

REST API Assignment Solution

1. Home Page - <http://localhost:8080/>

Welcome to Employee Management

[Employee Management API](#) [H2 – Console](#) [Login](#)

View Users & Roles

Users		Roles
Username (Id)	Password	Name (Id)
admin (1)	\$2a\$10\$JlVb0gK22hH3S8YWJabF8uLwmzVoiP.a.ttj1Nm5W8axofeKGO8Y6	ADMIN (1)
user (2)	\$2a\$10\$DNzZd20lUdZ0aL3eTTcjPeh5p1jrLuaUinC1e3XmFOQuR2zlAXPFW	USER (2)

Note: Default username and password is same

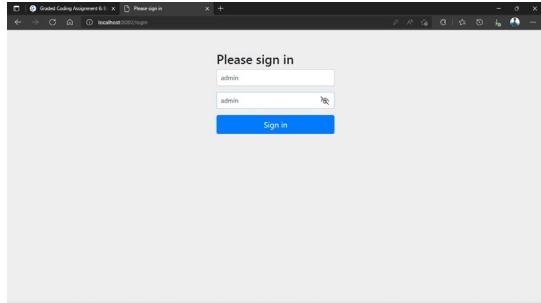
View Employees

ID	First Name	Last Name	Email
1	Harshit	Choudhary	Harshit@greatlearning.in
2	Ankit	Garg	Ankit@greatlearning.in
3	Monica	Sharma	Monica@greatlearning.in
4	Bhavya	Shetty	BHavya@greatlearning.in

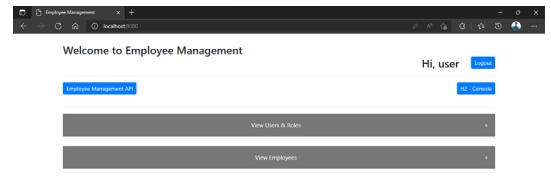
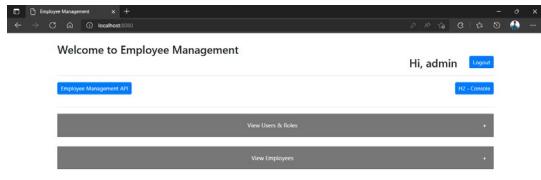
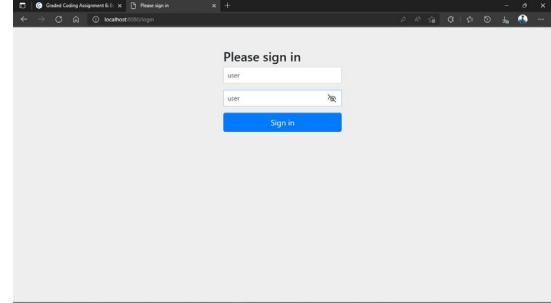
2. Login Page - <http://localhost:8080/login/>

The screenshot shows a single browser window displaying the 'Please sign in' form. The URL in the address bar is `localhost:8080/login`. The form contains two input fields: 'Username' and 'Password', and a blue 'Sign in' button.

Admin Login



User Login



3. Employee Management API - <http://localhost:8080/swagger-ui.html>

The screenshot shows the Swagger UI interface for the Employee Management API. At the top, there's a green header bar with the 'swagger' logo and a dropdown menu labeled 'EmployeeManagement-API'. Below the header, the title 'EmployeeManagement API' is displayed with a '1.0' badge. A note indicates the base URL is 'localhost:8080' and provides a link to the API specification at 'http://localhost:8080/v2/api-docs?group=EmployeeManagement-API'.

Under the title, there are links for 'EmployeeManagement API for developers', 'Terms of service', 'EmployeeManagement API - Website', 'Send email to EmployeeManagement API', and 'EmployeeManagement License'.

The main content area is organized into sections for different controllers:

- employee-controller** (Employee API):
 - DELETE** /employees/deleteById deleteById
 - GET** /employees/findAll findAll
 - GET** /employees/findByFirstNameContainsAllIgnoreCase findByFirstNameContainsAllIgnoreCase
 - GET** /employees/findById findById
 - POST** /employees/save save
 - GET** /employees/sortByFirstNameDirection sortByFirstNameDirection
 - PUT** /employees/updateById updateById
- role-controller** (Role API):
 - DELETE** /roles/deleteById deleteById
 - GET** /roles/findAll findAll
 - GET** /roles/findById findById
 - POST** /roles/save save
 - PUT** /roles/updateById updateById
- user-controller** (User API):
 - DELETE** /users/deleteById deleteById
 - GET** /users/findAll findAll
 - GET** /users/findById findById
 - POST** /users/save save
 - PUT** /users/updateById updateById
- Models**: This section displays the schema definitions for the Employee, Role, and User entities.
 - Employee** (Employee.v):

```
Employee v {  
    email string  
    firstName string  
    id integer($int32)  
    lastName string  
}
```
 - Role** (Role.v):

```
Role v {  
    id integer($int32)  
    name string  
}
```
 - User** (User.v):

```
User v {  
    id integer($int32)  
    password string  
    userRoles v (Role v {  
        id integer($int32)  
        name string  
    })  
    username string  
}
```

4. Admin Operations

• Employee

➤ List All

The image displays two side-by-side screenshots of the EmployeeManagement API's Swagger documentation, specifically focusing on the 'employee-controller' endpoint.

Left Screenshot (employee-controller):

- Endpoint:** /employees/findAll
- Method:** GET
- Responses:**
 - 200: OK (Example Value: [{}])
 - 401: Unauthorized
 - 403: Forbidden
 - 404: Not Found
- Other Endpoints:** /employees/deleteById, /employees/findByName, /employees/findByNameContainsAllIgnoreCase, /employees/findById, /employees/save, /employees/sortByFirstNameDirection, /employees/updateById

Right Screenshot (employee-controller):

- Endpoint:** /employees/findAll
- Method:** GET
- Responses:**
 - 200: OK (Example Value: [{}])
 - 401: Unauthorized
 - 403: Forbidden
 - 404: Not Found
- Other Endpoints:** /employees/deleteById, /employees/findByName, /employees/findByNameContainsAllIgnoreCase, /employees/findById, /employees/save, /employees/sortByFirstNameDirection, /employees/updateById

Common Headers (Visible in both screenshots):

```
cache-control: no-store, max-age=0, must-revalidate  
connection: keep-alive  
content-type: application/json  
date: Sat, 01 Jan 2022 00:00:00 GMT  
host: localhost:8080  
keep-alive: timeout=60  
origin: http://localhost:8080  
transfer-encoding: chunked  
upgrade-insecure-requests: 1  
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4666.64 Safari/537.36  
x-content-type-options: nosniff  
x-frame-options: sameorigin  
x-xss-protection: 1; mode=block
```

➤ Find by First Name

swagger Select a spec EmployeeManagement API

EmployeeManagement API ⓘ
[Base URL: localhost:8080]
https://localhost:8080/api/doc/api.html#/EmployeeManagementAPI

