Task 3 Web API Documentation

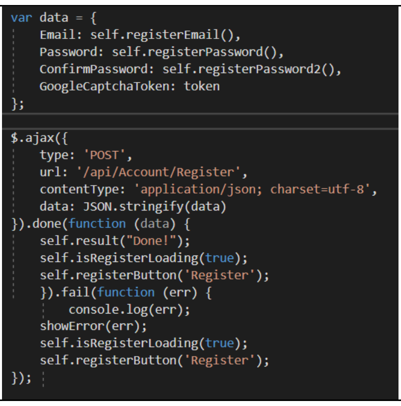
Function	**API**	**Description**
Register	POST /api/Account/Register	Creates new user in AspNetUsers
Login	POST /Token	Authentication and log in for users
Logout	POST /api/Account/Logout	Logs out users
Getting values/Invoke API	GET /api/values	Returns an array of string values

Register

Registration helps to save the user's credentials and allow them to login. On successful registration, the user's credentials (Email and Password) can then be used in order to Login the application.

URL	/api/Account/Register
Method	POST
sample body *(object)*	`{ "Email": "test@example.com", "Password": "P@ssw0rd", "ConfirmPassword": "P@ssw0rd", "GoogleCaptchaToken": {GOOGLE_TOKEN_RECEIVED} }`
contentType	application/json charset=UTF-8
Success Response	Code: 200 Content: -

Error Responses	1. **Any values that do not pass validation**
	Code: 400
	Sample Content:
	`{ "Message": "The request is invalid.", "ModelState": { "": [ "Email 'testingMyApiAgainexample.com' is invalid." ] } }`
	2. **Server Errors**
	Code: 500
	Sample Content:
	`{ "Message": "An error has occurred.", "ExceptionMessage": "An exception occurred while initializing the database. See the InnerException for details.", "ExceptionType": "System.Data.DataException", "StackTrace": ….. }`

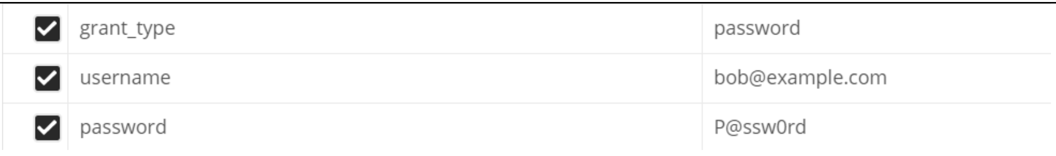
Sample Call	

```
var data = {
    Email: self.registerEmail(),
    Password: self.registerPassword(),
    ConfirmPassword: self.registerPassword2(),
    GoogleCaptchaToken: token
};
```

```
$.ajax({
    type: 'POST',
    url: '/api/Account/Register',
    contentType: 'application/json; charset=utf-8',
    data: JSON.stringify(data)
}).done(function (data) {
    self.result("Done!");
    self.isRegisterLoading(true);
    self.registerButton('Register');
    }).fail(function (err) {
        console.log(err);
    showError(err);
    self.isRegisterLoading(true);
    self.registerButton('Register');
});
```

Login

Login helps user to be authorized to carry out specific tasks that they can only do when logged in.

URL	/Token
Method	POST
sample body *(x-www-form-urlencoded)*	
contentType	application/x-www-form-urlencoded charset=UTF-8

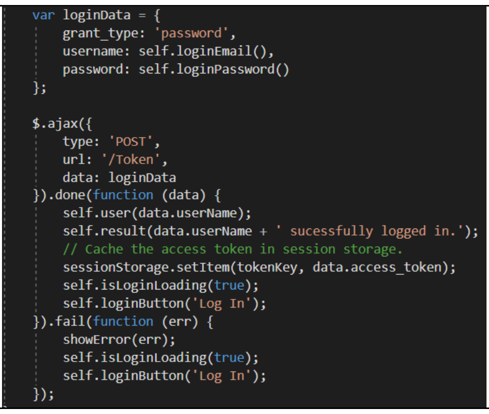
✓	grant_type	password
✓	username	bob@example.com
✓	password	P@ssw0rd

Success Response

Code: 200
Content:
```
{
   "access_token": {ACCESS_TOKEN_RECEIVED},
   "expires_in": 1209599,
   "userName": "kim@example.com",
   ".issued": "Fri, 08 Jan 2021 08:35:27 GMT",
   ".expires": "Fri, 22 Jan 2021 08:35:27 GMT"
}
```

Error Responses

1. **Any values that do not pass validation**
   Code: 400
   Sample Content:
   ```
   {
      "error": "invalid_grant",
      "error_description": "The user name or password is incorrect."
   }
   ```
2. **Server Errors**
   Code: 500
   Sample Content:
   ```
   {
      "Message": "An error has occurred.",
      "ExceptionMessage": "An exception occurred while initializing the database. See the InnerException for details.",
      "ExceptionType": "System.Data.DataException",
      "StackTrace": .....
   }
   ```

Sample Call	

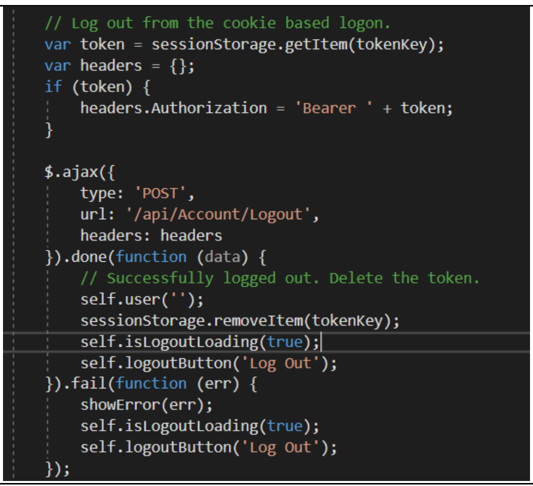
```
var loginData = {
    grant_type: 'password',
    username: self.loginEmail(),
    password: self.loginPassword()
};

$.ajax({
    type: 'POST',
    url: '/Token',
    data: loginData
}).done(function (data) {
    self.user(data.userName);
    self.result(data.userName + ' sucessfully logged in.');
    // Cache the access token in session storage.
    sessionStorage.setItem(tokenKey, data.access_token);
    self.isLoginLoading(true);
    self.loginButton('Log In');
}).fail(function (err) {
    showError(err);
    self.isLoginLoading(true);
    self.loginButton('Log In');
});
```

Logout

Log out helps user to remove their credentials from being used by other unauthorized personnel to carry out tasks.

