

웹프레임워크2(N)

-과제2-

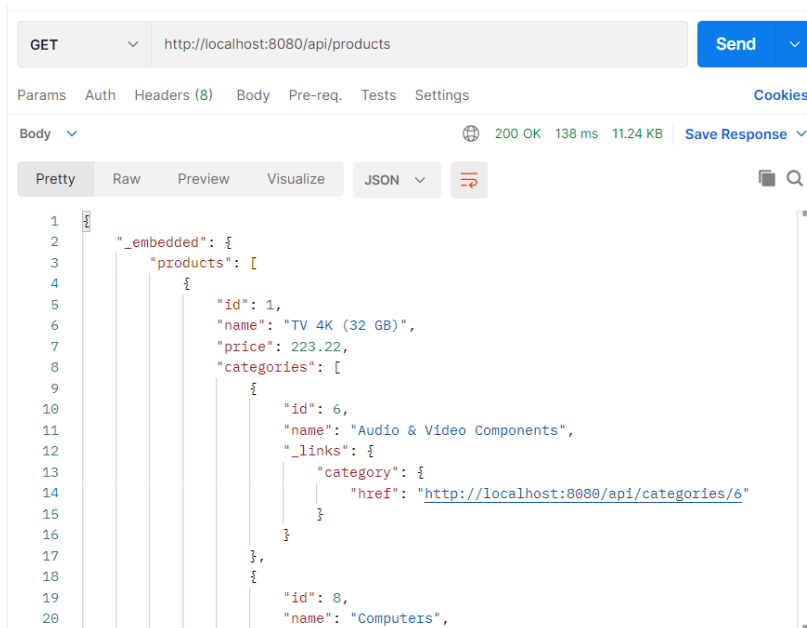
학번: 1771324

이름: 김수연

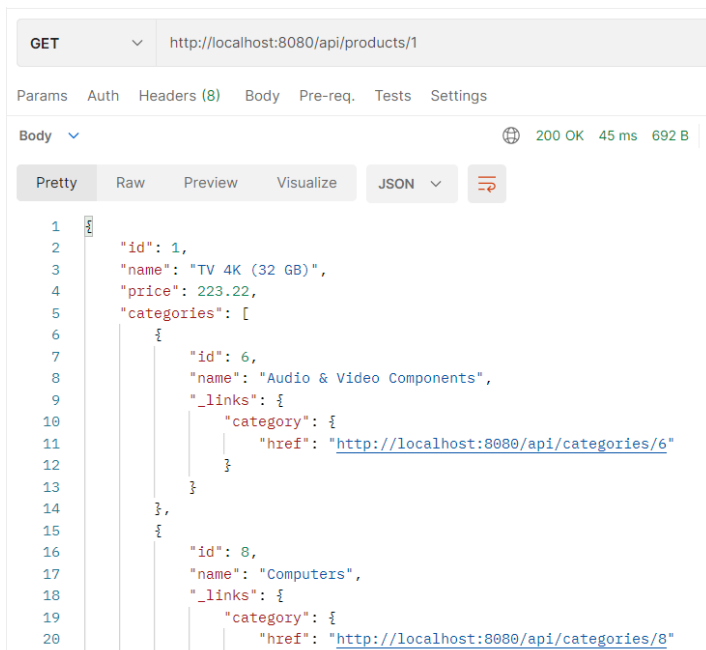
1) github의 readme.md 파일 내용 중에서 "6. 실습할 Rest API"에 따라 postman을 사용하여 웹 요청을 보내고 응답 결과를 보여라. (Products에 대한 요청 5개, Category에 대한 요청 10개)

Products:

1. Get, <http://localhost:8080/api/products>



2. Get, <http://localhost:8080/api/products/1>



3. Post, <http://localhost:8080/api/products>, id 100이 생성되는지 확인

{ "name": "삼성컴퓨터", "price": 150 } // body는 postman에서 raw, json format으로 전송

The image shows two screenshots of the Postman interface. The top screenshot shows a POST request to `http://localhost:8080/api/products` with a JSON body: `{ "name": "삼성컴퓨터", "price": 150 }`. The bottom screenshot shows the response, which is a 201 Created status with a JSON body: `{ "id": 100, "name": "삼성컴퓨터", "price": 150.0, "_links": { "self": { "href": "http://localhost:8080/api/products/100" } } }`.

