Mobile Application Development
Integration HTTP Android Client (Retrofit)

Waterford Institute of Technology

November 1, 2016

John Fitzgerald

HTTP client

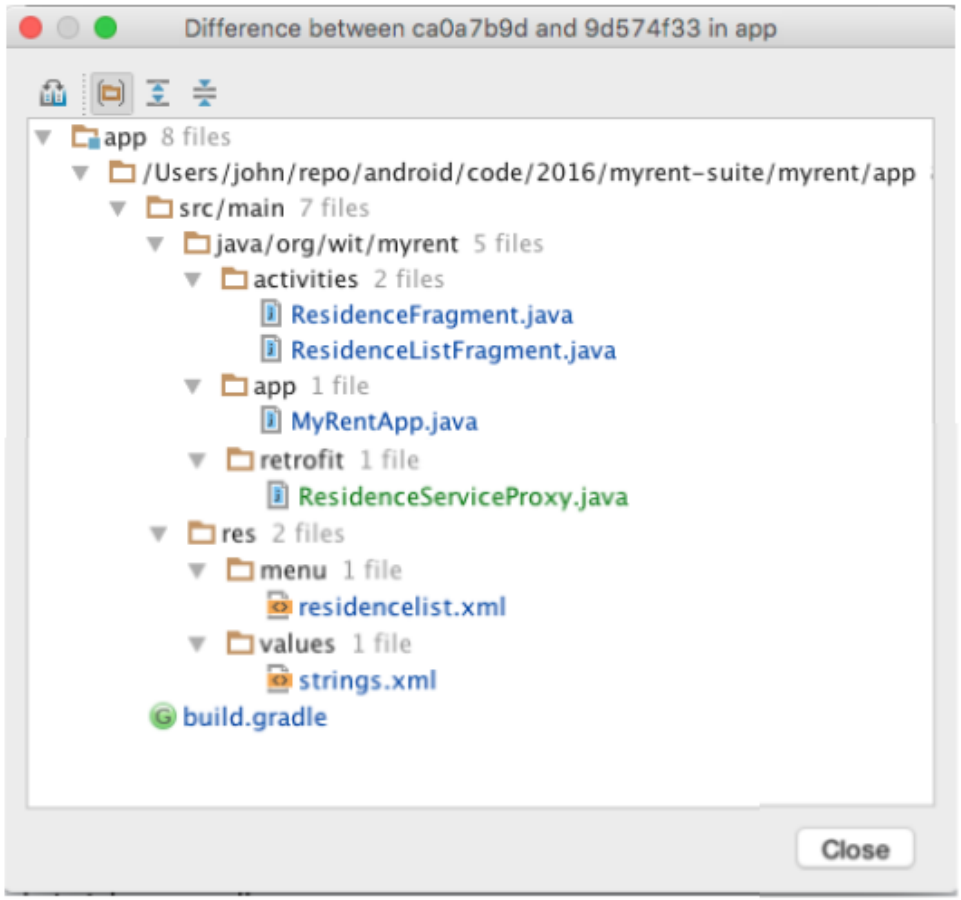
Learning objectives

- Refactor MyRent by integrating Retrofit HTTP client.
- SQLite acts as local cache.
- Model state in MyRent service updated by network calls.
- Local cache kept in sync through manual update.

