

# Aufruf von Webservices per Javascript

## Beschreibung

Webservices können per Script aufgerufen werden und dienen dazu Informationen auszulesen, Daten zu setzen, Aufgaben zu assignen, etc.

## Webservice Name

Eine Liste der Webservice Namen kann im Firebug eingesehen werden. Je nach ausgewähltem Webservice Namen findet man dann die vorhandenen Methoden.

webService	Object { lastSOAPCall=(), ArchivistManager=(), BlogManager=(), mehr... }
ArchivistManager	Object { url="http://localhost:8080/loom-es/ArchivistManager", ns="http://www.loom-gmbh.de/ArchivistManager", archiveEndedProcessInstances-function(), mehr... }
BlogManager	Object { url="http://localhost:8080/loom-es/BlogManager", ns="http://www.loom-gmbh.de/BlogManager", createBlogArticle-function(), mehr... }
BusinessCalendar	Object { url="http://localhost:8080/loom-es/BusinessCalendar", ns="http://www.loom-gmbh.de/BusinessCalendar", createHolidayToCalendarProfile-function(), mehr... }
DMSManager	Object { url="http://localhost:8080/loom-es/DMSManager", ns="http://www.loom-gmbh.de/DMSManager", getFormFields-function(), mehr... }
DbConnectionManager	Object { url="http://localhost:8080/loom-es/DbConnectionManager", ns="http://www.loom-gmbh.de/DbConnectionManager", getQueries-function(), mehr... }
DemotManager	Object { url="http://localhost:8080/loom-es/DemotManager", ns="http://www.loom-gmbh.de/DemotManager", createDemotEntity-function(), mehr... }
DocumentManager	Object { url="http://localhost:8080/loom-es/DocumentManager", ns="http://www.loom-gmbh.de/DocumentManager", convertProcessDocumentsToPDF-function(), mehr... }
EffortManager	Object { url="http://localhost:8080/loom-es/EffortManager", ns="http://www.loom-gmbh.de/EffortManager", createEffortForTaskInstance-function(), mehr... }
EntityViewManager	Object { url="http://localhost:8080/loom-es/EntityViewManager", ns="http://www.loom-gmbh.de/EntityViewManager", getViewsForEntityId-function(), mehr... }
GenericEntityManager	Object { url="http://localhost:8080/loom-es/GenericEntityManager", ns="http://www.loom-gmbh.de/GenericEntityManager", archive-function(), mehr... }
IdentityManager	Object { url="http://localhost:8080/loom-es/IdentityManager", ns="http://www.loom-gmbh.de/IdentityManager", getUsersFromGroupWithGroupedCator-function(), mehr... }
IssueManager	Object { url="http://localhost:8080/loom-es/IssueManager", ns="http://www.loom-gmbh.de/IssueManager", assignMe-function(), mehr... }
LicenceManager	Object { url="http://localhost:8080/loom-es/LicenceManager", ns="http://www.loom-gmbh.de/LicenceManager", checkLicenceForNewProcessDefinition-function(), mehr... }
LinkManager	Object { url="http://localhost:8080/loom-es/LinkManager", ns="http://www.loom-gmbh.de/LinkManager", getWebByTaskID-function(), mehr... }
LookupManager	Object { url="http://localhost:8080/loom-es/LookupManager", ns="http://www.loom-gmbh.de/LookupManager", createLookupForLookupGroup-function(), mehr... }
LoomEmailManager	Object { url="http://localhost:8080/loom-es/LoomEmailManager", ns="http://www.loom-gmbh.de/LoomEmailManager", getEmail-function(), mehr... }
LoomInitManager	Object { url="http://localhost:8080/loom-es/LoomInitManager", ns="http://www.loom-gmbh.de/LoomInitManager", initDB-function(), mehr... }
ModelerManager	Object { url="http://localhost:8080/loom-es/ModelerManager", ns="http://www.loom-gmbh.de/ModelerManager", createModel-function(), mehr... }
NodeManager	Object { url="http://localhost:8080/loom-es/NodeManager", ns="http://www.loom-gmbh.de/NodeManager", findActualSectorNodeMappingByNode-function(), mehr... }
PortalManager	Object { url="http://localhost:8080/loom-es/PortalManager", ns="http://www.loom-gmbh.de/PortalManager", addCadgetToPortalTab-function(), mehr... }
ProcessDefinitionManager	Object { url="http://localhost:8080/loom-es/ProcessDefinitionManager", ns="http://www.loom-gmbh.de/ProcessDefinitionManager", findLatestParticipatingProcessDefinitions-function(), mehr... }
ProcessDefinitionTemplateManager	Object { url="http://localhost:8080/loom-es/ProcessDefinitionTemplateManager", ns="http://www.loom-gmbh.de/ProcessDefinitionTemplateManager", getProcessDefinitionTemplateList-function(), mehr... }
ProcessInstanceManager	Object { url="http://localhost:8080/loom-es/ProcessInstanceManager", ns="http://www.loom-gmbh.de/ProcessInstanceManager", addProcessContextEntity-function(), mehr... }
ProcessVariableManager	Object { url="http://localhost:8080/loom-es/ProcessVariableManager", ns="http://www.loom-gmbh.de/ProcessVariableManager", getProcessContextEntity-function(), mehr... }
SearchManager	Object { url="http://localhost:8080/loom-es/SearchManager", ns="http://www.loom-gmbh.de/SearchManager", searchByClassNames-function(), mehr... }
SectionNodeMappingManager	Object { url="http://localhost:8080/loom-es/SectionNodeMappingManager", ns="http://www.loom-gmbh.de/SectionNodeMappingManager", getAIMyLocksOnProcessInstance-function(), mehr... }
SuperClientManager	Object { url="http://localhost:8080/loom-es/SuperClientManager", ns="http://www.loom-gmbh.de/SuperClientManager", createSuperClient-function(), mehr... }
SystemManager	Object { url="http://localhost:8080/loom-es/SystemManager", ns="http://www.loom-gmbh.de/SystemManager", changePassword-function(), mehr... }
TaskManager	Object { url="http://localhost:8080/loom-es/TaskManager", ns="http://www.loom-gmbh.de/TaskManager", addAdhocTaskInstance-function(), mehr... }
TimeServiceManager	Object { url="http://localhost:8080/loom-es/TimeServiceManager", ns="http://www.loom-gmbh.de/TimeServiceManager", createTimeInfo-function(), mehr... }
UpdateManager	Object { url="http://localhost:8080/loom-es/UpdateManager", ns="http://www.loom-gmbh.de/UpdateManager", update-function(), mehr... }
lastSOAPCall	[ null, function() ]

