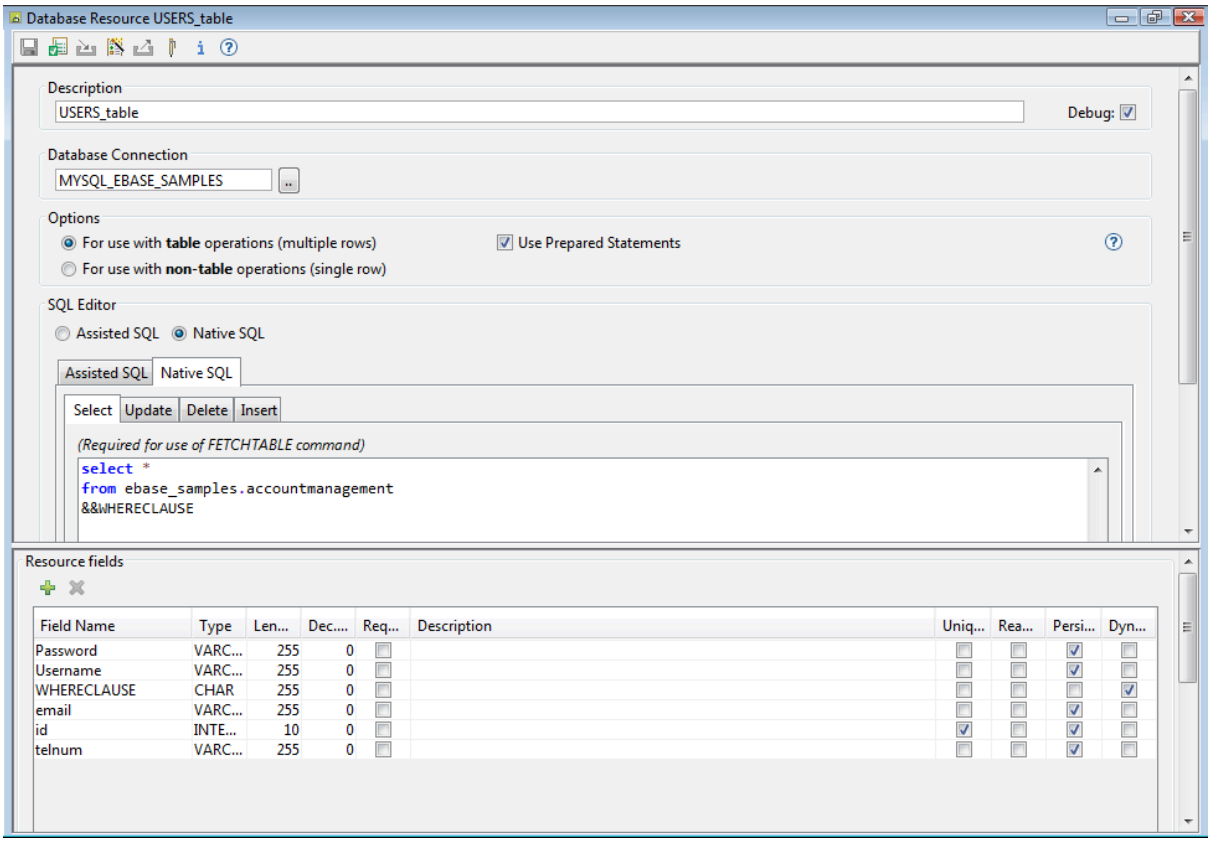
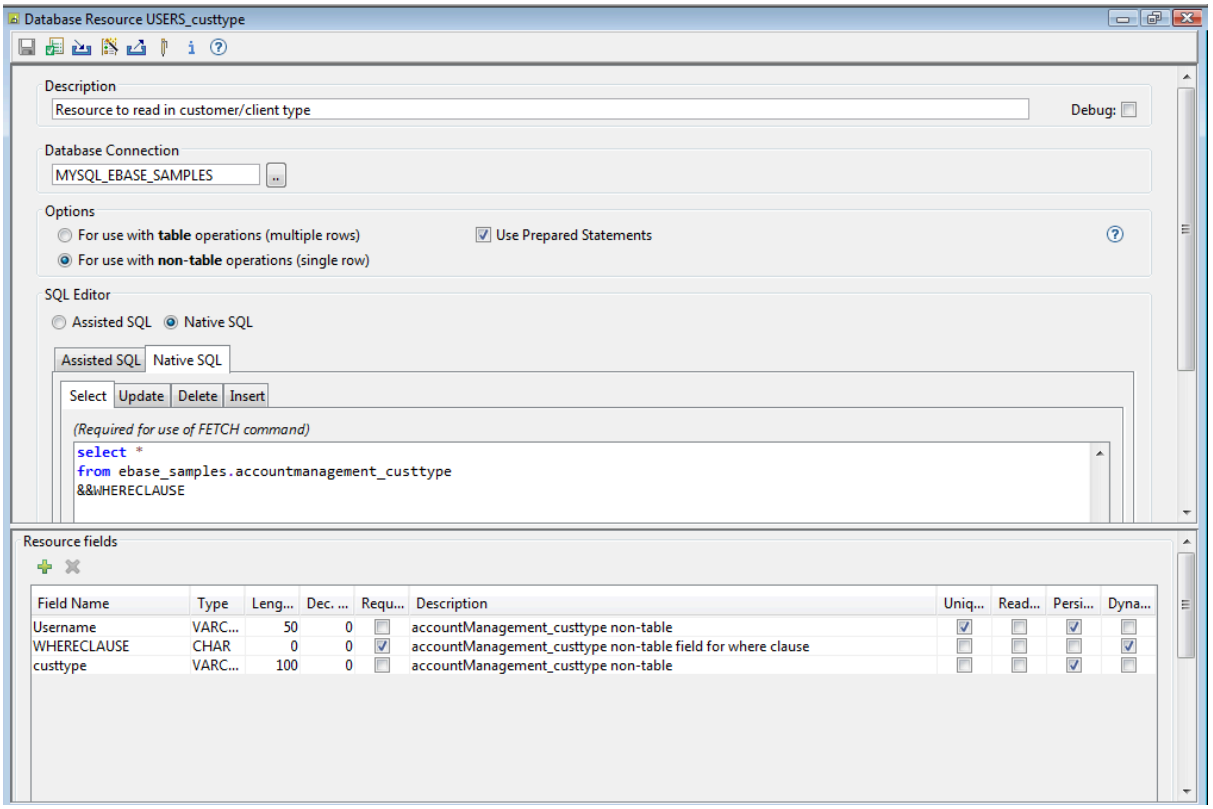
1). MySQL database setup.
Define table(s) to handle security

```
CREATE TABLE `accountmanagement` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `Username` varchar(50) DEFAULT NULL,
  `Password` varchar(50) CHARACTER SET latin1 DEFAULT NULL,
  `email` varchar(100) DEFAULT NULL,
