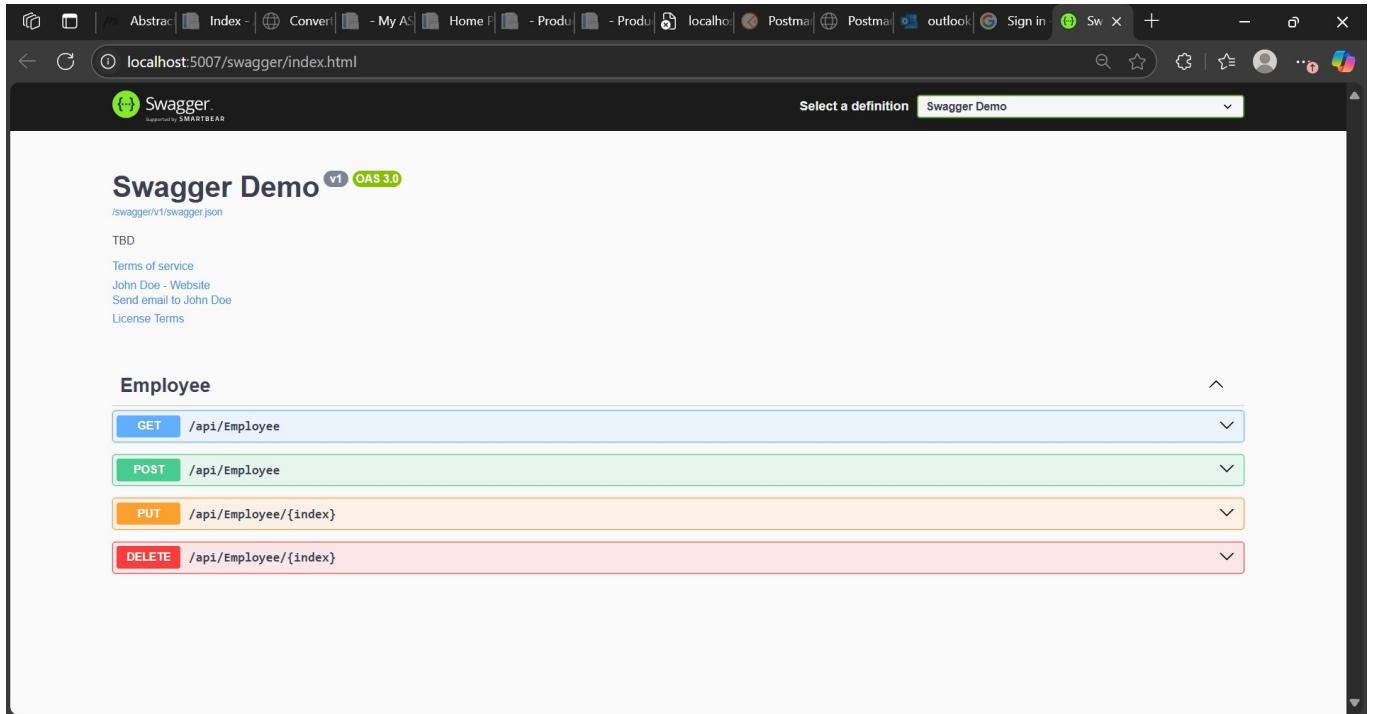


WEB API

1. Lab1



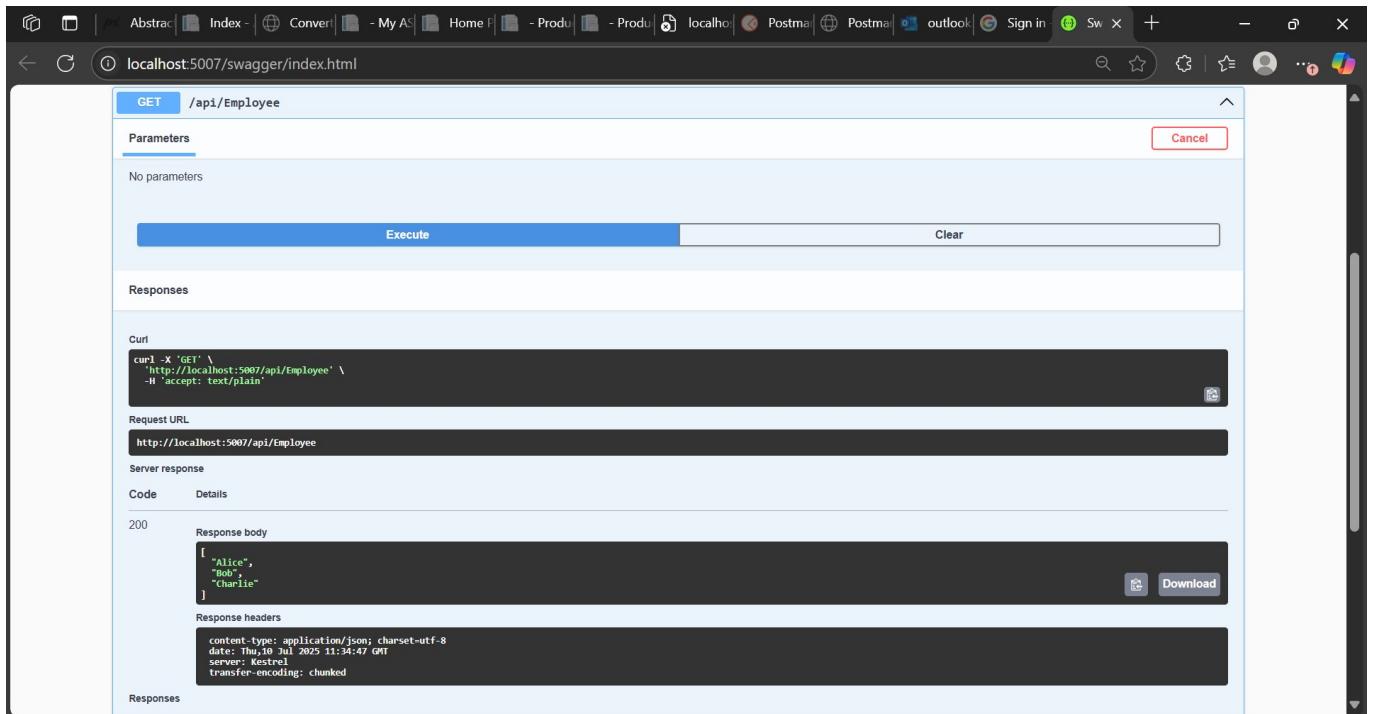
The screenshot shows the Swagger UI homepage for a 'Swagger Demo' application. At the top, there's a navigation bar with links like 'Abstract', 'Index', 'Convert', 'My API', 'Home', 'Produ...', 'localhost', 'Postma...', 'Postma...', 'outlook', 'Sign in', and 'Sw...'. Below the navigation, the title 'Swagger Demo' is displayed with a 'v1 OAS 3.0' badge. A dropdown menu 'Select a definition' is set to 'Swagger Demo'. The main content area is titled 'Employee' and lists four API operations:

- GET /api/Employee** (blue button)
- POST /api/Employee** (green button)
- PUT /api/Employee/{index}** (orange button)
- DELETE /api/Employee/{index}** (red button)

2. Lab2

SWAGGER UI

GET



The screenshot shows the detailed view for the 'GET /api/Employee' endpoint in the Swagger UI. The top section shows the method 'GET' and the URL '/api/Employee'. Below this, under 'Parameters', it says 'No parameters'. There are two buttons at the bottom: 'Execute' (blue) and 'Clear' (white). The 'Responses' section is expanded, showing the 'Curl' command:

```
curl -X 'GET' \
  'http://localhost:5007/api/Employee' \
  -H 'accept: text/plain'
```

