**// https://localhost:7089/api/v1/users?page=1&pageSize=10**

**app.MapGet("/api/v1/users", UserEndpointsLogic.GetUsersLogicAsync)**

**.RequireAuthorization("EmployeeOrAdminWithUpdatedPasswordPolicy")**

**.WithName("GetUsers");**

* **Requires valid JWT token.**
* **Requires Claim ‘Admin’ or ‘Employee’ and ‘updated password’ = ‘true ‘.**
* **Authorized users: Admin, Employee.**
* **Retrieves all users matching filter (If Claim Admin). Retrieves all Clients matching filter( If Claim Employee).**
* **Can use query to filter for Firstname, Lastname, Phonenumber, Email.**
* **Includes pagination page&pageSize**

**// https://localhost:7089/api/v1/users/887ac10b-58cc-4372-a567-0e02b2c3d493**

**app.MapGet("/api/v1/users/{id}", UserEndpointsLogic.GetUserByIdLogicAsync)**

**.RequireAuthorization("AdminWithUpdatedPasswordPolicy")**

**.WithName("GetUserById");**

* **Requires valid JWT token.**
* **Requires Claim ‘updated password’ = ‘true ‘.**
* **Authorized users: Admin.**
* **Retrieves a single user by userId.**

**// https://localhost:7089/api/v1/users/b97ac10b-58cc-4372-a567-0e02b2c3d490 app.MapPut("/api/v1/users/{id}", UserEndpointsLogic.UpdateUserLogicAsync) .RequireAuthorization()**

**.EndpointValidationFilter<UserUpdateDTO>()**

**.WithName("UpdateUser");**

* **Requires valid JWT token.**
* **Requires Claim ‘updated password’ = ‘true ‘.**
* **Authorized users: Admin, Employee**
* **Updates a single user**
* **If Claim Admin: Can update all users.**
* **If Claim Employee: Can update only Clients.**
* **If Claim Client: Can update only self.**

**// https://localhost:7089/api/v1/users/6d7b1ca5-54f6-4859-a746-fc712d564128**

**app.MapDelete("/api/v1/users/{id}", UserEndpointsLogic.DeleteUserLogicAsync)**

**.RequireAuthorization("AdminWithUpdatedPasswordPolicy")**

**.WithName("DeleteUser");**

* **Requires valid JWT token.**
* **Requires Claim ‘updated password’ = ‘true ‘.**
* **Authorized users: Admin**
* **Deletes a single user or Employee. Admin cannot be deleted.**

**//** [**https://localhost:7089/api/v1/employees/**](https://localhost:7089/api/v1/employees/)**register**

**app.MapPost("/api/v1/employees/register", UserEndpointsLogic.RegisterEmployeeLogicAsync)**

**//.RequireAuthorization("AdminWithUpdatedPasswordPolicy")**

**.EndpointValidationFilter<UserRegistrationDTO>()**

**.WithName("RegisterEmployee");**

* **Requires valid JWT token.**
* **Requires Claim ‘updated password’ = ‘true ‘.**
* **Authorized users: Admin**
* **Registers a new Employee**

**// https://localhost:7089/api/v1/clients/**

**app.MapPost("/api/v1/clients/register", UserEndpointsLogic.RegisterClientLogicAsync)**

**.EndpointValidationFilter<UserRegistrationDTO>()**

**.WithName("RegisterClient");**

* **Authorized users: All**
* **Registers a new Client**

**// https://localhost:7089/api/v1/login**

**app.MapPost("/api/v1/login", UserEndpointsLogic.UserLoginLogicAsync)**

**.EndpointValidationFilter<LoginDTO>()**

**.WithName("UserLogin");**

* **Authorized users: All**
* **Login user returns valid JWT**

**// https://localhost:7089/api/v1/users/update-password**

**app.MapPost("/api/v1/users/update-password", UserEndpointsLogic.UpdatePasswordLogicAsync)**

**.RequireAuthorization()**

**.EndpointValidationFilter<UpdatePasswordDTO>()**

**.WithName("UpdatePassword");**

* **Requires valid JWT token.**
* **Authorized users: Admin, Employee, Client**
* **Updates a user’s password**
* **Any user can only update own password.**