

Developer Guide for MongoDB User Store

In this document contains all the important methods in MongoDB User Store and their roles in user store

According to the behaviour i categorized those methods into three categories such as,

1. Read Methods.
2. Write Methods.
3. Read - Write Methods.

So first let's look into the the read methods available in MongoDB User Store

Method Name	Return type	Available Parameters	Use
getUserPropertyValues	Map	String userName, String[] propertyNames, String profileName	get all the user claim values of belongs to a given profile of a given user
doCheckExistingRole	boolean	String roleName	check whether given role exists in MongoDB user store and return true if exists else false
isExistingMongoDBRole	boolean	RoleContext context	check for existing role in MongoDB user store role context and return true if exists else false
isValueExisting	boolean	String mongoQuery, DB db, Map params	check whether a given values exists in MongoDB by executing provided query and return true if exists else return false
doCheckExistingUser	boolean	String userName	check whether a given user exists in user store and return

			true if exists else return false
getUserListFromProperties	String[]	String property, String value, String profileName	retrieve all the users in user store who has the given claim value in their profile
doAuthenticate	boolean	String userName, Object credential	authenticate user with the given credentials and return true or false this method trigger at when logged into IS or any connected application to identity server
getProperty	String	DB dbConnection,int userId	retrieve user claim value to given user id
doGetExternalRoleListOfUser	String[]	String userName, String filter	retrieve all the roles of a given user in user store
getStringValuesFromDatabase	String[]	String mongoQuery, Map params,boolean findStatus,boolean multipleLookUps	retrieve all result as string from MongoDB database those findstatus is to decide whether it's a simple find or aggregate and multipleLookUps variable it to define whether it's a join or a single collection result
getDistinctStringValues	String[]	String mongoQuery, Map params	retrieve all distinct result as string from MongoDB database

doGetSharedRoleListOfUser	String[]	String userName,String tenantDomain, String filter	retrieve all shared roles from user store
getRoleNamesWithD	String[]	String mongoQuery,	retrieve all roles in

omain		String userName, int tenantId,boolean appendDn	user store with appending the user store domain to front of the role this appendDn variable is to check whether to domain append or not
getUserIDS	int[]	DB dbConnection,String[] userList	This is newly implemented method by me only for MongoDBUserStore Manager this will return array of user ids of given username list
doGetRoleNames	String[]	String filter, int maxItemLimit	get role names according to the given filter from userstore if filter contain "*" it will return all the roles and maxItemLimit is to limit number of roles display per page
doListUsers	String[]	String[] userNames	get all users according to the given filter from userstore if filter contain "*" it will return all the roles and maxItemLimit is to limit number of roles display per page
doGetDisplayNames ForInternalRole	String[]	String tenantDomain, String filter, int maxItemLimit	this is a get method which return all userNames
doGetSharedRoleNa mes	String[]	RoleContext ctx, String filter	retrieve all the shared roles of a given tenantDomain and

			display according to the filters and limit
getUserListOfMongoDBRole	String[]	String userName	retrieve all the roles in current role context according to filter,filter "*" means for all domains
getProfileNames	String[]	-	retrieve all the profiles of given user

getAllProfileNames	String[]	-	retrieve all the profiles in MongoDB user store
isReadOnly	boolean	-	check whether a user store is read only
getUserId	int	String userName	retrieve the userId of a given username from user store
getTenantId	int	String userName	retrieve the tenantId of a given username from user store
getProperties	Map	Tenant tenant	retrieve the properties of a given tenant from user store
isMultipleProfilesAllowed	boolean	String userName, String token	set true if multiple profiles allow else set false,default will be true
isValidRememberMeToken	boolean	-	check whether a given token is valid for a given user and return true if valid else false
getDefaultUserStoreProperties	Properties	Tenant tenant	- retrieve all default properties(Mandatory ,Optional and Advanced) for userstore

getProperties	Map	-	retrieve all the properties of tenant
isBulkImportSupported	boolean	-	return true for allow bulk import else false

getRealmConfiguration	RealmConfiguration	-	return realmConfiguration of user store
getRolesIDS	int[]	String[] roles	this is newly implemented method only for MongoDBUserStoreManager retrieve array of roleids of given role list
checkExistingUserName	boolean	String userName	this method will check whether user exists in user store
isExistingRememberMeToken	boolean	String userName, String token	check whether the given remember me token is already exists for a given user in user store
isExistingRole	boolean	String roleName, boolean shared	check whether the given rolename is in user store

isSharedGroupEnabled	boolean	-	check for shared groups enabled for user store
getMyDomainName	String	-	get the domain name of current user

Up to now we consider the read methods available in MongoDBUserStoreManager now let's look into write methods available in MongoDBUserStoreManager

Method Name	Return Type	Available Parameters	Use
preparePassword	String	String password, String saltValue	this method will salt the given password with given salt value,it will salt password according to hashing method provided in userstore
doAddUser	void	String userName, Object credential, String[] roleList, Map claims, String profileName, boolean requirePasswordChange	this method will responsible of adding new user to user store by invoking persistUser method
doUpdateCredential	void	String userName, Object newCredential, Object oldCredential	this method will responsible of update user credentials
doUpdateCredentialByAdmin	void	String userName, Object newCredential	this method will responsible of update admin user credentials
updateStringValuesToDatabase	void	DB dbConnection, String mongoQuery,Map params	update MongoDB database with provided String values by executing given MongoDB query
updateUserClaimValuesToDatabase	void	DB dbConnection,Map map,boolean isUpdateTrue	update MongoDB database with provided user claim values isUpdateTrue method will derieve whether this is a update or insert
deleteStringValuesFromDatabase	void	DB dbConnection, String mongoQuery,Object.. params	delete the documents content given records from database

