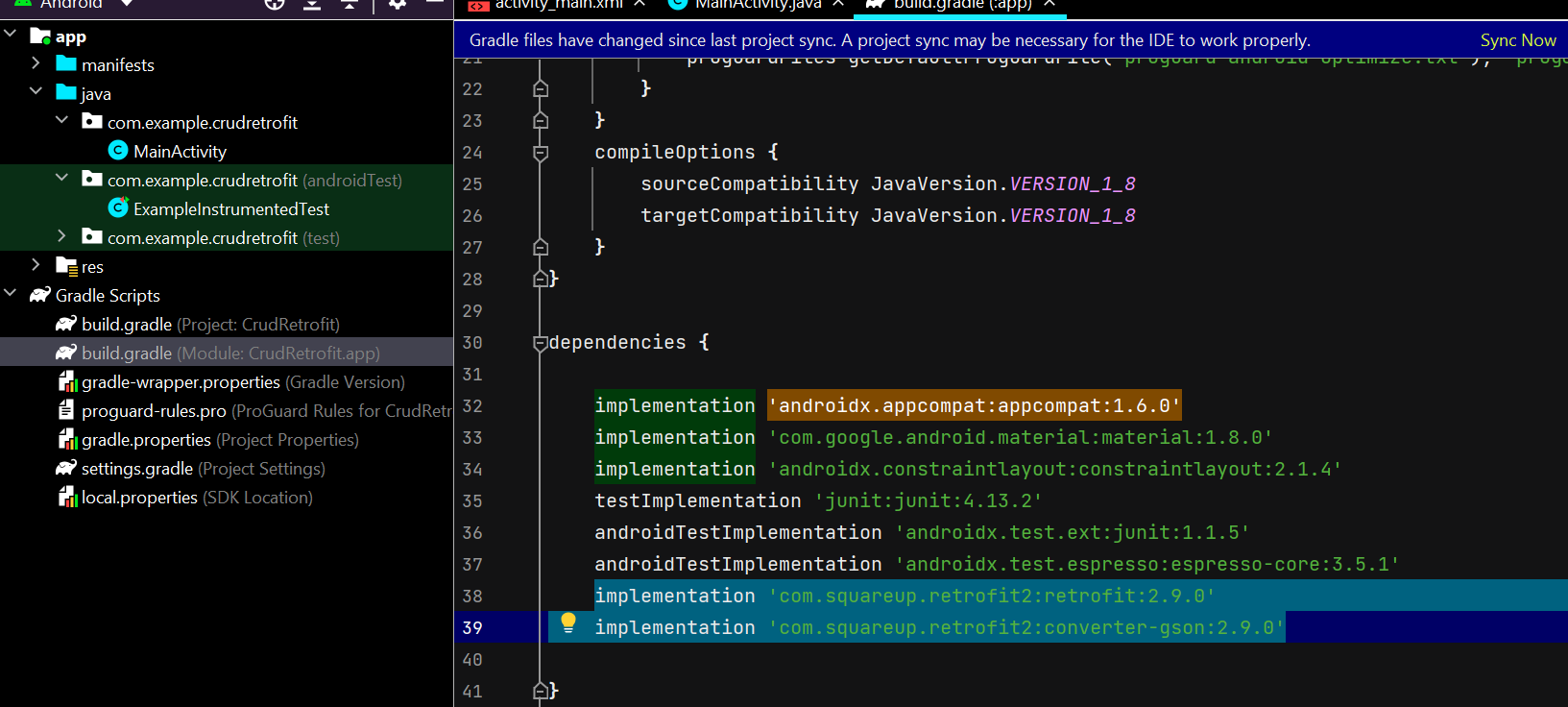
Contenido

1. Creación de proyecto backend con spring boot y postgress
2. Creación de proyecto Android
3. Agregar dependencias en el archivo gradle.build
4. Agregar nuevos paquetes
5. Crear la clase modelo
6. Crear interfaces con la implementación de los diferentes verbos
7. Agregar dependencias en el archivo gralde.build



La dependencia de retrofit2 sirve para construir cliente rest con Android y la librería gson sirve para leer el json devuelto por el servicio backend que para nuestro ejemplo es un servicio java en spring boot, luego de agregar las dependencias se debe oprimir el lin de sync que se encuentra en la parte superior derecha de la pantalla

