# Mastering ASP.NET Web API

Source code bundle for all chapters of this book.

Postman collection is included in bundle, import it for using with Postman REST client

Softwares Required

- .NET Core 2.0 Preview 2 SDK as per OS.
- Visual Studio 2017 Preview 3 IDE
- Visual Studio Code (for windows or non Windows OS)

## Puhoi folder source code for Chapter 4 - Controller, Actions & Models AND Chapter 7 - Chapter 7: Perform Unit & Integration Testing

## Chapter 3 - Anatomy of ASP.NET Core Web API

Project Name - MyFirstCoreAPI

Open the source code bundle in Visual Studio IDE

**How to run?**

- Choose *MyFirstCoreAPI* project, *Set as Startup Project* to run this project
- Press F5 to run the application. *Port number might differ*
- Use either Postman or any browser, access the endpoint http://localhost:49396/api/values to view results as shown in below image
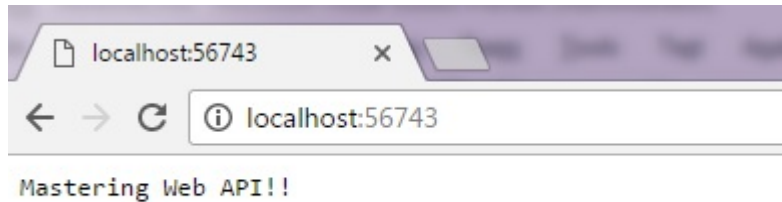
Basic Web API on Postman
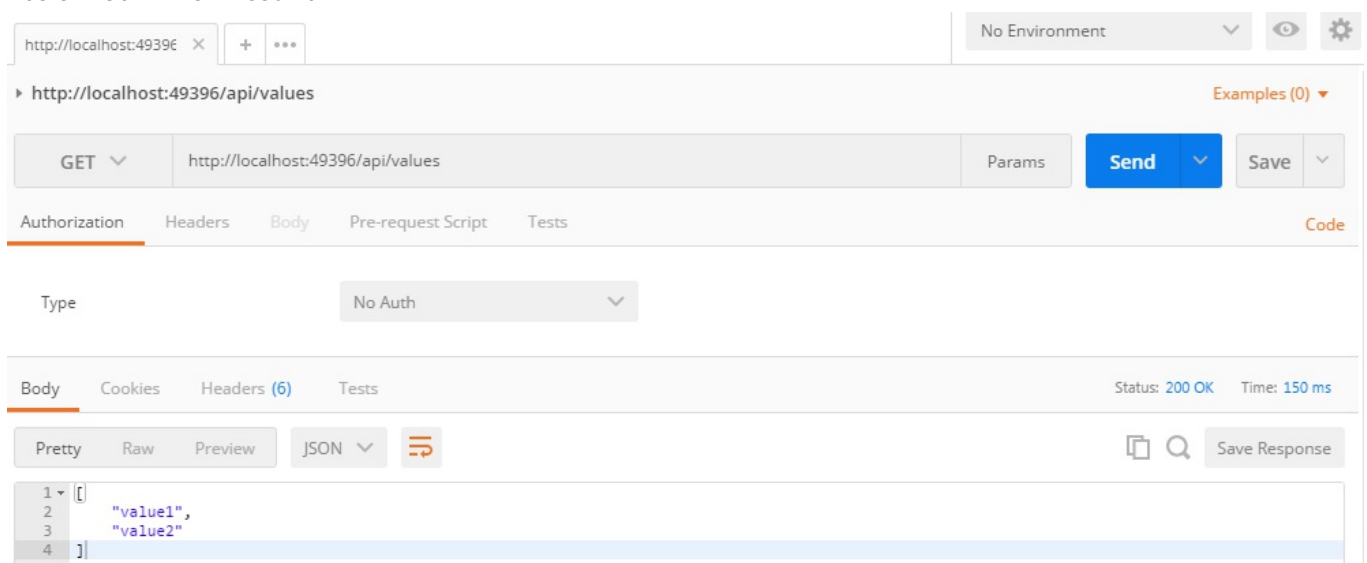