  `telnum` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `accountmanagement_custtype` (
  `Username` varchar(50) NOT NULL,
  `custtype` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`Username`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

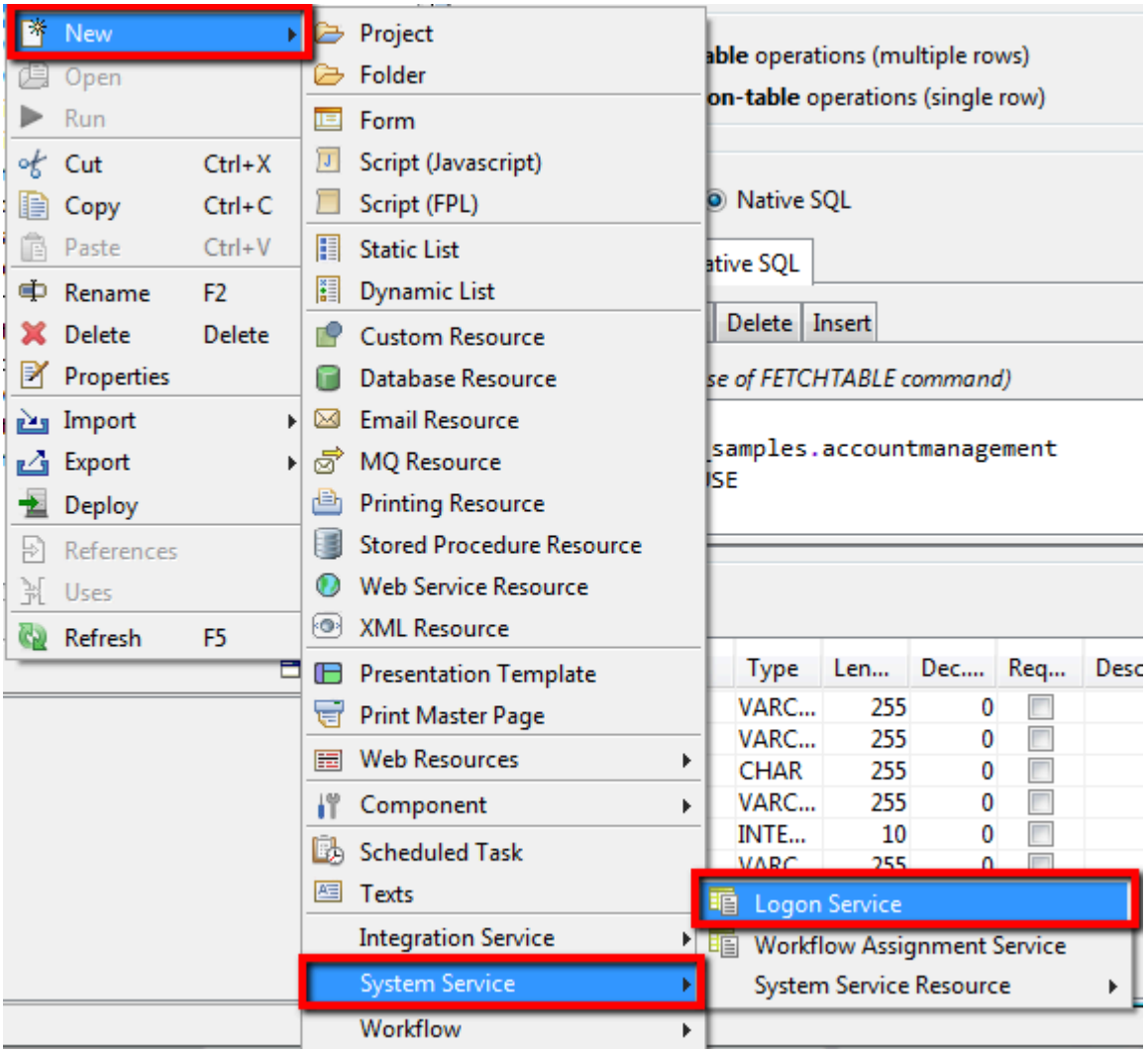
2). Define database resource based on database connection to your database repository to the table(s) defined in step 1.





