Flyway, açık kaynaklı database migration'ları yapmamızı sağlayan bir tool'dur.
Sadeliği ve konfigürasyondan bağımsız konvensiyonel bir mekanizmayı ön plana çıkarmayı amaçlar.

Ücretsiz olarak kullanabileceğimiz Kong API Gateway => https://konghq.com/products/api-gateway-platform

Kong açık kaynak kodlu ölçeklenebilir API Gateway ürünüdür. Clientlerınızın servislerinize bir endpoint üzerinden ulaşmasını sağlar yani artık clientleriniz doğrudan servislerinizi değil kong endpointini çağırır ve bu endpoint ilgili servise yönlendirme yapar. Kong yönlendirme yaparken size sağladığı pluginler ile authentication,security, response rate limiting, request size limiting, logging gibi çözümler sunar.

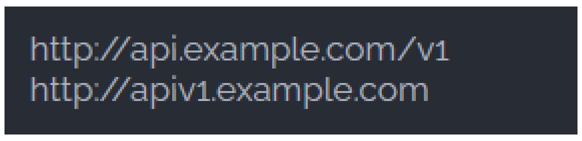
WSO2 API Gateway=>

https://apim.docs.wso2.com/en/latest/deploy-and-publish/deploy-on-gateway/api-gateway/overview-of-the-api-gateway/

https://restfulapi.net/versioning/

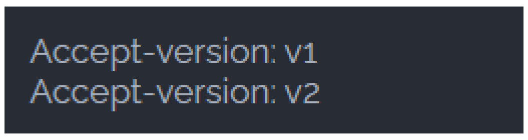
REST API nasıl versiyonlanmalı?

1. URI Versioning



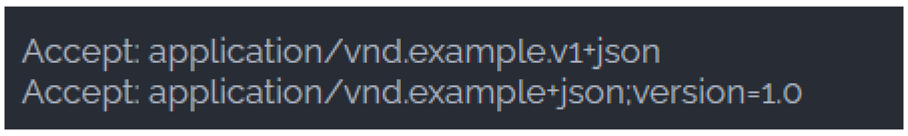
```
http://api.example.com/v1
http://apiv1.example.com
```

