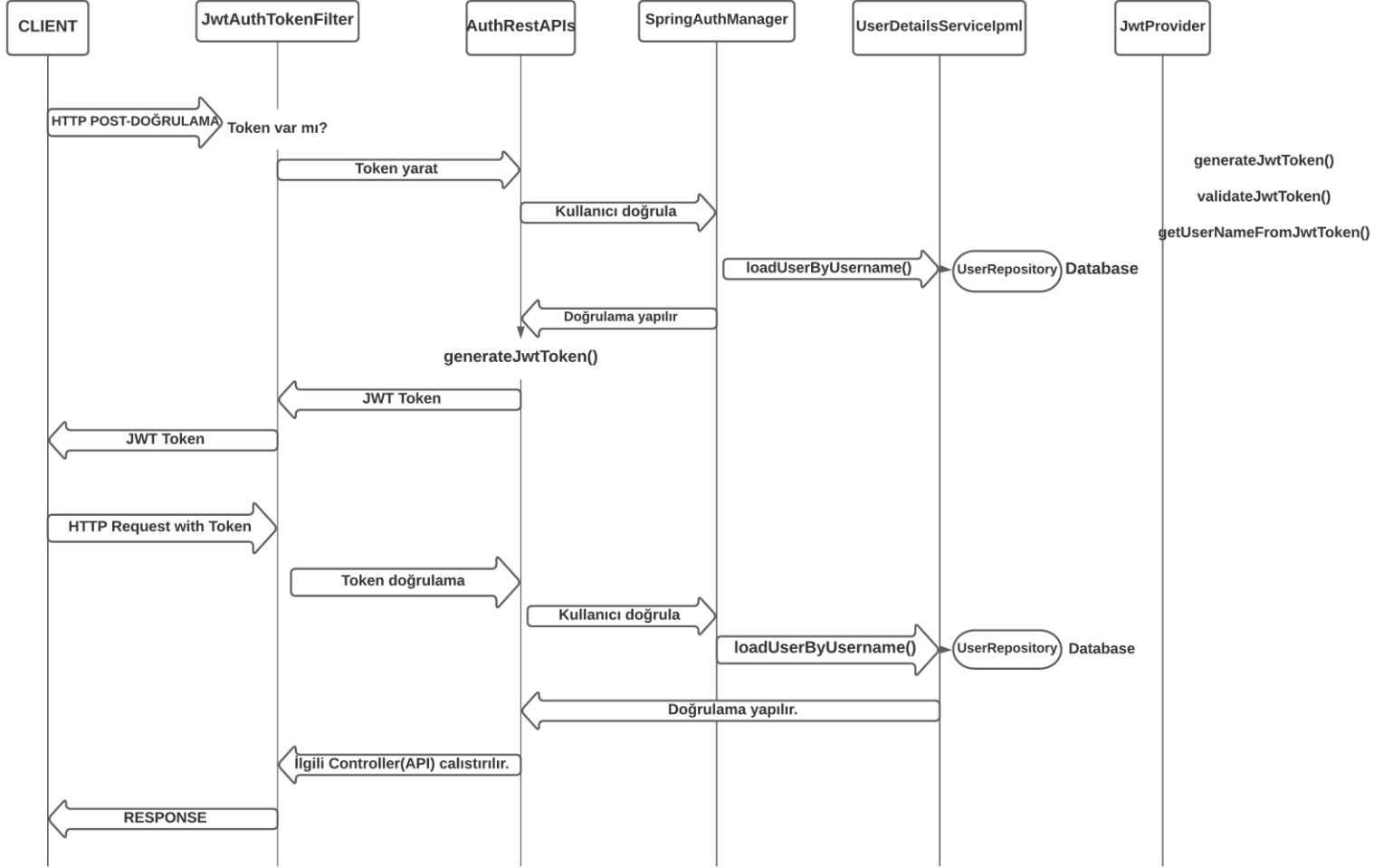


# MyUserServiceProject

## JWT Authotantication Şeması



## ***Request ve Response Formatları***

### **AuthRestAPIs**

URL Structure : <http://localhost:8080/api/auth>

**authenticateUser()** -> <http://localhost:8080/api/signin>

Parametreler: LoginForm( username ve password parametrelerini içinde tutan sınıf nesnesi. @RequestBody ile ilgili alanlar ilişilendirilir.)