POST Request:

```
1 {
2   "name": "삼성컴퓨터",
3   "price": 150
4 }
```

Response (201 Created):

```
1 {
2   "id": 100,
3   "name": "삼성컴퓨터",
4   "price": 150.0,
5   "_links": {
6     "self": {
7       "href": "http://localhost:8080/api/products/100"
8     }
9   }
10 }
```

GET Request:

```
821 {
822   },
823   {
824     "id": 100,
825     "name": "삼성컴퓨터",
826     "price": 150.0,
827     "_links": {
828       "self": {
829         "href": "http://localhost:8080/api/products/100"
830       }
831     }
832   }
833 ]
834 {,
```

4. Put, <http://localhost:8080/api/products/100>

{ "name": "LG컴퓨터", "price": 120 }

PUT <http://localhost:8080/api/products/100>

Params Auth Headers (10) **Body** Pre-req. Tests Settings

raw JSON

```
1 {
2   "name": "LG컴퓨터",
3   "price": 120
4 }
```

Body 200 OK 60 ms 46%

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 100,
3   "name": "LG컴퓨터",
4   "price": 120.0,
5   "_links": {
6     "self": {
7       "href": "http://localhost:8080/api/products/100"
8     }
9   }
10 }
```

GET <http://localhost:8080/api/products>

Params Auth Headers (10) **Body** Pre-req. Tests Settings

Body 200 OK 98 ms 11.35 KB

Pretty Raw Preview Visualize JSON