## Chapter 5 - Implementing Routing

Project Name - BasicRoutes

Open the source code bundle in Visual Studio IDE

**How to run?**

- Choose *BasicRoutes* project, *Set as Startup Project* to run this project
- Press F5 to run the application. *Port number might differ*
- Use any browser, access the endpoint http://localhost:56743/ to view results as shown in below image



Basic Routes

**Uncomment the below regions to learn respective concepts** Basic Routes & Constraints

```
public void Configure(IApplicationBuilder app, IHostingEnvironment env)
{
    BasicRoute

    RouteBuilder

    RouteConstraintsNotApplied

    RouteConstraintApplied

    CustomRouteConstraints
}
```
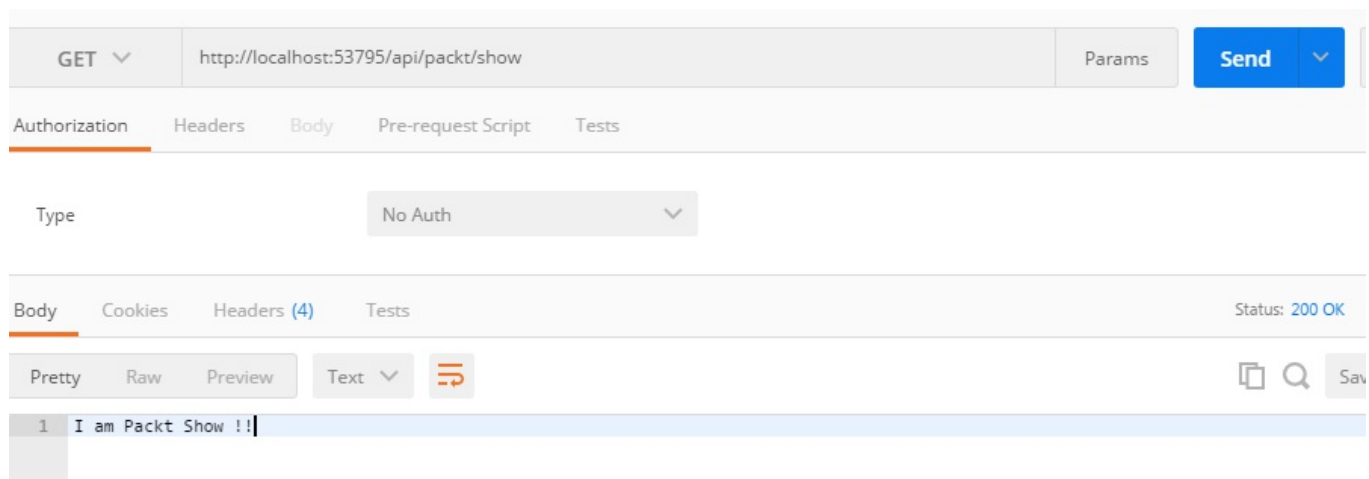
# Project Name - MyFirstCoreApiRoutes

