**MIND STORM SOFTWARE PVT LTD **  
**Google App Engine Training**

# ExamResults – Step 3 – Datastore Service

In this session, we are going to build the Data Layer for the application. In the previous session, we build the Web Interface for the ExamResults application but it was returning a dummy result. So what we plan to do here is the following:

1. Learn about Datastore Service in App Engine and build the Data Access Layer for Exam Result objects
2. Load the Datastore Service with sample data (4-5 Records)
3. Modify the Exam Results Servlet to query the results given a seat number and return the actual data. We will be using dummy data for the results, since we will be building out the database layer in the next session.

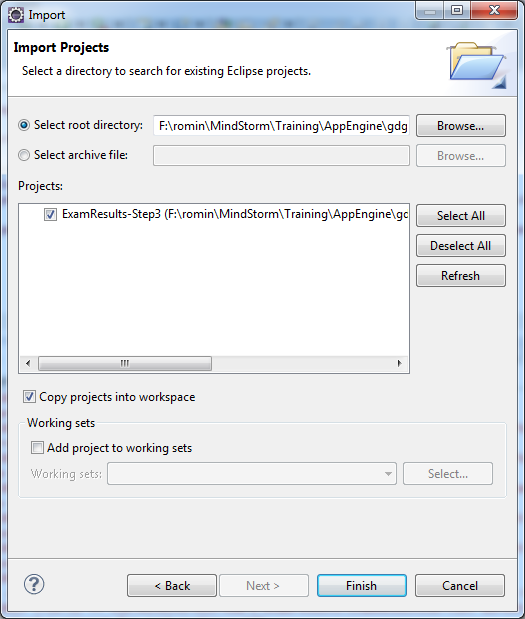
# ExamResults – Step 3 – See it in Action

In terms of the UI, it will still be the same, except that the data is now being returned from the Data store layer that we are going to be build.

Refer to **hands-on-exercises/Exam Results App/ExamResults-Step 3** for full project source code. You can also import it directly into Eclipse and study the code.

The Steps to import the project are:

1. Click on **File --> Import**
2. Select **General --> Existing Projects into Workspace.** Click on **Next.**
3. In the **Import Projects** dialog, Browse to the folder **hands-on-exercises/Exam Results App/ExamResults-Step 3** and tab out
4. It should show a dialog as given below. Select the Exam Results-Step3 project and ensure that Copy projects into workspace is checked as given below:

****

1. Click on Finish.
2. **Resolve AppEngine SDK:** Select your App Engine SDK. For that, select the Project, Right-click and go to Properties. Then visit Java Build Path and in the Order and Export tab, select the App Engine SDK that you have installed and click on OK.

# ExamResultEntity class

We are going to now persist the data in the datastore. You will find a class named ExamResultEntity.java, which will contain various methods for saving, updating, deleting and listing ExamResult entity classes.

Go to the Source folder and open up ExamResultEntity.java file. Notice that all we are doing is dealing with the Entity class and setting its various properties for adding or updating the record.

You will notice that we are using a class DBUtil.java that does the hard work of actually committing the changes and/or retrieving the data from the underlying datastore service.

# DBUtil.java

This class will provide utility methods that will be used by other modules in the application to add / modify/ delete /search for Exam Result datastore entities. This will therefore shield them from knowing the Datastore API in App Engine.

The class that you should study here are:

**src/com/mindstormsoftware/examresults/services/DBUtil.java**

It contains methods for the following:

* persistEntity(Entity)
* deleteEntity(Entity)
* findEntity(Key)
* listEntities(...)

Look at the methods in detail and you will find the Datastore API being used. You can use this generic class for any of your other projects also.

# Modify the Servlet (ExamResultsServlet.java) and results.jsp

The only change that we need to make to the ExamResultsServlet.java is to change the call from called getDummyResult(seatNumber) to the actual call shown below:

Entity \_result = ExamResultEntity.getExamResult(seatNumber);

The results.jsp file is changed so that the new Entity object is accessed and we only have to access the getProperty(...) method to extract out the different property values.