doDeleteUser	void	String userName	delete the given user from userstore
doDeleteUserClaimValue	void	String userName, String claimURI, String profileName	delete the given user claim value from given user's given profile
deleteProperty	void	DB dbConnection, String userName, String property, String profileName	delete the given property from given user's specified profile
doDeleteUserClaimValues	void	String userName, String[] claims, String profileName	delete the all the given claim values from specified profile in given user
doUpdateUserListOfRole	void	String roleName, String deletedUsers[], String[] newUsers	update the assigned user list for a given role
doUpdateRoleListOfUser	void	String userName, String[] deletedRoles, String[] newRoles	update the assigned role list for a given user
doAddRole	void	String roleName, String[] userList, boolean shared	add new role to user store
doAddSharedRole	void	String roleName, String[] userList	add new shared role to user store
doDeleteRole	void	String roleName	delete given role from user store
doUpdateRoleName	void	String roleName, String newRoleName	update exists rolename with a given name
updateUserValue	void	DB connection,String query,Map map	update the database with user values provided by executing the given query
addProperty	void	DB dbConnection, Map map	add given user claim value to user store

updateProperty	void	DB dbConnection, Map map	update given user claim value in user store
persistDomain	void	-	Add user store domain to main database
doInitialSetup	void	-	initialize hybridrolemanager and systemuserrolemana ger
addUser	void	String userName, Object credential, String[] roleList,Map claims, String profileName	add new user to user store
addInternalRole	void	String roleName, String[] userList,org.wso2.car bon.user.api.Permissi on[] permission	add internal role to primary database
filterSharedRoles	void	List sharedRoles, String tenantDomain	filter shared roles from roles available in user store
doInitialUserAdding	void	-	add hybriduser ,systemuser to IS
deletePersistedDomain	void	String domain	delete the given domain from system
updatePersistedDomain	void	String oldDomain, String newDomain	update the domain name in system
addRole	void	String roleName, String[] userList,org.wso2.car bon.user.api.Permissi on[] permissions	add new role to user store

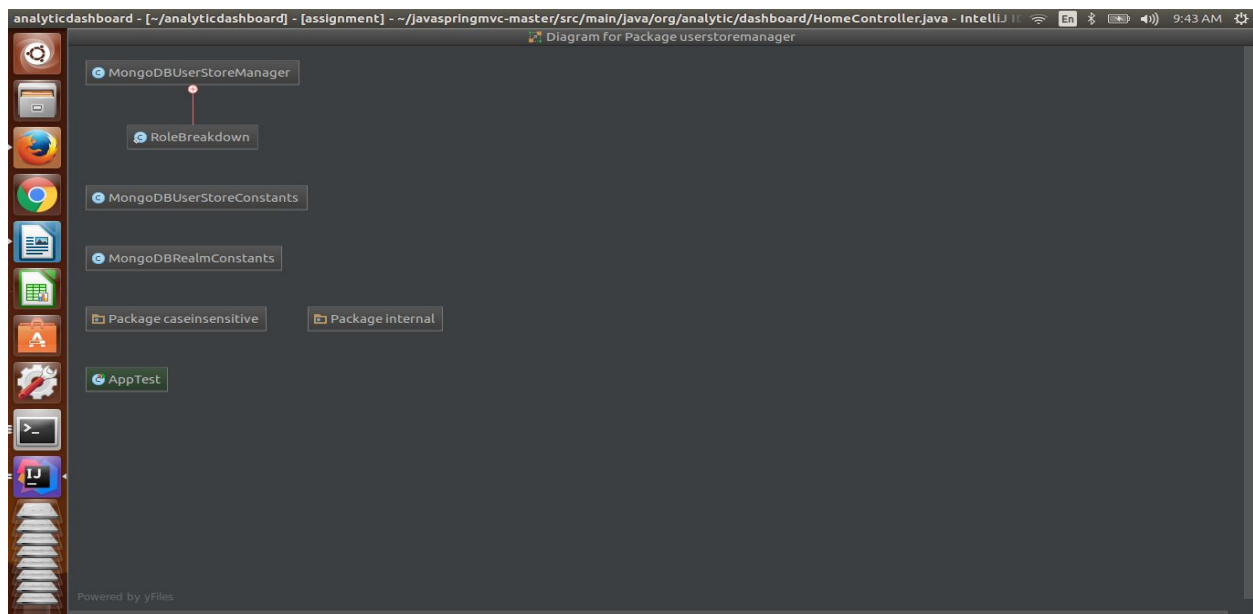
Now finally let's look into the Read - Write Methods Available in MongoDBUserStoreManager

Method Name	Return Type	Parameters	Use
createRoleContext	RoleContext	String roleName	create a new role context for the given user and return it
loadUserStoreSpecificDataSource	DB	-	load user store datasource
persistUser	DB	String userName, Object credential, String[] roleList, Map claims, String profileName, boolean requirePasswordChange	check for username,password validity and check whether user exists and then insert new user to user store

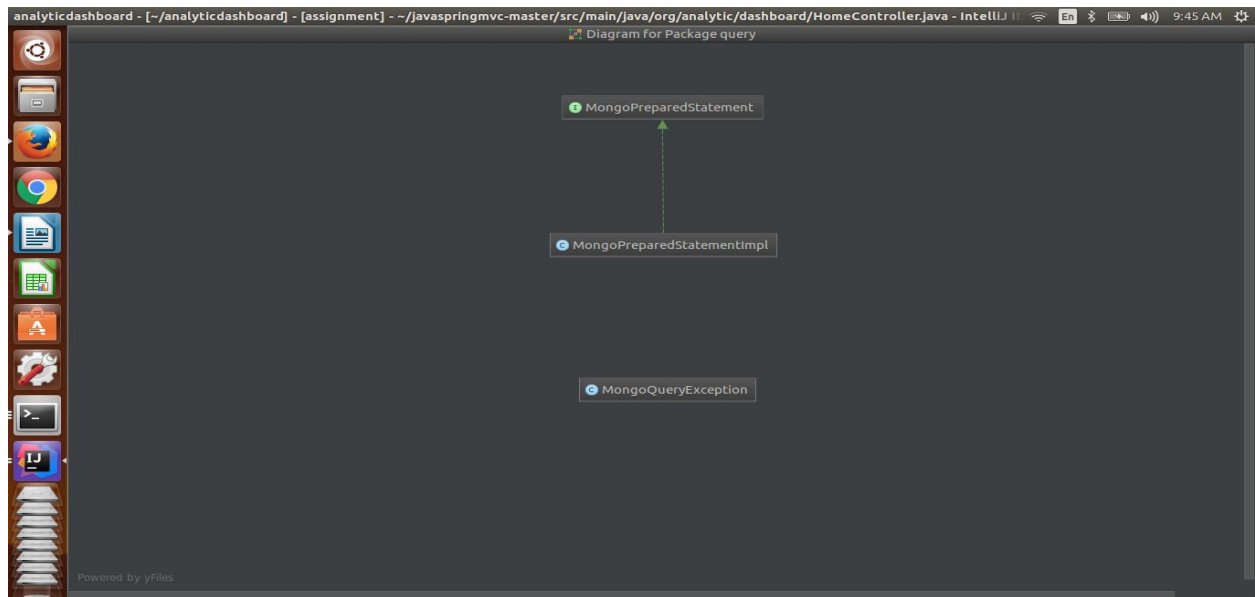
For more details about the methods and classes in MongoDBUserStoreManager you can refer the java doc[2] .