Open the source code bundle in Visual Studio IDE

**How to run?**

- Choose *MyFirstCoreApiRoutes* project, *Set as Startup Project* to run this project
- Press F5 to run the application. *Port number might differ*
- Use Postman or any browser, access the endpoint http://localhost:56743/api/packt/show to view results as shown in below image
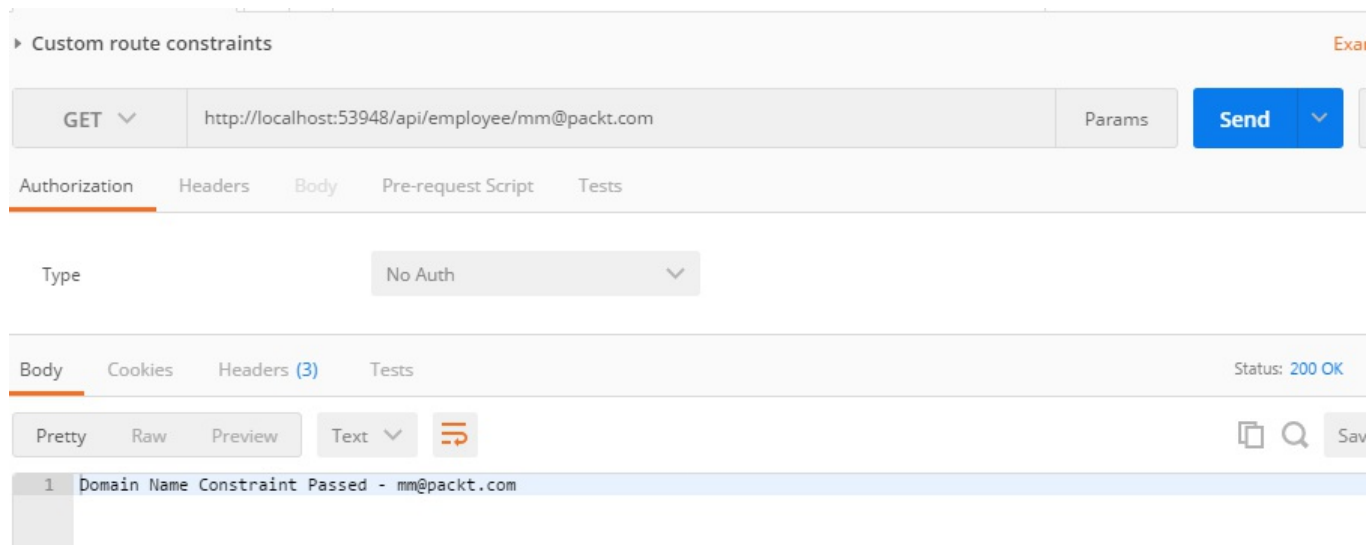
Attribute Routes

## Project Name - CustomRouteConstraints

Open the source code bundle in Visual Studio IDE

**How to run?**

- Choose *CustomRouteConstraints* project, *Set as Startup Project* to run this project
- Press F5 to run the application. *Port number might differ*
- Use Postman or any browser, access the endpoint http://localhost:53948/api/employee/mm@packt.com to view results as shown in below image

Custom Route Constraints



## Project Name - LinkGeneration

Open the source code bundle in Visual Studio IDE

**How to run?**

- Choose *LinkGeneration* project, *Set as Startup Project* to run this project
- Press F5 to run the application. *Port number might differ*
- Use Postman or any browser, access the endpoint http://localhost:54116/api/todos to view results as

shown in below image

Body of POST request

```
{
        "id":12,
        "name":"Book tickets"
}
```

Custom Route Constraints



# Chapter 6 - Middleware and Filters

Project Name - middleware-basics

Open the source code bundle in Visual Studio IDE

**How to run?**

- Choose *middleware-basics* project, *Set as Startup Project* to run this project
- Press F5 to run the application. *Port number might differ*
- Use any browser, access the endpoint. *Refer Chapter 6 for more explaination*