2. Request Header ile Versioning

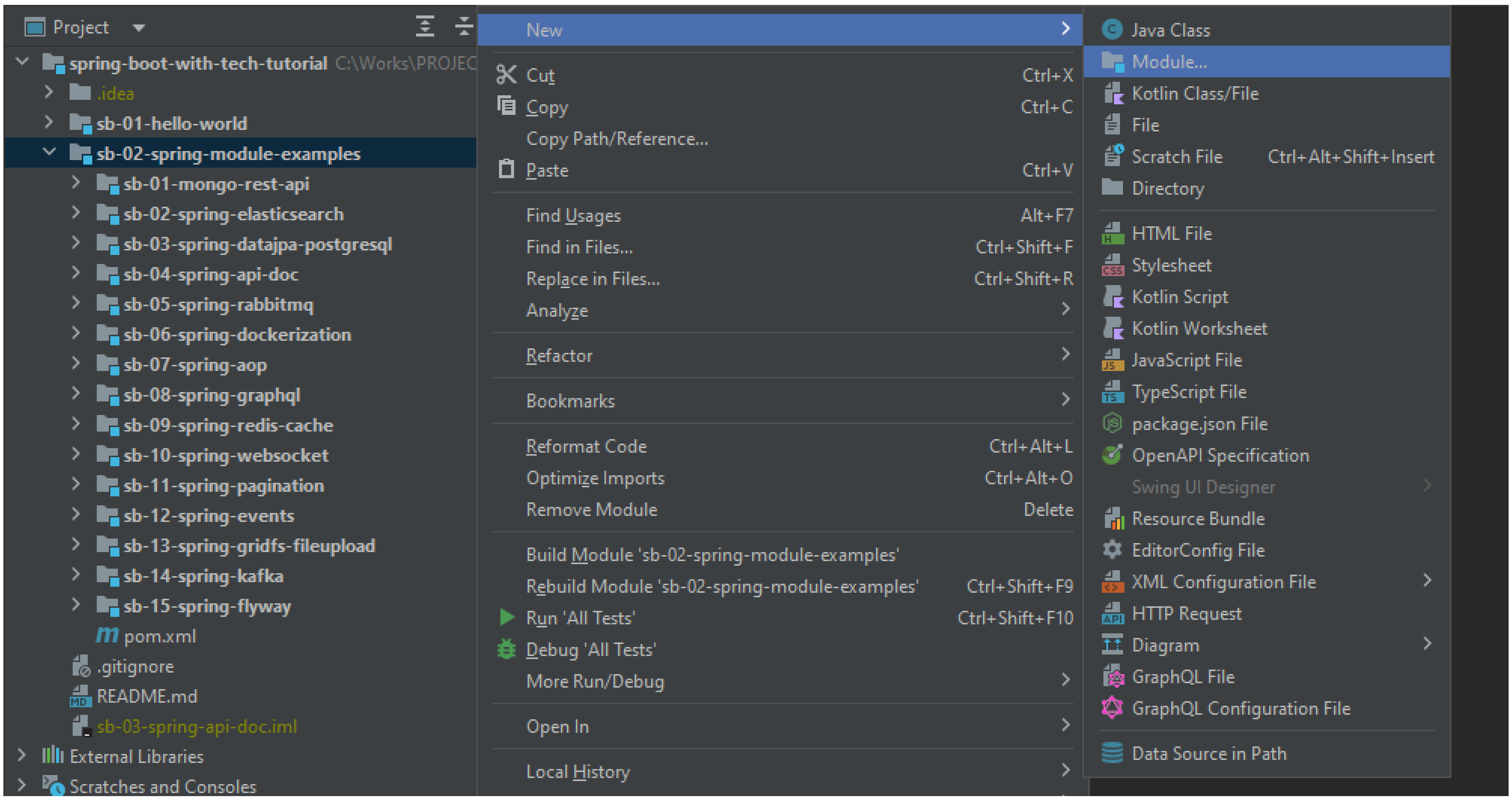


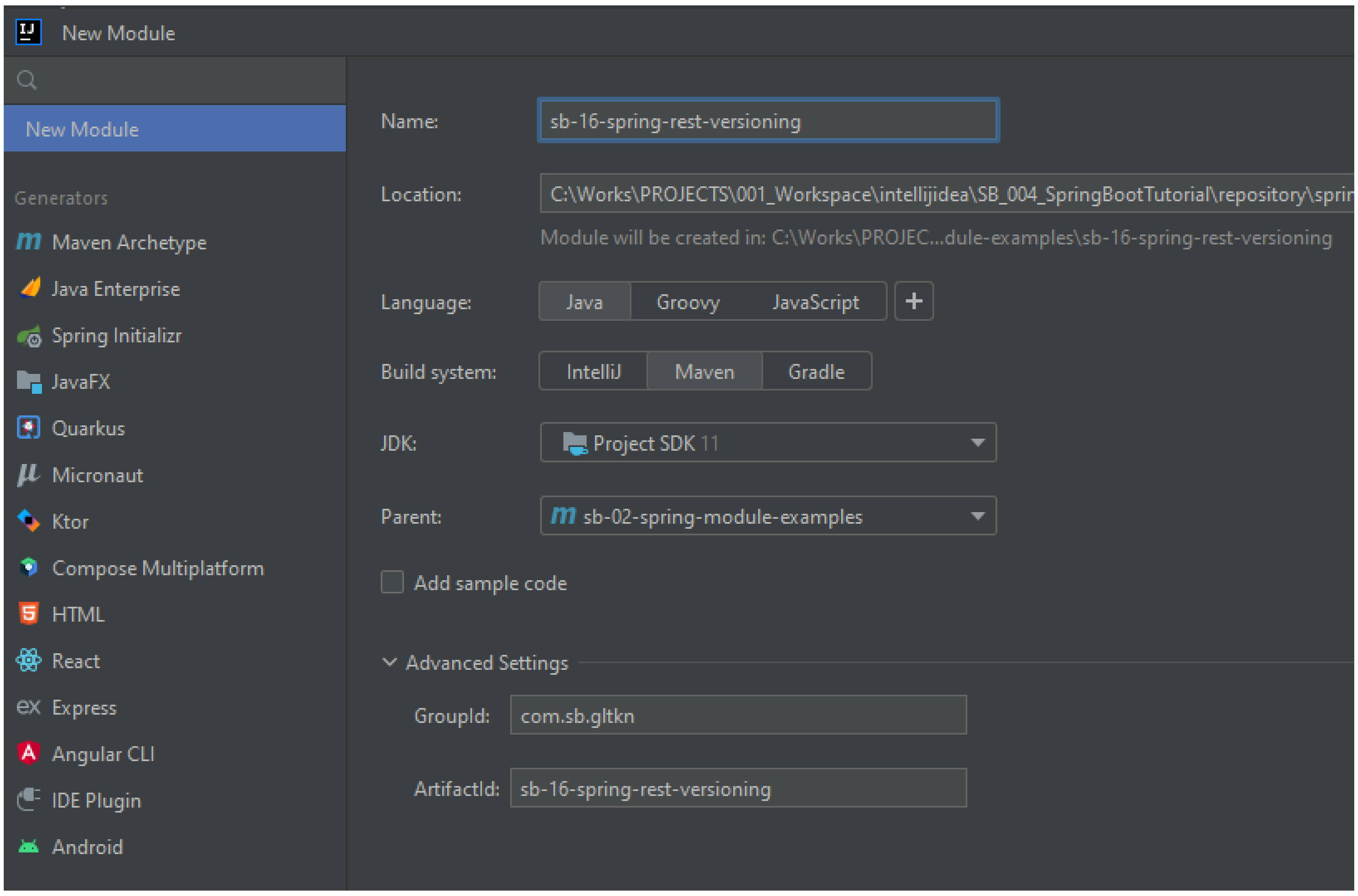
```
Accept-version: v1
Accept-version: v2
```

3. “Accept” header ile Versioning



```
Accept: application/vnd.example.v1+json
Accept: application/vnd.example+json;version=1.0
```





Name: sb-16-spring-rest-versioning

Location: C:\Works\PROJECTS\001_Workspace\intellijidea\SB_004_SpringBootTutorial\repository\sprin

Module will be created in: C:\Works\PROJEC...dule-examples\sb-16-spring-rest-versioning

