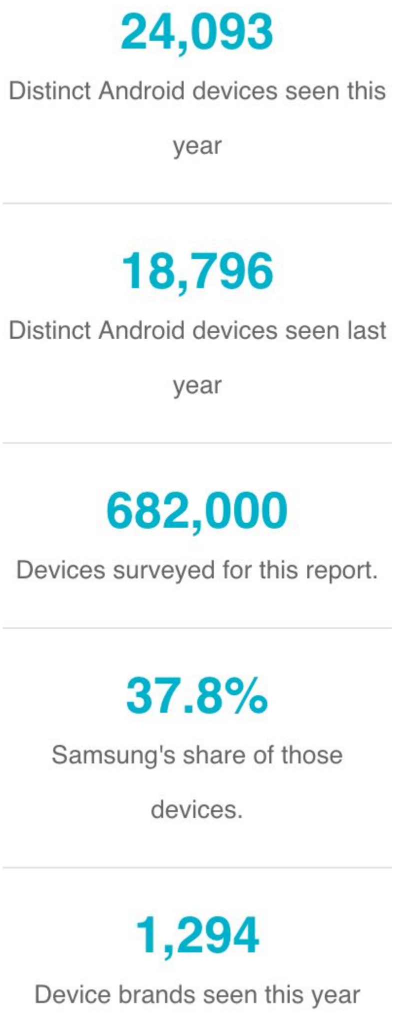
Android için Retrofit ile API’ leri Kullanmak

Ahmet Menteş
Android Developer
@ahmet_mentes
ahmetmentes@gmail.com

HIPO

Android Fragmentation Report by Open Signal 2015



24,093

Distinct Android devices seen this year

18,796

Distinct Android devices seen last year

682,000

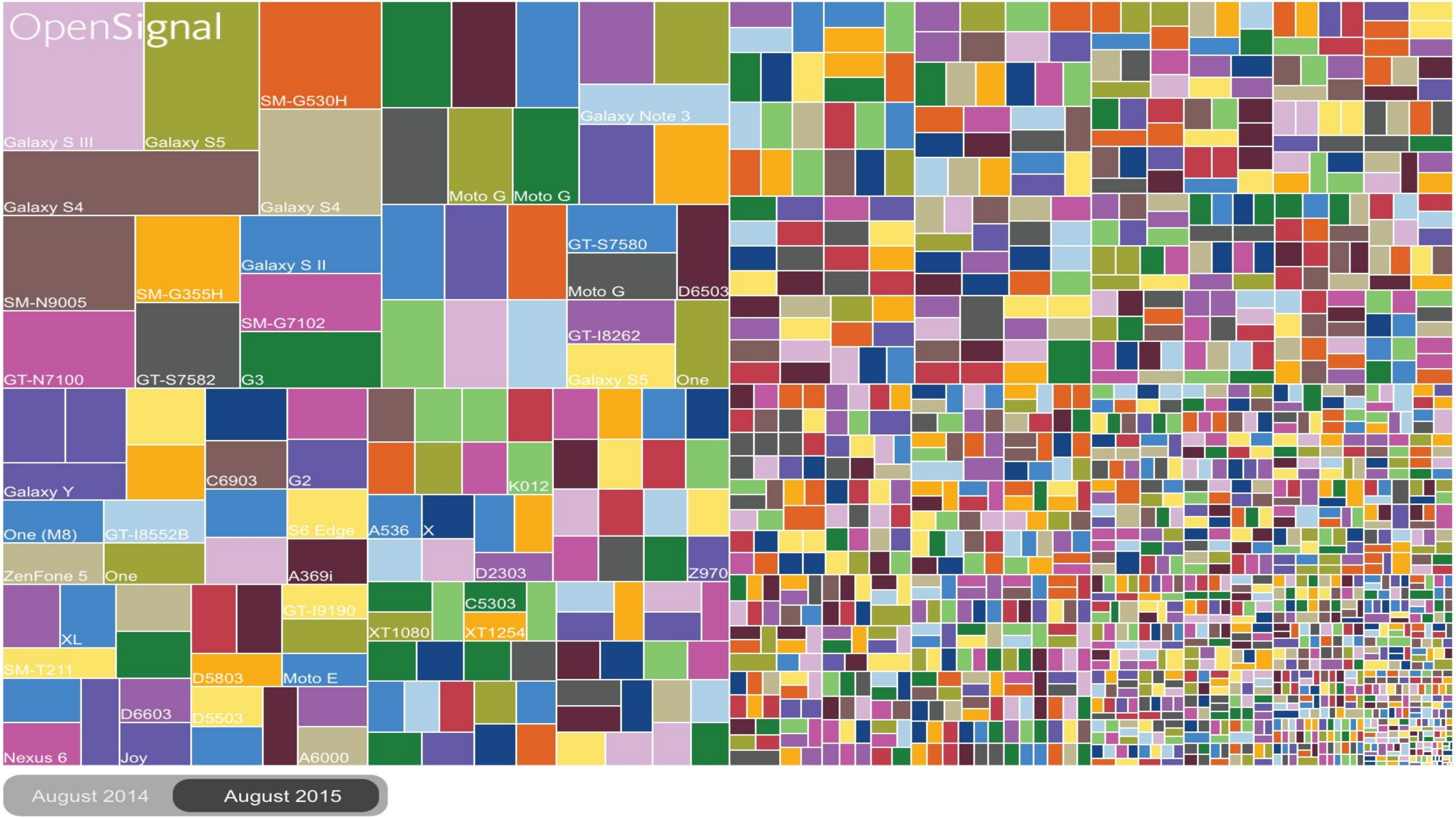
Devices surveyed for this report.

37.8%

Samsung's share of those devices.

