

## 2XX 응답 코드

Postman interface showing a successful POST request to `http://localhost:3000/items`. The request body is raw JSON:

```
1 {  
2   "name": "test"  
3 }
```

The response status is `201 Created` with a response body:

```
1 {  
2   "status": "success",  
3   "data": {  
4     "id": 0,  
5     "name": "test"  
6   }  
7 }
```

Postman interface showing a successful POST request to `http://localhost:3000/users`. The request body is raw JSON:

```
1 {  
2   "username": "Park"  
3 }
```

The response status is `201 Created` with a response body:

```
1 {  
2   "status": "success",  
3   "data": {  
4     "id": 1,  
5     "username": "Park"  
6   }  
7 }
```

Postman screenshot showing a collection named "3분반\_박세운\_202112016\_실습1". A GET request named "GET\_1" is selected, which sends a GET request to "http://localhost:3000/items". The "Query Params" tab shows a variable "baseURL" set to "http://localhost:3000" with a scope of "Collection". The response body is a JSON object:

```
1 {  
2     "status": "success",  
3     "data": [  
4         {  
5             "id": 0,  
6             "name": "test"  
7         }  
8     ]  
9 }
```

Postman screenshot showing a collection named "3분반\_박세운\_202112016\_실습1". A GET request named "GET\_2" is selected, which sends a GET request to "http://localhost:3000/users/1". The "Headers" tab shows 6 headers. The response body is a JSON object:

```
1 {  
2     "status": "success",  
3     "data": [  
4         {  
5             "id": 1,  
6             "username": "Park"  
7         }  
8     ]  
9 }
```

Postman screenshot showing a collection named "3분반\_박세운\_202112016.실습1". The PUT\_1 endpoint is selected, which sends a PUT request to http://localhost:3000/items/0 with a raw JSON body:

```
PUT /items/0
{
  "name": "testupdate"
}
```

The response is a 200 OK status with the following JSON body:

```
{
  "status": "success",
  "data": {
    "id": 0,
    "name": "testupdate"
  }
}
```

Postman screenshot showing a collection named "3분반\_박세운\_202112016.실습1". The PUT\_2 endpoint is selected, which sends a PUT request to http://localhost:3000/users/1 with a raw JSON body:

```
PUT /users/1
{
  "username": "Lee"
}
```

The response is a 200 OK status with the following JSON body:

```
{
  "status": "success",
  "data": {
    "id": 1,
    "username": "Lee"
  }
}
```

HTTP 3분단\_박세운\_202112016\_실습1 / **DELETE** / **DELETE\_1**

DELETE http://localhost:3000/items/0

Body Cookies Headers (7) Test Results 200 OK 5 ms 291 B Save Response

```
{ } JSON < Preview Visualize >
```

```
1 {
2   "status": "success",
3   "data": {
4     "id": 0,
5     "name": "testupdate"
6   }
7 }
```

HTTP 3분단\_박세운\_202112016\_실습1 / **DELETE** / **DELETE\_2**

DELETE http://localhost:3000/users/1

Body Cookies Headers (7) Test Results 200 OK 11 ms 288 B Save Response

```
{ } JSON < Preview Visualize >
```

```
1 {
2   "status": "success",
3   "data": {
4     "id": 1,
5     "username": "Lee"
6   }
7 }
```

## 4XX 응답 코드

The screenshot shows the Postman application interface. On the left, the sidebar displays '박세운's Workspace' with a collection named '3분반\_박세운\_202112016\_실습1'. In the main workspace, a POST request is being tested against the URL `http://localhost:3000/items`. The request method is set to POST, and the body is currently empty (raw JSON). The response status is 400 Bad Request, with a message indicating an error: "status": "error", "message": "Name is required", "data": null.

This screenshot shows another instance of the Postman application. The sidebar indicates '박세운's Workspace' with a collection named '3분반\_박세운\_202112016\_실습1'. A POST request is being tested against the URL `http://localhost:3000/users`. The request method is POST, and the body is empty (raw JSON). The response status is 400 Bad Request, with a message indicating an error: "status": "error", "message": "Username required", "data": null.

HTTP 3분반\_박세운\_202112016\_실습1 / GET / GET\_2

GET http://localhost:3000/users/2

Body Cookies Headers (7) Test Results 404

```
1: { "status": "error", "message": "User not found", "data": null }
```

HTTP 3분반\_박세운\_202112016\_실습1 / PUT / PUT\_1

PUT http://localhost:3000/items/1

Body Cookies Headers (8) Test Results 404

```
1: { "name": "testupdate" }
```

```
1: { "status": "error", "message": "Item not found", "data": null }
```

The screenshot shows the Postman interface with a dark theme. On the left, the sidebar displays '박세윤's Workspace' with collections like '3분반\_박세윤\_202112016\_실습1'. In the main area, a PUT request is being made to `http://localhost:3000/items/0`. The 'Body' tab is selected, showing a raw JSON payload with three empty lines. The response status is '400 Bad Request' with a message: 'status: "error", message: "Name is required", data: null'.

This screenshot shows another instance of the Postman interface. A PUT request is sent to `http://localhost:3000/users/2`. The 'Body' tab shows a raw JSON payload with three empty lines. The response status is '404 Not Found' with a message: 'status: "error", message: "User not found", data: null'.

HTTP 3분반\_박세운\_202112016\_실습1 / PUT / **PUT\_2**

PUT http://localhost:3000/users/1

Body (8) Body (raw) JSON

```
1 {  
2   "status": "error",  
3   "message": "Username required",  
4   "data": null  
5 }
```

400 Bad Request 6 ms 304 B Save Response

HTTP 3분반\_박세운\_202112016\_실습1 / DELETE / **DELETE\_1**

DELETE http://localhost:3000/items/1

Params

Key	Value	Description
Key	Value	Description

Body (8) Body (raw) JSON

```
1 {  
2   "status": "error",  
3   "message": "Item not found",  
4   "data": null  
5 }
```

404 Not Found 7 ms 299 B Save Response

The screenshot shows the Postman application interface. On the left, the sidebar displays "박세운's Workspace" with collections like "3분반\_박세운\_202112016\_실습1" containing various API endpoints such as DELETE, GET, PATCH, POST, and PUT. The main workspace shows a DELETE request to `http://localhost:3000/users/2`. The response details indicate a **404 Not Found** status with a message: "status": "error", "message": "User not found", "data": null.

Key	Value	Description
Key	Value	Description

Body:

```
{ } JSON
```

```
1 {
2     "status": "error",
3     "message": "User not found",
4     "data": null
5 }
```

## 5XX 응답 코드

The screenshot shows the Postman interface with a collection named "3분반\_박세운\_202112016\_실습1". A POST request is selected with the URL `http://localhost:3000/items`. The request body is set to raw JSON:

```
1 {  
2   "name": "error"  
3 }  
4 }
```

The response status is 500 Internal Server Error, with a response time of 57 ms and a response size of 309 B. The response body is:

```
1 {  
2   "status": "error",  
3   "message": "Server error",  
4   "data": null  
5 }
```

The screenshot shows the Postman interface with the same collection. A GET request is selected with the URL `http://localhost:3000/maintenance`. The request has a header "Content-Type: application/json" and a query parameter "key". The response status is 503 Service Unavailable, with a response time of 7 ms and a response size of 326 B. The response body is:

```
1 {  
2   "status": "error",  
3   "message": "Service temporarily unavailable",  
4   "data": null  
5 }
```

미들웨어 구현(메소드 호출 시 출력)

```
mid Time 1763183229399
request logging: GET /maintenance
app Time: 1763183229401
```