

## **How to Publish ASP.NET Core RESTful API to Azure**

In this tutorial I'm going to show how to:

- Create a SQL Database
- Create a .Net Core RESTful API
- Deploy REST API to Azure
- Test REST API using Postman

You can find the complete source code here:

<https://github.com/patricksameerajayalath/Net-Core-RESTAPI>

### **Step 01: Create a SQL Database**

We are going to create new SQL Database and add a table called Employee. We will be having bellow structure for an employee.

Column Name	Type
Name	String
Age	Int
DOB	Date
Manager	Bool

## Create a SQL Database.

- Resource group: patsam-rg
- Database name: patsam-sqlldb
- Server: patsam
- Region: Central US

[Home](#) > [SQL databases](#) > Create SQL Database

### Create SQL Database

Microsoft

[Basics](#) [Networking](#) [Additional settings](#) [Tags](#) [Review + create](#)

Create a SQL database with your preferred configurations. Complete the Basics tab then go to Review + Create to provision with smart defaults, or visit each tab to customize. [Learn more](#)

#### Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ⓘ

Visual Studio Enterprise – MPN

Resource group \* ⓘ

patsam-rg

[Create new](#)

#### Database details

Enter required settings for this database, including picking a logical server and configuring the compute and storage resources

Database name \*

patsam-sqlldb

Server \* ⓘ

(new) patsam ((US) West Central US)

[Create new](#)

Want to use SQL elastic pool? \* ⓘ

☐ Yes ☒ No

Compute + storage \* ⓘ

**Basic**

2 GB storage

[Configure database](#)

[Review + create](#)

Next : Networking >

Home > SQL databases

### SQL databases

100 Services List

+ Add Reservations Edit columns Refresh Assign tags Delete

Try our new Azure SQL resource browser! This experience offers a unified view of all your SQL Server resources in Azure as well as improved sorting and filtering. Click here to go to the new experience.

Subscriptions: Visual Studio Enterprise – MPN – Don't see a subscription? Open Directory + Subscription settings

Filter by name... All resource groups All locations All tags No grouping

1 items

Name	Status	Replication role	Server	Pricing tier	Location	Subscription
patmam-sqladb	Online	None	patmam	Basic	West Central US	Visual Studio Enterprise – MPN

---

Home > SQL databases > patmam-sqladb (patmam/patmam-sqladb)

### patmam-sqladb (patmam/patmam-sqladb)

SQL database

+ Add Reservations ... More

Search (Ctrl+F)

Overview Activity log Tags Diagnose and solve problems Quick start Query editor (preview)

Power Platform

- Power BI (preview)
- PowerApps (preview)
- Flow (preview)

Settings

- Configure
- Geo-Replication
- Connection strings
- Sync to other databases
- Add Azure Search
- Properties
- Locks
- Export template

Integrations

- Stream analytics (preview)

Security

- Advanced data security
- Auditing
- Dynamic Data Masking
- Transparent data encryption

Intelligent Performance

Copy Restore Export Set server firewall Delete Connect with... Feedback

Resource group (change): patmam-rg

Status: Online

Location: West Central US

Subscription (change): Visual Studio Enterprise – MPN

Subscription ID: 3822b9c0-bf7e-455e-a1d1-8717b1b4150b

Tags (change): Click here to add tags

Server name: patmam.database.windows.net

Elastic pool: No elastic pool

Connection strings: Show database connection strings

Pricing tier: Basic

Earliest restore point: No restore point available

Show data for last: 1 hour 24 hours 7 days

Aggregation type: Max

Compute utilization

0%

Database data storage

Used space: 4 MB

Allocated space: 16 MB

Maximum storage size: 2 GB

0.2% USED SPACE

Notifications (0) Database features (6)

All Security (4) Performance (1) Recovery (1)

Transparent data encryption

Encryption at rest for your databases, backups, and logs.

CONFIGURED

Advanced data security

Data Discovery & Classification, Vulnerability Assessment and Advanced Threat Protection.

NOT CONFIGURED

Configure the Firewall settings to allow accessing it.

Home > SQL databases > patmam-sqladb (patmam/patmam-sqladb) > Firewall settings

### Firewall settings

patmam (SQL server)

Save Discard + Add client IP

Connections from the IPs specified below provides access to all the databases in patmam.

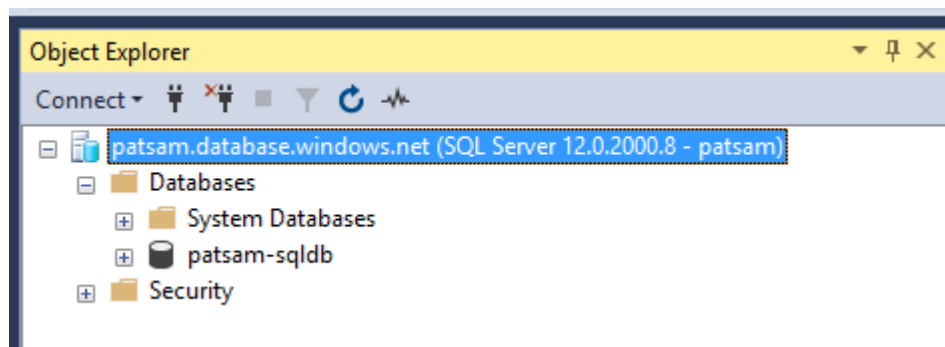
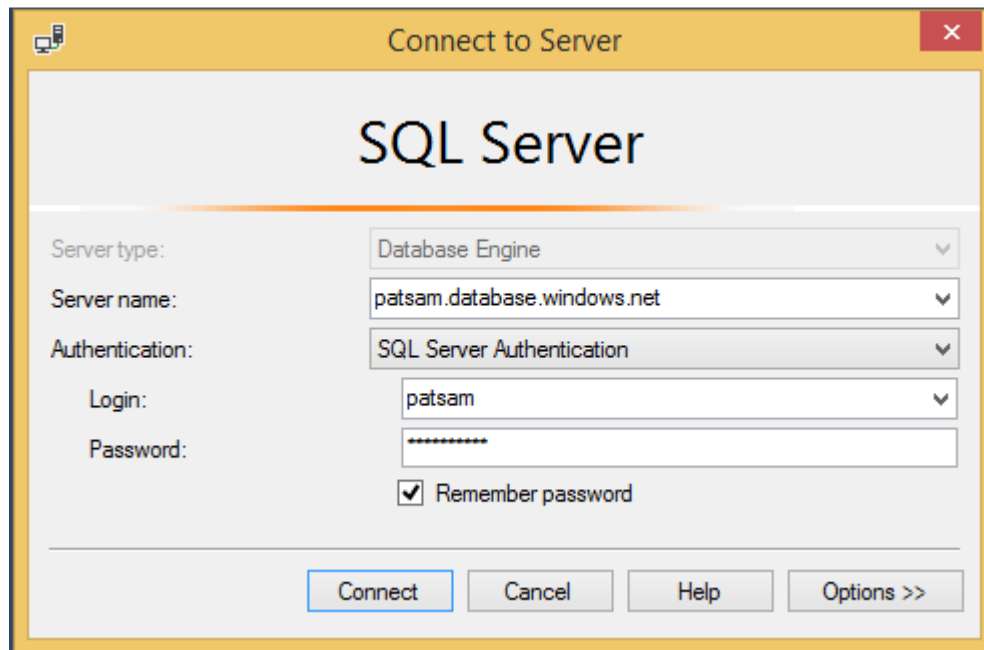
Allow Azure services and resources to access this server

