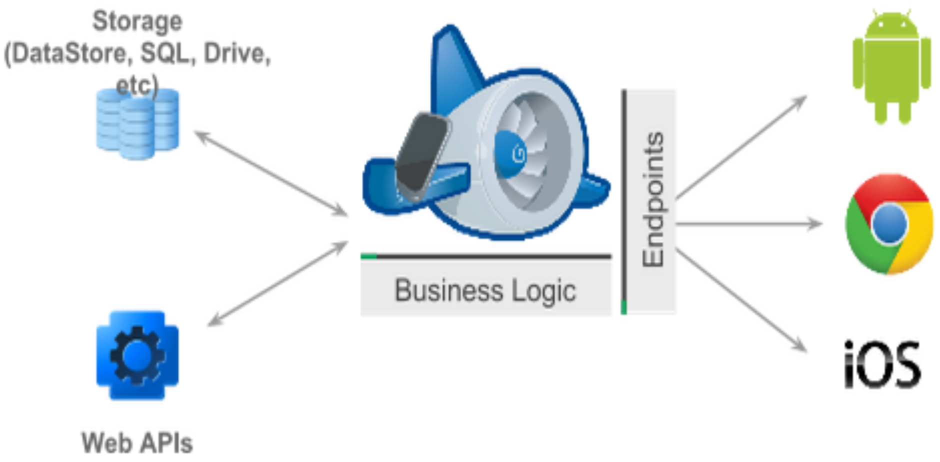
TODO List Manager on Steroids
Executable Prototype Document
Team 3.09

1. Introduction

The goal of the project is to extend the existent TODO List Application on Android to a Web application and to provide sync functionalities between the Android and the Web application, thereby enhancing the user-experience. For the Prototype we focused on the following two areas.

2. Sync

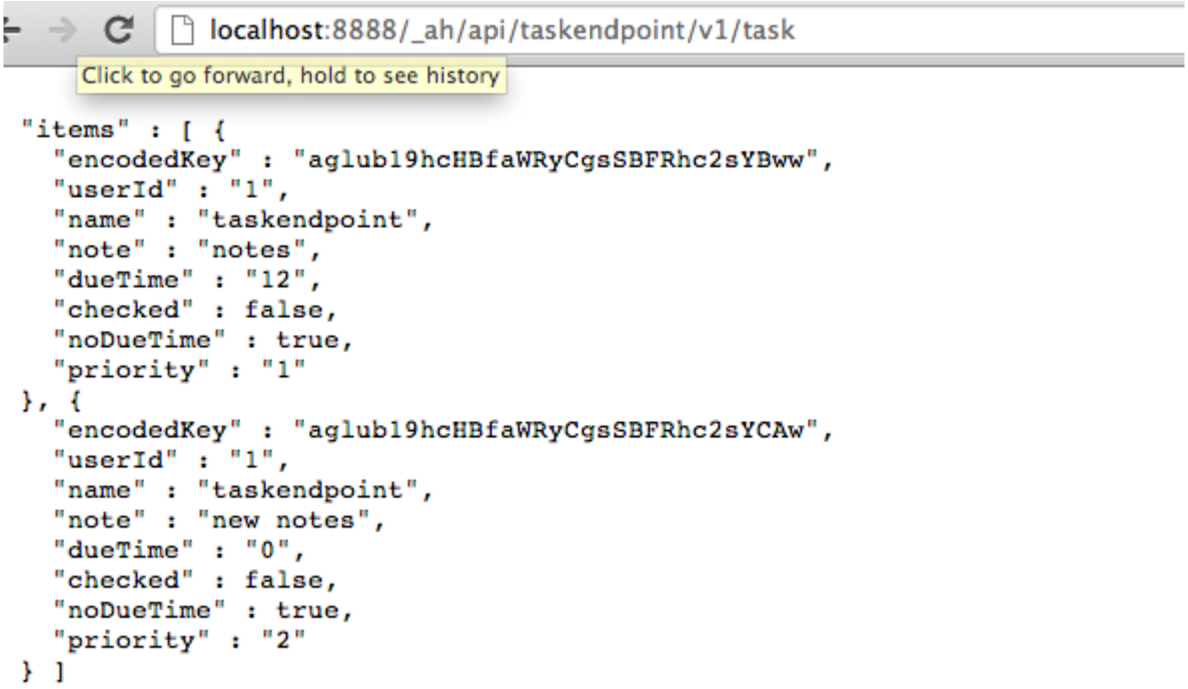
For communication between the two applications we decided to use the Google Cloud Endpoints that lets you create Apis in your Google app engine project. Cloud Endpoints can be used to create client side libraries for other web, android or iOS applications.



