Custom Processors

**Apache NiFi Processors and Controller Services**

A NiFi Processor is the basic building block for creating an Apache NiFi dataflow. Processors provide an interface through which NiFi provides access to a flowfile, its attributes and its content. Writing your own custom processor provides a way to perform different operations or to transform flowfile content according to specfic needs.

A NiFi Controller Service provides a shared starting point and functionality across Processors, other ControllerServices, and ReportingTasks within a single JVM. Controllers are used to provide shared resources, such as a database, ssl context, or a server connection to an external server and much more.

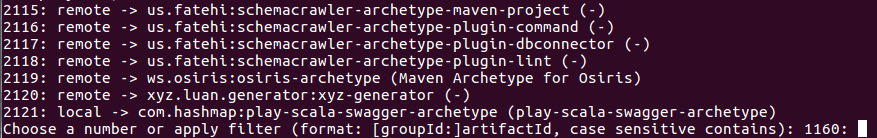
So, let’s dig into creating a custom processor, creating a custom controller service, and lastly creating a custom processor that will use a custom controller service.

**Steps for Creating a Custom Apache NiFi Processor**

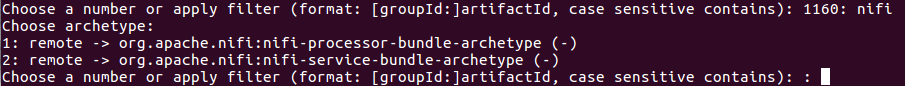
1. Maven archetype provides us the easiest way to get started with our own NiFi processor.

$ mvn archetype:generate

2. This will list out all the archetypes with maven, and we will get output something like this:



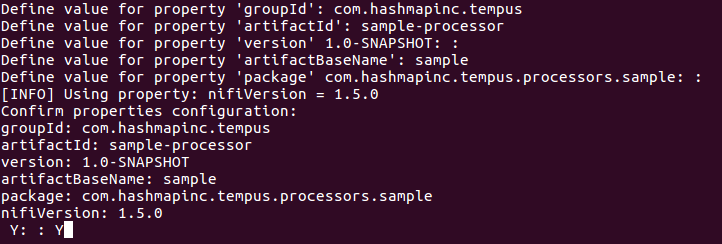
3. Apply filter type nifi and we will see the following output:



4. To create a processor select option 1, i.e org.apache.nifi:nifi-processor-bundle-archetype. This will list different versions of processor archetypes. We will select one according to our choice — default will be the latest version.



5. After selecting the version, we will see a menu to define different properties for a maven project. Properties like groupId, artifactId, version and few more as follows :



6. We can also use this command to generate maven archetype:  
$ mvn archetype:generate

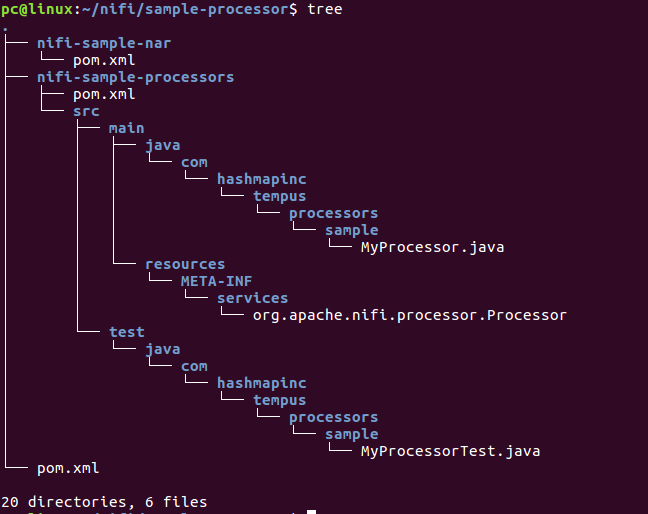
-DarchetypeGroupId=org.apache.nifi

-DarchetypeArtifactId=nifi-processor-bundle-archetype

-DarchetypeVersion=1.5.0

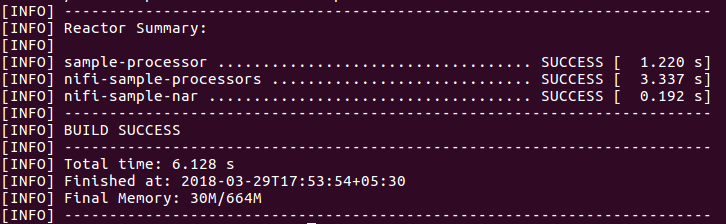
-DnifiVersion=1.5.0

7. After completing the steps, a maven project with default MyProcessor will be created. Directory structure will be as follows :



