


[개정판] Spring Boot를 이용한 RESTful API 개발

{ REST API } +  Spring
Boot

```
class Book {
    def setData(self, title, price, author):
        self.title = title
        self.price = price
        self.author = author
}

var fs = require('fs');
fs.readFile('./JONE.txt', 'utf-8', function (err, data) {
    console.log(data); // 3
});

@interface NextInnovationDelegate : NSObject <UIApplicationDelegate>
```

- Section 0: Web Service & Web Application
- Section 1: Spring Boot로 개발하는 RESTful API
- Section 2: User Service API 추가
- **Section 3: RESTful Service 기능 확장**
- Section 4: Spring Boot API 사용
- Section 5: JPA 사용
- Section 6: REST API 설계 가이드

3. Section

RESTful Service 기능 확장

- Validation
- Internationalization
- XML format으로 반환하기
- Filtering
- Version 관리

Validations 1/4

```
@PostMapping("/users")
public ResponseEntity<User> createUser(@Valid @RequestBody User user) {
    User savedUser = service.save(user);
}
```

```
public class User {
    private Integer id;

    @Size(min = 2)
    private String name;

    @Past
    private Date joinDate;
}
```

```
1 {
2   "name": "D",
3   "joinDate": "2023-11-20T03:21:32.896+0000"
4 }
```

```
1 {
2   "name": "Dowon Lee",
3   "joinDate": "2023-11-21T03:21:32.896+0000"
4 }
```

none form-data x-www-

```
1 {
2   "name": "Dowon Lee",
3   "joinDate": "2023-11-21T03:21:32.896+0000"
4 }
```

Body Cookies Headers (5) Test Results

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "type": "about:blank",
3   "title": "Bad Request",
4   "status": 400,
5   "detail": "Invalid request content.",
6   "instance": "/users"
7 }
```

Status: 400 Bad Request

Validations 2/4

- ResponseEntityExceptionHandler → handleMethodArgumentNotValid

```
@Override
protected ResponseEntity<Object> handleMethodArgumentNotValid(MethodArgumentNotValidException ex,
                                                                HttpHeaders headers,
                                                                HttpStatusCode status,
                                                                WebRequest request) {

    ExceptionResponse exceptionResponse = new ExceptionResponse(new Date(),
                                                                ex.getMessage(), ex.getBindingResult().toString());

    return new ResponseEntity(exceptionResponse, HttpStatus.BAD_REQUEST);
}
```

CustomizedResponseEntityExceptionHandler.java

Body Cookies Headers (4) Test Results

Status: 400 Bad Request Time: 130 ms Size: 1.23 KB Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "timestamp": "2023-11-20T11:11:03.404+00:00",
3   "message": "Validation failed for argument [0] in public org.springframework.http.ResponseEntity<kr.co.joneconsulting.newmyrestfulservice.bean.User> kr.co.joneconsulting.newmyrestfulservice.controller.UserController.createUser(kr.co.joneconsulting.newmyrestfulservice.bean.User): [Field error in object 'user' on field 'name': rejected value [D]; codes [Size.user.name,Size.name,Size.java.lang.String,Size]; arguments [org.springframework.context.support.DefaultMessageSourceResolvable: codes [user.name,name]; arguments []; default message [name],2147483647,2]; default message [크기가 2에서 2147483647 사이여야 합니다]] ",
4   "details": "org.springframework.validation.BeanPropertyBindingResult: 1 errors\nField error in object 'user' on field 'name': rejected value [D]; codes [Size.user.name,Size.name,Size.java.lang.String,Size]; arguments [org.springframework.context.support.DefaultMessageSourceResolvable: codes [user.name,name]; arguments []; default message [name],2147483647,2]; default message [크기가 2에서 2147483647 사이여야 합니다]"
5 }
```

