Retrofit

<dependency>

<groupId>com.squareup.retrofit2</groupId>

<artifactId>retrofit</artifactId>

<version>2.4.0</version>

</dependency>

<!-- https://mvnrepository.com/artifact/com.squareup.okhttp3/logging-interceptor -->

<dependency>

<groupId>com.squareup.okhttp3</groupId>

<artifactId>logging-interceptor</artifactId>

<version>3.11.0</version>

</dependency>

<!-- https://mvnrepository.com/artifact/com.squareup.retrofit2/converter-gson -->

<dependency>

<groupId>com.squareup.retrofit2</groupId>

<artifactId>converter-gson</artifactId>

<version>2.4.0</version>

</dependency>

<!-- https://mvnrepository.com/artifact/com.google.code.gson/gson -->

<dependency>

<groupId>com.google.code.gson</groupId>

<artifactId>gson</artifactId>

<version>2.8.5</version>

</dependency>

import okhttp3.OkHttpClient;

import okhttp3.logging.HttpLoggingInterceptor;

import retrofit2.Retrofit;

import retrofit2.converter.gson.GsonConverterFactory;

public class API {

    private static Retrofit retrofit = null;

    public static Retrofit getClient() {

        HttpLoggingInterceptor logging = new HttpLoggingInterceptor();

        logging.setLevel(HttpLoggingInterceptor.Level.BODY);

        OkHttpClient.Builder httpClient = new OkHttpClient.Builder();

        httpClient.addInterceptor(logging);

        if (retrofit != null) {

            retrofit = null;

        }

        retrofit = new Retrofit.Builder()

                .baseUrl("http://jsonbulut.com/json/")

                .addConverterFactory(GsonConverterFactory.create())

                .client(httpClient.build())

                .build();

        return retrofit;

    }

}

private static void getEmployeeById()

{

final String uri = "http://localhost:8080/springrestexample/employees/{id}";

Map<String, String> params = new HashMap<String, String>();

params.put("id", "1");

RestTemplate restTemplate = new RestTemplate();

EmployeeVO result = restTemplate.getForObject(uri, EmployeeVO.class, params);

System.out.println(result);

}