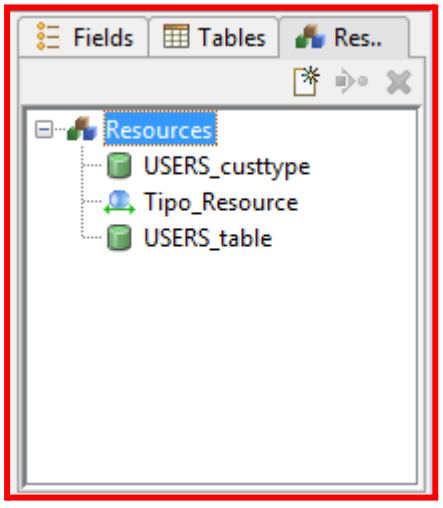
Please note the “WHERECLAUSE” field added in the tables above. This info is per comments by segi which is crucial.

3). Define a login service for your project.
Please check the version of ebase you using, most will have a project that has this already implemented (ver 5.0.0: ebaseSystemServices) that I believe becomes the default login service that can be used by projects deployed within that deployed webapp.
In the following section, we creating one ourselves just as a reference.

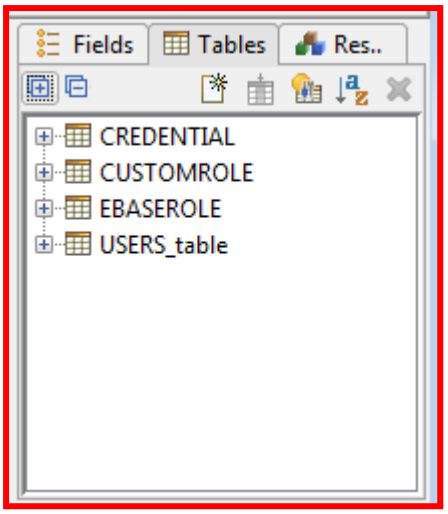


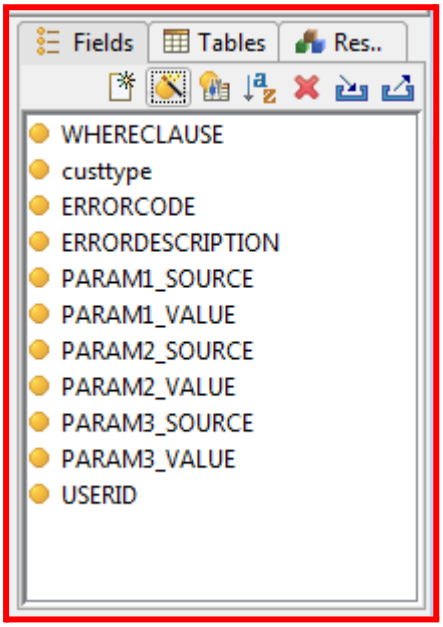
This operation creates the logon service as well as the logon service resource for it.

4). Add the database resource created in step 2 to the logon service resources.



