CREATE ROLE

Overview

This API allows users with the "Create role" permission to create a new role. When creating a role, the user must provide basic information about the role (name, code) and a list of permissions (Privileges) that will be assigned to that role.

The system will:

1. Authenticate the user with the "Create role" permission.
2. Check if the RoleCode already exists.
3. Check if each PrivilegeId in the list is valid.
4. Save the new role to the Roles table.
5. Save the associations between roles and permissions to the RolePrivileges table.
6. Return information about the created role, including a list of assigned permissions.

Context API

To call this API, the client must first know what permissions are available. Client should call GET /api/privilege API to get list of PrivilegeDto(ID, Name, Description) and then use IDs from that list to create new roles.

API Specification

API	URL
POST	/api/role/create
Permission	[AuthorizeByPrivilege("Create role")]

Request sample

```
{
  "roleName": "Kỹ thuật viên Lab",
  "roleCode": "LAB_USER",
  "roleDescription": "Vai trò dành cho kỹ thuật viên thực hiện xét nghiệm.",
  "privileges": [
    {
      "privilegeId": 1,
      "privilegeName": "View Patient",
      "description": "Quyền xem thông tin bệnh nhân"
    },
    {
      "privilegeId": 5,
      "privilegeName": "Manage Sample",
      "description": "Quyền quản lý mẫu xét nghiệm"
    }
  ]
}
```

Field	Description	Data Type	Examples
roleName	Display name of the role.	string	Kỹ thuật viên Lab
roleCode	Unique identifier of the role (for logic).	string	LAB_USER

roleDescription	Describe the role in detail.	string	Vai trò dành cho...
privileges	List of permissions assigned to this role.	List<PrivilegeDto>	[{ "privilegeId": 1 }, ...]
privileges[].privilegeId	Unique ID of the permission to be assigned.	integer	1

