessionId	The session identifier. `long`.
measureName	The name of the measure to be queried. `String`.
projectId	The project identifier. `int`.
startTime	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`.

Variable	Description
endTime	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`.
mergeSpan	The aggregation level to be queried from the database. The value reflects the amount of minutes that are aggregated into a single value in the time series. The method returns the actual values stored in the database for the aggregation level specified and therefore also depends on the settings of your installation for deleting old results. The following values are supported: • Unmerged (real) values= 0 • Quarter of an hour= 15 • Hourly = 60 • Daily = 1440 • Weekly = 10080

getMessages Method (svdata)

Class

svdata.

Action

Retrieves monitor execution messages for the specified execution set within the time frame specified.

Syntax

```
messages = sventities.getMessages(sessionId, execSetId, startTimeUTC,
endTimeUTC, rowLimit)
```

Variable	Description
messages	The execution messages that are associated with the given execution set and the time frame specified. *Message[]*.
sessionId	The session identifier. `long`.
execSetId	The execution set identifier. int.
startTimeUTC	The minimum time data should be returned for (in UTC). The time format used must be `YYYY-MM-DD HH:MM:SS.MMM`. String.
endTimeUTC	The maximum time data should be returned for (in UTC). The time format used must be `YYYY-MM-DD HH:MM:SS.MMM`. String.
rowLimit	The maximum number of messages to be returned. int.

getResultFiles Method (svdata)

Class

svdata.

Action

Retrieves information of result files for the execution set and the time frame specified.

Note: The result file contents is not included!

The result file contents can be downloaded via the following URL: `http://<yourserver>/servicesExchange?<authentication-info>&hid=pmResultFile&resultFileId=<resultFileId>` where `<authentication-info>` is one of the following two options:

- `userName=<plain-text username>&passWord=<plain-text password>` where `<plain-text username>` and `<plain-text password>` need to be replaced by valid credentials for your installation
- `sid=<session id>` where `<session id>` needs to be replaced by a valid session id retrieved via `logonUser()` and where `<resultFileId>` is a valid id of a result file in the SPM database.

Example

The request below will authenticate user the `admin` with the password `myAdminPwd` and download the contents of the result file with id 23, if it exists.

```
http://<yourserver>/servicesExchange?
userName=admin&passWord=myAdminPwd&hid=pmResultFile&resultFileId
=23
```

Syntax

```
resFileInfos = sventities.getResultFiles(sessionId, execSetId, startTimeUTC,
endTimeUTC, rowLimit)
```

Variable	Description
resFileInfos	The array of all result file information objects that are associated with the given execution set and the time frame specified. *`ResultFile[]`*.
sessionId	The session identifier. `long`.
execSetId	The execution set identifier. int.
startTimeUTC	The minimum time data should be returned for (in UTC). The time format used must be `YYYY-MM-DD HH:MM:SS.MMM`. String.
endTimeUTC	The maximum time data should be returned for (in UTC). The time format used must be `YYYY-MM-DD HH:MM:SS.MMM`. String.
rowLimit	The maximum number of result file information objects to be returned. int.

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
dataPoints	A merged time series that includes all the data that pass the specified filter conditions. *DataPoint[]*.
sessionId	The session identifier. `long`.
measureName	The name of the measure to be queried. `String`.
projectId	The project identifier. `int`.
transactionIds	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[]`.
locationIds	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[]`.
serverIds	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.
startTime	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`.
endTime	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`.
mergeSpan	The aggregation level to be queried from the database. The value reflects the amount of minutes that are aggregated into a single value in the time series. The method returns the actual values stored in the database for the aggregation level specified and therefore also depends on the settings of your installation for deleting old results. The following values are supported: • Unmerged (real) values= 0 • Quarter of an hour= 15 • Hourly = 60 • Daily = 1440 • 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
dataPoints	A merged time series that includes all the data that pass the specified filter conditions. *DataPoint[]*.
sessionId	The session identifier. `long`.
measureName	The name of the measure to be queried. `String`.
projectIds	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[]`.
transactionIds	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[]`.
locationIds	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[]`.
serverIds	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.
startTime	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`.
endTime	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`.
mergeSpan	The aggregation level to be queried from the database. The value reflects the amount of minutes that are aggregated into a single value in the time series. The method returns the actual values stored in the database for the aggregation level specified and therefore also depends on the settings of your installation for deleting old results. The following values are supported: • Unmerged (real) values= 0 • Quarter of an hour= 15 • Hourly = 60 • Daily = 1440 • Weekly = 10080

mergeSeries Method (svdata)

Class

svdata.

Action

Merges a specified series to a single `DataPoint`.

Syntax

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

Variable	Description
dataPoint	A merged time series that includes all the data as a single `DataPoint`. The returned data point uses *series*[0].`getTime`() as its time. *`DataPoint`*.
series	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. `String`.
`createMonitor`	Creates a new monitor with the default monitor settings and the project schedule. `int`.
`createMonitor`	Creates a new monitor with the specified test script. `int`.
`createMonitor`	Creates a new monitor with the specified test script and user group. `int`.
`deleteMonitor`	Deletes the monitor with the specified identifier.
`getLocationsOfSchedule`	Retrieves the identifiers of the locations of a schedule. `int[]`.
`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[]`*.
`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`.

Name	Description
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. `String`.
description	Gets or sets the description of the monitor setting. `String`.
value	Gets or sets the value of the monitor setting. `String`.
defaultValue	*Optional:* Gets or sets the default value of the monitor setting. `String`.
dataType	Gets or sets the data type of the monitor setting. `String`.
allowedValues	*Optional:* Gets the allowed values 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.

