

# 1. Employee management API

---

## 1.1. Login into store

### 1.1.1. Request

- Path: api/v1/login
- Method: POST
- Header:
  - Content-type: application/json
  - Authorization:
- Body:

```
{  
  "username": "admin",  
  "password": "admin@gmail.com"  
}
```

### 1.1.2. Response

- On success:

```
{  
  "success": true,  
  "id": 1,  
  "name": "admin",  
  "accessToken":  
  "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiIxIiwiaWF0IjoxNTQzMzM5NjExLCJleHAiOi0E1NDM5N  
  DQ0MTF9.sM5jlIKZbRivFXDhMIBgPf4vj-6-  
  7Prc33Ms9IE9fmWpdhVc_Q2GMoxdIvOKmz1SA061txj8DBUZPcHG3gHAhA",  
  "tokenType": "Bearer"  
}
```

- On fail (unauthorized):

```
{  
  ...  
  "status": 401,  
  "error": "Unauthorized",  
  "message": "Sorry, You're not authorized to access this resource.",  
}
```

```
    ...  
}
```

- On fail (one field is not in right format):

```
{  
    ...  
    "status": 400,  
    "error": "Bad Request",  
    ...  
}
```

## 1.2. Get your own summary profile

### 1.2.1. Request

- Path: api/v1/me/profile
- Method: GET
- Header:
  - Content-type:
  - Authorization: Bearer JWT
- Body:

### 1.2.2. Response

- On success:

```
{  
    "success": true,  
    "id": 4,  
    "username": "admin",  
    "name": null  
}
```

- On fail (unauthorized):

```
{  
    ...  
    "status": 401,  
    "error": "Unauthorized",  
    "message": "Sorry, You're not authorized to access this resource.",  
    ...  
}
```

## 1.3. Get your own information

### 1.3.1. Request

- Path: api/v1/me
- Method: GET
- Header:
  - Content-type:
  - Authorization: Bearer JWT
- Body:

### 1.3.2. Response

- On success:

```
{
  "success": true,
  "id": 4,
  "name": null,
  "username": "admin",
  "salary": null,
  "email": "admin@gmail.com",
  "address": null,
  "mobileNo": null,
  "roles": [
    "ROLE_ADMIN"
  ]
}
```

- On fail (unauthorized):

```
{
  ...
  "status": 401,
  "error": "Unauthorized",
  "message": "Sorry, You're not authorized to access this resource.",
  ...
}
```

## 1.4. Update your own information

### 1.4.1. Request

- Path: api/v1/me
- Method: PUT
- Header:
  - Content-type: application/json
  - Authorization: Bearer JWT
- Body: (you can choose any fields you want to change, all fields are not compulsory)

```
{
  "name": "Admin dep trai",
  "username": "admin",
  "email": "admin@gmail.com",
  "address": "Hanoi",
  "mobileNo": "0917915518"
}
```

#### 1.4.2. Response

- On Success:

```
{
  "success": true,
  "code": "update_successful",
  "message": "update profile successful"
}
```

- On fail (unauthorized):

```
{
  ...
  "status": 401,
  "error": "Unauthorized",
  "message": "Sorry, You're not authorized to access this resource.",
  ...
}
```

- On fail (one field is not in right format):

```
{
  ...
  "status": 400,
  "error": "Bad Request",
}
```

```
    ...  
}
```

## 1.5. Change your own password

### 1.5.1. Request

- Path: /me/change\_password
- Method: PUT
- Header:
  - Content-type: application/json
  - Authorization: Bearer JWT
- Body:

```
{  
    "oldPassword": "admin@gmail.com",  
    "newPassword": "admin@password"  
}
```

