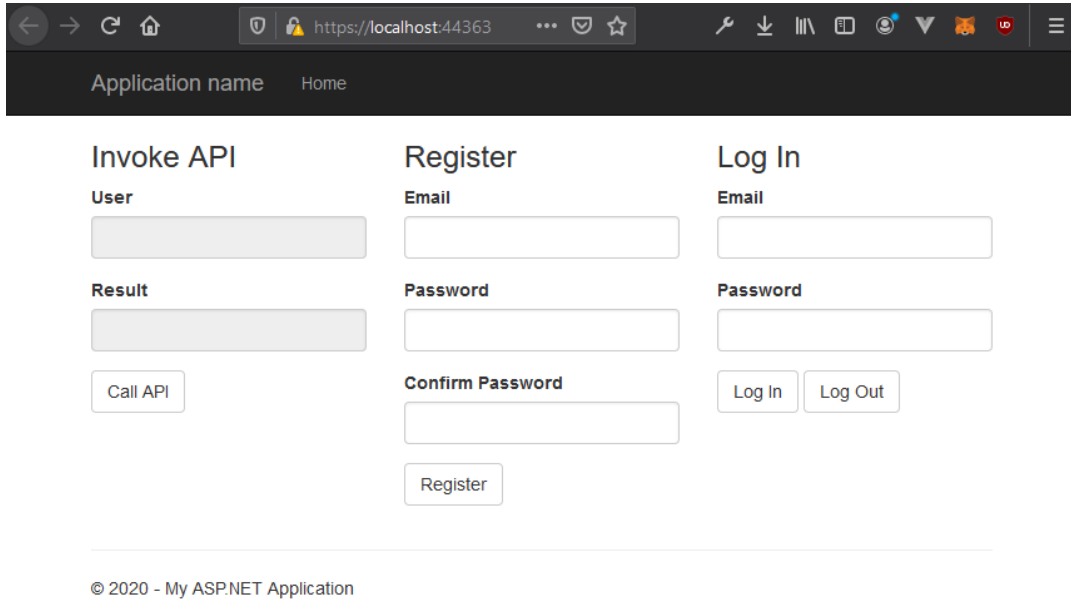


## Task 3

### Browser

Debug run the application and you will see the following page on load.



The screenshot shows a web browser window with the address bar displaying `https://localhost:44363`. The browser's address bar includes navigation buttons (back, forward, refresh, home) and a search bar. The page content is divided into three main sections: **Invoke API**, **Register**, and **Log In**.

**Invoke API** section:

- User**: A text input field.
- Result**: A text input field.
- Call API**: A button.

**Register** section:

- Email**: A text input field.
- Password**: A text input field.
- Confirm Password**: A text input field.
- Register**: A button.

**Log In** section:

- Email**: A text input field.
- Password**: A text input field.
- Log In**: A button.
- Log Out**: A button.

At the bottom of the page, there is a footer that reads: © 2020 - My ASP.NET Application.

## Register

Register account by entering email and password twice.

### Invoke API

User

Result

Done!

Call API

### Register

Email

Password

Confirm Password

Register

### Log In

Email

Password

Log In

Log Out

Register can fail if email is already existed in the database.

### Invoke API

User

Result

400: Bad Request

Call API

The request is invalid.

Name admin@mail.com is already taken.

Email 'admin@mail.com' is already taken.

### Register

Email

Password

Confirm Password

Register

### Log In

Email

Password

Log In

Log Out

Register can fail if password fails the validation process.

## Invoke API

User

Result

400: Bad Request

Call API

The request is invalid.

The Password must be at least 6 characters long.

## Register

Email

admin@mail.com

Password

••••

Confirm Password

••••

Register

## Log In

Email

Password

Log In

Log Out

## Invoke API

User

Result

400: Bad Request

Call API

The request is invalid.

Passwords must have at least one non letter or digit character.  
Passwords must have at least one lowercase ('a'-'z'). Passwords must have at least one uppercase ('A'-'Z').