1,294

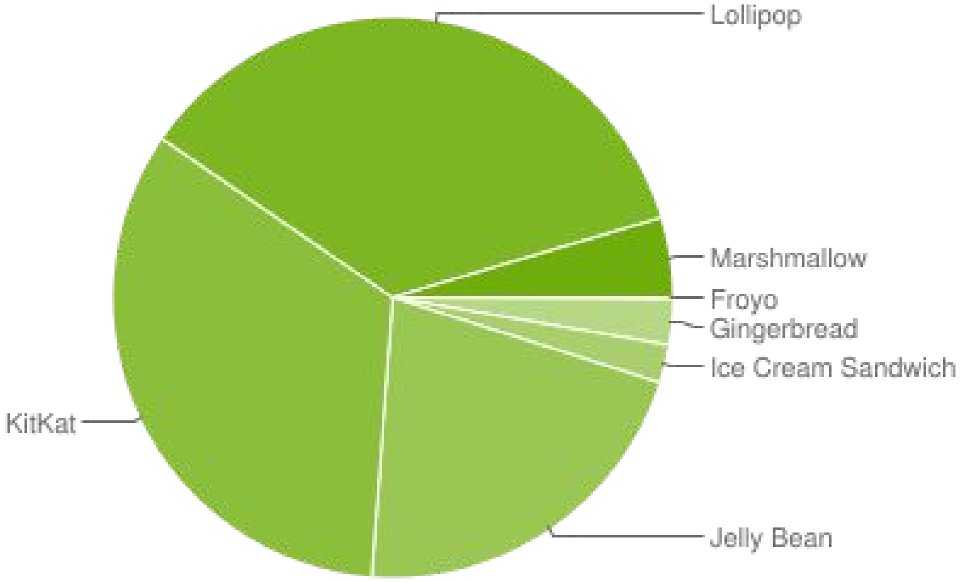
Device brands seen this year


OpenSignal
Galaxy S III
Galaxy S5
SM-G530H
Galaxy Note 3
Galaxy S4
Galaxy S4
Moto G
Moto G
SM-N9005
SM-G355H
Galaxy S II
GT-S7580
Moto G
D6503
SM-G7102
GT-I8262
GT-N7100
GT-S7582
G3
Galaxy S5
One
C6903
G2
K012
Galaxy Y
One (M8)
GT-I8552B
S6 Edge
A536
X
ZenFone 5
One
A369i
D2303
Z970
XL
GT-I9190
C5303
XT1080
XT1254
SM-T211
D5803
Moto E
D6603
D5503
Nexus 6
Joy
A6000
August 2014
August 2015

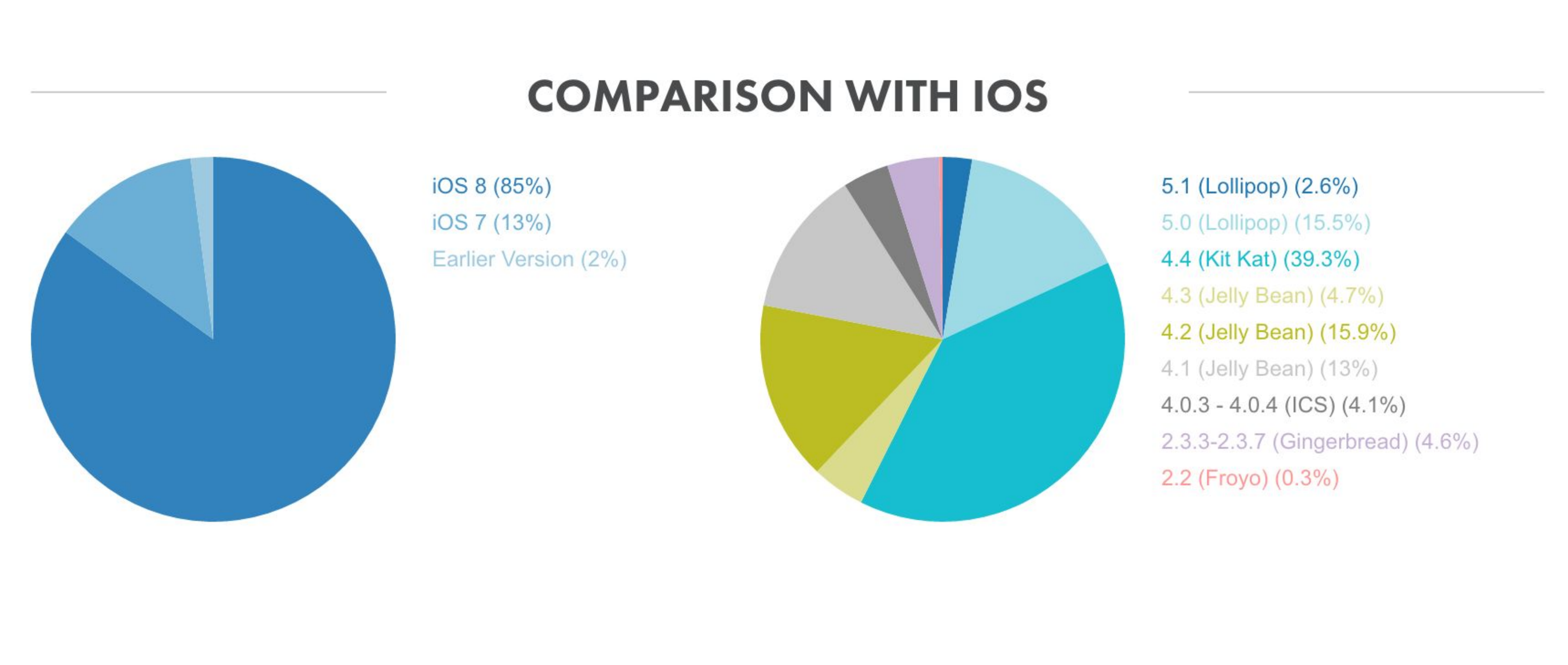
Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	2.6%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2.2%
4.1.x	Jelly Bean	16	7.8%
4.2.x		17	10.5%
4.3		18	3.0%
4.4	KitKat	19	33.4%
5.0	Lollipop	21	16.4%
5.1		22	19.4%
6.0	Marshmallow	23	4.6%

Data collected during a 7-day period ending on April 4, 2016.

Any versions with less than 0.1% distribution are not shown.