5). Add the corresponding table and fields of the database resource to the tables and Field tab.



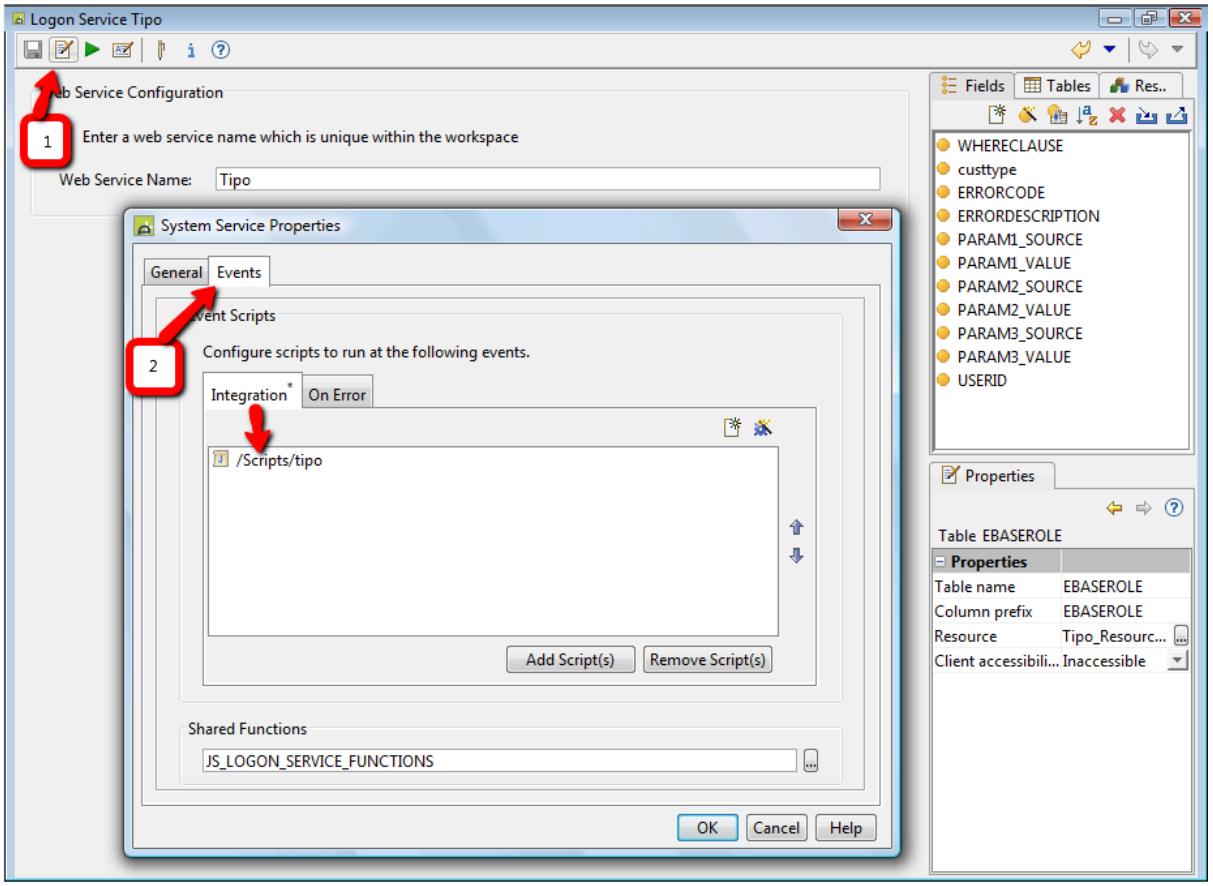


6). Script.
In this example, I copied the code from the supplied example ebaseSystemServices and amended as follows

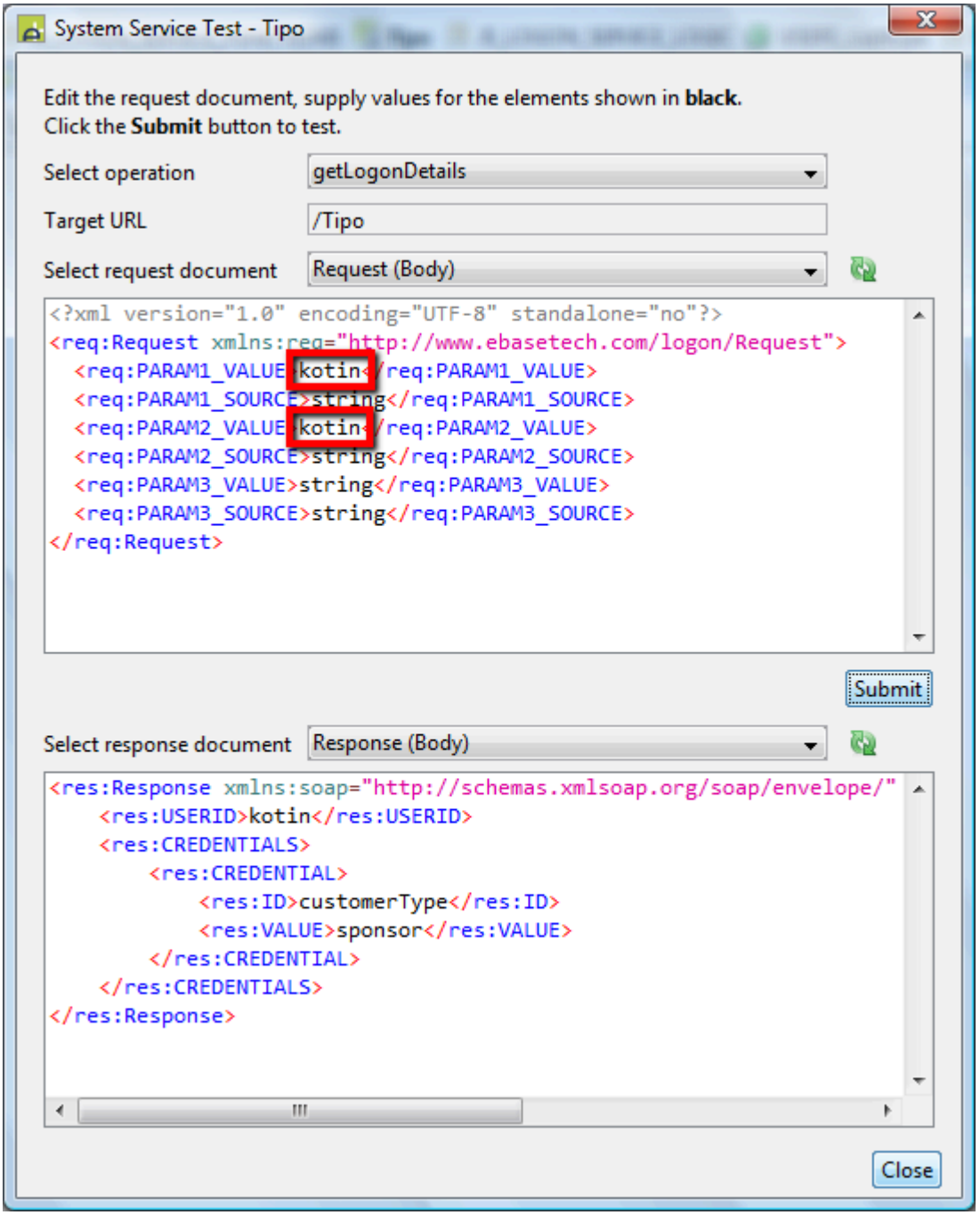
```
importPackage(com.ebasetech.xi.api);
importPackage(com.ebasetech.xi.services);