Language: Java | Groovy | JavaScript | +

Build system: IntelliJ | Maven | Gradle

JDK: Project SDK 11

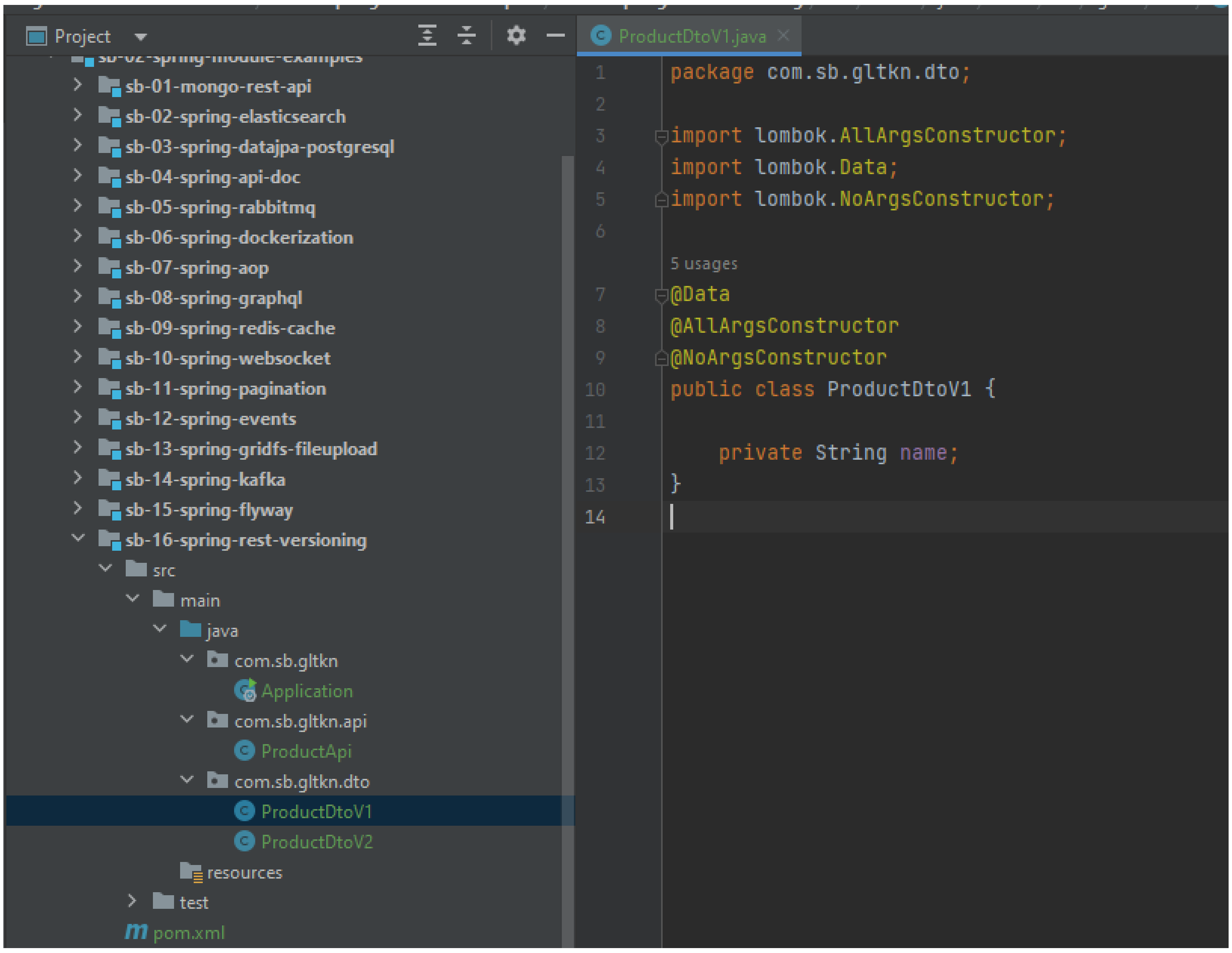
Parent: sb-02-spring-module-examples

□ Add sample code

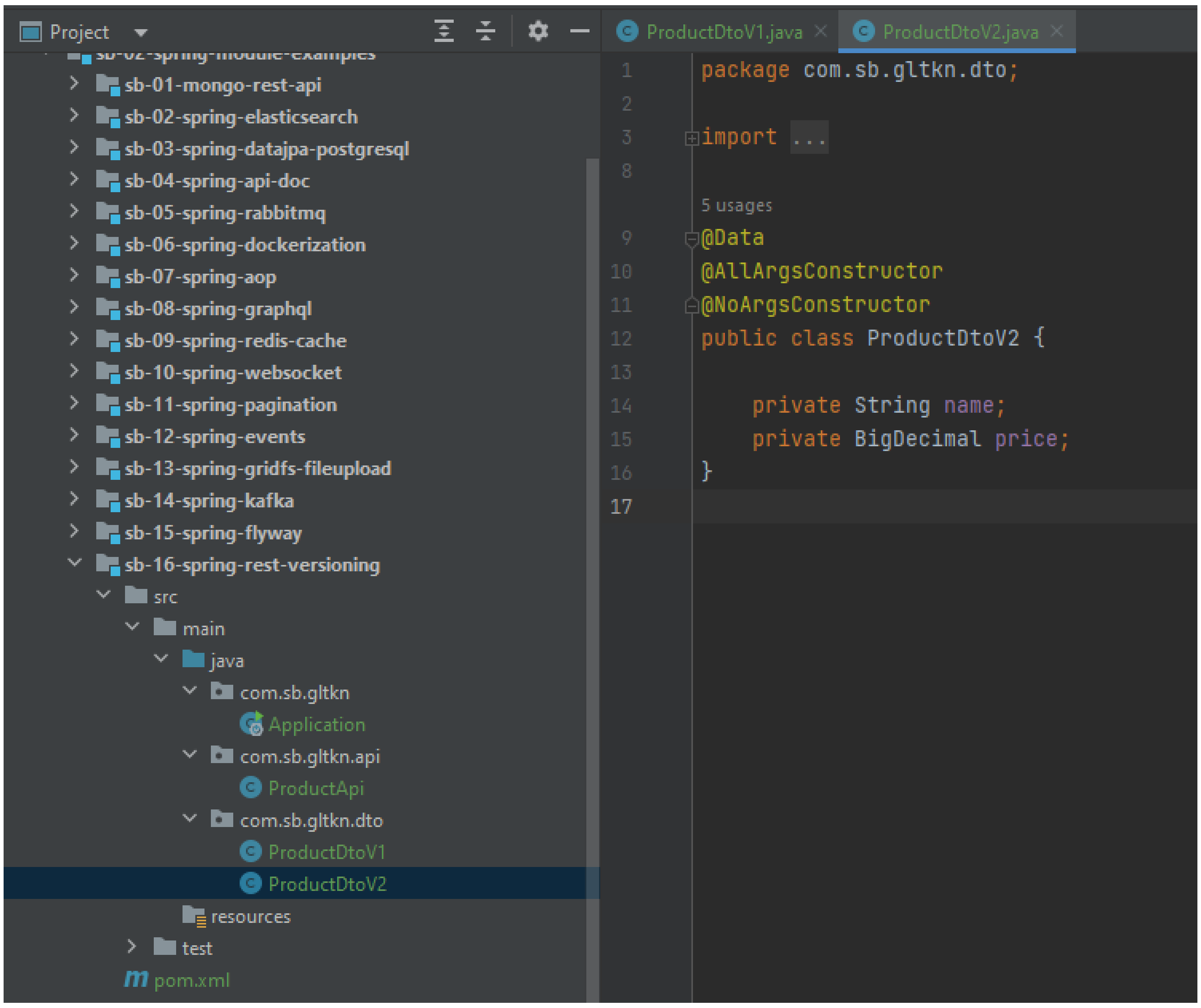
Advanced Settings

GroupId: com.sb.gltkn

ArtifactId: sb-16-spring-rest-versioning

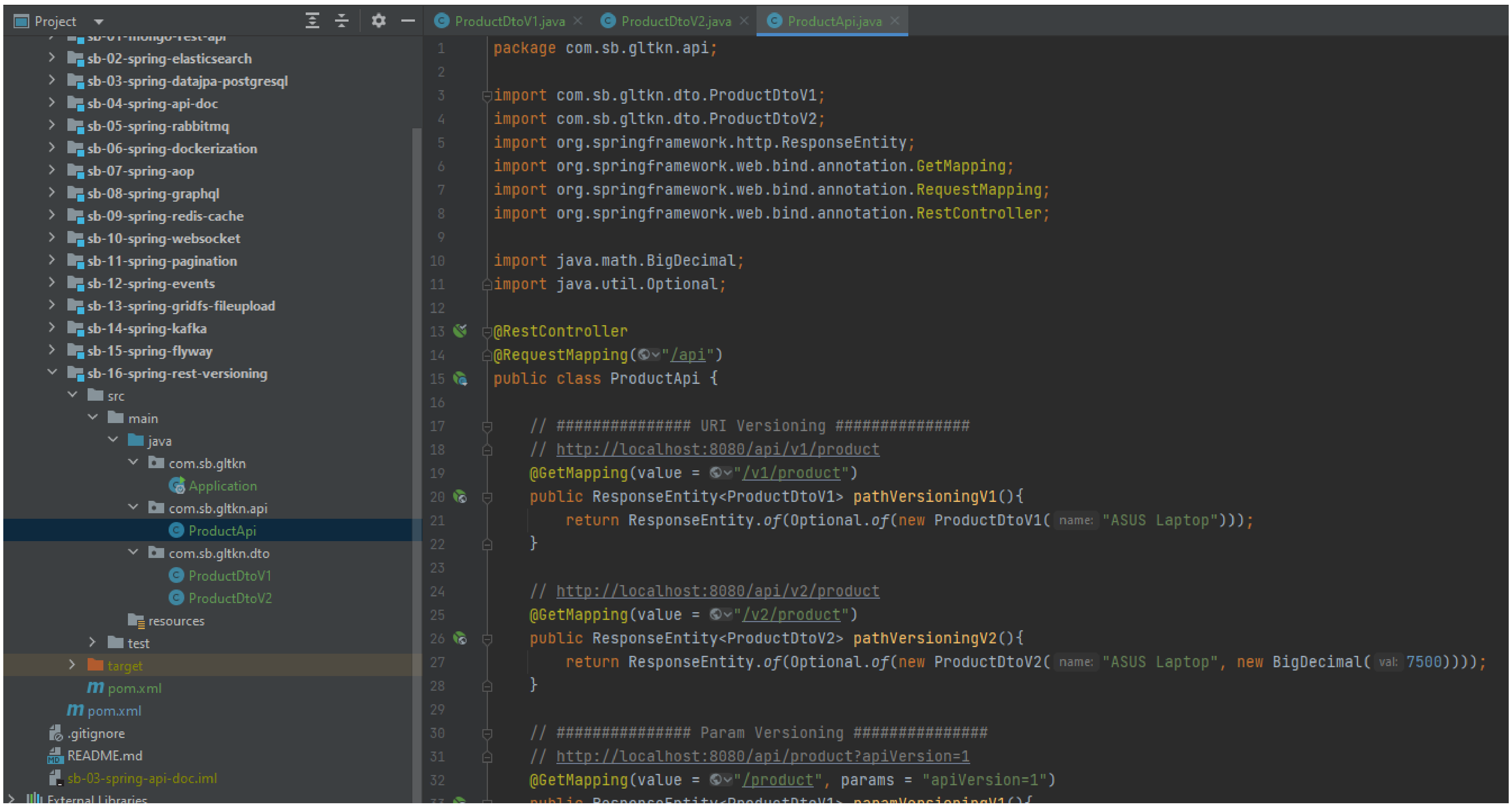


```java
package com.sb.gltkn.dto;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class ProductDtoV1 {

    private String name;
}
```



```
sb-02-spring-module-examples
  sb-01-mongo-rest-api
  sb-02-spring-elasticsearch
  sb-03-spring-datajpa-postgresql
  sb-04-spring-api-doc
  sb-05-spring-rabbitmq
  sb-06-spring-dockerization
  sb-07-spring-aop
  sb-08-spring-graphql
  sb-09-spring-redis-cache
  sb-10-spring-websocket
  sb-11-spring-pagination
  sb-12-spring-events
  sb-13-spring-gridfs-fileupload
  sb-14-spring-kafka
  sb-15-spring-flyway
  sb-16-spring-rest-versioning
    src
      main
        java
          com.sb.gltkn
            Application
          com.sb.gltkn.api
            ProductApi
          com.sb.gltkn.dto
            ProductDtoV1
            ProductDtoV2
        resources
      test
    pom.xml
```

```java
package com.sb.gltkn.dto;

import ...

5 usages
@Data
@AllArgsConstructor
@NoArgsConstructor
public class ProductDtoV2 {

    private String name;
    private BigDecimal price;
}
```