Response sample

```
{
  "statusCode": 200,
  "message": "Role has been created successfully.",
  "data": {
    "roleName": "Kỹ thuật viên Lab",
    "roleCode": "LAB_USER",
    "roleDescription": "Vai trò dành cho kỹ thuật viên thực hiện xét nghiệm.",
    "privileges": [
      {
        "privilegeId": 1,
        "privilegeName": "View Patient",
        "description": "Quyền xem thông tin bệnh nhân"
      },
      {
        "privilegeId": 5,
        "privilegeName": "Manage Sample",
        "description": "Quyền quản lý mẫu xét nghiệm"
      }
    ]
  },
  "responsedAt": "2025-10-29T04:15:00.123Z"
}
```
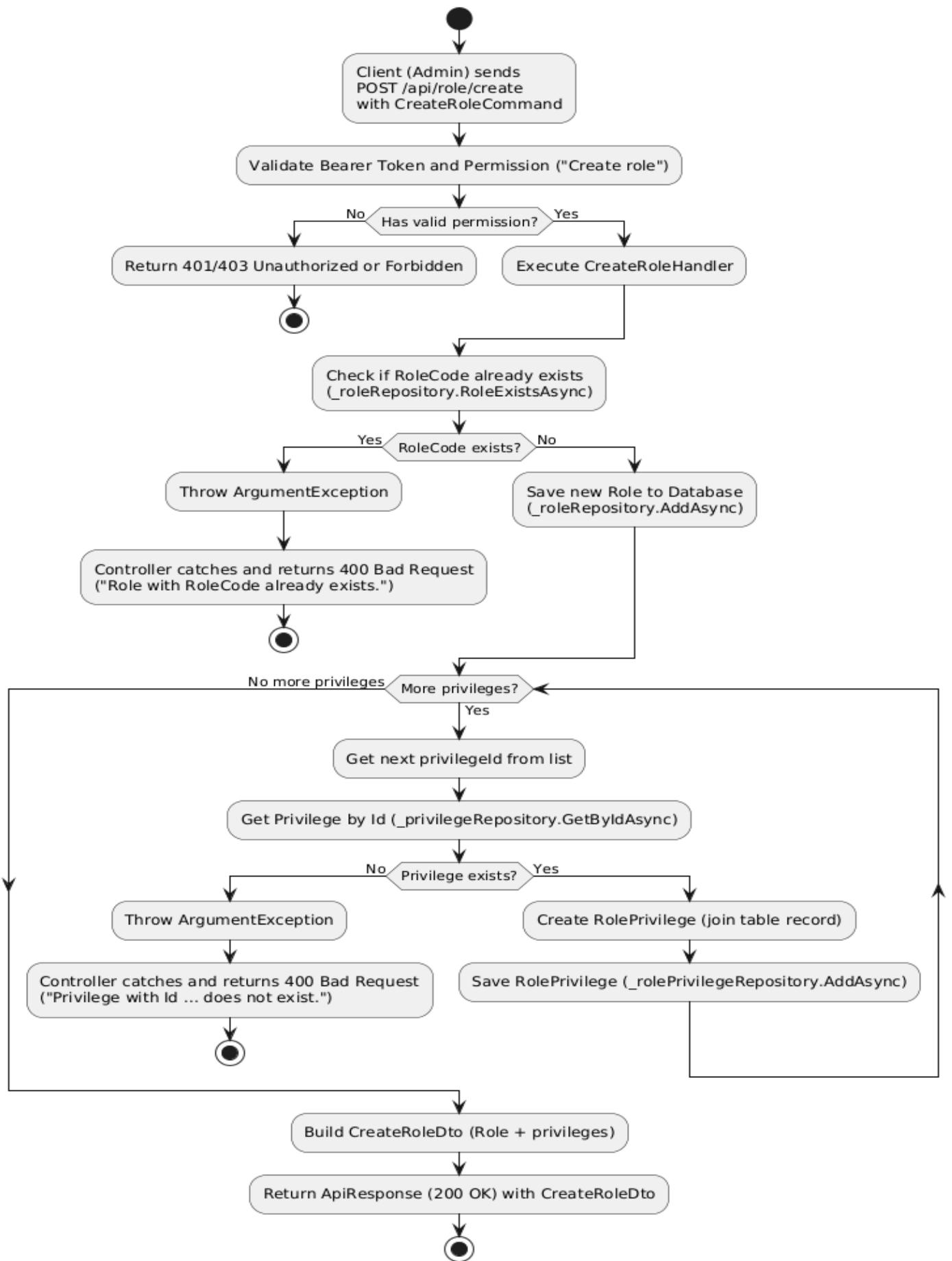
Validation

Status code	Description	Examples
400	Error: RoleCode already exists.	{ "statusCode": 400, "message": "Role with RoleCode 'LAB_USER' already exists.", ...}
400	Error: One of the PrivilegeId does not exist.	{ "statusCode": 400, "message": "Privilege with Id '99' does not exist.", ...}

| 401 | Unauthorized. User is not logged in. | Error: response status is 401 |

| 403 | Forbidden. The user is logged in but does not have the "Create role" permission. | Error: response status is 403 |