## Webservice Methode

Nach der Auswahl eines Namens findet man die Methoden. Hier am Beispiel des IdentityManagers

▼ IdentityManager	Object { url="http://localhost:8080/loam-ws/IdentityManager", ns="http://www.loam-gmbh.de/IdentityManager", getUsersFromGroupWithGroupIndicator-function(), mehr... }
ns	"http://www.loam-gmbh.de/IdentityManager"
url	"http://localhost:8080/loam-ws/IdentityManager"
▶ addIdentityToGroup	function()
▶ addIdentityToGroupInRole	function()
▶ createAbsence	function()
▶ createAbsenceRuleForUser	function()
▶ createGroup	function()
▶ createRole	function()
▶ createUnconfirmedUser	function()
▶ createUser	function()
▶ findMembershipsByGroupAndUser	function()
▶ getAllAbsenceRulesOfUser	function()
▶ getAllAbsencesOfUser	function()
▶ getAllActiveUsers	function()
▶ getAllGroups	function()
▶ getAllIdentities	function()
▶ getAllUserReferences	function()
▶ getAllUsers	function()
▶ getCurrentAbsence	function()
▶ getCurrentClient	function()
▶ getCurrentGroups	function()
▶ getCurrentIdentities	function()
▶ getCurrentUser	function()
▶ getCurrentUserProfileAsWSEntity	function()
▶ getCurrentUserProfileImplementation	function()
▶ getGroupByName	function()
▶ getGroupByUser	function()
▶ getIdentitiesByExpression	function()
▶ getIdentityTree	function()
▶ getMyPermissions	function()
▶ getPermissionsForIdentity	function()
▶ getRepresentativeForProcessDefinitionAndUser	function()
▶ getRolesForIdentity	function()
▶ getUserByGroupAndRole	function()
▶ getUserByName	function()
▶ getUsersByGroup	function()
▶ getUsersCurrentlyLoggedIn	function()
▶ getUsersFromGroupWithGroupIndicator	function()
▶ importUsersFromLdap	function()
▶ importUsersFromLdapByStep	function()
▶ isUserInRole	function()
▶ login	function()
▶ logout	function()
▶ removeMembership	function()
▶ setRoleToIdentity	function()
▶ switchUserRoleBlock	function()
IssueManager	Object { url="http://localhost:8080/loam-ws/IssueManager", ns="http://www.loam-gmbh.de/IssueManager", assignMe-function(), mehr... }

## Parameter

Per Klick auf die Methode kann man sehen ob und welche Parameter mit übergeben werden müssen.

## Beispiel

In diesem Beispiel wird im IdentityManager die Methode `getUserByName` aufgerufen. Diese Methode ruft eine Callbackmethode (`userCallback`) im Script auf. Die Callback bekommt den gefundenen User mit übergeben. Innerhalb der Callback kann mit den Informationen des bekommenen Users gearbeitet werden.

```
var userCallback=function(userGot){
  //do sth
}
webService.IdentityManager.getUserByName("tim.support", userCallback);
```

From: <https://wiki.tim-solutions.de/> - **TIM Wiki** / **NEW TIM 6 Documentation**

Permanent link: <https://wiki.tim-solutions.de/doku.php?id=software:tim:webservices>

Last update: **2021/07/01 09:52**



