

Google Cloud Messaging(GCM) Setup

Step 1:

- Go to <https://console.developers.google.com/project>
- Create a new project.
NOTE: Write down the Project number created, it will be used in the 'Android Application' section.
- Go to 'APIs and auth' -> Enable 'Google Cloud Messaging for Android'
- Go to 'Credentials' ->
'Create new Client ID'
'Select 'Installed Application.'
'Fill required project information(name)'
'Fill the android package name'
'Fill the SHA1 fingerprint key' - (Follow guide to get your project's SHA1 key
<http://hackerspace.lifehacker.com/the-easy-way-to-find-your-md5-and-sha1-in-eclipse-for-a-1505263073>)

Step 2:

Create a 'Public API key' from same 'Google Console' for 'Server Key'.

(Provide the EC2 Instance IP address, as explained in the 'Backend Server' Section)

NOTE: Write down the GCM API key which will be used as explained in Step 7 of 'Backend Server' section.

Backend Server

We have chosen Amazon Web Server services to host our backend server which responds to PHP requests and also, the database to store persistent information about our application users with active connection users.

Step 1: Launch an EC2 Instance

Follow the guide:

<http://www.comtechies.com/2012/12/how-to-launch-ubuntu-instance-on-amazon.html>

NOTE:

- There is an option to use the service for free for a year. But, note to create an instance which is eligible for 'free-tier' service.
- Remember to allow - TCP, FTP, SFTP, HTTP connections in the security group step.
- Safeguard your '.pem' key obtained during key-pair generation; will be used to communicate with your EC2 instance.
- Write down your Public IP information from the AWS management Panel.

Step 2: Connecting to the EC2 instance

Follow the guide:

<http://www.comtechies.com/2012/12/how-to-connect-amazon-ec2-ubuntu.html>

Or use 'ssh' from terminal

Format : `ssh -i myKey.pem ec2-user@IP`

Step 3: Setting up the EC2 instance as web server with ability to respond to PHP requests and communicate with MySQL database

Follow guide:

<http://www.nczonline.net/blog/2011/07/21/quick-and-dirty-spinning-up-a-new-ec2-web-server-in-five-minutes/>

From the section : 'Setting up your virtual machine'

Step 4: Creating an MySQL RDS instance

Follow guide:

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_CreateInstance.html

NOTE:

- Write down the public URL for your MySQL RDS instance from the AWS management panel.
(Will be of format 'abc.xyz.us-west-2.rds.amazonaws.com')

Step 5: Connecting to your MySQL RDS instance

Follow guide:

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ConnectToInstance.html

Or

From terminal:

`mysql -h abc.xyz.us-west-2.rds.amazonaws.com -P 3306 -u 'database username' -p`

Step 6: Initialize the database by issuing the following commands in the terminal after connecting to MySQL

```
CREATE DATABASE VBG;
```

```
USE VBG
```

```
CREATE TABLE users (  
id INT(10) NOT NULL AUTO_INCREMENT,  
username VARCHAR(50) NOT NULL,  
password TEXT NOT NULL,  
status VARCHAR(1) NOT NULL DEFAULT 'C',  
gcmid TEXT DEFAULT NULL,  
created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
PRIMARY KEY (id),  
UNIQUE KEY username (username)  
);
```

```
CREATE TABLE GuideClient (  
id INT(10) NOT NULL AUTO_INCREMENT,  
guidename VARCHAR(50) NOT NULL,  
guideip VARCHAR(50) DEFAULT NULL,  
clientname VARCHAR(50) NOT NULL,  
clientip VARCHAR(50) DEFAULT NULL,  
created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
PRIMARY KEY (id)  
);
```

Step 7: Edit Some Configuration files

In the dblInfo.php

```
$servername = " "; //Amazon RDS URL. [EX: abc.xyz.us-west-2.rds.amazonaws.com]  
$username = " "; //Amazon RDS username(NOTE: It is not amazon account Username) [EX: admin]  
$password = " "; //Amazon RDS password(NOTE: It is not amazon account Username) [EX: admin]  
$dbname = " "; //Amazon RDS database name. [EX: VBG]  
$portno = " "; //Amazon RDS port number. [Default: 3306]
```

In the GCM_API_key.php

```
$apiKey = " "; //Project API key obtained from Google API console for your application
```

Step 8: Connect to EC2 instance and copy files

From terminal:

```
sftp -o IdentityFile=myKey.pem -C ec2-user@IP  
put localfilename remotefilename
```

Step 9: SSH and move file to 'www' directory

```
sudo mv * /var/www/html/
```

Android Application:

1. Eclipse setup:

Software Used

Our project made use of Eclipse. The procedure to install the IDE is as follows:

1. Download Eclipse(3.4) from : <http://www.eclipse.org/downloads/>
2. Install Jdk6 preferably if you don't have it already installed:
<http://java.sun.com/javase/downloads/index.jsp>
3. Download the Android SDK for your eclipse version and OS from

http://developer.android.com/sdk/download.html?v=android-sdk_r04-windows.zip

4. Install the ADT Plugin for Eclipse

Android offers a custom plugin for the Eclipse IDE, called Android Development Tools (ADT) which is designed to give you a powerful, integrated environment in which to build Android applications. It extends the capabilities of Eclipse to let you quickly develop new Android projects, create an application user interface, add components based on the Android Framework API and debug using the Android SDK tools.

Now, you just need to modify your Eclipse preferences to point to the Android SDK directory:

2. Project Setup:

- Install the Android SDK package and make sure you install the following two packages:

- google_play_services
- appcompat_v7

- Now reference these two projects in our project: **VideoBasedGuidance**.

- Setting up a firebase server

Create a new user account in the firebase website <https://www.firebase.com/> and note down the server URL.

- Import the project **VideoBasedGuidance** into the eclipse workspace.
- Also import **google_play_services** and **appcompmt_v7** to your workspace. These are the dependencies on our project
- **src** directory contains the source code.
- Two packages of our interest are: *com.vbg.guide* and *com.fsaduk.vbg*

Also we should set up the backend server.

The IP address you got from step 1 of backend server setup should be used in the android source code as follows:

- Go to the package **com.fsaduk.vbg** and open the file **URL.Java**
- Change the PROJECT_NUMBER of your project
- Now ,change the IP address of the backend server and the firebase as instructed in the URL.java file.
- This finishes the setup of the android source code.
- Now plug in the phone using USB cable and right click on the project: Run as-> Android application.

- This will install the application on your android device.