Activity Diagram

```
                            ●

                  ┌─────────────────────┐
                  │ Client (Admin) sends │
                  │ POST /api/role/create│
                  │ with CreateRoleCommand│
                  └─────────────────────┘
                            │
          ┌──────────────────────────────────────────┐
          │ Validate Bearer Token and Permission ("Create role") │
          └──────────────────────────────────────────┘
                            │
        No        ◇ Has valid permission? ◇        Yes
      ┌───────────                         ───────────┐
      │                                               │
┌──────────────────────────────┐        ┌──────────────────────────┐
│ Return 401/403 Unauthorized or Forbidden │   │ Execute CreateRoleHandler │
└──────────────────────────────┘        └──────────────────────────┘
      │                                               │
      ◉                                  ┌──────────────────────────────┐
                                         │ Check if RoleCode already exists │
                                         │ (_roleRepository.RoleExistsAsync)│
                                         └──────────────────────────────┘
                                                      │
                    Yes       ◇ RoleCode exists? ◇       No
              ┌──────────────                      ──────────────┐
              │                                                  │
    ┌──────────────────────┐                      ┌──────────────────────────┐
    │ Throw ArgumentException │                    │ Save new Role to Database │
    └──────────────────────┘                      │ (_roleRepository.AddAsync)│
              │                                    └──────────────────────────┘
    ┌──────────────────────────────────────┐                   │
    │ Controller catches and returns 400 Bad Request │          │
    │ ("Role with RoleCode already exists.")│                   │
    └──────────────────────────────────────┘                   │
              │                                                 │
              ◉                                                 │
                                                                │
   No more privileges   ◇ More privileges? ◇ ◄──────────────────┘
  ┌──────────────────                 
  │                       Yes │
  │               ┌──────────────────────────┐
  │               │ Get next privilegeId from list │
  │               └──────────────────────────┘
  │                           │
  │       ┌────────────────────────────────────────────┐
  │       │ Get Privilege by Id (_privilegeRepository.GetByIdAsync) │
  │       └────────────────────────────────────────────┘
  │                           │
  │        No      ◇ Privilege exists? ◇      Yes
  │      ┌──────────                      ──────────┐
  │      │                                          │
  │ ┌──────────────────────┐          ┌──────────────────────────────┐
  │ │ Throw ArgumentException │        │ Create RolePrivilege (join table record) │
  │ └──────────────────────┘          └──────────────────────────────┘
  │      │                                          │
  │ ┌──────────────────────────────────────┐  ┌──────────────────────────────────────────┐
  │ │ Controller catches and returns 400 Bad Request │ │ Save RolePrivilege (_rolePrivilegeRepository.AddAsync) │
  │ │ ("Privilege with Id ... does not exist.")│    └──────────────────────────────────────────┘
  │ └──────────────────────────────────────┘          │
  │      │                                             │
  │      ◉                                             │
  │                                                    │
  └────────────────────────┬───────────────────────────┘
                           │
              ┌──────────────────────────────┐
              │ Build CreateRoleDto (Role + privileges) │
              └──────────────────────────────┘
                           │
          ┌────────────────────────────────────────┐
          │ Return ApiResponse (200 OK) with CreateRoleDto │
          └────────────────────────────────────────┘
                           │
                           ◉
```

Sequence Diagram

Client → RoleController: POST /api/role/create (CreateRoleCommand)

[AuthorizeByPrivilege("Create role")]

RoleController → IMediator: Send(command)

IMediator → CreateRoleHandler: Handle(command)

CreateRoleHandler → IRoleRepository: RoleExistsAsync(command.RoleCode)

IRoleRepository → Database: SELECT * FROM Roles WHERE RoleCode = ?

Database → IRoleRepository: null

IRoleRepository → CreateRoleHandler: false

CreateRoleHandler → IRoleRepository: AddAsync(new Role)

IRoleRepository → Database: INSERT INTO Roles (...)

Database → IRoleRepository: Success

loop [For each privilege in command.Privileges]

CreateRoleHandler → IPrivilegeRepository: GetByIdAsync(priv.PrivilegeId)

IPrivilegeRepository → Database: SELECT * FROM Privileges WHERE Id = ?

Database → IPrivilegeRepository: PrivilegeEntity

IPrivilegeRepository → CreateRoleHandler: PrivilegeEntity

CreateRoleHandler → IRolePrivilegeRepository: AddAsync(new RolePrivilege)

IRolePrivilegeRepository → Database: INSERT INTO RolePrivileges (...)

Database → IRolePrivilegeRepository: Success

CreateRoleHandler → IMediator: CreateRoleDto

IMediator → RoleController: CreateRoleDto

RoleController → RoleController: new ApiResponse<CreateRoleDto>(200, "Success")

RoleController → Client: 200 OK (ApiResponse)