Microservices

Spring Boot

Get Course along with Student

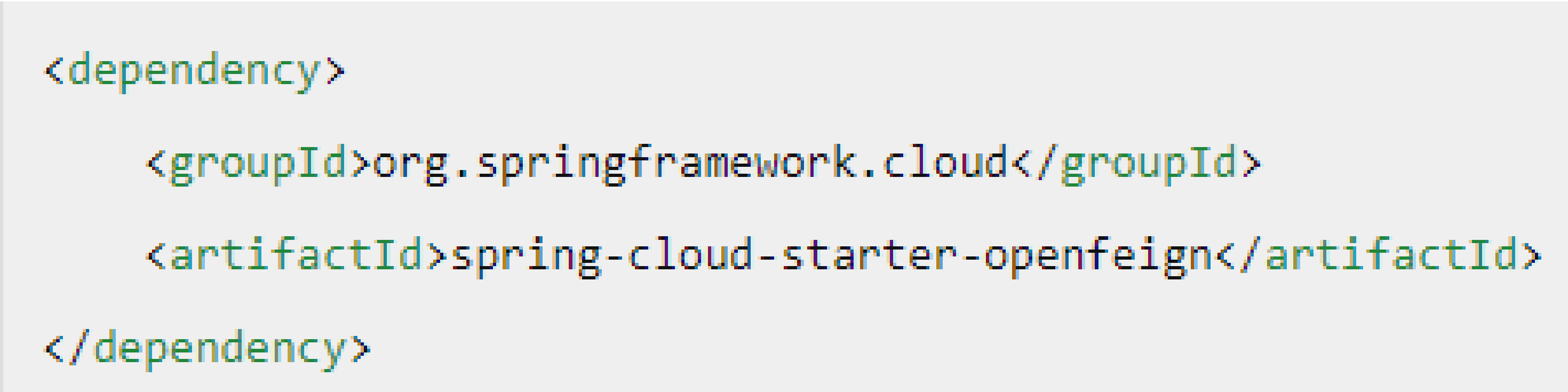
Using Spring Cloud Open Feign

Open Fiegn

Spring Cloud OpenFeign is a declarative REST client for Spring Boot applications.

It simplifies the process of creating REST clients by allowing developers to define an interface and annotate it.

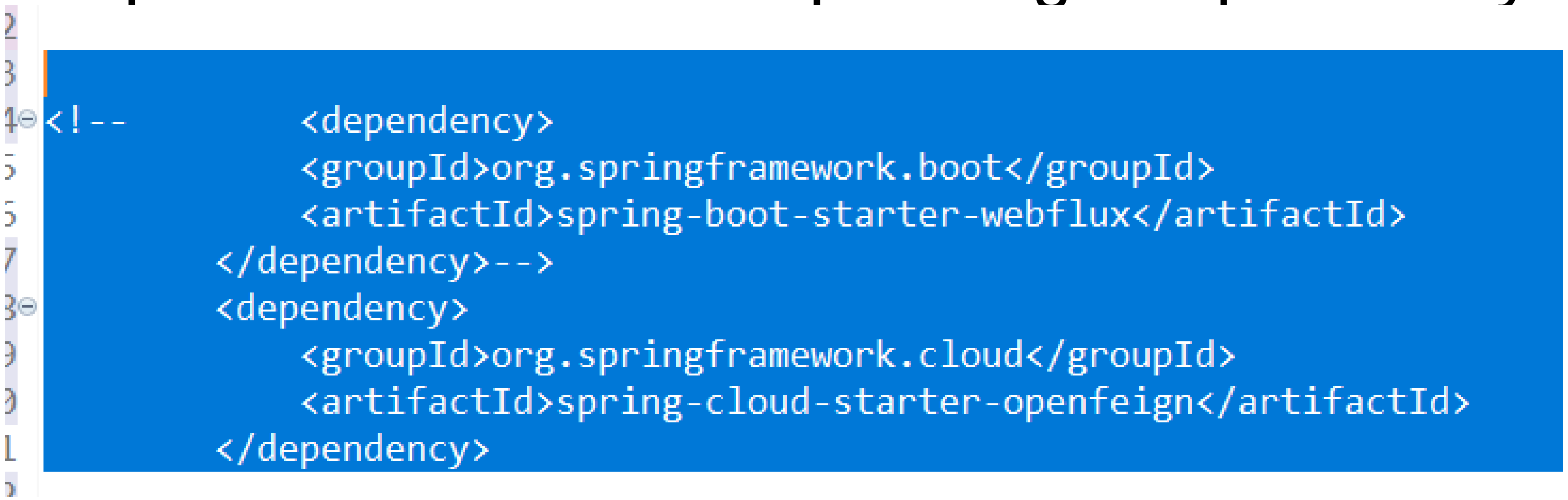
The implementation is generated at runtime, reducing boilerplate code.



```
<dependency>

    <groupId>org.springframework.cloud</groupId>

    <artifactId>spring-cloud-starter-openfeign</artifactId>

</dependency>
```

