

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 dep trai",
  "username": "admin",
  "salary": null,
  "email": "admin@gmail.com",
  "address": "Hanoi",
  "mobileNo": "0917915518",
  "roles": [
    "ROLE_ADMIN"
  ],
  "tokenType": "Bearer",
  "accessToken":
"eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiIxIiwiaWF0IjoxNTQzNTEyNTYzLCJleHAiOiJE1NDQxMTczNjN9.S4zVTJY0ejIjTWbNni1Wo6bQijeRb9fkhs_mi8JAIBx_QbCCPe4Vo_ITbooRWtoDwlihUvfzq9wgFXQxwUJd2w"
}
```

- On fail (unauthorized):

```
{
  "success": false,
  "code": "unauthorized",
  "message": "You're not authorized to access this resource"
}
```

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

```
{
  "success": false,
  "code": "argument_not_valid",
  "message": {object_name} + {default_message} + "and" + ...
}
```

1.2. Get your own information

1.2.1. Request

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

1.2.2. Response

- On success:

```
{
  "success": true,
  "id": 1,
  "name": "Admin dep trai",
  "username": "admin",
  "salary": null,
  "email": "admin@gmail.com",
  "address": "Hanoi",
  "mobileNo": "0917915518",
  "roles": [
    "ROLE_ADMIN"
  ]
}
```

- On fail (unauthorized):

```
{
  "success": false,
  "code": "unauthorized",
  "message": "You're not authorized to access this resource"
}
```

1.3. Update your own information

1.3.1. 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"
}
```

1.3.2. Response

- On Success:

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

- On fail (unauthorized)

```
{
  "success": false,
  "code": "unauthorized",
}
```

```
{
  "message": "You're not authorized to access this resource"
}
```

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

```
{
  "success": false,
  "code": "argument_not_valid",
  "message": {object_name} + {default_message} + "and" + ...
}
```

1.4. Change your own password

1.4.1. Request

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

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

1.4.2. Response

- On success:

```
{
  "tokenType": "Bearer",
  "accessToken":
  "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiIxIiwiaWF0IjoxNTQzNTEyNjQxLCJleHAiOiE1NDQxMTc0NDY5LjUyMi3nwFVmNtJM-04dknYrCgm8eYe2utlj2k82SEj5lCFvzvypFKuLXxvdAS-0LEYn1DCvMH0Q-L_TlZ5idDww"
}
```

- On fail (unauthorized):

```
{
  "success": false,
  "code": "unauthorized",
  "message": "You're not authorized to access this resource"
}
```

- On fail (old password is not correct):

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

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

```
{
  "success": false,
  "code": "argument_not_valid",
  "message": {object_name} + {default_message} + "and" + ...
}
```

1.5. Delete your own account

1.5.1. Request

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

1.5.2. Response

- On success:

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

- On fail (unauthorized):

```
{
  "success": false,
  "code": "unauthorized",
  "message": "You're not authorized to access this resource"
}
```

1.6. Create new account

1.6.1. 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"
  ]
}
```

1.6.2. Response

- On success:

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

- On fail (unauthorized):

```
{
  "success": false,
  "code": "unauthorized",
  "message": "You're not authorized to access this resource"
}
```

- On fail (username is taken):

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

- On fail (email is taken):

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

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

```
{
  "success": false,
  "code": "argument_not_valid",
  "message": {object_name} + {default_message} + "and" + ...
}
```

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

```
{
  "success": false,
  "code": "wrong_role",
  "message": "role " + {wrong_role} + " does not exist"
}
```

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

```
{
  "success": false,
```

```
"code": "access_denied",
"message": "You don't have permission to access this resource"
}
```

1.7. Get other user's information

1.7.1. Request

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

1.7.2. 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):

```
{
  "success": false,
  "code": "unauthorized",
  "message": "You're not authorized to access this resource"
}
```

- On fail (id does not exist):


```
{
  "success": false,
  "code": "wrong_user_id",
  "message": "user id " + {id} + " does not exist"
}
```

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

```
{
  "success": false,
  "code": "access_denied",
  "message": "You don't have permission to access this resource"
}
```

1.8. Update other user's information

1.8.1. 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"
  ]
}
```

1.8.2. Response

- On success:

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

- On fail (unauthorized):

```
{
  "success": false,
  "code": "unauthorized",
  "message": "You're not authorized to access this resource"
}
```

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

```
{
  "success": false,
  "code": "argument_not_valid",
  "message": {object_name} + {default_message} + "and" + ...
}
```

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

```
{
  "success": false,
  "code": "wrong_role",
  "message": "role " + {wrong_role} + " does not exist"
}
```

- On fail (id does not exist):

```
{
  "success": false,
  "code": "wrong_user_id",
  "message": "user id " + {id} + " does not exist"
}
```

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

```
{
  "success": false,
```

```
"code": "access_denied",  
"message": "You don't have permission to access this resource"  
}
```

1.9. Get all users' information

1.9.1. Request

- Path: /api/v1/users?page=0&size=3
- Method: GET
- Header:
 - Content-type:
 - Authorization: Bearer JWT
- Body:

1.9.2. Response

- On success:

```
{  
  "success": true,  
  "users": [  
    {  
      "id": 1,  
      "name": "Admin dep trai",  
      "username": "admin",  
      "salary": null,  
      "email": "admin@gmail.com",  
      "address": "Hanoi",  
      "mobileNo": "0917915518",  
      "roles": [  
        "ROLE_ADMIN"  
      ]  
    },  
    {  
      "id": 4,  
      "name": "cashier",  
      "username": "cashier",  
      "salary": 1000,  
      "email": "cashier@gmail.com",  
      "address": "hoa binh",  
      "mobileNo": "0916167558",  
      "roles": [  
        "ROLE_CASHIER"  
      ]  
    },  
    {
```

```
        "id": 3,
        "name": "ceo la",
        "username": "ceo",
        "salary": 1000,
        "email": "ceo@gmail.com",
        "address": "hoa binh",
        "mobileNo": "0916167558",
        "roles": [
            "ROLE_MANAGER",
            "ROLE_ADMIN"
        ]
    }
]
```

- On fail (unauthorized):

```
{
  "success": false,
  "code": "unauthorized",
  "message": "You're not authorized to access this resource"
}
```

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

```
{
  "success": false,
  "code": "access_denied",
  "message": "You don't have permission to access this resource"
}
```

- On fail (page number is less than zero):

```
{
  "success": false,
  "code": "page_num_unacceptable",
  "message": "Page number cannot be less than zero."
}
```

- On fail (page size is greater than MAX_PAGE_SIZE):

```
{
  "success": false,
  "code": "page_size_unacceptable",
}
```

```
"message": "Page size must not be greater than 50"
}
```

1.10. Delete other users's account

1.10.1. Request

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

1.10.2. Response

- On success:

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

- On fail (unauthorized):

```
{
  "success": false,
  "code": "unauthorized",
  "message": "You're not authorized to access this resource"
}
```

- On fail (id does not exist):

```
{
  "success": false,
  "code": "wrong_user_id",
  "message": "user id " + {id} + " does not exist"
}
```

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

```
{
  "success": false,
  "code": "access_denied",
  "message": "You don't have permission to access this resource"
}
```

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):

```
{
  "success": false,
  "code": "unauthorized",
}
```

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

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

```
{
  "success": false,
  "code": "argument_not_valid",
  "message": {object_name} + {default_message} + "and" + ...
}
```

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

```
{
  "success": false,
  "code": "access_denied",
  "message": "You don't have permission to access this resource"
}
```

2.2. Get all customer summary information

2.2.1. Request

- Path: /api/v1/customers?page=0&size=3
- Method: GET
- Header:
 - Content-type:
 - Authorization: Bearer JWT
- Body:

2.2.2. Response

- On success:

```
{
  "success": true,
  "customers": [
    {
      "success": null,
      "id": 3,
      "name": "customer_3",
      "email": "customer_3@gmail.com",
      "address": "ha noi",
    }
  ]
}
```

```
        "mobileNo": "0916167887"
      },
      {
        "success": null,
        "id": 2,
        "name": "customer_2",
        "email": "customer_2@gmail.com",
        "address": "ha noi",
        "mobileNo": "0916167887"
      },
      {
        "success": null,
        "id": 1,
        "name": "customer_1",
        "email": "customer_1@gmail.com",
        "address": "ha noi",
        "mobileNo": "0916167887"
      }
    ]
  }
}
```

- On fail (unauthorized):

```
{
  "success": false,
  "code": "unauthorized",
  "message": "You're not authorized to access this resource"
}
```

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

```
{
  "success": false,
  "code": "access_denied",
  "message": "You don't have permission to access this resource"
}
```

2.3. Get a customer's information

2.3.1. Request

- Path: /api/v1/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):

```
{
  "success": false,
  "code": "unauthorized",
  "message": "You're not authorized to access this resource"
}
```

- On fail (id does not exist):

```
{
  "success": false,
  "code": "wrong_customer_id",
  "message": "customer id " + {id} + " does not exist"
}
```

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

```
{
  "success": false,
  "code": "access_denied",
  "message": "You don't have permission to access this resource"
}
```

2.4. Update a customer's information

2.4.1. Request

- Path: /api/v1/customers/{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):

```
{
  "success": false,
  "code": "unauthorized",
  "message": "You're not authorized to access this resource"
}
```

- On fail (id does not exist):

```
{
  "success": false,
  "code": "wrong_customer_id",
  "message": "customer id " + {id} + " does not exist"
}
```

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

```
{
  "success": false,
  "code": "access_denied",
  "message": "You don't have permission to access this resource"
}
```

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

```
{
  "success": false,
  "code": "argument_not_valid",
  "message": {object_name} + {default_message} + "and" + ...
}
```

2.5. Delete a customer

2.1.1. Request

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

2.1.2. Response

- On success:

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

- On fail (unauthorized):

```
{
  "success": false,
  "code": "unauthorized",
  "message": "You're not authorized to access this resource"
}
```

- On fail (id does not exist):

```
{
  "success": false,
  "code": "wrong_customer_id",
  "message": "customer id " + {id} + " does not exist"
}
```

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

```
{
  "success": false,
  "code": "access_denied",
  "message": "You don't have permission to access this resource"
}
```