JavaScript adapters

Overview

Prerequisite: Make sure to read the Adapters Overview tutorial first.

JavaScript adapters provide templates for connection to various back-ends, such as HTTP, SQL, Cast Iron, SAP JCo, and SAP Netweaver. JavaScript adapters also provide a service discovery wizard, which you can use to autogenerate adapters for connecting to WSDL services and more.

Agenda

- Anatomy of JavaScript adapters
- Structure of a JavaScript adapter procedure
- JavaScript adapter procedures
- XML structure of JavaScript adapters
- JavaScript adapter types

Anatomy of JavaScript adapters

Each adapter consists of the following elements:

- An XML file, which describes the connectivity options and lists the procedures that are exposed to the application or other adapters
- A JavaScript file, which contains the implementation of procedures that are declared in the XML file
- Zero, one, or more XSL files, which contain a transformation scheme for retrieved raw XML data

Data that is retrieved by an adapter can be returned raw or preprocessed by the adapter itself. In either case, it is presented to the application as a **JSON object**.

JavaScript adapter procedures

Procedures are declared in XML and are implemented with server-side JavaScript, for the following purposes:

- To provide adapter functions to the application
- To call back-end services to retrieve data or to perform actions

By using server-side JavaScript, a procedure can process the data before or after it calls the service. You can apply more filtering to retrieved data by using simple XSLT code.

JavaScript adapter procedures are implemented in JavaScript. However, because an adapter is a server-side entity, it is possible to use Java in the adapter code.

XML structure of JavaScript adapters

```
<wl:adapter name="HelloWorld">
  <displayName />
  <description />
  <connectivity>
    <connectionPolicy>
    ...
    <loadConstraints>
    ...
  </connectivity>
  <procedure />
  <procedure />
  ...
</wl:adapter>
```

- `name:` Mandatory. The name of the adapter
- `displayName`: Optional. The name that is displayed in the MobileFirst Console
- `description`: Optional. Additional information that is displayed in the MobileFirst Console
- `connectivity`:
  - Defines the connection properties and load constraints of the back-end system.
  - When the back-end system requires user authentication, defines how user credentials are obtained.
- `procedure`: Declares a service for accessing a back-end application. One entry for each adapter procedure.

```
<procedure name="procedure1"></procedure>
<procedure name="procedure2"></procedure>
```

Structure of a JavaScript adapter procedure

Each procedure that is declared in the adapter XML file must have a corresponding function in the JavaScript file.
The `WL.Server` API defines a procedure logic in JavaScript.

```
function procedure1(param) {
  return WL.Server.invokeSQLStatement({
    preparedStatement: procedure1Statement
,
    parameters: [param]
  });
}
```

Using global variables

Depending on your infrastructure and configuration, your MobileFirst server may be running with `SessionIndependent` set to true, where each request may reach a different node and HTTP sessions are not used.
In such cases you should not rely on global variables to keep data from one request to the next.

JavaScript adapter types