```
821 {
822   },
823   {
824     "id": 100,
825     "name": "LG컴퓨터",
826     "price": 120.0,
827     "_links": {
828       "self": {
829         "href": "http://localhost:8080/api/products/100"
830       }
831     }
832   }
833 ]
```

5. Delete, <http://localhost:8080/api/products/100>

DELETE

▼

http://localhost:8080/api/products/100

ParamsAuthHeaders (10)Body ●Pre-req.TestsSettings

Body ▼

204 No Content

PrettyRawPreviewVisualizeText ▼

1

Category:

1) category CRUD

- 1-1) Get, <http://localhost:8080/api/categories>

GET

▼

http://localhost:8080/api/categories

ParamsAuthHeaders (7)BodyPre-req.TestsSettings

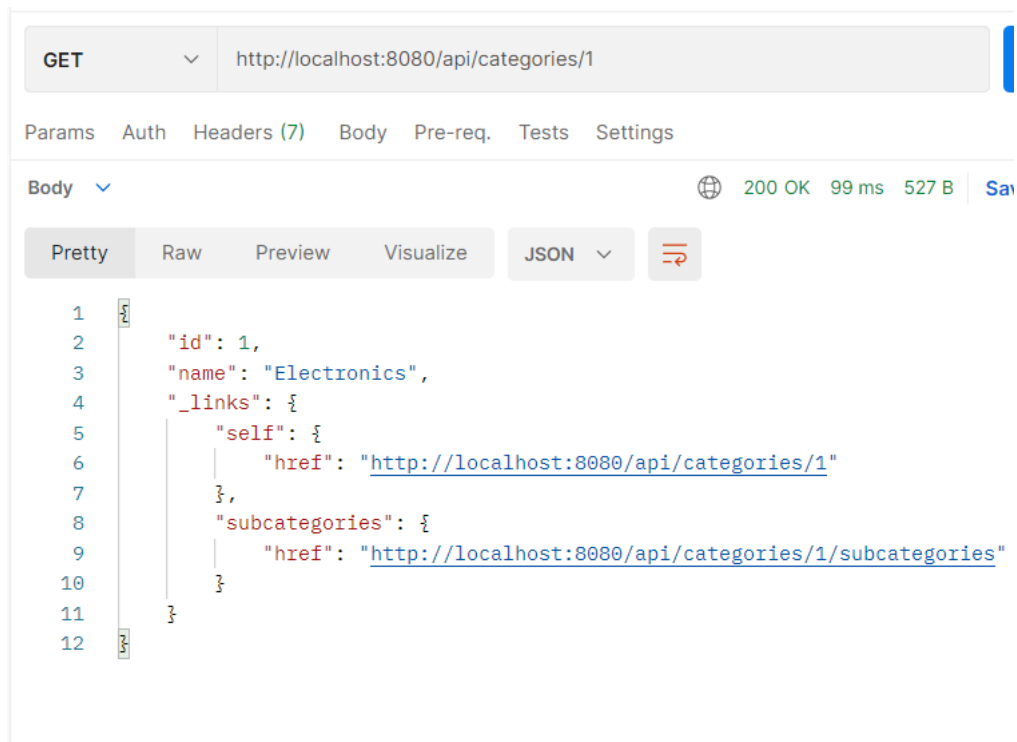
Body ▼

200 OK1445 ms10.26 KB

PrettyRawPreviewVisualizeJSON ▼

12
2 "_embedded": {
3 "categories": [
4 {
5 "id": 1,
6 "name": "Electronics",
7 "_links": {
8 "self": {
9 "href": "http://localhost:8080/api/categories/1"
10 },
11 "subcategories": {
12 "href": "http://localhost:8080/api/categories/1/
 subcategories"
13 }
14 }
15 },
16 {
17 "id": 2,
18 "name": "Clothing, Shoes & Accessories",
19 "_links": {

1-2) Get, <http://localhost:8080/api/categories/1>



GET <http://localhost:8080/api/categories/1>

Params Auth Headers (7) Body Pre-req. Tests Settings

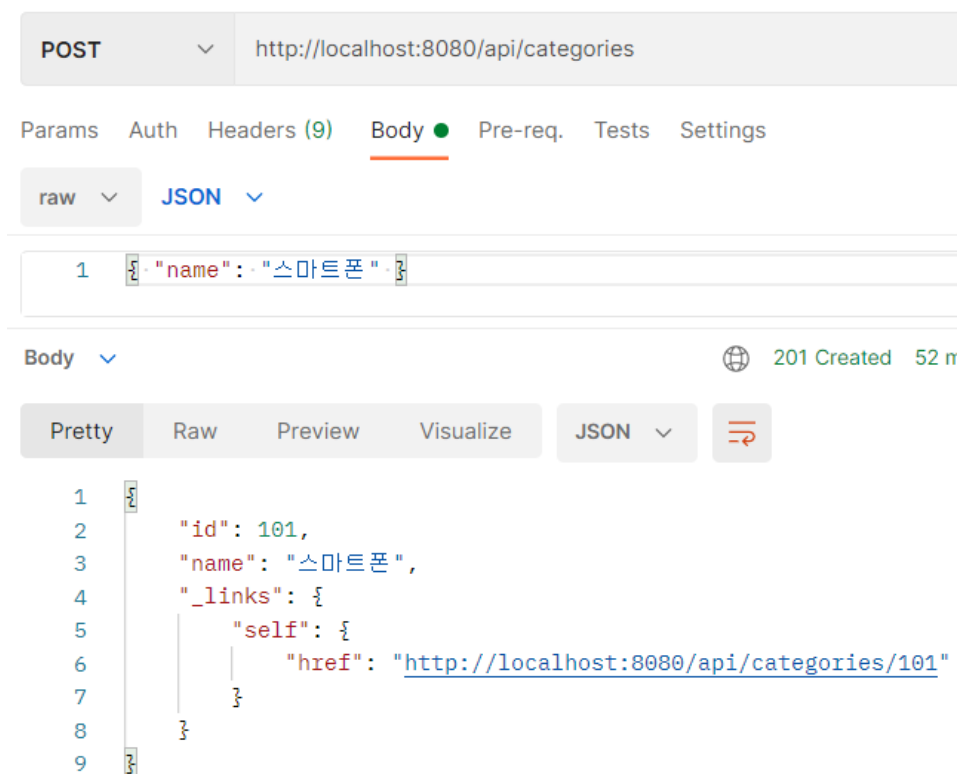
Body [v](#) [200 OK](#) [99 ms](#) [527 B](#) [Save](#)

Pretty Raw Preview Visualize JSON [v](#) [≡](#)

```
1 {
2   "id": 1,
3   "name": "Electronics",
4   "_links": {
5     "self": {
6       "href": "http://localhost:8080/api/categories/1"
7     },
8     "subcategories": {
9       "href": "http://localhost:8080/api/categories/1/subcategories"
10    }
11  }
12 }
```

1-3) Post, <http://localhost:8080/api/categories>, id 101 카테고리 생성됨

{ "name": "스마트폰" }



POST <http://localhost:8080/api/categories>

Params Auth Headers (9) Body [●](#) Pre-req. Tests Settings

raw [v](#) JSON [v](#)