try {

        /**
        * Required by the login process to be set!
        */
 fields.USERID.value=fields.PARAM1_VALUE.value;

        /**
```

```
		* Using Native SQL with a database resource field of WHERECLAUSE defined.
		* Note the seetings required for this since is crucial!
		*/
  var n = "WHERE BINARY Username=\"" + fields.PARAM1_VALUE.value + "\" AND Password=aes_encrypt('"+
fields.PARAM2_VALUE.value +"','definesecuritystrip')";

 tables.USERS_table.WHERECLAUSE.value=n;

		tables.USERS_table.fetchTable();
		log("COUNT (tables.USERS_table.fetchTable()): "+ tables.USERS_table.rowCount);

		if ( tables.USERS_table.rowCount > 0 ) {
			/**
			 * successful login
			 */

			// Required for Ebase SecurityManager to know that the authentication was successfull
   fields.USERID.value=tables.USERS_table.Username.value;

   //Gather customer/client type data
   var n = "WHERE BINARY Username=\"" + fields.USERID.value +"\"";
		fields.WHERECLAUSE.value = n;

		var loginResult = resources.USERS_custtype.fetch();
			log("AK: status of reading in the customer/client type: "+ loginResult + ". Customer/Client type: "+
fields.custtype.value);

   // Comment this out to add credentials here as needed which can be accessed by any application to get
information about the currently logged in user by doing system.securityManager.getCredential("REALNAME");
   tables.CREDENTIALS.insertRow();
   tables.CREDENTIALS.ID.value="customerType";
   tables.CREDENTIALS.VALUE.value=fields.custtype.value;
   tables.CREDENTIALS.updateTable();

		} else {
			/**
			 * login failure
			 */
			fields.USERID.value = null;
   fields.ERRORCODE.value = "999991";
   fields.ERRORDESCRIPTION.value = "Your username or password is not correct";
   print("Login failed with the username " + fields.PARAM1_VALUE.value);
		}

} catch (e) {
  fields.USERID.value = null;
  fields.ERRORCODE.value = "999991";
  fields.ERRORDESCRIPTION.value = "Your username or password is not correct" + e;
}
```

Add the script to the logon service.



You can then test the logon service by typing in the values of userid and password. Please see output which shows the credentials as the output.


System Service Test - Tipo
Edit the request document, supply values for the elements shown in **black**.
Click the **Submit** button to test.
Select operation
getLogonDetails
Target URL
/Tipo
Select request document
Request (Body)
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<req:Request xmlns:req="http://www.ebasetech.com/logon/Request">
<req:PARAM1_VALUE>kotin</req:PARAM1_VALUE>
<req:PARAM1_SOURCE>string</req:PARAM1_SOURCE>
<req:PARAM2_VALUE>kotin</req:PARAM2_VALUE>
<req:PARAM2_SOURCE>string</req:PARAM2_SOURCE>
<req:PARAM3_VALUE>string</req:PARAM3_VALUE>
<req:PARAM3_SOURCE>string</req:PARAM3_SOURCE>
</req:Request>
Submit
Select response document
Response (Body)
<res:Response xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
<res:USERID>kotin</res:USERID>
<res:CREDENTIALS>
<res:CREDENTIAL>
<res:ID>customerType</res:ID>
<res:VALUE>sponsor</res:VALUE>
</res:CREDENTIAL>
</res:CREDENTIALS>
</res:Response>
Close