Android vs iOS



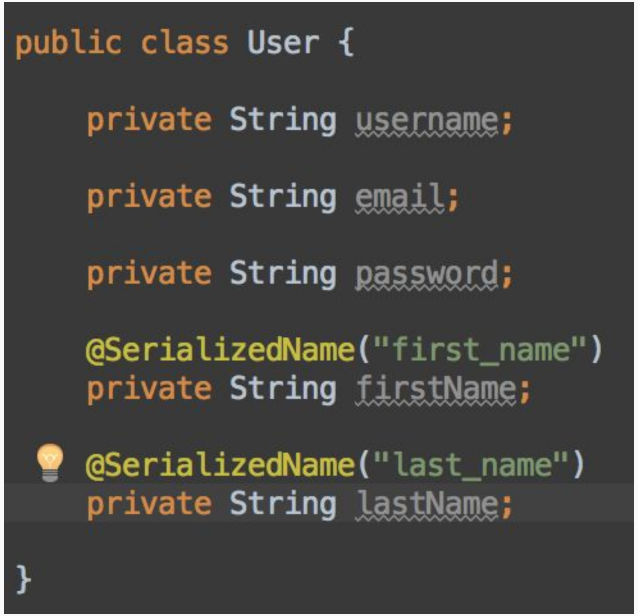
COMPARISON WITH IOS

Geçmişte

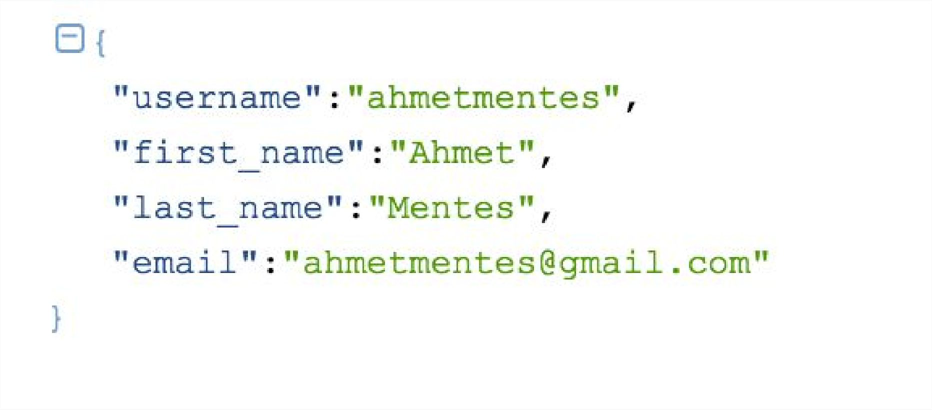
- Apache Http Client
- HttpUrlConnection
- OkHttp (Square)
  - OkHttp-engine (Android Kitkat 4.4)
- Volley (Networking Library by Google)

Gson

A Java serialization/deserialization library that can convert Java Objects into JSON and back



```java
public class User {

    private String username;

    private String email;

    private String password;

    @SerializedName("first_name")
    private String firstName;

    @SerializedName("last_name")
    private String lastName;

}
```



```json
{
    "username":"ahmetmentes",
    "first_name":"Ahmet",
    "last_name":"Mentes",
    "email":"ahmetmentes@gmail.com"
}
```

Retrofit

Retrofit turns your HTTP API into a Java interface.

A type-safe HTTP client for Android and Java

Retrofit 1

- Made open source on October 13th, 2010
- Originally a Bob Lee joint
- Took over / stole stewardship mid-2012
- Released 1.0 on May 13th, 2013
- 18 releases post-1.0 with tons of features / fixes

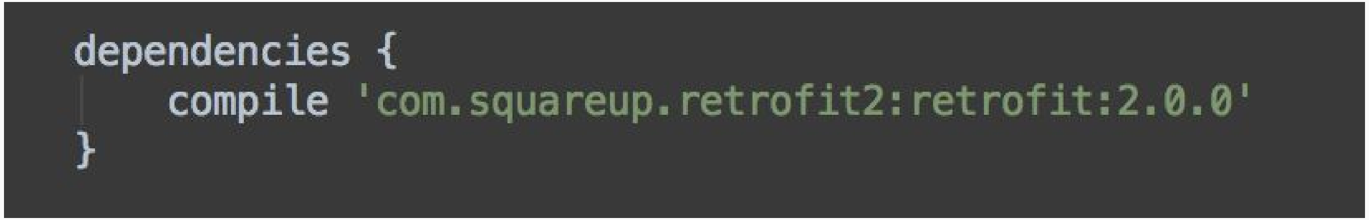
Retrofit 2

- Released on March 11th, 2016

Retrofit

Initialization

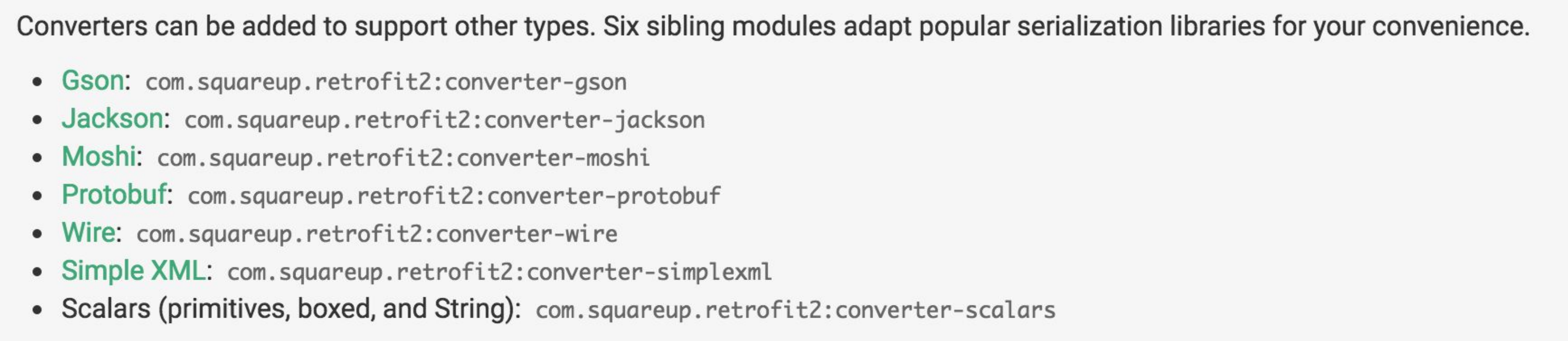
Add retrofit dependency to your app’s build.gradle file



```
dependencies {
    compile 'com.squareup.retrofit2:retrofit:2.0.0'
}
```