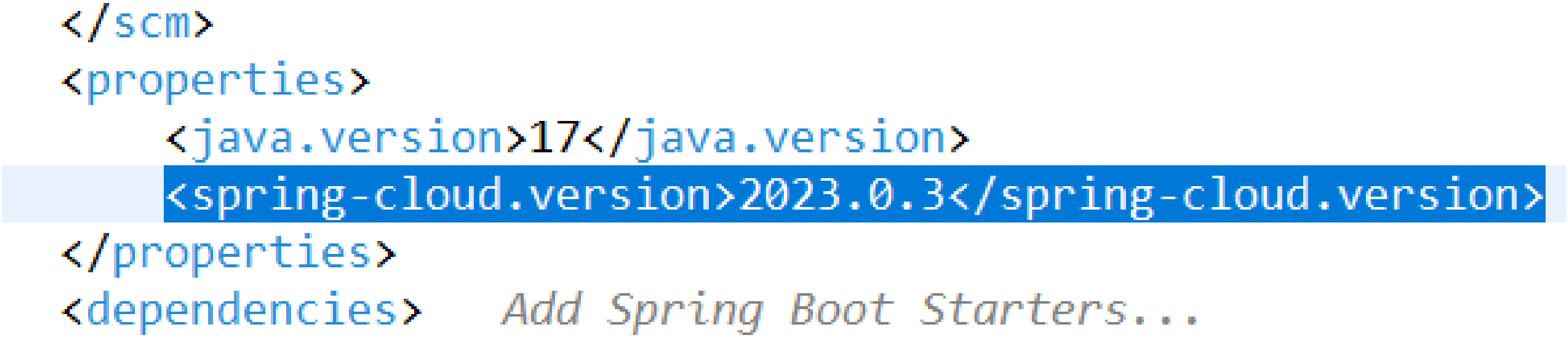
Updating pom.xml

- Replace Webflux with OpenFeign dependency



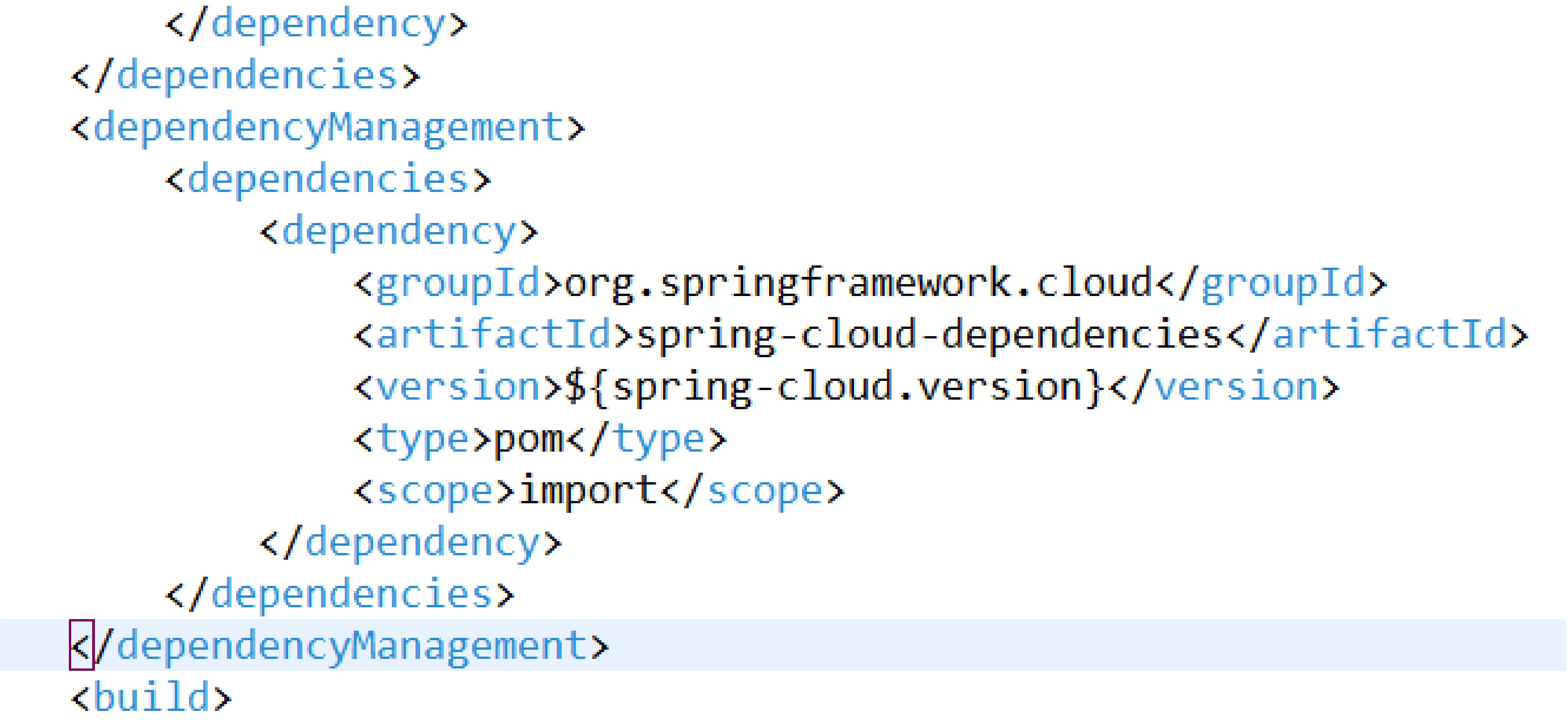
```
<!--            <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-webflux</artifactId>
        </dependency>-->
        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-starter-openfeign</artifactId>
        </dependency>
```

- Include spring cloud version in properties



```
</scm>
<properties>
    <java.version>17</java.version>
    <spring-cloud.version>2023.0.3</spring-cloud.version>
</properties>
<dependencies>    Add Spring Boot Starters...
```