Below the curl command, the 'Request URL' is shown as 'http://localhost:5007/api/Employee'. The 'Server response' section shows the status code '200' and the response body:

```
[{"name": "Alice"}, {"name": "Bob"}, {"name": "Charlie"}]
```

There are 'Download' and 'Copy' buttons next to the response body. The 'Response headers' section shows:

```
content-type: application/json; charset=utf-8
date: Thu,10 JUL 2025 11:34:47 GMT
server: Kestrel
transfer-encoding: chunked
```

POST

The screenshot shows the Postman interface with a POST request to `/api/Employee`. The 'Parameters' tab is selected, showing a parameter named `name` with the value `David`. Below the parameters are sections for 'Responses', 'Curl', 'Request URL', 'Server response', and detailed response headers and body.

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:5007/api/Employee?name=David' \
  -H 'Accept: */*' \
  -d ''
```

Request URL

`http://localhost:5007/api/Employee?name=David`

Server response

Code Details

200 Response body

```
Employee added.
```

Response headers

```
content-type: text/plain; charset=utf-8
date: Thu, 10 Jul 2025 11:36:40 GMT
server: Kestrel
transfer-encoding: chunked
```

PUT

The screenshot shows the Postman interface with a PUT request to `/api/Employee/{index}`. The 'Parameters' tab is selected, showing two parameters: `index` (value 2) and `newName` (value Boba). Below the parameters are sections for 'Responses', 'Curl', 'Request URL', 'Server response', and detailed response headers and body.

Responses

Curl

```
curl -X 'PUT' \
  'http://localhost:5007/api/Employee/2?newName=Boba' \
  -H 'Accept: */*'
```

Request URL

`http://localhost:5007/api/Employee/2?newName=Boba`

Server response

Code Details

200 Response body