ON OFF

Client IP address

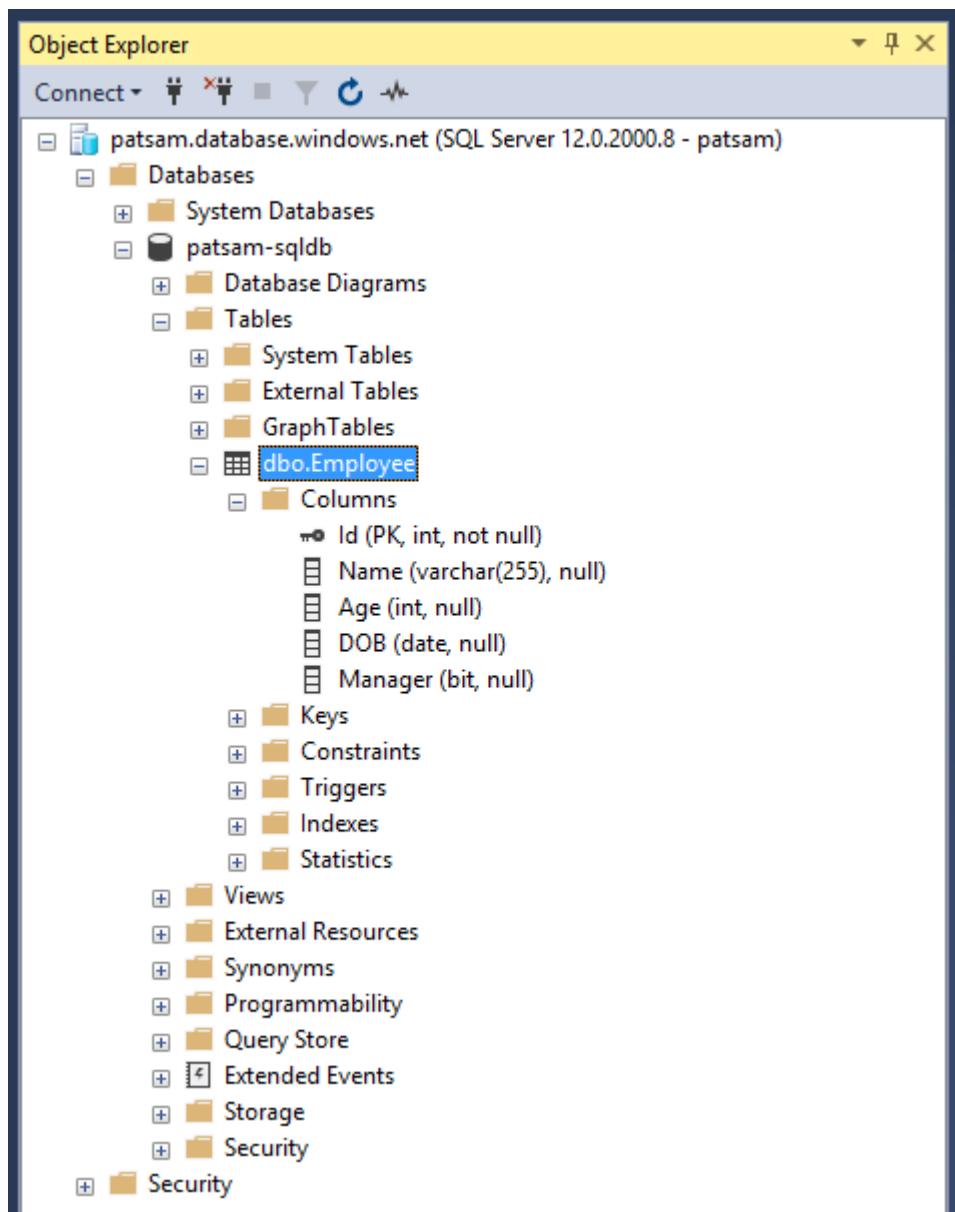
Rule name	Start IP	End IP	
			...
All	0.0.0.0	255.255.255.255	...

Connect to the Database through SQL Server Management Studio.



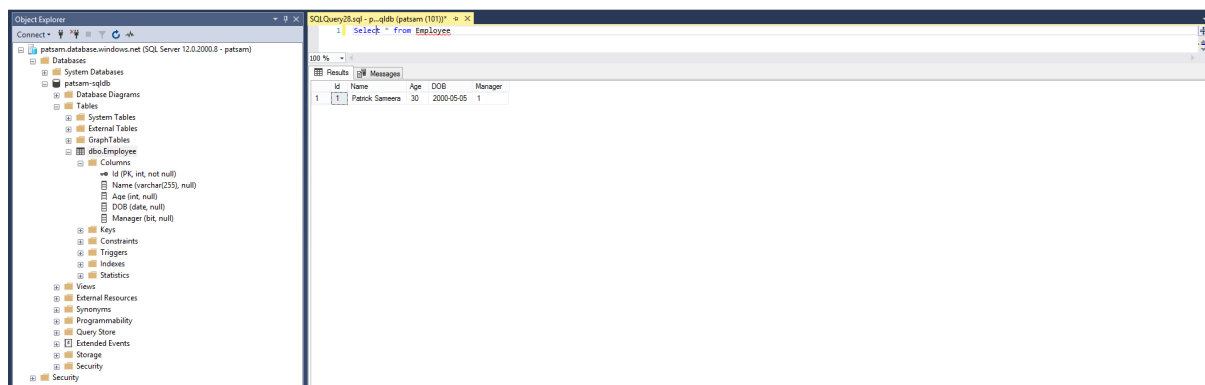
Next create a new table called Employee.

```
CREATE TABLE Employee (  
    Id Int IDENTITY(1,1) PRIMARY KEY,  
    Name Varchar(255),  
    Age Int,  
    DOB Date,  
    Manager Bit  
);
```



Insert some data to the table.

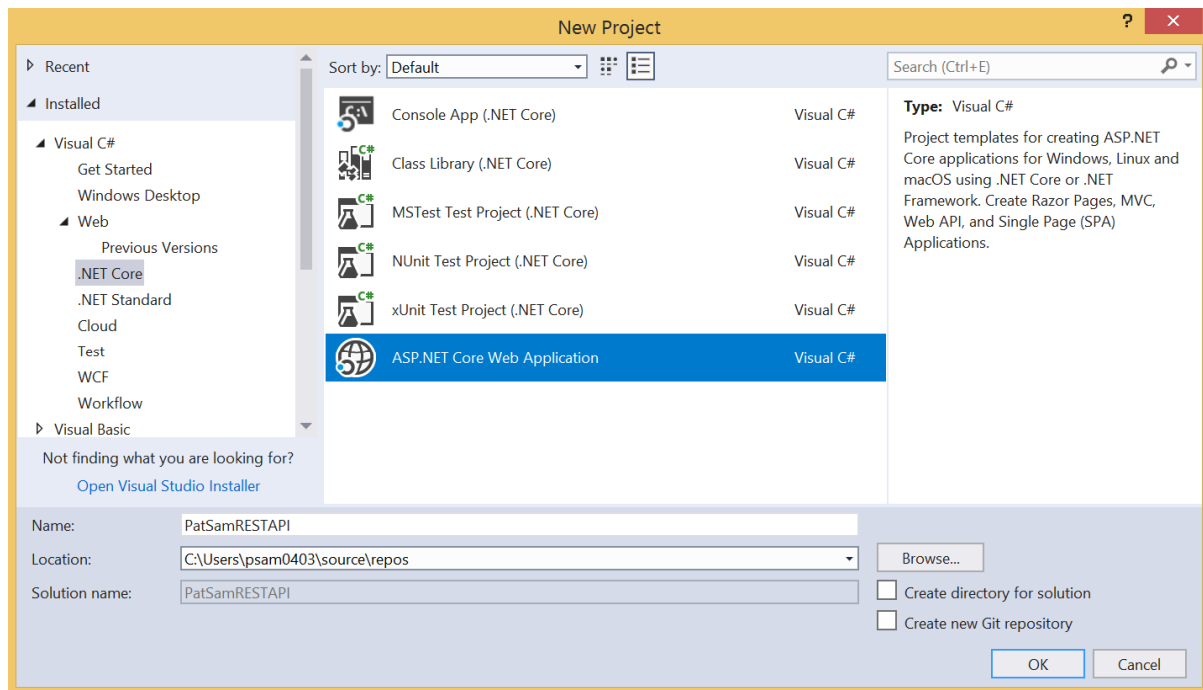
*Insert Into Employee*  
*Values ('Patrick Sameera', 30, '05/05/2000', 1)*



