Connect SIF3 Framework Consumers to HITS

# HITS environment

To be able to connect your consumer to HITS you must have a connection to a HITS Environment via the HITS Dashboard. You should have received a link to that Dashboard from NSIP. If you do not have that information, please contact NSIP to get you access to HITS.

For the rest of this document it is assumed that you have access to the HITS Dashboard.

The URL to the HITS Dashboard is of the form [http://hits.dev.nsip.edu.au/devdash/index.html?token=<some\_token](http://hits.dev.nsip.edu.au/devdash/index.html?token=%3csome_token)>. Use this URL in a browser to go to the HITS Dashboard that is for your personal use. Once on the HITS Dashboard, you should see your SIF3 HITS Environment information. It will look something like this:

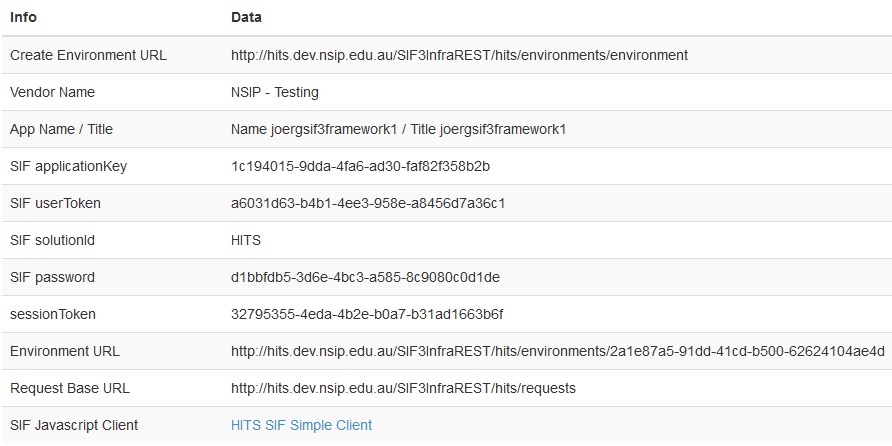


Figure : SIF3 HITS Environment Page

# Configure SIF3 Framework

To configure the SIF3 Framework to connect to the HITS environment listed in previous section you need to do the following steps:

**Step 1**: Set Solution ID in Environment Template

In your framework you should have a directory where the consumer’s SIF3 Environment templates are stored. This would generally be at the following location <installDIR>/config/environments/consumer/template. There is most likely an environment template called HITS.xml, demo.xml or devLocal.xml. If the HITS.xml doesn’t exist follow the steps below:

1. Create a copy of one of these templates in the same directory and name it something like HITS.xml
2. Open the newly created HITS.xml file and put the value of the “SIF solutionId” as seen from Figure 1 into the <solutionId> node. Your HITS.xml file should look something like this:

<environment xmlns="http://www.sifassociation.org/infrastructure/3.1">

<solutionId>**HITS**</solutionId>

<authenticationMethod>Basic</authenticationMethod>

<instanceId/>

<userToken/>

<consumerName></consumerName>

<applicationInfo>

<applicationKey></applicationKey>

<supportedInfrastructureVersion>3.1</supportedInfrastructureVersion>

<dataModelNamespace>http://www.sifassociation.org/au/datamodel/3.4</dataModelNamespace>

<transport>REST</transport>

<applicationProduct>

<vendorName>Systemic Pty Ltd</vendorName>

<productName>Test Driver</productName>

<productVersion>0.1alpha</productVersion>

</applicationProduct>

</applicationInfo>

</environment>

You can change the values under the <applicationProduct> node to any value that is applicable to you. Leave everything else as is.

1. Save your HITS.xml file.

**Step 2**: Configure the consumer’s properties file.

The consumer’s property file can be found in the directory <installDIR>/config/consumers. Let’s assume you have a properties file called StudentConsumer.properties. You need to set a few properties in that properties file with the values from Figure 1. The table below states the name of the property and what it must be set to.

|  |  |
| --- | --- |
| **Property Name** | **Value** |
| env.xml.file.name | Name of xml file created in Step 2 (i.e. HITS.xml) |
| env.application.key | Value of “SIF applicationKey” from Figure 1 |
| env.pwd | Value of “SIF password” from Figure 1 |
| env.userToken | Value of “SIF userToken” from Figure 1 |
| env.baseURI | Value of “Create Environment URL” from Figure 1 |
| env.authentication.method\* | Basic |
| env.create.conflictIsError\* | false |

\*This value is not shown in Figure 1. Please use the value given in this table.

If the “sessionToken” and “Environment URL” from Figure 1 are set then it means that a SIF Environment is already created and you want your consumer to connect to this existing environment. In this case you need to set the following additional properties in the StudentConsumer.properties file:

|  |  |
| --- | --- |
| **Property Name** | **Value** |
| env.use.existing\* | true |
| env.existing.sessionToken | Value of “sessionToken” from Figure 1 |
| env.existing.environmentURI | Value of “Environment URL” from Figure 1 |

\*This value is not shown in Figure 1. Please use the value given in this table.

**Step 3**: Start your Consumer

After you have applied the configurations in the previous two steps you should be able to start your consumer. Please ensure that your consumer uses the correct properties file (the one you changed in step 2 above). Verify this by looking at the consumer executable and ensure that the ConsumerLoader is initialised with the correct property file (example below):

ConsumerLoader.initialise("**StudentConsumer**");

Note: You MUST NOT provide the “.properties” file extension!