```
1 { "name": "스마트폰" }
```

Body [v](#) [201 Created](#) [52 m](#)

Pretty Raw Preview Visualize JSON [v](#) [≡](#)

```
1 {
2   "id": 101,
3   "name": "스마트폰",
4   "_links": {
5     "self": {
6       "href": "http://localhost:8080/api/categories/101"
7     }
8   }
9 }
```

1-4) Put, <http://localhost:8080/api/categories/101>

{ "name": "태블릿" }

The screenshot shows a REST client interface. At the top, a PUT request is configured for the URL `http://localhost:8080/api/categories/101`. The 'Body' tab is selected, and the request body is set to raw JSON: `{ "name": "태블릿" }`. Below the request, the response is displayed in the 'Body' tab. The response status is `200 OK` with a response time of `42 ms`. The response body is a JSON object:

```
{
  "id": 101,
  "name": "태블릿",
  "_links": {
    "self": {
      "href": "http://localhost:8080/api/categories/101"
    }
  }
}
```

1-5) Delete, <http://localhost:8080/api/categories/101>

The screenshot shows a REST client interface. At the top, a DELETE request is configured for the URL `http://localhost:8080/api/categories/101`. Below the request, the response is displayed in the 'Body' tab. The response status is `204 No Content`. The response body is empty, indicated by the number `1` on the first line.

2) Add / Remove child categories

2-1) Get, <http://localhost:8080/api/categories/1/subcategories>

GET <http://localhost:8080/api/categories/1/subcategories>

Params Auth Headers (9) Body ● Pre-req. Tests Settings

Body ▾ 200 OK 60 ms 2.51 KB S

Pretty Raw Preview Visualize JSON ▾

```
1 {
2   "_embedded": {
3     "categories": [
4       {
5         "id": 6,
6         "name": "Audio & Video Components",
7         "products": [
8           {
9             "id": 1,
10            "name": "TV 4K (32 GB)",
11            "price": 223.22,
12            "_links": {
13              "product": {
14                "href": "http://localhost:8080/api/produ
15              }
16            }
17          },
18          {
19            "id": 2,
20            "name": "RCA ANT751 OUTDOOR ANTENNA OPTIMIZED FOR
```

2-2) subcategory를 생성한 후(id=102), category(id=1)에 연결한다

Post, <http://localhost:8080/api/categories>

{ "name": "스마트폰" }

POST <http://localhost:8080/api/categories>

Params Auth Headers (9) Body ● Pre-req. Tests Settings

raw ▾ JSON ▾

```
1 { "name": "스마트폰" }
```

Body ▾ 201 Created 56 ms


Pretty Raw Preview Visualize JSON ▾


```
1 {
2   "id": 102,
3   "name": "스마트폰",
4   "_links": {
5     "self": {
6       "href": "http://localhost:8080/api/categories/102"
7     }
8   }
9 }
```

Post, <http://localhost:8080/api/categories/1/subcategories/102>

POST <http://localhost:8080/api/categories/1/subcategories/102>

Params Auth Headers (8) Body Pre-req. Tests Settings


Body  201 Created 41 ms 532 B [Sa](#)


Pretty Raw Preview Visualize JSON 

```
1 {
2   "id": 1,
3   "name": "Electronics",
4   "_links": {
5     "self": {
6       "href": "http://localhost:8080/api/categories/1"
7     },
8     "subcategories": {
9       "href": "http://localhost:8080/api/categories/1/subcategories"
10    }
11  }
12 }
```

GET <http://localhost:8080/api/categories/1/subcategories>

Params Auth Headers (7) Body Pre-req. Tests Settings

Body  200 OK 73 ms 2.67 KB [Save](#)


Pretty Raw Preview Visualize JSON 