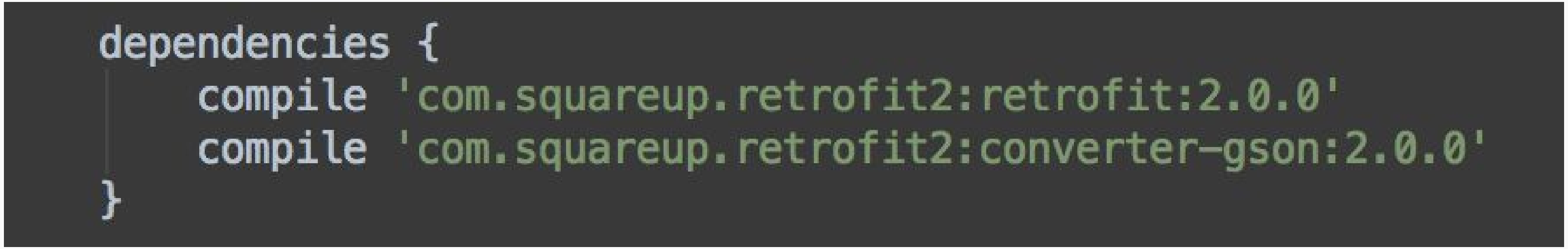
Retrofit

Initialization



Converters can be added to support other types. Six sibling modules adapt popular serialization libraries for your convenience.

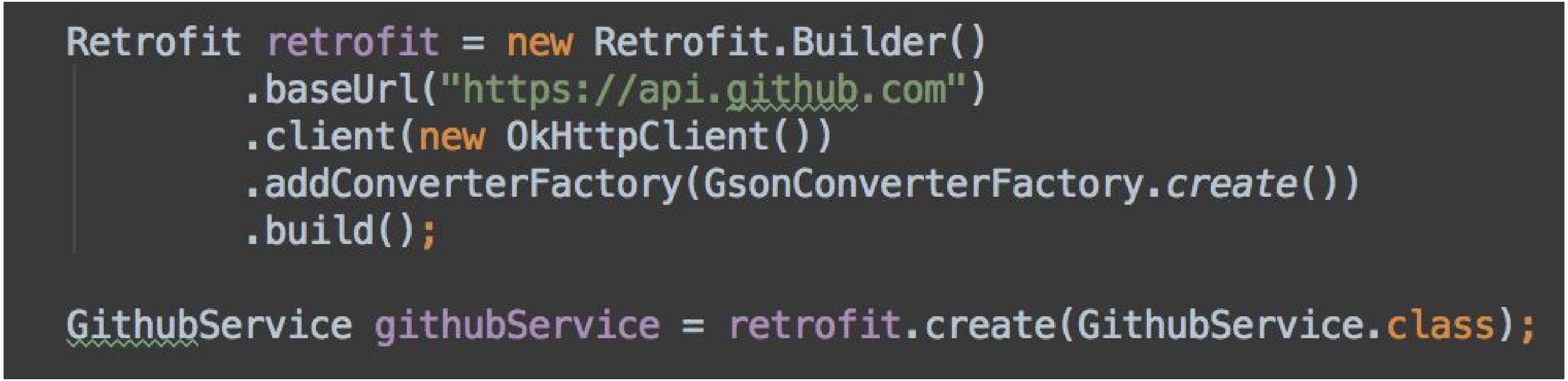
- Gson: `com.squareup.retrofit2:converter-gson`
- Jackson: `com.squareup.retrofit2:converter-jackson`
- Moshi: `com.squareup.retrofit2:converter-moshi`
- Protobuf: `com.squareup.retrofit2:converter-protobuf`
- Wire: `com.squareup.retrofit2:converter-wire`
- Simple XML: `com.squareup.retrofit2:converter-simplexml`
- Scalars (primitives, boxed, and String): `com.squareup.retrofit2:converter-scalars`



```
dependencies {
    compile 'com.squareup.retrofit2:retrofit:2.0.0'
    compile 'com.squareup.retrofit2:converter-gson:2.0.0'
}
```