URL	/api/Account/Logout
Method	POST
header	headers.authorization = ‘Bearer ‘ + {ACCESS_TOKEN_FROM_SESSIONSTORAGE}
Success Response	Code: 200 Content: -

Error Responses	1. **If token does not exist (No user logged in), token is not authorized**
	Code: 401
	Sample Content:
	{
	"Message": "Authorization has been denied for this request."
	}
	2. **Server Errors**
	Code: 500
	Sample Content:
	{
	"Message": "An error has occurred.",
	"ExceptionMessage": "An exception occurred while initializing the database. See the InnerException for details.",
	"ExceptionType": "System.Data.DataException",
	"StackTrace": …..
	}

Sample Call	

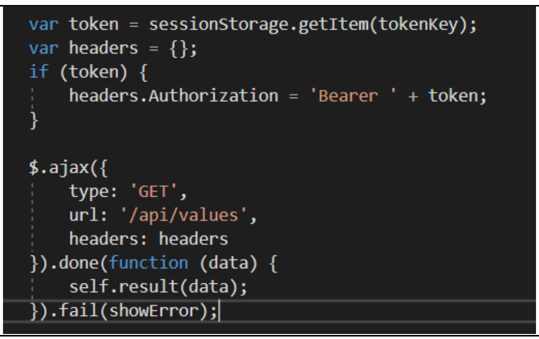
```
// Log out from the cookie based logon.
var token = sessionStorage.getItem(tokenKey);
var headers = {};
if (token) {
    headers.Authorization = 'Bearer ' + token;
}

$.ajax({
    type: 'POST',
    url: '/api/Account/Logout',
    headers: headers
}).done(function (data) {
    // Successfully logged out. Delete the token.
    self.user('');
    sessionStorage.removeItem(tokenKey);
    self.isLogoutLoading(true);
    self.logoutButton('Log Out');
}).fail(function (err) {
    showError(err);
    self.isLogoutLoading(true);
    self.logoutButton('Log Out');
});
```

Invoke API

This API can only be accessed by authorized users who are logged in.

URL	/api/values
Method	GET
header	headers.authorization = 'Bearer ' + {ACCESS_TOKEN_FROM_SESSIONSTORAGE}
Success Response	Code: 200 Content: ["value1", "value2"]

Error Responses	1. **If token does not exist (No user logged in), token is not authorized**
	Code: 401
	Sample Content:
	{
	"Message": "Authorization has been denied for this request."
	}
	2. **Server Errors**
	Code: 500
	Sample Content:
	{
	"Message": "An error has occurred.",
	"ExceptionMessage": "An exception occurred while initializing the database. See the InnerException for details.",
	"ExceptionType": "System.Data.DataException",
	"StackTrace": .....
	}

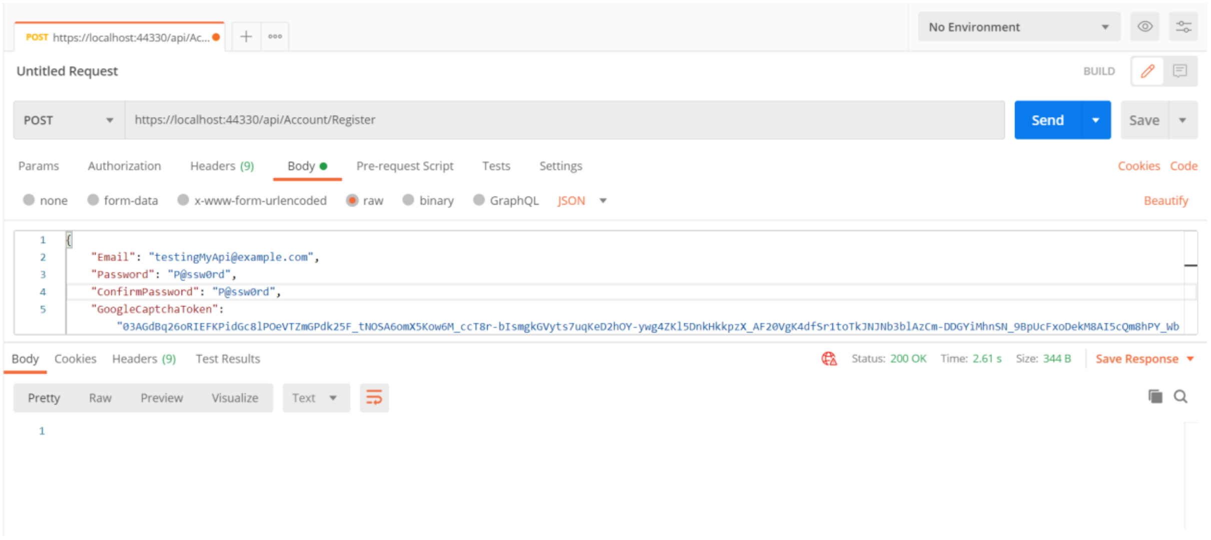
Sample Call		

```
var token = sessionStorage.getItem(tokenKey);
var headers = {};
if (token) {
    headers.Authorization = 'Bearer ' + token;
}

$.ajax({
    type: 'GET',
    url: '/api/values',
    headers: headers
}).done(function (data) {
    self.result(data);
}).fail(showError);
```

