

# Chapter 4-1 CRUD

서비스의 자원

- 자원과 자원사이의 관계
- 사용자가 원하는 자원의 관리

Create, Read, Update, Delete

- 어떤 자원을 사용자에게 제공할 때 사용할 수 있는 4가지 기능 !

CRUD 실습

post package 생성

PostDto.java file 생성

→ Dot : data transfer object (데이터를 주고 받는데 사용하는 객체)

```
package dev.aquashdw.crud.post;

public class PostDto {
    private String title;
    private String content;
    private String writer;

    public PostDto() {
    }

    public PostDto(String title, String content, String writer) {
        this.title = title;
        this.content = content;
        this.writer = writer;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getContent() {
        return content;
    }

    public void setContent(String content) {
        this.content = content;
    }
}
```

```

    public String getWriter() {
        return writer;
    }

    public void setWriter(String writer) {
        this.writer = writer;
    }

    @Override
    public String toString() {
        return "PostDto{" +
            "title='" + title + '\'' +
            ", content='" + content + '\'' +
            ", writer='" + writer + '\'' +
            '}';
    }
}

```

## PostController.java 생성

```

@Controller
@ResponseBody
@RequestMapping("post")
public class PostController {
    private static final Logger logger = LoggerFactory.getLogger(PostController.class);

}

```

Controller, ResponseBody를 해놓으면 RestController가 Controller에 있는 모든 함수에 ResponseBody를 넣는 것과 같다. 여기서도 마찬가지로 Controller, ResponseBody를 해놓으면 이 클래스 안에 있는 모든 함수들이 ResponseBody가 붙어있는 형태로 함수 선언이 된다.

RequestMapping을 클래스에 붙이면 클래스 내용물에 있는 RequestMapping에 따라 앞쪽에 모두 post가 붙는다.

```

private final List<PostDto> postList; - 인터페이스

public PostController() {
    postList = new ArrayList<>(); - 구현체
}

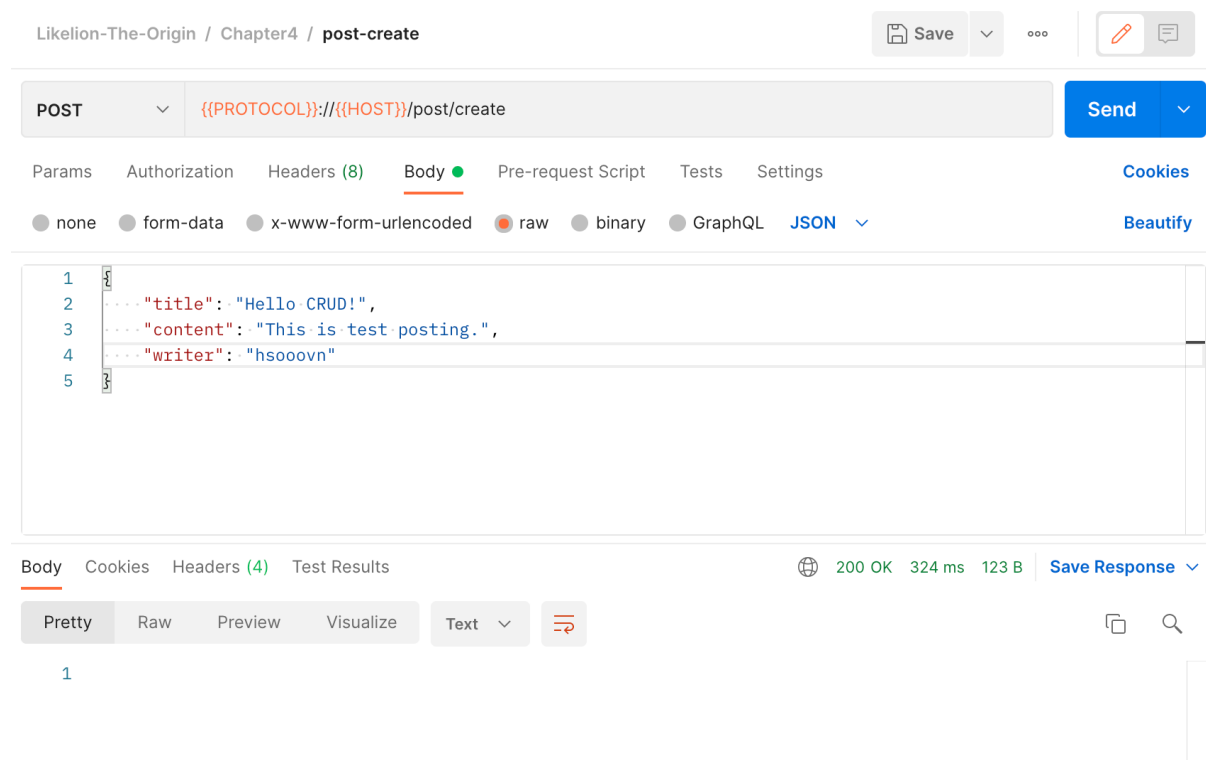
```

```
@PostMapping("create")
public void createPost(@RequestBody PostDto postDto){
    this.postList.add(postDto);
}
```