As described in above those are the important methods used in MongoDBUserStoreManager
Class Diagram of MongoDBUserStoreManager

[org.wso2.carbon.mongodb.userstoremanager package](#)



org.wso2.carbon.mongodb.query.package



now let's look into their behaviour of some of the methods used in `MongoDBUserStoreManager` in `RemoteUserStoreManager` SOAP service in SOAP UI you can check these in soap ui once you enable admin services[1] in wso2 identity server and their after once you deploy the identity server using following url you can test in SOAP UI

<https://localhost:9443/services/RemoteUserStoreManagerService?wsdl>

1. getAllProfiles

The screenshot shows the SoapUI 5.2.1 interface. The left sidebar contains a project tree with a folder named 'MongoDBUserStoreManager' and a sub-folder 'RemoteUserStoreManagerService'. Under this sub-folder, there is a list of endpoints, including 'getAllProfiles'. The main window displays 'Request 1' for the endpoint 'https://localhost:9443/services/RemoteUserStoreManagerService.RemoteUserStoreManagerServiceHttpsSoap11Endpoint'. The request is a SOAP message with the following XML structure:

```
<?xml version='1.0'?>
<soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>
  <soap:Header/>
  <soap:Body>
    <ser:getAllProfileNames/>
  </soap:Body>
</soap:Envelope>
```

The response is also a SOAP message with the following XML structure:

```
<?xml version='1.0'?>
<soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>
  <soap:Body>
    <ns:getAllProfileNamesResponse xmlns:ns='http://service.ws.um.carbon.wso2.org'>
      <ns:return default='ns:return'>
        <ns:getAllProfileNamesResponse/>
      </ns:return>
    </ns:getAllProfileNamesResponse>
  </soap:Body>
</soap:Envelope>
```

The bottom panel shows the 'Request Properties' table:

Property	Value
Name	Request 1
Description	
Message Size	235
Encoding	UTF-8

The bottom status bar shows the following logs:

```
SoapUI log http log jetty log error log wsrm log memory log
```

2. getHybridRoles

The screenshot shows the SoapUI 5.2.1 interface. The left sidebar contains a project tree with a folder named 'MongoDBUserStoreManager' and a sub-folder 'RemoteUserStoreManagerService'. Under this sub-folder, there is a list of endpoints, including 'getHybridRoles'. The main window displays 'Request 1' for the endpoint 'https://localhost:9443/services/RemoteUserStoreManagerService.RemoteUserStoreManagerServiceHttpsSoap11Endpoint'. The request is a SOAP message with the following XML structure:

```
<?xml version='1.0'?>
<soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>
  <soap:Header/>
  <soap:Body>
    <ser:getHybridRoles/>
  </soap:Body>
</soap:Envelope>
```

The response is also a SOAP message with the following XML structure:

```
<?xml version='1.0'?>
<soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>
  <soap:Body>
    <ns:getHybridRolesResponse xmlns:ns='http://service.ws.um.carbon.wso2.org'>
      <ns:return>Internal/everyone</ns:return>
      <ns:return>Internal/testInternalRole</ns:return>
    </ns:getHybridRolesResponse>
  </soap:Body>
</soap:Envelope>
```

The bottom panel shows the 'Request Properties' table:

Property	Value
Name	Request 1
Description	
Message Size	231
Encoding	UTF-8

The bottom status bar shows the following logs:

```
SoapUI log http log jetty log error log wsrm log memory log
```

3. getPasswordExpirationTime

The screenshot shows the SoapUI 5.2.1 interface. The left sidebar displays a project tree with a folder named 'RemoteUserStoreManagerService'. Under this folder, there is a 'Request 1' for the 'getPasswordExpirationTime' endpoint. The main window shows the raw XML of the request and response. The request is a SOAP envelope with a header and a body containing a 'getPasswordExpirationTime' element. The response is a SOAP envelope with a header and a body containing a 'getPasswordExpirationTimeResponse' element. The response body includes a 'return' element with the value '1'. The bottom panel shows the 'Request Properties' table and a log of the request and response.

Property	Value
Name	Request 1
Description	
Message Size	346
Encoding	UTF-8

Request Properties

Properties

Request 1

Raw XML

```
<?xml version='1.0' encoding='UTF-8'>
<soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>
  <soap:Header/>
  <soap:Body>
    <ser:getPasswordExpirationTime>
      <!-- Optional: -->
      <ser:username?</ser:username>
    </ser:getPasswordExpirationTime>
  </soap:Body>
</soap:Envelope>
```

Response time: 92ms (312 bytes)

Mon Jun 13 13:32:40 EDT 2016:DEBUG:Authorization challenge processed
Mon Jun 13 13:32:40 EDT 2016:DEBUG:Authentication scope: BASIC 'WSO2 Identity Server'@localhost:9443
Mon Jun 13 13:32:40 EDT 2016:INFO:localhost:9443 requires authentication with the realm 'WSO2 Identity Server'
Mon Jun 13 13:32:40 EDT 2016:DEBUG:Found credentials
Mon Jun 13 13:32:40 EDT 2016:DEBUG:Attempt 2 to execute request
Mon Jun 13 13:32:40 EDT 2016:DEBUG:Sending request: POST /services/RemoteUserStoreManagerService.RemoteUserStoreManagerServiceHttpsSoap11Endpoint/...
Mon Jun 13 13:32:40 EDT 2016:DEBUG:Receiving response: HTTP/1.1 200 OK
Mon Jun 13 13:32:40 EDT 2016:DEBUG:Connection can be kept alive indefinitely
Mon Jun 13 13:32:40 EDT 2016:INFO:Got response for [RemoteUserStoreManagerServiceSoap11Binding.getPasswordExpirationTime:Request 1] in 92ms (312 bytes)

