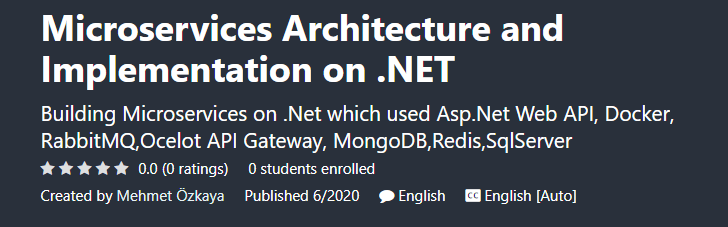
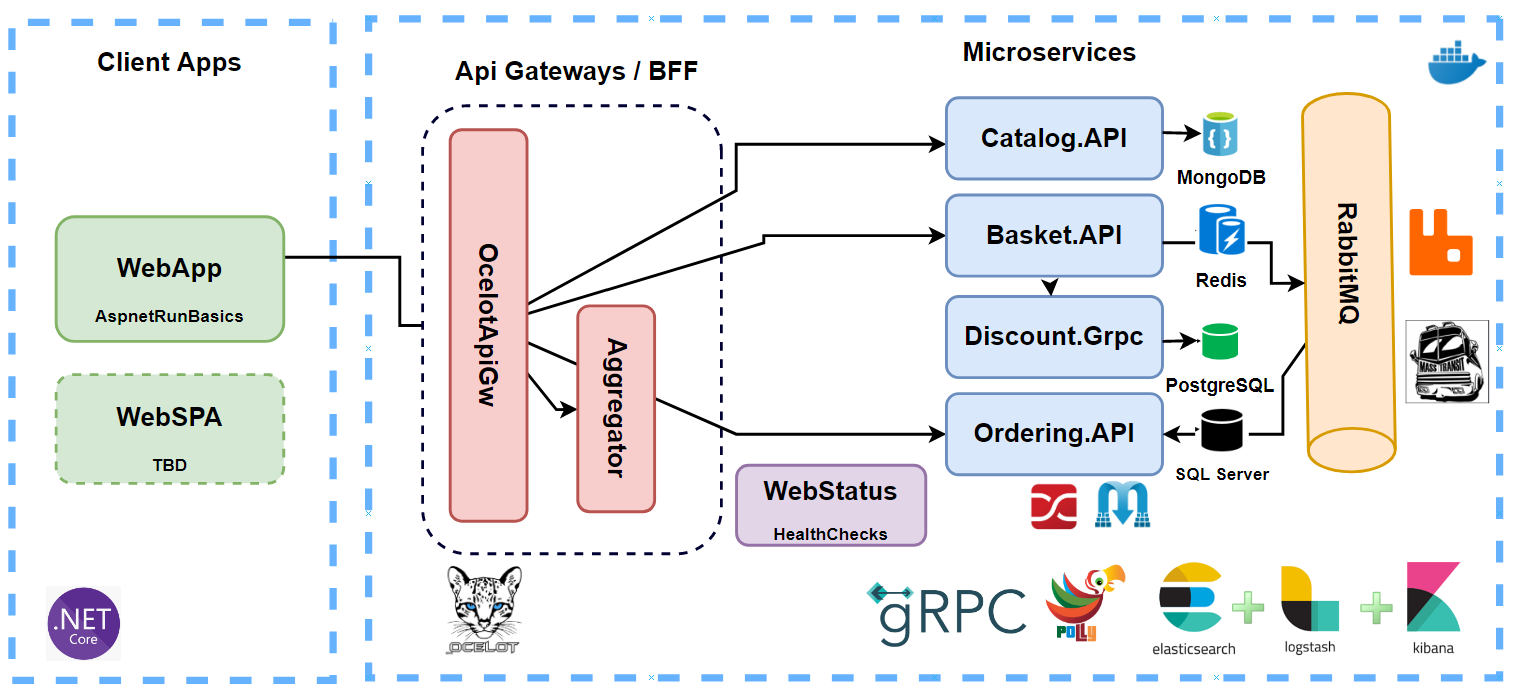
**AspnetMicroservices**

This repository prepared for the below udemy course.