Task 3 Postman Testing Results

Register (https://localhost:44330/api/Account/Register)

1. Successful Registration with all the data (Email, Password, Confirm Password, GoogleCaptchaToken) passing validation.
   Response: 200 OK



Untitled Request

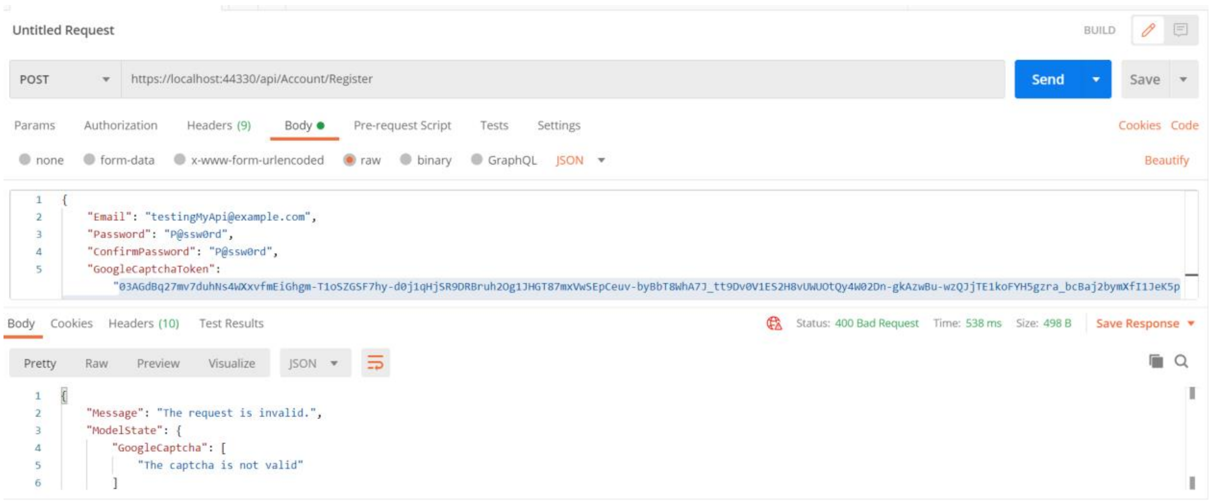
POST https://localhost:44330/api/Account/Register

Body: raw JSON

```
{
    "Email": "testingMyApi@example.com",
    "Password": "P@ssw0rd",
    "ConfirmPassword": "P@ssw0rd",
    "GoogleCaptchaToken":
        "03AGdBq26oRIEFKPidGc8lPOeVTZmGPdk25F_tNOSA6omX5Kow6M_ccT8r-bIsmgkGVyts7uqKeD2hOY-ywg4ZKl5DnkHkkpzX_AF20VgK4dfSr1toTkJNJNb3blAzCm-DDGYiMhnSN_9BpUcFxoDekM8AI5cQm8hPY_Wb
```

Status: 200 OK Time: 2.61 s Size: 344 B

2. Registration failed, due to Invalid Google Captcha Token.
   Response: 400 Bad Request



Untitled Request

POST https://localhost:44330/api/Account/Register

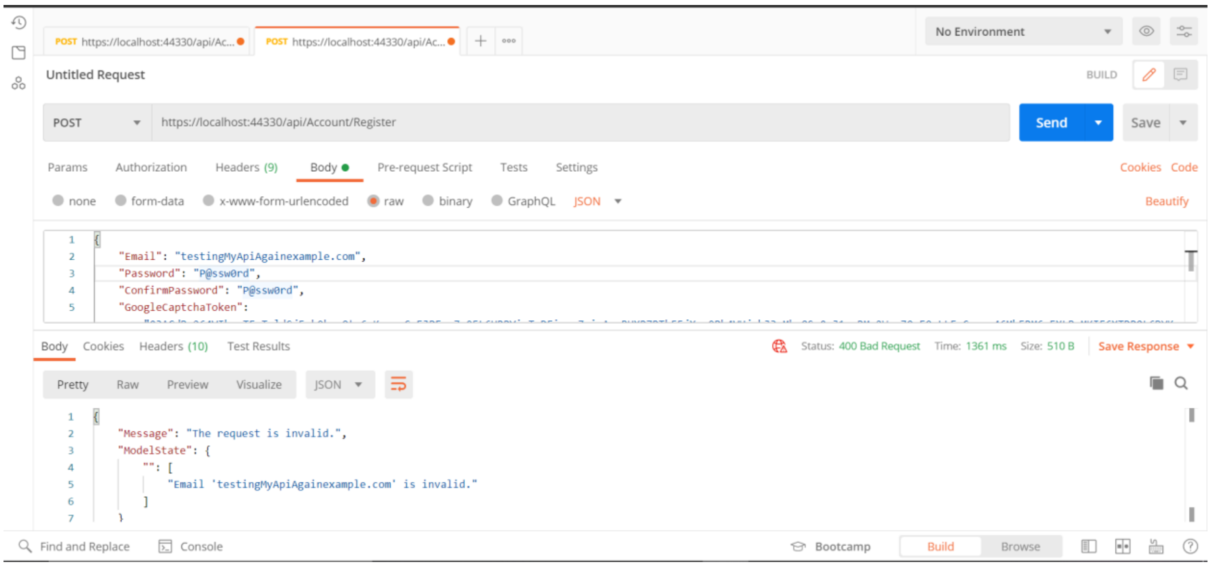
Body: raw JSON

```
{
    "Email": "testingMyApi@example.com",
    "Password": "P@ssw0rd",
    "ConfirmPassword": "P@ssw0rd",
    "GoogleCaptchaToken":
        "03AGdBq27mv7duhNs4WXxvfmEiGhgm-T1oSZGSF7hy-d0j1qHjSR9DRBruh2Og1JHGT87mxVwSEpCeuv-byBbT8WhA7J_tt9Dv0V1ES2H8vUWUOtQy4W02Dn-gkAzwBu-wzQJjTE1koFYH5gzra_bcBaj2bymXfI1JeK5p
```

