

## 2XX 응답 코드

The screenshot shows the Postman interface with a dark theme. On the left, the sidebar displays '박세윤's Workspace' with a collection named '3분반\_박세윤\_202112016\_실습1'. Under this collection, there are several requests: 'DELETE', 'GET', 'PATCH', 'POST', and 'PUT'. The 'POST' request is currently selected. The main panel shows a POST request to 'http://localhost:3000/items'. The 'Body' tab is active, showing a raw JSON payload:

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

The response status is '201 Created' with a response time of 99 ms and a size of 290 B. The response body is:

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

This screenshot shows another instance of the Postman interface with a dark theme. The sidebar displays '박세윤's Workspace' with a collection named '3분반\_박세윤\_202112016\_실습1'. The 'POST' request under this collection is selected. The main panel shows a POST request to 'http://localhost:3000/users'. The 'Body' tab is active, showing a raw JSON payload:

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

The response status is '201 Created' with a response time of 12 ms and a size of 294 B. The response body is:

```
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 is selected, with the URL set to "http://localhost:3000/items". A variable "baseURL" is defined under "Query Params" with a value of "http://localhost:3000". 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 is selected, with the URL set to "http://localhost:3000/users/1". The response body is a JSON object:

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

Home Workspaces API Network

朴세윤's Workspace

3분반\_박세윤\_202112016.실습1 / PUT / PUT\_1

PUT http://localhost:3000/items/0

Body (8)

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

200 OK 9 ms 291 B

PUT PUT\_1

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

Cloud View Find and replace Console

Home Workspaces API Network

朴세윤's Workspace

3분반\_박세윤\_202112016.실습1 / PUT / PUT\_2

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

Body (8)

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

200 OK 10 ms 288 B

PUT PUT\_1

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

Cloud View Find and replace Console

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

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

Body Cookies Headers (7) Test Results

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

200 OK 5 ms 291 B Save Response

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

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

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

Body Cookies Headers (7) Test Results

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

200 OK 11 ms 288 B Save Response

```
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'. The 'POST' tab is selected. A POST request is being made to 'http://localhost:3000/items'. The 'Body' tab is active, showing a raw JSON payload:

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

The response status is '400 Bad Request' with a duration of '5 ms' and a size of '303 B'. The response body is identical to the request body.

This screenshot shows another instance of the Postman application. The sidebar and collection structure are identical to the first screenshot. A POST request is being made to 'http://localhost:3000/users'. The 'Body' tab is active, showing a raw JSON payload:

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

The response status is '400 Bad Request' with a duration of '8 ms' and a size of '304 B'. The response body is identical to the request body.

Postman interface showing a collection named "3분반\_박세운\_202112016\_실습1". A GET request is made to `http://localhost:3000/users/2`. The response is a 404 Not Found error with the message "User not found".

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

Postman interface showing a collection named "3분반\_박세운\_202112016\_실습1". A PUT request is made to `http://localhost:3000/items/1` with a JSON body containing "name": "testupdate". The response is a 404 Not Found error with the message "Item not found".

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

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

The screenshot shows the Postman interface with a collection named "3분반\_박세윤\_202112016\_실습1". A PUT request is being made to `http://localhost:3000/items/0`. The request body is currently empty ({}). The response status is 400 Bad Request, with the message: "status": "error", "message": "Name is required", "data": null.

```
PUT http://localhost:3000/items/0
{
}
400 Bad Request
{
  "status": "error",
  "message": "Name is required",
  "data": null
}
```

The screenshot shows the Postman interface with the same collection. A PUT request is being made to `http://localhost:3000/users/2`. The request body is currently empty ({}). The response status is 404 Not Found, with the message: "status": "error", "message": "User not found", "data": null.

```
PUT http://localhost:3000/users/2
{
}
404 Not Found
{
  "status": "error",
  "message": "User not found",
  "data": null
}
```

The screenshot shows the Postman interface with a collection named "3분반\_박세운\_202112016\_실습1". A PUT request is being tested with the URL `http://localhost:3000/users/1`. The request body is set to raw JSON:

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

The response status is 400 Bad Request, and the response body is:

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

The screenshot shows the Postman interface with the same collection. A DELETE request is being tested with the URL `http://localhost:3000/items/1`. The request body is set to raw JSON:

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

The response status is 404 Not Found, and the response body is:

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

The screenshot shows the Postman application interface. On the left, the sidebar displays "박세윤's Workspace" with collections like "3분반\_박세윤\_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 "Params" tab is selected, showing a table with one row: Key (Key) and Value (Value). The "Body" tab is selected, showing a JSON response:

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

The status bar at the bottom indicates a 404 Not Found response with 6 ms duration and 299 B size.

## 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" with value "Value".

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 }
```