Divolte Workshop: Exercise 2

In this exercise we're going to extend the schmema and mapping so that we have better quality data in the clickstream. We are going to:

1. Add some additional fields to the schema that describe where in the site page-views are occurring;
2. Update our mapping to derive this information from the location URLs.

Step 1: Updating the Schema

The first thing we're going to do is update our schema to include the additional fields. In your favourite editor:

1. Open up and edit: `divolte/ShopEventRecord.avsc`
2. Add in the following fields:

```
{ "name": "pageType",        "type": "string",           "default": "unknown" },
{ "name": "page",            "type": ["null", "int"],    "default": null },
{ "name": "productId",       "type": ["null", "string"], "default": null },
{ "name": "downloadId",      "type": ["null", "string"], "default": null },
{ "name": "category",        "type": ["null", "string"], "default": null }
```

Don't forget to add a trailing comma (`,`) at the end of the last field before these.

1. Restart the stack with your new schema and check the Divolte logs to make sure everything started up properly:

   ```
   % ./refresh
   % docker-compose logs -f divolte
   ```

If everything startup properly you should see something like:

```
[main] INFO  [SchemaRegistry]: Loading Avro schema from path: /etc/shop/divolte/ShopEventRecord.avsc
[main] INFO  [SchemaRegistry]: Loaded schemas used for mappings: [shop_mapping]
…
[main] INFO  [Server]: Initialized sources: [browser]
[main] INFO  [Server]: Starting server on 0.0.0.0:8290
```

If this does not happen there should be something in the logs indicating the problem.

1. Browse to the shop (http://localhost:9011/), and click around a bit.
2. After waiting for a minute or so, examine the events again:

   ```
   % docker-compose run divolte show-avro
   ```

The most recent events should now include the additional fields, although for now they're empty...

Step 2: Updating the Mapping

In the first step we updated our schema to include some additional fields. In this step we're going to update the mapping so that these are filled in.

1. Open up and edit: `divolte/mapping.groovy`
2. Look for the `EXERCISE 2: INSERT SECTION HERE` comment and add in the following:

```
section {
    def locationUri = parse location() to uri
    def locationPath = locationUri.path()
    when locationPath.equalTo('/') apply {
        map 'home' onto 'pageType'
        // Skip rest of section.
        exit()
    }
    // Regular expression for extracting pieces out of the path.
    def pathMatcher = match '^/([a-z]+)/([^/]+)?[/]?([^/]+)?.*' against locationPath
    when pathMatcher.matches() apply {
        map pathMatcher.group(1) onto 'pageType'
        when pathMatcher.group(1).equalTo('category') apply {
            map pathMatcher.group(2) onto 'category'
            when pathMatcher.group(3).isAbsent() apply {
                map 0 onto 'page'
            }
            map { parse pathMatcher.group(3) to int32 } onto 'page'
            exit()
        }
        when pathMatcher.group(1).equalTo('product') apply {
            map pathMatcher.group(2) onto 'productId'
            exit()
        }
        when pathMatcher.group(1).equalTo('download') apply {
            map pathMatcher.group(2) onto 'downloadId'
            exit()
        }
        exit()
    }
}
```

1. Once again restart the stack with your new schema and check the Divolte logs to make sure everything started up properly:

   ```
   % ./refresh
   % docker-compose logs -f divolte
   ```

2. Browse to the shop (http://localhost:9011/), and click around a bit.
3. After waiting for a minute or so, examine the events again:

```
% docker-compose run divolte show-avro
```

The most recent events should now include the additional fields with values filled in from the mapping.

Things to Think About

- If we had a search function that worked by submitting to `/search?q=SOMETHING`, how would you extend the mapping?
- How would you include the search query itself in events?