Retrofit

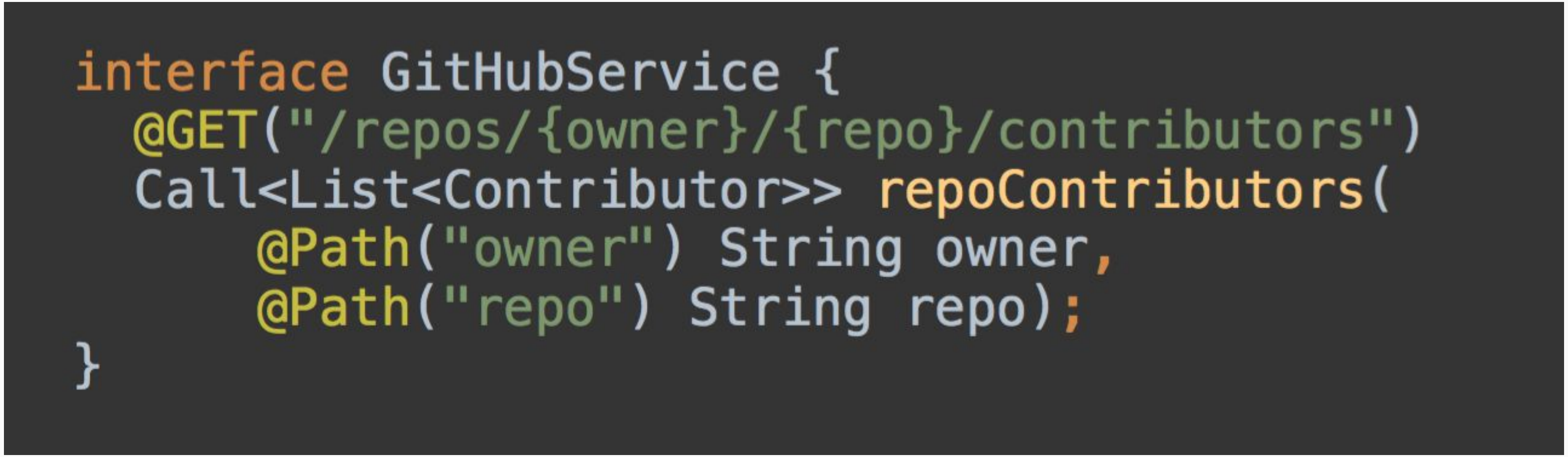
Initialization



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

GithubService githubService = retrofit.create(GithubService.class);
```

Retrofit

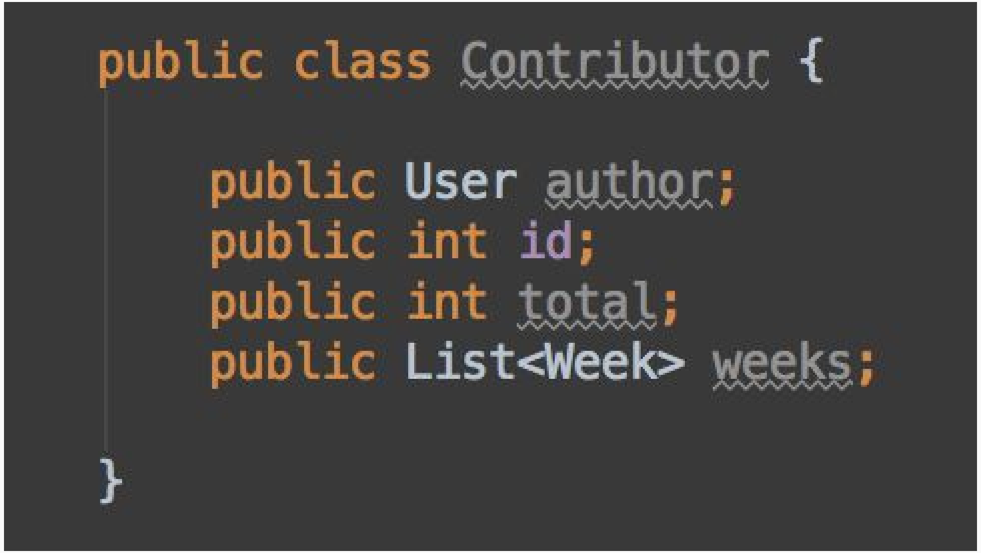
- Interface service declarations
- Method and parameter annotations customize request



```
interface GitHubService {
  @GET("/repos/{owner}/{repo}/contributors")
  Call<List<Contributor>> repoContributors(
      @Path("owner") String owner,
      @Path("repo") String repo);
}
```

Retrofit

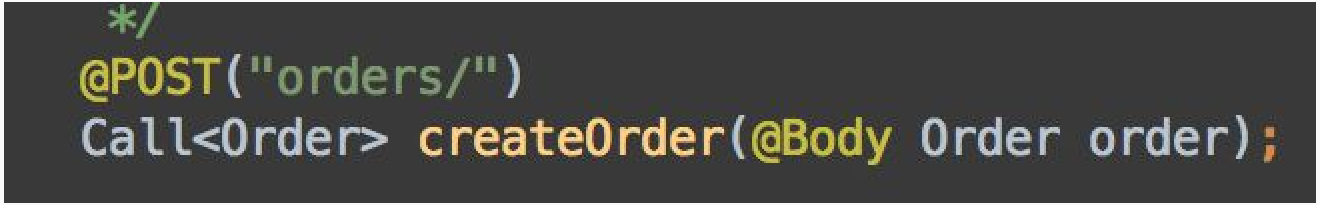
- Interface service declarations
- Method and parameter annotations customize request



```
public class Contributor {

    public User author;
    public int id;
    public int total;
    public List<Week> weeks;