4, Agregar nuevos paqueters

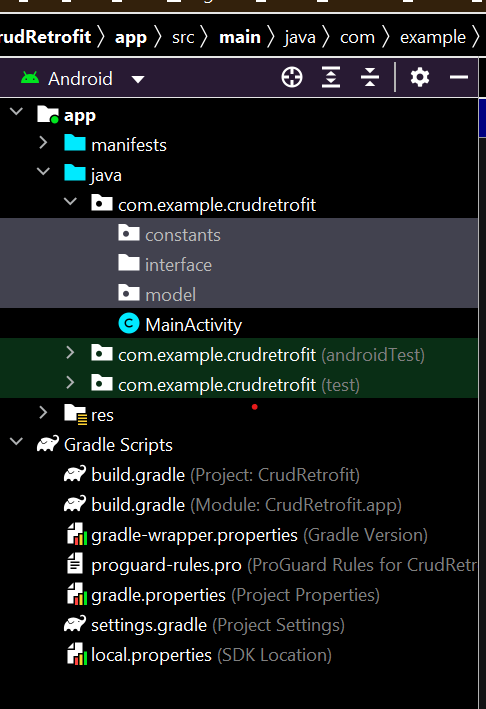
Se deben agregar nuevos paquetes al proyecto , lo cual es necesario para construir un proyecto con los mejores estándares de la industria basado en los principios de responsabilidad única, open closed , alta cohesión , bajo acoplamiento ,entre otros

Para este proyecto se crearán los paquetes:

Model: entidades de la base de datos y objetos POJOS

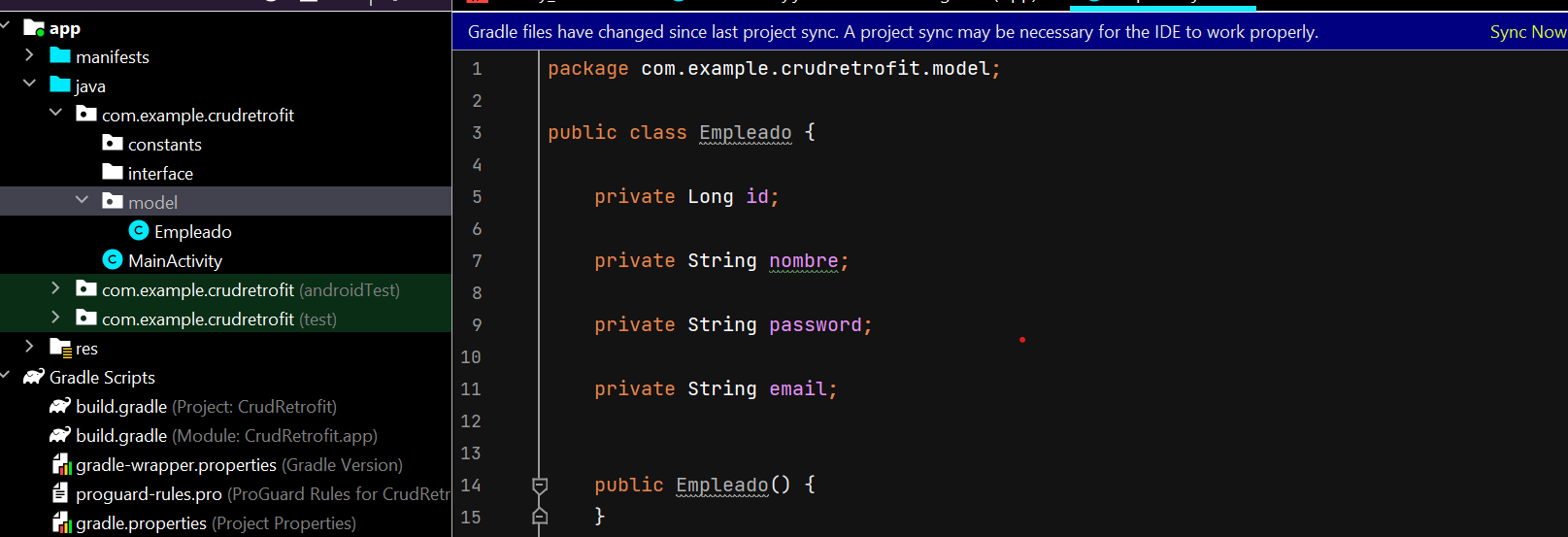
Constant: para gestionar las constantes dentro del proyecto

