

22306A1012

Sr. No.	Title/Aim of The Practical	Page No.	Date	Signature
1	Installation of .NET SDK & Building Console Application.		24/02/2023	B2
2	Building ASP.NET MVC Application.		21/03/2023	B2
3	Working with Docker hub.		10/04/2023	B2
4	Working with Docker Desktop App.		11/04/2023	B2
5	Building ASP.Net core REST API.		12/04/2023	B2
6	Working with Members API of TeamService.		13/04/2023	B2
7	Running Location Service.		17/04/2023	B2
8	Create a backing Service using ASP .NET.		18/04/2023	B2
9	Building an ASP.NET Core Web Application & Invoking REST APIs from JavaScript.		19/04/2023	B2
10	Working with Docker Volumes and Networks.		20/04/2023	B2.

## Practical 1

Gim : Installation of .Net SDK and building First console application

### Description

The .Net SDK is set of libraries and tools that allow developers to create .Net applications and libraries. It contains following components that are used to build and run application

#### - .Net CLI

The .Net command line interface is cross platform toolchain for developing, building running .NET application

#### - .Net runtime and libraries

The .NET runtime which is installed on a machine for use by framework dependent apps has expensive standard set of class libraries

#### - The dotnet driver

The driver is named dotnet and has two responsibilities, either running a framework dependent app or executing command

### .Net CLI commands

- |                          |          |
|--------------------------|----------|
| - new                    | - sln    |
| - restore                | - help   |
| - <del>build</del> build | - store  |
| - Publish                | - watch  |
| - run                    | - format |
| - test                   |          |
| - vstest                 |          |
| - Pack                   |          |
| - migrate                |          |
| - clear                  |          |

Aim: Building First ASP.NET MVC

Description: The model view controller (MVC) is software design pattern that separates an application data, user interface and control logic into three interconnected components.

- \* The three components of MVC pattern are:-
  - 1] Model: The model represents application data and business logic. It is responsible for managing data, processing it and providing it to other components
  - 2] View: The view is responsible for presenting data to user in user friendly format. It retrieves data from model and presents it to user in way that makes sense
  - 3] Controller: The controller is responsible for handling user input updating model and updating view. It receives inputs from the user via view and update model accordingly.
- \* The MVC pattern provides several benefits:
  - 1] Separation of concerns: The pattern separates application data, user interface, control logic
  - 2] Code reusability: Each component of pattern can be reuse in other application
  - 3] Scalability: The pattern can be scaled to handle large application with complex data structure

Aim: working with Docker hub

Description:

- Give overview of Docker

Docker is open source platform that allows developer to package and deploy application in lightweight portable containers

Docker uses containerization technology to create isolated environments for application and their dependencies allowing easy deployment across different operating systems and cloud platform

One of the key benefits of Docker is portability. Containers can be easily moved between environment such as deployment, testing and production without requiring changes to application code

Overall, Docker simplifies the process of deploying and managing application making it popular choice of modern software development

11/4/23

## Practical no 4

Aim: working with Docker Desktop APP

Description: Give overview of Docker desktop app

### - Container:

A container packages up code and its dependencies so application runs quickly and reliably from one computing environment to another. By default, container is relatively well isolated from other container and its host machine.

### - Image

An image is read only template with instruction creating Docker container. You might create own images or you might only use those creating by others publish in registry.

### - Volumes:

- Volumes are preferred mechanism for persisting data generated by and used by Docker container.
- Volumes are easier to back up or migrate than bind mounts.

Practical No 5

12/14/23

Aim: building ASP .NET core REST API

ASP .NET core is modern, cross-platform framework for building web application including RESTful API. Restful API are popular approach for creating web services that conforms these constraints

ASP .NET core Rest API use HTTP method perform CRUD operation on resources exposed by API

The API can be consumed by clients such as web application, mobile apps or other services. clients sends request to API and receive response in requested format

Overall ASP .NET core REST API are powerful and flexible way to build modern web services that can be consumed by wide range of clients ecosystem

## Practical no 6

Aim: Working with members API of teamservice

The members API of teamService is RESTFUL API that allows users to perform CRUD operations

The API allows users to create new members with an ID, First name and last name

- clients can create new member by sending POST request to the endpoints
- clients can retrieve all members of team by sending GET request to the endpoint
- clients can delete member by sending DELETE request to endpoint

Overall the members API of Teamservice provides a simple and straight forward way for users to manage member of team

## Practical 7

Aim: Running location service

Building the service:

- We also want to maintain and query locations of all our team members
- Upgrade team service to contain locations
- It makes more sense to put responsibility of location management into own service
- This service will manage location history of individuals
- Eg. the public API for location service

REST API for location service

Resource	method	Description
[location/{memberID}/latest]	✓ GET	Retrieves location
[location/{memberID}]	POST	Adds location
[locations/{memberID}]	GET	Retrieves location

## Practical 8

Aim: Create backing service using ASP.NET

Give overview of Event sourcing

- Number of requirements for event-sourced system

1] Ordered:

Event streams are ordered. Performing calculations against same set of events but in different sequence

2] Idempotent

Any function that operates on event stream must always return exact same result for identical ordered event streams

3] Isolated

Any function that produces result based on make use of external information

All data required for calculation must be present in events

## Practical 9

Aim: Building on ASP.NET core web application involving REST API from javascript

The basic steps for invoking REST API from Javascript are

- 1] Determine endpoints URL of REST API you want to call
- 2] choose appropriate HTTP method to use request
- 3] Add any necessary request headers such as authorization or content
- 4] If API requires it add request body with any data that needs to be sent to server.

## Practical 10

Aim: Working with Docker volumes  
and networks

Explain the concept of Docker volumes  
and networks

### - Docker Volumes.

A volume in Docker is way to persist data outside of container file system  
volumes allow you share data between container

### - Docker networks

- A network in Docker is way to connect container together and allow them to communicate with each others
- By default, Docker creates bridge network for each host, which allows containers to communicate with each other

## **Practical No: 1**

**Date: 24/02/2023**

**Aim:** Installation of .NET SDK & Building First Console APP

### **Description:**

Give Overview .NET SDK

### **Code & Output:**

Net SDK Download

Link:

<https://dotnet.microsoft.com/learn/dotnet/hello-world-tutorial/install>

#### 1. Download and install

To start building .NET apps you just need to download and install the .NET SDK (Software Development Kit).

#### 2. Check everything installed correctly

Once you've installed, open a new command prompt and run the following command:

Command prompt> dotnet

```
D:\>dotnet

Usage: dotnet [options]
Usage: dotnet [path-to-application]

Options:
  -h|--help           Display help.
  --info              Display .NET information.
  --list-sdks         Display the installed SDKs.
  --list-runtimes     Display the installed runtimes.

path-to-application:
  The path to an application .dll file to execute.
```

#### 3. Create your app

In your command prompt, run the following commands:

Command prompt

>dotnet new console -o myApp

>cd myApp

The main file in the myApp folder is Program.cs. By default, it already contains the necessary code to write "Hello World!" to the Console.

## Program.cs

```
using System;

namespace MyApp
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}

D:\>dotnet new console -o MyApp
The template "Console App" was created successfully.

Processing post-creation actions...
Restoring D:\myApp\MyApp.csproj:
  Determining projects to restore...
  Restored D:\myApp\MyApp.csproj (in 54 ms).
Restore succeeded.

D:\>cd MyApp
D:\myApp>
```

## 4. Run your app

In your command prompt, run the following command:

Command prompt

```
> dotnet run
```

```
D:\myApp>dotnet run
Hello, World!

D:\myApp>
```

```
>dotnet restore
```

```
>dotnet run
```

```
y D:\myApp>dotnet restore
  Determining projects to restore...
  All projects are up-to-date for restore.

s D:\myApp>dotnet run
  Ninad Karlekar 22306A1012

D:\myApp>
```

## **Print even odd number from 1 to 30**

### **CODE**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int i = 0;
            Console.WriteLine("Print even odd number from 1 to 30");
            Console.WriteLine("\nNinad Karlekar 22306A1012");

            Console.WriteLine("\nEven Numbers :");
            for (i = 1; i <= 30; i++)
            {
                if (i % 2 == 0)
                {
                    Console.Write(i + " ");
                }
            }
            Console.WriteLine("\nOdd Numbers :");
            for (i = 1; i <= 30; i++)
            {
                if (i % 2 != 0)
                {
                    Console.Write(i + " ");
                }
            }
            Console.WriteLine();
        }
    }
}
```

```
D:\myApp>cd ..

D:\>dotnet new console -o myevenodd
The template "Console App" was created successfully.

Processing post-creation actions...
Restoring D:\myevenodd\myevenodd.csproj:
  Determining projects to restore...
  Restored D:\myevenodd\myevenodd.csproj (in 56 ms).
Restore succeeded.
```

```
D:\myevenodd>dotnet restore
  Determining projects to restore...
  All projects are up-to-date for restore.