새로 생성된 리스트에 요청보내진 postDto가 들어가게 된다.

실행 !

PostMan



```
] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring DispatcherServlet 'dispatcherServlet'
] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
] o.s.web.servlet.DispatcherServlet        : Completed initialization in 13 ms
```

servlet이 실행되면서 정상적으로 작동하는 것을 확인할 수 있음.

아래의 코드를 추가해 내용 확인

```
logger.info(postDto.toString());
```

```

cat.[localhost].[/]      : Initializing Spring DispatcherServlet 'dispatcherServlet'
t.DispatcherServlet      : Initializing Servlet 'dispatcherServlet'
t.DispatcherServlet      : Completed initialization in 3 ms
rud.post.PostController  : PostDto{title='Hello CRUD!', content='This is test posting.', writer='hsooovn'}

```

\*사용자에게 제공하는 - PostMapping

\*사용자가 조회하는 - GetMapping

추가

```

@GetMapping("read-all")
public List<PostDto> readPostAll(){
    logger.info("in read all");
    return this.postList;
}

@GetMapping("read-one")
public PostDto readPostOne(@RequestParam("id") int id){
    logger.info("in read one");
    return this.postList.get(id);
}

@PostMapping("update")
public void updatePost(
    @RequestParam("id") int id,
    @RequestBody PostDto postDto){
    PostDto targetPost = this.postList.get(id);
    if (postDto.getTitle() != null){
        targetPost.setTitle(postDto.getTitle());
    }
    if (postDto.getContent() != null){
        targetPost.setContent(postDto.getContent());
    }
    this.postList.set(id, targetPost);
}

@DeleteMapping("delete")
public void deletePost(@RequestParam("id") int id){
    this.postList.remove(id);
}

```

update 함수

RequestParam - URL에서 쿼리 파라미터를 말하고,

RequestBody - post요청의 바디를 말하고 있음

id : 리스트의 인덱스 값을 그대로 사용

`targetPost` : 업데이트를 하기 위한 목적 게시글

If문 두 번 : 게시자는 바꿀 수 없다는 것을 가정하고 !

→ `targetPost` 내용 대신, `PostDto`를 다시 사용한다면 `title`과 `content`가 없으면 `targetPost`의 내용도 바뀌기 때문에

## Postman

Likelion-The-Origin / Chapter4 / post-read

GET `{{PROTOCOL}}://{{HOST}}/post/read-one?id=0` Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	id	0			
	Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 54 ms 240 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "title": "Hello CRUD!",
3   "content": "This is test posting.",
4   "writer": "hsooovn"
5 }
```

GET {{PROTOCOL}}://{{HOST}}/post/read-all

Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (5) Test Results

 200 OK 42 ms 242 B Save Response

Pretty

Raw

Preview

Visualize

JSON

v



```
1 [
2   {
3     "title": "Hello CRUD!",
4     "content": "This is test posting.",
5     "writer": "hsooovn"
6   }
7 ]
```

POST {{PROTOCOL}}://{{HOST}}/post/create

Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

Beautify

```
1 {
2   "title": "Second Post",
3   "content": "CRUD is hard.",
4   "writer": "hsooovn"
5 }
```

Body Cookies Headers (4) Test Results

 200 OK 11 ms 123 B Save Response

Pretty

Raw

Preview

Visualize

Text

v



```
1 |
```

Likelion-The-Origin / Chapter4 / post-read-all

Save Send

GET {{PROTOCOL}}://{{HOST}}/post/read-all

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 8 ms 311 B Save Response

Pretty Raw Preview Visualize JSON