Validations 3/4

```
@Override
protected ResponseEntity<Object> handleMethodArgumentNotValid(MethodArgumentNotValidException ex,
                                                                HttpHeaders headers,
                                                                HttpStatusCode status,
                                                                WebRequest request) {

    ExceptionResponse exceptionResponse = new ExceptionResponse(new Date(),
                                                                message: "Validation Failed", ex.getBindingResult().toString());

    return new ResponseEntity(exceptionResponse, HttpStatus.BAD_REQUEST);
}
```

CustomizedResponseEntityExceptionHandler.java

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 127 ms Size: 651 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "timestamp": "2023-11-20T11:06:16.398+00:00",
3   "message": "Validation Failed",
4   "details": "org.springframework.validation.BeanPropertyBindingResult: 1 errors\nField error in object 'user' on field 'name': rejected value [D]; codes [Size.user.name,Size.name,Size.java.lang.String,Size]; arguments [org.springframework.context.support.DefaultMessageSourceResolvable: codes [user.name,name]; arguments []; default message [name],2147483647,2]; default message [크기가 2에서 2147483647 사이여야 합니다]"
5 }
```

Validations 4/4

```
public class User {  
    private Integer id;  
  
    @Size(min = 2, message = "Name은 2글자 이상 입력해 주세요.")  
    private String name;  
}
```

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 124 ms Size: 643 B Save as Example

Pretty Raw Preview Visualize JSON ↺

```
1 {  
2   "timestamp": "2023-11-20T11:14:10.283+00:00",  
3   "message": "Validation Failed",  
4   "details": "org.springframework.validation.BeanPropertyBindingResult: 1 errors\nField error in object 'user' on field  
'name': rejected value [D]; codes [Size.user.name,Size.name,Size.java.lang.String,Size]; arguments [org.  
springframework.context.support.DefaultMessageSourceResolvable: codes [user.name,name]; arguments []; default  
message [name],2147483647,2]; default message [Name은 2글자 이상 입력해 주세요.]"  
5 }
```



```
@GetMapping(path = "/hello-world-internationalized")
public String helloWorldInternationalized() {
    return "Good morning";
}
```

HelloWorldController.java

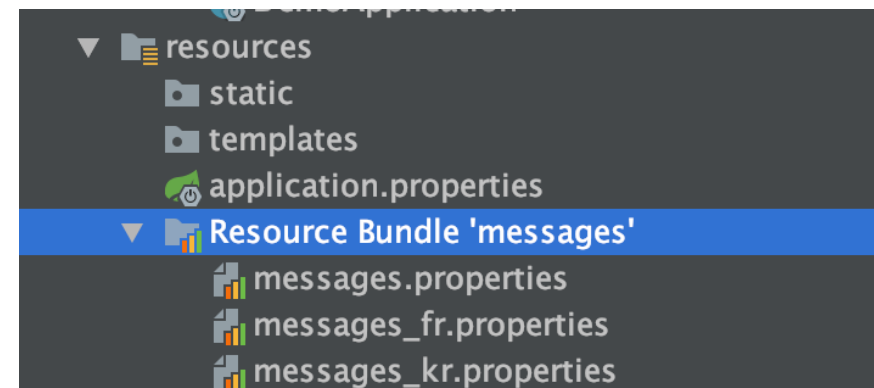
- Internationalization
- @Configuration 등록
 - LocaleResolver
 - Default Locale
 - **Locale.US** or **Locale.KOREA**
 - ResourceBundleMessageSource
- Usage
 - 1) generate message bundle files
 - 2) @Autowired MessageSource
 - 3) @RequestHeader(value = "Accept-Language", required = false) Local local
 - 4) messageSource.getMessage("greeting.message", null, local)

Internationalization 2/3

```
@SpringBootApplication
public class NewMyRestfulServiceApplication {

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

    @Bean
    public LocaleResolver localeResolver() {
        SessionLocaleResolver localeResolver = new SessionLocaleResolver();
        localeResolver.setDefaultLocale(Locale.US);
        return localeResolver;
    }
}
```



greeting.message=Hello
greeting.message=Bonjour
greeting.message=안녕하세요
greeting.message=こんにちは

■ application.yml