Status: 400 Bad Request Time: 538 ms Size: 498 B

```
{
    "Message": "The request is invalid.",
    "ModelState": {
        "GoogleCaptcha": [
            "The captcha is not valid"
        ]
```

3. Registration failed, due to Invalid Email address.
   Response: 400 Bad Request



POST https://localhost:44330/api/Account/Register

Untitled Request

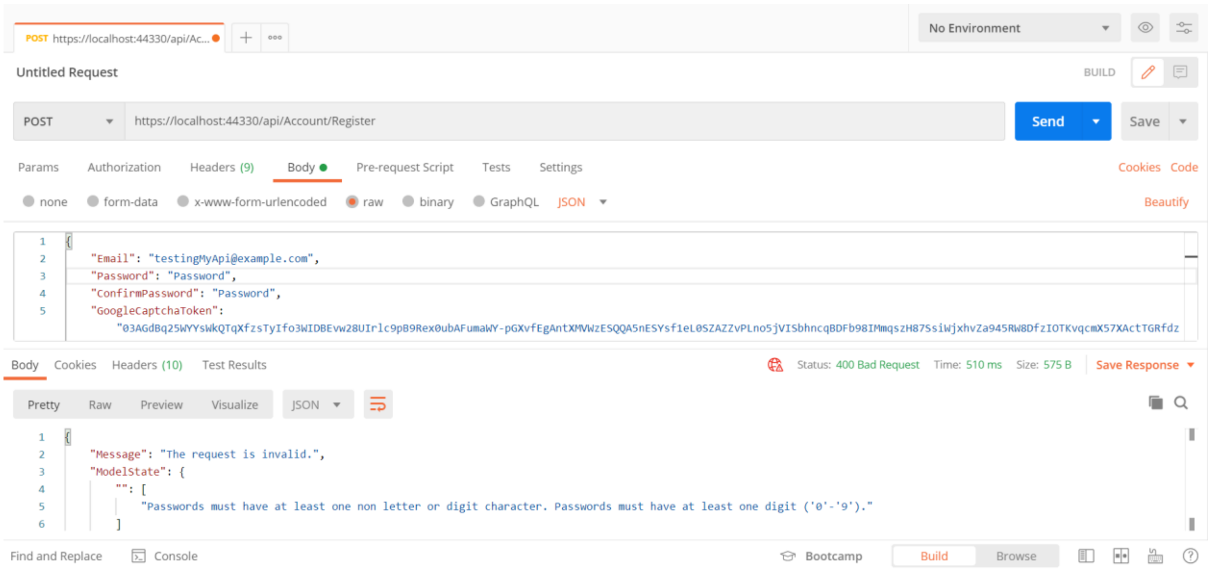
Body (raw, JSON):

```
{
    "Email": "testingMyApiAgainexample.com",
    "Password": "P@ssw0rd",
    "ConfirmPassword": "P@ssw0rd",
    "GoogleCaptchaToken":
```

Status: 400 Bad Request   Time: 1361 ms   Size: 510 B

```
{
    "Message": "The request is invalid.",
    "ModelState": {
        "": [
            "Email 'testingMyApiAgainexample.com' is invalid."
        ]
    }
```

4. Registration failed, due to Invalid Password.
   Response: 400 Bad Request



POST https://localhost:44330/api/Account/Register

Untitled Request

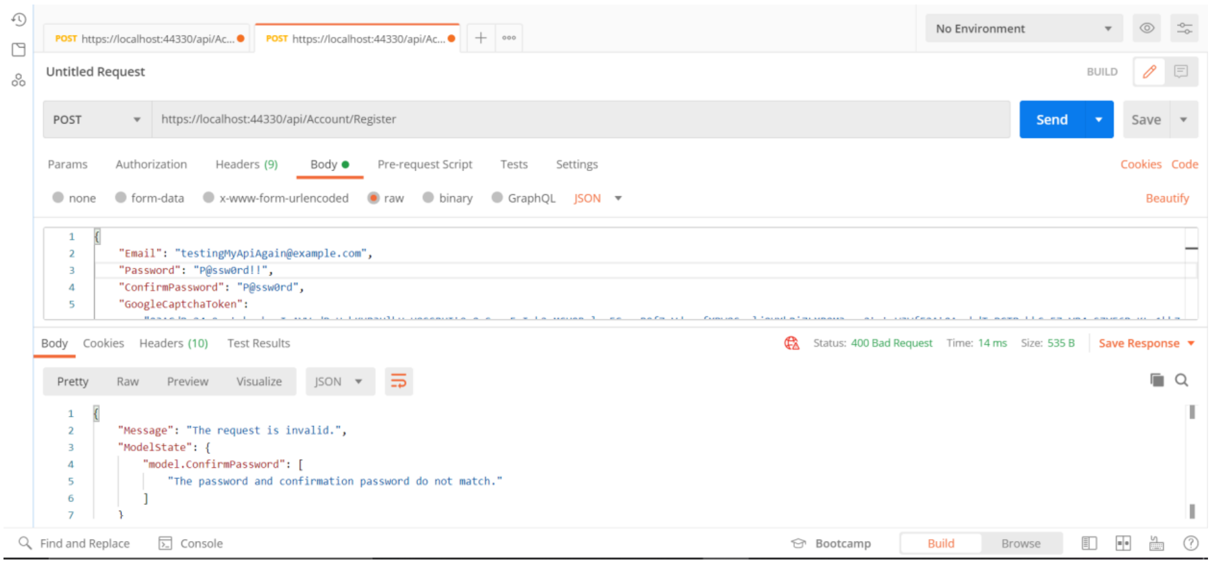
Body (raw, JSON):

```
{
    "Email": "testingMyApi@example.com",
    "Password": "Password",
    "ConfirmPassword": "Password",
    "GoogleCaptchaToken":
        "03AGdBq25WYYsWkQTqXfzsTyIfo3WIDBEvw28UIrlc9pB9Rex0ubAFumaWY-pGXvfEgAntXMVWzESQQA5nESYsf1eL0SZAZZvPLno5jVISbhncqBDFb98IMmqszH87SsiWjxhvZa945RW8DfzIOTKvqcmX57XActTGRfdz
```