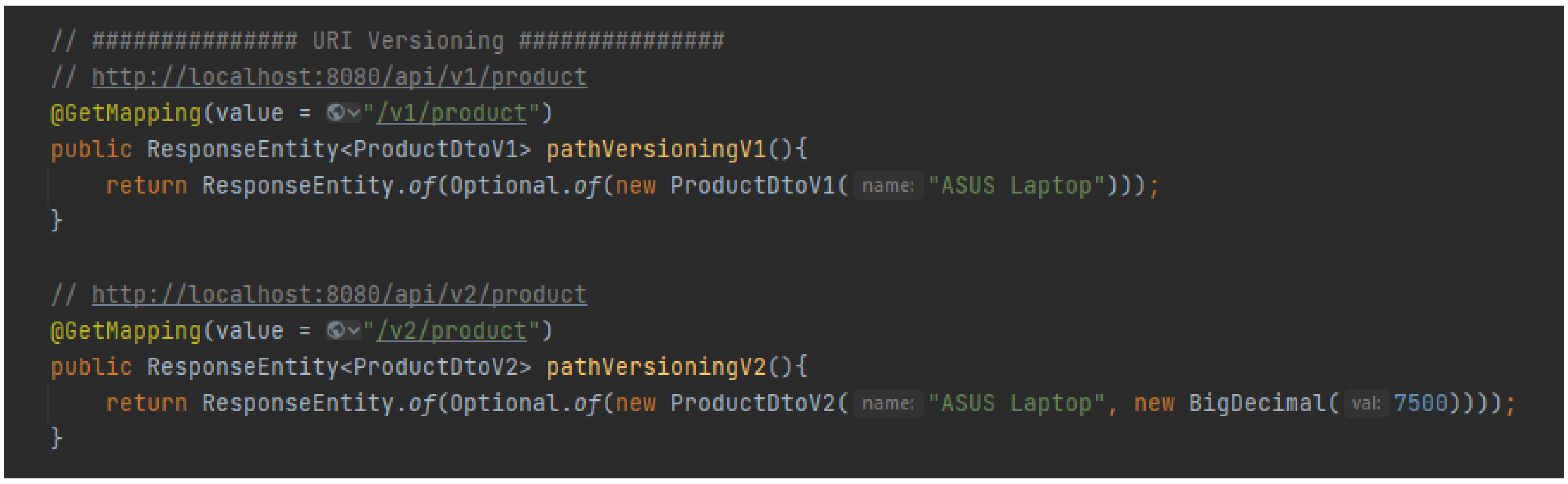
```
  sb-01-mongo-rest-api
  sb-02-spring-elasticsearch
  sb-03-spring-datajpa-postgresql
  sb-04-spring-api-doc
  sb-05-spring-rabbitmq
  sb-06-spring-dockerization
  sb-07-spring-aop
  sb-08-spring-graphql
  sb-09-spring-redis-cache
  sb-10-spring-websocket
  sb-11-spring-pagination
  sb-12-spring-events
  sb-13-spring-gridfs-fileupload
  sb-14-spring-kafka
  sb-15-spring-flyway
  sb-16-spring-rest-versioning
    src
      main
        java
          com.sb.gltkn
            Application
          com.sb.gltkn.api
            ProductApi
          com.sb.gltkn.dto
            ProductDtoV1
            ProductDtoV2
        resources
      test
    target
    pom.xml
  pom.xml
  .gitignore
  README.md
  sb-03-spring-api-doc.iml
External Libraries
```

```java
package com.sb.gltkn.api;

import com.sb.gltkn.dto.ProductDtoV1;
import com.sb.gltkn.dto.ProductDtoV2;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import java.math.BigDecimal;
import java.util.Optional;

@RestController
@RequestMapping("/api")
public class ProductApi {

    // ############### URI Versioning ###############
    // http://localhost:8080/api/v1/product
    @GetMapping(value = "/v1/product")
    public ResponseEntity<ProductDtoV1> pathVersioningV1(){
        return ResponseEntity.of(Optional.of(new ProductDtoV1( name: "ASUS Laptop")));
    }

    // http://localhost:8080/api/v2/product
    @GetMapping(value = "/v2/product")
    public ResponseEntity<ProductDtoV2> pathVersioningV2(){
        return ResponseEntity.of(Optional.of(new ProductDtoV2( name: "ASUS Laptop", new BigDecimal( val: 7500))));
    }

    // ############### Param Versioning ###############
    // http://localhost:8080/api/product?apiVersion=1
    @GetMapping(value = "/product", params = "apiVersion=1")
    public ResponseEntity<ProductDtoV1> paramVersioningV1(){
```