EmployeeManagement API for developers
Terms of service
EmployeeManagement API - website
Send email to EmployeeManagement API
EmployeeManagement License

employee-controller Employee API

DELETE /employees/deleteById deleteById

GET /employees/findAll findAll

GET /employees/findbyFirstNameContainsAllIgnoreCase findByFirstNameContainsIgnoreCase

Parameters

Name Description
firstName * required string (query)

Responses

Code Description
200 OK
Example Value (Model)

```
{ "id": 1, "firstName": "string", "lastName": "string", "middleName": "string" }
```

401 Unauthorized

403 Forbidden

404 Not Found

GET /employees/findbyId findById

POST /employees/save save

GET /employees/sortByFirstNameDirection sortByFirstNameDirection

PUT /employees/updateById updateById

role-controller Role API

user-controller User API

Models

swagger Select a spec EmployeeManagement API

EmployeeManagement API ⓘ
[Base URL: localhost:8080]
https://localhost:8080/api/doc/api.html#/EmployeeManagementAPI

EmployeeManagement API for developers
Terms of service
EmployeeManagement API - website
Send email to EmployeeManagement API
EmployeeManagement License

employee-controller Employee API

DELETE /employees/deleteById deleteById

GET /employees/findAll findAll

GET /employees/findbyFirstNameContainsIgnoreCase findByFirstNameContainsIgnoreCase

Parameters

Name Description
firstName * required string (query)

Responses

Code Details
200 Response body

```
[ { "id": 2, "firstName": "string", "lastName": "string", "middleName": "string" } ]
```

Request URL
<http://localhost:8080/employees/findbyFirstNameContainsIgnoreCase?firstName=arkit>

Server response
200 Response body

```
[ { "id": 2, "firstName": "string", "lastName": "string", "middleName": "string" } ]
```

Response headers
cache-control: no-cache, no-store, max-age=0, must-revalidate
connection: keep-alive
content-type: application/json
date: Sat, 01 Jun 2013 04:04:04 GMT
pragma: no-cache
status: 200
vary: Accept
x-frame-options: DENY
x-xss-protection: 1; mode=block

Code Description
200 OK
Example Value (Model)

```
[ { "id": 2, "firstName": "string", "lastName": "string", "middleName": "string" } ]
```

401 Unauthorized

403 Forbidden

404 Not Found

GET /employees/findbyId findById

POST /employees/save save

GET /employees/sortByFirstNameDirection sortByFirstNameDirection

PUT /employees/updateById updateById

role-controller Role API

user-controller User API

Models

➤ Find by Id

EmployeeManagement API

employee-controller Employee API

DELETE /employees/deleteById employeeId

GET /employees/findall

GET /employees/findbyFirstNameContainsAllIgnoreCase firstName containsAllIgnoreCase

GET /employees/findById findById

Parameters

Name	Description
employeeId * integer	employeeId

Responses

Code	Description
200	OK
401	Unauthorized
403	Forbidden
404	Not Found

POST /employees/save save

GET /employees/sortByFirstNameDirection sortByFirstNameDirection

PUT /employees/updateById updateById

role-controller Role API

user-controller User API

Models

EmployeeManagement API

employee-controller Employee API

DELETE /employees/deleteById employeeId

GET /employees/findall

GET /employees/findbyFirstNameContainsAllIgnoreCase firstName containsAllIgnoreCase

GET /employees/findById findById

Parameters

Name	Description
employeeId * integer	employeeId

Responses

Code	Description
200	OK
401	Unauthorized
403	Forbidden
404	Not Found

POST /employees/save save

GET /employees/sortByFirstNameDirection sortByFirstNameDirection

PUT /employees/updateById updateById

role-controller Role API

user-controller User API

Models

➤ Sort by First Name & Direction

The image displays three side-by-side screenshots of the EmployeeManagement API documentation in the Swagger UI interface. Each screenshot shows the /employees endpoint with various sorting parameters.

Screenshot 1 (Left): Shows the /employees endpoint with a 'Sort' dropdown set to 'firstName'. It includes a note: 'Sort employees by first name. Order direction can be asc or desc.'

Screenshot 2 (Middle): Shows the /employees endpoint with a 'Sort' dropdown set to 'lastName'. It includes a note: 'Sort employees by last name. Order direction can be asc or desc.'

Screenshot 3 (Right): Shows the /employees endpoint with a 'Sort' dropdown set to 'id'. It includes a note: 'Sort employees by id. Order direction can be asc or desc.'

All three screenshots show the same basic structure: a top navigation bar, a sidebar with 'employee-controller', 'role-controller', and 'user-controller' sections, and a main content area with tabs for 'Responses' and 'Code'.

➤ Add

The screenshot shows the EmployeeManagement API documentation for the employee-controller. It includes sections for DELETE /employees/deleteById, GET /employees/findAll, GET /employees/findByNameContainsIgnoreCase, GET /employees/findById, POST /employees/save, and a detailed view of the POST /employees/save method. The detailed view shows parameters for email, firstName, and lastName, and response codes for 200, 201, 401, 403, and 404.

This screenshot shows the same employee-controller section but with a different UI. It features a large input field for 'email' with placeholder text 'ac@gl.in'. Below it are fields for 'firstName' and 'lastName'. A 'Cancel' button is visible in the top right corner of the input field area.

➤ Update by id

EmployeeManagement API®

EmployeeManagement API for developers
Terms of service
Contact support, API, feedback
Send email to EmployeeManagement API
EmployeeManagement License

employee-controller Employee API

DELETE /employees/deleteById {modelId}

GET /employees/findAll {model}

GET /employees/findFirstWhereContainsAllIgnoreCase {firstNameContainsIgnoreCase}

GET /employees/find.byId {id}

POST /employees/save {model}

GET /employees/sortByFirstNameDirection {sortByFirstNameDirection}

PUT /employees/updateById {updateModel}

Parameters

Name	Description
employee *	employee
(model)	Example value : Model
Response content type	application/json

Responses

Code	Description
200	OK
201	Created
401	Unauthorized
403	Forbidden
404	Not Found

role-controller Role API

user-controller User API

Models

EmployeeManagement API®

EmployeeManagement API for developers
Terms of service
Contact support, API, feedback
Send email to EmployeeManagement API
EmployeeManagement License

employee-controller Employee API

DELETE /employees/deleteById {modelId}

GET /employees/findAll {model}

GET /employees/findFirstWhereContainsAllIgnoreCase {firstNameContainsIgnoreCase}

GET /employees/find.byId {id}

POST /employees/save {model}

GET /employees/sortByFirstNameDirection {sortByFirstNameDirection}

PUT /employees/updateById {updateModel}

Parameters

Name	Description
employee *	employee
(model)	Example value : Model
Response content type	application/json

Responses

Code	Description
200	OK
201	Created
401	Unauthorized
403	Forbidden
404	Not Found

curl
curl -X PUT "http://localhost:8080/employees/updateById" -H "accept: */*" -H "Content-Type: application/json" -d " {"email": "anushka@greatlearning.in", "firstName": "Anushka", "lastName": "Chaturvedi", "modelId": 5, "password": "Cheetured1"}"

Request URL
<http://localhost:8080/employees/updateById>

Server response

Code	Details
200	Response body

```
{
  "id": 5,
  "firstName": "Anushka",
  "lastName": "Chaturvedi",
  "modelId": 5,
  "password": "Cheetured1"
}
```