Status: 400 Bad Request   Time: 510 ms   Size: 575 B

```
{
    "Message": "The request is invalid.",
    "ModelState": {
        "": [
            "Passwords must have at least one non letter or digit character. Passwords must have at least one digit ('0'-'9')."
        ]
```

5. Registration failed, due to password and confirm password not matching.
   Response: 400 Bad Request



POST https://localhost:44330/api/Account/Register

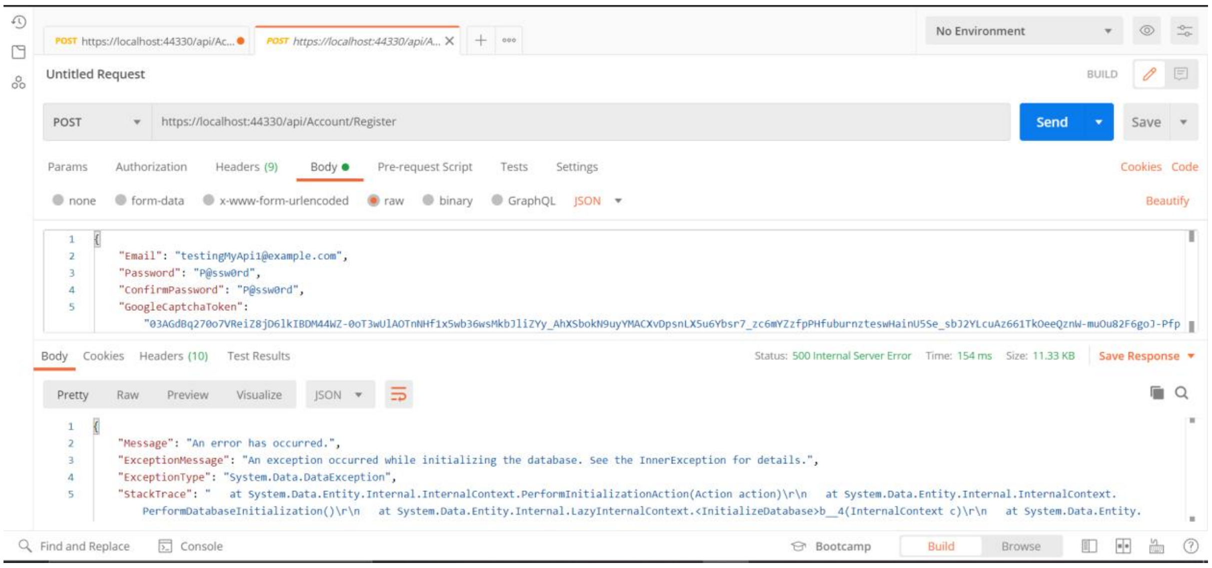
Body (raw, JSON):

```
{
    "Email": "testingMyApiAgain@example.com",
    "Password": "P@ssw0rd!!",
    "ConfirmPassword": "P@ssw0rd",
    "GoogleCaptchaToken":
```

Status: 400 Bad Request Time: 14 ms Size: 535 B

```
{
    "Message": "The request is invalid.",
    "ModelState": {
        "model.ConfirmPassword": [
            "The password and confirmation password do not match."
        ]
    }
```

6. Registration failed, due to server errors. (E.g. Database .mdf file unable to be found by Visual Studio, database connection issues)
   Response: 500 Internal Server Error



POST https://localhost:44330/api/Account/Register

Body (raw, JSON):

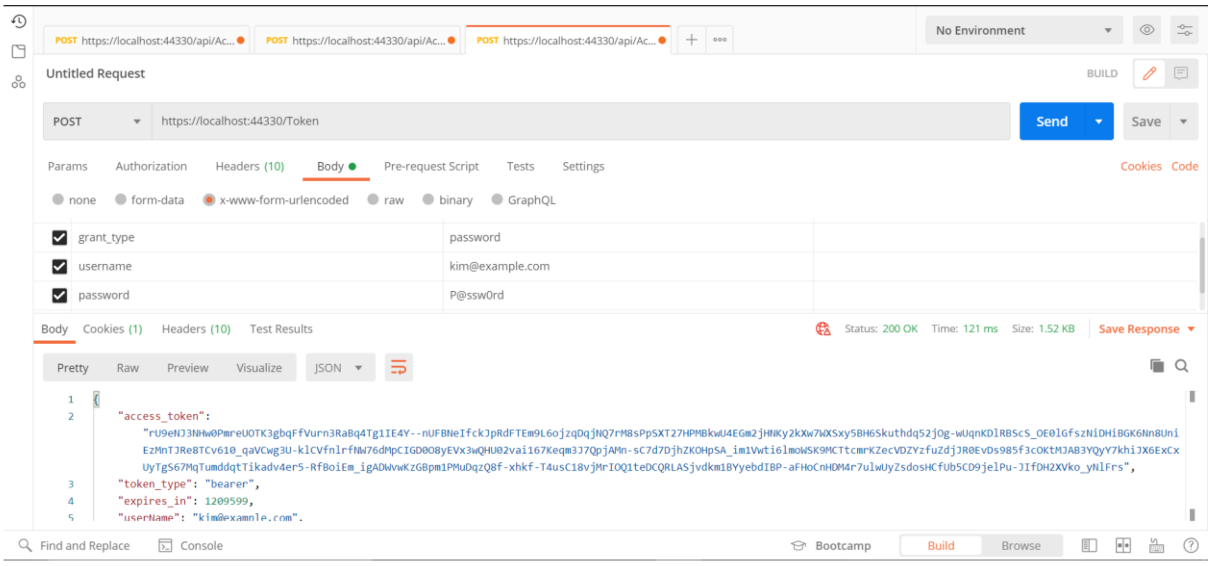
```
{
    "Email": "testingMyApi1@example.com",
    "Password": "P@ssw0rd",
    "ConfirmPassword": "P@ssw0rd",
    "GoogleCaptchaToken":
        "03AGdBq27o7VReiZ8jD6lkIBDM44WZ-0oT3wUlAOTnNHf1x5wb36wsMkbJliZYy_AhXSbokN9uyYMACXvDpsnLX5u6Ybsr7_zc6mYZzfpPHfuburnzteswHainU5Se_sbJ2YLcuAz661TkOeeQznW-muOu82F6goJ-Pfp
```

