FREE eBook

LEARNING retrofit2

Free unaffiliated eBook created from **Stack Overflow contributors.**

#retrofit2

Table of Contents

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: retrofit2

It is an unofficial and free retrofit2 ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official retrofit2.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with retrofit2

Remarks

Overview :

Retrofit : Type-safe HTTP client for Android and Java by Square, Inc.

GitHub Repo : https://github.com/square/retrofit

It is one of the widely used networking library in Android. By its design it is very flexible in nature and offer wide range of plug and play feature like support for various Json Parser like GSON,Jackson,moshi , support for Rx-Java , etc

Versions

Version	Release Date
2.3.0	2017-05-13
2.2.0	2017-02-21
2.1.0	2016-06-15

Examples

Setup

What is retrofit?

The official Retrofit page describes itself as:

> A type-safe REST client for Android and Java.

This library makes downloading **JSON** or **XML** data from a web API fairly straightforward. Once the data is downloaded then it is parsed into a Plain Old Java Object (POJO) defined for each request using anyone of the adapter/parser listed here.

For Demo purpose we would be using GSON parser

Setup:

1. Add internet permission in manifest.xml:

```
<uses-permission android:name="android.permission.INTERNET" />
```

2. Add the following to your `build.gradle` file:

```
dependencies {
    compile 'com.squareup.retrofit2:retrofit:2.1.0'
    compile 'com.squareup.retrofit2:converter-gson:2.1.0'
}
```

3. Create proper POJO(Model) based on your **Json** response:

   If your json response is:

```
{
    "CategoryModel": {
        "debug": "on",
        "window": {
            "title": "Sample Konfabulator Widget",
            "name": "main_window",
            "width": 500,
            "height": 500
        },
        "image": {
            "src": "Images/Sun.png",
            "name": "sun1",
            "hOffset": 250,
            "vOffset": 250,
            "alignment": "center"
        },
        "text": {
            "data": "Click Here",
            "size": 36,
            "style": "bold",
            "name": "text1",
            "hOffset": 250,
            "vOffset": 100,
            "alignment": "center",
            "onMouseUp": "sun1.opacity = (sun1.opacity / 100) * 90;"
        }
    }
}
```

Then you can use website like JsonOnlineEditor or JsonView to format your json which will help to create your model Or use jsonschema2pojo to convert your Json to POJO using GSON annotations :

```
public class CategoryModel {

    @SerializedName("debug")
    private String debug;

    @SerializedName("window")
    private Window window;

    @SerializedName("image")
    private Image image;

    @SerializedName("text")
    private Text text;
}
```

4. Then we need an instance of Retrofit which acts as controller for all the request and response.

   Note : We prefer to create this controller as singleton which is very helpful if we want to set some additional property of the client .

```
public static final String BASE_URL = "http://test.com"

Retrofit retrofit = new Retrofit.Builder()
            .baseUrl(BASE_URL)
            .addConverterFactory(GsonConverterFactory.create())
            .build();
```

5. Next create **Interface** class where define all api calls with request,response type and request params for each call .(We need to create an interface for managing url calls like `GET`,`POST` ..etc.)

```
public interface IPlusService {
    @GET("/api/category")
    Call<CategoryModel> getAllCategory();
}
```

6. Create network/request client with retrofit instance :

```
IPlusService requestClient = retrofit.create(IPlusService.class);
```

7. Call your web-service in your `Fragment/Activity`:

```
requestClient.getAllCategory().enqueue(new Callback<ResponseBody>() {
    @Override
    public void onResponse(Call<CategoryModel> call, Response<CategoryModel> response) {
        // DO success handling
    }

    @Override
    public void onFailure(Call<CategoryModel> call, Throwable t) {
        // DO failure handling
    }
});
```

Good resources:

1. https://github.com/codepath/android_guides/wiki/Consuming-APIs-with-Retrofit

2. http://www.vogella.com/tutorials/Retrofit/article.html

3. http://www.androidhive.info/2016/05/android-working-with-retrofit-http-library/

4. https://futurestud.io/tutorials/retrofit-getting-started-and-android-client

Read Getting started with retrofit2 online: https://riptutorial.com/retrofit2/topic/6965/getting-started-with-retrofit2

Credits

S. No	Chapters	Contributors
1	Getting started with retrofit2	Amir, Code_Life, Community, spuente