Project Name - builtin-middleware

Open the source code bundle in Visual Studio IDE

**How to run?**

- Choose *builtin-middleware* project, *Set as Startup Project* to run this project
- Press F5 to run the application. *Port number might differ*
- Use any browser, access the endpoint. *Refer Chapter 6 for more explaination*
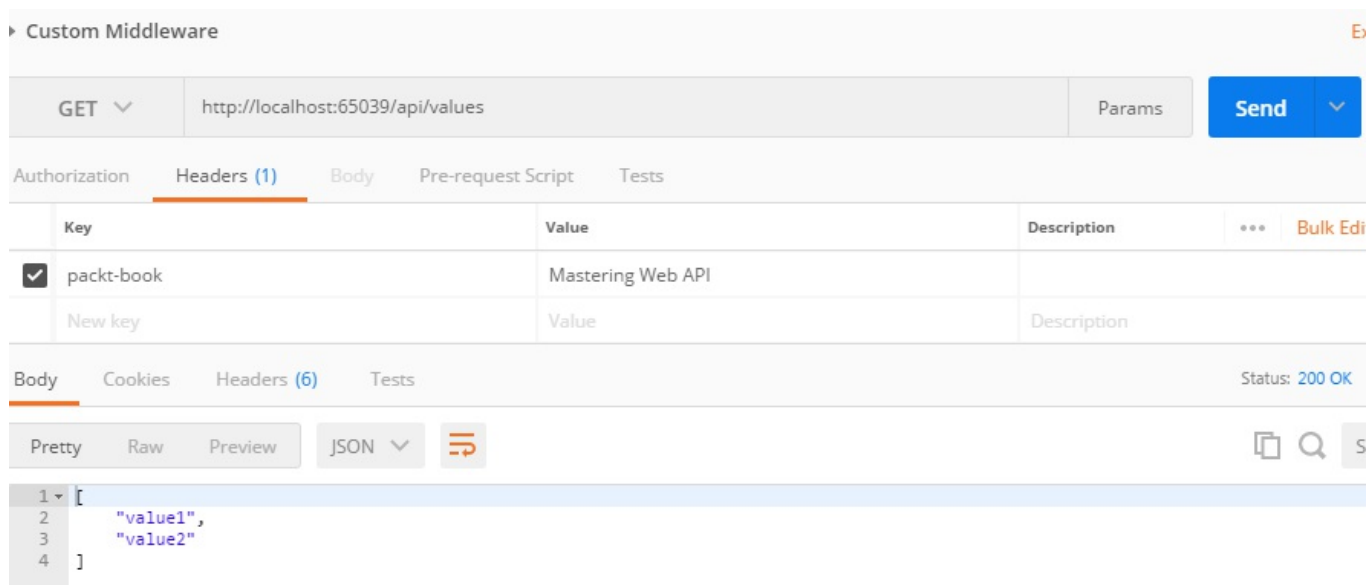
## Project Name - custom-middleware

Open the source code bundle in Visual Studio IDE

**How to run?**

- Choose *custom-middleware* project, *Set as Startup Project* to run this project
- Press F5 to run the application. *Port number might differ*
- Use Postman or any browser, access the endpoint http://localhost:65039/api/values

Add Headers **packt-book** with value **Mastering Web API** and call endpoint

Custom Middleware in action



## Project Name - httpmodule_to_middleware

Open the source code bundle in Visual Studio IDE

**How to run?**

- Choose *httpmodule_to_middleware* project, *Set as Startup Project* to run this project
- Press F5 to run the application. *Port number might differ*
- Use any browser, access the endpoint. *Refer Chapter 6 for more explaination*

## Project Name - filters-demo

Open the source code bundle in Visual Studio IDE

**How to run?**

- Choose *filters-demo* project, *Set as Startup Project* to run this project
- Press F5 to run the application. *Port number might differ*
- Use Postman or any browser, access the endpoint http://localhost:65265/api/values.

Custom Filters in action

Custom Exception Filter



```
1 ▾ {
2      "code": 500,
3      "message": "Employee Id Cannot be Zero"
4 }
```

# Chapter 8 - Web API Security

Project Name - PersonalBudget

Open the source code bundle in Visual Studio IDE

**How to run?**

- Choose *PersonalBudget* project, *Set as Startup Project* to run this project
- Press F5 to run the application. *Port number might differ*

**CORS**

*WebDemoAPP1 & WebDemoAPP2* used to call Web API using AJAX.