Responses

Code	Description
200	OK
201	Created
401	Unauthorized
403	Forbidden
404	Not Found

role-controller Role API

user-controller User API

Models

➤ Delete by id

EmployeeManagement API

employee-controller Employee API

DELETE /employees/deleteById `employeeId`

Parameters

Name	Description
<code>employeeId</code> <small>required</small>	employed

Responses

Code	Description
200	ok
204	No Content
401	Unauthorized
403	Forbidden

Code Examples

- `GET /employees/findAll`
- `GET /employees/findByNameContainsAllIgnoreCase`
- `GET /employees/findById employeeId`
- `POST /employees/save new`
- `GET /employees/sortByFirstNameDirection sortByFirstNameDirection`
- `PUT /employees/updateById updateId`

role-controller Role API

user-controller User API

Models

EmployeeManagement API

employee-controller Employee API

DELETE /employees/deleteById `employeeId`

Parameters

Name	Description
<code>employeeId</code> <small>required</small>	employed

Responses

Code	Description
200	ok
204	No Content
401	Unauthorized
403	Forbidden

Curl Example:

```
curl -X DELETE "http://localhost:8080/employees/deleteById?employeeId=5" -H "Accept: */*
```

Request URL: <http://localhost:8080/employees/deleteById?employeeId=5>

Server response

Code	Details
200	<p>Response body</p> <pre>Deleted employee Id = 5</pre> <p>Response headers</p> <pre>cache-control: no-cache, no-store, max-age=0, must-revalidate content-type: application/json content-length: 23 date: Mon, 01 Mar 2022 06:59:32 GMT date: Sat, 01 Jan 1970 00:00:00 GMT keep-alive: timeout=60 vary: Origin, Access-Control-Request-Method, Access-Control-Request-Headers x-content-type-options: nosniff x-frame-options: SAMEORIGIN x-xss-protection: 1; mode=block</pre>

Responses

Code	Description
200	ok
204	No Content
401	Unauthorized
403	Forbidden

Code Examples

- `GET /employees/findAll`
- `GET /employees/findByNameContainsAllIgnoreCase`
- `GET /employees/findById employeeId`
- `POST /employees/save new`
- `GET /employees/sortByFirstNameDirection sortByFirstNameDirection`
- `PUT /employees/updateById updateId`

role-controller Role API

user-controller User API

Models

• Role

➤ List All

The screenshot shows two side-by-side views of the EmployeeManagement API documentation using the Swagger interface.

Left View (EmployeeManagement API v1):

- Role Controller:** Shows the `/roles/deleteById` endpoint with a `DELETE` method. It includes a curl example and a response section with status codes 200, 401, 403, 404, and 405.
- User Controller:** Shows the `/roles/findAll` endpoint with a `GET` method. It includes a curl example and a response section with status codes 200, 401, 403, 404, and 405.
- Models:** A section for defining data models.

Right View (EmployeeManagement API v2):

- Role Controller:** Shows the `/roles/deleteById` endpoint with a `DELETE` method. It includes a curl example and a response section with status codes 200, 401, 403, 404, and 405.
- User Controller:** Shows the `/roles/findAll` endpoint with a `GET` method. It includes a curl example and a response section with status codes 200, 401, 403, 404, and 405.
- Models:** A section for defining data models.

The interface includes tabs for selecting the API version (v1 or v2) and provides links to the API's GitHub repository, terms of service, and license information.

➤ Find by id

The screenshot shows the EmployeeManagement API's roles endpoint in the Swagger UI. It highlights the `DELETE /roles/deleteById` method. The parameters section shows `roleId` as required, with a type of `integer`. The responses section shows four status codes: 200 (OK), 401 (Unauthorized), 403 (Forbidden), and 404 (Not Found). Below the method details, there are buttons for `POST /roles/save` and `PUT /roles/updateById`.

The screenshot shows the EmployeeManagement API's roles endpoint in the Swagger UI, specifically the `GET /roles/findById` method. It requires a parameter `roleId` of type `integer`. The responses section includes a detailed example for the 200 OK response, which includes headers like `Content-Type: application/json`, `Cache-Control: no-cache, no-store, max-age=0, must-revalidate`, and `Expires: 0`, along with a JSON object example: `{"id": 1, "name": "ADMIN"}`. Other status codes shown are 401 (Unauthorized), 403 (Forbidden), and 404 (Not Found).

➤ Add

EmployeeManagement API

employee-controller Employee API

role-controller Role API

user-controller User API

Models

EmployeeManagement API

employee-controller Employee API

role-controller Role API

user-controller User API

Models

➤ Update by id

EmployeeManagement API

employee-controller Employee API

role-controller Role API

Responses

Code	Description
200	OK
201	Created
401	Unauthorized
403	Forbidden
404	Not Found

EmployeeManagement API

employee-controller Employee API

role-controller Role API

Responses

Code	Description
200	OK
201	Created
401	Unauthorized
403	Forbidden
404	Not Found

➤ Delete by id

EmployeeManagement API

`DELETE /roles/deleteById roleId`

Parameters

Name	Description
roleId	roleId

Responses

Code	Description	Response content type
200	OK	*/*
204	No Content	
401	Unauthorized	
403	Forbidden	

`GET /roles/findAll findAll`

`GET /roles/findById findById`

`POST /roles/save save`

`PUT /roles/updateById updateById`

employee-controller Employee API

role-controller Role API

Models

EmployeeManagement API

`DELETE /roles/deleteById roleId`

Parameters

Name	Description
roleId	roleId

Responses

Code	Description	Response content type
200	Deleted role id = 3	*/*
204	No Content	
401	Unauthorized	
403	Forbidden	

`curl -X DELETE "http://localhost:8080/roles/deleteById?roleId=3" -H "accept: */*`

Request URL

`http://localhost:8080/roles/deleteById?roleId=3`

Server response

Code **Details**

Code	Details
200	Deleted role id = 3

Responses

Code	Description
200	OK
204	No Content
401	Unauthorized
403	Forbidden

`GET /roles/findAll findAll`

`GET /roles/findById findById`

`POST /roles/save save`

`PUT /roles/updateById updateById`

employee-controller Employee API

role-controller Role API

Models

• User

➤ List All

The image displays two identical screenshots of the EmployeeManagement API's Swagger UI interface, illustrating the 'users' endpoint.

EmployeeManagement API (localhost:8080)

Endpoints:

- employee-controller** (Employee API)
 - DELETE /users/{deletedId}**: deletedId
 - GET /users/findAll**: findAll
- role-controller** (Role API)
- user-controller** (User API)
 - DELETE /users/{deletedId}**: deletedId
 - GET /users/findAll**: findAll

Responses:

Code Description Example Value (JSON)

200	OK	[Redacted]
401	Unauthorized	[Redacted]
403	Forbidden	[Redacted]
404	Not Found	[Redacted]

Code Description Example Value (JSON)

200	OK	[Redacted]
401	Unauthorized	[Redacted]
403	Forbidden	[Redacted]
404	Not Found	[Redacted]

Models:

The interface includes sections for Response headers (with examples like cache-control: no-cache, no-store, max-age=0, must-revalidate, content-type: application/json, pragma: no-cache, expires: 0, etag: '1234567890', if-none-match: '1234567890', if-modified-since: checked, vary: Origin, Access-Control-Request-Method, Access-Control-Request-Headers, x-frame-options: DENY, x-xss-protection: 1; mode=block) and Response models (User).

