**Feign Client :**

Feign is a Java library used to simplify making HTTP requests to external APIs. It allows you to define API clients using interfaces and annotations, making the code more readable and maintainable. A Feign client is an interface annotated with **@FeignClient**, specifying the external service's details and methods to call. Spring Cloud integrates Feign to make it easy to use in Spring Boot applications.

The **@EnableFeignClients** annotation in Spring Cloud is used to enable Feign clients in your Spring Boot application. Feign is a declarative web service client that makes it easier to create web service clients in Java by providing a simple way to define HTTP APIs as Java interfaces.

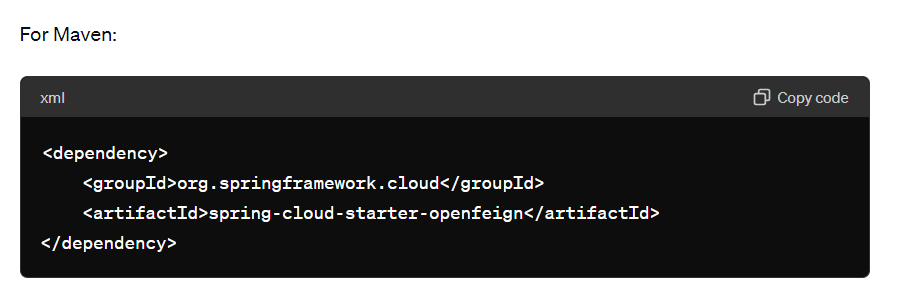
**Usage**

**Basic Setup**

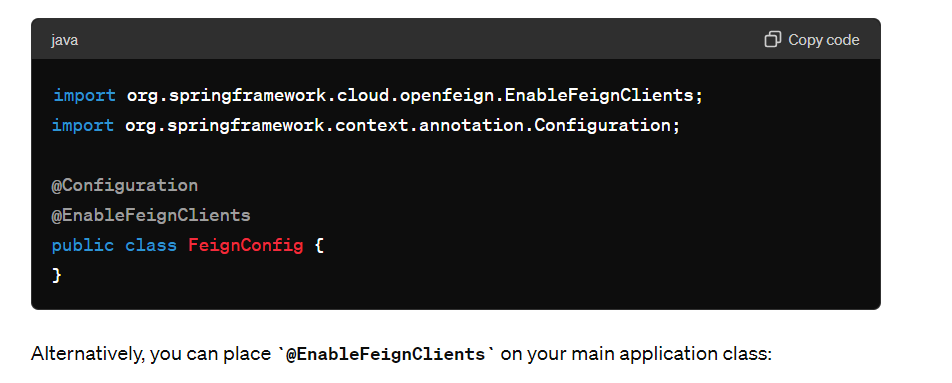
To use Feign in a Spring Boot application, you need to follow these steps:

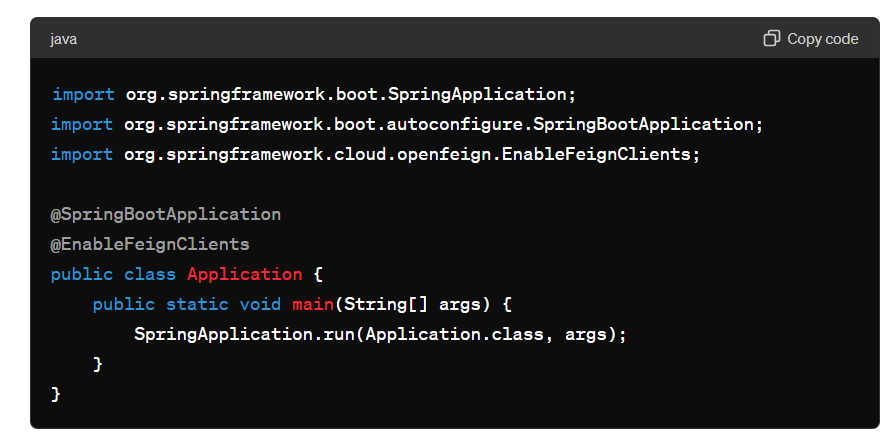
1. **Add Dependencies**: Include the necessary dependencies in your **pom.xml** (for Maven) or **build.gradle** (for Gradle).