8. To deploy MyProcessor in nifi, we will change our directory to our processor project directory and build the project:  
$ cd sample-processor  
$ mvn clean install

9. Output of the build command will be as follows:

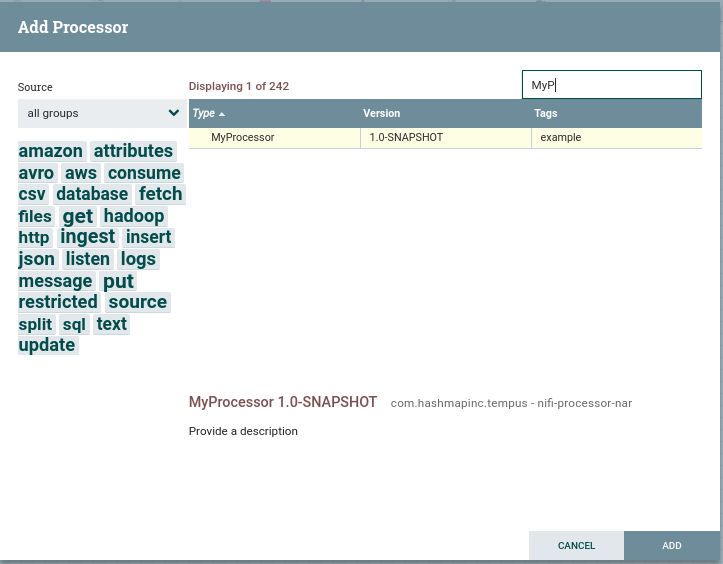


10. Now, copy the build nifi-sample-nar-1.0-SNAPSHOT.nar file to nifi lib directory and restart nifi:  
$ cp nifi-sample-nar/target/nifi-sample-nar-1.0-SNAPSHOT.nar NIFI\_HOME/lib  
$ NIFI\_HOME/bin/nifi.sh start

*A quick note — if you want to understand more about NiFi NAR files, please see the Hashmap blog post*[*NiFi NAR files Explained*](https://medium.com/hashmapinc/nifi-nar-files-explained-14113f7796fd)*.*

11. Open a browser and navigate to the nifi UI at <http://localhost:8080/nifi>

12. On the Add Processor window we will able to see our processor aptly named MyProcessor:



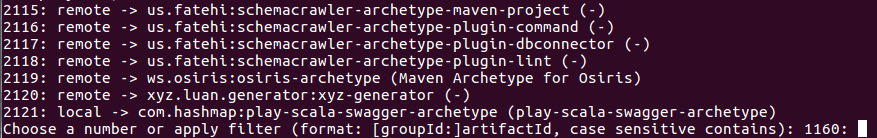
And there you go — you’ve just created a custom processor in Apache NiFi.

**Steps for Creating a Custom Apache NiFi Controller Service**

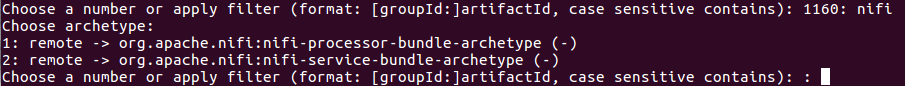
1. Similar to creating a custom processor, maven provides the easiest way to create a custom controller service.

$ mvn archetype:generate

2. This will list out all the archetypes with maven and we will get output something like this:



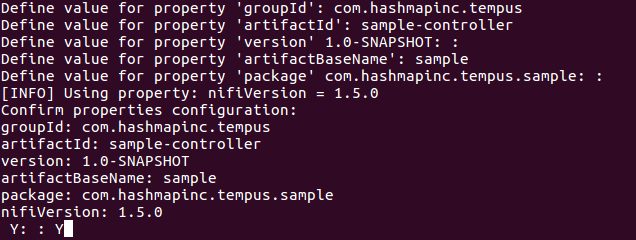
3. Apply filter type nifi and we will see the following output:



4. Select option 2, org.apache.nifi:nifi-service-bundle-archetype. This will list different versions of custom controller archetypes. We will select one according to our choice, and default will be the latest version.