## Register

Email

admin@mail.com

Password

••••••

Confirm Password

••••••

Register

## Log In

Email

Password

Log In

Log Out

## Log in

Log in to the account by entering email and password. (Bearer token will be created)

### Invoke API

User

admin@mail.com

Result

Call API

### Register

Email

Password

Confirm Password

Register

### Log In

Email

admin@mail.com

Password

.....

Log In

Log Out

Log in can fail if email does not exist or password is incorrect.

### Invoke API

User

Result

400: Bad Request

Call API

invalid\_grant

The user name or password is incorrect.

### Register

Email

Password

Confirm Password

Register

### Log In

Email

admin@mail.com

Password

.....

Log In

Log Out

## Call API

After logging in, user will be able to call API. (Authorized using bearer token)

### Invoke API

User

admin@mail.com

Result

Hello, admin@mail.com.

Call API

### Register

Email

Password

Confirm Password

Register

### Log In

Email

admin@mail.com

Password

••••••••

Log In

Log Out

API could not be called when user is not log on. (Due to lack of bearer token)

### Invoke API

User

Result

401: Unauthorized

Call API

Authorization has been denied for this request.

### Register

Email

Password

Confirm Password

Register

### Log In

Email

Password

Log In

Log Out

## Postman

Localhost:

<https://localhost:44363/>

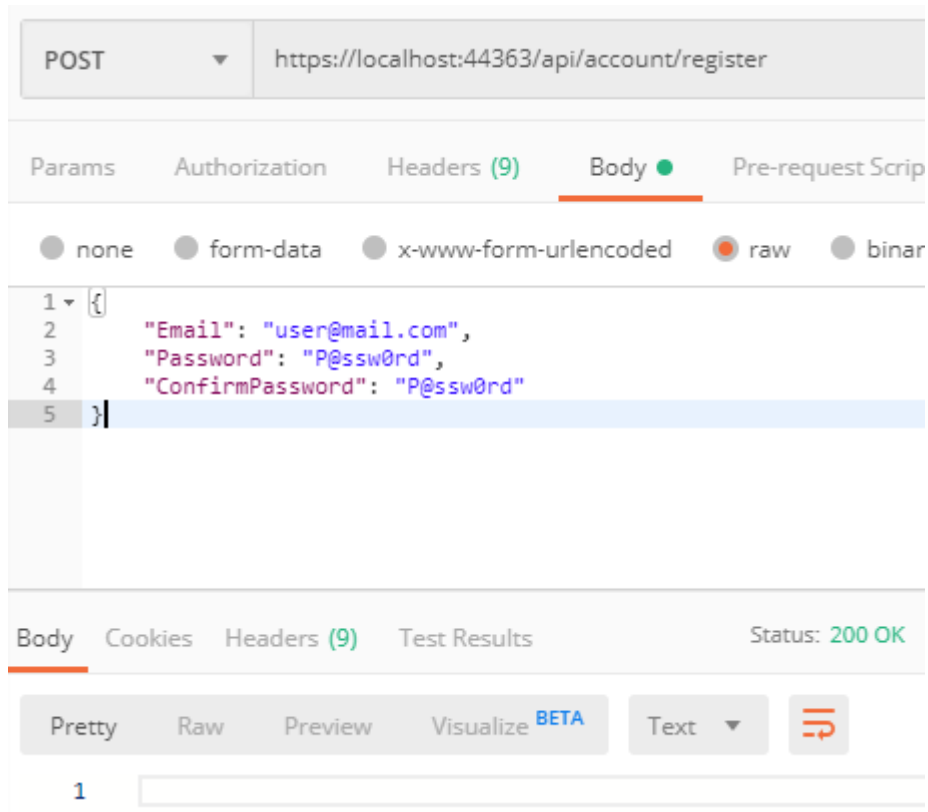
APIs:

Security Level	Type	API Route
Anonymous	POST	api/account/register
Anonymous	POST	token
Authorized	GET	api/values

POST: api/account/register

Link: <https://localhost:44363/api/account/register>

Register account by Post.



Exist email will lead to 400 Bad Request.