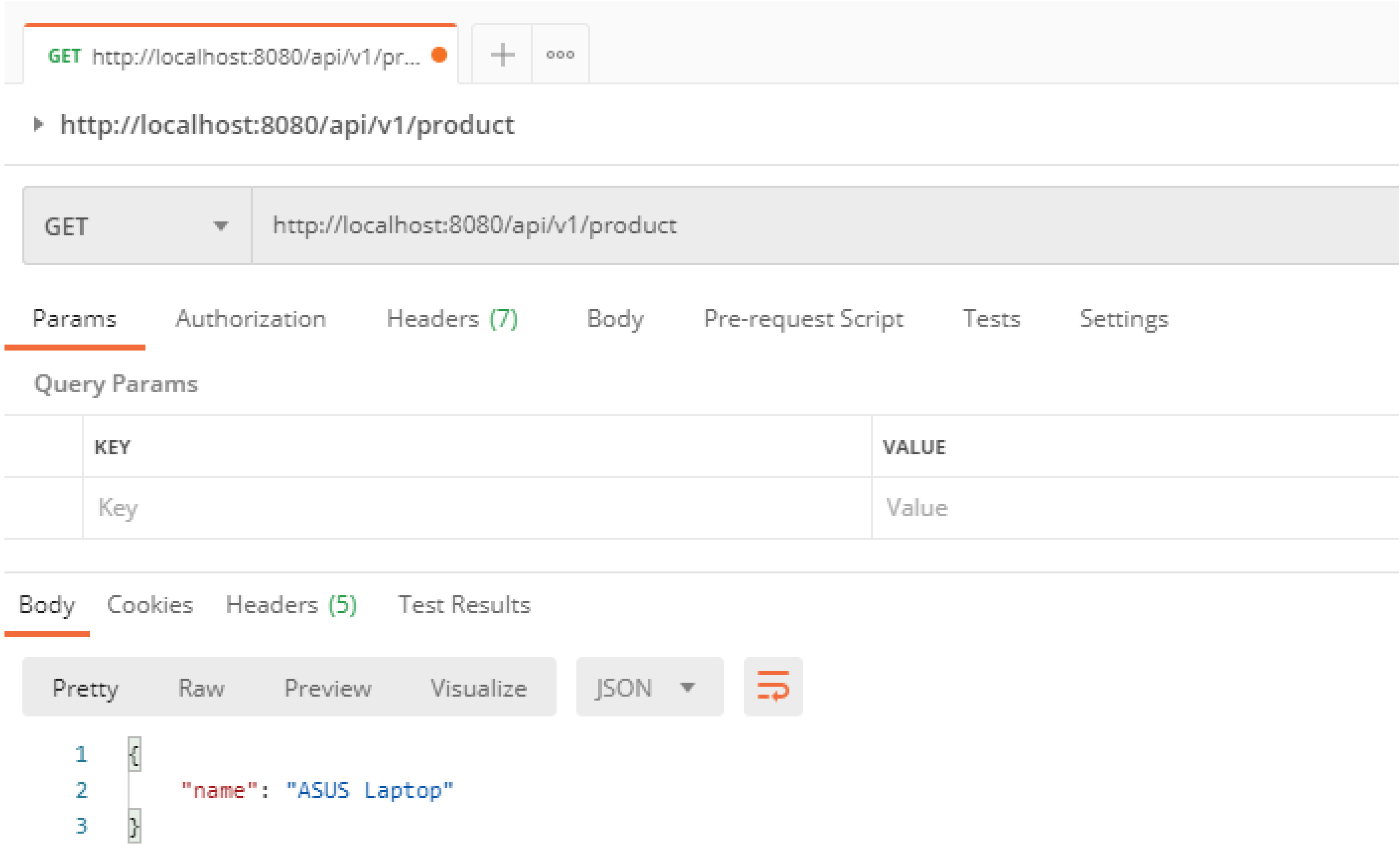
URI Versioning(Path Param)

http://localhost:8080/api/v1/product

http://localhost:8080/api/v2/product



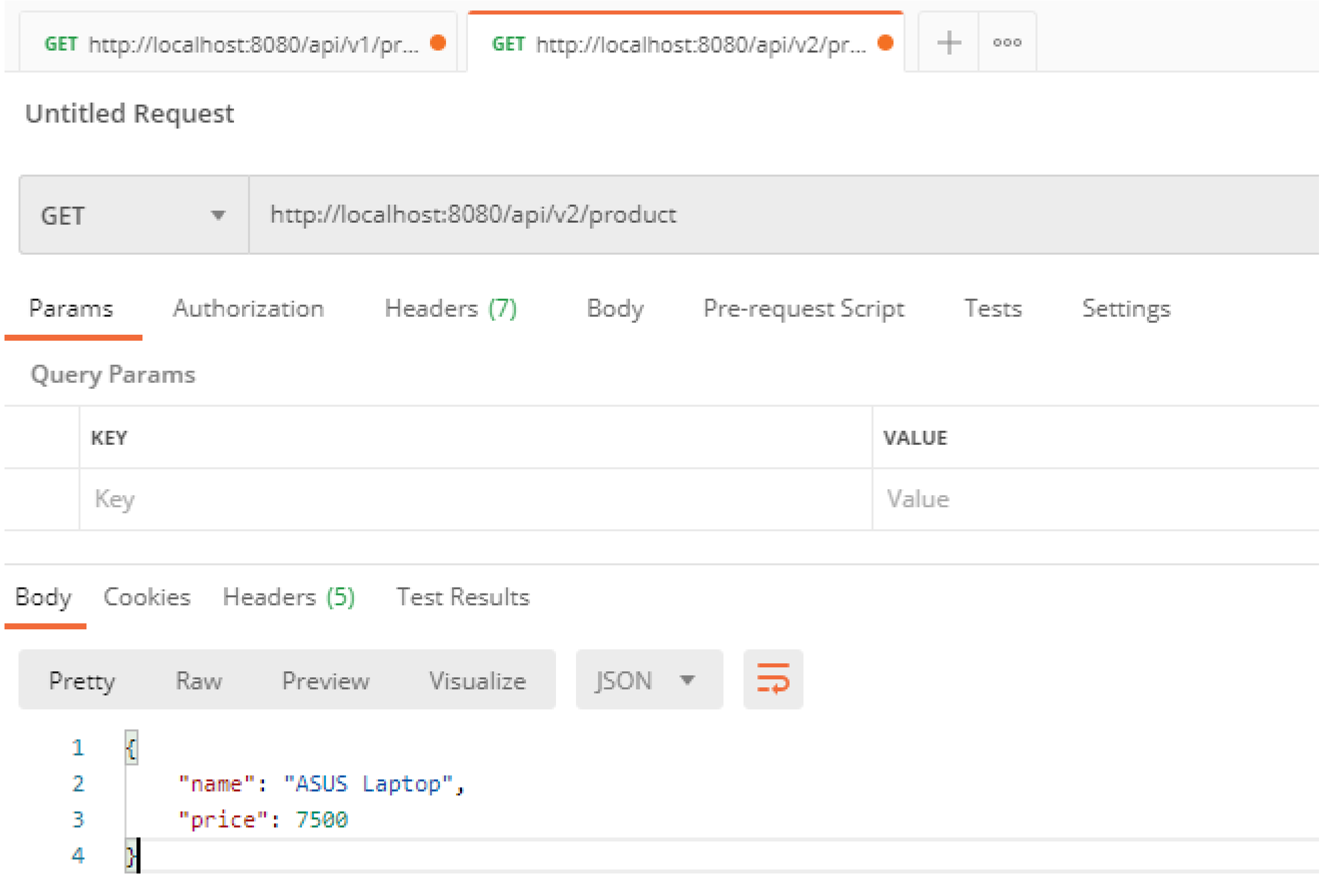
```
// ############### URI Versioning ###############
// http://localhost:8080/api/v1/product
@GetMapping(value = "/v1/product")
public ResponseEntity<ProductDtoV1> pathVersioningV1(){
    return ResponseEntity.of(Optional.of(new ProductDtoV1( name: "ASUS Laptop")));
}

// http://localhost:8080/api/v2/product
@GetMapping(value = "/v2/product")
public ResponseEntity<ProductDtoV2> pathVersioningV2(){
    return ResponseEntity.of(Optional.of(new ProductDtoV2( name: "ASUS Laptop", new BigDecimal( val: 7500))));
}
```