```
141 {
142   "parent": {
143     "href": "http://localhost:8080/api/categories/1"
144   }
145 },
146 {
147   "id": 102,
148   "name": "스마트폰",
149   "_links": {
150     "self": {
151       "href": "http://localhost:8080/api/categories/102"
152     },
153     "parent": {
154       "href": "http://localhost:8080/api/categories/1"
155     }
156   }
157 }
158 ]
159 },
160 "_links": {
161   "self": {
```

2-3) Delete, <http://localhost:8080/api/categories/1/subcategories/102>

DELETE <http://localhost:8080/api/categories/1/subcategories/102>

Params Auth Headers (7) Body Pre-req. Tests Settings

Body  204 No Content

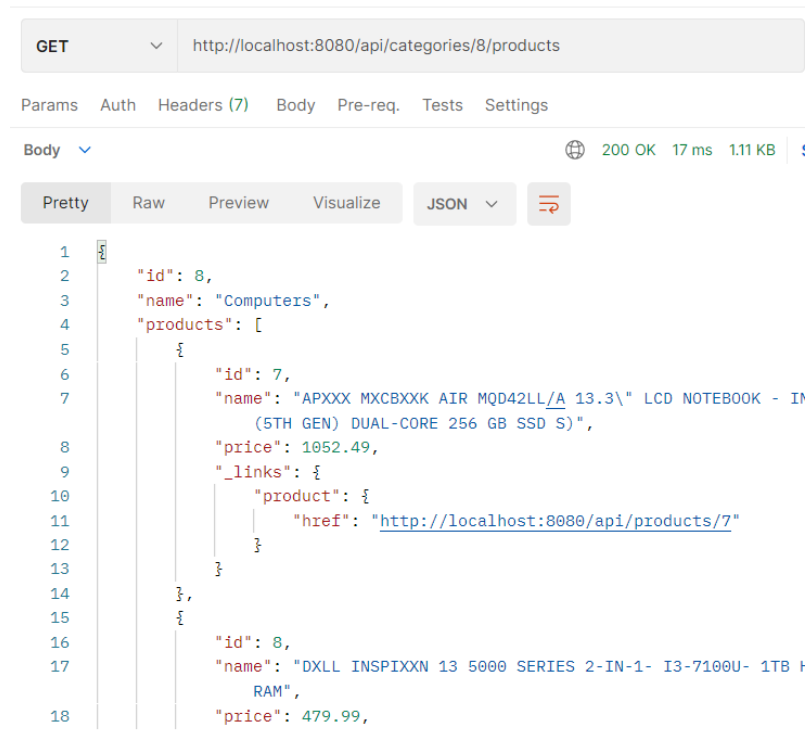
Pretty Raw Preview Visualize Text 

1

3) Link / Unlink products

3-1) Get, <http://localhost:8080/api/categories/8/products>

먼저 "Computer" 카테고리에 존재하는 Product조회한다



```
GET http://localhost:8080/api/categories/8/products

Params Auth Headers (7) Body Pre-req. Tests Settings

Body 200 OK 17 ms 1.11 KB

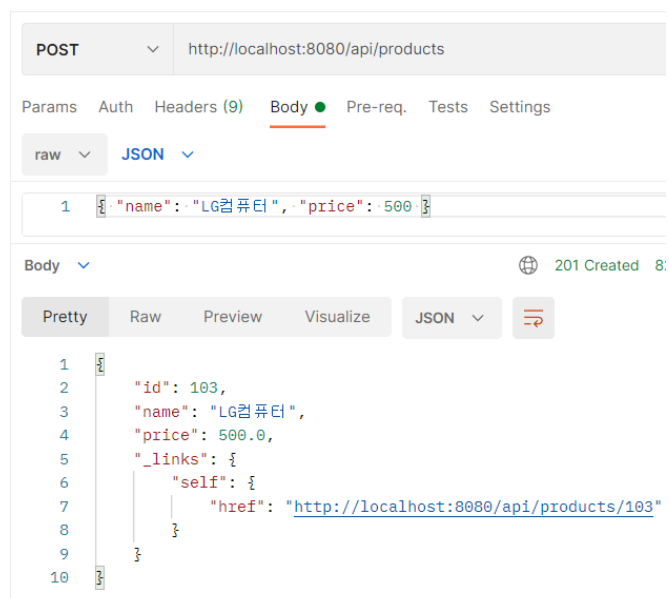
Pretty Raw Preview Visualize JSON

1 {
2   "id": 8,
3   "name": "Computers",
4   "products": [
5     {
6       "id": 7,
7       "name": "APXXX MXCBXXX AIR MQD42LL/A 13.3\" LCD NOTEBOOK - IN
8         (5TH GEN) DUAL-CORE 256 GB SSD S)",
9       "price": 1052.49,
10      "_links": {
11        "product": {
12          "href": "http://localhost:8080/api/products/7"
13        }
14      },
15    },
16    {
17      "id": 8,
18      "name": "DXLL INSPIXXN 13 5000 SERIES 2-IN-1- I3-7100U- 1TB +
19        RAM",
20      "price": 479.99,
```

3-2) Product(id=103)를 생성한 후, "Computer" 카테고리에 연결한다(Link)

Post, , <http://localhost:8080/api/products>

{ "name": "LG컴퓨터", "price": 500 }