The screenshot shows a REST client interface with a POST request to `https://localhost:44363/api/account/register`. The request body is a JSON object with the following fields:

```
{
  "Email": "user@mail.com",
  "Password": "P@ssw0rd",
  "ConfirmPassword": "P@ssw0rd"
}
```

The response status is **400 Bad Request**. The response body is a JSON object indicating an invalid request due to an existing email:

```
{
  "Message": "The request is invalid.",
  "ModelState": {
    "": [
      "Name user@mail.com is already taken.",
      "Email 'user@mail.com' is already taken."
    ]
  }
}
```

The interface includes tabs for Params, Authorization, Headers (9), Body, and Pre-request Script. The Body tab is selected, and the response is displayed in the Pretty view. The status bar at the bottom shows the status as 400 Bad Request.

Fail model validation will lead to 400 Bad Request.

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

- Method:** POST
- URL:** https://localhost:44363/api/account/register
- Body Tab:** Selected, showing a JSON request body:

```
1 {  
2   "Email": "user@mail.com",  
3   "Password": "P@ssw0rd",  
4   "ConfirmPassword": "password"  
5 }
```
- Response Tab:** Selected, showing a JSON response body:

```
1 {  
2   "Message": "The request is invalid.",  
3   "ModelState": {  
4     "model.ConfirmPassword": [  
5       "The password and confirmation password do not match."  
6     ]  
7   }  
8 }
```
- Status:** 400 Bad Request
- Time:** 25ms



## POST: token

Link: <https://localhost:44363/token>

Get token for API which need authorization by posting email and password. (Email is set as a ClaimType in the token)

POST ▼ <https://localhost:44363/token>

Params Authorization Headers (10) **Body** ● Pre-request Script Tests

● none ● form-data ● x-www-form-urlencoded ● **raw** ● binary ● Graph

1 `username=user@mail.com&password=P@ssw0rd&grant_type=password`

Body Cookies (1) Headers (10) Test Results Status: 200 OK Time: 41ms Si

Pretty Raw Preview Visualize BETA JSON ↻

```
1 {
2   "access_token":
3     "O7DjgAWaNG-Wi69N014dJzMKVKQtIV0aoD9EtVB1-G1HV4B8qY3ynsMpRL2Pf4J
4     JQrBr_0OGZo7blpAB5HstF_1ykkKf8i7z5GC941XbYEvn0NC4uIYF1pJU49Xjp5y
5     PD4nabbgo1_SFf4T-tsDncYwkf2X0N1VxeFBeRqS_OUQYFtRwpfgI4VvSz5moMMa
6     awjP9l6qPpQaTEi1PqEj5Ac8veRoRXVvJdb9Ajq4Tq1I3sgxN34eV5xLswfUkDs9
7     hLvAIqyEhvgH70gKJxoFv5k9J-fVnVLRccKZuG8WFyH9g2OoLYQk1oe5AUaZ23Sj
8     EgwGIId-DchZ52de9gOaP89AraVyFYNUc2BepfqHgU",
9   "token_type": "bearer",
10  "expires_in": 1209599,
11  "userName": "user@mail.com",
12  ".issued": "Fri, 03 Jan 2020 16:54:56 GMT",
13  ".expires": "Fri, 17 Jan 2020 16:54:56 GMT"
```

Email that does not exist or incorrect password will lead to 400 Bad Request.

The screenshot displays a REST client interface for a POST request to `https://localhost:44363/token`. The request body is a URL-encoded string: `username=user@mail.com&password=P@ssw0rd&grant_type=password`. The response status is `400 Bad Request` with a time of `27ms`. The response body is a JSON object indicating an invalid grant.

