1. Add the following two libraries to our project.
    a. Retrofit -> For the network calls
    b. Gson -> For easy parsing of JSON data
2. We need to create the Model class (All the objects than we going to get from the retrofit response)
3. Created the API interface (Is when we make the @GET @POST @PUT @DELETE
    a. Here we need to add the BASE_URL (It's the http://... before the api call)
    b. Letter we make aout annotations @GET, @POST, @PUT, @DELETE with the name of the api inside the parentheses:
    @GET("marvel")
    Call<List<Hero>> getHeroes();
4. Now we need to make the API call
    a. We need to create a Retrofit object, it need the next component after new:
        i. Retrofit.Builder()
        ii. .baseUrl(Api.BASE_URL)
        iii. .addConverterFactory(GsonConverterFactory.create())
        iv. .build();
    b. We need to create the api interface, we'll get it from the instance of retrofit.create() and we'll send by parameter the api.class
    Api api = retrofit.create(Api.class);
5. Now we need to create the **Call** object, with this we set what we want to get, and we need to set it we the method from the api interface
    a. **Call<List<Hero>> call = api.getHeroes();**
    b. Finally we use the object call to enqueue the response, this interface will have two override onResponse and onFailure

```
//then finallly we are making the call using enqueue()
//it takes callback interface as an argument
//and callback is having two methods onRespnose() and onFailure
//if the request is successfull we will get the correct response and onResponse will be executed
//if there is some error we will get inside the onFailure() method
call.enqueue(new Callback<List<Hero>>() {
    @Override
    public void onResponse(Call<List<Hero>> call, Response<List<Hero>> response) {

        //In this point we got our hero list
        //thats damn easy right ;)
        List<Hero> heroList = response.body();

        //now we can do whatever we want with this list

    }

    @Override
    public void onFailure(Call<List<Hero>> call, Throwable t) {
        Toast.makeText(getApplicationContext(), t.getMessage(), Toast.LENGTH_SHORT).show();
    }
});
```

Don’t forget to add Internet permission in the Manifest :D