5. After selecting the version, we will get a menu to define different properties for our maven project. Properties like groupId, artifactId, version and few more as follows :



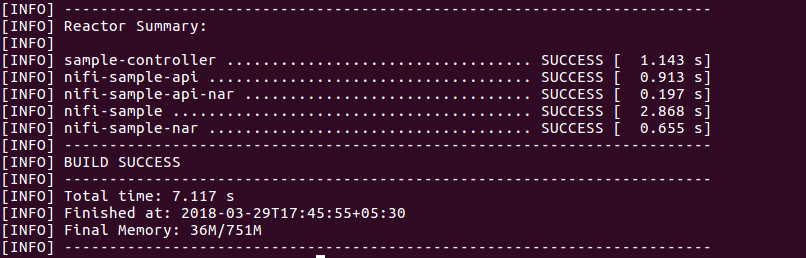
6. We can also use the command to generate maven archetype :  
$ mvn archetype:generate -DarchetypeGroupId=org.apache.nifi -DarchetypeArtifactId=nifi-service-bundle-archetype -DarchetypeVersion=1.5.0 -DnifiVersion=1.5.0

7. After completing the steps, a maven project with be created with default StandardMyService and MyService as controller APIs. The directory structure will be as follows:



8. To use this controller service in nifi, simply change our directory to the processor directory and build the project:  
$ cd sample-controller  
$ mvn clean install

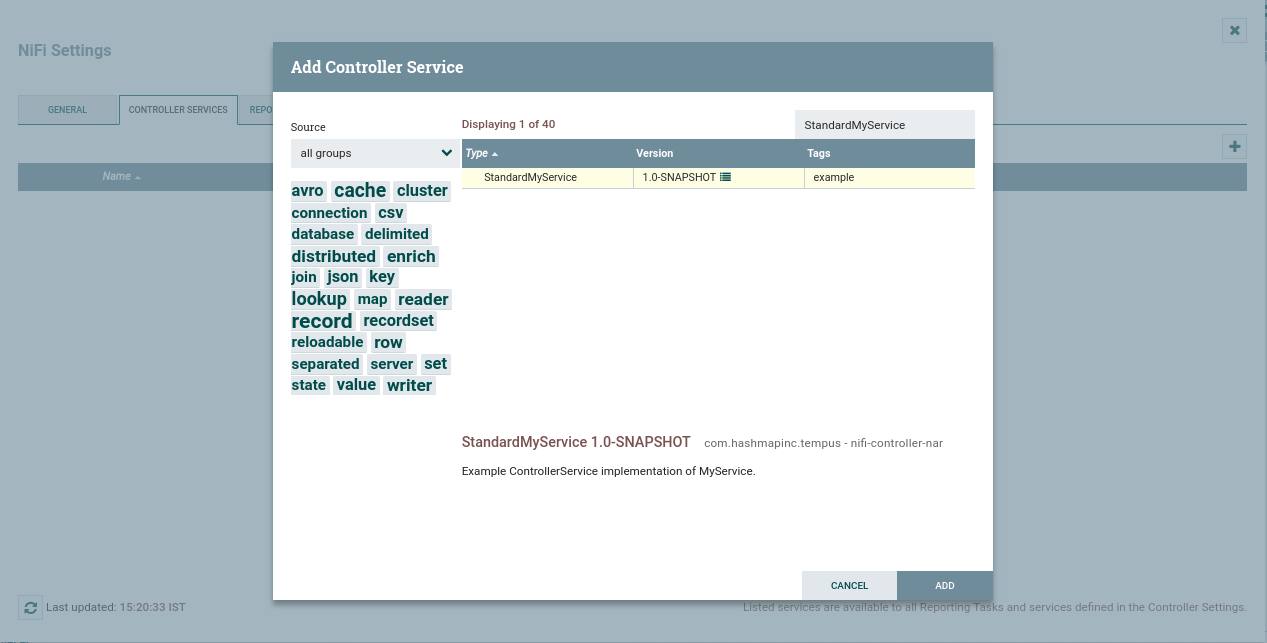
9. Output of the build command will be as follows:



10. Now, copy the build nar files to the nifi lib directory and restart nifi:  
$ cp nifi-sample-api-nar/target/nifi-sample-api-nar-1.0-SNAPSHOT.nar NIFI\_HOME/lib  
$ cp nifi-sample-nar/target/nifi-sample-nar-1.0-SNAPSHOT.nar NIFI\_HOME/lib  
$ NIFI\_HOME/bin/nifi.sh start

11. Open a browser and navigate to the nifi ui <http://localhost:8080/nifi>

12. Upon opening the controller service window we will see our StandardMyService listed as follows:



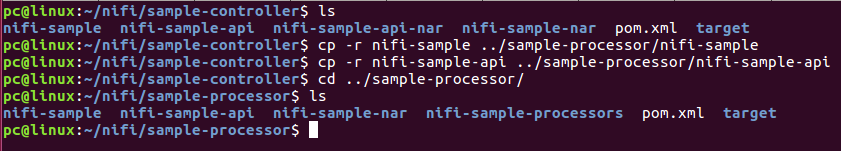
We’ve now created custom Controller Service in Apache NiFi.

**Steps for Creating a Custom Processor Using a Custom Controller Service with Apache Nifi**

Now that we’ve created both a custom NiFi processor and a custom NiFi controller service, let’s focus on creating a custom Processor using a custom Controller Service.

For this we will be using our previously created sample-processor and sample-controller projects.

1. First, copy the nifi-sample and nifi-sample-api from the sample-controller project directory to the sample-processor project directory:  
   $ ls  
   $ cp -r nifi-sample ../sample-processor/nifi-sample  
   $ cp -r nifi-sample-api ../sample-processor/nifi-sample-api  
   $ cd ../sample-processor/  
   $ ls

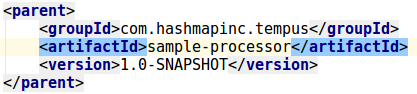


2. The Directory Tree will be as follows :

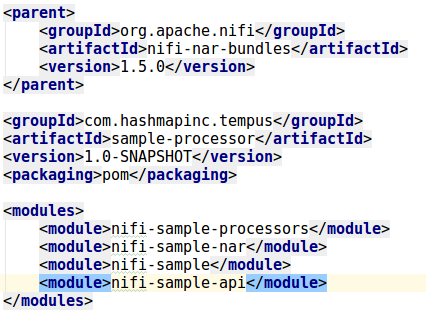


3. Open sample-processor in your IDE of choice — personally, I prefer Intellij. We are going to edit some files — feel free to use any other editor you like.

4. Open the pom file in nifi-sample and change its parent artifactId from sample-controller to sample-processor. Follow the same step with the nifi-sample-api pom file.



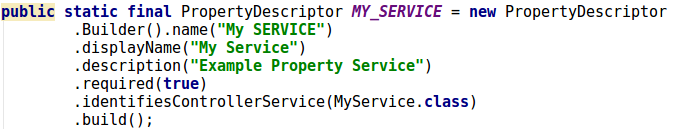
5. Now open the main project sample-processor pom file and add two modules to it nifi-sample and nifi-sample-api as follows:



6. To use the Controller Service in MyProcessor, we need to add the nifi-sample controller module as a dependency to the pom file of nifi-sample-processor:

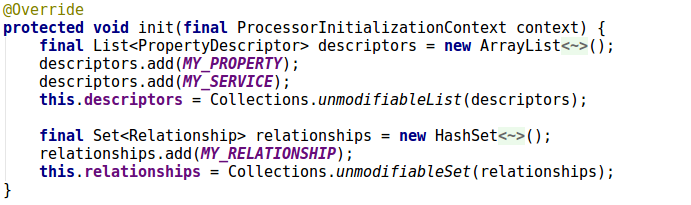


7. To add the Controller Service to the processor as an property, open MyProcessor.java and add following line:

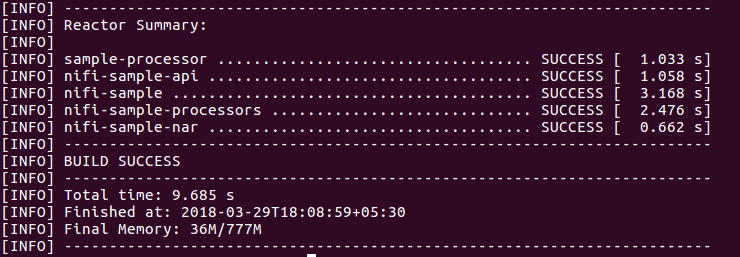


NOTE : Do not forget to import : com.hashmapinc.tempus.sample.MyService

8. Add the MY\_SERVICE property to the init() function of MyProcessor :



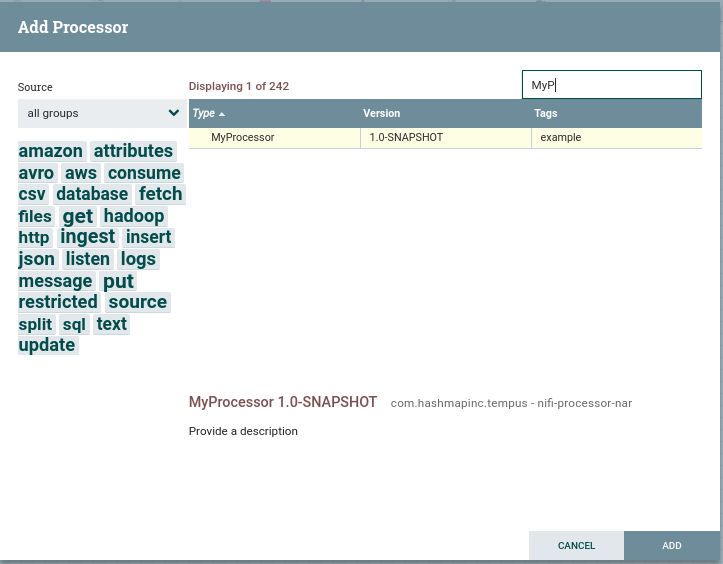
9. Now, move back to terminal and let’s build our maven project by using mvn clean install.



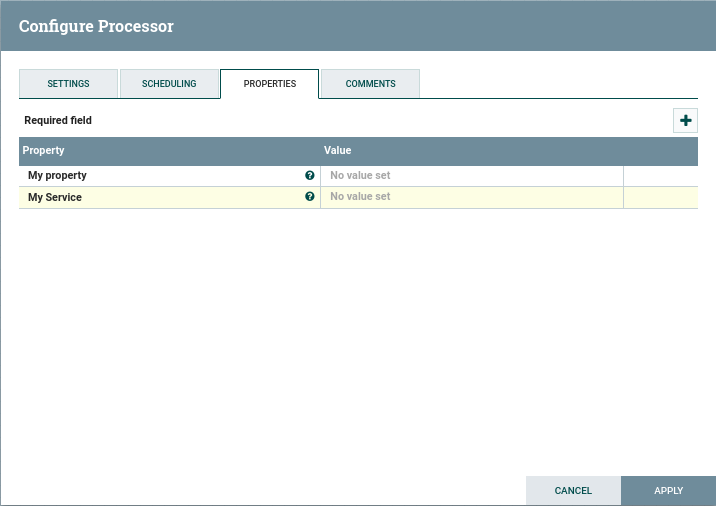
10. Copy the nifi-sample-nar-1.0-SNAPSHOT.nar file to nifi lib directory and restart nifi:  
$ cp cp nifi-sample-nar/target/nifi-sample-nar-1.0-SNAPSHOT.nar NIFI\_HOME/lib  
$ NIFI\_HOME/bin/nifi.sh start

11. Open a browser and navigate to nifi ui <http://localhost:8080/nifi>

12. On the Add Processor window we’ll see our processor named MyProcessor:

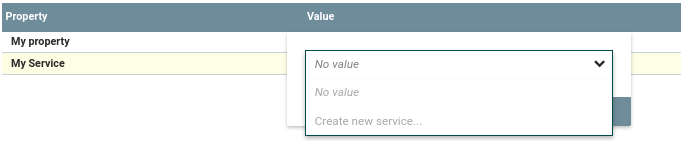


13. After adding the processor, right click on MyProcessor and open its configuration window. Under the Properties tab we will see something like this:

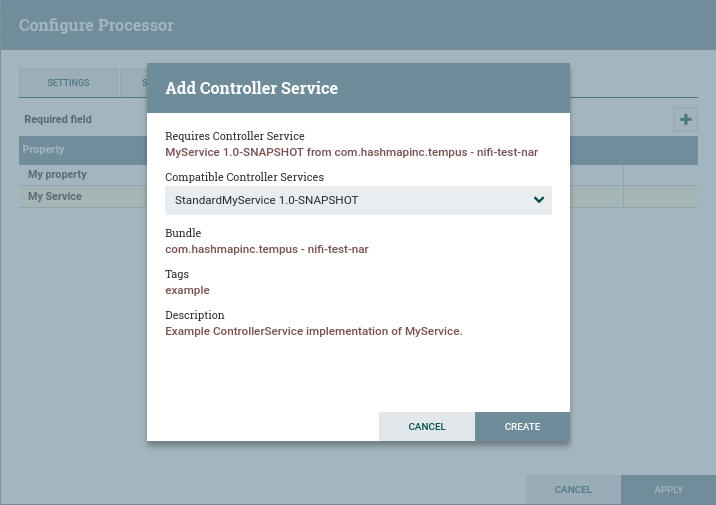


14. Set My Property as a random text value.

15. Clicking on My Service Property will give an option to create a new service, click on that.



16. The above step will open the below window to create a service, click on Create…

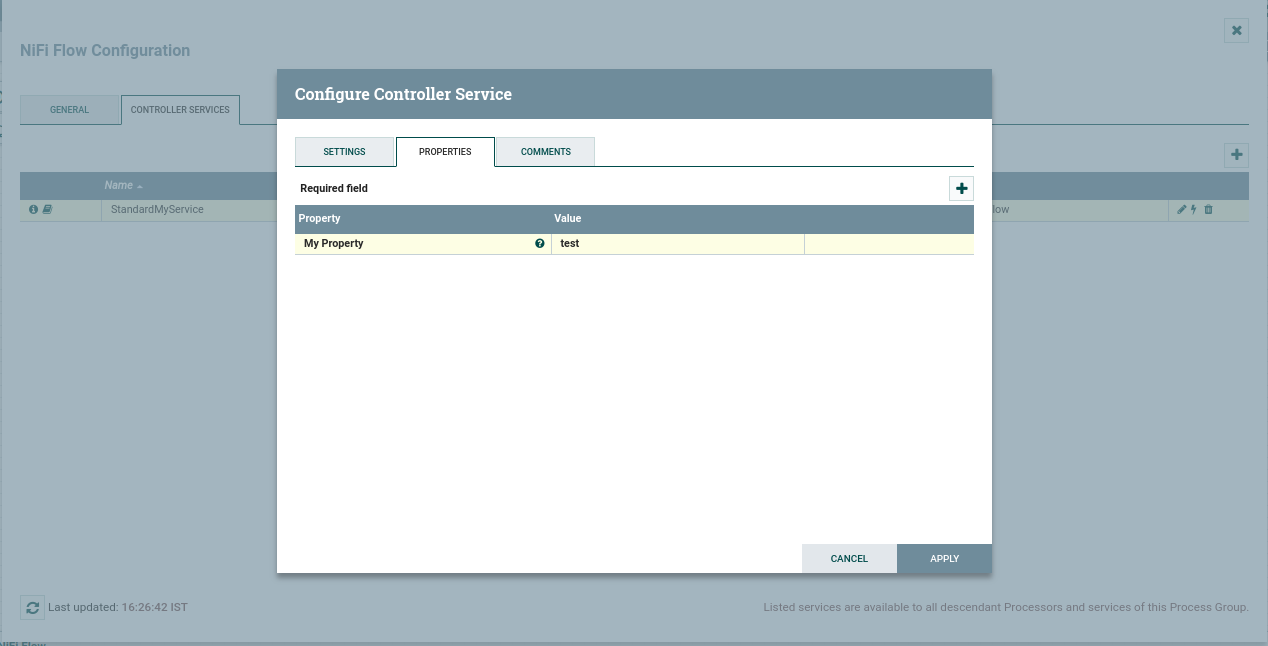


17. Click on the arrow to configure the Service :





18. This will open a window and just click on the pencil icon on right most column to configure My Property



19. Now, click on the lightning symbol to enable the service.