SoapUI log http log jetty log error log wsrm log memory log

4. getProfileNames

The screenshot shows the SoapUI 5.2.1 interface. The left sidebar displays a project tree with a folder named 'RemoteUserStoreManagerService'. Under this folder, there is a 'Request 1' for the 'getProfileNames' endpoint. The main window shows the raw XML of the request and response. The request is a SOAP envelope with a header and a body containing a 'getProfileNames' element. The response is a SOAP envelope with a header and a body containing a 'getProfileNamesResponse' element. The response body includes a 'return' element with the value 'default'. The bottom panel shows the 'Request Properties' table and a log of the request and response.

Property	Value
Name	Request 1
Description	
Message Size	326
Encoding	UTF-8

Request Properties

Properties

Request 1

Raw XML

```
<?xml version='1.0' encoding='UTF-8'>
<soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>
  <soap:Header/>
  <soap:Body>
    <ser:getProfileNames>
      <!-- Optional: -->
      <ser:username?</ser:username>
    </ser:getProfileNames>
  </soap:Body>
</soap:Envelope>
```

Response time: 127ms (577 bytes)

Mon Jun 13 13:33:53 EDT 2016:DEBUG:Authorization challenge processed
Mon Jun 13 13:33:53 EDT 2016:DEBUG:Authentication scope: BASIC 'WSO2 Identity Server'@localhost:9443
Mon Jun 13 13:33:53 EDT 2016:INFO:localhost:9443 requires authentication with the realm 'WSO2 Identity Server'
Mon Jun 13 13:33:53 EDT 2016:DEBUG:Found credentials
Mon Jun 13 13:33:53 EDT 2016:DEBUG:Attempt 2 to execute request
Mon Jun 13 13:33:53 EDT 2016:DEBUG:Sending request: POST /services/RemoteUserStoreManagerService.RemoteUserStoreManagerServiceHttpsSoap11Endpoint/...
Mon Jun 13 13:33:53 EDT 2016:DEBUG:Receiving response: HTTP/1.1 200 OK
Mon Jun 13 13:33:53 EDT 2016:DEBUG:Connection can be kept alive indefinitely
Mon Jun 13 13:33:53 EDT 2016:INFO:Got response for [RemoteUserStoreManagerServiceSoap11Binding.getProfileNames:Request 1] in 127ms (577 bytes)

SoapUI log http log jetty log error log wsrm log memory log

5. getProperties

SoapUI 5.2.1

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy

Search Forum Online Help

Navigator

- deleteUserClaimValues
- getAllProfileNames
- Request 1
- getHybridRoles
- Request 1
- getPasswordExpirationTime
- Request 1
- getProfileNames
- Request 1
- getProperties
- Request 1
- getRoleListofUser
- getRoleNames
- getTenantId
- getTenantIdofUser
- getUserClaimValue
- getUserClaimValuesForC
- getUserid
- getUserList
- getUserListofRole
- isExistingRole
- isExistingUser
- isReadOnly
- listUsers
- setUserClaimValue
- setUserClaimValues
- updateCredential

Request Properties

Property	Value
Name	Request 1
Description	
Message Size	533
Encoding	UTF-8

Request 1

Raw XML

```
<?xml version='1.0' encoding='UTF-8'>
<soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>
  <soap:Header/>
  <soap:Body>
    <ser:getProperties>
      <i--Optional:-->
        <ser:tenant>
          <i--Optional:-->
            <xsd:active></xsd:active>
          <i--Optional:-->
            <xsd:realmConfig>
              </xsd:realmConfig>
            </ser:tenant>
          </ser:getProperties>
        </soap:Body>
      </soap:Envelope>
    </soap:Body>
  </soap:Envelope>
</?xml>
```

Authorization: Basic

Username: admin

Password: ****

response time: 114ms (517 bytes)

Mon Jun 13 13:36:16 EDT 2016:DEBUG:Authentication scope: BASIC 'WSO2 Identity Server'@localhost:9443

Mon Jun 13 13:36:16 EDT 2016:INFO:localhost:9443 requires authentication with the realm 'WSO2 Identity Server'

Mon Jun 13 13:36:16 EDT 2016:DEBUG:Found credentials

Mon Jun 13 13:36:16 EDT 2016:DEBUG:Attempt 2 to execute request

Mon Jun 13 13:36:16 EDT 2016:DEBUG:Sending request: POST /services/RemoteUserStoreManagerService.RemoteUserStoreManagerServiceHttpsSoap11Endpoint/

Mon Jun 13 13:36:16 EDT 2016:DEBUG:Receiving response: HTTP/1.1 200 OK

Mon Jun 13 13:36:16 EDT 2016:DEBUG:Connection can be kept alive indefinitely

Mon Jun 13 13:36:16 EDT 2016:INFO:Got response for [RemoteUserStoreManagerServiceSoap11Binding.getProperties:Request 1] in 114ms (517 bytes)

Mon Jun 13 13:36:50 EDT 2016:DEBUG:Connection closed

SoapUI log http log jetty log error log wsrm log memory log

6. getRoleNames

SoapUI 5.2.1

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy

Search Forum Online Help

Navigator

- deleteUserClaimValues
- getAllProfileNames
- Request 1
- getHybridRoles
- Request 1
- getPasswordExpirationTime
- Request 1
- getProfileNames
- Request 1
- getProperties
- Request 1
- getRoleListofUser
- getRoleNames
- Request 1
- getTenantId
- getTenantIdofUser
- getUserClaimValue
- getUserClaimValuesForC
- getUserid
- getUserList
- getUserListofRole
- isExistingRole
- isExistingUser
- isReadOnly
- listUsers
- setUserClaimValue