Status: 500 Internal Server Error Time: 154 ms Size: 11.33 KB

```
{
    "Message": "An error has occurred.",
    "ExceptionMessage": "An exception occurred while initializing the database. See the InnerException for details.",
    "ExceptionType": "System.Data.DataException",
    "StackTrace": "   at System.Data.Entity.Internal.InternalContext.PerformInitializationAction(Action action)\r\n   at System.Data.Entity.Internal.InternalContext.
        PerformDatabaseInitialization()\r\n   at System.Data.Entity.Internal.LazyInternalContext.<InitializeDatabase>b__4(InternalContext c)\r\n   at System.Data.Entity.
```

Login (https://localhost:44330/Token)

1. Successful Login with all the data (username, password, grant_type) passing validation.
   Response:200 OK



Untitled Request

POST https://localhost:44330/Token

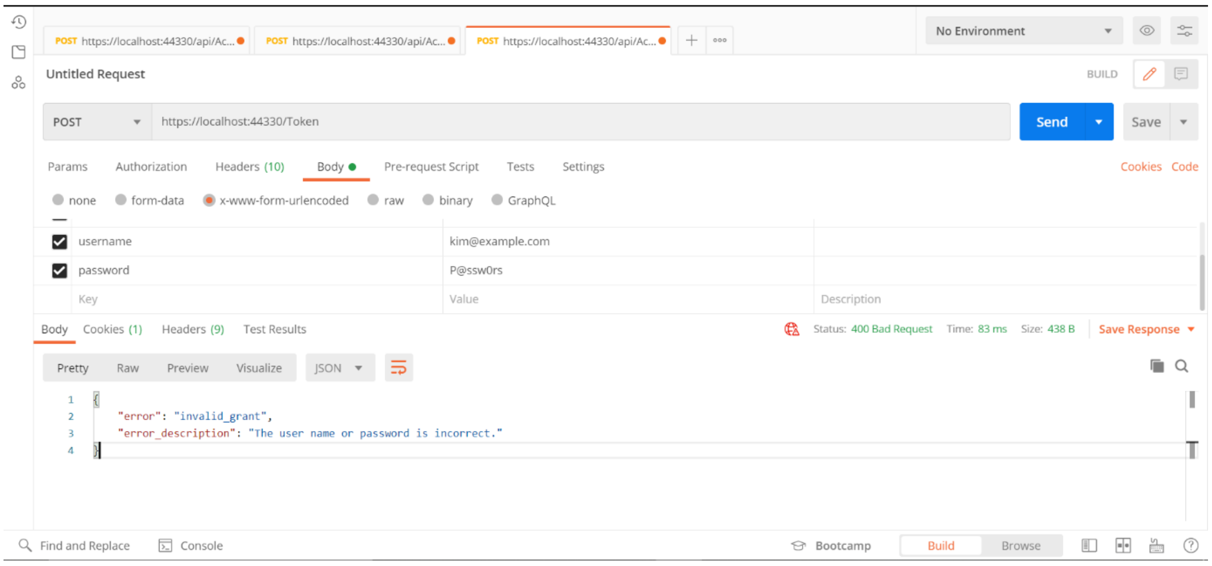
Body: x-www-form-urlencoded

grant_type	password
username	kim@example.com
password	P@ssw0rd

Status: 200 OK Time: 121 ms Size: 1.52 KB

```
{
    "access_token":
        "rU9eNJ3NHw0PmreUOTK3gbqFfVurn3RaBq4Tg1IE4Y--nUFBNeIfckJpRdFTEm9L6ojzqDqjNQ7rM8sPpSXT27HPMBkwU4EGm2jHNKy2kXw7WXSxy5BH6Skuthdq52jOg-wUqnKDlRBScS_OE0lGfszNiDHiBGK6Nn8Uni
        EzMnTJRe8TCv610_qaVCwg3U-klCVfnlrfNW76dMpCIGD0O8yEVx3wQHU02vai167Keqm3J7QpjAMn-sC7d7DjhZKOHpSA_im1Vwti6lmoWSK9MCTtcmrKZecVDZYzfuZdjJR0EvDs985f3cOKtMJAB3YQyY7khiJX6ExCx
        UyTgS67MqTumddqtTikadv4er5-RfBoiEm_igADWvwKzGBpm1PMuDqzQ8f-xhkf-T4usC18vjMrIOQ1teDCQRLASjvdkm1BYyebdIBP-aFHoCnHDM4r7ulwUyZsdosHCfUb5CD9jelPu-JIfDH2XVko_yNlFrs",
    "token_type": "bearer",
    "expires_in": 1209599,
    "userName": "kim@example.com",
```