For Maven:

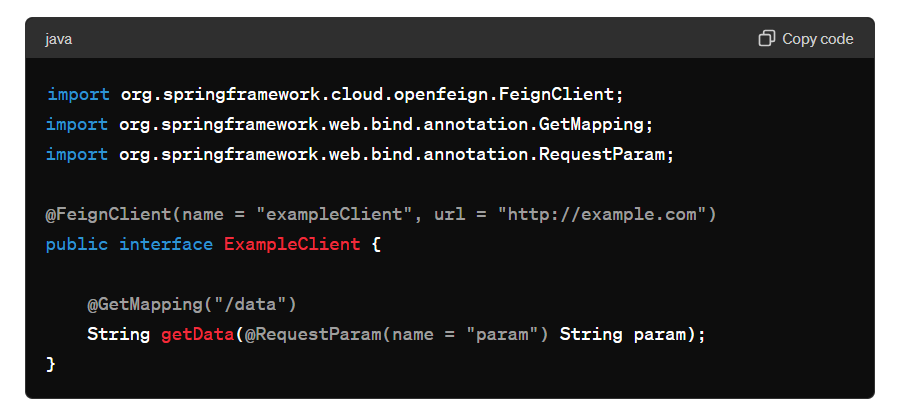


**Enable Feign Clients**: Annotate a configuration class with **@EnableFeignClients** to scan for interfaces that declare they are Feign clients





**Define a Feign Client**: Create an interface and annotate it with **@FeignClient**. This annotation marks it as a Feign client and allows you to specify the service it will call.

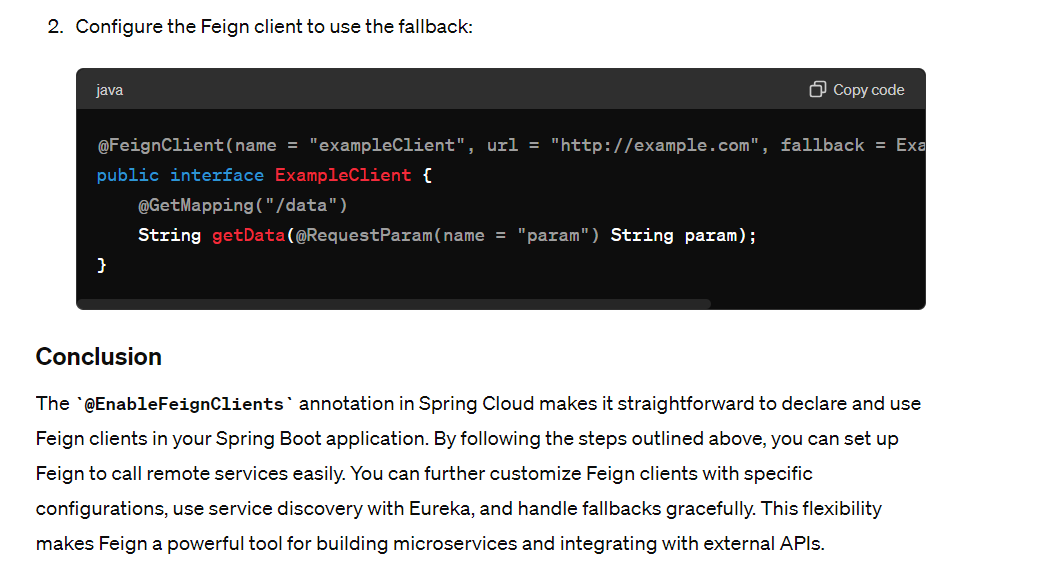


#### Handling Fallbacks

Feign provides a way to handle fallbacks if the remote service is unavailable. You can specify a fallback class in the **@FeignClient** annotation.

1. Define a fallback class:





@FeignClient(name = "exampleClient", url = "http://example.com", fallback = ExampleClientFallback.class)