

CommandLineRunner es ideal cuando trabajamos en un proyecto con Spring Boot, sobre todo si necesitamos controlar hasta los aspectos más minuciosos de una aplicación, por ejemplo necesitamos controlar lo que sucede con la aplicación al momento de iniciarla.

Para eso nos sirve CommandLineRunner, con esa interfaz podemos definir un método en nuestra clase principal que se encargue controlar lo que sucede al momento de iniciar la aplicación, veamos:

```
package com.example;

import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;

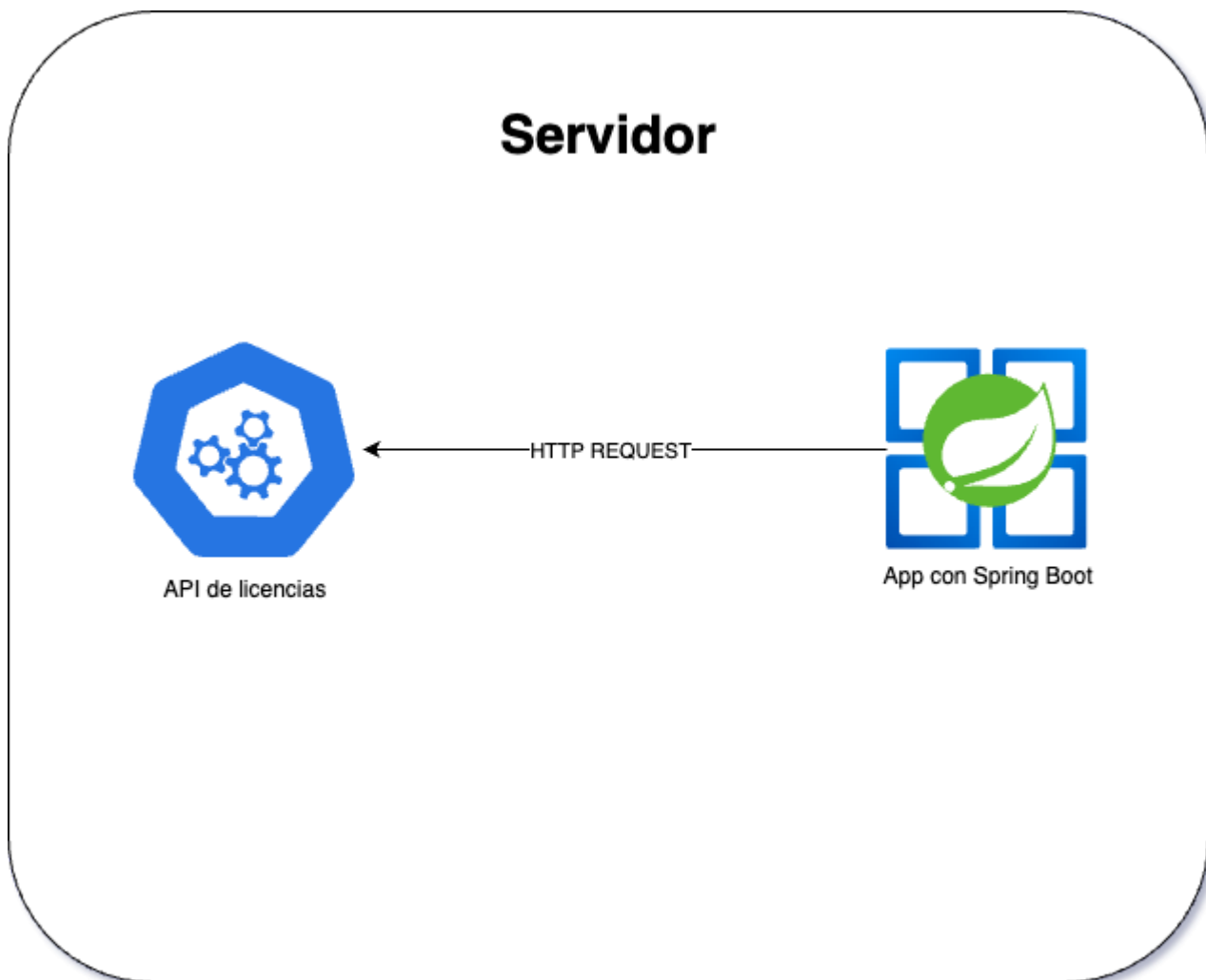
@SpringBootApplication
public class ExampleApplication implements CommandLineRunner {

    public static void main(String[] args) {
        SpringApplication.run(ExampleApplication.class, args);
    }

    @Override
    public void run(String... args) throws Exception {
        //Aquí podemos agregar lógica.
    }

}
```

En el código anterior se muestra la implementación de la clase lo interesante aquí es que podemos controlar el inicio de la aplicación, hagamos un caso de uso sencillo.



Necesitamos conectarnos a un API para validar si la licencia de un cliente en particular se encuentra activa, y en caso de no estar activar la aplicación no se debe iniciar, para lograrlo necesitamos modificar el método run y vamos a asumir que tenemos un servicio que hace el trabajo de conectarse al API antes mencionada, dicho servicio nos devuelve un boolean.

```
@Autowired
private LicenseService licenseService;

@Override
public void run(String... args) throws Exception {
    if(!licenseService.isLicenseActive()) {
        throw new Exception("App can't be started because license is
invalid");
    }
}
```

```
}  
}
```

En caso de que el método `isLicenseActive()` retorne `false`, lanzaremos una excepción y esto va a ocasionar que la aplicación no pueda ejecutarse, es maravilloso.

Realmente las posibilidades no son pocas así que te voy a dejar 5 casos de uso reales donde puedes llegar a implementar `CommandLineRunner`

1. Inicialización de Datos en la Base de Datos

A menudo, en las aplicaciones necesitamos cargar datos iniciales en la base de datos al arrancar la aplicación. Esto puede incluir datos de configuración, datos de prueba o datos esenciales para el funcionamiento de la aplicación.

2. Carga de Configuraciones Externas

Caso de Uso:

Si tu aplicación necesita cargar configuraciones desde un archivo externo o un servicio web al iniciar, puedes usar `CommandLineRunner` para hacerlo.

3. Enviar Notificaciones o Alertas al Inicio

Caso de Uso:

Puedes necesitar enviar notificaciones o alertas (por ejemplo, al equipo de DevOps) cuando la aplicación se inicia para indicar que el servicio está en funcionamiento.

4. Ejecutar Tareas de Limpieza

Caso de Uso:

Si necesitas realizar alguna limpieza de recursos temporales o viejos al inicio de la aplicación, puedes hacerlo usando `CommandLineRunner`.

5. Configuración de Servicios Externos

Caso de Uso:

A veces, tu aplicación puede depender de servicios externos que necesitan ser configurados

o inicializados al arrancar. Por ejemplo, conectar y configurar un servidor de mensajería o un sistema de archivos distribuido.