2. Failed Login, due to either password or username being invalid or user not existing in the database
   Response: 400 Bad Request



Untitled Request

POST https://localhost:44330/Token

Body: x-www-form-urlencoded

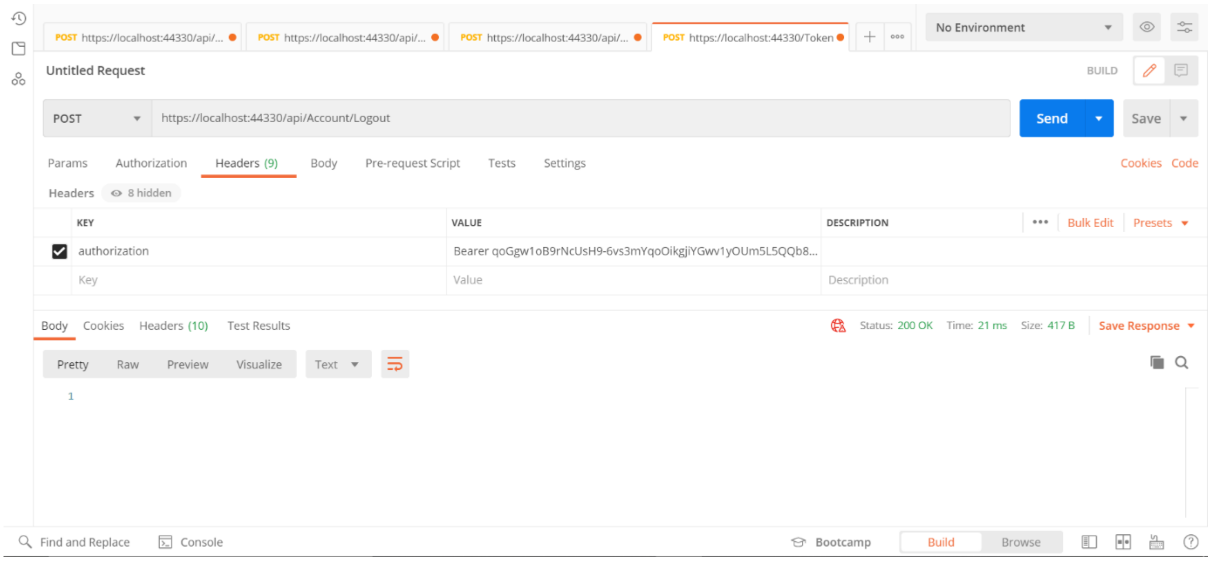
username	kim@example.com
password	P@ssw0rs

Status: 400 Bad Request Time: 83 ms Size: 438 B

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

Logout (https://localhost:44330/api/Account/Logout)

1. Successful Logout with all the data (Bearer token) passing authorization.
   Response: 200 OK



Untitled Request

POST https://localhost:44330/api/Account/Logout

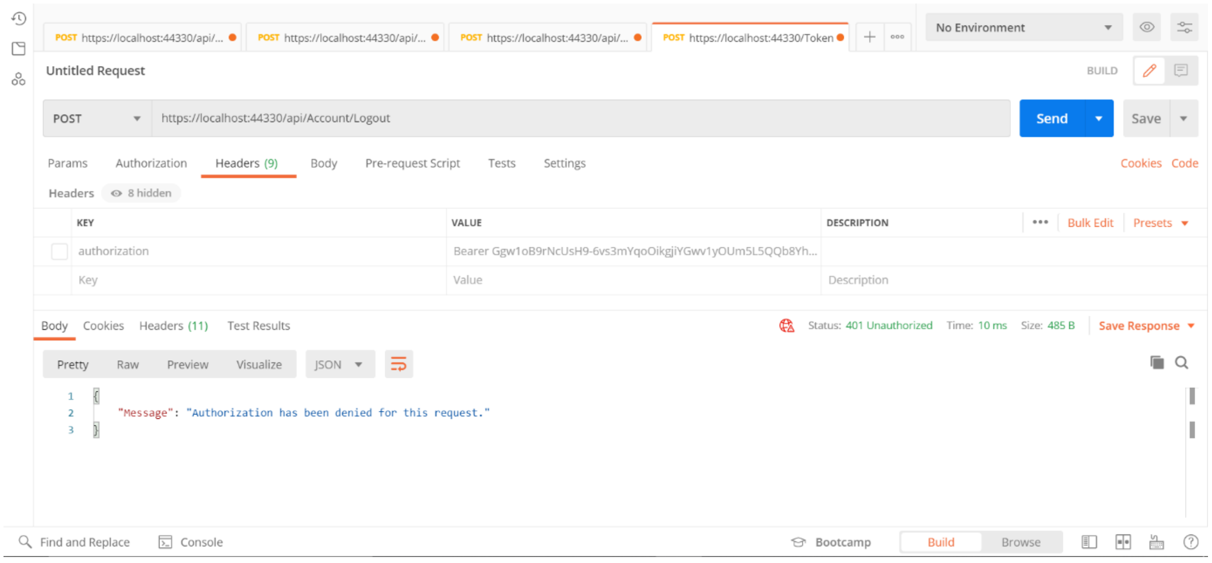
Headers (9) — 8 hidden

	KEY	VALUE	DESCRIPTION
✓	authorization	Bearer qoGgw1oB9rNcUsH9-6vs3mYqoOikgjiYGwv1yOUm5L5QQb8...	
	Key	Value	Description