Interface: capa de seguridad y abstracción de complejidad de las capas mas internas

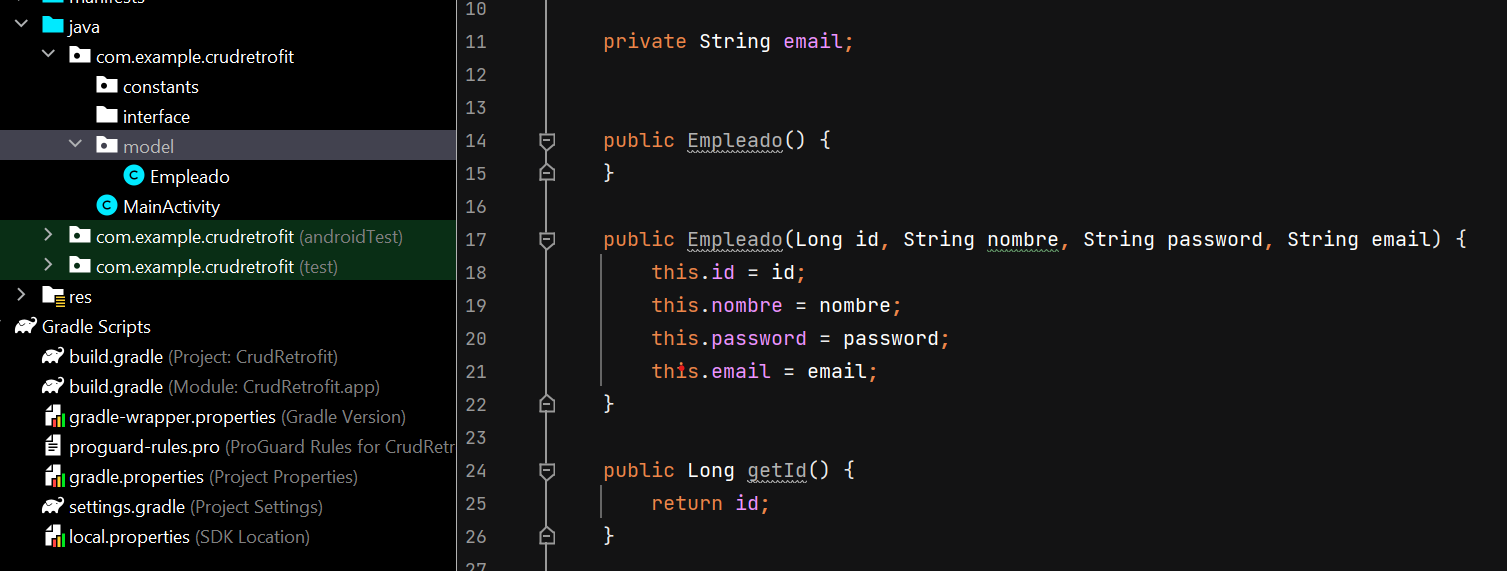


5. Crear la clase modelo

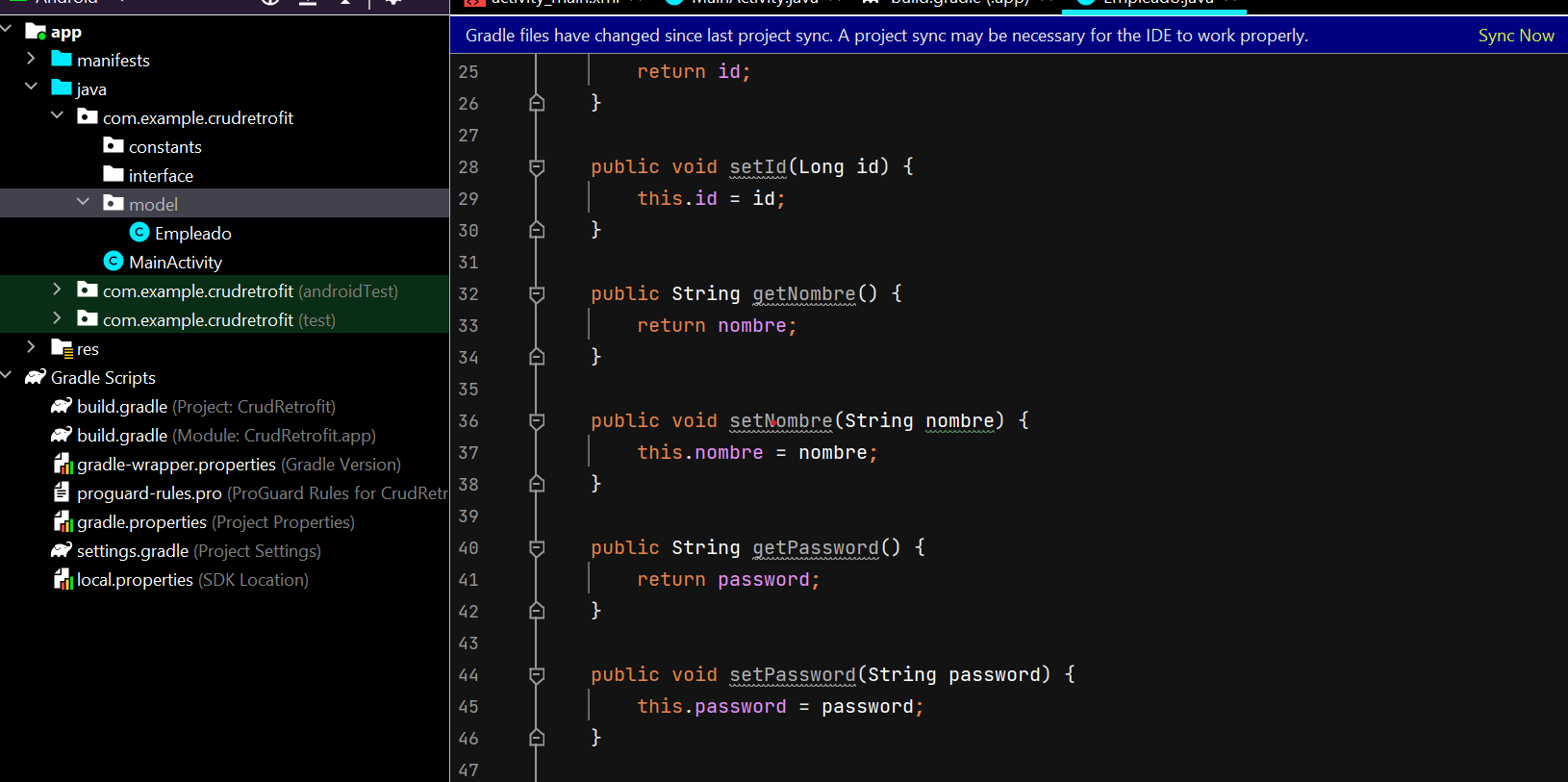
Dentro del paquete modelo , se debe crear la clase Empleado , que finalmente es la clase que hemos usado en el backend y que desde postman enviamos al servicio para que nos respondiera al verbo http invocado, en este caso debe llamarse igual y sus atributos tambié