}
```

Retrofit

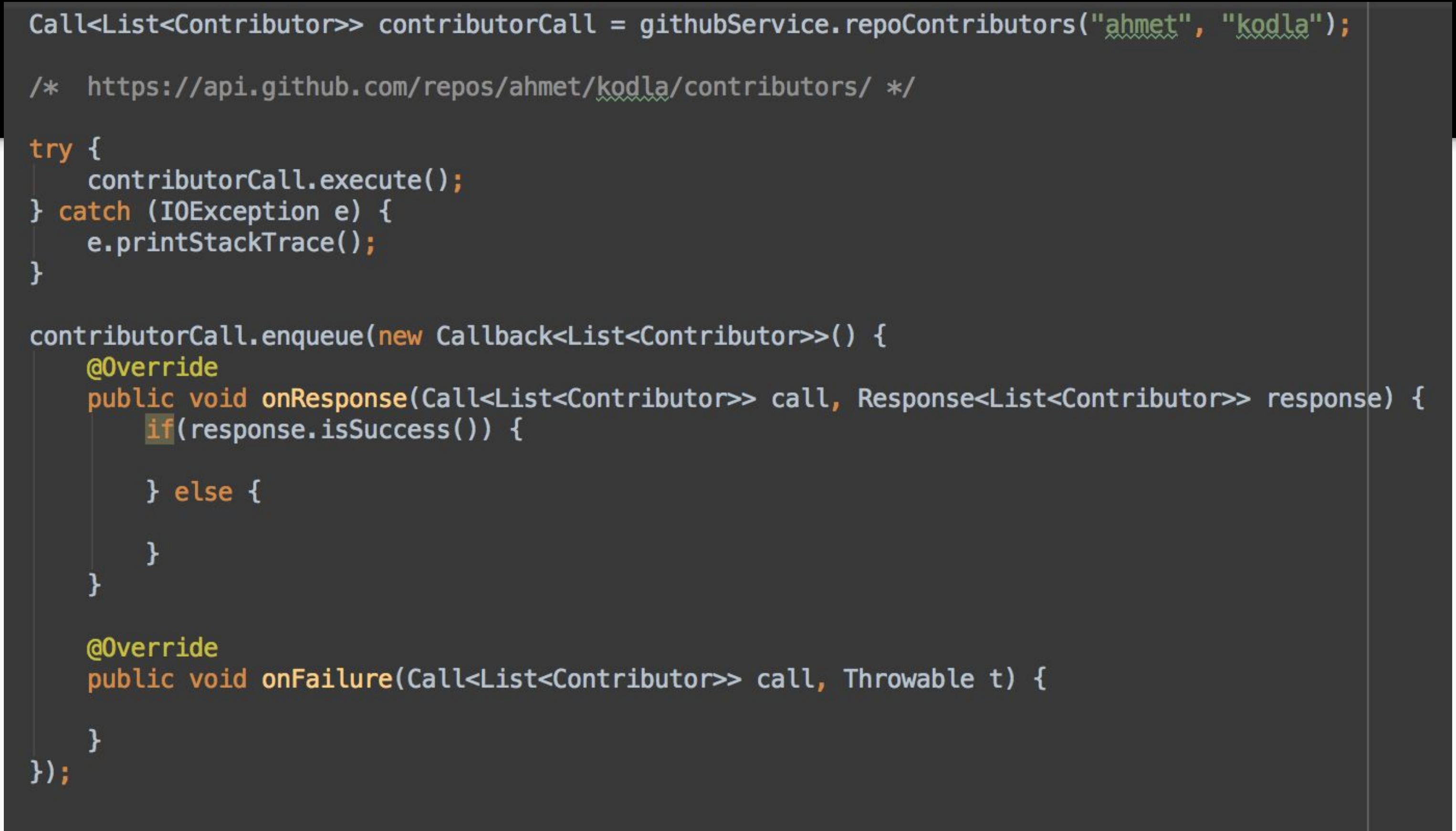
Method Types : GET, POST, PUT, DELETE and HEAD



```
 */
@POST("orders/")
Call<Order> createOrder(@Body Order order);
```

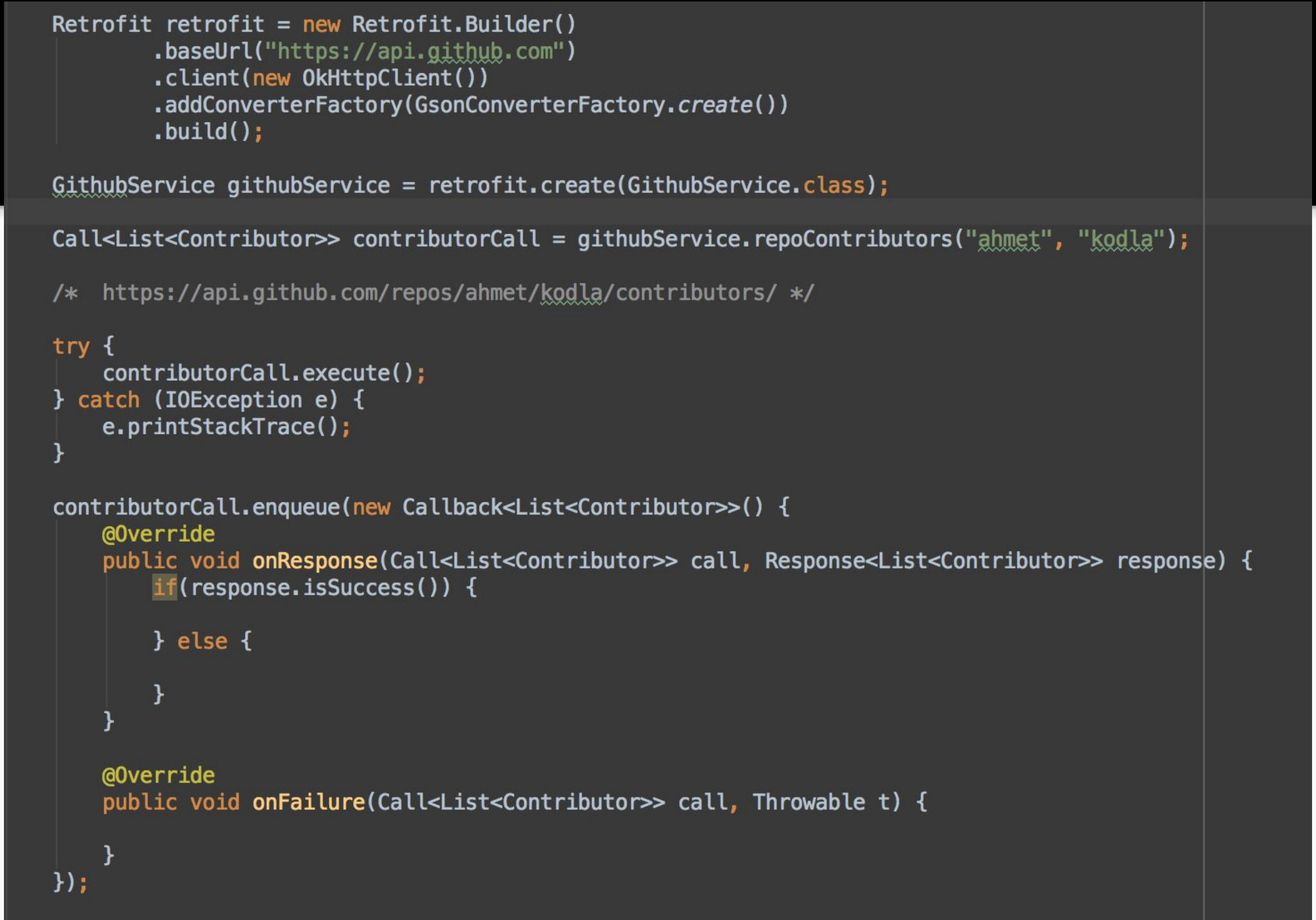
Retrofit

- Request, Response



