Parameterization

A key feature for scripting dynamic content

Script recordings contain unique IDs that are only valid for one session during the recording.

- replay of the script will fail with the unique ID from recording
- unique IDs must be adapted dynamically during each replay

SAP End-User Experience Monitoring handles this dynamically generated IDs in the recording with during the process of Parameterization. The EEM Editor is looking for the first occurrence of an ID in a server response and tries to find all client requests where this ID is used for the communication between client and server. In simple words: The EEM Editor finds out where to get the value for a variable and where the variable is used later in the EEM Script.

There are two strategies for Parameterization:

- Automatic parameterization: EEM Editor comes with a set of pre-defined parameterization rules (AutoParam rules) to make well-known unique IDs dynamic.
  AutoParam rules are applied during import of the recording.
  Custom parameterization rules can be defined to cover additional IDs that must be replaced (e.g. To cover non-SAP solutions)
- Manual parameterization: Editor can be used to manually turn unique IDs dynamic

Automatic parameterization

During import of a recording the Editor tries to apply all parameterization rules. As result unique IDs are retrieved from responses of preceding requests.

Activating Parameterization Rules

The list of active search rules can be configured

- Configuration scope Editor
- By default almost all rules are active
- Disable in case of conflicts
- UI is intentionally read-only
After changing the param rules they can be re-applied to the recording without repeating the import completely.

Extending the AutoParam Rules

Additional AutoParam rules can be defined by, for instance, using the Create New Rule button in the Parameterization Rules popup. (This creates a file named AutoParams.xml in the Editor's global directory - e.g. the folder containing ConfigGlobal.xml).
How to fill the rule:

- **Rule type** (remark: after the type selection, a help panel is open)
  - *Simple*: The search values to be extracted from the response of a recording are identified by the surrounding text, given as left boundary and right boundary.
  - *Regex*: The search values are specified via a regular expression. A search is generated for each left/right boundary value combination. At run time, the same regular expression is used for the search.
  - *Regex2*: The search values are specified via a regular expression, as for *Regex*. But at run time, the same regular expression is used for the search. This rule should be preferred over *Regex*.

- **Rule name**: choose any name that does not exist yet

- **DefinePattern** (type regex only): Must be a valid Java regular expression containing exactly three groups:
  - Left-bound group
  - Match group (value of this group is extracted into the variable)
  - Right-bound group
  - Regular Expression
  - If you want to do additional grouping, e.g. for different alternatives of the left or right boundary, make sure to use non-capturing groups: `(?:X)`

- **LeftBoundText / RightBoundText** (type simple only): delimiter of the searched value

- **InHeader**: enable searching in response header

- **InBody**: enable searching in response body

- **IgnoreCase**: ignore the case during the parameterization.

- **ReplaceLB / ReplaceRB**: Restrict the insertion of variables for a value in requests to occasions of the string `ReplaceLB+value+ReplaceRB`.

- **RedefinitionPolicy**: *ReuseFirstDefinition* or *UseLatestDefinition*

Click Re-parameterize to apply the new rule

**Advanced**

As an advanced feature, you can also directly create/modify the AutoParams.xml file.

Here is an example of the file:
AutoParams.xml

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<AutoParamRules xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.sap.com/solman/eem/autoparams2.0.xsd">
  <Group Name="Customized">
    <Rule Type="Regex2" Name="eemtest_regex2" DefinePattern="((?:l1)|(?:l2))(.*?)(?:r1)|(?:r2))="/>
    <Rule Type="Regex" Name="eemtest_regex" DefinePattern="((?:l1)|(?:l2))(.*?)(?:r1)|(?:r2))="/>
    <Rule Type="Simple" Name="eemtest_simpleheader" LeftBoundText="l1" RightBoundText="r1" InBody="false" />
    <Rule Type="Simple" Name="eemtest_simplebody" LeftBoundText="l1" RightBoundText="r1" InHeader="false" ReplaceLB="a " ReplaceRB=" b" />
  </Group>
</AutoParamRules>
```

Put AutoParams.xml into the Editor's global directory

**Manual Parameterization**

**Important:** manual parameterization is NOT persisted after a Re-parameterize of the script!

**Identify a parameter that must be set dynamically**

- Review request URLs and parameters for strings that looks like a **unique ID**
- Disabling all static requests and hiding all disabled requests will make this operation easier
- Copy the Id candidate to clipboard and keep message ID in mind

![Image showing manual parameterization process]

**Define a search to message**

- Add a search to the message whose response delivers the parameter. This search assigns the parameter value to a variable:
  1. Edit the Searches for the message
  2. Create a new Search (here select the suitable type: Left Bound Right Bound, Regex, Rule, ...) 
  3. Indicate a name for the search
  4. Provide the mandatory parameters (depending on the selected type)
5. Indicate where to search: header or Body
6. Indicate what to do in case the search fails at runtime
7. Indicate the variable name and instance number (0=first occasion)

Replace recorded parameter by variable

- Replace the actual parameter value in the request by a reference to the variable. Use the variable defined in the search of the previous step