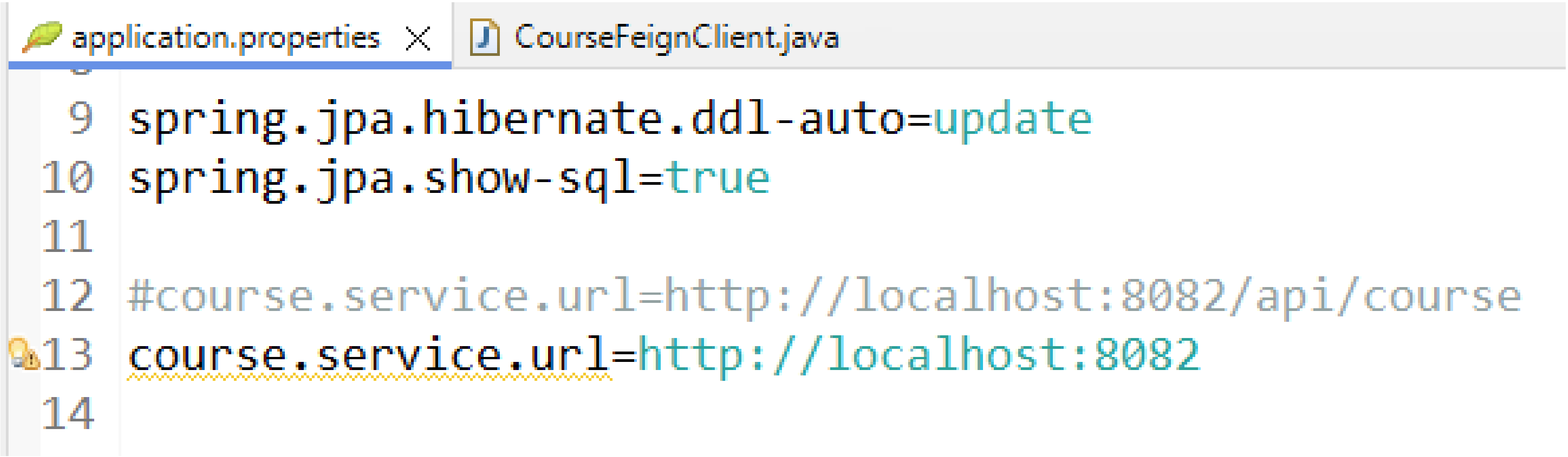
Add spring cloud dependencies management



```
        </dependency>
    </dependencies>
    <dependencyManagement>
        <dependencies>
            <dependency>
                <groupId>org.springframework.cloud</groupId>
                <artifactId>spring-cloud-dependencies</artifactId>
                <version>${spring-cloud.version}</version>
                <type>pom</type>