Refrence: http://endpoints-trusted-tester.appspot.com/

Google End Points is a new technology (June-July 2012), therefore it requires the developer to be registered in the Trusted Tester program in order for google servers to host the Apis created. I am still waiting to be approved to join this program and for this reason was unable to complete the sync functionality between the android and web applications.

However I did create an API for the users which is present in the WTM project under com.sdp.wtm.TaskEndPoint.
hosting this application locally and going to the http://localhost:8888/_ah/api/taskendpoint/v1/task shows the following results:

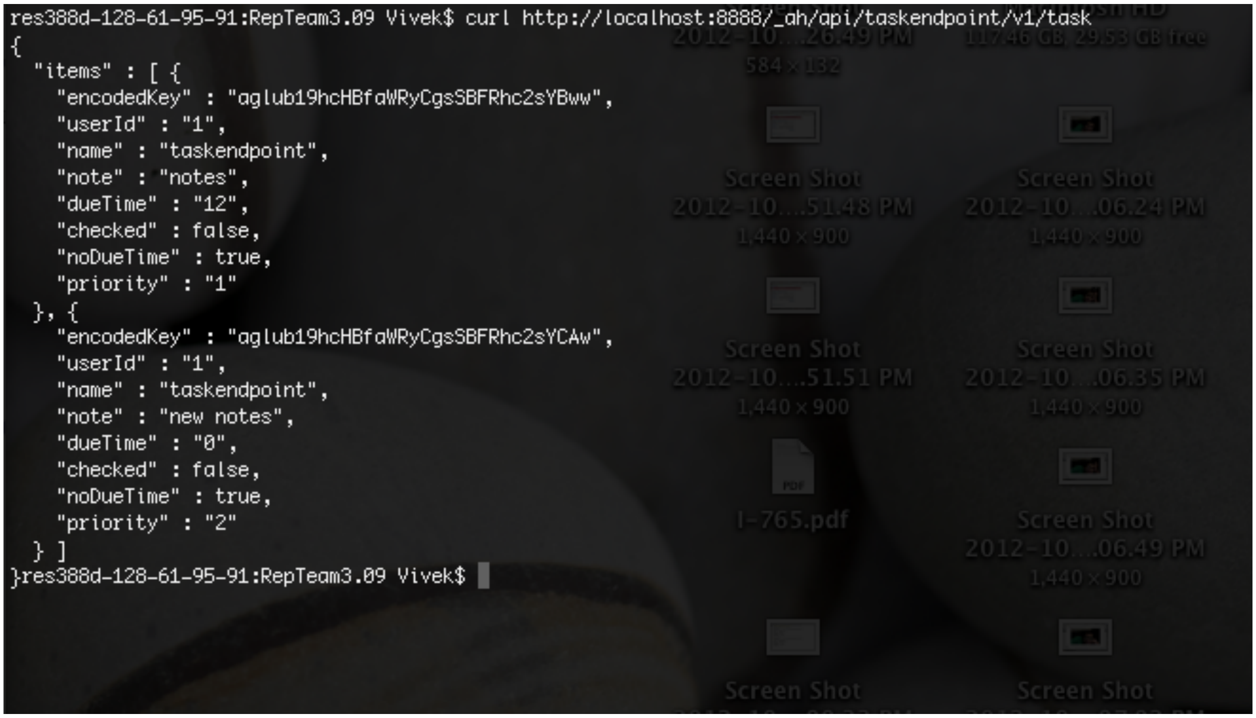

localhost:8888/_ah/api/taskendpoint/v1/task
Click to go forward, hold to see history

```
"items" : [ {
  "encodedKey" : "aglub19hcHBfaWRyCgsSBFRhc2sYBww",
  "userId" : "1",
  "name" : "taskendpoint",
  "note" : "notes",
  "dueTime" : "12",
  "checked" : false,
  "noDueTime" : true,
  "priority" : "1"
}, {
  "encodedKey" : "aglub19hcHBfaWRyCgsSBFRhc2sYCAw",
  "userId" : "1",
  "name" : "taskendpoint",
  "note" : "new notes",
  "dueTime" : "0",