Android MyRent Client

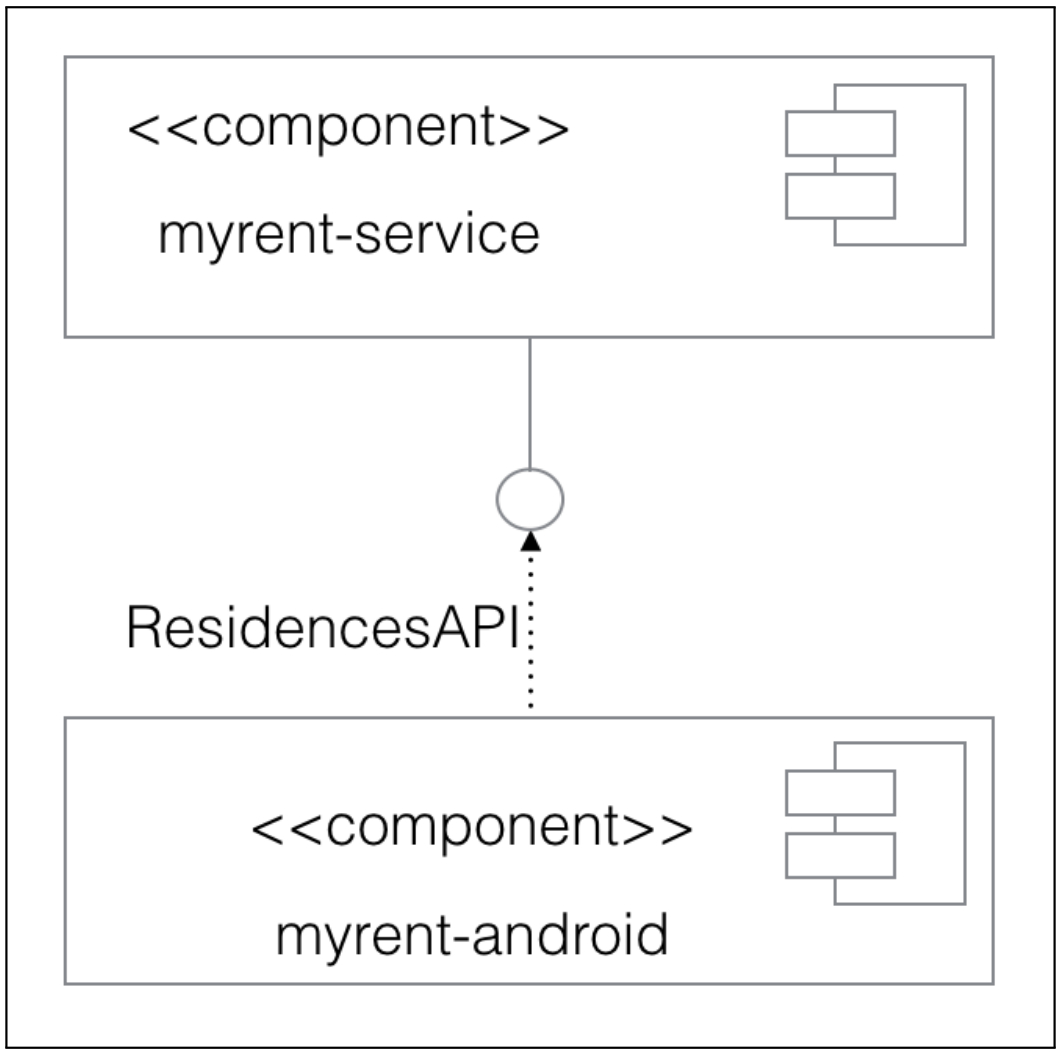
New and changed files

Structure at conclusion refactoring: proxy only new file.



MyRent service app

Android Client consumes service api

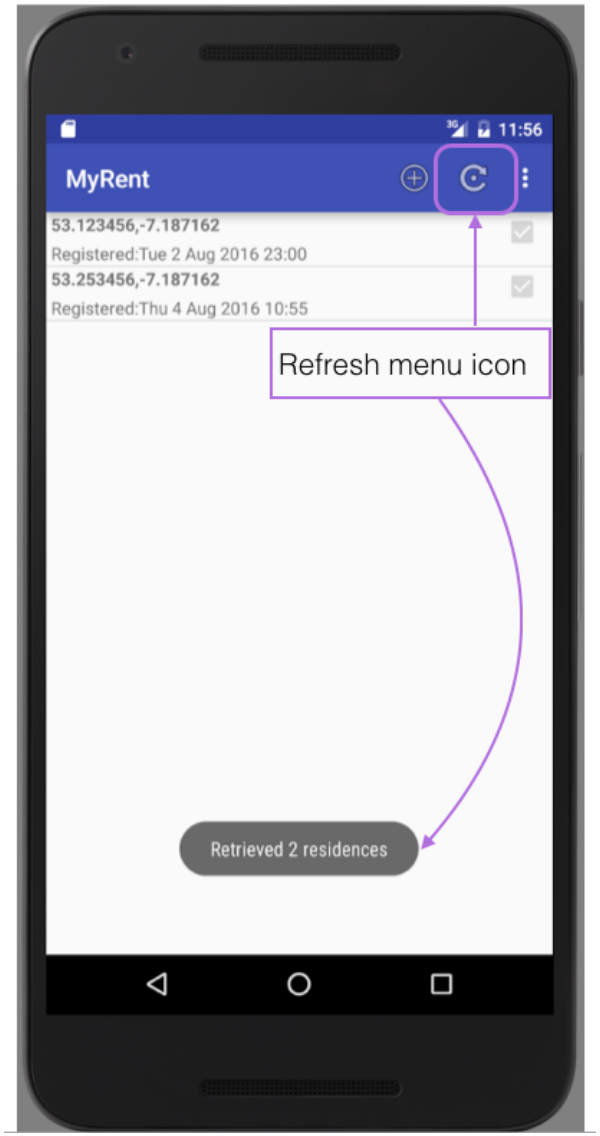


Android MyRent Client

Refresh Menu Item

```
<string name="refresh">Refresh
    Residence List</string>
```

```
<item
  android:id="@+id/action_refresh"
  android:icon=
   "@android:drawable/ic_menu_rotate"
  app:showAsAction="always"
  android:title="@string/refresh"/>
```