Return: ResponseEntity

**registerUser()** -> <http://localhost:8080/api/signup>

Parametreler: SignUpForm ( name, username, email, password alanlarını içinde tutan sınıf nesnesi. @RequestBody ile ilgili alanlar ilişilendirilir.)

Return: ResponseEntity

### **TestRestAPIs**

URL Structure : <http://localhost:8080/api/test>

**userAccess()** -> <http://localhost:8080/api/test/user>

User veya admin rolüne sahip kullanıcıların ulaşabileceği alan.

Return: String

**projectManagementAccess()** -> <http://localhost:8080/api/test/pm>

pm veya admin rolüne sahip kullanıcıların ulaşabileceği alan.

Return: String

**adminAccess()** -> <http://localhost:8080/api/test/admin>

admin rolüne sahip kullanıcıların ulaşabileceği alan.

Return: String

**getAll()** -> <http://localhost:8080/api/test/admin/getAll>

Varolan User'ları List olarak döner.

Return: List<User>

**getAllPageable ()** -> <http://localhost:8080/api/test/admin/getAllPageable>

Parametreler: int pageNo, int pageSize

Varolan User'ları pageable List olarak döner.

Return: List<User>

**getAllSortable ()** -> <http://localhost:8080/api/test/admin/getAllSorted>

Varolan User'ları sıralayarak List döner.

Return: List<User>

**FindByEmail()** -> <http://localhost:8080/api/test/findByEmail>

Parametreler: String email

Gönderilen email'e sahip kullanıcıyı döner.

Return: User

**FindByUsername()** -> <http://localhost:8080/api/test/findByUsername>

Parametreler: String username

Gönderilen username'e sahip kullanıcıyı döner.

Return: User

**findByld () ->** <http://localhost:8080/api/test/admin/findByld>

Gönderilen id'ye sahip kullanıcıyı döner.

Return: User







### Example Responses

The screenshot shows a REST client interface. The top bar indicates the URL is `http://localhost:8080/` and the environment is `No Environment`. The request is a `POST` to `http://localhost:8080/api/auth/signup`. The body is set to `JSON (application/json)` and contains the following JSON:

```
1 {
2   "name": "admin",
3   "username": "admin",
4   "email": "admin@admin",
5   "role": ["admin", "user"],
6   "password": "123456"
7 }
8
```

The response is a `200 OK` with a time of `463 ms`. The response body is shown in `JSON` format:

```
1 {
2   "message": "User registered successfully"
3 }
```

Data Output		Explain	Messages	Notifications
 id [PK] integer 	email character varying (255) 	password character varying (255) 	username character varying (255) 	name character varying (255) 
1	1		[null]	[null]
2	2 string	string	[null]	[null]
3	3 string	string	[null]	[null]
4	4 string	string	[null]	[null]
5	5 string	string	[null]	[null]
6	6 string	string	[null]	[null]
7	7 sema@sema	\$2a\$10\$A2ABGSL7HxkFCH8p44yl1ewnIAoDIGa1hJa4x5nMkoVbLkgM8aJyC	smnr	sema
8	8 admin@admin	\$2a\$10\$ouZryjM9ZctfVIX.ZdO4Oc35YLMuopmH.jheQUUXYqg.BlaAv5p6	admin	admin

http://localhost:8080/ + ... No Environment

POST ▼ http://localhost:8080/api/auth/signin Params Send ▼

Authorization Headers (1) **Body** ● Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```
1 {
2   "name": "admin",
3   "username": "admin",
4   "email": "admin@admin",
5   "role": ["admin", "user"],
6   "password": "123456"
7 }
8 }
```

**Body** Cookies Headers (13) Test Results Status: 200 OK Time: 269 ms

Pretty Raw Preview JSON ▼ 🔍

```
1 {
2   "username": "admin",
3   "authorities": [
4     {
5       "authority": "ROLE_ADMIN"
6     },
7     {
8       "authority": "ROLE_USER"
9     }
10  ],
11   "accessToken": "eyJhbGciOiJIUzUxMiJ9.eyJ3ZWI101JhZG1pb1IsIm1hdCI6MTYzNjJkNDEyOCw1ZXhwIjoxNjMzMDAwNTI4fQ.sdBZyd1ILk1pUyRCCTT1AmAK9JvQ1pnGck1ib7W1rt_ekDUG3-KABgXrrt1toVAH_vkxuihlsU16fHocbAPTA",
12   "tokenType": "Bearer"
13 }
```