Status: 200 OK Time: 21 ms Size: 417 B

```
1
```

2. Failed Logout with bearer token value passed not being authorized.
   Response: 401 Unauthorized



Untitled Request

POST https://localhost:44330/api/Account/Logout

Headers (9) — 8 hidden

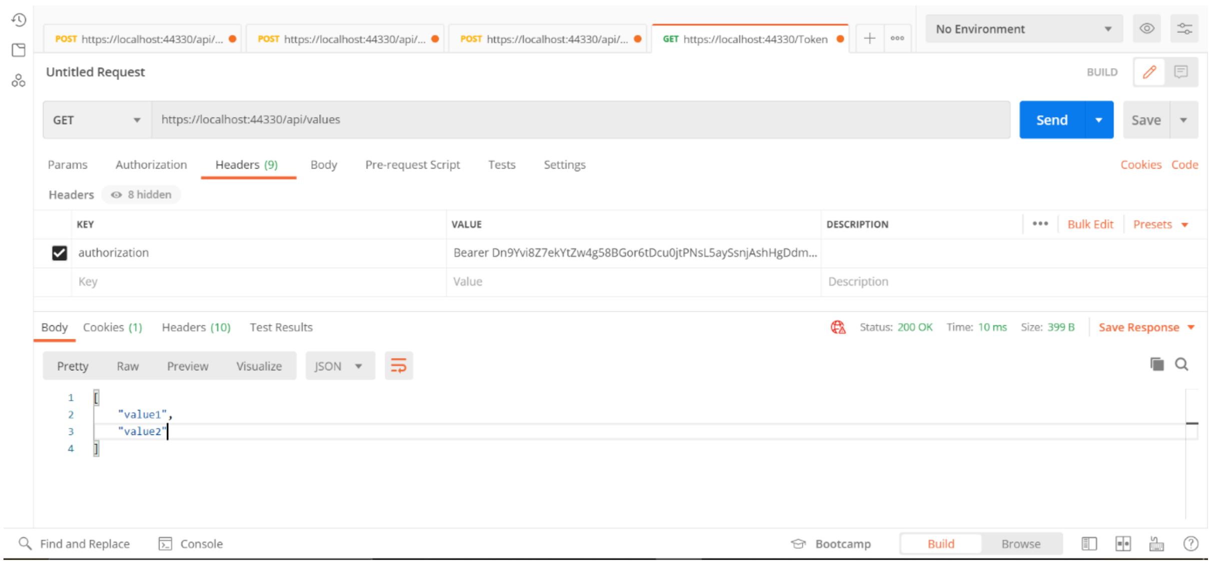
	KEY	VALUE	DESCRIPTION
	authorization	Bearer Ggw1oB9rNcUsH9-6vs3mYqoOikgjiYGwv1yOUm5L5QQb8Yh...	
	Key	Value	Description

Status: 401 Unauthorized Time: 10 ms Size: 485 B

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

Invoke API (https://localhost:44330/api/values)

1. Successful retrieval of all the data, with user (Bearer token) passing authorization.
   Response: 200 OK



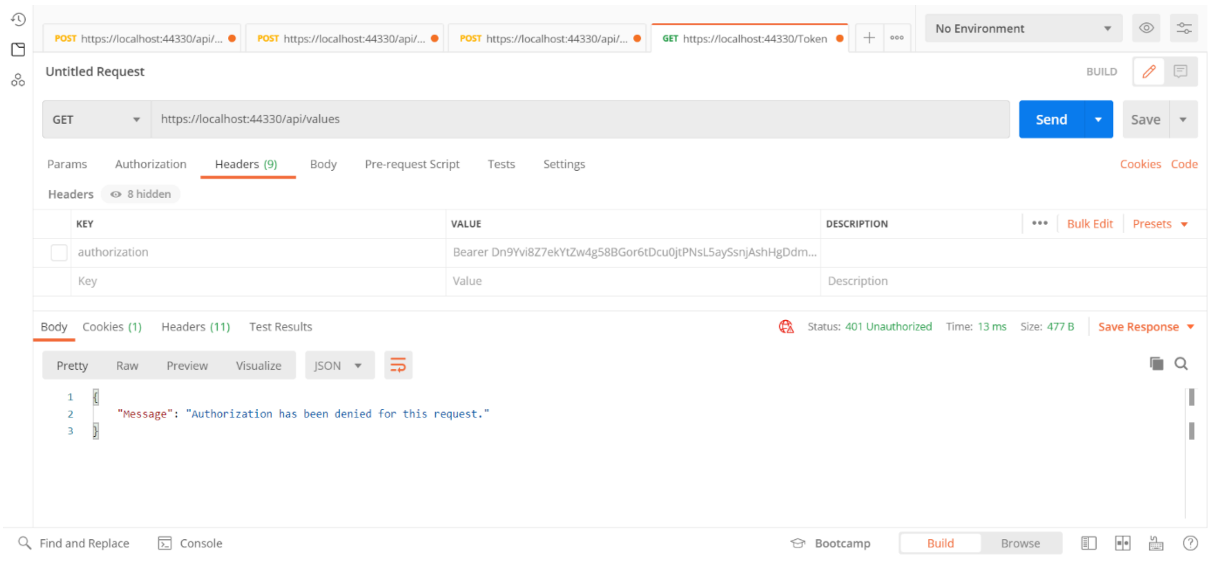
GET https://localhost:44330/api/values

KEY	VALUE	DESCRIPTION
✓ authorization	Bearer Dn9Yvi8Z7ekYtZw4g58BGor6tDcu0jtPNsL5aySsnjAshHgDdm...	
Key	Value	Description

Status: 200 OK Time: 10 ms Size: 399 B

```
[
    "value1",
    "value2"
]
```

2. Failed Retrieval of the data, with user (Bearer token) failing authorization or no user is logged in.
   Response: 401 Unauthorized



GET https://localhost:44330/api/values

KEY	VALUE	DESCRIPTION
authorization	Bearer Dn9Yvi8Z7ekYtZw4g58BGor6tDcu0jtPNsL5aySsnjAshHgDdm...	
Key	Value	Description

Status: 401 Unauthorized Time: 13 ms Size: 477 B

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