```
spring:
  messages:
    basename: messages
```

Internationalization 3/3

```
@GetMapping(path = "/hello-world-internationalized")
public String helloWorldInternationalized(
    @RequestHeader(name="Accept-Language", required=false) Locale locale) {
    return messageSource.getMessage( code: "greeting.message", args: null, locale);
}
```

GET 127.0.0.1:8088/hello-world-internationalized

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

	Key	Value	Description
	Key	Value	Description

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize Text

1 Hello

Status: 200 OK

<input checked="" type="checkbox"/>	Accept-Language	ja
	Key	Value

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize Text

1 こんにちは

<input checked="" type="checkbox"/>	Accept-Language	kr
	Key	Value

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize Text

1 안녕하세요

XML Format

- pom.xml
 - add dependency

```
<dependency>
  <groupId>com.fasterxml.jackson.dataformat</groupId>
  <artifactId>jackson-dataformat-xml</artifactId>
  <version>2.13.4</version>
</dependency>
```

GET 127.0.0.1:8088/users

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Key	Value
Accept	application/xml

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize XML

```
1 <List>
2   <item>
3     <id>1</id>
4     <name>Kenneth</name>
5     <joinDate>2023-11-20T11:40:22.095+00:00</joinDate>
6   </item>
7   <item>
8     <id>2</id>
9     <name>Alice</name>
10    <joinDate>2023-11-20T11:40:22.095+00:00</joinDate>
11  </item>
12  <item>
13    <id>3</id>
14    <name>Elena</name>
15    <joinDate>2023-11-20T11:40:22.095+00:00</joinDate>
16  </item>
17 </List>
```

Static Filtering 1/3

```
public class User {  
    private Integer id;  
  
    @Size(min = 2, message = "Name은 2글자 이상 입력해 주세요.")  
    private String name;  
  
    @Past  
    private Date joinDate;  
  
    private String password;  
    private String ssn;
```

```
public class UserDaoService {  
    private static List<User> users = new ArrayList<>();  
  
    private static int usersCount = 3;  
  
    static {  
        users.add(new User( id: 1, name: "Kenneth", new Date(), passw  
        users.add(new User( id: 2, name: "Alice", new Date(), passw  
        users.add(new User( id: 3, name: "Elena", new Date(), passw  
    }  
}
```

```
[  
  {  
    "id": 1,  
    "name": "Kenneth",  
    "joinDate": "2023-11-20T11:45:02.386+00:00",  
    "password": "test1",  
    "ssn": "111111-1111111"  
  },  
  {  
    "id": 2,  
    "name": "Alice",  
    "joinDate": "2023-11-20T11:45:02.386+00:00",  
    "password": "test2",  
    "ssn": "222222-2222222"  
  },  
  {  
    "id": 3,  
    "name": "Elena",  
    "joinDate": "2023-11-20T11:45:02.386+00:00",  
    "password": "test3",  
    "ssn": "333333-3333333"  
  }  
]
```

Static Filtering 2/3

■ @JsonIgnore

```
public class User {  
    private Integer id;  
  
    @Size(min = 2, message = "Name은 2글자 이상 입력해 주세요.")  
    private String name;  
  
    @Past  
    private Date joinDate;  
  
    @JsonIgnore  
    private String password;  
    @JsonIgnore  
    private String ssn;  
}
```

GET 127.0.0.1:8088/users

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

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": 1,
4     "name": "Kenneth",
5     "joinDate": "2023-11-20T11:46:26.006+00:00"
6   },
7   {
8     "id": 2,
9     "name": "Alice",
10    "joinDate": "2023-11-20T11:46:26.006+00:00"
11  },
12  {
13    "id": 3,
14    "name": "Elena",
15    "joinDate": "2023-11-20T11:46:26.006+00:00"
16  }
17 ]
```