POST `https://localhost:44363/token`

Params Authorization Headers (10) **Body** Pre-request Script Tests

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ Graph

1 `username=user@mail.com&password=P@ssw0rd&grant_type=password`

Body Cookies (1) Headers (9) Test Results Status: **400 Bad Request** Time: **27ms**

Pretty Raw Preview Visualize **BETA** JSON

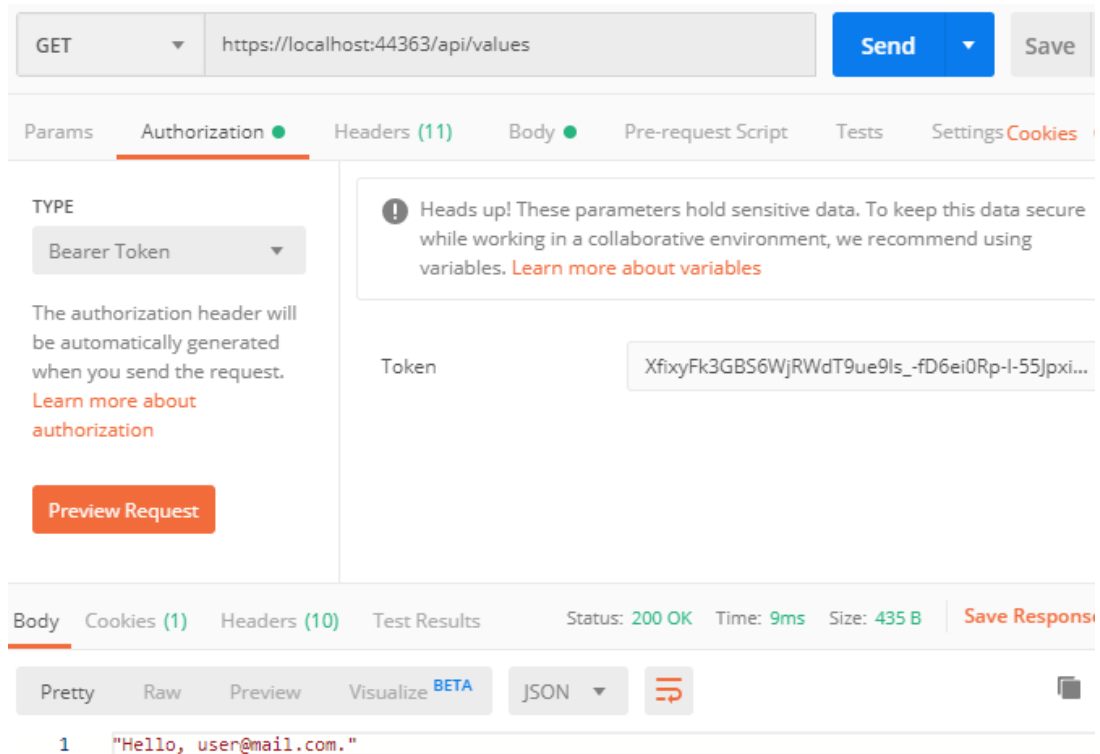
```
1 {
2   "error": "invalid_grant",
3   "error_description": "The user name or password is incorrect."
4 }
```

GET: api/values

Link: <https://localhost:44363/api/values>

(Requires Bearer token)

Get request with bearer token. (Email is derived from a ClaimType in the token)



The screenshot displays the Swagger client interface for a REST API. At the top, the method is set to GET and the URL is `https://localhost:44363/api/values. The 'Authorization' tab is selected, showing a 'Bearer Token' type. A warning message states: 'Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables'. The token value is XfixyFk3GBS6WjRWdT9ue9ls_-fD6ei0Rp-l-55Jpxi.... A 'Preview Request' button is visible. The 'Body' tab is selected at the bottom, showing the response status '200 OK', time '9ms', and size '435 B'. The response body is displayed in JSON format as "Hello, user@mail.com.".`

Get request without bearer token will lead to 403 Unauthorized .

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

- Request Bar:** Method is GET, URL is `https://localhost:44363/api/values`. Buttons for "Send" and "Save" are present.
- Request Configuration:** Tabs include Params, Authorization (selected), Headers (10), Body, Pre-request Script, Tests, Settings, and Cookies. The Authorization tab shows "TYPE" set to "Inherit auth from pa..." and a message: "The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)".
- Response Information:** A bar at the bottom shows "Status: 401 Unauthorized", "Time: 8ms", and "Size: 509 B". A "Save Response" button is on the right.
- Response Body:** The "Body" tab is selected, showing a JSON response in "Pretty" format: 

```
{  "Message": "Authorization has been denied for this request."}
```