### 1.5.2. Response

- On success:

```
{  
    "success": true,  
    "id": 1,  
    "name": "admin",  
    "accessToken":  
    "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiIxIiwiaWF0IjoxNTQzMzM5NjExLCJleHAiOiE1NDM5N  
    DQ0MTF9. sM5jliKZbRivFXDhMIBgPf4vj-6-  
    7Prc33Ms9IE9fmWpdhVc_Q2GMoxdIvOKmz1SA061txj8DBUZPcHG3gHAhA",  
    "tokenType": "Bearer"  
}
```

- On fail (unauthorized):

```
{  
    ...  
    "status": 401,  
    "error": "Unauthorized",  
    "message": "Sorry, You're not authorized to access this resource.",  
    ...  
}
```

- On fail (old password is not correct):

```
{
  "success": false,
  "code": "change_password_fail",
  "message": "old password is not correct"
}
```

- On fail (one field is not in right format):

```
{
  ...
  "status": 400,
  "error": "Bad Request",
  ...
}
```

## 1.6. Delete your own account

### 1.6.1. Request

- Path: /api/v1/me
- Method: DELETE
- Header:
  - Content-type:
  - Authorization: Bearer JWT
- Body:

### 1.6.2. Response

- On success:

```
{
  "success": true,
  "code": "success",
  "message": "delete user successful"
}
```

- On fail (unauthorized):

```
{
  ...
  "status": 401,
  "error": "Unauthorized",
  "message": "Sorry, You're not authorized to access this resource.",
  ...
}
```

## 1.7. Create new account

### 1.7.1. Request

- Path: /api/v1/users
- Method: POST
- Header:
  - Content-type: application/json
  - Authorization: Bearer JWT
- Body: (username, email and password are compulsory, other fields can be skipped)

```
{
  "name": "ceo la",
  "username": "ceo",
  "password": "md5@password",
  "salary": 1000,
  "email": "ceo@gmail.com",
  "address": "hoa binh",
  "mobileNo": "0916167558",
  "roles": [
    "ROLE_ADMIN", "ROLE_MANAGER"
  ]
}
```

### 1.7.2. Response

- On success:

```
{
  "success": true,
  "id": 9
}
```

- On fail (unauthorized):

```
{
  ...
  "status": 401,
  "error": "Unauthorized",
  "message": "Sorry, You're not authorized to access this resource.",
  ...
}
```

- On fail (username is taken):

```
{
  "success": false,
  "code": "username_is_taken",
  "message": "Username is already taken!"
}
```

- On fail (email is taken):

```
{
  "success": false,
  "code": "email_is_taken",
  "message": "email is already taken!"
}
```

- On fail (one field is not in right format):

```
{
  ...
  "status": 400,
  "error": "Bad Request",
  ...
}
```

- On fail (role is not in list roleName):

```
{
  "success": false,
  "code": "something_wrong",
  "message": "something wrong with role field"
}
```

- On fail (you don't have role admin):



```
{
  ...
  "status": 403,
  "error": "Forbidden",
  "message": "Forbidden",
  ...
}
```

## 1.8. Get other user's information

### 1.8.1. Request

- Path: /api/v1/users/{id}
- Method: GET
- Header:
  - Content-type:
  - Authorization: Bearer JWT
- Body:

### 1.8.2. Response

- On success:

```
{
  "success": true,
  "id": 9,
  "name": "ceo la",
  "username": "ceo",
  "salary": 1000,
  "email": "ceo@gmail.com",
  "address": "hoa binh",
  "mobileNo": "0916167558",
  "roles": [
    "ROLE_MANAGER",
    "ROLE_ADMIN"
  ]
}
```

- On fail (unauthorized):

```
{
  ...
  "status": 401,
  "error": "Unauthorized",
  ...
}
```

```
    "message": "Sorry, You're not authorized to access this resource.",  
    ...  
}
```

- On fail (id does not exist):

```
{  
  "success": false,  
  "code": "something_wrong",  
  "message": "something wrong with user id"  
}
```

- On fail (you don't have role admin):

```
{  
  ...  
  "status": 403,  
  "error": "Forbidden",  
  "message": "Forbidden",  
  ...  
}
```