1. npm install from command line (NodeJS should be installed)
2. npm run dev to start the web apps

However WebDemoAPP2 is not present in Origin policy of Web API. CORS not enabled in this project

**Data Protection API**

- Choose *PersonalBudget* project, *Set as Startup Project* to run this project
- Press F5 to run the application. *Port number might differ*
- Using POST method, pass on below body contents. It gets added, returns id in encrypted format.
- Later you can use GET method to retrieve records

```
{
        "name":"Food",
        "amount":100
}
```

Data Protection API in action

POST ∨   http://localhost:51053/api/budgetcategory   Params   Send ∨

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

○ form-data   ○ x-www-form-urlencoded   ● raw   ○ binary   JSON (application/json) ∨

1 ▾ {
2       "name":"Food",
3       "amount":100|
4   }

Body   Cookies   Headers (4)   Tests                              Status: 200 OK   Ti

Pretty   Raw   Preview   JSON ∨

1 ▾ {
2       "encryptId": "CfDJ8CUx7sCYTQNEh3cmGnIZISRB4h0MTNuzH93dPCC
            -cWjrPQ1VX5zAiIUHY2wutZwaKoRHKwL2queY8yG19yK93KkFrK4KZMNjUSbw_54MnhJo0D7XizvL1AYFHOALAYiDtw",
3       "name": "Food",
4       "amount": 100
5   }

**JWT - JSON Web Tokens**

- Choose *PersonalBudget* project, *Set as Startup Project* to run this project
- Press F5 to run the application. *Port number might differ*
- Using POST method, pass on below body contents. It returns JWT if authentication succcess

```
{
        "userName":"mithunvp",
        "password":"abcd123"
}
```

Generating JSON Web Token (JWT) in action

## Validating JWT in action



**Claims based Authentication**

- Choose *PersonalBudget* project, *Set as Startup Project* to run this project
- Press F5 to run the application. *Port number might differ*
- Using POST method, pass on below body contents. It returns JWT if authentication succcess

```
{
        "userName":"yogi",
        "password":"abcd123"
}
```

If you the JWT generated to add BudgetCategory, it fails because its not Super User

# Chapter 9 - Integration with database

(EF6 example) Project Name - AdvWorks.WebAPI

Open the source code bundle in Visual Studio IDE

**How to run?**

- Ensure that AdventureWorks2014 database is setup on your machine
- Choose *AdvWorks.WebAPI* project, *Set as Startup Project* to run this project
- Press F5 to run the application. *Port number might differ*

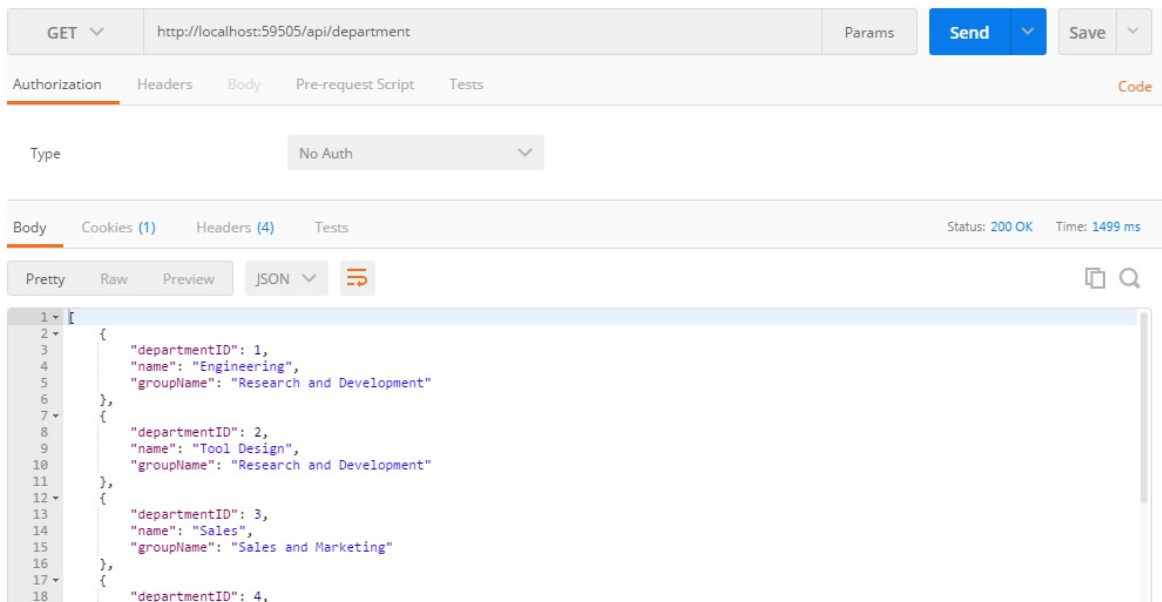ASP.NET Core Web API with EF 6 on full framework in action