```
Call<List<Contributor>> contributorCall = githubService.repoContributors("ahmet", "kodla");

/*  https://api.github.com/repos/ahmet/kodla/contributors/ */

try {
    contributorCall.execute();
} catch (IOException e) {
    e.printStackTrace();
}

contributorCall.enqueue(new Callback<List<Contributor>>() {
    @Override
    public void onResponse(Call<List<Contributor>> call, Response<List<Contributor>> response) {
        if(response.isSuccess()) {

        } else {

        }
    }

    @Override
    public void onFailure(Call<List<Contributor>> call, Throwable t) {

    }
});
```

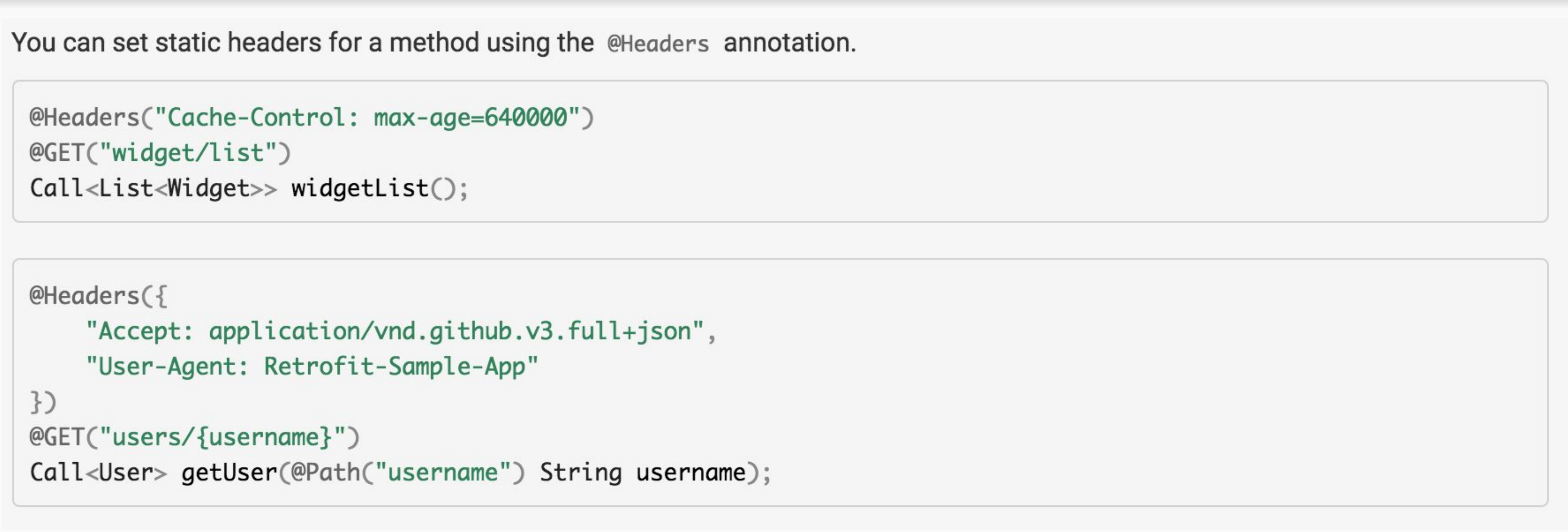
Retrofit



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

GithubService githubService = retrofit.create(GithubService.class);

Call<List<Contributor>> contributorCall = githubService.repoContributors("ahmet", "kodla");

/*  https://api.github.com/repos/ahmet/kodla/contributors/ */

try {
    contributorCall.execute();
} catch (IOException e) {
    e.printStackTrace();
}

contributorCall.enqueue(new Callback<List<Contributor>>() {
    @Override
    public void onResponse(Call<List<Contributor>> call, Response<List<Contributor>> response) {
        if(response.isSuccess()) {

        } else {

        }
    }

    @Override
    public void onFailure(Call<List<Contributor>> call, Throwable t) {

    }
});
```

Retrofit

- Header Manipulation



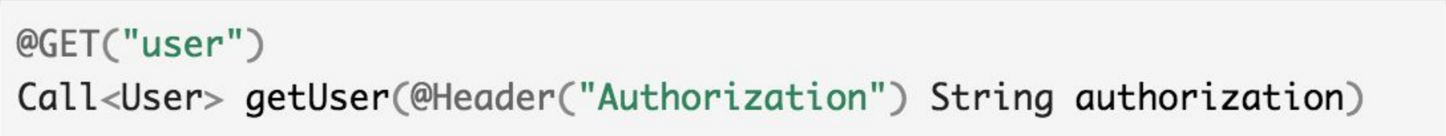
You can set static headers for a method using the `@Headers` annotation.

```
@Headers("Cache-Control: max-age=640000")
@GET("widget/list")
Call<List<Widget>> widgetList();
```

```
@Headers({
    "Accept: application/vnd.github.v3.full+json",
    "User-Agent: Retrofit-Sample-App"
})
@GET("users/{username}")
Call<User> getUser(@Path("username") String username);
```