```
POST http://localhost:8080/api/products

Params Auth Headers (9) Body Pre-req. Tests Settings

raw JSON

1 { "name": "LG컴퓨터", "price": 500 }

Body 201 Created 8:


Pretty Raw Preview Visualize JSON


1 {
2   "id": 103,
3   "name": "LG컴퓨터",
4   "price": 500.0,
5   "_links": {
6     "self": {
7       "href": "http://localhost:8080/api/products/103"
8     }
9   }
10 }
```

Post, <http://localhost:8080/api/categories/8/products/103>

POST <http://localhost:8080/api/categories/8/products/103>

Params Auth Headers (8) Body Pre-req. Tests Settings


Body  201 Created 56 ms 583 B


Pretty Raw Preview Visualize JSON 

```
1  {
2    "id": 103,
3    "name": "LG컴퓨터",
4    "price": 500.0,
5    "categories": [
6      {
7        "id": 8,
8        "name": "Computers",
9        "_links": {
10         "category": {
11           "href": "http://localhost:8080/api/categories/8"
12         }
13       }
14     ],
15     "_links": {
16       "self": {
17         "href": "http://localhost:8080/api/products/103"
18       }
19     }
20   }
```

GET <http://localhost:8080/api/categories/8/products>

Params Auth Headers (7) Body Pre-req. Tests Settings

Body  200 OK 32 ms 1.22 KB

Pretty Raw Preview Visualize JSON 

```
42   {
43     }
44   },
45   {
46     "id": 103,
47     "name": "LG컴퓨터",
48     "price": 500.0,
49     "_links": {
50       "product": {
51         "href": "http://localhost:8080/api/products/103"
52       }
53     }
54   }
55 ],
56 "_links": {
57   "self": {
58     "href": "http://localhost:8080/api/categories/8"
59   }
60 }
61 }
```

2) charlie 계정(권한은 admin, 패스워드는 charliepw 로 설정)을 data.sql에 저장한다. 이 때 패스워드는 BCryptPasswordEncoder를 이용하여 반드시 해쉬 값을 저장해야 한다. 그리고 "/api/products", HTTP GET method를 호출하였을 때 로그인 화면이 보일 수 있도록 WebSecurityConfig.java 파일을 수정하라. 그리고 브라우저에서 로그인이 성공적으로 이루어짐을 보여라.

```
@SpringBootTest
class EcommerceSpringRestApplicationTests {

    @Autowired
    private PasswordEncoder encoder;

    @Test
    void generateHashedPassword() {
        String pwd = encoder.encode(charSequence: "charliepw");
        System.out.println(pwd);
    }
}
```

```
INSERT INTO users (id, email, password) VALUES
(1, 'alice@hansung.ac.kr', '$2a$10$KNV8XoSu1uGkcgQwMImd/03HIES5L/Fx5XFxFkUb/5b5pMW0DxT66'),
(2, 'bob@hansung.ac.kr', '$2a$10$QuYYcgaP6LFfV53Qssvtp.hcj1BJVmQz07pCValc27jm4SsI25Lf2'),
(3, 'trudy@hansung.ac.kr', '$2a$10$UXt0RhsuLW0fhWxhcDEhX09CUQkIWPxRrGMJYQhMW/5TBd7rYbsFq'),
(4, 'charlie@hansung.ac.kr', '$2a$10$LYJ0rQRWkU8sVFKcUdA26u0h1hk20jvS2oLnLiMxFtCELUyP94qHy');

insert into user_role(user_id, role_id) values (1,1), (1,2), (2,2), (3,2), (4,1);
```

```
};