  "checked" : false,
  "noDueTime" : true,
  "priority" : "2"
} ]
```

This image shows the two tasks I have added to the DataStore using the taskendpoint.list() API.

You can also see this through the command line using a simple curl command as shown below:

curl http://localhost:8888/_ah/api/taskendpoint/v1/task



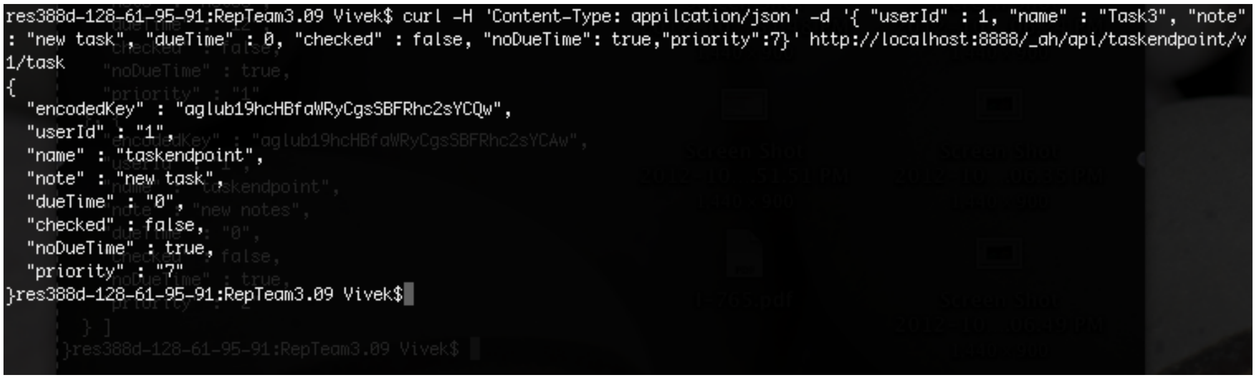
```
res388d-128-61-95-91:RepTeam3.09 Vivek$ curl http://localhost:8888/_ah/api/taskendpoint/v1/task
{
  "items" : [ {
    "encodedKey" : "aglub19hcHBfaWRyCgsSBFRhc2sYBww",
    "userId" : "1",
    "name" : "taskendpoint",
    "note" : "notes",
    "dueTime" : "12",
    "checked" : false,
    "noDueTime" : true,
    "priority" : "1"
  }, {
    "encodedKey" : "aglub19hcHBfaWRyCgsSBFRhc2sYCAw",
    "userId" : "1",
    "name" : "taskendpoint",
    "note" : "new notes",
    "dueTime" : "0",
    "checked" : false,
    "noDueTime" : true,
    "priority" : "2"
  } ]
}res388d-128-61-95-91:RepTeam3.09 Vivek$
```

Note if you are using Windows, you may have to install curl from here
Scroll down to Win32 - Generic:
http://curl.haxx.se/download.html

Notice that the same two tasks show up.