                <scope>import</scope>
            </dependency>
        </dependencies>
    </dependencyManagement>
    <build>
```

In Properties file

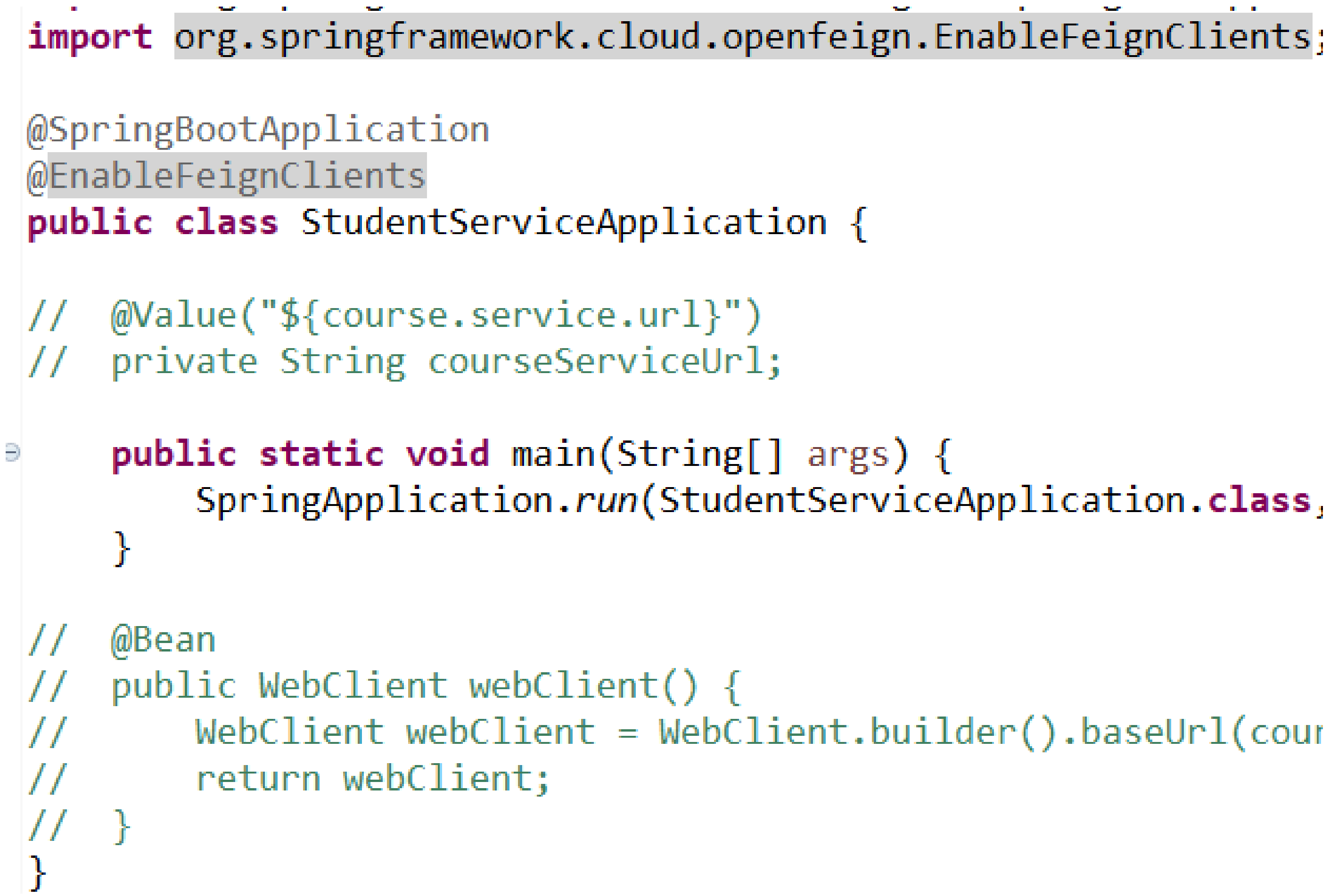
- Replace the url to base path
- To use an attribute “path” in OpenFeign Client



```
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true

#course.service.url=http://localhost:8082/api/course
course.service.url=http://localhost:8082

