Mock-Ups

Mock-ups are used to cut the dependency to services by emulating their response. This is useful for regression testing or development if the services are yet not there or if their behavior is stateful.

The mock-up example used in this section consists of a simple input form to create a support case followed by a confirmation dialog. Before the dialog is opened the **loadData** service is called to fetch additional information. This behavior is modelled with the state chart below:
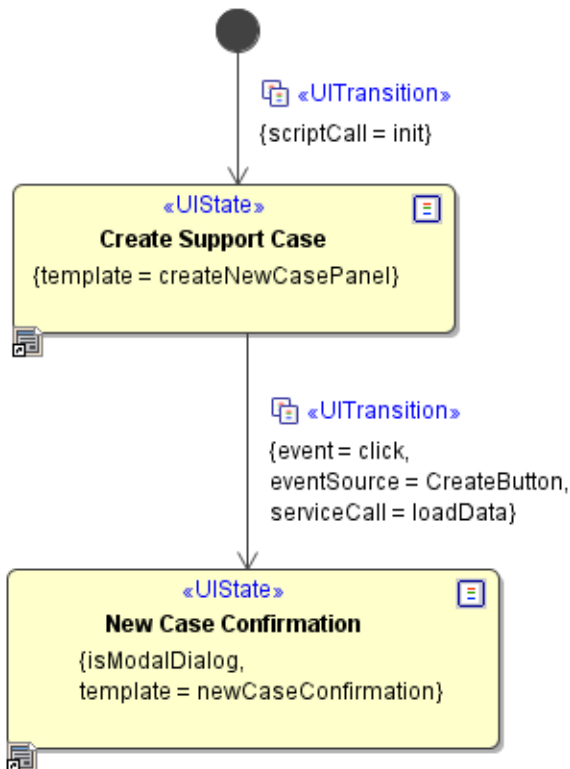
Executing the mock-up example in Firebug shows the response of the **loadData** service:
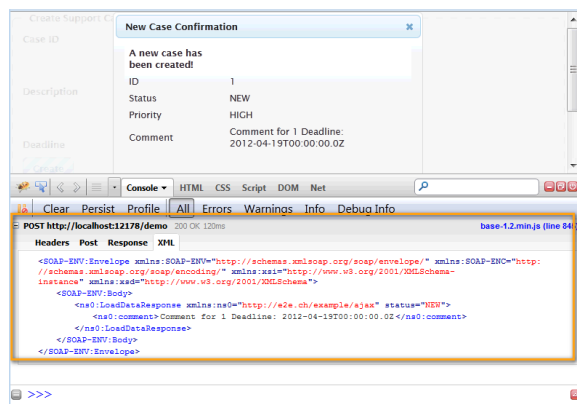

If we want to avoid calling this service we can use the controller operation **e2eSetMockup(key, value)**. This operation requires the following parameters:

- Mock-up key: this string is derived from the service port and operation name: key = portTypeName + '.' + operationName
- Mock-up value: This is the service response. The Firebug window above shows an example.

In the uiMockup.xml example we activate the mock-up by using the URL: ?mockup=true" class="external-link" rel="nofollow">http://localhost:12496/ui/UIMockup.html**?mockup=true**.

This URL sets the public **mockup** controller attribute to true (see containment tree below). As the state machine above indicates, the JavaScript operation **init** is called before reaching the first UI state. In this operation we call the **e2eSetMockup** operation to set the service response value. The mock-up key is "TestPort.loadData" because the SOAP operation **loadData** is member of the **TestPort** class. The following figure shows the **init()** operation:
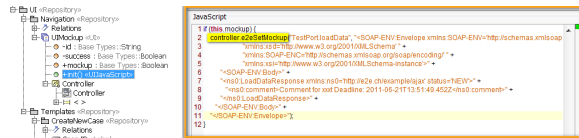
When running the example with **mockup=true**, the Firebug console now shows that the service is not called anymore but instead the mock-up value has been returned:
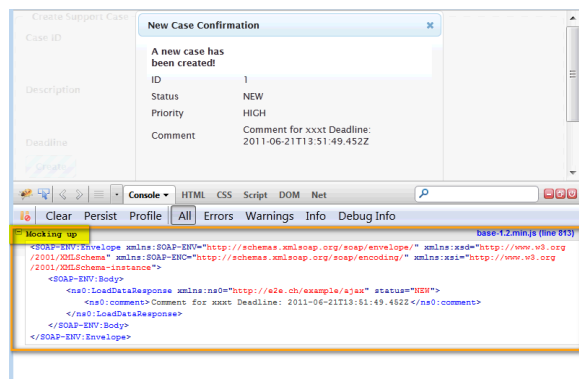
Using the controller **mockup** attribute is just an example how to switch between mock-up and non-mock-up mode. Another possibility is to enumerate test cases and to return another mock-up value for each test case.