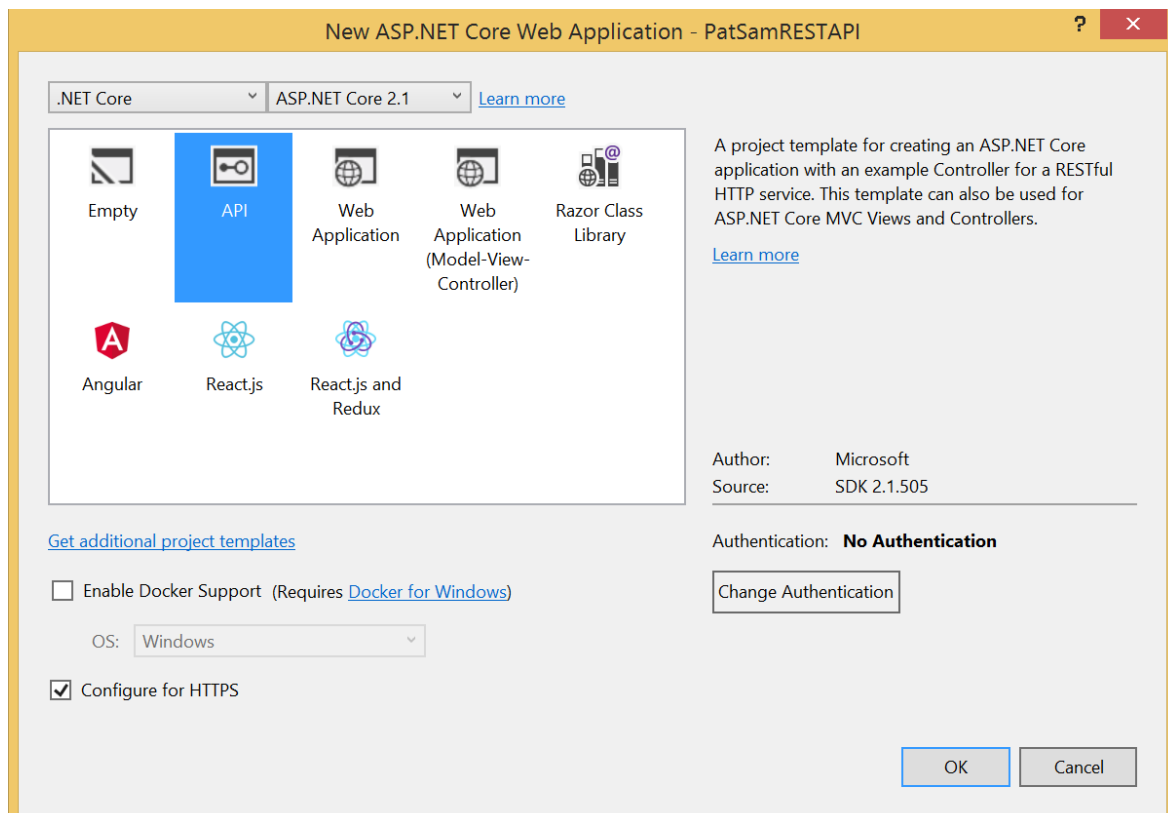
Patrick Sameera Jayalath

## Step 02: Create a .Net Core RESTful API

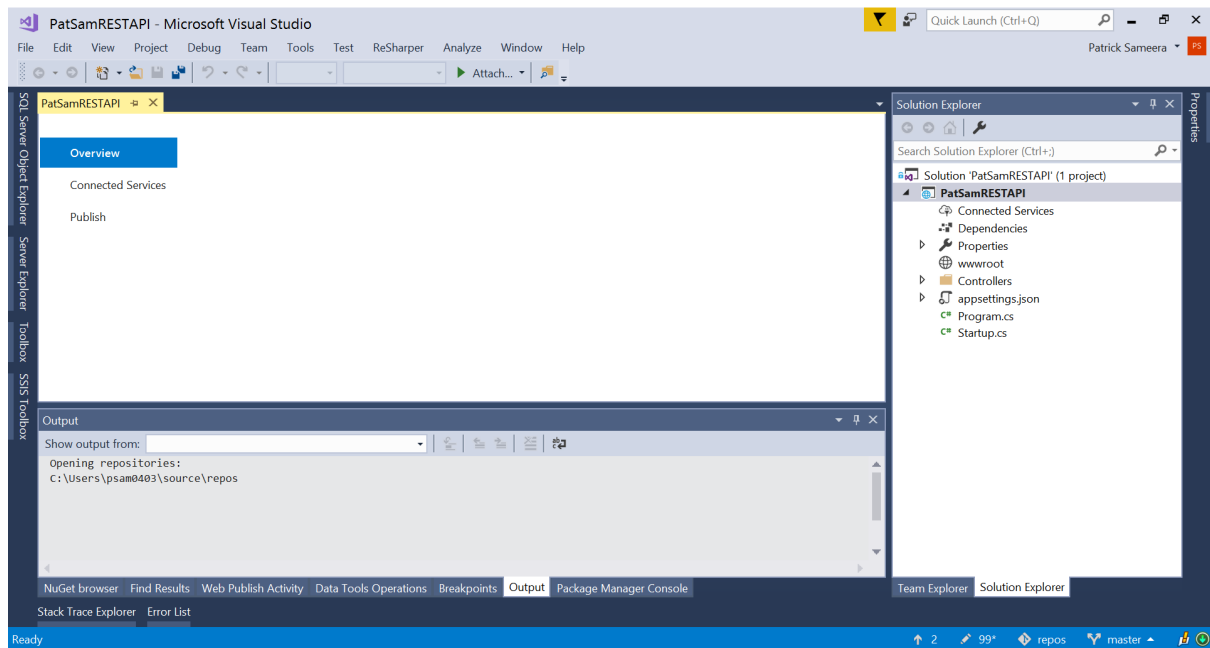
First, we need to create a ASP.NET Core Web Application.



- .Net Core 2.1
- API
- Configure for HTTPS ticked



This is how the project folder structure looks like.



Run the project to make sure it compiles and runs properly.



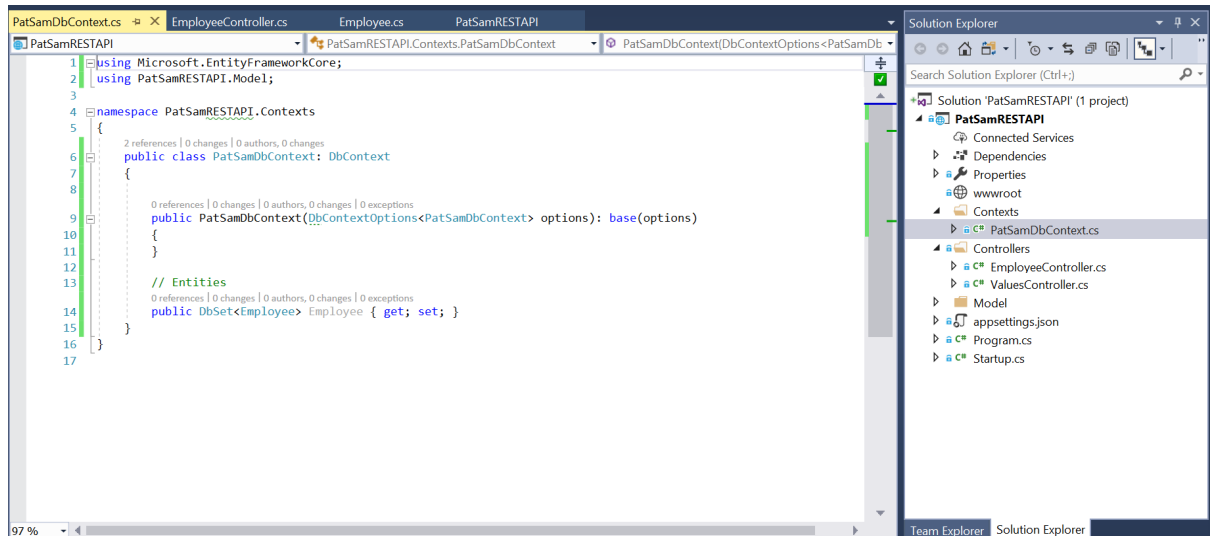
Next add/do bellow changes to the code.



## PatSamDbContext.cs

Create a folder called “Contexts” and add the PatSamDbContext.cs inside that.

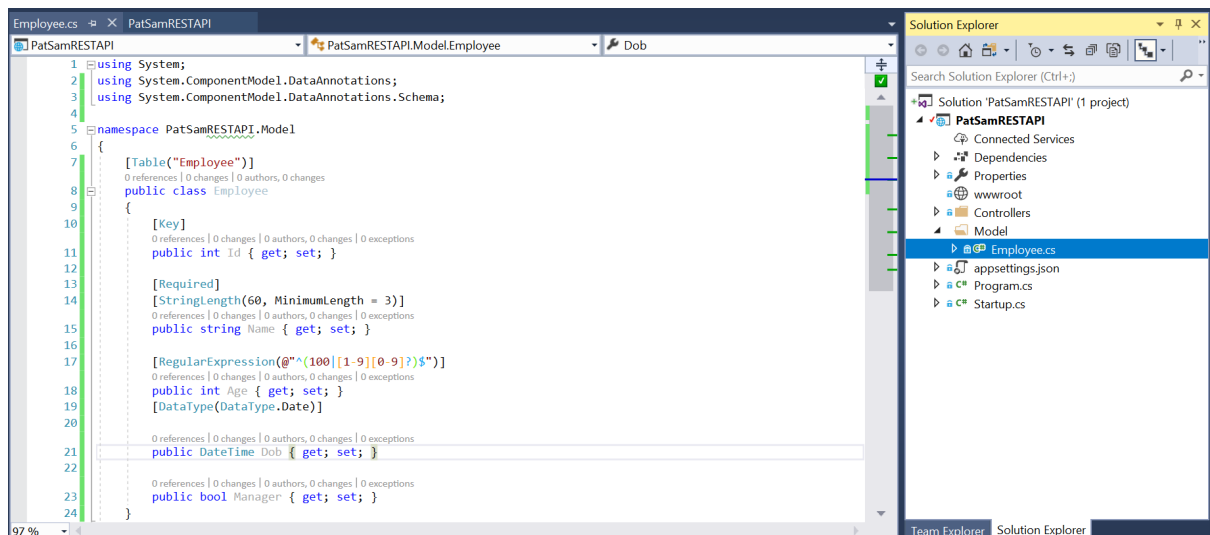
<https://github.com/patricksameerajayalath/Net-Core-RESTAPI/blob/master/Contexts/PatSamDbContext.cs>



## Employee.cs

Next create a folder called “Model” and add the Employee.cs inside that.