GET http://localhost:8080/api/v1/product

KEY	VALUE
Key	Value

```
{
    "name": "ASUS Laptop"
}
```



GET http://localhost:8080/api/v2/product

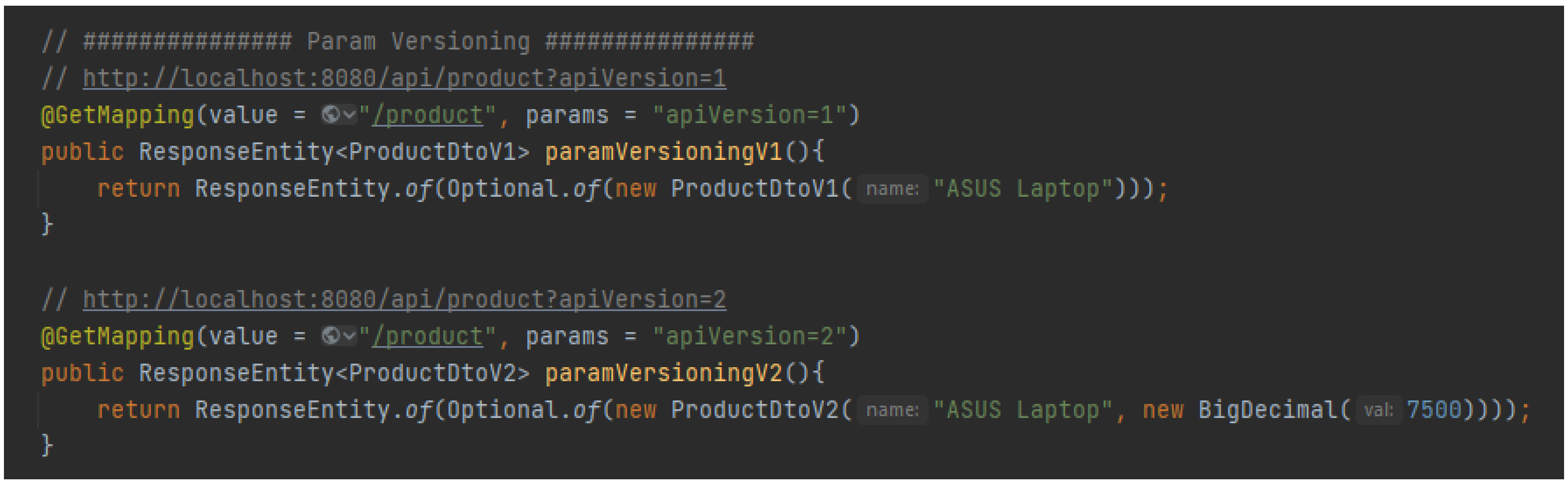
KEY	VALUE
Key	Value

```
{
    "name": "ASUS Laptop",
    "price": 7500
}
```

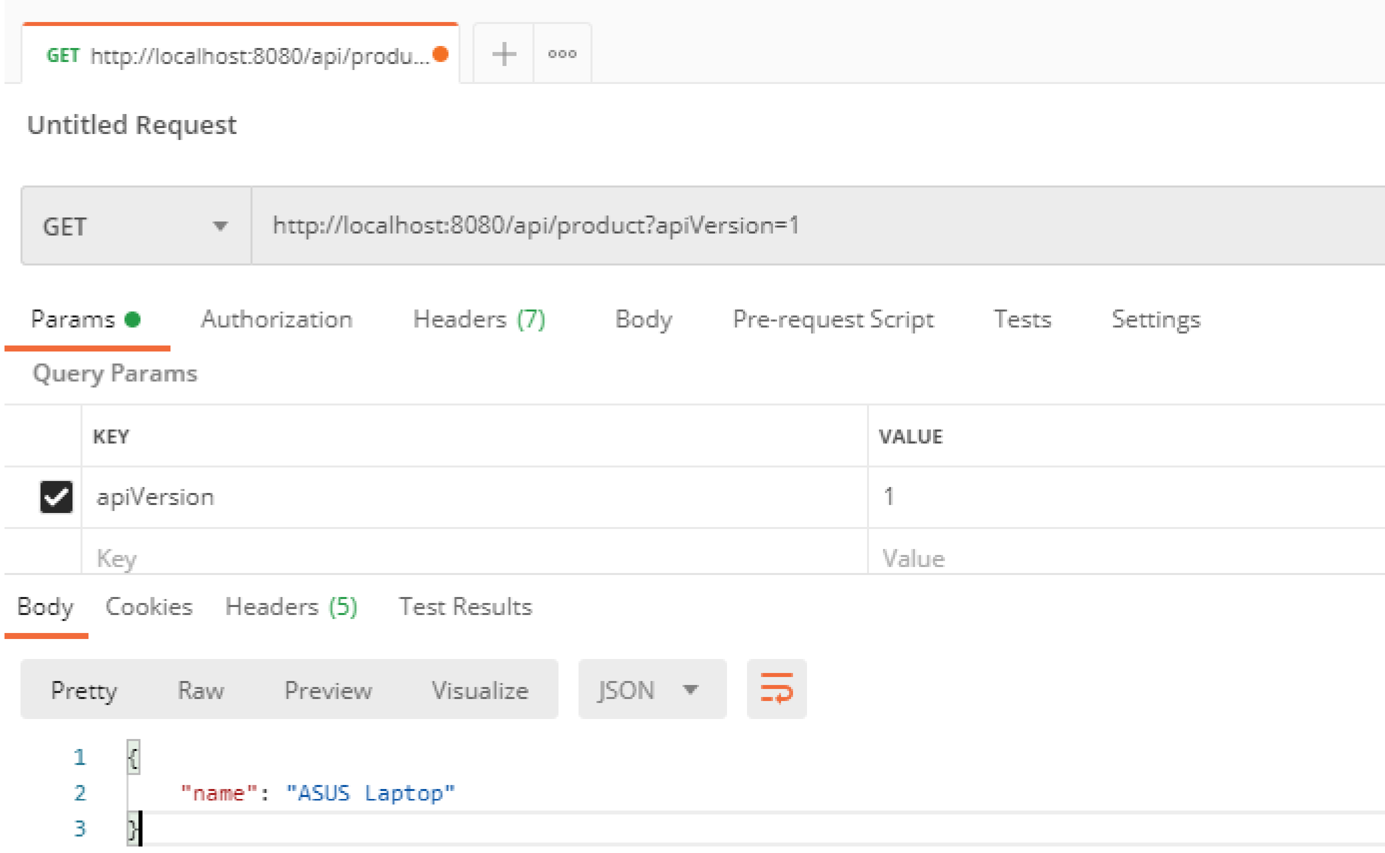
URI Versioning(Query Param)

