



Session Variables Tutorial

Session variables are a special type of substitution variables in DigDash Enterprise. Other types of substitution variables are user variables (`${user.<varname>}`), date variables (`${date.<format>}`) and server variables (`${server.<varname>}`).

Session variables syntax: `${session.<varname>}`

Like other variables, they can be used in almost any part of a flow: data model, visualization properties and flow properties.

They are used (substituted) during a flow and cube refresh.

This document illustrates a possible usage for the session variables.

I. LIFE CYCLE

The scope of session variables is a user session.

1.1 Instanciation

It is created once a user logs in, for this and only this session (the value is not shared across different sessions).

The initial value is an empty string, but if a user variable with the **same name** exists for this user, then the initial value will be the value of the user variable.

Example:

A user has a **myvar** attribute defined to "test": `${user.myvar}` is "test"

Consequently the initial value of the `${session.myvar}` is "test", while the initial value of another variable `${session.myemptyvar}` is an empty string.

1.2 Value Change

If the session variable value is changed during a session, the change is visible only for this session. Other sessions will keep their own variable values.

1.3 End Of Life

The variable value is deleted when the session is closed.

II. USAGE

Session variable can be used everywhere a user variable can be used in a flow, non-exhaustively:

- Data sources: queries, file paths, row filters, derived measures formulas and filters, targets, groups, etc.
- Visualization parameters: axis filters,
- Flow parameters: export entries name

Basically, session variables are very close to user variables in their usage. The difference is that it is easier to temporarily change a session variable value during a user session.

It makes session variables a powerful tool when coupled with flows and cubes that are refreshed on open.

The following tutorial will explain how to interactively refresh flows with parameters, using session variables and DigDash Javascript API.

III. TUTORIAL: INTERACTIVE REFRESH OF A FLOW WITH PARAMETERS

III.1 Dependencies

This tutorial makes use of:

- Linked data sources paths (aka WLNK)
- Scheduling options on a flow and on a data source
- The editor widget in Dashboard Editor
- DigDash Javascript API method from an hyperlink in a Dashboard

III.2 Context

In this deployment you have a set of folders, each one containing a CSV file named **data.csv**. All files **data.csv** files are exactly of the same structure but represent different scenarios:

- Scenario1
 - data.csv
- Scenario2
 - data.csv
- Scenario3
 - data.csv
- etc.

The set of folders itself is not restricted and can expand in the future.

You want your users to be able to look at different scenarios metrics (once at a time) in a dashboard page.

One possible way to do it would be to concatenate all **data.csv** files into one big data source, with a new scenario column. This will create a large cube containing all the scenarios data.

The drawback is if the data set is very large and/or there are a lot of scenarios. The refresh time can be very long for the concatenated source. And this is not optimal because the users may not look at all scenarios everyday.

The session variables can help you configuring a data source and flows that will have the scenario name (folder name) as a refresh parameter. Then a scenario is refreshed only on a user's request for that specific scenario.

III.3 Users Configuration

1. Connect to Digdash Enterprise User Management page
2. At the bottom of the left column, add a new user parameter named **scenarioVar**.
3. Set its default value to "Scenario1".

Important:

Please note that creating a new user variable is taken into account at user's next login. So if your user is logged in the Enterprise Administrator, and/or Dashboard Editor/Viewer, you must log out and reconnect before continuing this tutorial.

III.4 Data Source Configuration

1. Connect to the Enterprise Administrator. If it was already connected, disconnect and reconnect to take the new user variable **scenarioVar** into account.
2. Create a new document server that points to the folders containing all scenarios folders.
3. Create a new CSV File data source
4. Choose the document server you created for this example
5. Add a new document to it: Choose the "new link" option in the new document dialog. Click on Browse to select the source for the link, and choose one of the **data.csv** file.

The link content should look like this: 12345678|Scenario1/data.csv

6. Now we want this link to represent not only **data.csv** from Scenario1 but **data.csv** from *any* scenario:

=> Change the link to: 12345678|**#{session.scenarioVar}**/data.csv

Note: This will work only if you previously created the user variable **scenarioVar** and set its default value to an existing scenario folder. If there is an error while creating the link, you probably missed the user variable creation in previous paragraph. Please note again that creating a new user variable is taken into account at user's next login.

7. The **data.csv** file from Scenario1 is loaded and previewed (Scenario1 is the default value for **scenarioVar**). You can configure your data source as usual.

Important:

Before finishing the data source configuration, schedule the data source (*Refresh* tab in the second data source panel) to **refresh on open**. You may limit the frequency of the refresh to once a day.

III.5 Flow Configuration

Create a new flow using the new data source as usual.

Important:

Make sure you schedule the flow (*Schedule* tab in the properties dialog of the flow) to **refresh on open**. You may limit the frequency of the refresh to once a day.

III.6 Dashboard Editor

1. Connect to the Dashboard Editor
2. Add the new flow on a blank page. It should refresh/display the data generated from Scenario1/data.csv because Scenario1 is the default value you set on **scenarioVar** user variable
3. Edit the portlet properties to copy the flow identifier in the clipboard

Now we want to let the user change the scenario value. For this we are going to create a text field where the user can enter the scenario name he wants to see:

1. Add a new "Editor" portlet (static content) to the page. This type of portlet is used to add static content to a dashboard page (title, description, images...). Here we are using this portlet to add an interface to change the current scenario.
2. Edit the Editor and click on the HTML button to directly edit the editor's HTML
3. Add a text field, an OK button and the Javascript logic by typing the following HTML code (*do not forget to set the correct flow ID in the link's javascript below, in place of "FlowId"*):

```
<input type="text" id="scenarioName"
value=""></input>

<input type="button" value="OK"
onclick="ddCtrl.setSessionAttribute('scenarioVar',
document.getElementById('scenarioName').value);
ddCtrl.refreshFlow('FlowId');"></input>
```