➤ Find by id

EmployeeManagement API

`curl -X GET "http://localhost:8080/users/findById?userId=1"`

`GET /users/findById userId`

Parameters

Name	Description
userId	integer (Required)

Responses

Code	Description
200	<code>OK</code>
401	<code>Unauthorized</code>
403	<code>Forbidden</code>
404	<code>Not Found</code>

`POST /users/save save`

`PUT /users/updateById updateById`

Models

EmployeeManagement API

`curl -X DELETE "http://localhost:8080/users/deleteById?userId=1"`

`DELETE /users/deleteById userId`

Parameters

Name	Description
userId	integer (Required)

Responses

Code	Description
200	<code>OK</code>
401	<code>Unauthorized</code>
403	<code>Forbidden</code>
404	<code>Not Found</code>

`POST /users/save save`

`PUT /users/updateById updateById`

➤ Add

EmployeeManagement API

employee-controller Employee API

role-controller Role API

user-controller User API

DELETE /users/deleteById deleteId

GET /users/findAll findAll

GET /users/findById findId

POST /users/save save

Parameters

Name	Description
password *	password
userRoles *	userRoles
username *	username

Responses

Code	Description
200	OK
201	Created
401	Unauthorized
403	Forbidden
404	Not Found

PUT /users/updateById updateId

Models

EmployeeManagement API

employee-controller Employee API

role-controller Role API

user-controller User API

DELETE /users/deleteById deleteId

GET /users/findAll findAll

GET /users/findById findId

POST /users/save save

Parameters

Name	Description
password *	password
userRoles *	userRoles

Responses

Code	Description
200	OK
201	Created
401	Unauthorized
403	Forbidden
404	Not Found

PUT /users/updateById updateId

Models

➤ Update by id

EmployeeManagement API

employee-controller Employee API

role-controller Role API

user-controller User API

DELETE /users/deleteById deletedId

GET /users/findAll findAll

GET /users/findById findById

POST /users/save save

PUT /users/updateById updateById

Parameters

password	Description
password	password
userId	userId
userRoles	userRoles
username	username

Responses

Code	Description
200	ok
201	Created
401	Unauthorized
403	Forbidden
404	Not Found

EmployeeManagement API

employee-controller Employee API

role-controller Role API

user-controller User API

DELETE /users/deleteById deletedId

GET /users/findAll findAll

GET /users/findById findById

POST /users/save save

PUT /users/updateById updateById

Parameters

password	Description
password	password
userId	userId
userRoles	userRoles
username	username

Responses

Code	Description
200	ok
201	Created
401	Unauthorized
403	Forbidden
404	Not Found

➤ Delete by id

EmployeeManagement API

EmployeeManagement API for developers

Try it out!

employee-controller Employee API

role-controller Role API

user-controller User API

DELETE /users/{userId}

Parameters

Name	Description
userId *	userId

Responses

Code	Description	Response content type
200	ok	*/*
204	No Content	*/*
401	Unauthorized	*/*
403	Forbidden	*/*

curl -X DELETE "http://localhost:8080/users/{userId}" -H "accept: */*"

GET /users/findAll

GET /users/findById {userId}

POST /users/save save

PUT /users/updateById {updateId}

Models

EmployeeManagement API

EmployeeManagement API for developers

Try it out!

employee-controller Employee API

role-controller Role API

user-controller User API

DELETE /users/{userId}

Parameters

Name	Description
userId *	userId

Value: 3

Execute Clear

Responses

Code	Description	Response content type
200	ok	*/*

Curl

curl -X DELETE "http://localhost:8080/users/{userId}" -H "accept: */*"

Request URL: http://localhost:8080/users/{userId}=3

Server response

Code Details

200 Response body

Deleted user id = 3

Download

Responses

Code	Description
200	ok
204	No Content
401	Unauthorized
403	Forbidden

curl -X DELETE "http://localhost:8080/users/{userId}" -H "accept: */*"

GET /users/findAll

GET /users/findById {userId}

POST /users/save save

PUT /users/updateById {updateId}

Models

5. User Operations

• Employee

➤ List All

The screenshot shows two side-by-side views of the EmployeeManagement API's Swagger UI interface.

Left View (EmployeeController API):

- DELETE /employees/deleteById {id}**: Shows a red highlighted row.
- GET /employees/findAll {field}**: Shows a blue highlighted row.
- Responses**:
 - 200**: Example value:

```
[{"name": "string", "id": "string", "age": 0, "lastName": "string"}]
```
 - 401**: Unauthorized
 - 403**: Forbidden
 - 404**: Not Found
- GET /employees/findByFirstNameContainsAllIgnoreCase {firstName}**
- GET /employees/findById {id}**
- POST /employees/save {av}**
- PUT /employees/sortByFirstNameDirection {sortByFirstNameDirection}**
- PUT /employees/updateById {updatedId}**: Shows an orange highlighted row.

Right View (EmployeeController API):

- DELETE /employees/deleteById {id}**: Shows a red highlighted row.
- GET /employees/findAll {field}**: Shows a blue highlighted row.
- Responses**:
 - 200**: Response body:

```
[{"id": 1, "name": "string", "lastName": "string", "age": 0, "email": "string"}, {"id": 2, "name": "string", "lastName": "string", "age": 0, "email": "string"}, {"id": 3, "name": "string", "lastName": "string", "age": 0, "email": "string"}, {"id": 4, "name": "string", "lastName": "string", "age": 0, "email": "string"}]
```

Code comments:

```
code.controller("EmployeeController", ["$scope", "$http", "localStorage", "maxAge", "mustRevalidate", "connection", "keepalive", "keepaliveTime", "keepaliveCount", "keepaliveInterval", "keepaliveTimeout", "keepaliveEnabled", "keepaliveRequesting", "keepaliveLastRequest", "keepaliveLastResponse", "keepaliveLastError", "keepaliveLastRequestTime", "keepaliveLastResponseTime", "keepaliveLastErrorTime", "keepaliveLastRequestMethod", "Access-Control-Request-Headers", "Access-Control-Request-Method", "Access-Control-Request-Header", "x-content-type-options", "x-nosniff", "x-xss-protection", "moder-block"])
```
 - 401**: Unauthorized
 - 403**: Forbidden
 - 404**: Not Found
- GET /employees/findByFirstNameContainsAllIgnoreCase {firstName}**
- GET /employees/findById {id}**
- POST /employees/save {av}**
- PUT /employees/sortByFirstNameDirection {sortByFirstNameDirection}**
- PUT /employees/updateById {updatedId}**: Shows an orange highlighted row.