http://localhost:8080/api/product?apiVersion=1

http://localhost:8080/api/product?apiVersion=2



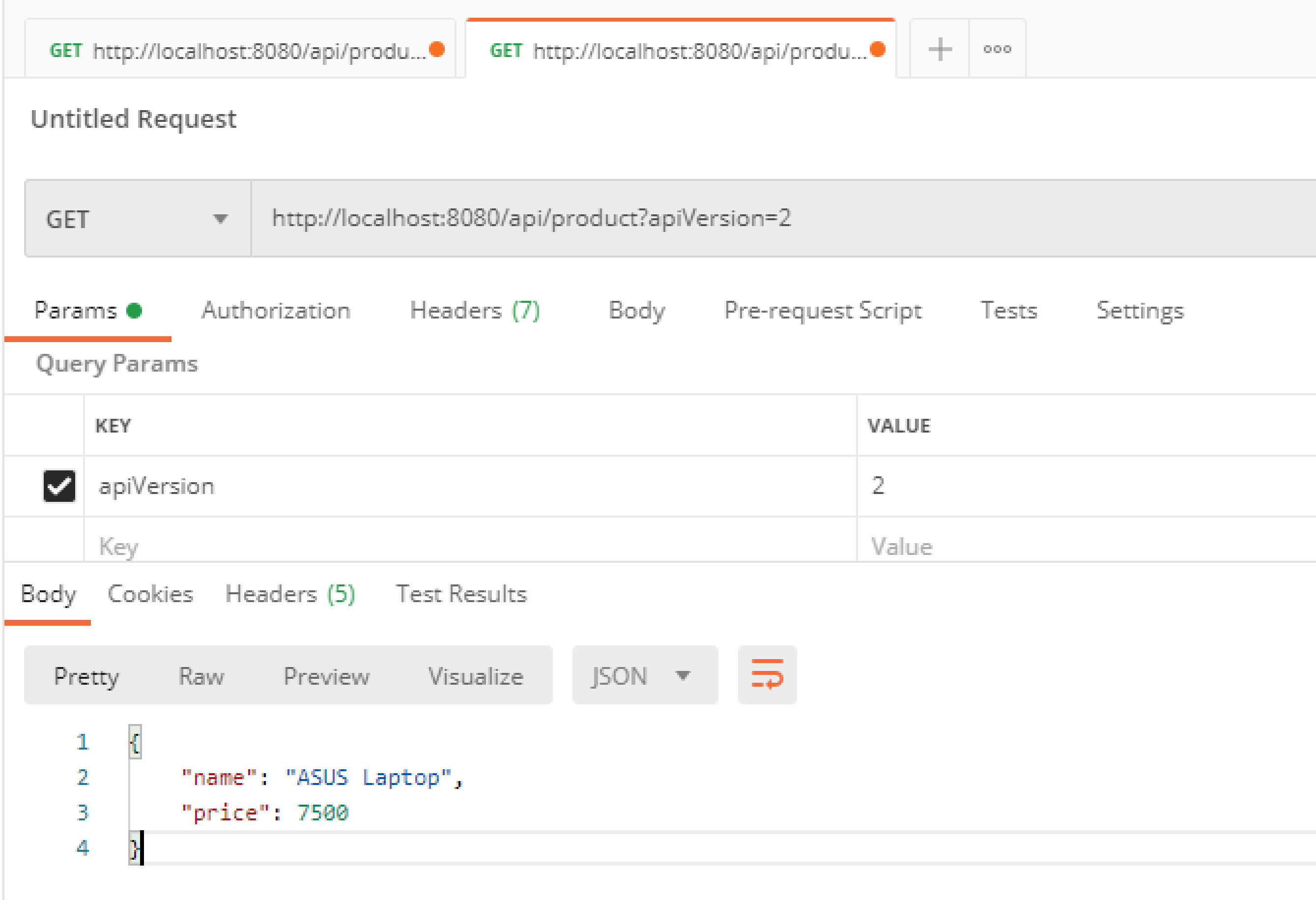
```java
// ############### Param Versioning ###############
// http://localhost:8080/api/product?apiVersion=1
@GetMapping(value = "/product", params = "apiVersion=1")
public ResponseEntity<ProductDtoV1> paramVersioningV1(){
    return ResponseEntity.of(Optional.of(new ProductDtoV1("ASUS Laptop")));
}

// http://localhost:8080/api/product?apiVersion=2
@GetMapping(value = "/product", params = "apiVersion=2")
public ResponseEntity<ProductDtoV2> paramVersioningV2(){
    return ResponseEntity.of(Optional.of(new ProductDtoV2("ASUS Laptop", new BigDecimal(7500))));
}
```