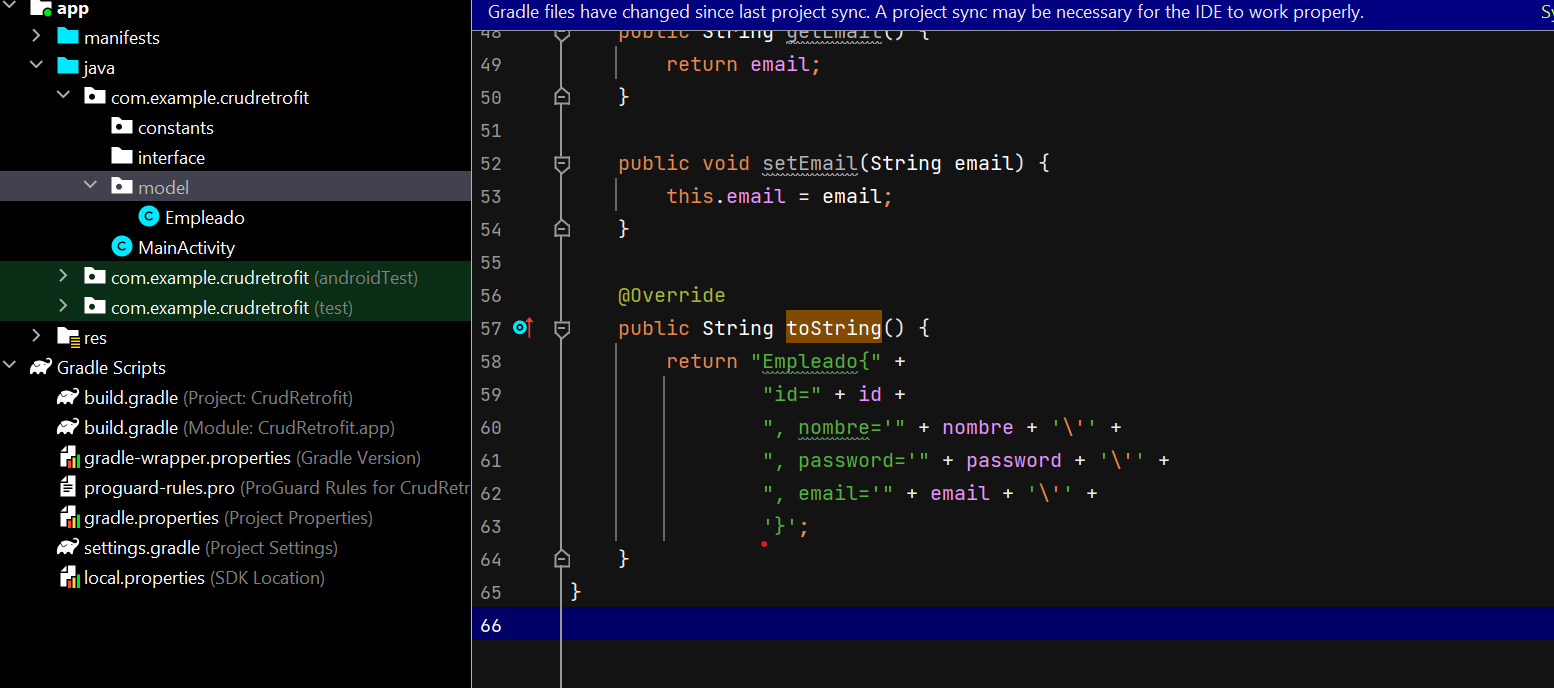
Luego se debe agregar un método constructor vacio y uno con todos los atributos