➤ Find by First Name

swagger Select a spec EmployeeManagement API

EmployeeManagement API ⓘ
[Base URL: localhost:8080]
https://localhost:8080/api/doc/api.html#/EmployeeManagementAPI

EmployeeManagement API for developers
Terms of service
EmployeeManagement API - website
Send email to EmployeeManagement API
EmployeeManagement License

employee-controller Employee API

DELETE /employees/deleteById deleteById

GET /employees/findAll findAll

GET /employees/findbyFirstNameContainsAllIgnoreCase findByFirstNameContainsIgnoreCase

Parameters

Name Description
firstName * required string (query)

Responses

Code Description
200 OK

Example Value (Model)

```
{ "id": 1, "firstName": "string", "lastName": "string", "middleName": "string" }
```

401 Unauthorized

403 Forbidden

404 Not Found

GET /employees/findbyId findById

POST /employees/save save

GET /employees/sortByFirstNameDirection sortByFirstNameDirection

PUT /employees/updateById updateById

role-controller Role API

user-controller User API

Models

swagger Select a spec EmployeeManagement API

EmployeeManagement API ⓘ
[Base URL: localhost:8080]
https://localhost:8080/api/doc/api.html#/EmployeeManagementAPI

EmployeeManagement API for developers
Terms of service
EmployeeManagement API - website
Send email to EmployeeManagement API
EmployeeManagement License

employee-controller Employee API

DELETE /employees/deleteById deleteById

GET /employees/findAll findAll

GET /employees/findbyFirstNameContainsIgnoreCase findByFirstNameContainsIgnoreCase

Parameters

Name Description
firstName * required string (query)

Responses

Code Details
200 Response body

Request URL <http://localhost:8080/employees/findbyFirstNameContainsIgnoreCase?firstName=arkit>

Server response

```
[ { "id": 2, "firstName": "arkit", "lastName": "ver", "middleName": "weltgreen" } ]
```

Response headers

```
cache-control: no-cache, no-store, max-age=0, must-revalidate  
connection: keep-alive  
content-type: application/json  
date: Sat, 01 Jun 2013 04:04:04 GMT  
pragma: no-cache  
status: 200  
vary: Accept  
x-frame-options: DENY  
x-xss-protection: 1; mode=block
```

200 OK

Example Value (Model)

```
{ "id": 2, "firstName": "arkit", "lastName": "ver", "middleName": "weltgreen" }
```

401 Unauthorized

403 Forbidden

404 Not Found

GET /employees/findbyId findById

POST /employees/save save

GET /employees/sortByFirstNameDirection sortByFirstNameDirection

PUT /employees/updateById updateById

role-controller Role API

user-controller User API

Models

➤ Find by Id

EmployeeManagement API

employee-controller Employee API

DELETE /employees/deleteById employeeId

GET /employees/findall

GET /employees/findbyFirstNameContainsAllIgnoreCase firstName containsAllIgnoreCase

GET /employees/findById findById

Parameters

Name	Description
employeeId * integer	employeeId

Responses

Code	Description
200	OK
401	Unauthorized
403	Forbidden
404	Not Found

POST /employees/save save

GET /employees/sortByFirstNameDirection sortByFirstNameDirection

PUT /employees/updateById updateById

role-controller Role API

user-controller User API

Models

EmployeeManagement API

employee-controller Employee API

DELETE /employees/deleteById employeeId

GET /employees/findall

GET /employees/findbyFirstNameContainsAllIgnoreCase firstName containsAllIgnoreCase

GET /employees/findById findById

Parameters

Name	Description
employeeId * integer	employeeId

Responses

Code	Description
200	OK
401	Unauthorized
403	Forbidden
404	Not Found

POST /employees/save save

GET /employees/sortByFirstNameDirection sortByFirstNameDirection

PUT /employees/updateById updateById

role-controller Role API

user-controller User API

Models

➤ Sort by First Name & Direction

The image displays three side-by-side screenshots of the EmployeeManagement API documentation in the Swagger UI interface. Each screenshot shows the same endpoint: `/employees`.

Left Screenshot: Shows the `/employees` endpoint with a sorting parameter. The URL is `http://localhost:8080/api/v1/employees?sort=firstName,asc`. The description indicates it sorts employees by first name in ascending order.

Middle Screenshot: Shows the `/employees` endpoint with a sorting parameter. The URL is `http://localhost:8080/api/v1/employees?sort=lastName,desc`. The description indicates it sorts employees by last name in descending order.

Right Screenshot: Shows the `/employees` endpoint with a sorting parameter. The URL is `http://localhost:8080/api/v1/employees?sort=middleName,asc`. The description indicates it sorts employees by middle name in ascending order.

All three screenshots show the same detailed API documentation, including examples, responses, and code snippets for both GET and POST requests.

➤ Add

The screenshot shows the EmployeeManagement API documentation for the employee-controller. It includes a list of endpoints such as /employees/deleteById, /employees/findAll, /employees/findByNameContainsIgnoreCase, /employees/findId, /employees/save, and /employees/sortByFirstNameDirection. Each endpoint has its method, URL, and a detailed description of its parameters and responses.

The screenshot shows the EmployeeManagement API documentation for the employee-controller. It includes a list of endpoints such as /employees/deleteById, /employees/findAll, /employees/findByNameContainsIgnoreCase, /employees/findId, /employees/save, and /employees/sortByFirstNameDirection. Each endpoint has its method, URL, and a detailed description of its parameters and responses.

➤ Update by id

The screenshot shows the EmployeeManagement API's employee-controller endpoint. It highlights the **PUT /employees/:updateId** operation. A modal window is open, showing a JSON example for updating an employee's first name:

```
{
  "firstName": "John"
}
```

The modal also includes fields for 'Parameter content type' (application/json) and 'Response content type' (application/json).

Below the operation, there are response examples for various status codes (200, 201, 401, 403, 404) with their respective descriptions and sample responses.

The screenshot shows the EmployeeManagement API's employee-controller endpoint. It highlights the **PUT /employees/:updateId** operation. A modal window is open, showing a JSON example for updating an employee's first name:

```
{
  "firstName": "John"
}
```

The modal also includes fields for 'Parameter content type' (application/json) and 'Response content type' (application/json).

Below the operation, there are response examples for various status codes (200, 201, 401, 403, 404) with their respective descriptions and sample responses.