Retrofit

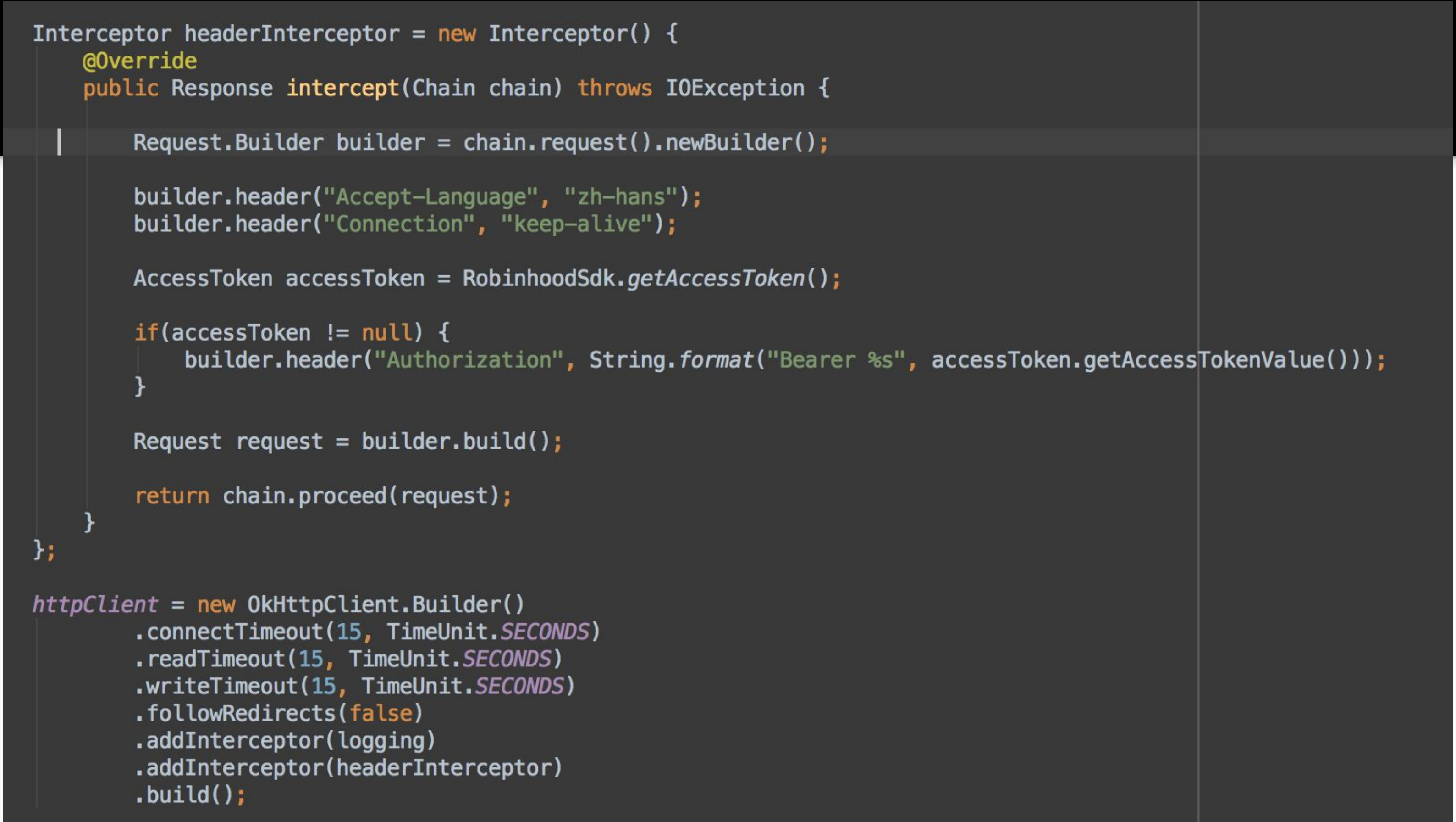
- Dynamic Headers



```
@GET("user")
Call<User> getUser(@Header("Authorization") String authorization)
```

Retrofit

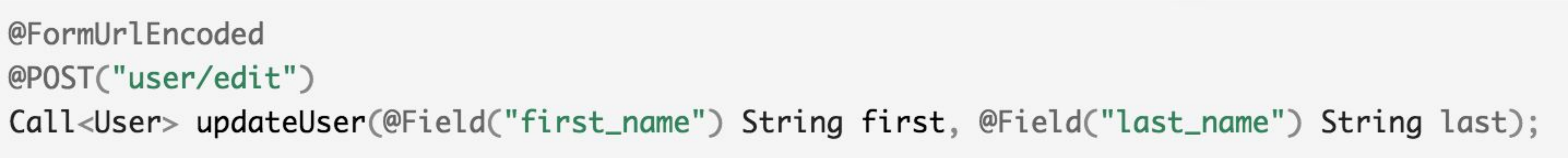
- Common Headers



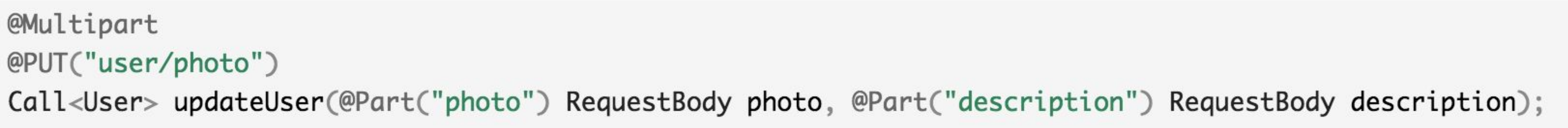
```
Interceptor headerInterceptor = new Interceptor() {
    @Override
    public Response intercept(Chain chain) throws IOException {

        Request.Builder builder = chain.request().newBuilder();

        builder.header("Accept-Language", "zh-hans");
        builder.header("Connection", "keep-alive");

        AccessToken accessToken = RobinhoodSdk.getAccessToken();

        if(accessToken != null) {
            builder.header("Authorization", String.format("Bearer %s", accessToken.getAccessTokenValue()));
        }

        Request request = builder.build();

        return chain.proceed(request);
    }
};

httpClient = new OkHttpClient.Builder()
        .connectTimeout(15, TimeUnit.SECONDS)
        .readTimeout(15, TimeUnit.SECONDS)
        .writeTimeout(15, TimeUnit.SECONDS)
        .followRedirects(false)
        .addInterceptor(logging)
        .addInterceptor(headerInterceptor)
        .build();
```

Retrofit

- FormUrlEncoded and Multipart



```
@FormUrlEncoded
@POST("user/edit")
Call<User> updateUser(@Field("first_name") String first, @Field("last_name") String last);
```



```
@Multipart
@PUT("user/photo")
Call<User> updateUser(@Part("photo") RequestBody photo, @Part("description") RequestBody description);
```

Questions?

- androidistanbul.slack.com -> androidistanbul.herokuapp.com
- Twitter: @anddevist
- Github: @androidistanbul
- http://www.meetup.com/Android-Istanbul-Meetup/