```

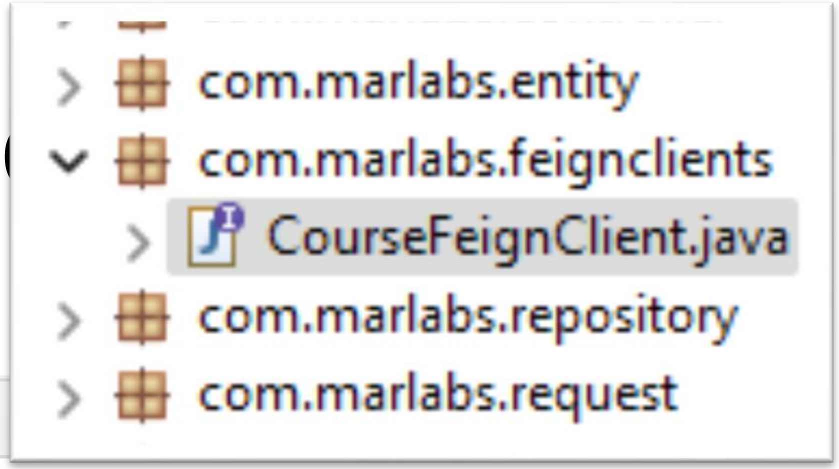
In Main Class

- Comment or Delete @Value & @Bean WebFlux
- Include @EnableFiegnClients

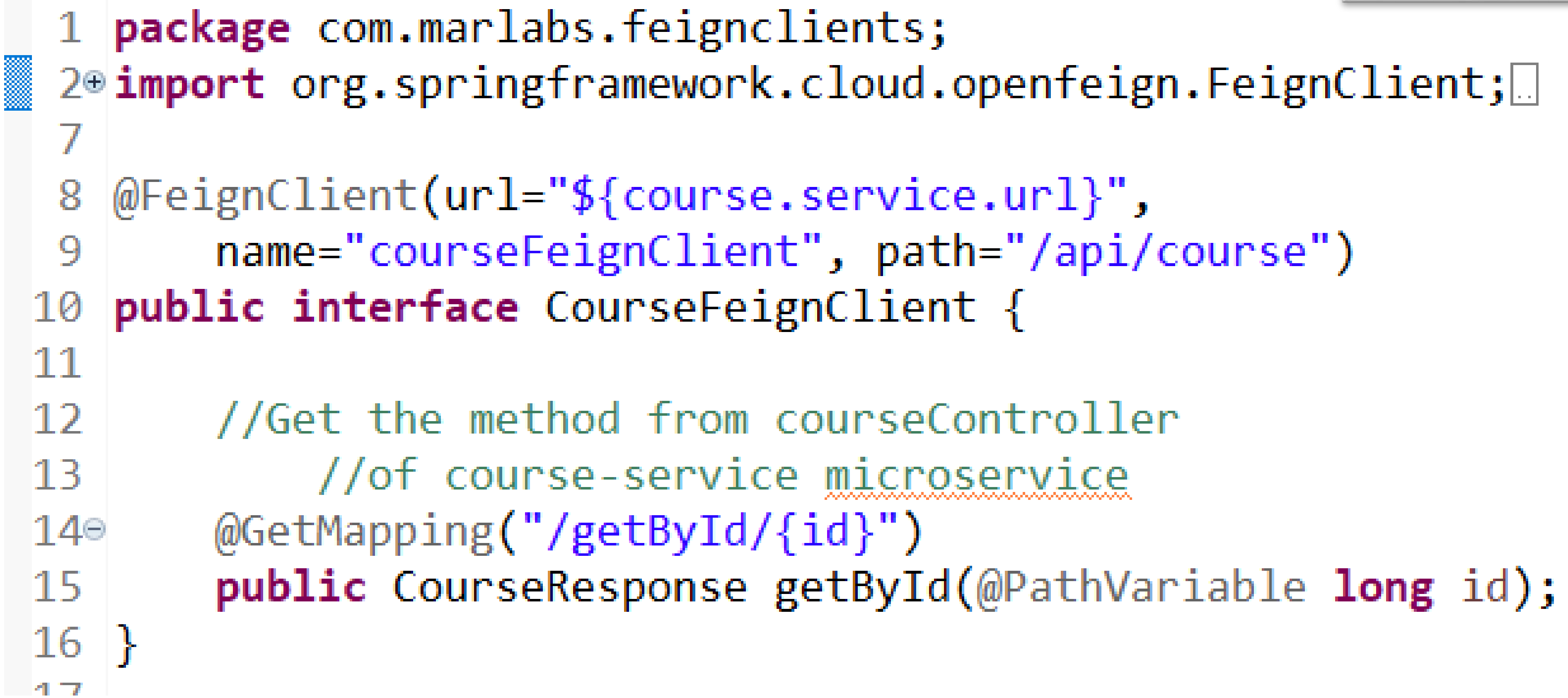


```
import org.springframework.cloud.openfeign.EnableFeignClients;

@SpringBootApplication
@EnableFeignClients
public class StudentServiceApplication {

//  @Value("${course.service.url}")
//  private String courseServiceUrl;

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

//  @Bean
//  public WebClient webClient() {
//      WebClient webClient = WebClient.builder().baseUrl(cour
//      return webClient;
//  }
}
```