Static Filtering 3/3

■ @JsonIgnoreProperties

```
@Data
@AllArgsConstructor
@JsonIgnoreProperties(value = {"password", "ssn"})
public class User {
    private Integer id;

    @Size(min = 2, message = "Name은 2글자 이상 입력해 주세요.")
    private String name;

    @Past
    private Date joinDate;

    // @JsonIgnore
    private String password;

    // @JsonIgnore
    private String ssn;
}
```

GET 127.0.0.1:8088/users/1

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

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ↻

```
1 {
2   "id": 1,
3   "name": "Kenneth",
4   "joinDate": "2023-11-20T11:48:08.819+00:00"
5 }
```


Dynamic Filtering 1/3

- Retrieve user info for admin

```
@Data
@AllArgsConstructor
@NoArgsConstructor
@JsonFilter("UserInfo")
public class AdminUser {
    private Integer id;

    @Size(min = 2, message = "Name은 2글자 이상")
    private String name;
}
```

AdminUser.java

```
@GetMapping("/users/{id}")
public MappingJacksonValue retrieveUser(@PathVariable int id) {
    User user = service.findOne(id);

    AdminUser adminUser = new AdminUser();
    if (user == null) {
        throw new UserNotFoundException(String.format("ID[%s] not found", id));
    } else {
        BeanUtils.copyProperties(user, adminUser);

        SimpleBeanPropertyFilter filter = SimpleBeanPropertyFilter
            .filterOutAllExcept("id", "name", "joinDate", "ssn");

        FilterProvider filters = new SimpleFilterProvider().addFilter("UserInfo", filter);

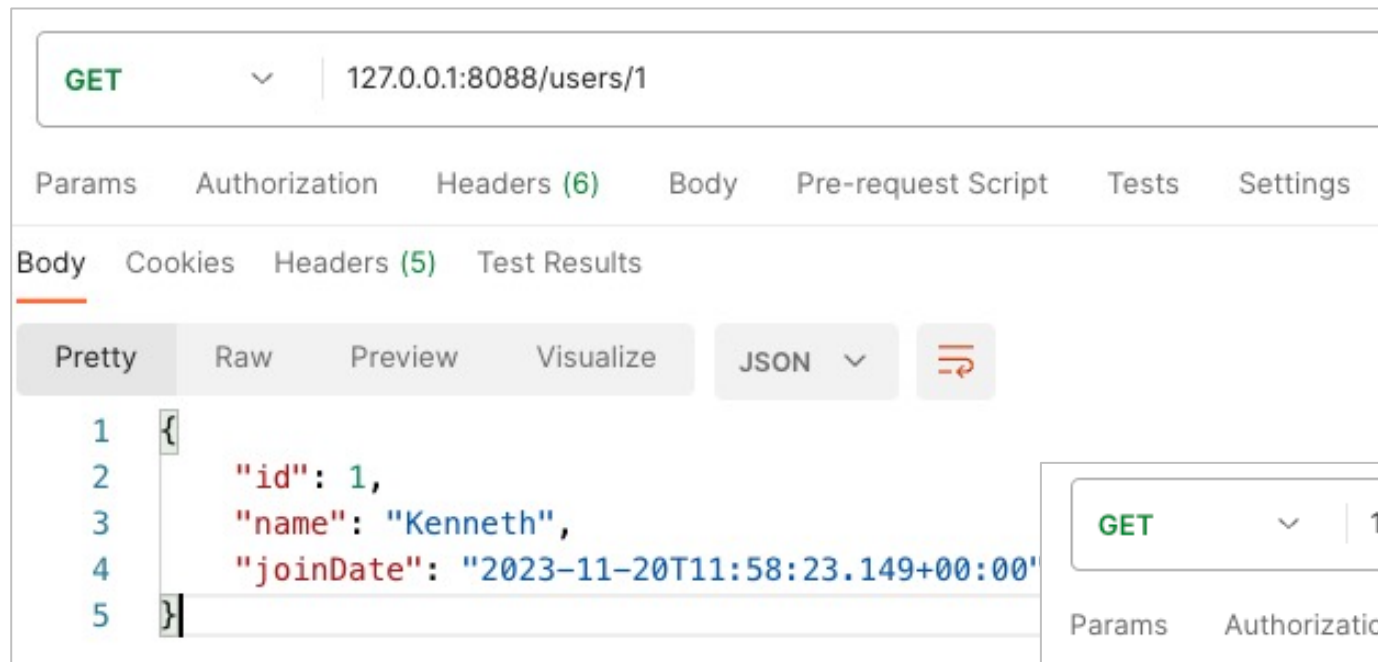
        MappingJacksonValue mapping = new MappingJacksonValue(adminUser);
        mapping.setFilters(filters);

        return mapping;
    }
}
```