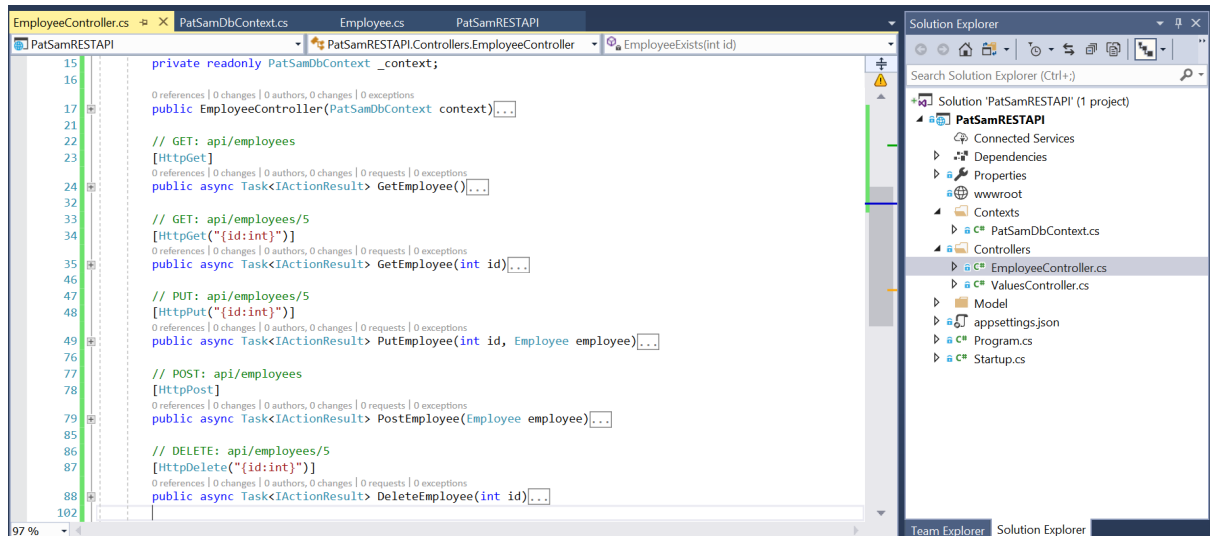
<https://github.com/patricksameerajayalath/Net-Core-RESTAPI/blob/master/Model/Employee.cs>



## EmployeeController.cs

Next create EmployeeController.cs

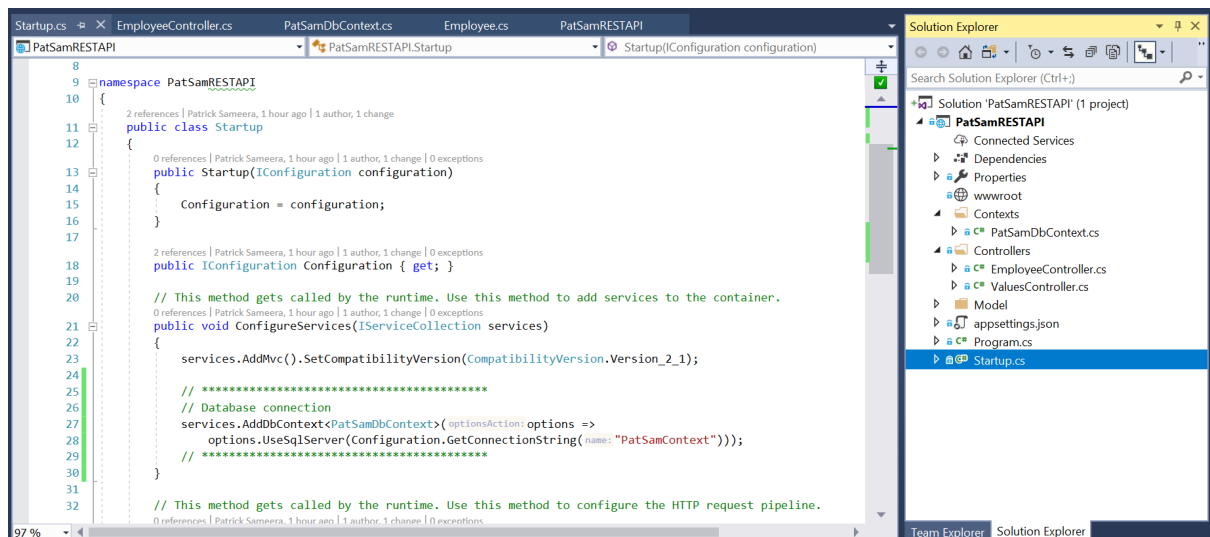
<https://github.com/patricksameerajayalath/Net-Core-RESTAPI/blob/master/Controllers/EmployeeController.cs>



## Startup.cs

Do bellow changes in Startup.cs

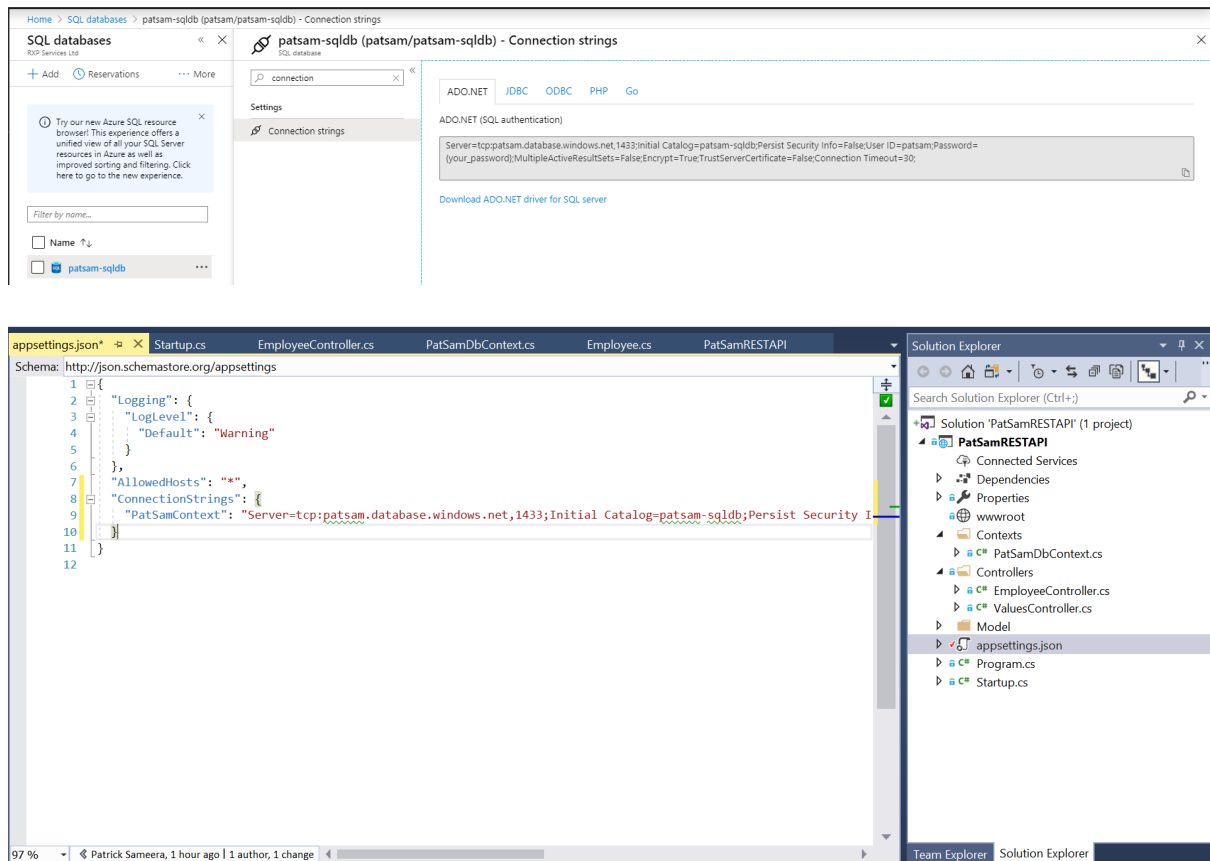
<https://github.com/patricksameerajayalath/Net-Core-RESTAPI/blob/master/Startup.cs>



## appsettings.json

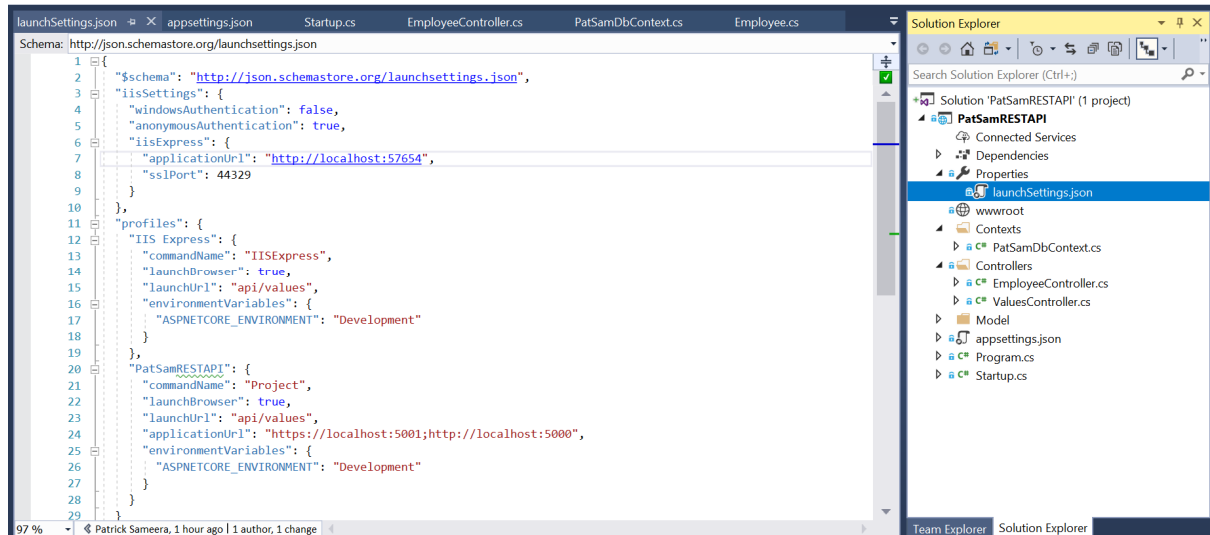
Add the Database connection string to the appsettings.json file.