➤ Delete by id

EmployeeManagement API

employee-controller Employee API

DELETE /employees/deleteById : deleteId

Parameters

employeeId	required
Type	integer (int32)
Description	employed

Responses

Code	Description
200	ok
Example Value	"string"

204	No Content
-----	------------

401	Unauthorized
-----	--------------

403	Forbidden
-----	-----------

GET /employees/findAll : findAll

GET /employees/findByNameContainsAllIgnoreCase : findByNameContainsIgnoreCase

GET /employees/findById : findById

POST /employees/save : save

GET /employees/sortByFirstNameDirection : sortByFirstNameDirection

PUT /employees/updateById : updateId

role-controller Role API

user-controller User API

Models

EmployeeManagement API

employee-controller Employee API

DELETE /employees/deleteById : deleteId

Parameters

employeeId	required
Type	integer (int32)
Description	employed

Responses

Code	Description
200	ok
Example Value	"string"

204	No Content
-----	------------

401	Unauthorized
-----	--------------

403	Forbidden
-----	-----------

Curl

```
curl -X DELETE "http://localhost:8080/employees/deleteById?employeeId=5" -H "accept: */*"
```

Request URL

Server response

Code

403	Error:
Response body	[{"message": "2022-01-01T07:35:00.443+00:00", "status": 403, "error": "Forbidden", "exception": "com.fasterxml.jackson.databind.JsonMappingException", "path": "/employees/deleteById"}]
Response headers	Cache-control: no-cache, no-store, max-age=0, must-revalidate Connection: keep-alive Content-Type: application/json Date: Sat, 01 Jan 2022 07:35:00 GMT Keep-Alive: timeout=60 Server: tomcat/9 Transfer-Encoding: chunked Vary: Origin, Access-Control-Request-Method, Access-Control-Request-Headers Access-Control-Allow-Methods: GET, POST Access-Control-Allow-Headers: Content-Type, origin, accept Access-Control-Max-Age: 3600 Access-Control-Expose-Headers: Content-Type Content-Type: application/json Content-Length: 21 Date: Sat, 01 Jan 2022 07:35:00 GMT Last-Modified: Sat, 01 Jan 2022 07:35:00 GMT Server: Apache-Tomcat/9.0.52 X-Content-Type-Options: nosniff X-Frame-Options: SAMEORIGIN X-XSS-Protection: 1; mode=block

Responses

Code	Description
200	ok
Example Value	"string"

204	No Content
-----	------------

401	Unauthorized
-----	--------------

403	Forbidden
-----	-----------

GET /employees/findAll : findAll

GET /employees/findByNameContainsAllIgnoreCase : findByNameContainsIgnoreCase

GET /employees/findById : findById

POST /employees/save : save

GET /employees/sortByFirstNameDirection : sortByFirstNameDirection

PUT /employees/updateById : updateId

role-controller Role API

user-controller User API

Models

• Role

➤ List All

The image displays two side-by-side screenshots of the EmployeeManagement API documentation in Swagger UI, specifically focusing on the 'roles' endpoint.

Left Screenshot (EmployeeManagement API v1):

- Base URL:** http://localhost:8080/api/doc/api/EmployeeManagement-API
- API Version:** v1
- Operations:**
 - DELETE /roles/deleteById**: Deletes a role by ID. Response status codes: 200 (OK), 401 (Unauthorized), 403 (Forbidden), 404 (Not Found).
 - GET /roles/findAll**: Finds all roles. Response status codes: 200 (OK), 401 (Unauthorized), 403 (Forbidden).
 - POST /roles/save**: Saves a new role. Response status codes: 200 (OK), 401 (Unauthorized), 403 (Forbidden).
 - PUT /roles/updateById**: Updates an existing role by ID. Response status codes: 200 (OK), 401 (Unauthorized), 403 (Forbidden), 404 (Not Found).
- Models:** A section for defining data models.

Right Screenshot (EmployeeManagement API v2):

- Base URL:** http://localhost:8080/api/doc/api/EmployeeManagement-API
- API Version:** v2
- Operations:**
 - DELETE /roles/deleteById**: Deletes a role by ID. Response status codes: 200 (OK), 401 (Unauthorized), 403 (Forbidden).
 - GET /roles/findAll**: Finds all roles. Response status codes: 200 (OK), 401 (Unauthorized), 403 (Forbidden).
 - POST /roles/save**: Saves a new role. Response status codes: 200 (OK), 401 (Unauthorized), 403 (Forbidden).
 - PUT /roles/updateById**: Updates an existing role by ID. Response status codes: 200 (OK), 401 (Unauthorized), 403 (Forbidden), 404 (Not Found).
- Server Response Headers:** Shows example headers for a successful response (status 200 OK).
- Responses:** A detailed breakdown of the response body and headers for each status code (200, 401, 403, 404).
- Code Examples:** curl command examples for each operation.

➤ Find by id

The screenshot shows the EmployeeManagement API's roles endpoint in the Swagger UI. It highlights the `DELETE /roles/deleteById` method. The parameters section shows `roleId` as required, with a type of `integer`. The responses section shows four status codes: 200 (OK), 401 (Unauthorized), 403 (Forbidden), and 404 (Not Found). Below the method details, there are buttons for `POST /roles/save` and `PUT /roles/updateById`.

The screenshot shows the EmployeeManagement API's roles endpoint in the Swagger UI, specifically the `GET /roles/findById` method. It requires a parameter `roleId` of type `integer`. The responses section includes a detailed example for the 200 OK response, which includes headers like `Content-Type: application/json`, `Cache-Control: no-cache, no-store, max-age=0, must-revalidate`, and `Expires: 0`, along with a JSON object example: `{"id": 1, "name": "ADMIN"}`. Other status codes shown are 401 (Unauthorized), 403 (Forbidden), and 404 (Not Found).

➤ Add

EmployeeManagement API

employee-controller Employee API

role-controller Role API

user-controller User API

Models

EmployeeManagement API

employee-controller Employee API

role-controller Role API

user-controller User API

Models

➤ Update by id

The screenshot shows the **EmployeeManagement API** interface in Swagger UI. The **roles** endpoint is selected. The **PUT /roles/updateById/{id}** method is highlighted. The request body example is shown as:

```
{
  "id": 3,
  "name": "string"
}
```

The **Responses** section shows responses for various status codes: 200 (OK), 201 (Created), 401 (Unauthorized), 403 (Forbidden), and 404 (Not Found).

This screenshot shows the same **EmployeeManagement API** interface, but with a different request body example:

```
{
  "id": 3,
  "name": "string"
}
```

The **Responses** section shows the same set of status codes: 200, 201, 401, 403, and 404.

➤ Delete by id

EmployeeManagement API

`DELETE /roles/deleteById :roleId`

Parameters

Name	Description
roleId	

Responses

Code	Description
200	OK
204	No Content
401	Unauthorized
403	Forbidden

Buttons: GET /roles/findAll, GET /roles/findById, POST /roles/save, PUT /roles/updateById

EmployeeManagement API

`DELETE /roles/deleteById :roleId`

Parameters

Name	Description
roleId	

Responses

Code	Description
200	OK
204	No Content
401	Unauthorized
403	Forbidden

Buttons: GET /roles/findAll, GET /roles/findById, POST /roles/save, PUT /roles/updateById

• User

➤ List All

The image displays two identical screenshots of the EmployeeManagement API's Swagger UI interface, illustrating the 'users' endpoint.

EmployeeManagement API (localhost:8080)

Endpoints:

- employee-controller** (Employee API)
 - DELETE /users/{deletedId}**: deletedId
 - GET /users/findAll**: findAll
- role-controller** (Role API)
- user-controller** (User API)
 - DELETE /users/{deletedId}**: deletedId
 - GET /users/findAll**: findAll

Responses:

Code Description Example Value (JSON)

200	OK	[Redacted]
401	Unauthorized	[Redacted]
403	Forbidden	[Redacted]
404	Not Found	[Redacted]

Code Description Example Value (JSON)

200	OK	[Redacted]
401	Unauthorized	[Redacted]
403	Forbidden	[Redacted]
404	Not Found	[Redacted]

Models:

The interface includes sections for Response headers (with examples like Cache-Control: no-cache, no-store, max-age=0, must-revalidate, Content-Type: application/json, Date: Mon, 27 Mar 2017 07:23:08 GMT, Expires: Mon, 27 Mar 2017 07:23:08 GMT, Pragma: no-cache, X-Content-Type-Options: nosniff, Vary: Origin, Access-Control-Request-Method: GET, Access-Control-Request-Headers: X-Frame-Options: DENY, X-Content-Type-Options: nosniff), Response body (with examples of JSON arrays for users and roles), and Response headers (with examples of HTTP headers).

