

JWT AUTHENTICATION

No parameters

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:5010/Product/GetAll' \
-H 'accept: */*'
```

Request URL

```
http://localhost:5010/Product/GetAll
```

Server response

Code

Details

401

Undocumented

Error: Unauthorized

Response headers

```
content-length: 0
date: Thu, 18 May 2023 10:26:51 GMT
server: Kestrel
www-authenticate: Bearer
```

No parameters

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:5010/Product/GetAll' \
-H 'accept: */*'
```

Request URL

```
http://localhost:5010/Product/GetAll
```

Server response

Code

Details

401

Undocumented

Error: Unauthorized

Response headers

```
content-length: 0
date: Thu, 18 May 2023 10:26:51 GMT
server: Kestrel
www-authenticate: Bearer
```

The screenshot displays a REST client interface with the following details:


- Request URL:** `http://localhost:5010/User/Authenticate`
- Method:** POST
- Headers:**
 - `Accept: */*`
 - `Content-Type: application/json`
- Body:**

```
{
  "username": "Induja",
  "password": "Induja"
}
```
- Status:** 200 OK
- Response Headers:**
 - `content-type: application/json; charset=utf-8`
 - `date: Thu, 18 May 2023 10:27:53 GMT`
 - `server: Kestrel`
 - `transfer-encoding: chunked`
- Response Body:**


```
{
  "juttoken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbmlxdWVfbmFtZSI6ImluZWhtYYSIsInJvbGUlOiJhZGIpbGlzInSiZiI6MTY4NDQwNTY3MiwiaWF0IjoxNjg0MDA2ODczLCJpYXQiOiE2ODQwMDU2NzN9.fsa3eeNRPqTIIscS243pWvdXKf52D1TwaBvTS3gm",
  "refreshToken": "hQUxpsQWdzijU4h7BaT0LC1433L1Pr2B1b7P8m/sK0="
}
```

CODE

FORMS AUTHENTICATION

 Travel Reimbursement

Login



Oops...

Invalid Detail!

OK

☐ Remember password
[Forgot password?](#)

Login

ROLE BASE AUTHENTICATION

```
[HttpPost]
0 references
public async Task<ActionResult> LoginManager(string Name, string EmailId, string userPassword, LoginViewModel modelLogin)
{
    var user = await _Context.Signup_Detail.FirstOrDefaultAsync(u => u.Name == Name && u.Email == EmailId && u.Password == userPassword);
    if (user != null)
    {
        //if(modelLogin.EmailID==EmailID && modelLogin.Password==Password){
        HttpContext.Session.SetString("Name", user.Name);
        List<Claim> claims = new List<Claim>() {
            new Claim(ClaimTypes.NameIdentifier, EmailId)
            // new Claim("OtherProperties", "Example Role")
        };
        ClaimsIdentity claimsIdentity = new ClaimsIdentity(claims,
            CookieAuthenticationDefaults.AuthenticationScheme);

        AuthenticationProperties properties = new AuthenticationProperties()
        {
            AllowRefresh = true,
            IsPersistent = modelLogin.KeepLoggedIn
        };
        await HttpContext.SignInAsync(CookieAuthenticationDefaults.AuthenticationScheme,
            new ClaimsPrincipal(claimsIdentity), properties);
        _logger.LogInformation("{EmailID} logged in successfully.", EmailId);
        return RedirectToAction("Index", "RaiseBill");
    }
    TempData["msg"] = "Incorrect EmailID/Password";
}
```