New package with an interfa

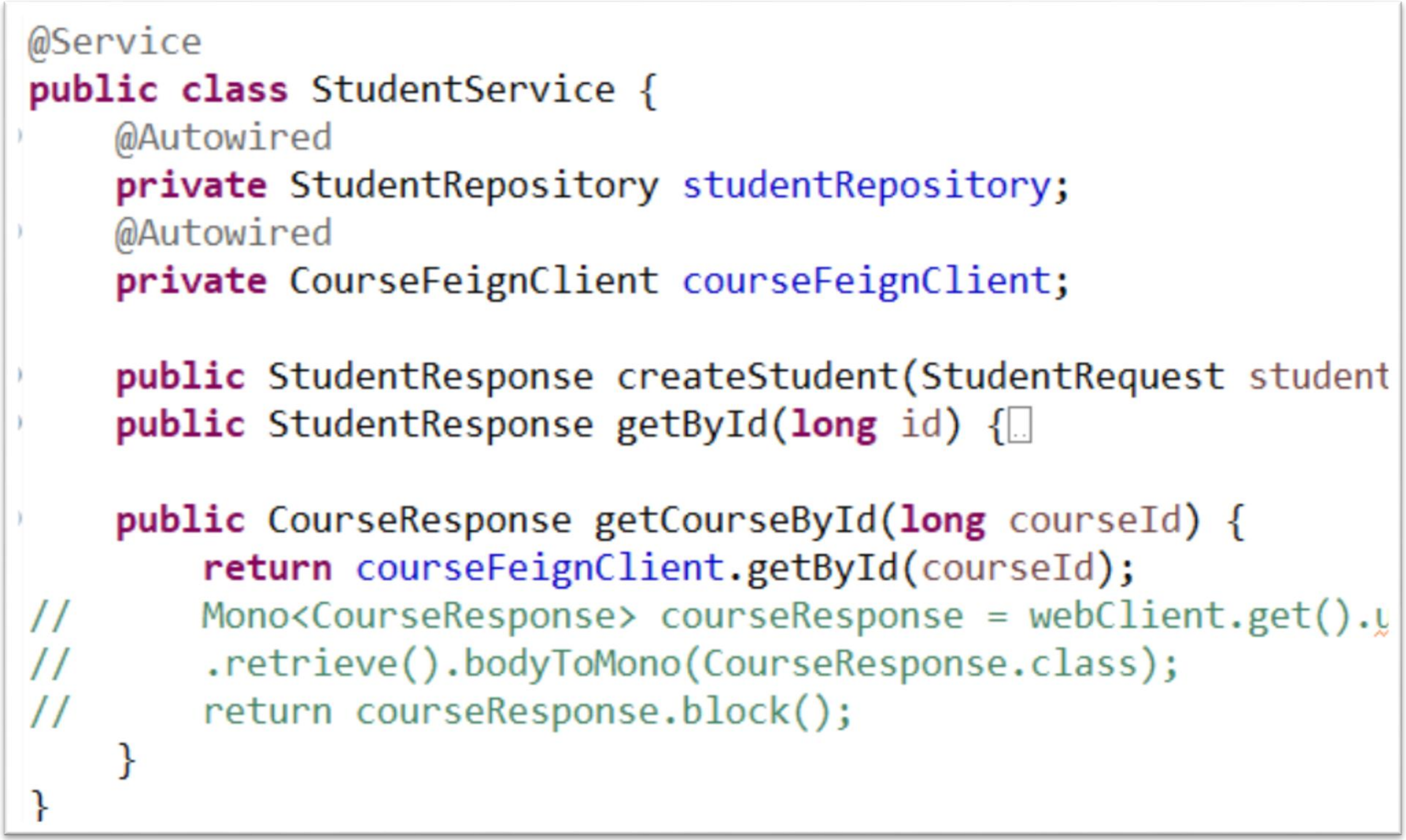


CourseFeignClient.java

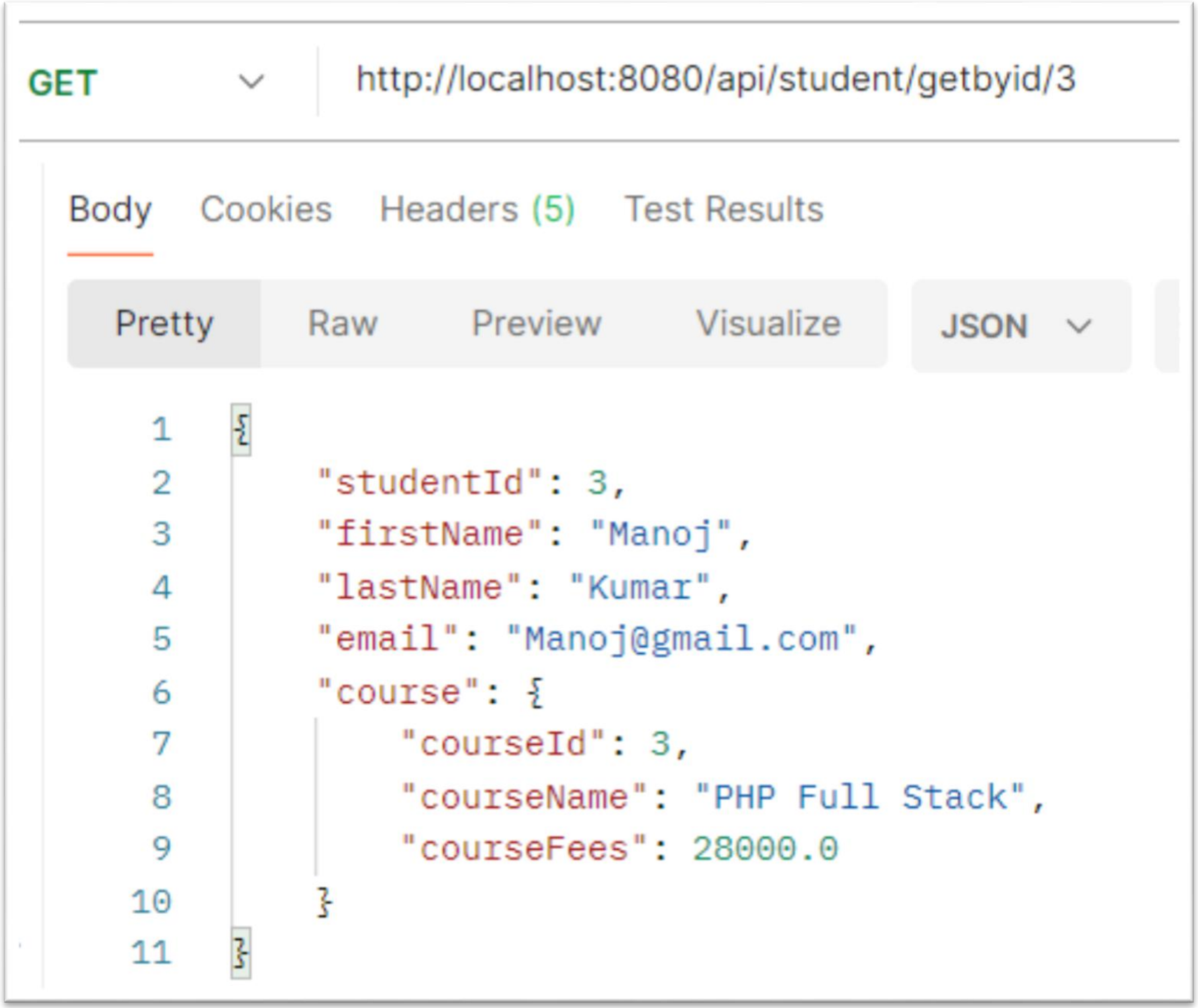


```java
package com.marlabs.feignclients;
import org.springframework.cloud.openfeign.FeignClient;

@FeignClient(url="${course.service.url}",
        name="courseFeignClient", path="/api/course")
public interface CourseFeignClient {

        //Get the method from courseController
            //of course-service microservice
        @GetMapping("/getById/{id}")
        public CourseResponse getById(@PathVariable long id);
}
```

Autowire and update Student Service



```
@Service
public class StudentService {
    @Autowired
    private StudentRepository studentRepository;
    @Autowired
    private CourseFeignClient courseFeignClient;

    public StudentResponse createStudent(StudentRequest student
    public StudentResponse getById(long id) {□

    public CourseResponse getCourseById(long courseId) {
        return courseFeignClient.getById(courseId);
//      Mono<CourseResponse> courseResponse = webClient.get().u
//      .retrieve().bodyToMono(CourseResponse.class);
//      return courseResponse.block();
    }
}
```



http://localhost:8080/api/student/getbyid/3

```
{
    "studentId": 3,
    "firstName": "Manoj",
    "lastName": "Kumar",
    "email": "Manoj@gmail.com",
    "course": {
        "courseId": 3,
        "courseName": "PHP Full Stack",
        "courseFees": 28000.0
    }
}
```