ICP 10: RESTful Services, Retrofit

Joe Moon
Email: jmn5y@umsystem.edu
Github: https://github.com/joemoon-0/WebMobile-2022Spring/tree/main/2022Spring-Mobile/Mobile_ICP10

Nathan Cheney
Email: ncxn8@umsystem.edu
Github:

Introduction

RESTful APIs are used for applications to gain access to a greater set of information that it can then display to the user. For Android applications, the Retrofit class is used to create a type-safe HTTP client for Android by converting the HTTP API into a Java Interface. This method is used with the Github API to display basic user data.

Retrofit Implementation

The first step involved implementing Retrofit into the application. This was done by adding it to the dependencies list in the Gradle Scripts.

```
35      implementation 'com.squareup.retrofit2:retrofit:2.9.0'
36      implementation 'com.squareup.retrofit2:converter-gson:2.9.0'
```

build.gradle (Module: Mobile_ICP.app)

In addition, internet permission was enabled to give the app internet access.

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

AndroidManifest.xml

Interface Classes

In order to capture the information from the API, an interface is used. A **User** class is created that serves as the object for each request. It is basic, only capturing the username and id.

```
5   public class User {
6
7       private int id;
8
9       @SerializedName("login")
10      private String userName;
11
12      public int getId() { return id; }
15
16      public String getUserName() { return userName; }
19  }
```

User.java

In addition, an interface is used to retrieve the data from the API and store it within a List of User objects.

```java
public interface ApiCollections {

    @GET("users")
    Call<List<User>> getData();

}
```

ApiCollections.java

Within the MainActivity, a retrofit object is created and used to make the API call, convert it to JSON, and then make the data accessible through the interface.

```java
Retrofit retrofit = new Retrofit.Builder()
        .baseUrl("https://api.github.com/")
        .addConverterFactory(GsonConverterFactory.create())
        .build();

ApiCollections apiCollections = retrofit.create(ApiCollections.class);

Call<List<User>> usersCall = apiCollections.getData();
```

MainActivity.java

Using the **usersCall** object, a callback is made to display the data into a textView that is then displayed to the user. The output is shown below.