Después se deben incluir los métodos getter y setter

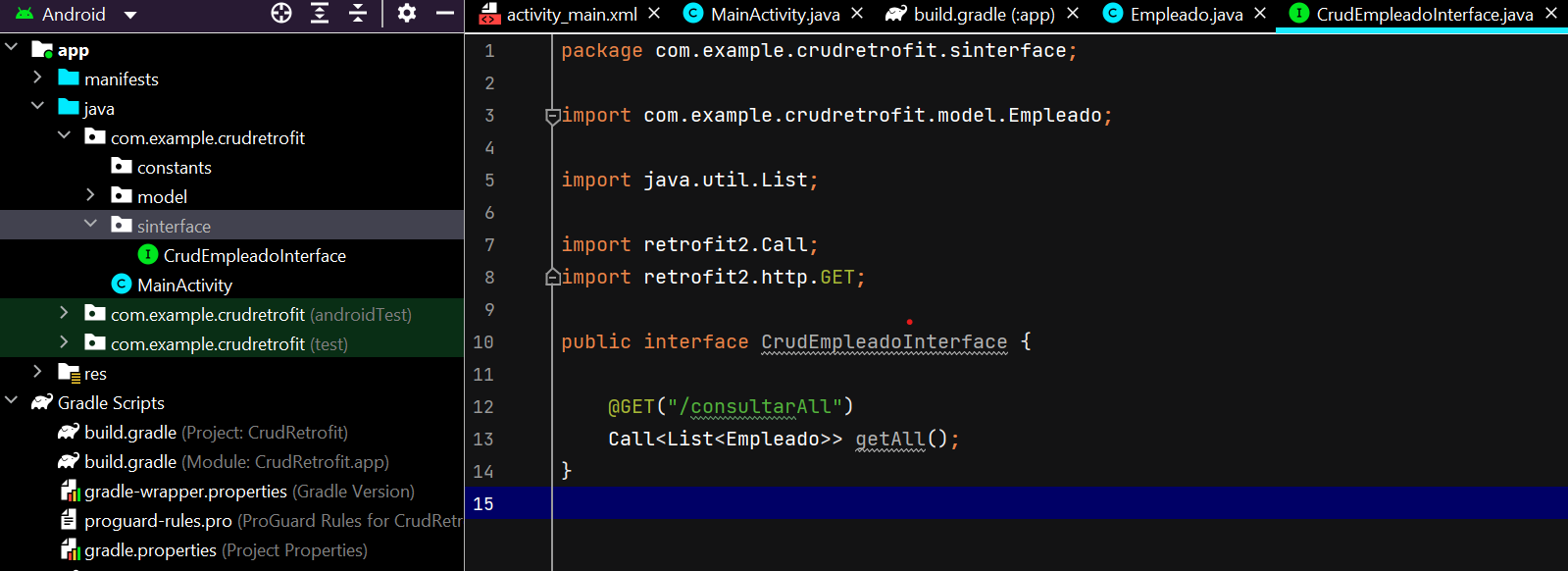


Y por ultimo sobre escribimos el método toString



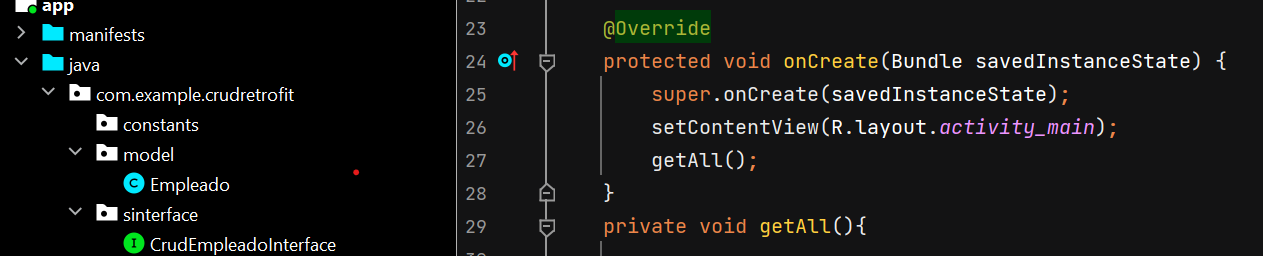
1. Crear interfaces con la implementación de los diferentes verbos

CrudEmpleadoInterface

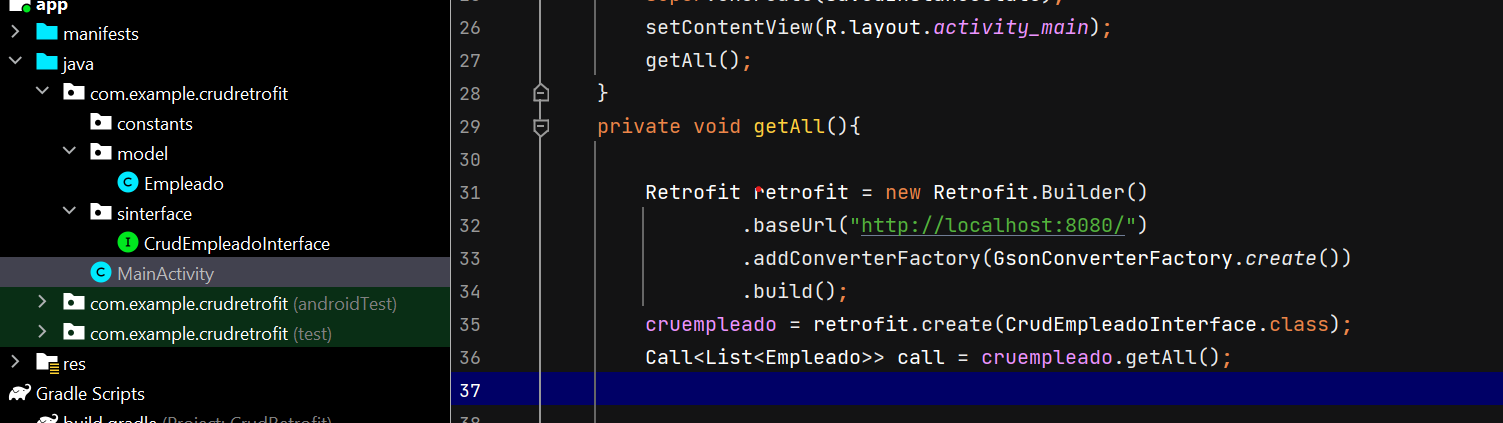


Inicialmente implementamos el método que nos permite obtener la lista de empleados , allí debemos tener en cuenta importar la funcionalidad Call que permite hacer llamado a servicios rest y la funcionalidad http.GET que le dice a Android que se va hacer un petición de tipo get