<https://github.com/patricksameerajayalath/Net-Core-RESTAPI/blob/master/appsettings.json>



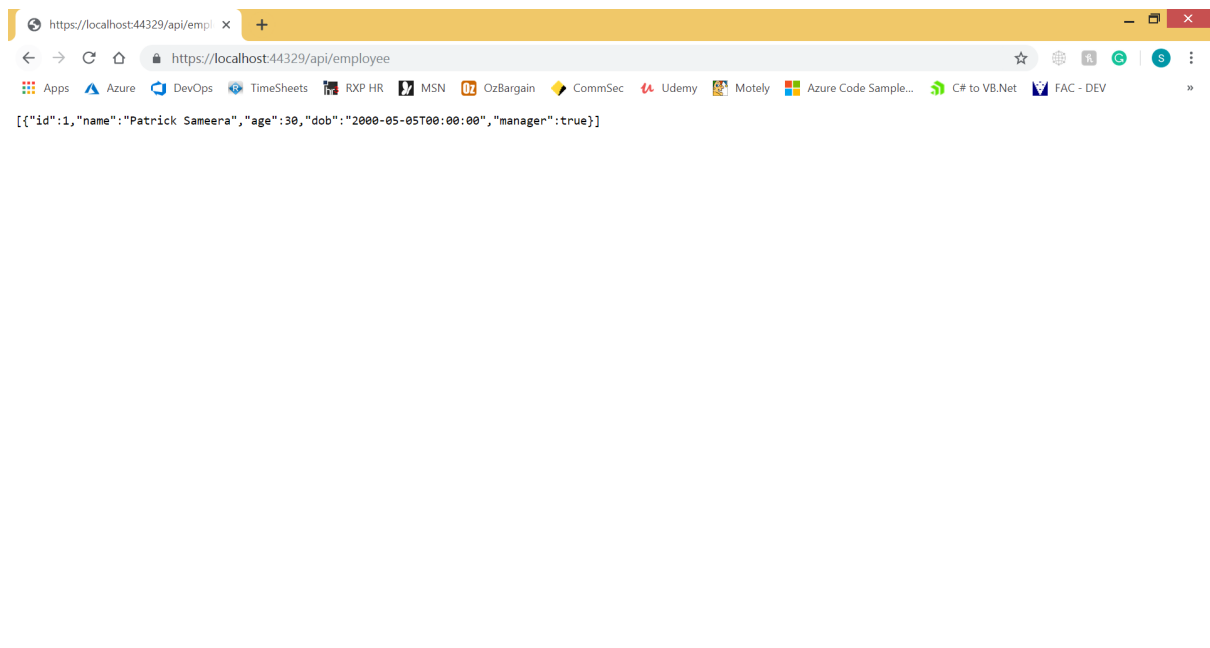
## launchSettings.json

Default URL `https://localhost:{port}/api/values` comes from the `launchUrl` property in the `Properties\launchSettings.json`:



Run the Project.

<https://localhost:44329/api/employee>



### Step 03: Deploy REST API to Azure

Create API App.

- Resource group: patsam-rg (existing)
- App service plan: patsam-sp (existing)

[Home](#) > API App

## API App

Create

**App name \***

✓

.azurewebsites.net

**Subscription \***

▼

**Resource Group \*** ⓘ

☐ Create new ☒ Use existing

▼

**\*App Service plan/Location** >

patsam-sp(Central US)

**Application Insights** >

patsamrestapi

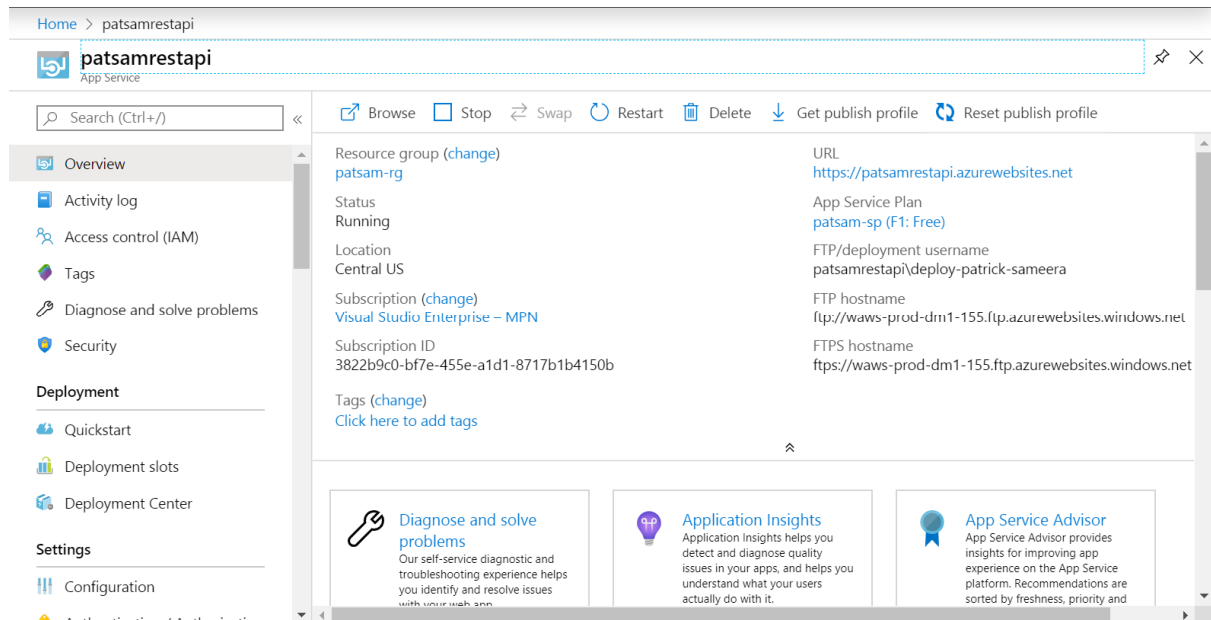
Create

Automation options

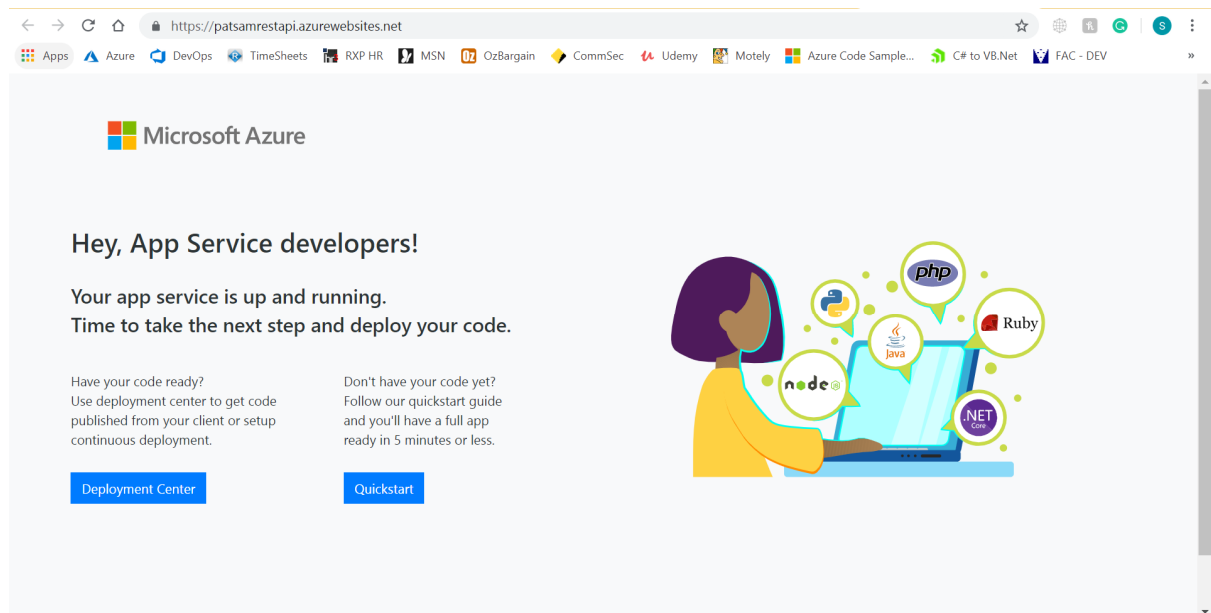
Once the API App got created successfully go inside the resource.

Click Browse to API App.