D:\myevenodd>dotnet run
Print even odd number from 1 to 30

Ninad Karlekar 22306A1012

Even Numbers :
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
Odd Numbers :
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29

D:\myevenodd>
```

**Practical No: 2**

**Date: 21/03/2023**

**Aim: Building First ASP.NET MVC App**

**Description:**

Give Overview of MVC Model

**Code & Output:**

- 1)Install .Net Core Sdk (Link: <https://dotnet.microsoft.com/learn/dotnet/hello-world-tutorial/install>)
- 2)Create folder MyMVC folder in D: drive or any other drive
- 3)Open command prompt and perform following operations

Command: to create mvc project

dotnet new mvc --auth none

```
D:\22306A1012\MyMVC>dotnet new mvc --auth none
The template "ASP.NET Core Web App (Model-View-Controller)" was created
This template contains technologies from parties other than Microsoft
for details.

Processing post-creation actions...
Restoring D:\22306A1012\MyMVC\MyMVC.csproj:
  Determining projects to restore...
    Restored D:\22306A1012\MyMVC\MyMVC.csproj (in 73 ms).
Restore succeeded.
```

- 4)Go to controllers folder and modify HomeController.cs file to match following:

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;
using MyMVC.Models;
namespace MyMVC.Controllers
{
  public class HomeController : Controller
  {
    public String Index()
    { return "Hello World"; }
  }
}
```

- 5) Run the project

```
D:\22306A1012\MyMVC>dotnet run
Building...
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5192
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: D:\22306A1012\MyMVC
warn: Microsoft.AspNetCore.HttpsPolicy.HttpsRedirectionMiddleware
      Failed to determine the https port for redirect.
```

localhost:5192

# Hello World

6) Now go back to command prompt and stop running project using CTRL+C

7) Go to models folder and add new file StockQuote.cs to it with following content

```
using System;
namespace MyMVC.Models
{
    public class StockQuote
    {
        public string Symbol {get;set;}
        public int Price{get;set;}
    }
}
```

8) Now Add View to folder then home folder in it and modify index.cshtml file to match following

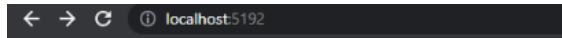
```
@{
    ViewData["Title"] = "Home Page";
}
<div>
    Symbol: @Model.Symbol <br/>
    Price: $@Model.Price <br/>
</div>
```

9) Now modify HomeController.cs file to match following:

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;
using MyMVC.Models;
namespace MyMVC.Controllers
{
    public class HomeController : Controller
    {
        public async Task<IActionResult> Index()
        {
            var model= new StockQuote{ Symbol='HLLO', Price=3200};
            return View(model);
        }
    }
}
```

10)Now run the project using  
dotnet run

11)Now go back to browser and refresh to get modified view response

A screenshot of a web browser window. The address bar at the top shows the URL "localhost:5192". The main content area displays the text "MyMVC" followed by a horizontal line and "Symbol: HLLO" and "Price: \$3200".

# MyMVC

Symbol: HLLO

Price: \$3200

---

## PRACTICE TASK WITH NAME AND ROLL NUMBER

CODE :

1)Install .Net Core Sdk (Link: <https://dotnet.microsoft.com/learn/dotnet/hello-world-tutorial/install>)

2)Create folder MyMVC folder in D: drive or any other drive

3)Open command prompt and perform following operations

Command: to create mvc project

dotnet new mvc --auth none

Output:

```
D:\22306A1012\NinadNew>dotnet new mvc --auth none
The template "ASP.NET Core Web App (Model-View-Controller)" was cr
This template contains technologies from parties other than Micros
ices for details.

Processing post-creation actions...
Restoring D:\22306A1012\NinadNew\NinadNew.csproj:
  Determining projects to restore...
    Restored D:\22306A1012\NinadNew\NinadNew.csproj (in 54 ms).
Restore succeeded.
```

4)Go to controllers folder and modify HomeController.cs file to match following:

5)Go to models folder and add new file StockQuote.cs to it with following content  
using System;

namespace MyMVC.Models

{

```
public class StockQuote
{
    public string Name {get;set;}
    public int RollNo {get;set;}
}
```

6)Now Add View to folder then home folder in it and modify index.cshtml file to match following

```
@{
    ViewData["Title"] = "Home Page";
}
<div>
    Name: @Model.Name <br/>
    RollNo: $@Model.RollNo <br/>
</div>
```

7)Now modify HomeController.cs file to match following:

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;
using MyMVC.Models;
namespace MyMVC.Controllers
{
    public class HomeController : Controller
    {
        public async Task<IActionResult> Index()
        {
            var model= new StockQuote{ Name ="Ninad", RollNo=1012};
            return View(model);
        }
    }
}
```