GET http://localhost:8080/api/product?apiVersion=1

KEY	VALUE
☑ apiVersion	1
Key	Value

```
{
    "name": "ASUS Laptop"
}
```

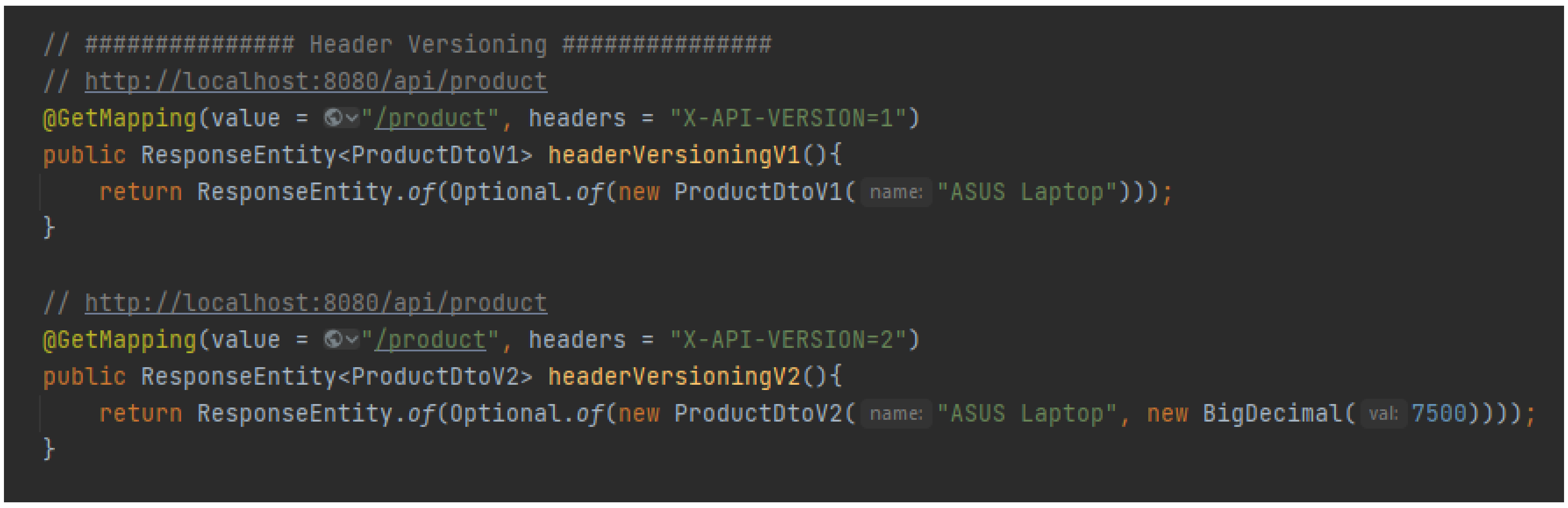


GET http://localhost:8080/api/product?apiVersion=2

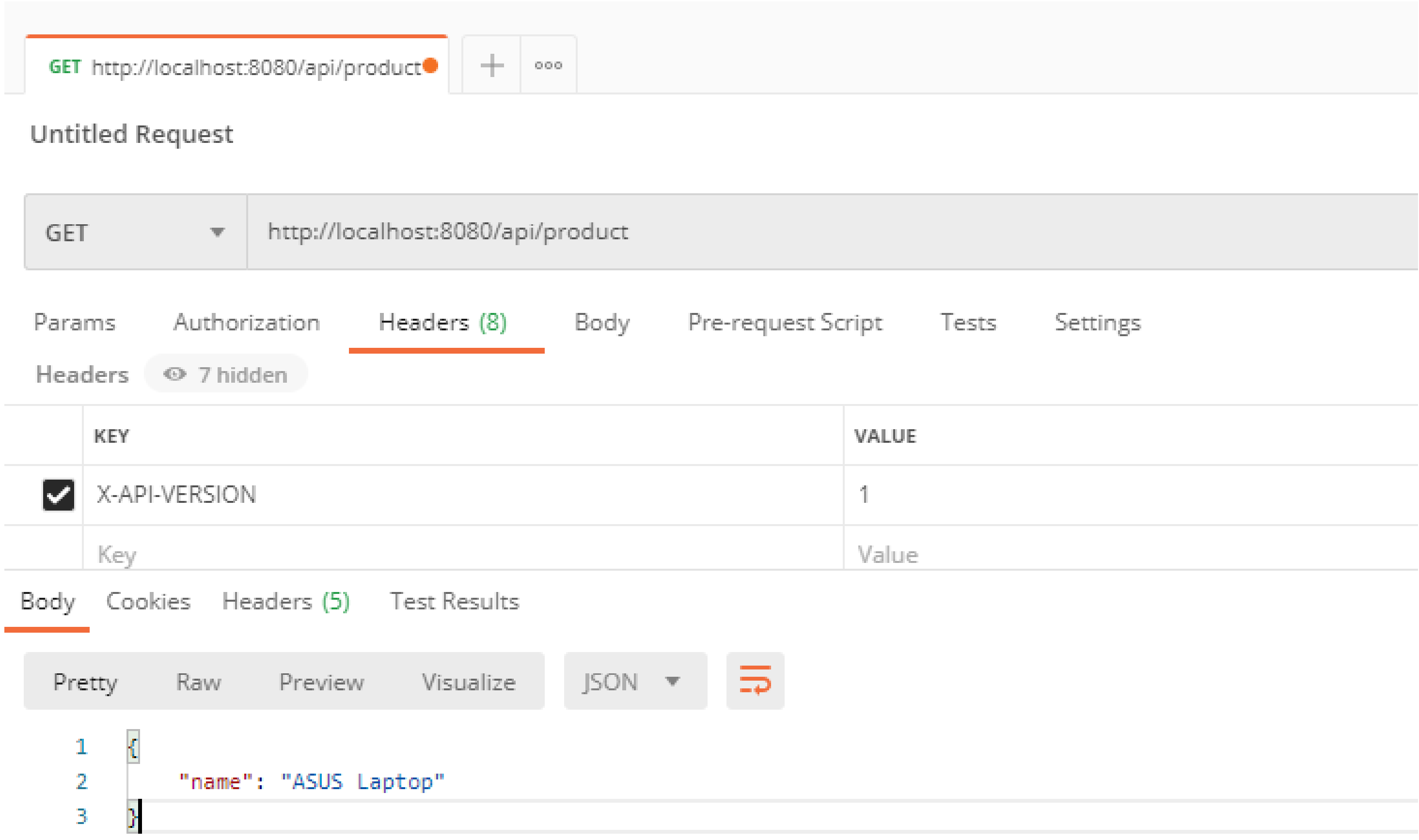
KEY	VALUE
☑ apiVersion	2
Key	Value

```
{
    "name": "ASUS Laptop",
    "price": 7500
}
```

Request Header Versioning



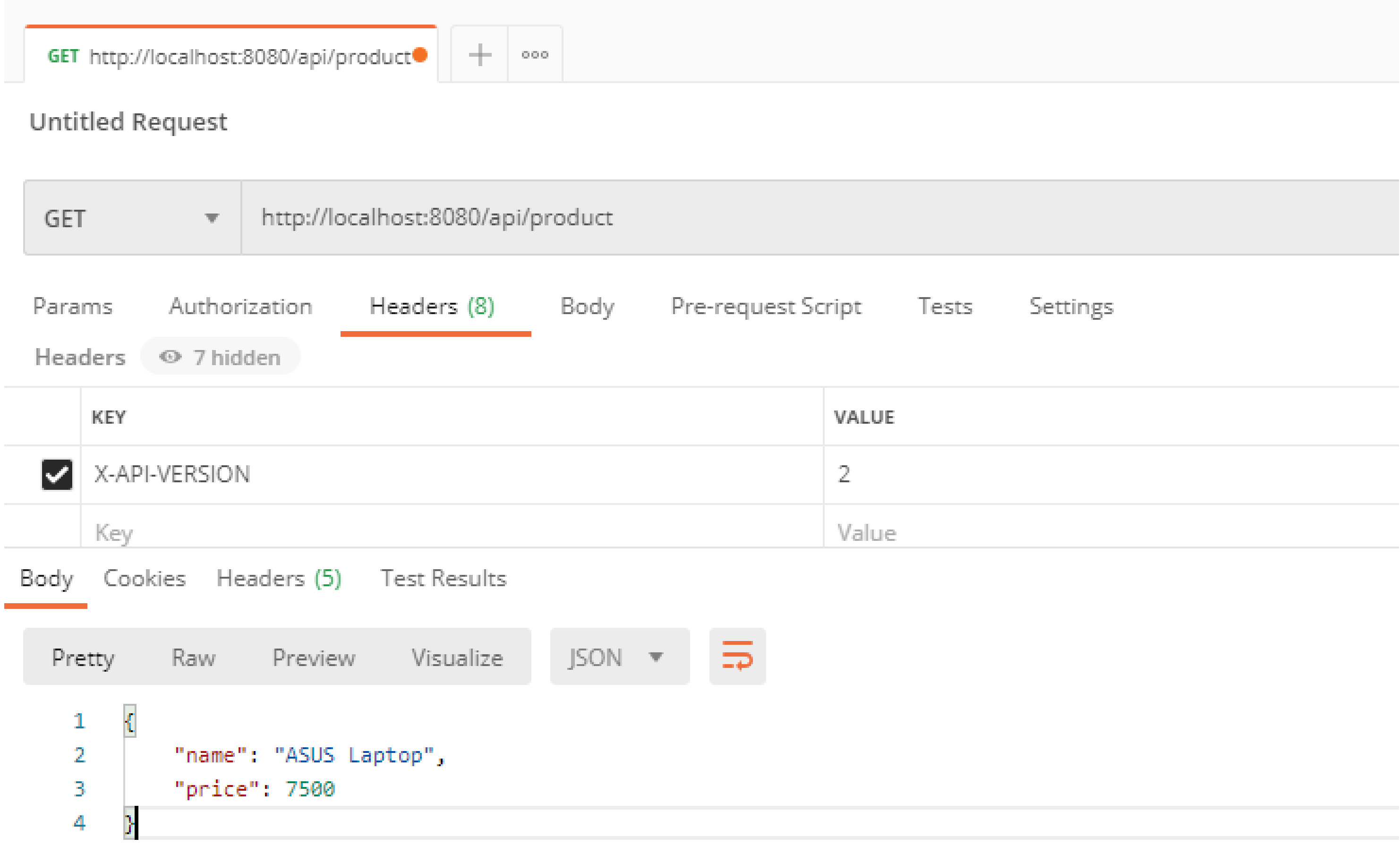
```java
// ############### Header Versioning ###############
// http://localhost:8080/api/product
@GetMapping(value = "/product", headers = "X-API-VERSION=1")
public ResponseEntity<ProductDtoV1> headerVersioningV1(){
    return ResponseEntity.of(Optional.of(new ProductDtoV1( name: "ASUS Laptop")));
}

// http://localhost:8080/api/product
@GetMapping(value = "/product", headers = "X-API-VERSION=2")
public ResponseEntity<ProductDtoV2> headerVersioningV2(){
    return ResponseEntity.of(Optional.of(new ProductDtoV2( name: "ASUS Laptop", new BigDecimal( val: 7500))));
}
```



GET http://localhost:8080/api/product

	KEY	VALUE
☑	X-API-VERSION	1
	Key	Value

```json
{
    "name": "ASUS Laptop"
}
```



GET http://localhost:8080/api/product

	KEY	VALUE
☑	X-API-VERSION	2
	Key	Value

```json
{
    "name": "ASUS Laptop",
    "price": 7500
}
```