Request Properties

Property	Value
Name	Request 1
Description	
Message Size	229
Encoding	UTF-8

Request 1

Raw XML

```
<?xml version='1.0' encoding='UTF-8'>
<soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>
  <soap:Header/>
  <soap:Body>
    <ser:getRoleNames>
      </ser:getRoleNames>
    </soap:Body>
  </soap:Envelope>
</?xml>
```

Authorization: Basic

Username: admin

response time: 951ms (897 bytes)

Mon Jun 13 13:38:57 EDT 2016:DEBUG:Authorization challenge processed

Mon Jun 13 13:38:57 EDT 2016:DEBUG:Authentication scope: BASIC 'WSO2 Identity Server'@localhost:9443

Mon Jun 13 13:38:57 EDT 2016:INFO:localhost:9443 requires authentication with the realm 'WSO2 Identity Server'

Mon Jun 13 13:38:57 EDT 2016:DEBUG:Found credentials

Mon Jun 13 13:38:57 EDT 2016:DEBUG:Attempt 2 to execute request

Mon Jun 13 13:38:57 EDT 2016:DEBUG:Sending request: POST /services/RemoteUserStoreManagerService.RemoteUserStoreManagerServiceHttpsSoap11Endpoint/

Mon Jun 13 13:38:58 EDT 2016:DEBUG:Receiving response: HTTP/1.1 200 OK

Mon Jun 13 13:38:58 EDT 2016:DEBUG:Connection can be kept alive indefinitely

Mon Jun 13 13:38:58 EDT 2016:INFO:Got response for [RemoteUserStoreManagerServiceSoap11Binding.getRoleNames:Request 1] in 951ms (897 bytes)

SoapUI log http log jetty log error log wsrm log memory log

7. getTenantId

The screenshot shows the SoapUI 5.2.1 interface. The left sidebar contains a tree view of test cases and requests. The main window displays a SOAP request and response for the `getTenantId` endpoint. The request is a POST to `https://localhost:9443/services/RemoteUserStoreManagerService.RemoteUserStoreManagerServiceHttpsSoap11Endpoint/` with the following XML body:

```
<?xml version='1.0' encoding='UTF-8'>
<soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>
  <soap:Header/>
  <soap:Body>
    <ser:getTenantId/>
  </soap:Body>
</soap:Envelope>
```

The response is a SOAP envelope with the following XML body:

```
<?xml version='1.0' encoding='UTF-8'>
<soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>
  <soap:Body>
    <ns:getTenantIdResponse xmlns:ns='http://service.ws.um.carbon.wso2.org/'>
      <ns:return>-1234</ns:return>
    </ns:getTenantIdResponse>
  </soap:Body>
</soap:Envelope>
```

The response time is 248ms (287 bytes). The log shows the following messages:

```
Mon Jun 13 13:42:03 EDT 2016:DEBUG:Authorization challenge processed
Mon Jun 13 13:42:03 EDT 2016:DEBUG:Authentication scope: BASIC 'WSO2 Identity Server'@localhost:9443
Mon Jun 13 13:42:03 EDT 2016:INFO:localhost:9443 requires authentication with the realm 'WSO2 Identity Server'
Mon Jun 13 13:42:03 EDT 2016:DEBUG:Found credentials
Mon Jun 13 13:42:03 EDT 2016:DEBUG:Attempt 2 to execute request
Mon Jun 13 13:42:03 EDT 2016:DEBUG:Sending request: POST /services/RemoteUserStoreManagerService.RemoteUserStoreManagerServiceHttpsSoap11Endpoint/
Mon Jun 13 13:42:03 EDT 2016:DEBUG:Receiving response: HTTP/1.1 200 OK
Mon Jun 13 13:42:03 EDT 2016:DEBUG:Connection can be kept alive indefinitely
Mon Jun 13 13:42:03 EDT 2016:INFO:Got response for [RemoteUserStoreManagerServiceSoap11Binding.getTenantId:Request 1] in 248ms (287 bytes)
```

8. getRolesListOfUser

The screenshot shows the SoapUI 5.2.1 interface. The left sidebar contains a tree view of test cases and requests. The main window displays a SOAP request and response for the `getRolesListOfUser` endpoint. The request is a POST to `https://localhost:9443/services/RemoteUserStoreManagerService.RemoteUserStoreManagerServiceHttpsSoap11Endpoint/` with the following XML body:

```
<?xml version='1.0' encoding='UTF-8'>
<soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>
  <soap:Header/>
  <soap:Body>
    <ser:getRolesListOfUser/>
  </soap:Body>
</soap:Envelope>
```

The response is a SOAP envelope with the following XML body:

```
<?xml version='1.0' encoding='UTF-8'>
<soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>
  <soap:Body>
    <ns:getRolesListOfUserResponse xmlns:ns='http://service.ws.um.carbon.wso2.org/'>
      <ns:return>MONGOSTORE/staff</ns:return>
      <ns:return>Internal/everyone</ns:return>
      <ns:return>MONGOSTORE/admin</ns:return>
    </ns:getRolesListOfUserResponse>
  </soap:Body>
</soap:Envelope>
```

The response time is 378ms (669 bytes). The log shows the following messages:

```
Mon Jun 13 13:43:33 EDT 2016:DEBUG:Authorization challenge processed
Mon Jun 13 13:43:33 EDT 2016:DEBUG:Authentication scope: BASIC 'WSO2 Identity Server'@localhost:9443
Mon Jun 13 13:43:33 EDT 2016:INFO:localhost:9443 requires authentication with the realm 'WSO2 Identity Server'
Mon Jun 13 13:43:33 EDT 2016:DEBUG:Found credentials
Mon Jun 13 13:43:33 EDT 2016:DEBUG:Attempt 2 to execute request
Mon Jun 13 13:43:33 EDT 2016:DEBUG:Sending request: POST /services/RemoteUserStoreManagerService.RemoteUserStoreManagerServiceHttpsSoap11Endpoint/
Mon Jun 13 13:43:34 EDT 2016:DEBUG:Receiving response: HTTP/1.1 200 OK
Mon Jun 13 13:43:34 EDT 2016:DEBUG:Connection can be kept alive indefinitely
Mon Jun 13 13:43:34 EDT 2016:INFO:Got response for [RemoteUserStoreManagerServiceSoap11Binding.getRolesListOfUser:Request 1] in 378ms (669 bytes)
```