[](https://www.udemy.com/course/microservices-architecture-and-implementation-on-dotnet/?couponCode=FA24745CC57592AB612A)

**UDEMY COURSE WITH DISCOUNTED - Step by Step Development of this repository ->**[**https://www.udemy.com/course/microservices-architecture-and-implementation-on-dotnet/?couponCode=FA24745CC57592AB612A**](https://www.udemy.com/course/microservices-architecture-and-implementation-on-dotnet/?couponCode=FA24745CC57592AB612A)

See the overall picture of **implementations on microservices with .net tools** on real-world **e-commerce microservices** project;

[](https://user-images.githubusercontent.com/1147445/110304529-c5b70180-800c-11eb-832b-a2751b5bda76.png)

There is a couple of microservices which implemented **e-commerce** modules over **Catalog, Basket, Discount** and **Ordering** microservices with **NoSQL (MongoDB, Redis)** and **Relational databases (PostgreSQL, Sql Server)** with communicating over **RabbitMQ Event Driven Communication** and using **Ocelot API Gateway**.

**Refer the main repository ->**[**https://github.com/aspnetrun/run-aspnetcore-microservices**](https://github.com/aspnetrun/run-aspnetcore-microservices)

**Run The Project**

You will need the following tools:

* [Visual Studio 2019](https://visualstudio.microsoft.com/downloads/)
* [.Net Core 5 or later](https://dotnet.microsoft.com/download/dotnet-core/5)
* [Docker Desktop](https://www.docker.com/products/docker-desktop)

**Installing**

Follow these steps to get your development environment set up: (Before Run Start the Docker Desktop)

1. Clone the repository
2. Once Docker for Windows is installed, go to the **Settings > Advanced option**, from the Docker icon in the system tray, to configure the minimum amount of memory and CPU like so:

* **Memory: 4 GB**
* CPU: 2

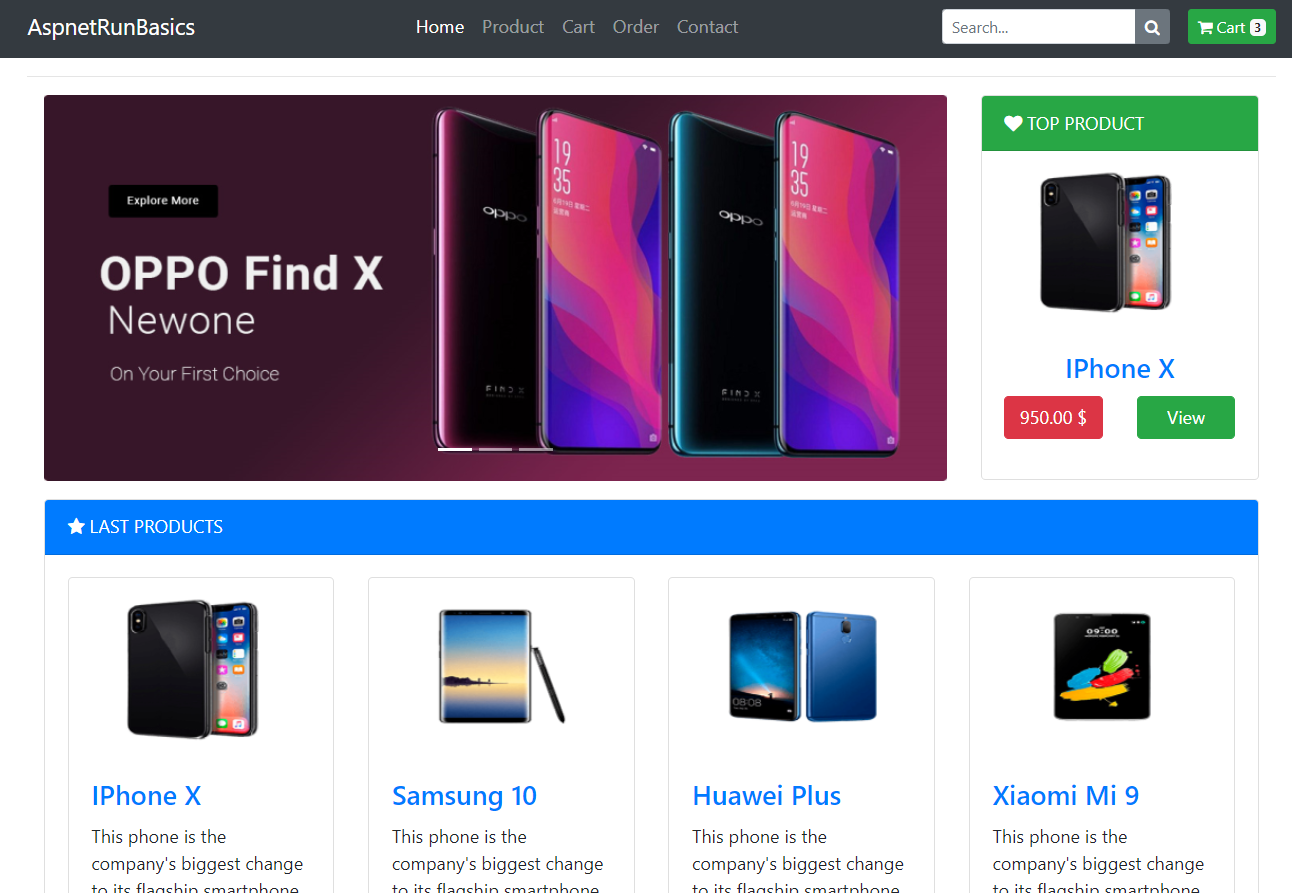
1. At the root directory which include **docker-compose.yml** files, run below command:

docker-compose -f docker-compose.yml -f docker-compose.override.yml up -d

1. Wait for docker compose all microservices. That’s it! (some microservices need extra time to work so please wait if not worked in first shut)
2. You can **launch microservices** as below urls:

* **Catalog API ->**[**http://host.docker.internal:8000/swagger/index.html**](http://host.docker.internal:8000/swagger/index.html)
* **Basket API ->**[**http://host.docker.internal:8001/swagger/index.html**](http://host.docker.internal:8001/swagger/index.html)
* **Discount API ->**[**http://host.docker.internal:8002/swagger/index.html**](http://host.docker.internal:8002/swagger/index.html)
* **Ordering API ->**[**http://host.docker.internal:8004/swagger/index.html**](http://host.docker.internal:8004/swagger/index.html)
* **Shopping.Aggregator ->**[**http://host.docker.internal:8005/swagger/index.html**](http://host.docker.internal:8005/swagger/index.html)
* **API Gateway ->**[**http://host.docker.internal:8010/Catalog**](http://host.docker.internal:8010/Catalog)
* **Rabbit Management Dashboard ->**[**http://host.docker.internal:15672**](http://host.docker.internal:15672/) -- guest/guest
* **Portainer ->**[**http://host.docker.internal:9000**](http://host.docker.internal:9000/) -- admin/admin1234
* **pgAdmin PostgreSQL ->**[**http://host.docker.internal:5050**](http://host.docker.internal:5050/) -- [admin@aspnetrun.com](mailto:admin@aspnetrun.com)/admin1234
* **Elasticsearch ->**[**http://host.docker.internal:9200**](http://host.docker.internal:9200/) -- To Be Develop
* **Kibana ->**[**http://host.docker.internal:5601**](http://host.docker.internal:5601/) -- To Be Develop
* **Web Status ->**[**http://host.docker.internal:8007**](http://host.docker.internal:8007/) -- To Be Develop
* **Web UI ->**[**http://host.docker.internal:8006**](http://host.docker.internal:8006/)

1. Launch [http://host.docker.internal:8007](http://host.docker.internal:8007/) in your browser to view the Web Status. Make sure that every microservices are healthy.
2. Launch [http://host.docker.internal:8006](http://host.docker.internal:8006/) in your browser to view the Web UI. You can use Web project in order to **call microservices over API Gateway**. When you **checkout the basket** you can follow **queue record on RabbitMQ dashboard**.

[](https://user-images.githubusercontent.com/1147445/81381837-08226000-9116-11ea-9489-82645b8dbfc4.png)

Note: If you are running this application in macOS then use docker.for.mac.localhost as DNS name in .env file and the above URLs instead of host.docker.internal.

**Authors**

* **Mehmet Ozkaya** - *Initial work* - [mehmetozkaya](https://github.com/mehmetozkaya)

See also the list of [contributors](https://github.com/aspnetrun/run-core/contributors) who participated in this project. Check also [gihtub page of repository.](https://aspnetrun.github.io/run-aspnetcore-angular-realworld/)