```

1  {
2    {
3      "title": "Hello CRUD!",
4      "content": "This is test posting.",
5      "writer": "hsooovn"
6    },
7    {
8      "title": "Second Post",
9      "content": "CRUD is hard.",
10     "writer": "hsooovn"
11   }

```

read와 read-all에서 바뀐 내용 확인



GET



{{PROTOCOL}}://{{HOST}}/post/read-one?id=1

Send



Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	id	1			
	Key	Value	Description		

Body

Cookies

Headers (5)

Test Results



200 OK 7 ms 232 B

Save Response

Pretty

Raw

Preview

Visualize

JSON



```
1 {
2   "title": "Second Post",
3   "content": "CRUD is hard.",
4   "writer": "hsooovn"
5 }
```



POST



{{PROTOCOL}}://{{HOST}}/post/update?id=1

Send



Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON



Beautify

```
1 {
2   ... "content": "CRUD is easy."
3 }
```

Body

Cookies

Headers (4)

Test Results



200 OK 15 ms 123 B

Save Response

Pretty

Raw

Preview

Visualize

Text



```
1
```





GET {{PROTOCOL}}://{{HOST}}/post/read-all

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (5) Test Results

200 OK 10 ms 311 B

Save Response

Pretty

Raw

Preview

Visualize

JSON



```
1 [
2   {
3     "title": "Hello CRUD!",
4     "content": "This is test posting.",
5     "writer": "hsooovn"
6   },
7   {
8     "title": "Second Post",
9     "content": "CRUD is easy.",
10    "writer": "hsooovn"
11  }
```



GET {{PROTOCOL}}://{{HOST}}/post/read-one?id=1

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	id	1			
	Key	Value	Description		

Body Cookies Headers (5) Test Results

200 OK 8 ms 232 B

Save Response

Pretty

Raw

Preview

Visualize

JSON



```
1 {
2   "title": "Second Post",
3   "content": "CRUD is easy.",
4   "writer": "hsooovn"
5 }
```

POST ⌵ {{PROTOCOL}}://{{HOST}}/post/update?id=0Send ⌵Params ● Authorization Headers (8) Body ● Pre-request Script Tests Settings

Cookies

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL JSON ⌵

Beautify

```
1 {
2   ... "title": "updated TITLE!"
3 }
```

Body Cookies Headers (4) Test Results 200 OK 18 ms 123 B [Save Response](#) ⌵

Pretty

Raw

Preview

Visualize

Text ⌵

1 |

GET ⌵ {{PROTOCOL}}://{{HOST}}/post/read-allSend ⌵Params ⌵ Authorization Headers (6) Body Pre-request Script Tests Settings

Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 29 ms 314 B [Save Response](#) ⌵

Pretty

Raw

Preview

Visualize

JSON ⌵

```
1 {
2   {
3     "title": "updated TITLE!",
4     "content": "This is test posting.",
5     "writer": "hsooovn"
6   },
```

Likelion-The-Origin / Chapter4 / post-read

GET

{{(PROTOCOL)}}://{{(HOST)}}/post/read-one?id=0

Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	id	q			
	Key	Value	Description		

Body Cookies Headers (5) Test Results

200 OK 9 ms 243 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "title": "updated TITLE!",
3   "content": "This is test posting.",
4   "writer": "hsooovn"
5 }
```

## delete

Likelion-The-Origin / Chapter4 / post-delete

DELETE

{{(PROTOCOL)}}://{{(HOST)}}/post/delete?id=0

Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have a body

Body Cookies Headers (4) Test Results

200 OK 15 ms 123 B Save Response

Pretty Raw Preview Visualize Text

```
1
```

Likelion-The-Origin / Chapter4 / post-read-all

Save ...

GET {{PROTOCOL}}://{{HOST}}/post/read-all Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 7 ms 234 B Save Response

Pretty Raw Preview Visualize JSON

```

1  [
2    {
3      "title": "Second Post",
4      "content": "CRUD is easy.",
5      "writer": "hsooovn"
6    }
7  ]

```

## ! 주의사항 !

임시적으로 진행

실제 사용되는 서비스에서 내부에서 List를 사용하는 경우는 드물다.

진짜로 사용자의 영구적인 자원을 관리하는 데에 있어서 클래스 안에 정의되어 있는 멤버 변수 List안에 데이터를 넣는 행위는 실제로 진행되지 않음 !!

직관적으로 보이기 위해 read, create를 사용

→ RESTAPI에서 말하는 것과 다름 !