JDG + EAP Lab 7 Guide

This explains the steps for lab 6, either follow them step-by-step or if you feel adventurous try to accomplish goals without the help of the step-by-step guide.

Background

myTODO application is a success, but we don't know much about our users. The marketing department has requirements for tracking : which are the devices (Smartphones and Tablets, Computers) users are accessing myTODO with, the OS of those devices and their preferred browsers.

Use-case

We will implement a solution to store this user device information. To minimize any impact to performance, user device information should be stored unstructured, and via Map/Reduce pattern we can structure the data further and interprete it using reporting tools. The user device information is captured from the User-Agent HTTP header that browsers typically provide.

These are the main tasks of lab 7

1. Create a local library mode cache together with the RemoteCache
2. Extend the REST layer to store the unstructured request data in a local cache
3. Provide a BiSerivce (Business Intelligence Service) class that can structure the data and generate summarized views of the data.
4. Provide a BiEndpoint (REST Service) to enable UI access to the BiService

Step-by-Step

1. Open `~/jdg-labs/projects/lab7` project in JBoss Developer Studio

2. Open `/src/main/java/org/jboss/infinispan/demo/Config.java` and fix the `getLocalCacheManager` which should look like this:

```
private EmbeddedCacheManager getLocalCacheManager() {
    GlobalConfiguration glob = new GlobalConfigurationBuilder()
        .globalJmxStatistics().allowDuplicateDomains(true).enable().build();

    org.infinispan.configuration.cache.Configuration loc = new org.infinispan.configuration.cache.ConfigurationBuilder()
        .expiration().lifespan(1,TimeUnit.DAYS)
        .build();

    return new DefaultCacheManager(glob, loc, true);
}
```

3. Add these `import` statements

```
import org.infinispan.configuration.global.GlobalConfiguration;
import org.infinispan.configuration.global.GlobalConfigurationBuilder;
import org.infinispan.manager.DefaultCacheManager;
import org.infinispan.manager.EmbeddedCacheManager;
import java.util.concurrent.TimeUnit;
```

4. Open the `org.jboss.infinispan.demo.rest.TaskEndpoint.java` file, and make the following changes

```
@Inject private Cache<Long, String> requestCache;
```

Here the request cache is injected.

5. Now add the following line to all REST methods (ie: update(), listAll(), create() methods)

```
requestCache.putAsync(System.nanoTime(), headers.getRequestHeader("user-agent").get(0));
```

6. Add the following `import` statement

```
import org.infinispan.Cache;
```

7. Open `org.jboss.infinispan.demo.BIService.java` and make the following changes

8. Remove `@SuppressWarnings("unused")`

9. Add these `import` statements

```
import javax.inject.Inject;
import org.infinispan.AdvancedCache;
import org.infinispan.Cache;
import org.infinispan.distexec.mapreduce.MapReduceTask;
import org.jboss.infinispan.demo.mapreduce.CountReducer;
import org.jboss.infinispan.demo.mapreduce.UserBrowserVendorCountMapper;
import org.jboss.infinispan.demo.mapreduce.UserOSCountMapper;
```

10. Change the implementation of `getRequestStatiscsPerOs()` method to

    ```
    public Map<String,Integer> getRequestStatiscsPerOs() {
        return new MapReduceTask<Long, String, String, Integer>(requestCache.getAdvancedCache())
                .mappedWith(new UserOSCountMapper())
                .reducedWith(new CountReducer())
                .execute();
    }
    ```

    Change the implementation of `getRequestStatiscsPerBrowser()` method to

    ```
    public Map<String,Integer> getRequestStatiscsPerBrowser() {
        return new MapReduceTask<Long, String, String, Integer>(requestCache.getAdvancedCache())
                .mappedWith(new UserBrowserVendorCountMapper())
                .reducedWith(new CountReducer())
                .execute();
    }
    ```

    Change the implementation of `generateTestData()` method to

    ```
    public void generateTestData() {
                    Random random = new Random(System.currentTimeMillis());

                    for(int i=0;i<5000;i++) {
                            int agent = random.nextInt(fakeUserAgents.length);
                            requestCache.put(new Long(i), fakeUserAgents[agent] );
                    }
            }
    ```

11. Replace this code

    ```
    Map browserMap = new HashMap();
            Map osMap = new HashMap();
    ```

    with this

    ```
    @Inject
            Cache < Long,String > requestCache;
    ```

12. Analyse what the mapping classes `UserOSCountMapper` and `UserBrowserVendorCountMapper` aim to achieve

13. Open `CountReducer.java` and add the implementation like below:

    ```
    @Override
    public Integer reduce(String reducedKey, Iterator<Integer> iter) {
        int sum = 0;
        while (iter.hasNext()) {
            Integer i = (Integer) iter.next();
            sum += i;
        }
        return sum;
    }
    ```

14. Open `TaskServiceTest.java` and add `import org.infinispan.Cache;`

15. Analyse the `testRequestCache()` method

16. Run the JUnit test

17. Deploy the application using the following command from projects/lab7 dir

    ```
    $ mvn clean package jboss-as:deploy
    ```

18. Congratulations, you have completed all the labs.