

Building An Application Using DataSets

Exercise Objectives

In this exercise, you will:

- Add DataSets to the example project
 - Implement using DataSets in Flowlets
 - Ingest data to the DataSets using `curl` commands
-

Exercise Steps

To the previous example, now add DataSets

- Specify imports
- Modify the `ApplicationSpecification`
- Modify the `Update` Flowlet to use the DataSets

Add these imports:

```
import com.continuity.api.data.dataset.SimpleTimeseriesTable;
import com.continuity.api.data.dataset.TimeseriesTable;
import com.continuity.api.data.dataset.table.Get;
import com.continuity.api.data.dataset.table.Increment;
import com.continuity.api.data.dataset.table.Row;
import com.continuity.api.data.dataset.table.Table;
import com.google.common.base.Splitter;
import com.google.common.collect.Iterables;
```

Exercise Steps

Change the `ApplicationSpecification` replacing `.noDataSet()` with:

```
.withDataSets()  
  .add(new Table("sentiments"))  
  .add(new SimpleTimeseriesTable("text-sentiments"))
```

Change the `Update` class to use the `DataSets` by adding:

```
@UseDataSet("sentiments")  
private Table sentiments;  
  
@UseDataSet("text-sentiments")  
private SimpleTimeseriesTable textSentiments;
```

Exercise Steps

Change the `while` statement of the `process` method to read:

```
while (sentimentItr.hasNext()) {
    String text = sentimentItr.next();
    Iterable<String> parts = Splitter.on("---").split(text);
    if (Iterables.size(parts) == 2) {
        String sentence = Iterables.get(parts, 0);
        String sentiment = Iterables.get(parts, 1);
        sentiments.increment(new Increment("aggregate", sentiment, 1));
        textSentiments.write(new TimeseriesTable.Entry(sentiment.getBytes(Charsets.UTF_8),
                                                         sentence.getBytes(Charsets.UTF_8),
                                                         System.currentTimeMillis()));
    }
}
```

Build, Deploy and Run

- Check that Reactor is running
 - If Reactor was already running, stop any existing Flows; you can reset the Reactor using the link on the `Overview` tab of the Reactor Dashboard
 - Build the App using `mvn clean package`
 - Deploy the App by dragging and dropping
 - Send sentences with sentiments using `curl` and watch them run through the Flow system
 - Examples on the following slide
 - Note: The sentiments follow the sentence, and are either positive, negative or neutral, and are separated from the sentence by three dashes
-

Example Commands with Sentences

```
curl -o /dev/null -sL -w "%{http_code}\\n" -d  
  "Continuity Reactor is awesome---positive"  
  http://localhost:10000/v2/streams/sentence  
  
curl -o /dev/null -sL -w "%{http_code}\\n" -d  
  "I have hard time building apps on Hadoop---negative"  
  http://localhost:10000/v2/streams/sentence  
  
curl -o /dev/null -sL -w "%{http_code}\\n" -d  
  "Hadoop is a Big Data platform---neutral"  
  http://localhost:10000/v2/streams/sentence
```

Each command is to be on a single line

Exercise Summary

You should now be able to:

- Add DataSets to an example project
 - Implement using DataSets in Flowlets
 - Ingest data to the DataSets using `curl` commands
-

Exercise Completed

[Chapter Index](#)