9. getUserClaimValue

The screenshot shows the SoapUI 5.2.1 interface. The left sidebar contains a project tree with various test cases and requests. The main window displays a SOAP request for the `getUserClaimValue` endpoint. The request is configured with the following details:

- URL:** `https://localhost:9443/services/RemoteUserStoreManagerService.RemoteUserStoreManagerServiceHttpsSoap11Endpoint/`
- Authorization:** Basic
- Username:** admin
- Password:** ****

The request body is a SOAP envelope with the following XML structure:

```
<?xml version='1.0' encoding='UTF-8'>
<soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>
  <soap:Header/>
  <soap:Body>
    <ser:getUserClaimValue>
      <i-- Optional:-->
      <ser:userName>MONGOSTORE/shakti</i-- Optional:-->
      <ser:claim>http://wso2.org/claims/username</i-- Optional:-->
      <ser:profileName>MONGOSTORE/default</i-- Optional:-->
    </ser:getUserClaimValue>
  </soap:Body>
</soap:Envelope>
```

The response time is 122ms (301 bytes). The log shows the following sequence of events:

- Mon Jun 13 13:52:41 EDT 2016:DEBUG:Authorization challenge processed
- Mon Jun 13 13:52:41 EDT 2016:DEBUG:Authentication scope: BASIC 'WSO2 Identity Server'@localhost:9443
- Mon Jun 13 13:52:41 EDT 2016:INFO:localhost:9443 requires authentication with the realm 'WSO2 Identity Server'
- Mon Jun 13 13:52:41 EDT 2016:DEBUG:Found credentials
- Mon Jun 13 13:52:41 EDT 2016:DEBUG:Attempt 2 to execute request
- Mon Jun 13 13:52:41 EDT 2016:DEBUG:Sending request: POST /services/RemoteUserStoreManagerService.RemoteUserStoreManagerServiceHttpsSoap11Endpoint/
- Mon Jun 13 13:52:41 EDT 2016:DEBUG:Receiving response: HTTP/1.1 200 OK
- Mon Jun 13 13:52:41 EDT 2016:DEBUG:Connection can be kept alive indefinitely
- Mon Jun 13 13:52:41 EDT 2016:INFO:Got response for [RemoteUserStoreManagerServiceSoap11Binding.getUserClaimValue:Request 1] in 122ms (301 bytes)

10. getUserClaimValues

The screenshot shows the SoapUI 5.2.1 interface. The left sidebar contains a project tree with various test cases and requests. The main window displays a SOAP request for the `getUserClaimValues` endpoint. The request is configured with the following details:

- URL:** `https://localhost:9443/services/RemoteUserStoreManagerService.RemoteUserStoreManagerServiceHttpsSoap11Endpoint/`
- Authorization:** Basic
- Username:** admin
- Password:** ****

The request body is a SOAP envelope with the following XML structure:

```
<?xml version='1.0' encoding='UTF-8'>
<soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>
  <soap:Header/>
  <soap:Body>
    <ser:getUserClaimValues>
      <i-- Optional:-->
      <ser:userName>MONGOSTORE/shakti</i-- Optional:-->
    </ser:getUserClaimValues>
  </soap:Body>
</soap:Envelope>
```

The response time is 331ms (9460 bytes). The log shows the following sequence of events:

- Mon Jun 13 13:53:39 EDT 2016:DEBUG:Authorization challenge processed
- Mon Jun 13 13:53:39 EDT 2016:DEBUG:Authentication scope: BASIC 'WSO2 Identity Server'@localhost:9443
- Mon Jun 13 13:53:39 EDT 2016:INFO:localhost:9443 requires authentication with the realm 'WSO2 Identity Server'
- Mon Jun 13 13:53:39 EDT 2016:DEBUG:Found credentials
- Mon Jun 13 13:53:39 EDT 2016:DEBUG:Attempt 2 to execute request
- Mon Jun 13 13:53:39 EDT 2016:DEBUG:Sending request: POST /services/RemoteUserStoreManagerService.RemoteUserStoreManagerServiceHttpsSoap11Endpoint/
- Mon Jun 13 13:53:39 EDT 2016:DEBUG:Receiving response: HTTP/1.1 200 OK
- Mon Jun 13 13:53:39 EDT 2016:DEBUG:Connection can be kept alive indefinitely
- Mon Jun 13 13:53:39 EDT 2016:INFO:Got response for [RemoteUserStoreManagerServiceSoap11Binding.getUserClaimValues:Request 1] in 331ms (9460 bytes)

11. getClaimValuesForClaim

The screenshot shows the SoapUI 5.2.1 interface. The left sidebar contains a tree view of requests, with 'Request 1' selected under the 'getClaimValuesForClaim' endpoint. The main window displays the raw XML of the request and response. The request is a SOAP envelope with a header and a body containing a 'ser:getUserClaimValuesForClaims' element. The response is a SOAP envelope with a header and a body containing a 'ns:getUserClaimValuesForClaimsResponse' element. The response body includes a 'return' element with a 'value' attribute set to 'Shakthi'.

Request 1

```
<?xml version='1.0' encoding='UTF-8'>
<soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>
  <soap:Header/>
  <soap:Body>
    <ser:getUserClaimValuesForClaims>
      <!--Optional-->
      <!--Zero or more repetitions-->
      <ser:claims=http://ws2.org/claims/>
      <!--Optional-->
      <ser:profileName=MONGOSTORE/default>
    </ser:getUserClaimValuesForClaims>
  </soap:Body>
</soap:Envelope>
```

Response 1

```
<?xml version='1.0' encoding='UTF-8'>
<soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>
  <soap:Header/>
  <soap:Body>
    <ns:getUserClaimValuesForClaimsResponse xmlns:ns='http://service.ws.um.carbon.com/'>
      <return xsi:type='ax2660:ClaimValue' xmlns:xsi='http://www.w3.org/2001/XMLSchema'>
        <ax2660:claimURI=http://ws2.org/claims/givenname/>
        <ax2660:value>Shakthi</ax2660:value>
      </return>
    </ns:getUserClaimValuesForClaimsResponse>
  </soap:Body>
</soap:Envelope>
```