8)Now run the project using

dotnet run

```
D:\22306A1012\NinadNew>dotnet run
Building...
D:\22306A1012\NinadNew\Models\StockQuote.cs(6,15): warning CS0168: The variable 'model' is assigned but its value is never used
D:\22306A1012\NinadNew\Controllers\HomeController.cs(12,14): warning CS0168: The variable 'model' is assigned but its value is never used
and will run synchronously. Consider using the 'await' operator or 'Task.Run()' to do CPU-bound work on a background thread. [D:\22306A1012\NinadNew\bin\Debug\netcoreapp3.1\HomeController.dll]
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5010
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: D:\22306A1012\NinadNew
```

NinadNew

Name: Ninad Karlekar

RollNo: 1012

## Practical No: 3

Ninad Karlekar 22306A1012

Date: 10/04/2023

### Aim Working with Docker

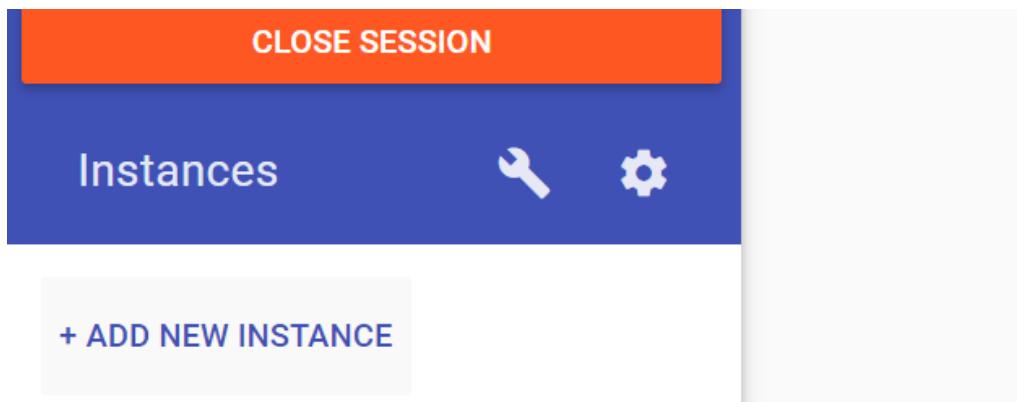
#### Description:

Give Overview of Docker

#### 3 a) Method 1: To pull and push images using docker

#### Code & Output:

1. create Docker Hub account (sign up)
2. login to <https://labs.play-with-docker.com/>
3. add new instance



4. perform following

#### To pull and push images using docker

Command: to check docker version

docker –version

```
[node1] (local) root@192.168.0.8 ~
$ docker --version
Docker version 20.10.17, build 100c701
[node1] (local) root@192.168.0.8 ~
$
```

#### Command: to pull readymade image

## **docker pull rocker/verse**

```
$ docker pull rocker/verse
Using default tag: latest
latest: Pulling from rocker/verse
2ab09b027e7f: Pull complete
1875e7997572: Pull complete
d35813be1883: Pull complete
5132964ecc30: Pull complete
f9ae60b737bd: Pull complete
cf514b297e6d: Pull complete
0dal1c500dd4b: Pull complete
16bff0b45131: Pull complete
719977ec8b94: Pull complete
558a8d6d50e1: Pull complete
Digest: sha256:d95aec1a79700121b86ba4e7031d17561a209ee10ae48f844ec26c84d39496ea
Status: Downloaded newer image for rocker/verse:latest
docker.io/rocker/verse:latest
[nodel1] (local) root@192.168.0.8 ~
$
```

## **Command: to check images in docker**

### **docker images**

```
$ docker images
REPOSITORY      TAG          IMAGE ID      CREATED       SIZE
rocker/verse    latest        7291950d643e   8 days ago   3.41GB
[nodel1] (local) root@192.168.0.8 ~
$
```

## **Now Login to docker hub and create repository**

### **Output:**

Create repository

Namespace  Repository Name\*

Description

Visibility  
Using 0 of 1 private repositories. [Get more](#)

Public Appears in Docker Hub search results  Private Only visible to you

[Cancel](#) [Create](#)