AdminUserController.java

Dynamic Filtering 1/2

- Retrieve user info for admin



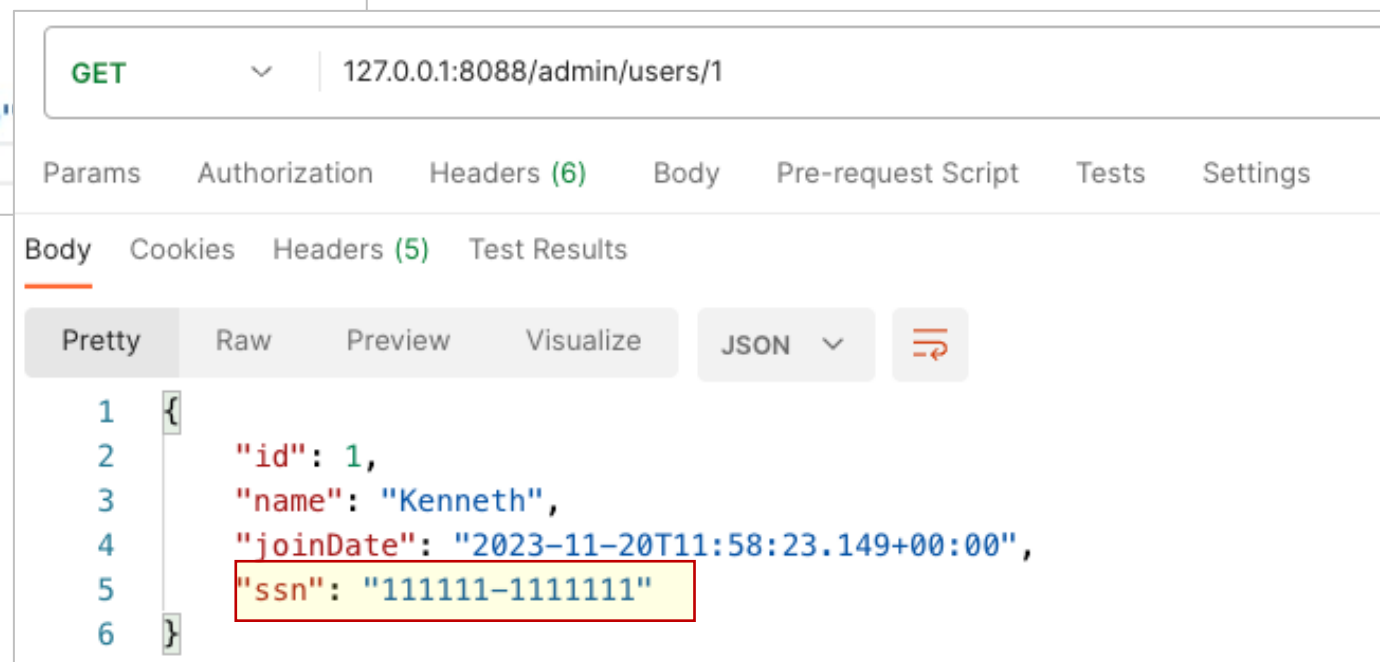
GET 127.0.0.1:8088/users/1

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

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ↺

```
1 {
2   "id": 1,
3   "name": "Kenneth",
4   "joinDate": "2023-11-20T11:58:23.149+00:00"
5 }
```