Request Properties

Property	Value
Name	Request 1
Description	
Message Size	565
Encoding	UTF-8

Log

```
Mon Jun 13 13:55:57 EDT 2016:DEBUG:Authentication scope: BASIC 'WSO2 Identity Server'@localhost:9443
Mon Jun 13 13:55:57 EDT 2016:INFO:localhost:9443 requires authentication with the realm 'WSO2 Identity Server'
Mon Jun 13 13:55:57 EDT 2016:DEBUG:Found credentials
Mon Jun 13 13:55:57 EDT 2016:DEBUG:Attempt 2 to execute request
Mon Jun 13 13:55:57 EDT 2016:DEBUG:Sending request: POST /services/RemoteUserStoreManagerService.RemoteUserStoreManagerServiceHttpsSoap11Endpoint/
Mon Jun 13 13:55:57 EDT 2016:DEBUG:Receiving response: HTTP/1.1 200 OK
Mon Jun 13 13:55:57 EDT 2016:DEBUG:Connection can be kept alive indefinitely
Mon Jun 13 13:55:57 EDT 2016:INFO:Got response for [RemoteUserStoreManagerServiceSoap11Binding.getUserClaimValuesForClaims:Request 1] in 105ms (780 bytes)
Mon Jun 13 13:56:30 EDT 2016:DEBUG:Connection closed
```

12. getUserListOfRole

The screenshot shows the SoapUI 5.2.1 interface. The left sidebar contains a tree view of requests, with 'Request 1' selected under the 'getUserListOfRole' endpoint. The main window displays the raw XML of the request and response. The request is a SOAP envelope with a header and a body containing a 'ser:getUserListOfRole' element. The response is a SOAP envelope with a header and a body containing a 'ns:getUserListOfRoleResponse' element. The response body includes a 'return' element with a 'value' attribute set to 'Shakthi'.

Request 1

```
<?xml version='1.0' encoding='UTF-8'>
<soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>
  <soap:Header/>
  <soap:Body>
    <ser:getUserListOfRole>
      <!--Optional-->
      <ser:roleName=MONGOSTORE/staff/>
      <ser:getUserListOfRole/>
    </ser:getUserListOfRole>
  </soap:Body>
</soap:Envelope>
```

Response 1

```
<?xml version='1.0' encoding='UTF-8'>
<soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>
  <soap:Header/>
  <soap:Body>
    <ns:getUserListOfRoleResponse xmlns:ns='http://service.ws.um.carbon.com/'>
      <return>
        <ns:return>MONGOSTORE/asantha</ns:return>
        <ns:return>MONGOSTORE/shakthi</ns:return>
        <ns:return>MONGOSTORE/iran</ns:return>
        <ns:return>MONGOSTORE/kamal</ns:return>
        <ns:return>MONGOSTORE/tarzan</ns:return>
        <ns:return>MONGOSTORE/saman</ns:return>
        <ns:return>MONGOSTORE/chalitha</ns:return>
        <ns:return>MONGOSTORE/jayan</ns:return>
      </return>
    </ns:getUserListOfRoleResponse>
  </soap:Body>
</soap:Envelope>
```

Request Properties

Property	Value
Name	Request 1
Description	
Message Size	345
Encoding	UTF-8

Log

```
Mon Jun 13 14:05:00 EDT 2016:DEBUG:Authorization challenge processed
Mon Jun 13 14:05:00 EDT 2016:DEBUG:Authentication scope: BASIC 'WSO2 Identity Server'@localhost:9443
Mon Jun 13 14:05:00 EDT 2016:INFO:localhost:9443 requires authentication with the realm 'WSO2 Identity Server'
Mon Jun 13 14:05:00 EDT 2016:DEBUG:Found credentials
Mon Jun 13 14:05:00 EDT 2016:DEBUG:Attempt 2 to execute request
Mon Jun 13 14:05:00 EDT 2016:DEBUG:Sending request: POST /services/RemoteUserStoreManagerService.RemoteUserStoreManagerServiceHttpsSoap11Endpoint/
Mon Jun 13 14:05:00 EDT 2016:DEBUG:Receiving response: HTTP/1.1 200 OK
Mon Jun 13 14:05:00 EDT 2016:DEBUG:Connection can be kept alive indefinitely
Mon Jun 13 14:05:00 EDT 2016:INFO:Got response for [RemoteUserStoreManagerServiceSoap11Binding.getUserListOfRole:Request 1] in 106ms (871 bytes)
```


13. isExistingRole

The screenshot displays the SoapUI 5.2.1 interface. The left sidebar shows a project tree with a request named 'isExistingRole' selected. The main window shows the 'Request 1' tab with the following XML payload:

```
<?xml version='1.0' encoding='UTF-8'>
<soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>
  <soap:Header/>
  <soap:Body>
    <ser:isExistingRole>
      <!--Optional:-->
      <ser:roleName>MONGOSTORE/staff</ser:roleName>
    </ser:isExistingRole>
  </soap:Body>
</soap:Envelope>
```

The 'Authorization' tab shows 'Basic' authentication with 'Username: admin' and 'Password: ****'. The 'Response' tab shows the following XML payload:

```
<?xml version='1.0' encoding='UTF-8'>
<soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>
  <soap:Header/>
  <soap:Body>
    <ns:isExistingRoleResponse xmlns:ns='http://service.ws.um.carbon.com/'>
      <ns:returntrue/>
    </ns:isExistingRoleResponse>
  </soap:Body>
</soap:Envelope>
```