➤ Find by id

EmployeeManagement API

employee-controller Employee API

role-controller Role API

user-controller User API

DELETE /users/deleteById deleteById

GET /users/findAll findAll

GET /users/findById findById

Parameters

Name	Description
userId * required	userId

Responses

Code	Description
200	OK
401	Unauthorized
403	Forbidden
404	Not Found

POST /users/save save

PUT /users/updateById updateById

Models

EmployeeManagement API

employee-controller Employee API

role-controller Role API

user-controller User API

DELETE /users/deleteById deleteById

GET /users/findAll findAll

GET /users/findById findById

Parameters

Name	Description
userId * required	userId

Responses

Code	Description
200	OK
401	Curl
403	Request URL
404	Server response

Curl

```
curl -X GET "http://localhost:8080/users/findById?userId=1" -H "accept: */*"
```

Request URL

http://localhost:8080/users/findById?userId=1

Server response

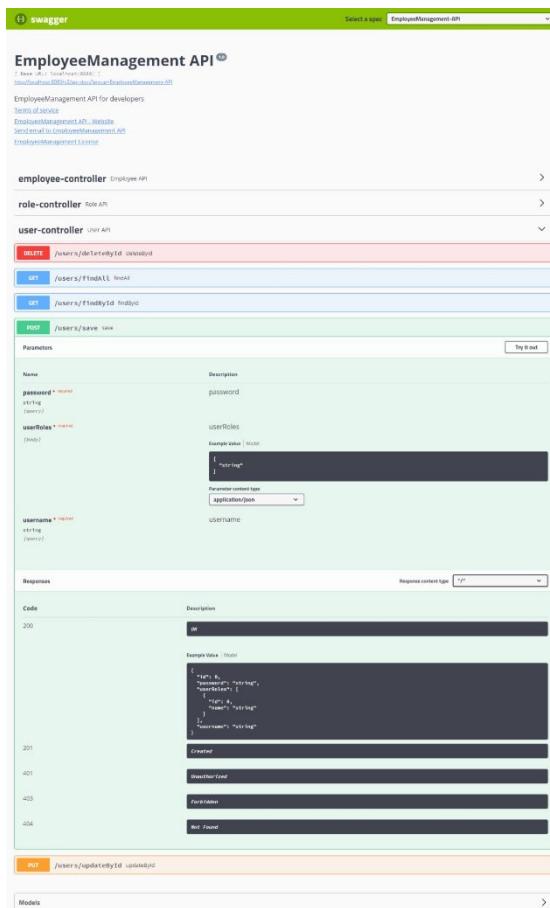
Code	Details
200	<p>Response body</p> <pre>{ "id": 1, "name": "John", "password": "\$2a\$10\$QwcaM8HNCuCt6QpxvGt11VlyxKegfddzrVwN.gherC", "userRole": [{ "id": 1, "name": "Admin" }] }</pre> <p>Download</p> <p>Response headers</p> <pre>cache-control: no-cache, no-store, max-age=0, must-revalidate connection: keep-alive content-type: application/json date: Sat, 01 Jun 2022 07:22:48 GMT keep-alive: timeout=60 transfer-encoding: chunked vary: Origin, X-Request-Method, Access-Control-Request-Headers x-content-type-options: nosniff x-frame-options: SAMEORIGIN x-xss-protection: 1; mode=block</pre>
200	<p>OK</p> <p>Example Value Model</p> <pre>{ "id": 1, "name": "John", "password": "string", "userRole": [{ "id": 1, "name": "string" }], "username": "string" }</pre>
401	Unauthorized
403	Forbidden
404	Not Found

POST /users/save save

PUT /users/updateById updateById

Models

➤ Add



employee-controller Employee API

role-controller Role API

user-controller User API

DELETE /users/deletebyId deleteId

GET /users/findAll findAll

GET /users/findByIdId findByIdId

POST /users/save save

Parameters

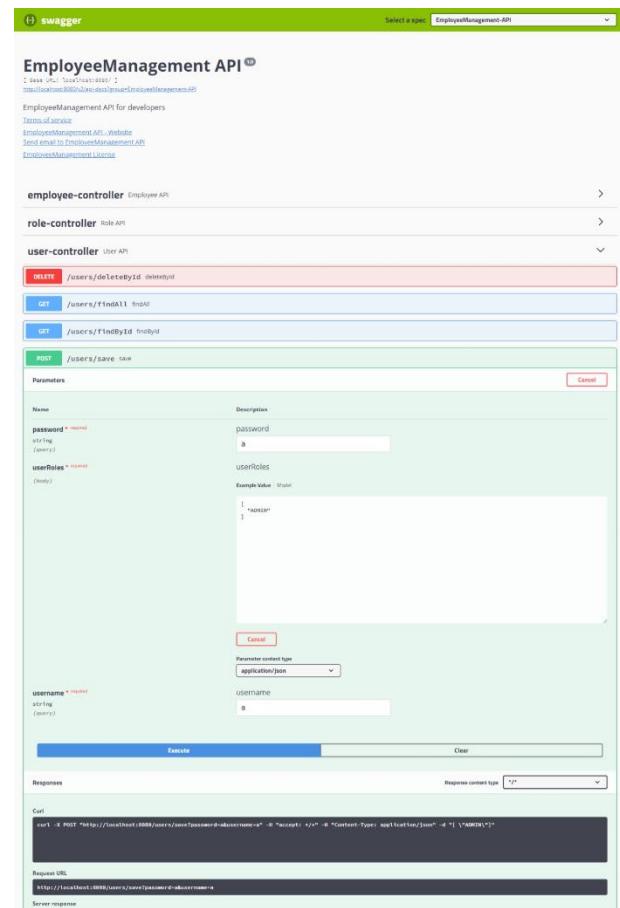
password *	password
username *	username
userRoles *	(body)

Responses

Code	Description
200	OK
Example Value	{ "id": 1, "name": "John Doe", "password": "123456", "userRole": [{ "id": 1, "name": "User", "parent": null }, { "id": 2, "name": "Admin", "parent": null }] }
201	Created
401	Unauthorized
403	Forbidden
404	Not Found

PUT /users/updatebyId updateId

Models



employee-controller Employee API

role-controller Role API

user-controller User API

DELETE /users/deletebyId deleteId

GET /users/findAll findAll

GET /users/findByIdId findByIdId

POST /users/save save

Parameters

password *	password
username *	username
userRoles *	(body)