GET 127.0.0.1:8088/admin/users/1

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

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ↺

```
1 {
2   "id": 1,
3   "name": "Kenneth",
4   "joinDate": "2023-11-20T11:58:23.149+00:00",
5   "ssn": "111111-111111"
6 }
```

Dynamic Filtering 3/3

- Retrieve users list for admin

```
@GetMapping("/users")
public MappingJacksonValue retrieveAllUsers() {
    List<User> users = service.findAll();

    List<AdminUser> adminUsers = new ArrayList<>();
    AdminUser adminUser = null;
    for (User user : users) {
        adminUser = new AdminUser();
        BeanUtils.copyProperties(user, adminUser);
        adminUsers.add(adminUser);
    }

    SimpleBeanPropertyFilter filter = SimpleBeanPropertyFilter
        .filterOutAllExcept("id", "name", "joinDate", "ssn");

    FilterProvider filters = new SimpleFilterProvider().addFilter(id: "UserInfo", filter);

    MappingJacksonValue mapping = new MappingJacksonValue(adminUsers);
    mapping.setFilters(filters);

    return mapping;
}
```

```
[
  {
    "id": 1,
    "name": "Kenneth",
    "joinDate": "2023-11-20T12:12:25.182+00:00",
    "ssn": "111111-111111"
  },
  {
    "id": 2,
    "name": "Alice",
    "joinDate": "2023-11-20T12:12:25.182+00:00",
    "ssn": "222222-222222"
  },
  {
    "id": 3,
    "name": "Elena",
    "joinDate": "2023-11-20T12:12:25.182+00:00",
    "ssn": "333333-333333"
  }
]
```

Versioning 1/6

■ 버전 관리

- <https://developers.facebook.com/docs/apps/versions>
- <https://developers.kakao.com/docs/latest/ko/kakaologin/rest-api>

■ /admin/users/{id} → /admin/**v1**/users/{id}

```
@GetMapping("/users/{id}")
public MappingJacksonValue retrieveUser(@PathVariable int id) {
    User user = service.findOne(id);
```

URI 변경

```
@GetMapping("/v1/users/{id}")
public MappingJacksonValue retrieveUser(@PathVariable int id) {
    User user = service.findOne(id);
```

■ Add a new user bean

```
@Data
@AllArgsConstructor
@NoArgsConstructor
@JsonFilter("UserInfoV2")
public class AdminUserV2 extends AdminUser {
    private String grade;
}
```

Versioning 2/6

■ URI versioning

```
@GetMapping("/v2/users/{id}")
public MappingJacksonValue retrieveUserV2(@PathVariable int id) {
    User user = service.findOne(id);

    if (user == null) {
        throw new UserNotFoundException(String.format("ID[%s] not found", id));
    }
}
```

```
AdminUserV2 adminUserV2 = new AdminUserV2();
BeanUtils.copyProperties(user, adminUserV2); // id, name, joinDate, ssn
adminUserV2.setGrade("VIP"); // grade
```

```
SimpleBeanPropertyFilter filter = SimpleBeanPropertyFilter
    .filterOutAllExcept("id", "name", "joinDate", "grade");
```

```
FilterProvider filters = new SimpleFilterProvider().addFilter(id: "UserInfoV2", fil
```

GET 127.0.0.1:8088/admin/v1/users/1

Params Authorization Headers (6) Body Pre-request Script

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "name": "Kenneth",
4   "joinDate": "2023-11-20T12:22:03.024+00:00",
5   "ssn": "111111-111111"
6 }
```

GET 127.0.0.1:8088/admin/v2/users/1

Params Authorization Headers (6) Body Pre-request Script

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "name": "Kenneth",
4   "joinDate": "2023-11-20T12:22:03.024+00:00",
5   "grade": "VIP"
6 }
```