Luego modificamos el archivo MainActivity y agregamos en esa clase el método que nos permitirá obtener todos los empleados. Para ello hacemos el llamado desde el método OnCreate , quien se ejecuta una vez se despliegue el app

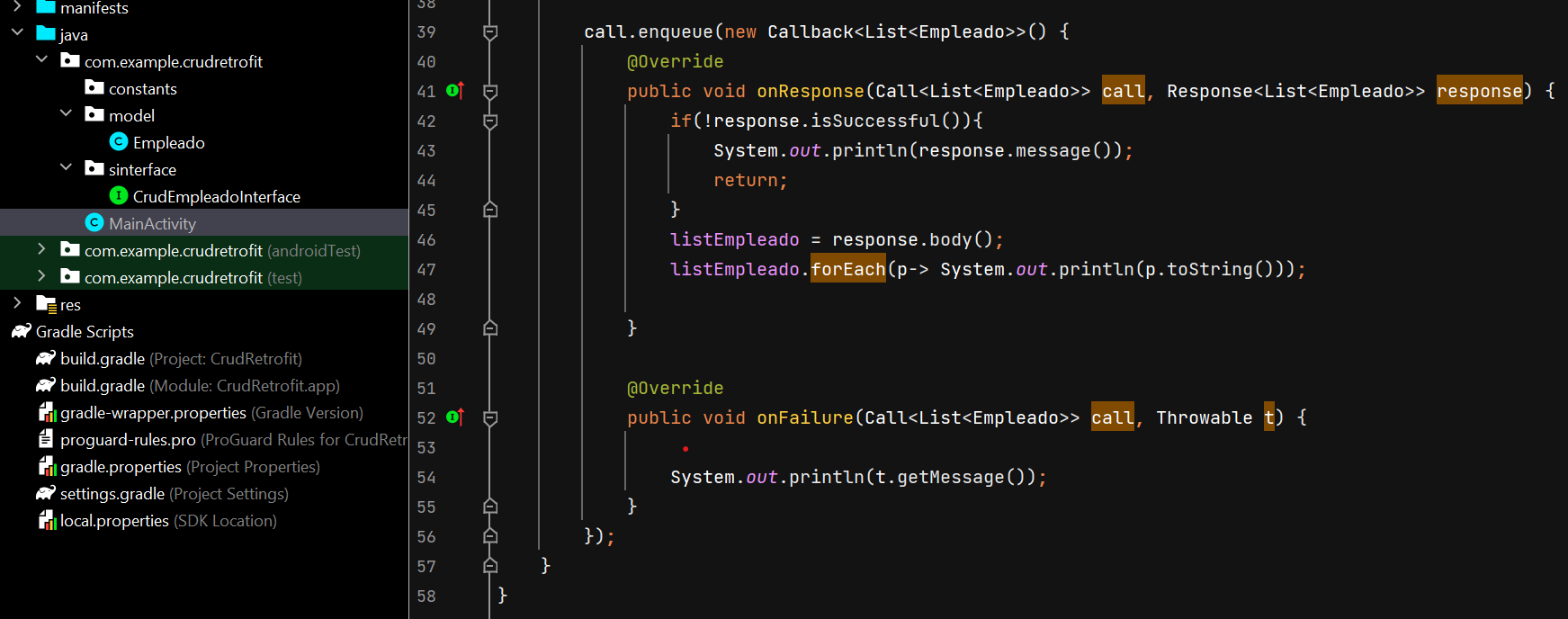


Luego contruimos la lógica del método



La primera parte de método contiene una instancia de la clase Retrofit con un llamado concatenado de alguno de sus métodos, por ejemplo el baseUrl define el path del servicio backend, el addConverterFactory le indica al programa que lo que reciba lo debe convertir a formato json y el build, significa que debe construir ese objeto con esas especificaciones.

Luego en la línea 35 se hace referencia al método que contruimos en la interface a través de la instancia del retrofit, en la línea 36 mediante la propiedad Call de retrofit preparamos en tiempo de ejecución el llamado al servicio, hasta allí aun no hemos llamado al backend.



Desde la línea 39 hasta la línea 56 atraves del método enqueue de la clase Call hacemos el llamado asíncrono del servicio backend y atraves de la implementación de los métodos onResponse y OnFaiture podemos capturar el mensaje de respuesta del servicio backend que puede ser exitoso o fallido.

