



Email: trainingatsmartbytes@gmail.com

Mock Services

1. Can you explain me on Mock Services?

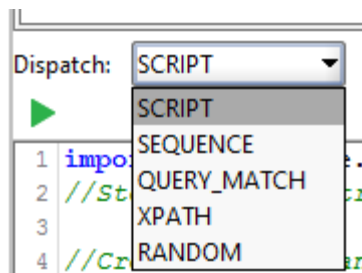
Mock services nothing but Simulation of actual web services. They come very handy when the actual service is not available. In mock service we may not achieve all the functionalities of the actual service, the reason is its very time consuming process (basically you are writing code for all scenarios, which is a duplicate effort).

Any example,

Let us say, we have a shopping cart service. During "check-out" process, we need to get payment confirmation from "Payment Gateway" service. This service might not be available for us during our tests. We can create a mock service, which can mimic the functionality of "makePayment" operation.

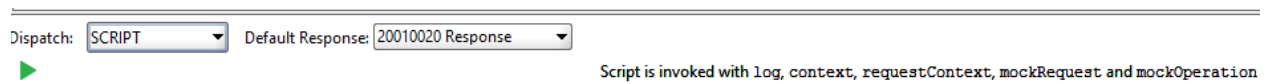
2. What are different dispatch strategies available ?

- Script (most commonly used)
- Sequence
- Random
- Xpath
- Query Match (uses XQuery expressions)



3. What are the available implicit objects in Script window (in mock service context) ?

- log
- context
- requestContext
- mockRequest
- mockOperation

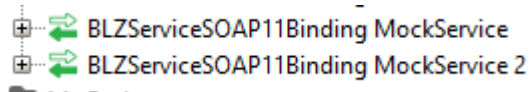


4. Can you explain steps involved in script to produce mock response?

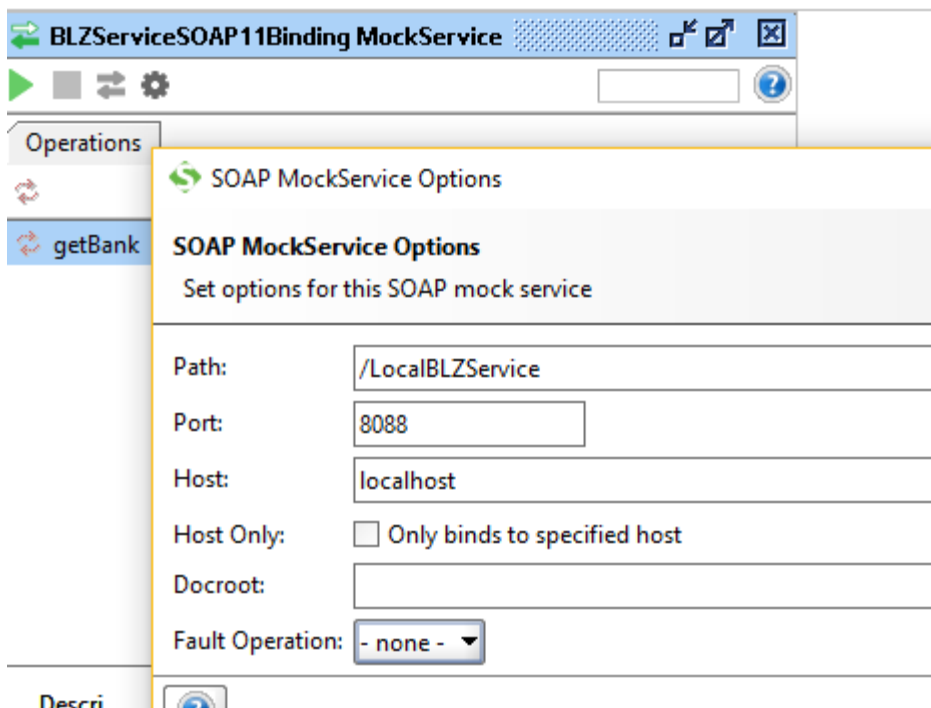
1. Instantiate class GroovyUtils
`groovyUtils = new GroovyUtils(context)`
2. Read mock request using
`def requestMessage = mockRequest.requestContent`
3. Convert the raw request into xml
`requestXml = groovyUtils.getXmlHolder(requestMessage)`
4. Extract the property from request
`requestXml.getNodeValue('///<property name>')`
5. Based on input combination, return appropriate response.

5. Can you create multiple mock services for same interface?

Yes (as long as mock service names are different)

**6. Can we change the "port" for mock service?**

Yes, not only port, we can control Path, Host as well.



7. Can you see the incoming requests in mock service?

Yes. Under "Message Log" window.

The screenshot displays the SoapUI MockService interface. On the left, a project tree shows a hierarchy starting with 'Demo' and 'Mock Services'. Under 'Mock Services', 'BLZServiceSOAP11Binding' is expanded, showing a 'getBank' operation with a 'Request 1'. Below this, 'BLZServiceSOAP12Binding' and 'BLZServiceSOAP11Binding MockService' are listed. The 'BLZServiceSOAP11Binding MockService' is selected, showing a 'getBank' operation with four responses: '20010020 Response', '55010111 Response', '55010424 Response', and '55010800 Response'. Below the tree, a 'My Project' section lists various test suites and services. At the bottom left, there are tabs for 'MockService Properties' and 'Custom Properties', with a table showing 'Property' and 'Value' columns.

The main window displays the 'BLZServiceSOAP11Binding MockService' running on port 8088. The 'Operations' tab is active, showing a list of operations with 'getBank' selected. Below the operations list, there are buttons for 'Descri...', 'Prop...', 'Start ...', 'Stop ...', 'OnReques...', and 'AfterReques...'. A checkbox labeled 'Enable' is checked. The 'Message Log' tab is active, showing a log entry: '2017-04-15 08:06:53.666: [getBank] 14ms'.

Property	Value
----------	-------