Cloud Power your Android Apps: Quick and Easy

Google Cloud Platform Developer Roadshow - 2014
Agenda

1. Mobile applications need scalable back ends
2. Using Cloud Backends
3. Consuming Backends in an Android App
4. Extending your App to your wrist with Android Wear
Mobile Applications need Scalable Backends
Isolated from outside world

Cloud Backend

83%

63%
Make applications more engaging

Off-device data, multi-device interactions
Want to focus on your code

Don't worry about load balancer configurations, CDNs, scaling...
Google Cloud Endpoints

Build Powerful Backends with APIs - Easily
Use familiar tools

*Android Studio* + Java
Start effortlessly

Auto-generated, mobile-optimized client libraries for your API
Scale automatically

*Google App Engine’s auto-scaling and high-availability*
Cloud Endpoints Architecture

- Mobile Backend Running on Google App Engine Instances
- Client libraries for your backend
- Backend running on Google App Engine instances
Cloud Endpoints Architecture

Client libraries for your backend

Backend running on Google App Engine instances
@Api(name = "myApi", version = "v1")
public class MyEndpoint {

    /** A simple Endpoint that takes a name and says "Hi" back */
    @ApiMethod(name = "sayHi")
    public MyBean sayHi(String name) {
        MyBean response = new MyBean();
        response.setData("Hi, " + name);

        return response;
    }
}

MyEndpoint.java
import com.example.MyBean;

public class MyEndpoint {
    @Api(name = "myApi", version = "v1")
    public class MyEndpoint {

        /** A simple Endpoint that takes a name and says "Hi" back */
        @ApiMethod(name = "sayHi")
        public MyBean sayHi(String name) {
            MyBean response = new MyBean();
            response.setData("Hi, " + name);
            return response;
        }
    }
}
@Api(name = "myApi", version = "v1")
public class MyEndpoint {
  /** A simple Endpoint that takes a name and says "Hi" back */
  @ApiMethod(name = "sayHi")
  public MyBean sayHi(String name) {
    MyBean response = new MyBean();
    response.setData("Hi, " + name);

    return response;
  }
}
Cloud Endpoints Architecture
Cloud Endpoints Architecture

Client libraries for your backend

Backend running on Google App Engine instances
Demo
Consuming Backends in an Android App
Cloud Endpoints Architecture

Backend running on Google App Engine instances

Client libraries for your backend

iOS

Android

Chrome
Client libraries tailored for your backend
Don't worry about "plumbing"

Object serialization/deserialization, compression, SSL support; client library generation

Networking, battery life, information security
Scale unlimitedly as your needs grow

ree to get started, has full power of Google App Engine: autoscaling, high-availability, multiple storage options.
apply plugin: 'android'

android {
    compileSdkVersion 19
    buildToolsVersion "19.1.0"

    defaultConfig {
        applicationId "com.google.sample.walkshare.app"
        minSdkVersion 19
        targetSdkVersion 19
        versionCode 1
        versionName "1.0"
    }

    buildTypes {
        release
        runProguard false
        proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-
    }

    repositories {
        mavenCentral()
    }

    dependencies {
        flatDir{
            dir 'libs'
        }
    }
}
Extending your App for Wearables
Android Wear

Apps and Notifications on your Wrist
Cloud Endpoints Architecture

Client libraries for your backend

Backend running on Google App Engine instances
apply plugin: 'android'

android {
    compileSdkVersion 19
    buildToolsVersion "19.1.0"
    defaultConfig {
        applicationId "com.google.sample.walkshare.app"
        minSdkVersion 19
        targetSdkVersion 19
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release
        runProguard false
    }
}

repositories {
    mavenCentral()
    flatDir{
        dir 'libs'
    }
}

dependencies {
    filesTree(dir: 'libs', includes: ['*.jar'])
}
Recap

1. Mobile applications need scalable back ends
2. Using Cloud Backends
3. Consuming Backends in an Android App
4. Extending your App to your wrist with Android Wear
End