```
Employee updated.
```

Response headers

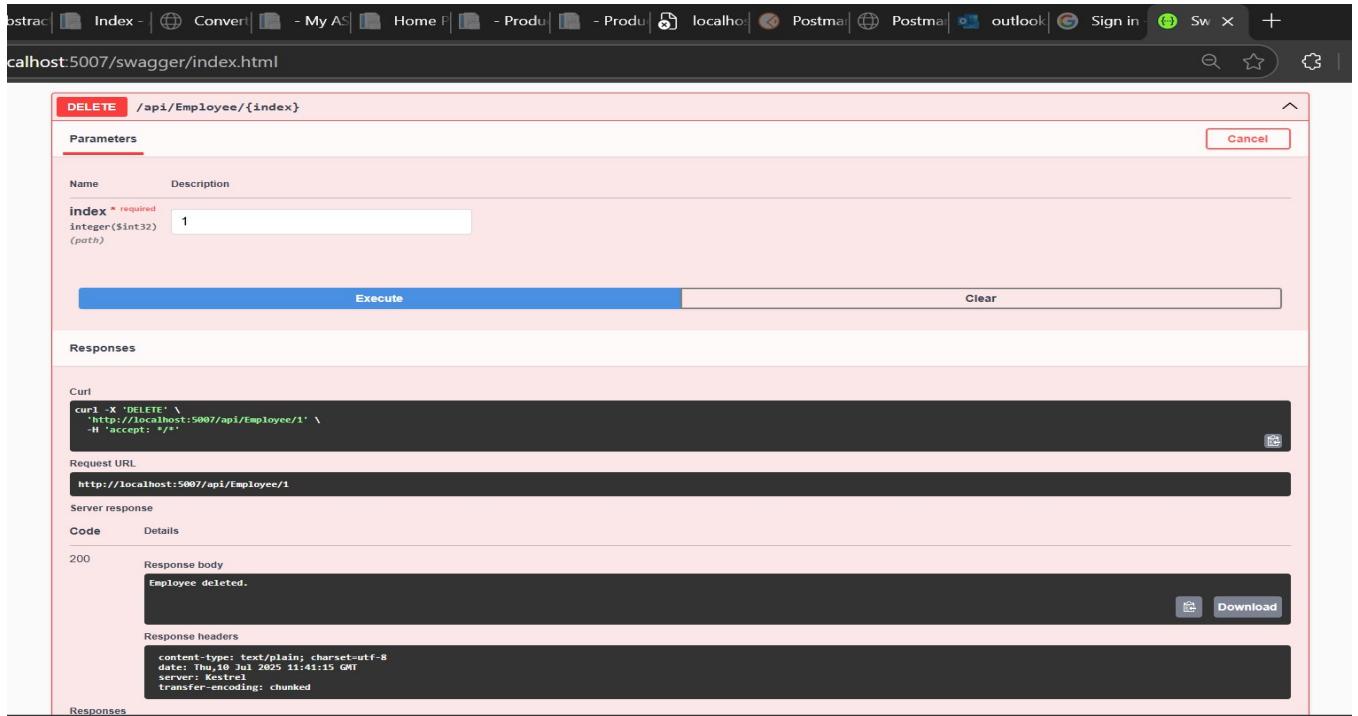
```
content-type: text/plain; charset=utf-8
date: Thu, 10 Jul 2025 11:38:56 GMT
server: Kestrel
transfer-encoding: chunked
```

Responses

Code Description

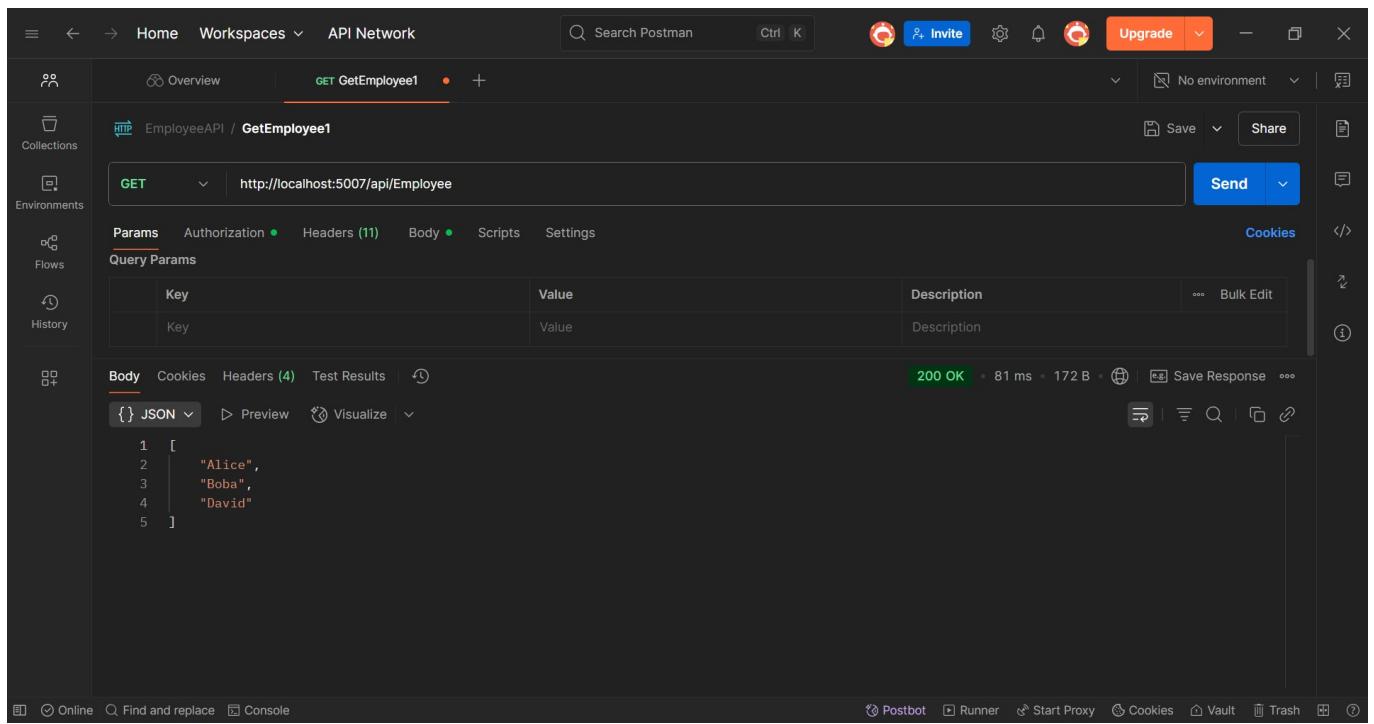
200 OK

DELETE



The screenshot shows the Postman interface for a DELETE request to the endpoint `/api/Employee/{index}`. The 'Parameters' section contains a required parameter `index` with a value of 1. Below the parameters are 'Responses' and 'Code' sections. The 'Code' section for status code 200 shows the response body: "Employee deleted." and the response headers: content-type: text/plain; charset=utf-8, date: Thu,10 Jul 2025 11:41:15 GMT, server: Kestrel, transfer-encoding: chunked.

POSTMAN GET



The screenshot shows the Postman interface for a GET request to the endpoint `http://localhost:5007/api/Employee`. The 'Params' tab is selected, showing a single parameter 'Key' with a value of 'Value'. The 'Body' tab displays a JSON response: [1, [2, "Alice", 3, "Boba", 4, "David", 5]]. The status bar at the bottom indicates a 200 OK response with 81 ms latency and 172 B size.

POST

The screenshot shows the Postman interface with a dark theme. On the left, there's a sidebar with 'Collections', 'Environments', 'Flows', and 'History'. The main area has a header 'Overview' and 'POST GetEmployee1'. Below it, a 'POST' request is shown with the URL 'http://localhost:5007/api/Employee'. The 'Body' tab is selected, showing raw JSON:

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

. The response section shows a green '200 OK' status with '65 ms' and '157 B' response time. The body of the response is 'Employee added.'

PUT

The screenshot shows the Postman interface with a dark theme. The sidebar is identical to the previous screenshot. The main area has a header 'Overview' and 'PUT GetEmployee1'. Below it, a 'PUT' request is shown with the URL 'http://localhost:5007/api/Employee/3'. The 'Body' tab is selected, showing raw JSON:

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

. The response section shows a green '200 OK' status with '197 ms' and '159 B' response time. The body of the response is 'Employee updated.'

DELETE

The screenshot shows the Postman interface with a DELETE request to `http://localhost:5007/api/Employee/3`. The response is a 200 OK status with the message "Employee deleted.".

3. Lab3

PART 1

The screenshot shows the Postman interface with a GET request to `http://localhost:5007/api/Employee/`. The response is a 200 OK status with a JSON array of employee data:

	id	name	salary	permanent	departmentName	skills	dateOfBirth
0	1	Alice	50000	true	HR	0 C# 1 SQL	1990-05-01T00:00:00