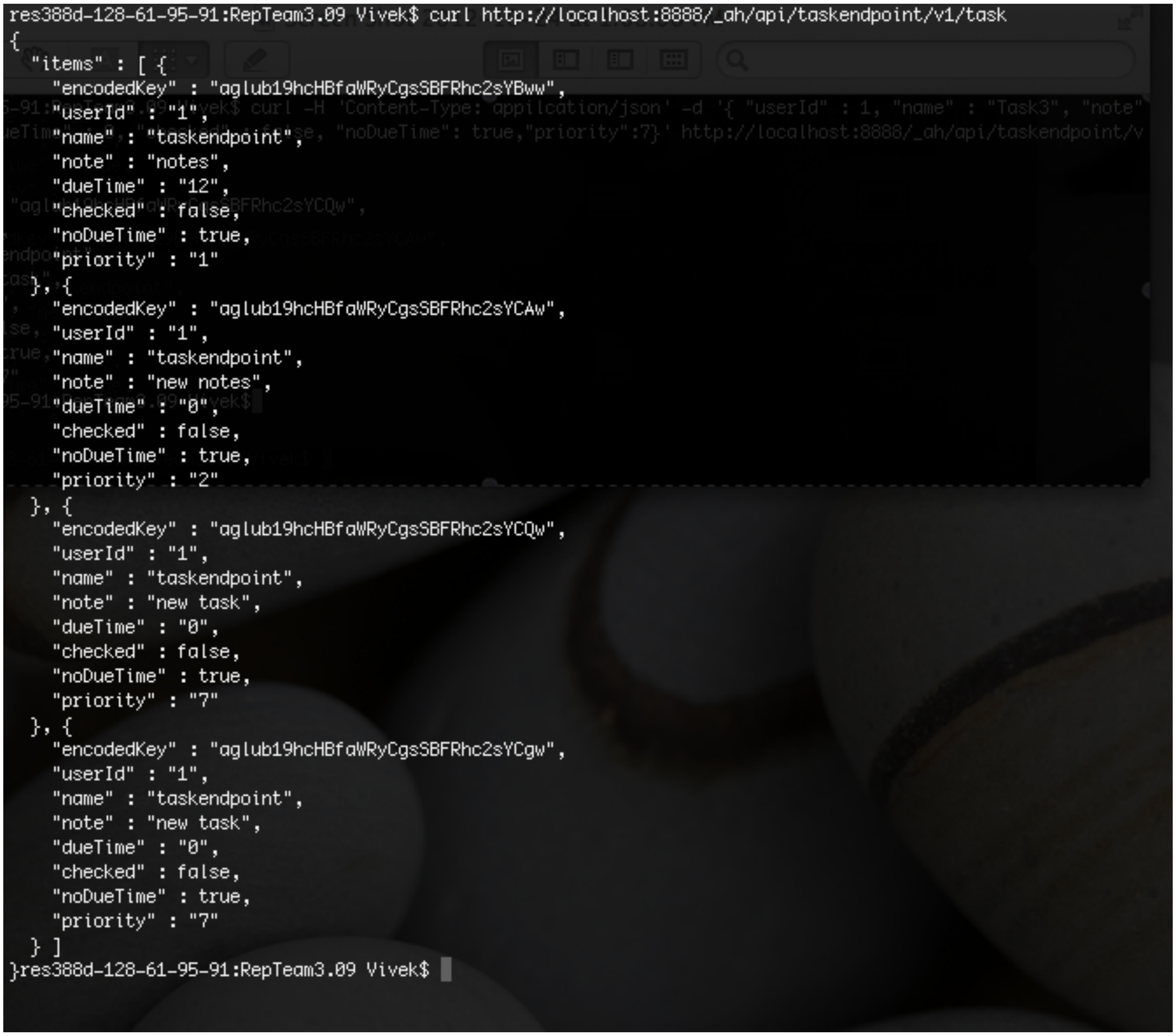
To add a new task, use the following curl command:

curl -H 'Content-Type: appilcation/json' -d '{ "userId" : 1, "name" : "Task3", "note": "new task", "dueTime" : 0, "checked" : false, "noDueTime": true,"priority":7}' http://localhost:8888/_ah/api/taskendpoint/v1/task



```
res388d-128-61-95-91:RepTeam3.09 Vivek$ curl -H 'Content-Type: appilcation/json' -d '{ "userId" : 1, "name" : "Task3", "note"
: "new task", "dueTime" : 0, "checked" : false, "noDueTime": true,"priority":7}' http://localhost:8888/_ah/api/taskendpoint/v
1/task
{
  "encodedKey" : "aglub19hcHBfaWRyCgsSBFRhc2sYCQw",
  "userId" : "1",
  "name" : "taskendpoint",
  "note" : "new task",
  "dueTime" : "0",
  "checked" : false,
  "noDueTime" : true,
  "priority" : "7"
}res388d-128-61-95-91:RepTeam3.09 Vivek$
```

As you can see, this creates a new task and adds it to the Datastore using the encodedKey that is generated. The api used to add a task is taskendpoint.insert(). Note that now two more tasks have been added, i.e 4 in total. Listing all the tasks again shows this:



```
res388d-128-61-95-91:RepTeam3.09 Vivek$ curl http://localhost:8888/_ah/api/taskendpoint/v1/task
{
  "items" : [ {
    "encodedKey" : "aglub19hcHBfaWRyCgsSBFRhc2sYBww",
    "userId" : "1",
    "name" : "taskendpoint",
    "note" : "notes",
    "dueTime" : "12",
    "checked" : false,
    "noDueTime" : true,
    "priority" : "1"
  }, {
    "encodedKey" : "aglub19hcHBfaWRyCgsSBFRhc2sYCAw",
    "userId" : "1",
    "name" : "taskendpoint",
    "note" : "new notes",
    "dueTime" : "0",
    "checked" : false,
    "noDueTime" : true,
    "priority" : "2"
  }, {
    "encodedKey" : "aglub19hcHBfaWRyCgsSBFRhc2sYCQw",
    "userId" : "1",
    "name" : "taskendpoint",
    "note" : "new task",
    "dueTime" : "0",
    "checked" : false,
    "noDueTime" : true,
    "priority" : "7"
  }, {
    "encodedKey" : "aglub19hcHBfaWRyCgsSBFRhc2sYCgw",
    "userId" : "1",
    "name" : "taskendpoint",
    "note" : "new task",
    "dueTime" : "0",
    "checked" : false,
    "noDueTime" : true,
    "priority" : "7"
  } ]
}res388d-128-61-95-91:RepTeam3.09 Vivek$
```

This now shows the four tasks in the Datastore.

In conclusion the Api created using Google Cloud endpoints works when used in the local host. However on deploying the project these Apis cannot be used since Google servers are not hosting these APIs till I obtain a trusted tester status. And for this reason I am unable to call these Apis from my android app.

I have already copied all the client side libraries for this API into my mobile application. So, once I obtain the trusted tester account, I can simply call these APIs, for example to call the insert from my mobile application, i will use the folllowing code in java.

```
Task task  = new Task;
task = taskendpoint.insert(task).execute();
String key = task.getEncodedKey(); // This can be stored to do other operations of the task
```

I also have the following Apis working:

```
public List<Task> listTask()
public Task getTask(@Named("id") String id)
public Task insertTask(Task task)
public Task updateTask(Task task)
public Task removeTask(@Named("id") String id)
```

These can be viewed here:
https://github.com/gatech/RepTeam3.09/blob/master/WTM/src/com/sdp/wtm/TaskEndpoint.java

Additional Information for the Trusted Tester Program

http://endpoints-trusted-tester.appspot.com/

https://docs.google.com/spreadsheet/viewform?pli=1&formkey=dDQ5QXJPTVFjSUhubl9WOC04TS1zSGc6MQ#gid=0

4. UI of Web Application

Creating basic UI for the web application, including a login and the tasks screen. Unfortunately, we couldn’t have the persistent layer, so there are just a few tasks hardcoded and users sign ups are not persistent. There’s one user hardcoded, “abcd” with password “dcbaaa”. In order to test it, launch ToDo Manager web project and navigate to:
http://127.0.0.1:8888/identification.html?gwt.codesvr=127.0.0.1:9997