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

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
`endType`	Gets or sets how long the monitor is to run. `int`.
`startTime`	Gets or sets the time in milliseconds at which the monitor is to run for the first time. `long`.
`endTime`	Gets or sets the time in milliseconds at which the monitor is to run for the last time. `long`.
`interval`	Gets or sets the time in milliseconds between two successive runs of the monitor. `long`.
`totalRuns`	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. `int`.
`nextTimestamp`	Gets the start time of the next scheduled run. `long`.

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	Value	Description
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
endType	Gets or sets how long the monitor is to run. `int`.
startTime	Gets or sets the time in milliseconds at which the monitor is to run for the first time. `long`.
endTime	Gets or sets the time in milliseconds at which the monitor is to run for the last time. `long`.
interval	Gets or sets the time in milliseconds between two successive runs of the monitor. `long`.
totalRuns	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. `int`.
nextTimestamp	Gets the start time of the next scheduled run. `long`.

addScheduleExclusion Method (svmonconfctrl)

Class

svmonconfctrl.

Action

Adds an exclusion to the specified schedule.

Syntax

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

Variable	Description
status	A status message that provides information regarding whether the requested process is delayed due to an unavailable application server. `String`.
sessionId	The session identifier. `long`.
scheduleId	The identifier of the schedule to which the exclusion is to be added. `int`.
exclusion	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
monitorId	The identifier of the created monitor. `int`.
sessionId	The session identifier. `long`.
projectId	The identifier of the project for which the monitor should be created. `int`.
essentialName	Hierarchy path and name of the essential or file pool entry. For example `CustomMonitors/performerProject.ltz` for a file pool entry or `Standard/Pinger/FtpPinger` for the ftp pinger essential. `String`.
name	The name of the new monitor. `String`.
`activate`	Whether the monitor is activated after creation or not. `boolean`.

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

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

deleteMonitor Method (svmonconfctrl)

Class

svmonconfctrl.

Action

Deletes the monitor with the specified identifier.

Syntax

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

Variable	Description
sessionId	The session identifier. `long`.
monitorId	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 = sccentities.getLocationsOfSchedule(sessionId, scheduleId)
```

Variable	Description
locationIds	The identifiers of the locations. `int[]`.
sessionId	The session identifier. `long`.
scheduleId	The identifier of the schedule. `int`.

getMonitorSchedule Method (svmonconfctrl)

Class

svmonconfctrl.

Action

Retrieves the schedule settings of the specified monitor.

Syntax

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

Variable	Description
scheduleSettings	The schedule of the given monitor, or null if the monitor uses the project schedule. *ScheduleSettings*.
sessionId	The session identifier. `long`.
monitorId	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 = sccentities.getMonitorSettings(sessionId, monitorId)
```

Variable	Description
monitorSettings	The settings of the specified monitor. *MonitorSetting[]*.
sessionId	The session identifier. `long`.
monitorId	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 = sccentities.getLocationsOfSchedule(sessionId, monitorId, settingName)
```

Variable	Description
monitorSettingSpec	The detailed specification of the specified monitor setting. *MonitorSettingSpec*.
sessionId	The session identifier. `long`.
monitorId	The identifier of the monitor. `int`.
settingName	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
schedule	The project schedule. *Schedule*.
sessionId	The session identifier. `long`.
projectId	The identifier of the project. `int`.
serverSide	When true, the project schedule for server-side monitoring is returned. When false, the project schedule for client-side monitoring is returned. `boolean`.

getScheduleExclusions Method (svmonconfctrl)

Class

svmonconfctrl.

Action

Retrieves all exclusions of the specified schedule.

Syntax

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

Variable	Description
runExclusions	The schedule exclusions. *RunExclusion[]*.
sessionId	The session identifier. `long`.
scheduleId	The identifier of the schedule. `int`.

getTestScripts Method (svmonconfctrl)

Class

svmonconfctrl.

Action

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

Syntax

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

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

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 = sccentities.getUsergroupNames(sessionId, projectId,
essentialName, scriptName)
```

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

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

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

Variable	Description
sessionId	The session identifier. `long`.
monitorId	The identifier of the monitor. `int`.

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
status	A status message that provides information regarding whether the requested process is delayed due to an unavailable application server. `String`.
sessionId	The session identifier. `long`.

Variable	Description
schedule	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
sessionId	The session identifier. `long`.
monitorId	The identifier of the monitor. `int`.

runMonitor Method (svmonconfctrl)

Class

`svmonconfctrl`.

Action

Runs the specified monitor.

Syntax

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

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

runMonitorsInProject Method (svmonconfctrl)

Class

`svmonconfctrl`.

Action

Runs the monitors that belong to the specified project.

Syntax

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

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

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
sessionId	The session identifier. `long`.
scheduleId	The identifier of the schedule. `int`.
locationIds	The identifiers of the locations to set in the schedule. `int[]`.

setMonitorState Method (svmonconfctrl)

Class

svmonconfctrl.

Action

Activates or deactivates the specified monitor.

Syntax

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

Variable	Description
status	A status message that provides information regarding whether the requested process is delayed due to an unavailable application server. `String`.
sessionId	The session identifier. `long`.

Variable	Description
monitorId	The identifier of the monitor. `int`.
active	When true, the monitor is activated. Otherwise, the monitor is deactivated. `boolean`.

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
sessionId	The session identifier. `long`.
monitorId	The identifier of the monitor. `int`.
monitorSettings	The values of the settings to be updated. *MonitorSetting[]*.

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.

Theses 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().getsccsystem();
  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

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V

W