<https://patsamrestapi.azurewebsites.net>

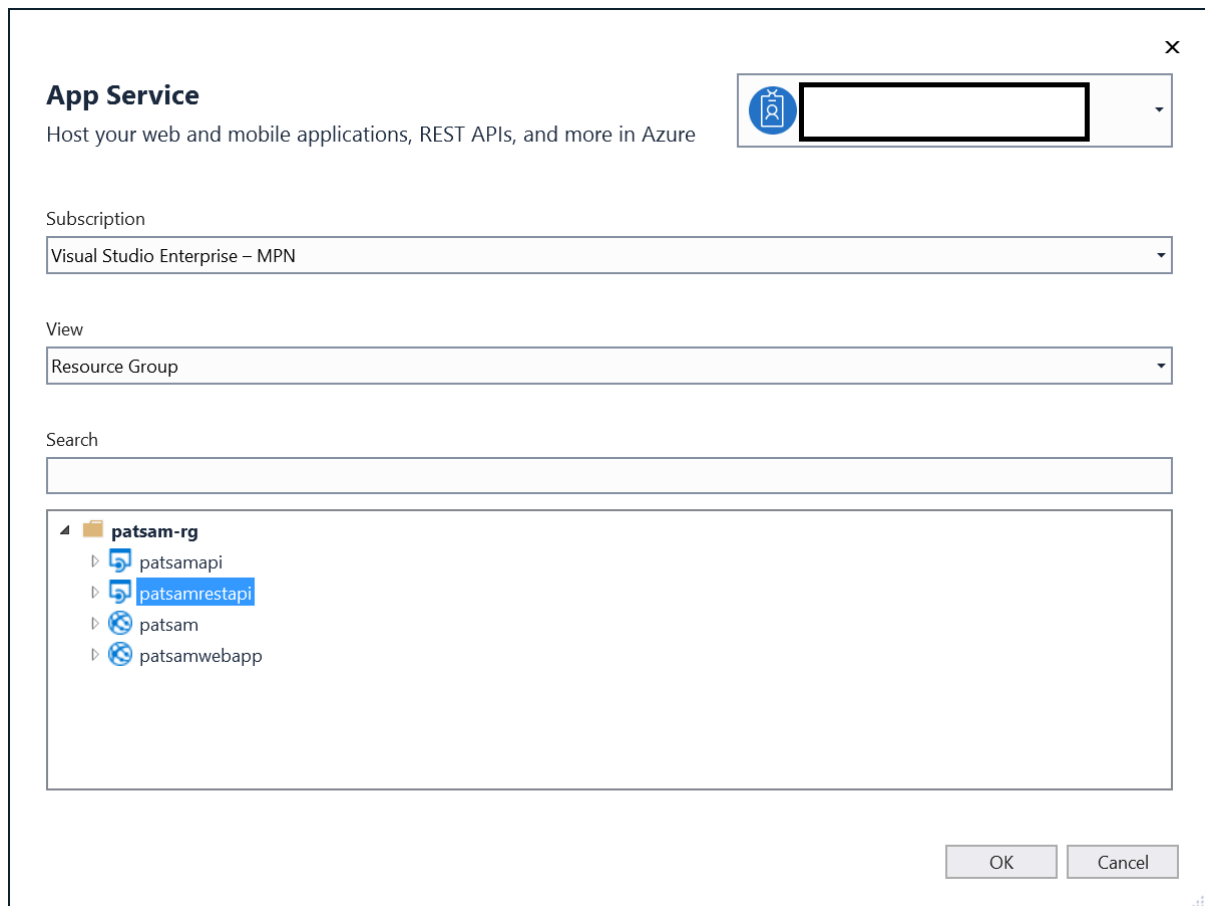


Make sure it works properly.

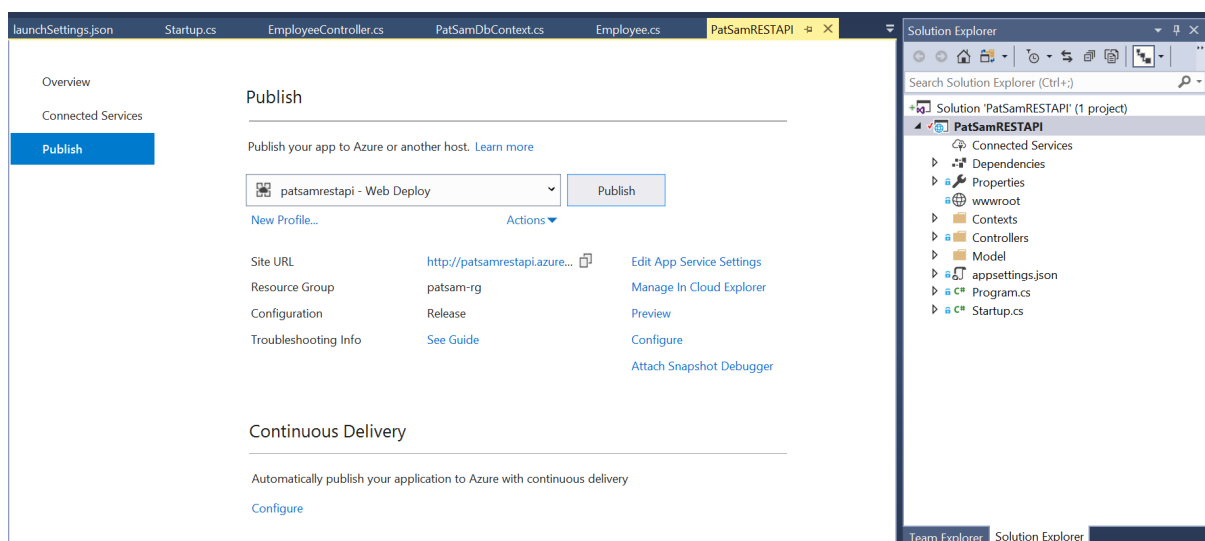


Next publish the REST API to the newly created API App.

Select the API App we created earlier.



Once the profile been set publish it.



Make sure it gets published successfully.



```
Output
Show output from: Build
Adding file (patsamapi\System.Composition.Hosting.dll).
Adding file (patsamapi\System.Composition.Runtime.dll).
Adding file (patsamapi\System.Composition.TypedParts.dll).
Adding file (patsamapi\web.config).
Publish Succeeded.
Web App was published successfully http://patsamapi.azurewebsites.net/
===== Build: 2 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
===== Publish: 1 succeeded, 0 failed, 0 skipped =====
```

Make sure the Web API works fine after the publish.

<https://patsamrestapi.azurewebsites.net/api/employee>



```
https://patsamrestapi.azurewebsites.net/api/employee
[{"id":1,"name":"Patrick Sameera","age":30,"dob":"2000-05-05T00:00:00","manager":true}]
```

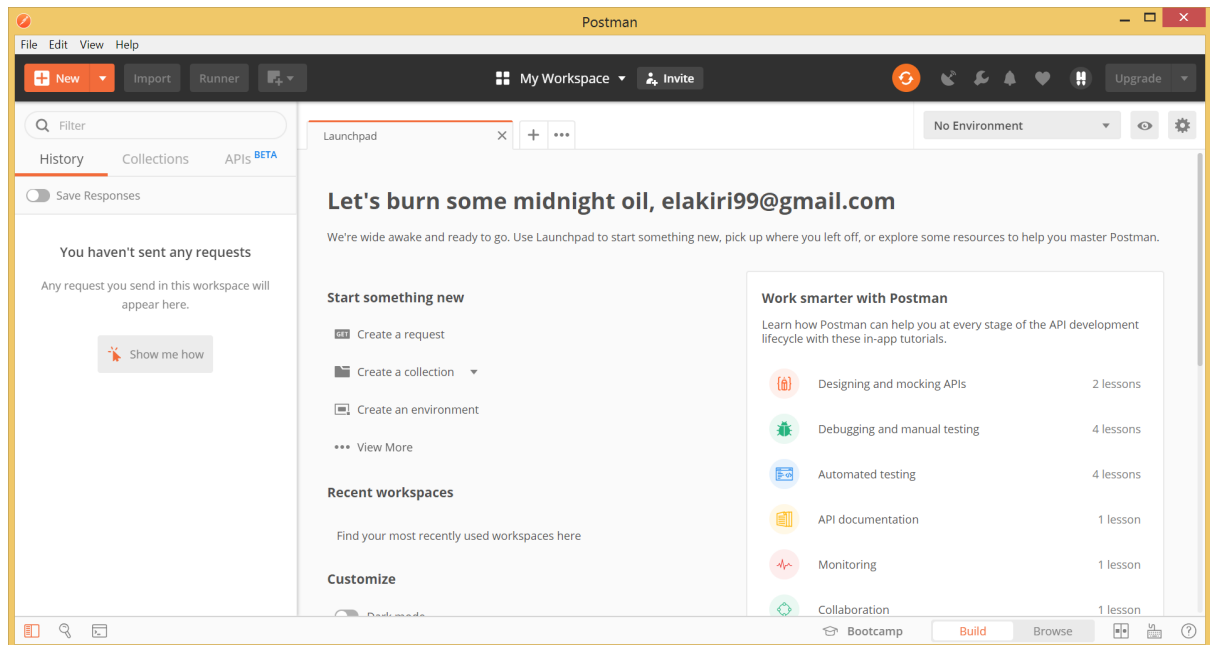


## Step 04: Test REST API using Postman

You can download Postman here:

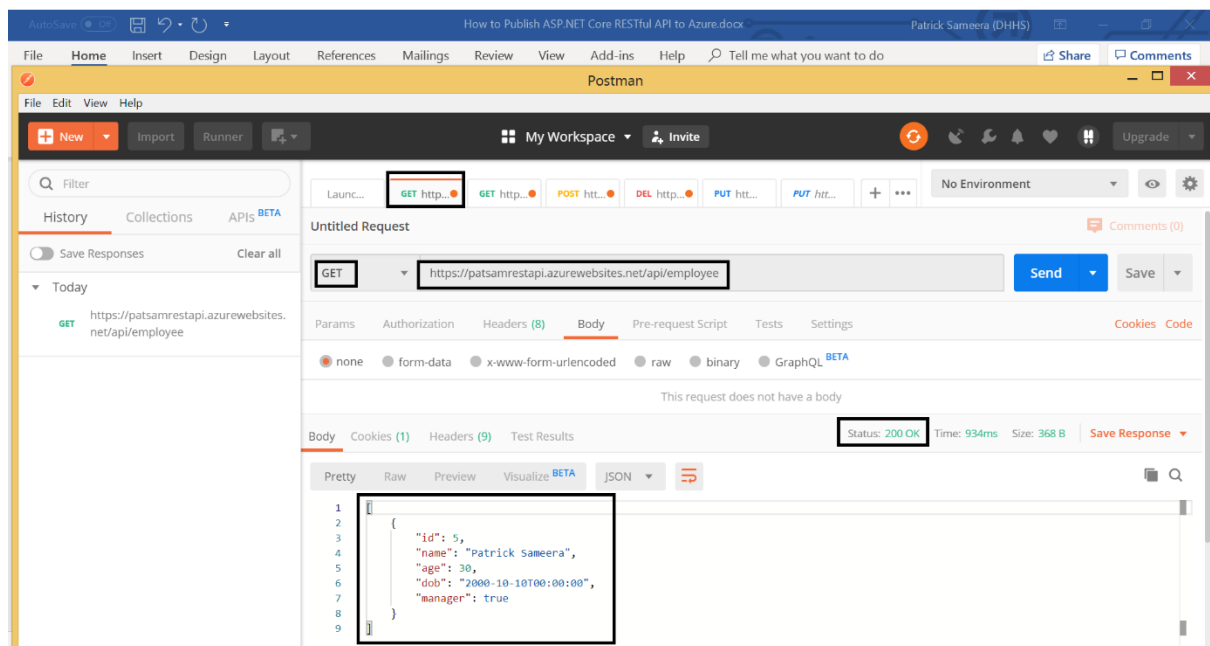
<https://www.getpostman.com/downloads/>

Open Postman.



## Get all employee details

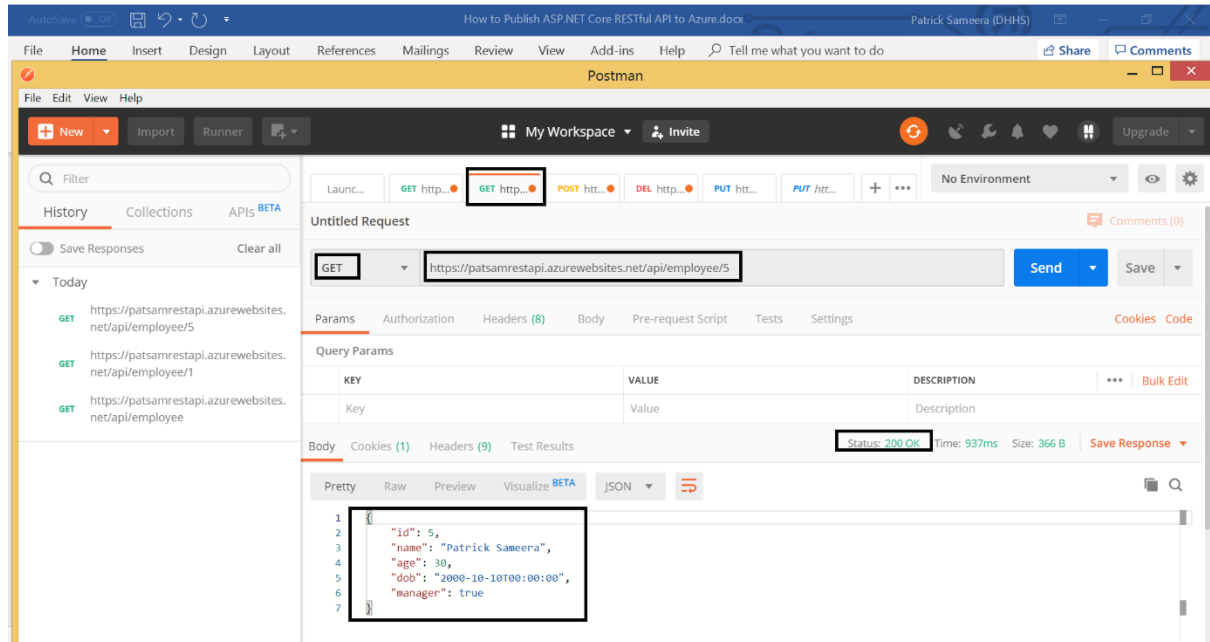
- GET
- <https://patsamrestapi.azurewebsites.net/api/employee>



Patrick Sameera Jayalath

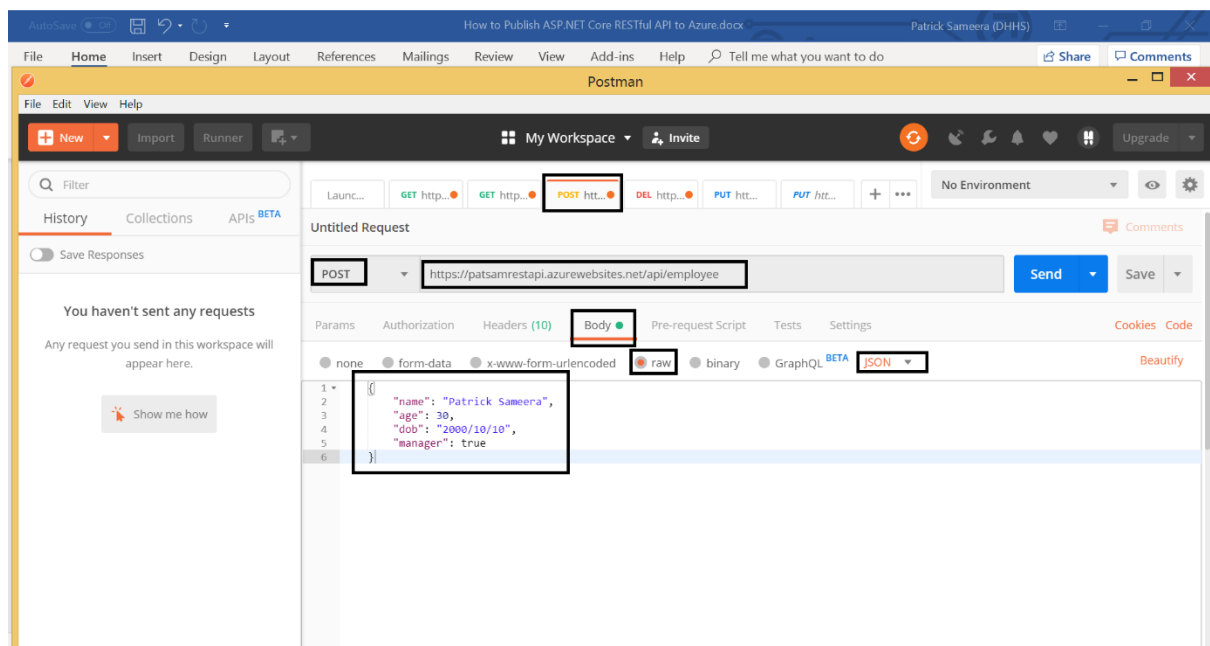
## Get single employee detail

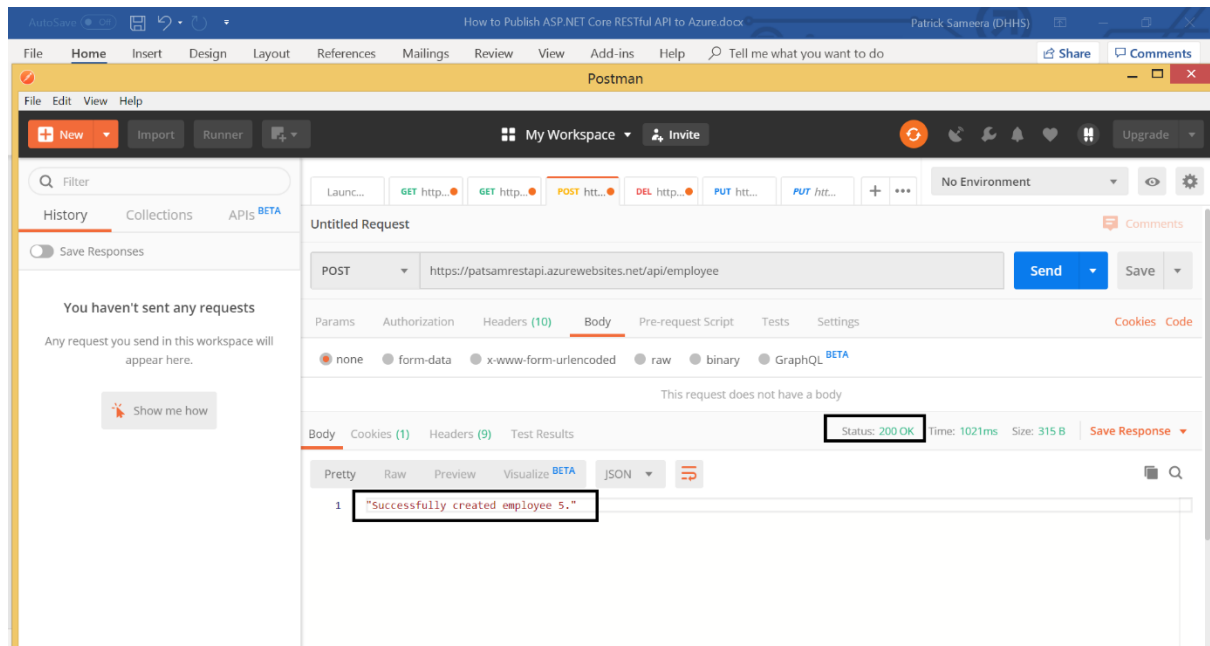
- GET
- <https://patsamrestapi.azurewebsites.net/api/employee/5>