1	2	Bob	60000	false	IT	0 Java 1 React	1988-10-15T00:00:00

PART 2

SWAGGER

The screenshot shows the Swagger UI homepage for a service named "Swagger Demo". On the left, there's a sidebar with a "TBD" status message and links to "Terms of service", "John Doe - Website", "Send email to John Doe", and "License Terms". The main content area is titled "Employee" and lists four HTTP methods for the endpoint "/api/Employee": GET, POST, PUT, and DELETE. To the right of the main content, a modal dialog titled "Available authorizations" is open, showing a "Bearer (apiKey)" section with a text input field containing "Bearer test123". There are "Authorize" and "Close" buttons at the bottom of the dialog.

The screenshot shows the detailed view for the GET /api/Employee endpoint. At the top, there are tabs for "GET" and "POST", with "GET" selected. Below the tabs, there are sections for "Parameters" (which shows "No parameters") and "Responses". Under "Responses", there are sections for "Curl" (containing a command-line example), "Request URL" (showing "http://localhost:5007/api/Employee"), and "Server response". The "Server response" section is expanded, showing a "Code" tab (selected) and a "Details" tab. The "Code" tab displays a response body with two JSON objects representing employees. The "Details" tab shows the response headers: "content-type: application/json; charset=utf-8", "date: Fri, 28 Oct 2025 12:19:12 GMT", "server: Kestrel", and "transfer-encoding: chunked". At the bottom, there are sections for "Responses", "Code", "Description", and "Links".

POSTMAN

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', 'API Network', and a search bar. On the left, there are sidebar panels for 'Collections', 'Environments', 'Flows', and 'History'. The main workspace shows a request for 'EmployeeAPI / GetEmployee1' with a 'GET' method and the URL 'http://localhost:5007/api/Employee/'. The 'Headers' tab is selected, showing a single header 'Authorization' with the value 'Bearer test123'. Below the headers, the 'Body' tab is selected, displaying a JSON response with two employees: Alice and Bob. The response structure includes fields like id, name, salary, permanent, departmentName, skills, and dateOfBirth. The status bar at the bottom indicates a 200 OK response with 18 ms latency and 424 B size.

PART 3

SWAGGER

The screenshot shows a browser window displaying a Swagger UI interface for an API endpoint. The URL in the address bar is 'localhost:5007/swagger/index.html'. The main area shows a 'GET /api/Employee/crash' operation. Under the 'Parameters' section, it says 'No parameters'. Below that is an 'Execute' button and a 'Clear' button. The 'Responses' section is collapsed. Under 'Curl', there is a code block:

```
curl -X 'GET' \
'http://localhost:5007/api/Employee/crash' \
-H 'accept: */*'
```

. Under 'Request URL', it shows 'http://localhost:5007/api/Employee/crash'. Under 'Server response', it shows a '400 Undocumented' error with the message 'Error: Bad Request' and 'Response body':

```
Invalid request - No Auth token
```

. Under 'Response headers', it shows:

```
content-type: text/plain; charset=utf-8
date: Thu, 10 Jul 2025 12:52:45 GMT
server: Kestrel
transfer-encoding: chunked
```

. The 'Responses' section is collapsed, showing a 'Code' table with rows for 400 (Undocumented) and 500 (Internal Server Error). The 'Links' column for the 500 row is 'No links'.

POSTMAN

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections for Collections, Environments, Flows, and History. The main area displays a collection named "EmployeeAPI / GetEmployee1". A GET request is shown with the URL `http://localhost:5007/api/Employee/crash`. The Headers tab is selected, showing an Authorization header with the value `Bearer test123`. The Body tab is also visible. Below the request, the response status is `500 Internal Server Error`, with a message: `Something went wrong. Please contact support.`. The bottom navigation bar includes links for Online, Find and replace, Console, Postbot, Runner, Start Proxy, Cookies, Vault, Trash, and Help.

4. Lab 4

SWAGGER

The screenshot shows the Swagger UI interface for the Employee API. At the top, there's a navigation bar with links for Prod, Prod, localho, Postm, Postm, outloo, Sign in, and localho. The main content area shows the API documentation for the `/api/Employee` resource. It includes sections for `GET /api/Employee/crash` and `PUT /api/Employee/{id}`. The `PUT` method has a parameter `id` (required, integer) and a request body example:

```
{
  "name": "Alice Updated",
  "salary": 50000,
  "department": true,
  "departmentName": "IT",
  "language": "SQL",
  "dateOfBirth": "1990-01-01T00:00:00"
}
```

Below the requests, there's a `Responses` section with a `Curl` example:

```
cURL -X PUT \
http://localhost:5007/api/Employee/1 \
-H 'Authorization: Bearer test123' \
-H 'Content-Type: application/json' \
-d '{
  "name": "Alice Updated",
  "salary": 50000,
  "department": true,
  "departmentName": "IT",
  "language": "SQL",
  "dateOfBirth": "1990-01-01T00:00:00"
}'
```

The `Server response` section shows a successful `200` response with the following JSON body:

```
{
  "id": 1,
  "name": "Alice Updated",
  "salary": 50000,
  "department": true,
  "departmentName": "IT",
  "language": [
    "C#",
    "SQL"
  ],
  "dateOfBirth": "1990-01-01T00:00:00"
}
```

At the bottom, there's a `Response` section for `Code: 200 OK` with the same JSON body and a `Links` section indicating "No links".

POSTMAN

The screenshot shows the Postman application interface. In the top navigation bar, 'Home' and 'Workspaces' are visible. The main workspace title is 'PUT GetEmployee1'. The request method is 'PUT' and the URL is 'http://localhost:5007/api/Employee/1'. The 'Body' tab is selected, showing a raw JSON payload:

```
1 {
2     "id": 1,
3     "name": "Alice Updated New",
4     "salary": 58000,
5     "permanent": true,
6     "departmentName": "HR",
7     "skills": ["C#", "SQL", "Azure"],
8     "dateOfBirth": "1990-05-01T00:00:00"
9 }
```

The response status is '200 OK' with a duration of '6 ms' and a size of '303 B'. Below the body, there's a preview of the JSON response:

```
1 {
2     "id": 1,
3     "name": "Alice Updated New",
4     "salary": 58000,
5     "permanent": true,
6     "departmentName": "HR",
7     "skills": [
```

On the left sidebar, 'Collections', 'Environments', and 'Flows' are listed. The bottom of the screen shows various icons for Postbot, Runner, Start Proxy, Cookies, Vault, and Trash.

5. Lab 5

The screenshot shows a browser window displaying the Swagger UI for a 'GET /api/Auth/get-token' endpoint. The 'Parameters' section indicates 'No parameters'. The 'Responses' section shows a '200' response with a 'Response body' containing a long JWT token. The 'Request URL' is 'http://localhost:5007/api/Auth/get-token'. The 'Server response' section shows the raw response headers and body. At the bottom right, there are 'Links' and other UI elements.

Vice
Website
to John Doe

ns

Available authorizations

Bearer (apiKey)

Enter 'Bearer (token)'
Name: Authorization
In: header
Value:
Bearer eyJhbGciOiJIUzI1Nils

Authorizer

/api/Auth/get-token

ers

parameters

Execute Clear

Cancel

Swagger/index.html

Employee

GET /api/Employee

Parameters

No parameters

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:5007/api/Employee' \
  -H 'Accept: text/plain' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vczNoZd1hcyStahMyb3MvZnQyY29tL3dzLxTwMDgvYDyval8lbnRpdi8vY2shaK1zl37vhGU01jBZ61pbIItIIVzZXJ2CT6TjEuMSIzInV4cCIGRtc1MjE1NxY3M1xdax'
```

Request URL

http://localhost:5007/api/Employee

Server response

Code	Details
200	Response body

```
[{"id": 1, "name": "Alice", "salary": 60000, "perm": true, "departmentname": "HR", "skills": ["HR", "CRM", "SQL"]}, {"id": 2, "name": "Bob", "salary": 60000, "perm": false, "departmentname": "IT", "skills": ["Java", "React"]}], "dateOfBirth": "1990-05-01T00:00:00"}]
```

Download

Response headers

```
content-type: application/json; charset=utf-8
date: Thu, 18 Jul 2025 14:29:32 GMT
server: Kestrel
transfer-encoding: chunked
```

Responses

Code Description

Links

After 2 minutes (expiration of token)

The screenshot shows the Swagger UI interface for a REST API. The URL in the browser is `host:5007/swagger/index.html`. The main section displays a **GET /api/Employee** endpoint. Under the **Responses** tab, the **401** status code is listed with the description "Error: Unauthorized". The response body shows the following JSON:

```
content-length: 0
date: Thu,10 Jul 2025 14:34:40 GMT
server: Kestrel
www-authenticate: Bearer error="invalid_token",error_description="The token expired at '07/10/2025 14:27:52'"
```

Below the responses, there is a **Responses** table with one row for the **200 OK** status code, which has a **text/plain** media type.