@Override
protected void configure(HttpSecurity http) throws Exception {
    http
        .csrf().disable()
        .authorizeRequests()
            .antMatchers(PUBLIC_MATCHERS).permitAll()
            .antMatchers(...antPatterns: "/", "/home", "/signup").permitAll()
            .antMatchers(...antPatterns: "/admin/**").hasRole("ADMIN")
            .antMatchers(...antPatterns: "/api/products").authenticated()
            .anyRequest().authenticated()
        .and()
}
```

```
GET http://localhost:8080/api/products

Params Auth Headers (8) Body Pre-req. Tests Settings

Body 200 OK 32 ms 1.85 KB

Pretty Raw Preview Visualize HTML

+IOBjI7aAzPcEZeedi5yW5i2yOq55WWLNGmvvx4Um1vskeMj0"
crossorigin="anonymous">
10 <link href="https://getbootstrap.com/docs/4.0/examples/signin/signin
    rel="stylesheet"
11     crossorigin="anonymous" />
12
13 <title>Please sign in</title>
14 </head>
15
16 <body>
17   <div class="container">
18     <form class="form-signin" method="post" action="/login">
19       <h2 class="form-signin-heading">Please sign in</h2>
20
21
22
23
24       <p>
25         <label for="username" class="sr-only">Username</label>
26         <input type="email" id="username" name="username"
```

localhost:8080/login

VER 한성 e-class 한성대학교 종합정... 직업훈련포털 HRD... GitHub

Please sign in

Username

charlie@hansung.ac.kr

Password

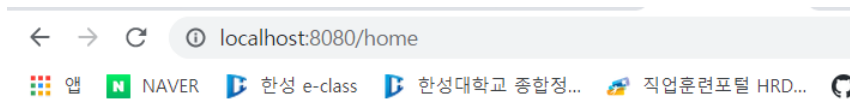
.....

Sign in

localhost:8080/api/products

NAVER 한성 e-class 한성대학교 종합정... 직업훈련포털 HRD... GitHub

```
{
  "_embedded": {
    "products": [
      {
        "id": 1,
        "name": "TV 4K (32 GB)",
        "price": 223.22,
        "categories": [
          {
            "id": 6,
            "name": "A Components",
            "_links": {
              "category": {
                "href": "http://localhost:8080/api/categories/6"
              }
            }
          },
          {
            "id": 8,
            "name": "Computers",
            "_links": {
              "category": {
                "href": "http://localhost:8080/api/categories/8"
              }
            }
          },
          {
            "id": 2,
            "name": "RCA ANT751 OUTDOOR ANTENNA OPTIMIZED RECEPTION - UPTO 3.3 FT",
            "price": 49.22,
            "categories": [
              {
                "id": 6,
                "name": "Audio & Video Components",
                "_links": {
                  "self": {
                    "href": "http://localhost:8080/api/categories/6"
                  }
                }
              },
              {
                "id": 3,
                "name": "APC REPLACEMENT BATTERY NO 24",
                "price": 299.0,
                "categories": [
                  {
                    "id": 6,
                    "name": "Audio & Video Components",
                    "_links": {
                      "category": {
                        "href": "http://localhost:8080/api/categories/6"
                      }
                    }
                  },
                  {
                    "id": 4,
                    "name": "CAXXX EOS REBEL T6 DSLR 18-55MM - 11590003",
                    "price": 379.95,
                    "categories": [
                      {
                        "id": 7,
                        "name": "Camera & Photo",
                        "_links": {
                          "category": {
                            "href": "http://localhost:8080/api/categories/7"
                          }
                        }
                      },
                      {
                        "id": 5,
                        "name": "RCA ANT751 OUTDOOR ANTENNA OPTIMIZED RECEPTION - UPTO 3.3 FT",
                        "price": 49.22,
                        "categories": [
                          {
                            "id": 7,
                            "name": "Camera & Photo",
                            "_links": {
                              "category": {
                                "href": "http://localhost:8080/api/categories/7"
                              }
                            }
                          },
                          {
                            "id": 6,
                            "name": "BEST CHOICE PRODUCTS 6-AXIS HEADLESS EDV DR NONE III/ IIIET LID CAMERA",
                            "price": 42.99,
                            "categories": [
                              {
                                "id": 7,
                                "name": "Camera & Photo",
                                "_links": {
                                  "category": {
                                    "href": "http://localhost:8080/api/categories/7"
                                  }
                                }
                              }
                            ]
                          }
                        ]
                      }
                    ]
                  }
                ]
              }
            ]
          }
        ]
      }
    ]
  }
}
```



안녕하세요. 스프링 부트 보안 사이트에 오신 것을 환영합니다.

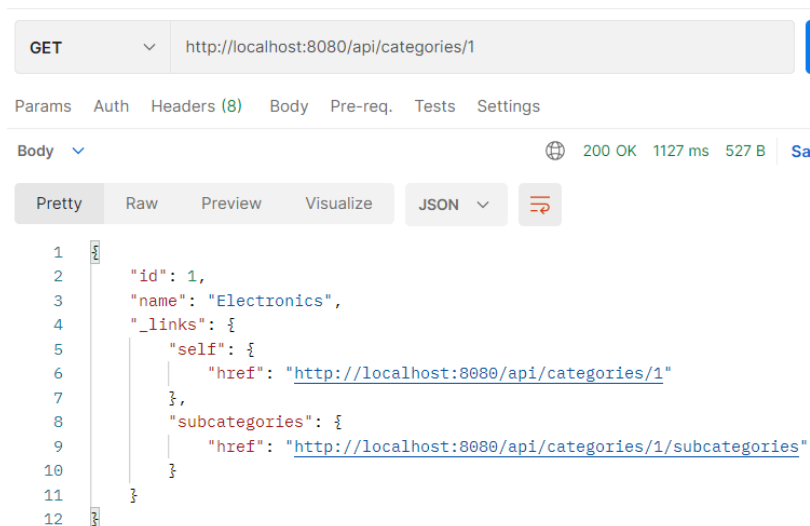
Welcome charlie@hansung.ac.kr

[Logout](#)

You will see this only if you are ADMIN

[Admin Home](#)

3) `/api/categories/1`, HTTP Get 메소드를 보냈을 경우, 응답 메시지(hypermedia)를 보여라. 그리고 응답 메시지 결과와 관련된 프로그램 소스를 지적하고 설명하라.



응답 메시지는 결과와 관련된 소스는 아래와 같다.