Versioning 3/6

- request parameter versioning

```
@GetMapping(value = "/users/{id}", params = "version=1")  
public MappingJacksonValue retrieveUser(@PathVariable int id) {  
    User user = service.findOne(id);
```

```
@GetMapping(value = "/users/{id}", params = "version=2")  
public MappingJacksonValue retrieveUserV2(@PathVariable int id) {  
    User user = service.findOne(id);
```

GET 127.0.0.1:8088/admin/users/1?version=1

Params • Authorization Headers (6) Body Pre-request Script

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 {  
2     "id": 1,  
3     "name": "Kenneth",  
4     "joinDate": "2023-11-20T12:25:07.814+00:00",  
5     "ssn": "111111-111111"  
6 }
```

GET 127.0.0.1:8088/admin/users/1?version=2

Params • Authorization Headers (6) Body Pre-request Script

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 {  
2     "id": 1,  
3     "name": "Kenneth",  
4     "joinDate": "2023-11-20T12:25:07.814+00:00",  
5     "grade": "VIP"  
6 }
```

Versioning 4/6

■ header versioning

```
@GetMapping(value = "/users/{id}", headers = "X-API-VERSION=1")  
public MappingJacksonValue retrieveUser(@PathVariable int id) {  
    User user = service.findOne(id);  
}
```

```
@GetMapping(value = "/users/{id}", headers = "X-API-VERSION=2")  
public MappingJacksonValue retrieveUserV2(@PathVariable int id) {  
    User user = service.findOne(id);  
}
```

GET 127.0.0.1:8088/admin/users/1

Params Authorization **Headers (7)** Body Pre-request Script Tests

Headers 6 hidden

	Key	Value
<input checked="" type="checkbox"/>	X-API-VERSION	1
	Key	Value

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 {  
2   "id": 1,  
3   "name": "Kenneth",  
4   "joinDate": "2023-11-20T12:28:33.763+00:00",  
5   "ssn": "111111-111111"  
6 }
```

GET 127.0.0.1:8088/admin/users/1

Params Authorization **Headers (7)** Body Pre-request Script Tests

Headers 6 hidden

	Key	Value
<input checked="" type="checkbox"/>	X-API-VERSION	2
	Key	Value

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 {  
2   "id": 1,  
3   "name": "Kenneth",  
4   "joinDate": "2023-11-20T12:28:33.763+00:00",  
5   "grade": "VIP"  
6 }
```


Versioning 5/6

- mime-type or accept header versioning

```
@GetMapping(value = "/users/{id}", produces = "application/vnd.company.appv1+json")
public MappingJacksonValue retrieveUser(@PathVariable int id) {
    User user = service.findOne(id);
```

GET 127.0.0.1:8088/admin/users/1

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Headers 6 hidden

	Key	Value
<input checked="" type="checkbox"/>	Accept	application/vnd.company.appv1+json

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "name": "Kenneth",
4   "joinDate": "2023-11-20T12:33:31.355+00:00",
5   "ssn": "111111-111111"
6 }
```

GET 127.0.0.1:8088/admin/users/1

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Headers 6 hidden

	Key	Value
<input checked="" type="checkbox"/>	Accept	application/vnd.company.appv2+json

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "name": "Kenneth",
4   "joinDate": "2023-11-20T12:33:31.355+00:00",
5   "grade": "VIP"
6 }
```

```
@GetMapping(value = "/users/{id}", produces = "application/vnd.company.appv2+json")
public MappingJacksonValue retrieveUserV2(@PathVariable int id) {
    User user = service.findOne(id);
```


■ Versioning

1) URI Versioning

- Twitter

2) Request Parameter versioning

- Amazon

일반 브라우저에서
실행 가능

3) (Custom) headers versioning

- Microsoft

4) Media type versioning (a.k.a “content negotiation” or “accept header”)

- GitHub

일반 브라우저에서
실행 불가

• Factors

- URI Pollution
- Misuse of HTTP Headers
- Caching
- Can we execute the request on the browser?
- API Documentation

• No Perfect solution