POSTMAN

The screenshot shows the Postman application interface. On the left, there are navigation tabs for Home, Workspaces, and API Network. The main workspace shows a collection named "EmployeeAPI" with a single endpoint named "GetEmployee1". The endpoint is a **GET** request to `http://localhost:5007/api/auth/get-token`. The "Params" tab is selected, showing a single parameter "Key" with the value "Key". The "Body" tab is also visible. Below the request details, the response section shows a **200 OK** status with a response time of 63 ms and a size of 406 B. The response body is displayed in raw JSON format:

```
eyJhbGciOiJTUzI1NiIsInR5cCI6IkpxVCJ9.
eyJodHRwOi8vc2NoZW1hcystaWByb3NvZnQuY29tL3dzLzIwMDgvMDYvaWR1bnRpdkhvY2xhaW1zL3JvbGUoJBZG1pbisIIVzZXJJZCI6IjEwMSIsImV4cCI6MTc1MjE1NzAxNiwiaXNzIjoiobXlTeXN0ZW0iLCJhdWQiOidteVvzzXJzIn0.K3vcDRUL0rpAZR1kjGVroe6UjjfLiSJU3zF03MCvsKU
```

The screenshot shows the Postman interface with the following details:

- Header Bar:** Home, Workspaces, API Network, Search Postman, Ctrl + K, Invite, Notifications, Upgrade.
- Sidebar:** Collections, Environments, Flows, History, and a Plus icon.
- Request Overview:** GET GetEmployee1, Overview tab selected.
- Request Details:** EmployeeAPI / GetEmployee1, Method: GET, URL: http://localhost:5007/api/Employee.
- Authorization Tab:** Auth Type: Bearer Token, Token input field containing a redacted token.
- Body Tab:** Body, Cookies, Headers (4), Test Results.
- Response:** Status: 200 OK, Response Time: 201 ms, Response Size: 424 B, Save Response button.
- Preview:** JSON response showing two employees:

	id	name	salary	permanent	departmentName	skills	dateOfBirth
0	1	Alice	50000	true	HR	0 C# 1 SQL	1990-05-01T00:00:00
1	2	Bob	60000	false	IT	0 Java 1 React	1988-10-15T00:00:00

After 2 minutes (expiration of token)

The screenshot shows the Postman application interface. On the left, there's a sidebar with icons for Overview, Collections, Environments, Flows, and History. The main area has a header with 'Home' and 'Workspaces' dropdowns, a search bar 'Search Postman', and a 'Ctrl + K' keyboard shortcut. The top navigation bar includes 'Invite', a gear icon, a bell icon, and an 'Upgrade' button.

The central workspace shows a collection named 'EmployeeAPI' with a single endpoint 'GetEmployee1'. The request method is 'GET' to the URL 'http://localhost:5007/api/Employee'. The 'Authorization' tab is selected, showing 'Bearer Token' as the auth type. A token field contains a long string of dots, indicating a redacted access token. The 'Headers' tab shows 11 items. The 'Body' tab is selected, and the 'Cookies' tab is visible on the right.

The response section shows a 401 Unauthorized status with a timestamp of 47 ms and a size of 214 B. There are 'Save Response' and 'Copy' buttons. Below the status, the raw response body is displayed as a long string of characters: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJodHRwOi8vc2NoZW1hc5taWNyb3NvZnQuY29tL3dzLzIwMDgvMDYvaWRlbnRpdkvY2xhaW1zL3JvbGUiOiJBZG1p...