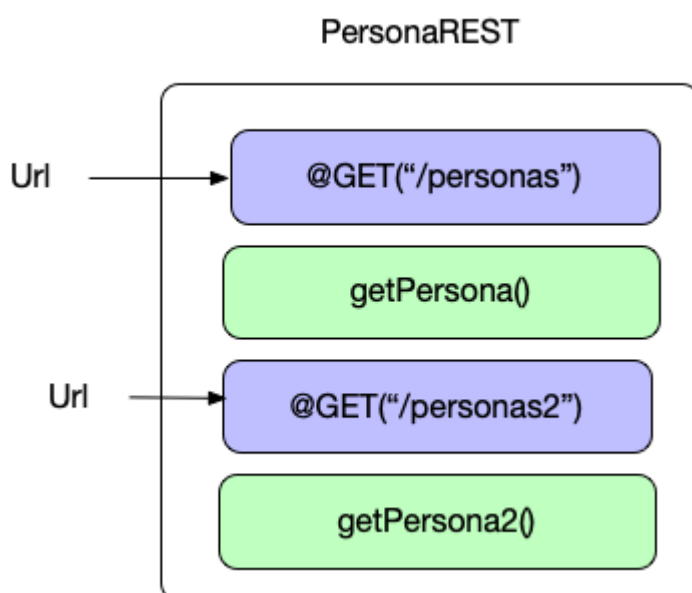


Retrofit es un cliente HTTP que nos permite realizar llamadas de forma sencilla a servicios REST. En muchos casos nos encontramos con que aplicaciones como Spring soportan RestTemplates o OpenFeign pero cuando salimos de Spring las cosas son más complejas y podemos necesitar una librería que haga las cosas relativamente sencillas. Retrofit es un ejemplo de este caso. Para ello lo primero siempre es configurar un proyecto Maven con su pom.xml.

```
<dependency>
    <groupId>com.squareup.retrofit2</groupId>
    <artifactId>retrofit</artifactId>
    <version>2.11.0</version>
</dependency>
```

Una vez añadida la dependencia lo único que tenemos que construir es un interface que tenga las propias anotaciones de Retrofit y permita hacer peticiones http.



```
import retrofit2.Call;
import retrofit2.http.GET;
```

```
public interface PersonaRest {  
  
    @GET("/personas")  
    Call<Persona> getPersona();  
  
    @GET("/personas2")  
    Call<Persona> getPersona2();  
}
```

Junto al interface necesitaremos una clase que define la información que queremos publicar:

```
public class Persona {  
  
    private String nombre;  
    private String apellidos;  
    private int edad;  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public void setNombre(String nombre) {  
        this.nombre = nombre;  
    }  
  
    public String getApellidos() {  
        return apellidos;  
    }  
  
    public void setApellidos(String apellidos) {  
        this.apellidos = apellidos;  
    }  
}
```

```

    public int getEdad() {
        return edad;
    }

    public void setEdad(int edad) {
        this.edad = edad;
    }

    public Persona(String nombre, String apellidos, int edad) {
        this.nombre = nombre;
        this.apellidos = apellidos;
        this.edad = edad;
    }

    public Persona() {
    }

}

```

Como podemos observar es muy sencillo cada anotación @GET se encarga de una url existen anotaciones para cada uno de los verbos habituales. Una vez tenemos el interface con Retrofit es muy sencillo realizar la llamada http.

```

import retrofit2.Call;
import retrofit2.Response;
import retrofit2.Retrofit;
import retrofit2.converter.jackson.JacksonConverterFactory;

import java.io.IOException;
import java.time.LocalDateTime;

```

```

public class Principal {
    public static void main(String[] args) throws IOException {

        Retrofit retrofit = new Retrofit.Builder()
            .baseUrl()
            .addConverterFactory(JacksonConverterFactory.create())
            .build();
        PersonaRest servicio= retrofit.create(PersonaRest.class);
        Call<Persona> personaLLamada=servicio.getPersona();
        Response<Persona> respuesta= personaLLamada.execute();
        Persona persona= respuesta.body();
        System.out.println(persona.getNombre());

    }
}

```

En este caso tenemos una url base “localhost:8080” y lo que hacemos es usar el interface PersonaRest para registrar las urls que puedan ser válidas. Realizada esta operación el siguiente paso es usar Retrofit y crear un servicio sobre el cual realizar la llamada a través del método getPersona().

Este método se encarga de obtener un objeto Call<Persona> que será el que nos abstraerá de la “llamada” REST . Es momento de invocar al método execute que invocará el servicio y nos traerá los datos que queremos mostrar en la consola. El resultado será :

pepe

Retrofit es una librería práctica cuando nos encontramos en entornos que no soportan de forma adecuada Spring Boot.

- [REST HTTP return codes y sus curiosidades](#)

- [Arquitecturas REST y sus niveles](#)
- [Curso Spring REST](#)
- [Utilizando Servlets 4 y HTTP 2 para peticiones push](#)
- [HTTP Response Status Codes y su uso](#)