# Load Initial Data

The first thing to do is to load the application with sample data. The sample data file is present in **WEB-INF\sampleData\examresults.dat**. To load the data, simply invoke <http://localhost:8888/loaddata.jsp>. On successful execution, this should load the ExamResult entity with 4 records.

The examresults.dat file is a simple text file where each row is a Exam Result for a particular seat. Each field is separated by a semi-colon (;). The file is shown below:

226510;Romin Irani;60;70;80;90;300;75

226511;Smarth Behl;70;70;70;70;300;70

226512;Prajyot Mainkar;80;80;80;80;320;80

226513;Jonathan D'Souza;90;90;90;90;360;90  
226514;Dhrumil Shah;95;95;95;95;380;95  
226515;Paresh Mayani;100;90;100;90;380;95

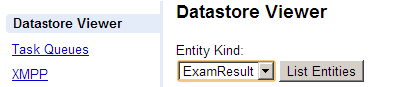
The loaddata.jsp file calls a utility class **DataLoader.java** that does the following:

1. Read the data file line by line.
2. Parse out the line into individual attributes and invoke the **ExamResultDAO.*INSTANCE*.add()** method to insert the records one by one.

# Test locally and Deploy

Once you made the changes, test out the changes locally first. Stop the Web Application if it is already running by clicking on the Stop icon in the Console window. Restart the application i.e. Right-click on Project and select **Run As 🡪 Web Application** and navigate to <http://localhost:8888>

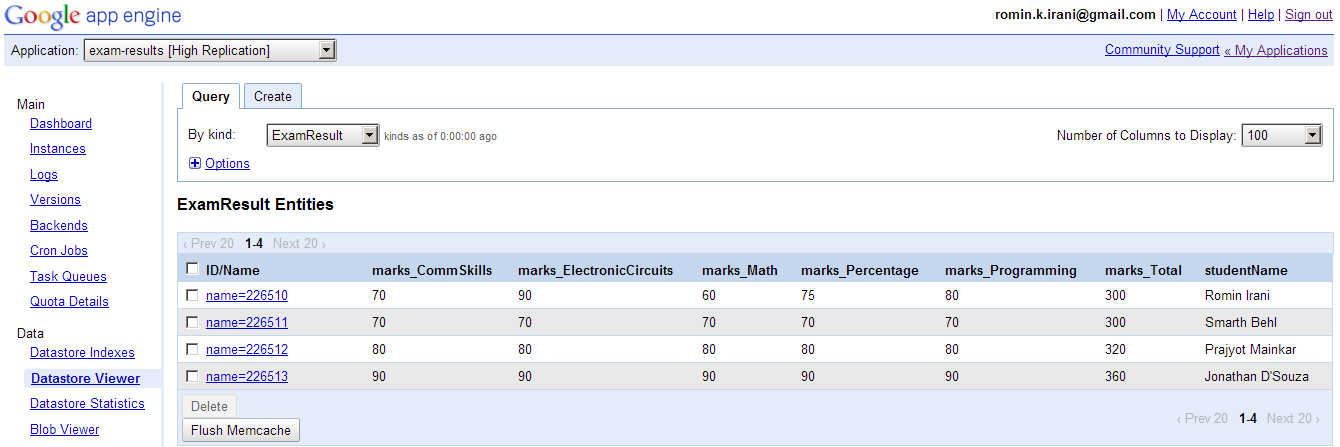
To view the data, you can go to the <http://localhost:8888/_ah/admin> and you should see the Entity **ExamResult** available as given below:



Click on the **List Entities** link to see the records.

You can also deploy your application to the cloud, if you wish. To deploy to the cloud, right-click on project and then **Google 🡪 Deploy to App Engine.**

**Remember to run the http://<appid>.appspot.com/loaddata.jsp file on the server so that it is also populated with dummy data. To view the data, login to Administrative Console in your app and go to Data 🡪 Datastore Viewer. You should see records populated as given under:**

****

# Next Session

In the next session, we shall see how to add XMPP support to the application. Users will then be able to get exam results directly from the Google Talk application.