## 1.9. Update other user's information

### 1.9.1. Request

- Path: /api/v1/users/{id}
- Method: PUT
- Header:
  - Content-type: application/json
  - Authorization: Bearer JWT
- Body: (you can choose any fields you want to change, all fields are not compulsory)

```
{  
  "name": "ceo la la",  
  "username": "ceola",  
  "password": "md5@password"  
  "salary": 100,  
  "email": "ceola@gmail.com",  
  "address": "hai phong",  
  "mobileNo": "0916157558",  
  "roles": [  
    "ROLE_CASHIER",  
  ]  
}
```

```
        "ROLE_ADMIN"  
    ]  
}
```

### 1.9.2. Response

- On success:

```
{  
  "success": true,  
  "code": "update_successful",  
  "message": "update successful"  
}
```

- On fail (unauthorized):

```
{  
  ...  
  "status": 401,  
  "error": "Unauthorized",  
  "message": "Sorry, You're not authorized to access this resource.",  
  ...  
}
```

- On fail (one field is not in right format):

```
{  
  ...  
  "status": 400,  
  "error": "Bad Request",  
  ...  
}
```

- On fail (role is not in list roleName):

```
{  
  "success": false,  
  "code": "something_wrong",  
  "message": "something wrong with role field"  
}
```

- On fail (id does not exist):

```
{
  "success": false,
  "code": "something_wrong",
  "message": "something wrong with user id"
}
```

- On fail (you don't have role admin):

```
{
  ...
  "status": 403,
  "error": "Forbidden",
  "message": "Forbidden",
  ...
}
```

## 1.10. Get all users' information

### 1.10.1. Request

- Path: /api/v1/users
- Method: GET
- Header:
  - Content-type:
  - Authorization: Bearer JWT
- Body:

### 1.10.2. Response

- On success:

```
{
  "success": true,
  "users": [
    {
      "id": 1,
      "username": "admin",
      "name": "admin"
    }
  ]
}
```

- On fail (unauthorized):

```
{
  ...
  "status": 401,
  "error": "Unauthorized",
  "message": "Sorry, You're not authorized to access this resource.",
  ...
}
```

- On fail (you don't have role admin):

```
{
  ...
  "status": 403,
  "error": "Forbidden",
  "message": "Forbidden",
  ...
}
```

## 1.11. Delete other users's account

### 1.11.1. Request

- Path: /api/v1/users/{id}
- Method: DELETE
- Header:
  - Content-type:
  - Authorization: Bearer JWT
- Body:

### 1.11.2. Response

- On success:

```
{
  "success": true,
  "code": "success",
  "message": "delete user successful"
}
```

- On fail (unauthorized):

```
{
  ...
  "status": 401,
  "error": "Unauthorized",
  "message": "Sorry, You're not authorized to access this resource.",
  ...
}
```

- On fail (id does not exist):

```
{
  "success": false,
  "code": "something_wrong",
  "message": "something wrong with user id"
}
```

- On fail (you don't have role admin):

```
{
  ...
  "status": 403,
  "error": "Forbidden",
  "message": "Forbidden",
  ...
}
```

## 2. Customer management API

---

### 2.1. Create new customer

#### 2.1.1. Request

- Path: /api/v1/customers
- Method: POST
- Header:
  - Content-type: application/json
  - Authorization: Bearer JWT
- Body:(name field is compulsory, other fields can be skipped)

```
{
  "name": "customer_1",
  "email": "customer_1@gmail.com",
  "address": "ha noi",
  "mobileNo": "0916167887"
}
```

### 2.1.2. Response

- On success:

```
{
  "success": true,
  "id": 1
}
```

- On fail (unauthorized):

```
{
  ...
  "status": 401,
  "error": "Unauthorized",
  "message": "Sorry, You're not authorized to access this resource.",
  ...
}
```

- On fail (one field is not in right format):

```
{
  ...
  "status": 400,
  "error": "Bad Request",
  ...
}
```