Ahora haremos unos ajustes finales para que funcione correctamente el proyecto.

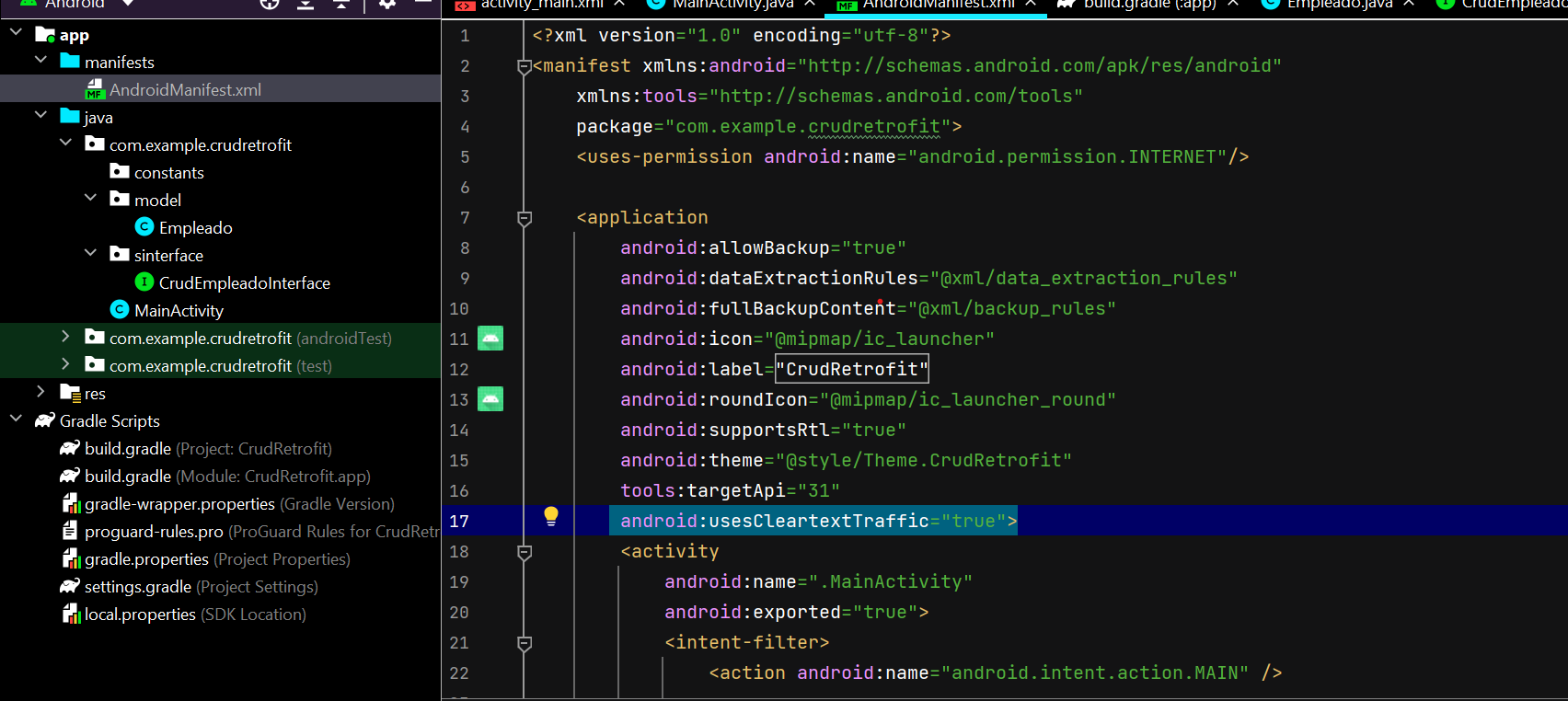
Lo primero que haremos en cambiar el System.ou.printl por Log de la siguiente forma:



Luego hacemos unos últimos ajustes en el archivo AndroidManifest.xml como se muestra a continuación

<uses-permission android:name="android.permission.INTERNET"/>

android:usesCleartextTraffic="true">



Si ejecutamos el proyecto y saca error en el servidor localhost , entonces debemos cambiar a localhost por la ip de nuestra máquina , exactamente en el método baseUrl dentro del método getAll en el archivo MainActivity, para conocer la ip de nuestra maquina digitamos en el cmd el comando ipconfig

En mi caso seria algo como esto :