```
return categoryService.getCategoryById(id)
    .map(categoryModelAssembler::toModel)
    .map(ResponseEntity::ok)
    .orElse(ResponseEntity.notFound().build());
```

이 코드에서 `categoryModelAssembler::toModel`는 `categoryModelAssembler`의 `toModel` 메소드를 사용하는 것이다. `toModel`에서 Entity를 DTO로 바꾸어주는데 여기에는 setter를 사용해 `id`, `name`과 같은 데이터 넣어줄 뿐만 아니라 아래의 코드를 사용해서 연관된 링크도 포함시킨다.

```
categoryModel.add( linkTo(methodOn(CategoryController.class)
    .retrieveCategory(entity.getId()))
    .withSelfRel() );
```

`withRel("parent")`와 `withRel("subcategories")`코드를 통해 링크 이름을 설정하고 `parent` 링크와 `subcategories` 링크도 포함하고 있다.

4) category에서 특정 product를 제거하는 코드를 작성하라. 즉, "/api/categories/{category_id}/products/{product_id}, DELETE method를 요청하였을 때 product가 제거되는 코드를(CategoryProductsController.java) 작성하고 결과를 보여라. 응답 상태 코드(status code)는 204 No Content. 로 보내도록 설정한다.

```
CategoryProductsController.java
87 @RequestMapping(path = "/{productid}", method = RequestMethod.DELETE)
88 public ResponseEntity<?> removeProduct(@PathVariable Long categoryid, @PathVariable Long productid) {
89     final Category category = categoryService.getCategoryById(categoryid)
90         .orElseThrow(() -> new NotFoundException(categoryid));
91
92     // Getting the requiring product; or throwing exception if not found
93     final Product product = productService.getProductById(productid)
94         .orElseThrow(() -> new NotFoundException(categoryid));
95
96     if (!productService.hasCategory(product, category)) {
97         throw new IllegalArgumentException(
98             "product " + product.getId() + " does not contain category " + category.getId());
99     }
100     productService.removeCategory(product, category);
101     return ResponseEntity.noContent().build();
102 }
103
```

DELETE

http://localhost:8080/api/categories/8/products/103

Params Auth Headers (8) Body Pre-req. Tests Settings

Body

204 No Content

Pretty

Raw

Preview

Visualize

Text