- On fail (you don't have role admin or cashier):

```
{
  ...
  "status": 403,
  "error": "Forbidden",
  "message": "Forbidden",
  ...
}
```

## 2.2. Get all customer summary information

### 2.2.1. Request

- Path: /api/v1/customers
- Method: GET
- Header:
  - Content-type:
  - Authorization: Bearer JWT
- Body:

### 2.2.2. Response

- On success:

```
{
  "success": true,
  "customer": [
    {
      "id": 2,
      "name": "customer_2"
    },
    {
      "id": 1,
      "name": "customer_1"
    }
  ]
}
```

- On fail (unauthorized):

```
{
  ...
  "status": 401,
  "error": "Unauthorized",
  "message": "Sorry, You're not authorized to access this resource.",
  ...
}
```

- On fail (you don't have role admin or cashier):

```
{
  ...
}
```



```
"status": 403,  
"error": "Forbidden",  
"message": "Forbidden",  
...  
}
```

## 2.3. Get a customer's information

### 2.3.1. Request

- Path: /api/v1/customer/{customer\_id}
- Method: GET
- Header:
  - Content-type:
  - Authorization: Bearer JWT
- Body:

### 2.3.2. Response

- On success:

```
{  
  "success": true,  
  "id": 1,  
  "name": "customer_1",  
  "email": "customer_1@gmail.com",  
  "address": "ha noi",  
  "mobileNo": "0916167887"  
}
```

- On fail (unauthorized):

```
{  
  ...  
  "status": 401,  
  "error": "Unauthorized",  
  "message": "Sorry, You're not authorized to access this resource.",  
  ...  
}
```

- On fail (customer\_id does not exist):

```
{
  "success": false,
  "code": "something_wrong",
  "message": "something wrong with customer id"
}
```

- On fail (you don't have role admin or cashier):

```
{
  ...
  "status": 403,
  "error": "Forbidden",
  "message": "Forbidden",
  ...
}
```

## 2.4. Update a customer's information

### 2.4.1. Request

- Path: /api/v1/customers/{customer\_id}
- Method: PUT
- Header:
  - Content-type: application/json
  - Authorization: Bearer JWT
- Body: (you can choose any fields you want to change, all fields are not compulsory)

```
{
  "name": "customer_5",
  "email": "customer_5@gmail.com",
  "address": "ha noi",
  "mobileNo": "0916167997"
}
```

### 2.4.2. Response

- On success:

```
{
  "success": true,
  "code": "update_successful",
}
```

```
  "message": "update successful"
}
```

- On fail (unauthorized):

```
{
  ...
  "status": 401,
  "error": "Unauthorized",
  "message": "Sorry, You're not authorized to access this resource.",
  ...
}
```

- On fail (customer\_id does not exist):

```
{
  "success": false,
  "code": "something_wrong",
  "message": "something wrong with customer id"
}
```

- On fail (you don't have role admin or cashier):

```
{
  ...
  "status": 403,
  "error": "Forbidden",
  "message": "Forbidden",
  ...
}
```

## 2.5. Delete a customer

### 2.1.1. Request

- Path: /api/v1/customers/
- Method: DELETE
- Header:
  - Content-type:
  - Authorization: Bearer JWT
- Body:

## 2.1.2. Response

- On success:

```
{
  "success": true,
  "code": "success",
  "message": "delete customer successful"
}
```

- On fail (unauthorized):

```
{
  ...
  "status": 401,
  "error": "Unauthorized",
  "message": "Sorry, You're not authorized to access this resource.",
  ...
}
```

- On fail (customer\_id does not exist):

```
{
  "success": false,
  "code": "something_wrong",
  "message": "something wrong with customer id"
}
```

- On fail (you don't have role admin or cashier):

```
{
  ...
  "status": 403,
  "error": "Forbidden",
  "message": "Forbidden",
  ...
}
```