(Dapper) Project Name - AdvWrksDapper

Open the source code bundle in Visual Studio IDE

**How to run?**

- Ensure that AdventureWorks2014 database is setup on your machine
- Choose *AdvWrksDapper* project, *Set as Startup Project* to run this project
- Press F5 to run the application. *Port number might differ*

Web API with Dapper for GET method

```
GET ∨    http://localhost:59505/api/department              Params    Send ∨    Save ∨

Authorization   Headers   Body   Pre-request Script   Tests                              Code

Type                    No Auth              ∨

Body   Cookies (1)   Headers (4)   Tests                      Status: 200 OK   Time: 1499 ms

Pretty   Raw   Preview   JSON ∨

 1  [
 2      {
 3          "departmentID": 1,
 4          "name": "Engineering",
 5          "groupName": "Research and Development"
 6      },
 7      {
 8          "departmentID": 2,
 9          "name": "Tool Design",
10          "groupName": "Research and Development"
11      },
12      {
13          "departmentID": 3,
14          "name": "Sales",
15          "groupName": "Sales and Marketing"
16      },
17      {
18          "departmentID": 4,
```

(EF Core) Project Name - PacktContactsCore

Open the source code bundle in Visual Studio IDE

**How to run?**

- Ensure that DB migrations step is performed. Refer *EF tools for database migrations and updates* in Chapter 9
- Choose *PacktContactsCore* project, *Set as Startup Project* to run this project
- Press F5 to run the application. *Port number might differ*

Use Postman to perform GET, POST, PUT, DELETE

# Chapter 9 - Modern Web Frontend

Additional Software Required along with previous mentioned

- Install latest version of NodeJS

```
username - mithunvp
password - abcd123
```

Project Name - ngPacktContacts (frontend) and PacktContacts of Chapter 12

Web frontend is built in Angular i.e. Angular 4

- Install Angular CLI by running npm install -g angular-cli

**How to run?**

- Deploy or run from VS the Chapter 12 project - **PacktContacts**
- Run the command npm install from root folder of Angular project **ngPacktContacts**. Will take few

minutes to download all packages
- Once done with download, run ng serve

*Change the endpoint Web API url used in frontend project according to your setup*

## Project Name - packtcontactapp (frontend) and PacktContacts of Chapter 12

Hybrid mobile frontend is built in Ionic 3

- Install Angular CLI by running npm install -g ionic

**How to run?**

- Deploy or run from VS the Chapter 12 project - **PacktContacts**
- Run the command npm install from root folder of Ionic 3 project **packtcontactapp**. Will take few minutes to download all packages
- Once done with download, run ionic serve

*Change the endpoint Web API url used in frontend project according to your setup*

## Project Name - PacktContactsJsUI (frontend) and PacktContacts of Chapter 12

ASP.NET web site uses JavaScript, JQuery or ReactJS to consume Web API

**How to run?**

- Deploy or run from VS the Chapter 12 project - **PacktContacts**
- Run the *PacktContactsJsUI* project to view below page, click respective UI frameworks to work with

*Change the endpoint Web API url used in frontend project according to your setup*

UI frameworks

ⓘ localhost:57341

# Mastering ASP.NET Core Web API

Consuming Web API using JavaScript or JQuery or ReactJS

Using Javascript   Using JQuery   Using ReactJS