Android MyRent Client

ResidenceServiceProxy

```
public interface ResidenceServiceProxy
{
  @POST("/api/residence")
  Call<Residence> createResidence(@Body Residence residence);

  @DELETE("/api/residences/{id}")
  Call<String> deleteResidence(@Path("id") Long id);

  @POST("/api/residence/update")
  Call<Residence> updateResidence(@Body Residence residence);

  @GET("/api/residences")
  Call<List<Residence>> getResidences();

}
```

Android MyRent Client

ResidenceServiceProxy imports

```
import org.wit.myrent.models.Residence;

import java.util.List;

import retrofit.Call;
import retrofit.http.Body;
import retrofit.http.DELETE;
import retrofit.http.GET;
import retrofit.http.POST;
import retrofit.http.Path;
```

Android MyRent Client

MyRentApp refactoring - service urls

```
public String service_url = "http://10.0.2.2:9000"; // Android Emulator
//public String service_url = "http://10.0.3.2:9000"; // Genymotion
//public String service_url = "https://myrent-service-2016.herokuapp.com/";
```

Android MyRent Client

MyRentApp refactoring - Gson, Retrofit

```
public class MyRentApp extends Application {
  public ResidenceServiceProxy residenceService;

  @Override
  public void onCreate() {
    Gson gson = new GsonBuilder().create();
    Retrofit retrofit = new Retrofit.Builder()
        .baseUrl(service_url)
        .addConverterFactory(GsonConverterFactory.create(gson))
        .build();

    residenceService = retrofit.create(ResidenceServiceProxy.class);

  }
}
```

Android MyRent Client

MyRentApp refactoring - imports

```
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;

import org.wit.myrent.retrofit.ResidenceServiceProxy;

import retrofit.GsonConverterFactory;
import retrofit.Retrofit;
```

Android MyRent Client

ResidenceFragment

```
// Implement Callback<Residence> interface
public class ResidenceFragment extends Fragment implements . . .
    Callback<Residence>
{
 . . .
}
```

```
import retrofit.Call;
import retrofit.Callback;
import retrofit.Response;
import retrofit.Retrofit;
```

Android MyRent Client

ResidenceFragment

```
@Override
public void onPause() {
  super.onPause();
  updateResidence(residence); // <<------Initiates network call process
  portfolio.updateResidence(residence);
}
```

```
public void updateResidence(Residence res) {
  Call<Residence> call = app.residenceService.updateResidence(res);
  call.enqueue(this);
}
```

Android MyRent Client

ResidenceFragment - CallBack methods implementation

```
@Override
public void onResponse(Response<Residence> response, Retrofit retrofit) {
  Residence returnedResidence = response.body();
  if (returnedResidence != null) {
    Toast.makeText(getActivity(), "Residence updated successfully", Toast.
        LENGTH_SHORT).show();
  }
  else {
    Toast.makeText(getActivity(), "Network call failed",
    Toast.LENGTH_SHORT).show();

  }
}

@Override
public void onFailure(Throwable t) {
  Toast.makeText(getActivity(), Network call failed", Toast.
      LENGTH_SHORT).show();

}
```

Android MyRent Client

ResidenceListFragment Retrofit integration

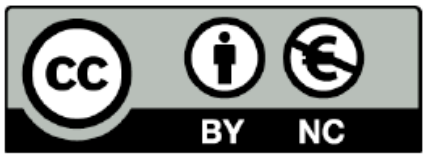
- A similar approach as ResidenceFragment.
- However, anonymous inner classes required.
- Delegate method to set listener accepts only one Callback.
- And required:
  - Callback<Residence>
  - Callback<List<Residence>>
  - Callback<String>
- This explained in detail in lab.

References

Retrofit from Square Open Source

1.Retrofit: HTTP Client for Android and Java

`https://square.github.io/retrofit/` [Accessed 2016-10-30]



Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning support unit