The 'Log' tab shows the following log entries:

```
Mon Jun 13 14:09:43 EDT 2016:DEBUG:Authorization challenge processed
Mon Jun 13 14:09:43 EDT 2016:DEBUG:Authentication scope: BASIC 'WSO2 Identity Server'@localhost:9443
Mon Jun 13 14:09:43 EDT 2016:INFO:localhost:9443 requires authentication with the realm 'WSO2 Identity Server'
Mon Jun 13 14:09:43 EDT 2016:DEBUG:Found credentials
Mon Jun 13 14:09:43 EDT 2016:DEBUG:Attempt 2 to execute request
Mon Jun 13 14:09:43 EDT 2016:DEBUG:Sending request: POST /services/RemoteUserStoreManagerService.RemoteUserStoreManagerServiceHttpsSoap11Endpoint/
Mon Jun 13 14:09:43 EDT 2016:DEBUG:Receiving response: HTTP/1.1 200 OK
Mon Jun 13 14:09:43 EDT 2016:DEBUG:Connection can be kept alive indefinitely
Mon Jun 13 14:09:43 EDT 2016:INFO:Got response for [RemoteUserStoreManagerServiceSoap11Binding.isExistingRole:Request 1] in 93ms (292 bytes)
```

14. isExistingUser

The screenshot displays the SoapUI 5.2.1 interface. The left sidebar shows a project tree with a request named 'isExistingUser' selected. The main window shows the 'Request 1' tab with the following XML payload:

```
<?xml version='1.0' encoding='UTF-8'>
<soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>
  <soap:Header/>
  <soap:Body>
    <ser:isExistingUser>
      <!--Optional:-->
      <ser:userName>MONGOSTORE/asantha</ser:userName>
    </ser:isExistingUser>
  </soap:Body>
</soap:Envelope>
```

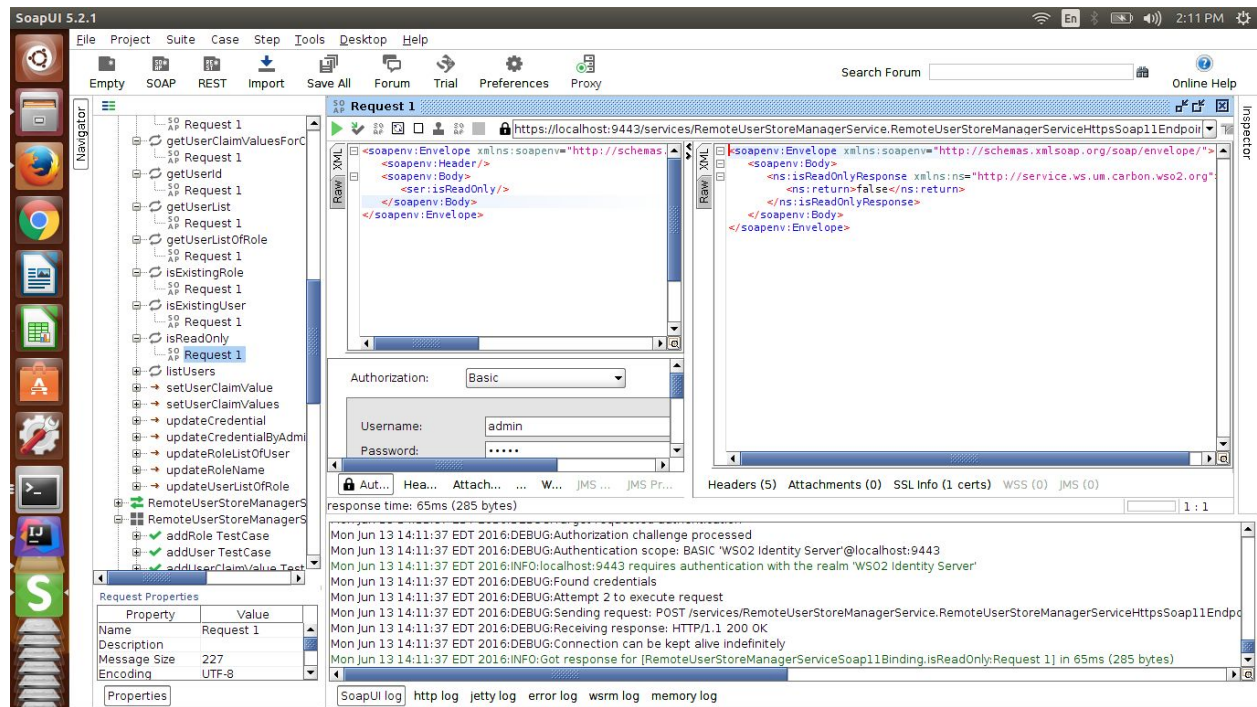
The 'Authorization' tab shows 'Basic' authentication with 'Username: admin' and 'Password: ****'. The 'Response' tab shows the following XML payload:

```
<?xml version='1.0' encoding='UTF-8'>
<soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>
  <soap:Header/>
  <soap:Body>
    <ns:isExistingUserResponse xmlns:ns='http://service.ws.um.carbon.com/'>
      <ns:returntrue/>
    </ns:isExistingUserResponse>
  </soap:Body>
</soap:Envelope>
```

The 'Log' tab shows the following log entries:

```
Mon Jun 13 14:10:35 EDT 2016:DEBUG:Authorization challenge processed
Mon Jun 13 14:10:35 EDT 2016:DEBUG:Authentication scope: BASIC 'WSO2 Identity Server'@localhost:9443
Mon Jun 13 14:10:35 EDT 2016:INFO:localhost:9443 requires authentication with the realm 'WSO2 Identity Server'
Mon Jun 13 14:10:35 EDT 2016:DEBUG:Found credentials
Mon Jun 13 14:10:35 EDT 2016:DEBUG:Attempt 2 to execute request
Mon Jun 13 14:10:35 EDT 2016:DEBUG:Sending request: POST /services/RemoteUserStoreManagerService.RemoteUserStoreManagerServiceHttpsSoap11Endpoint/
Mon Jun 13 14:10:35 EDT 2016:DEBUG:Receiving response: HTTP/1.1 200 OK
Mon Jun 13 14:10:35 EDT 2016:DEBUG:Connection can be kept alive indefinitely
Mon Jun 13 14:10:35 EDT 2016:INFO:Got response for [RemoteUserStoreManagerServiceSoap11Binding.isExistingUser:Request 1] in 65ms (292 bytes)
```

15. isReadOnly



References

- [1] <https://docs.wso2.com/display/IS500/Calling+Admin+Services>
- [2] http://googledrive.com/host/0B_ZE1ru1jkXbDVoSDIveV9yNk0