# JDG + EAP Lab 5 Guide

This explains the steps for lab 5, 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

The TODO application is a huge success and over 1 million users are today using it to synchronise the task list between different devices. Even though each entry are typically small the data is growing fast. By mid next year numbers of users and expected to grow to 5 millions. However the success gives new challenges. The memory used to store tasks is growing fast.

### Memory usage today in the future

The average length of a title is 20 characters (about 80 bytes). Together with two Date objects (32 bytes each) and the other fields we can assume that each Task object will take about 200 bytes of memory. The average number of tasks per user is 50.

Today the effective data storage used (data without meta-data) is about 10 GB of data [1 million x 50 tasks per user x 200 bytes per task / (1024^3)].

Mid next year the prediction is that we will have to be able to store 50 GB of data in the grid.

Given that TODO application is using distributed with 2 owners per object, the used storage is probably closer to 25 GB today and 125 GB mid next year.

The TODO application today runs on 4 mid sized virtual servers with 10 GB allocated to JVM on each host.

To meet the predicted increase we have a accouple of options:

1. Increase the number of server 5 times
2. Use eviction together with a store to limit the amount of data in memory
3. Layer our solution by moving to client/server mode for the JDG cache allowing applications for more effective JVM memory management.

## These are the main tasks of lab 3

1. List pros and cons with different solutions to the memory issue.
2. Setup a standalone node of JDG
3. Rewrite the application to use HotRod client instead of native client.

## Step-by-Step

1. Discussion pros and cons with different options for solution to the memory usage issue
2. Setup of a standalone node, but running the init-lab.sh for lab4

```
$ ./init-lab.sh --lab=5
```

3. Setup development dependencies
4. Setup runtime dependencies
5. Update the Config class and TaskService to use RemoteCache instead of Cache.
6. Configure a cache in JDG server.
7. Start the JDG server (and EAP if not already started)

```
$ ./target/xxx/bin/standalone.sh
```

8. Run the arquillian tests to verify that the application.
9. Deploy the application.
10. Congratulations you are done with lab 5.