## Add employee detail

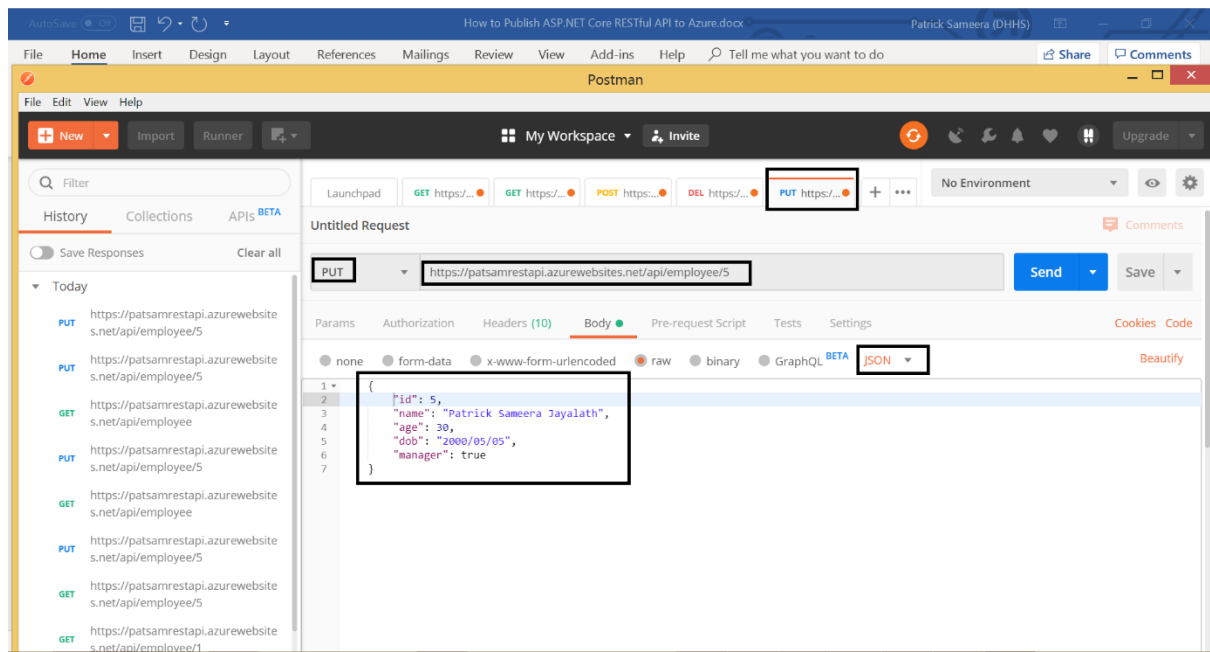
- POST
- <https://patsamrestapi.azurewebsites.net/api/employee>



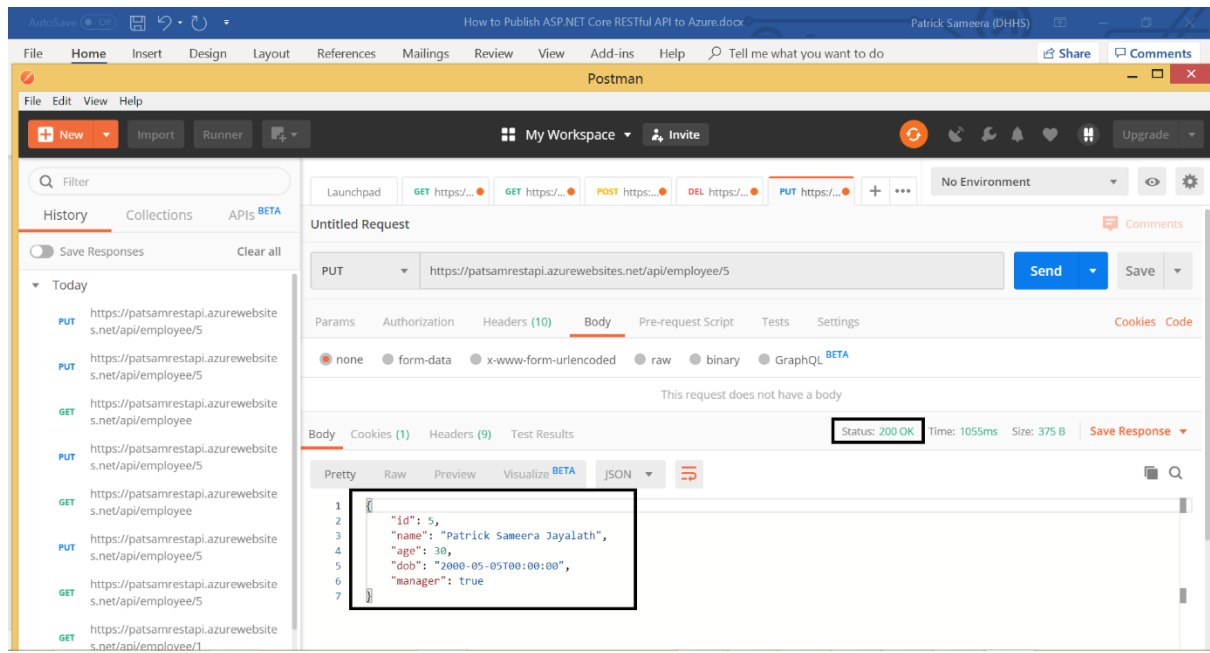


## Update employee detail

- PUT
- <https://patsamrestapi.azurewebsites.net/api/employee/5>

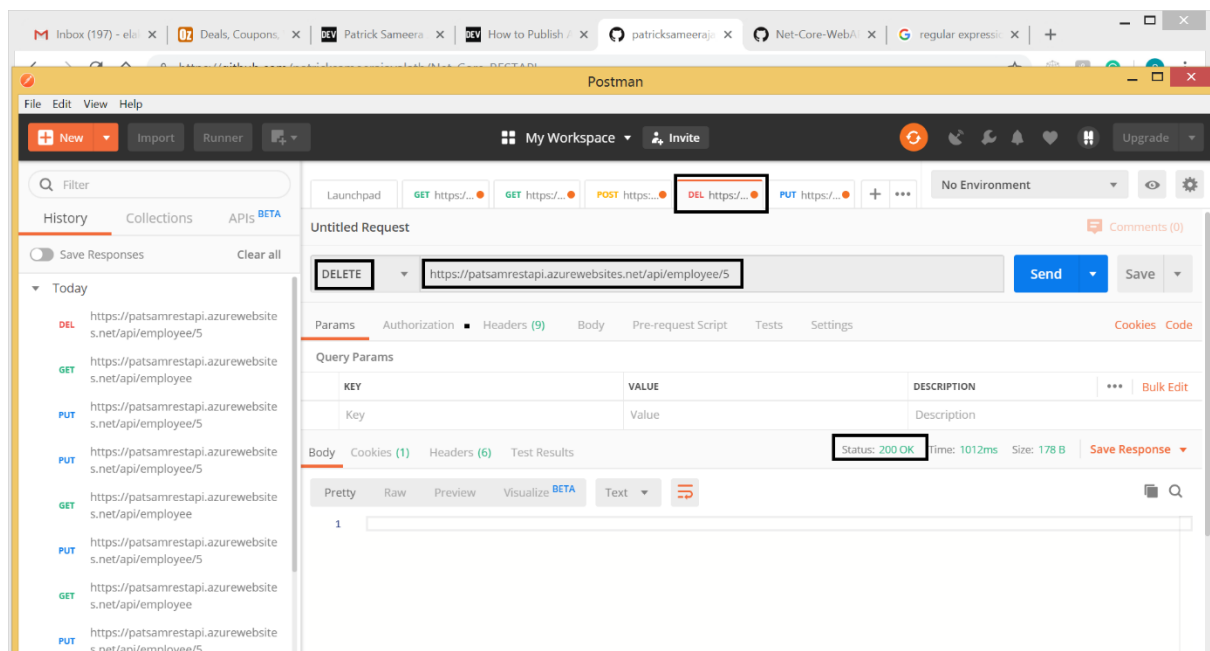


Patrick Sameera Jayalath



## Delete employee detail

- DELETE
- <https://patsamrestapi.azurewebsites.net/api/employee/5>



We can also check some API Validations through Postman.