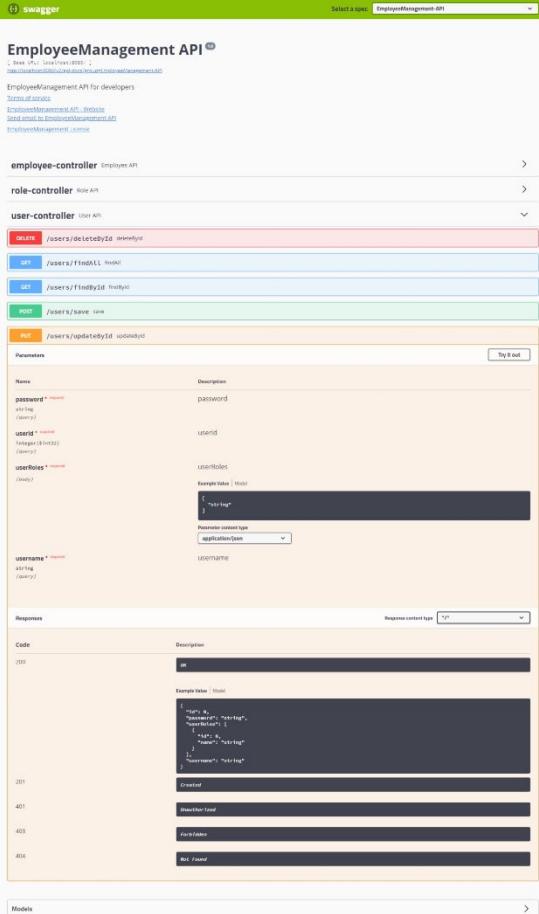
Responses

Code	Description
200	OK
Example Value	{ "id": 1, "name": "John Doe", "password": "123456", "userRole": [{ "id": 1, "name": "User", "parent": null }, { "id": 2, "name": "Admin", "parent": null }] }
201	Created
401	Unauthorized
403	Forbidden
404	Not Found

PUT /users/updatebyId updateId

Models

➤ Update by id



EmployeeManagement API

employee-controller Employee API

role-controller Role API

user-controller User API

DELETE /users/deletebyId deletebyId

GET /users/findall findAll

GET /users/findbyId findById

POST /users/save save

PUT /users/updatebyId updatebyId

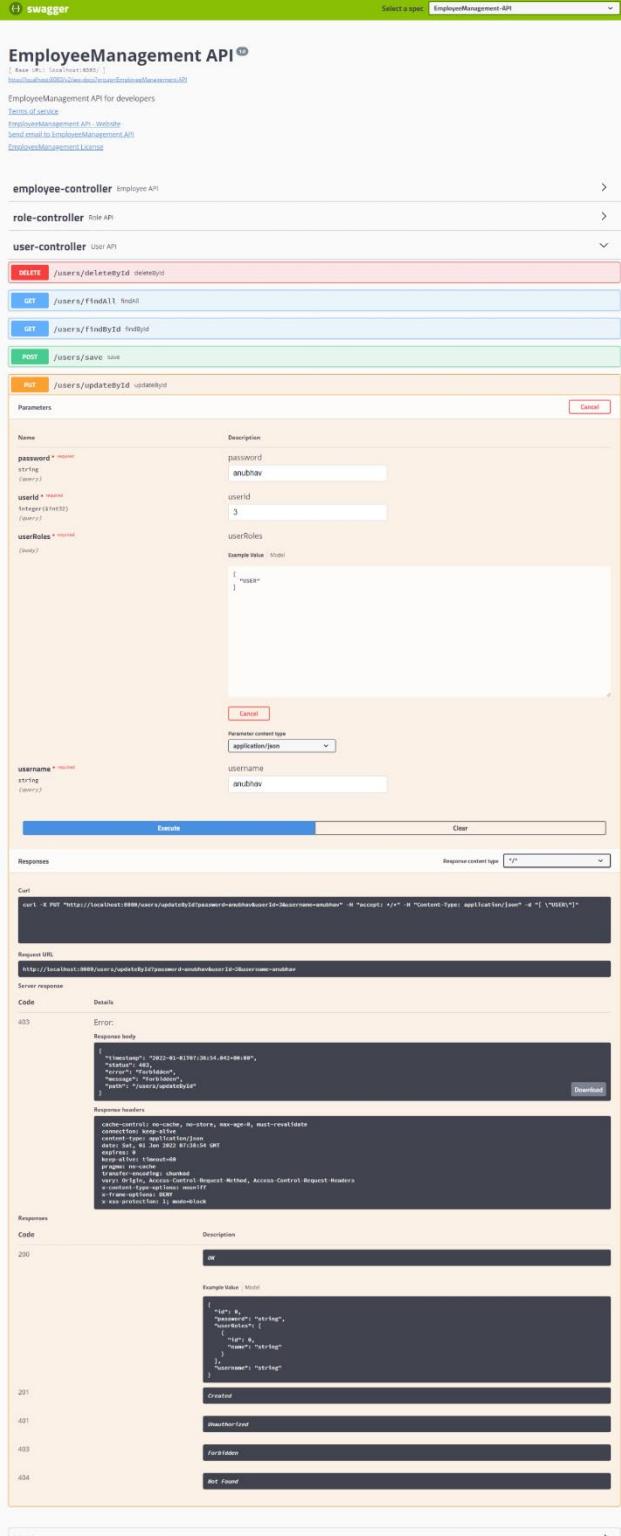
Parameters

password	Description
password	password
userId	userId
userRoles	userRoles
username	username

Responses

Code	Description
200	ok
Example Value	Model
201	Created
401	Unauthorized
403	Forbidden
404	Not Found

Models



EmployeeManagement API

employee-controller Employee API

role-controller Role API

user-controller User API

DELETE /users/deletebyId deletebyId

GET /users/findall findAll

GET /users/findbyId findById

POST /users/save save

PUT /users/updatebyId updatebyId

Parameters

password	Description
password	password
userId	userId
userRoles	userRoles
username	username

Responses

Code	Description
200	ok
Example Value	Model
201	Created
401	Unauthorized
403	Forbidden
404	Not Found

➤ Delete by id

EmployeeManagement API

DELETE /users/deleteById {userId}

Parameters

userId *	required
Integer(1 to 32)	(query)

Responses

Code	Description
200	OK
204	No Content
401	Unauthorized
403	Forbidden

Models

EmployeeManagement API

DELETE /users/deleteById {userId}

Parameters

userId *	required
Integer(3 int32)	(query)

Responses

Code	Description
200	OK
204	No Content
401	Unauthorized
403	Forbidden

Request URL

Server response

```

HTTP/1.1 200 OK
Content-Type: application/json
Date: Sat, 01 Jun 2022 07:30:57 GMT
Server: Apache/2.4.41 (Ubuntu)
Keep-Alive: timeout=60
Transfer-Encoding: chunked
Vary: Accept, Origin, X-Request-Method, Access-Control-Request-Headers
X-Content-Type-Options: nosniff
X-Xss-Protection: 1; mode=block

{
    "timestamp": "2022-01-01T07:30:57.473+00:00",
    "status": 403,
    "error": "Forbidden",
    "message": "User is deleted",
    "path": "/users/deleteById"
}
  
```

Responses

Code	Description
200	OK
204	No Content
401	Unauthorized
403	Forbidden

Models