## **Click on Create button**

## **Now check repository created**

---

## ninadstudy / repo1ma

### Description

First repo 

 Last pushed: a few seconds ago

**Command: to login to your docker account**

**docker login –username=ninadstudy**

**password:**

```
$ docker login --username=ninadstudy
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[node1] (local) root@192.168.0.28 ~
```

**Command : to tag image**

**docker tag 8c3e4e2c3e kbdocker11/repo1:firsttry**

```
[node1] (local) root@192.168.0.28 ~
$ docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
rocker/verse    latest       7291950d643e   8 days ago   3.41GB
[node1] (local) root@192.168.0.28 ~
$ docker tag 7291950d643e ninadstudy/repo1ma
[node1] (local) root@192.168.0.28 ~
$ docker tag 7291950d643e ninadstudy/repo1ma:firsttry
[node1] (local) root@192.168.0.28 ~
$
```

**Command: to push image to docker hub account**

**docker push kbdocker11/repo1:firsttry**

```

$ docker tag 7291950d643e ninadstudy/repolma:firsttry
[node1] (local) root@192.168.0.28 ~
$ docker push ninadstudy/repolma:firsttry
The push refers to repository [docker.io/ninadstudy/repolma]
9d1382211021: Mounted from rocker/verse
e4144a267522: Mounted from rocker/verse
7ee25b0c682a: Mounted from rocker/verse
elececf7a947: Mounted from rocker/verse
829d4370cc55: Mounted from rocker/verse
a295f65c9f46: Mounted from rocker/verse
14d748851378: Mounted from rocker/verse
96f921c0089d: Mounted from rocker/verse
98a6c148ed82: Mounted from rocker/verse
b93c1bd012ab: Mounted from rocker/verse
firsttry: digest: sha256:3e11e7dc9187d49a9fe36a349af8901a9901123783c14d049b31417386f72b4c size: 2428
[node1] (local) root@192.168.0.28 ~
$ █

```

## Check it in docker hub now

 ninadstudy / **repo1ma**

### Description

First repo 

 Last pushed: a minute ago

### Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
 firsttry		Image	---	a minute ago

## Click on tags and check



**ninadstudy/repo1ma:firsttry**

DIGEST: sha256 :3e11e7dc9187d49a9fe36a349af8901a9901123783c14d049b31417386f72b4c

OS/ARCH	COMPRESSED SIZE ⓘ	LAST PUSHED	TYPE
linux/amd64	1.24 GB	2 minutes ago by <a href="#">ninadstudy</a>	Image

[Image Layers](#)

[Vulnerabilities](#)

### **3 b) Method 2:Build an image then push it to docker and run it**

#### **Code & Output:**

**Command : to create docker file**

1. cat > Dockerfile <<EOF
2. FROM busybox
3. CMD echo "Hello Ninad."
4. EOF

```
[node1] (local) root@192.168.0.1
$ cat > Dockerfile <<EOF
> FROM busybox
> CMD echo "Hello NInad"
> EOF
[node1] (local) root@192.168.0.1
```

**Command : to build image from docker file**

**dokcer build -t ninadstudy/repo2**

```
[node1] (local) root@192.168.0.1
$ docker build -t ninadstudy/repo2 .
Sending build context to Docker daemon 11.26kB
Step 1/2 : FROM busybox
latest: Pulling from library/busybox
4b35f584bb4f: Pull complete
Digest: sha256:b5d6fe0712636ceb7430189de28819e195e8
Status: Downloaded newer image for busybox:latest
--> 7cfbbec8963d
Step 2/2 : CMD echo "Hello NInad"
--> Running in 1256e014e1bb
Removing intermediate container 1256e014e1bb
--> ddbceeac4d1f
Successfully built ddbceeac4d1f
Successfully tagged ninadstudy/repo2:latest
```

**Command: to check docker images**

**docker images**

```
[node1] (local) root@192.168.0.18 ~
$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
ninadstudy/repo2  latest   ddbceeac4d1f  9 seconds ago  4.86MB
busybox          latest   7cfbbec8963d  3 weeks ago   4.86MB
[node1] (local) root@192.168.0.18 ~
```

**Command: to push image to docker hub**

**docker push ninadstudy/repo2.**

```
Push an image or a repository to a registry
[node1] (local) root@192.168.0.18 ~
$ docker push ninadstudy/repo2
Using default tag: latest
The push refers to repository [docker.io/ninadstudy/repo2]
baacf561cff: Preparing
denied: requested access to the resource is denied
[node1] (local) root@192.168.0.18 ~
$ docker run ninadstudy/repo2
Hello NInad
[node1] (local) root@192.168.0.18 ~
```

**Now check it on docker hub**

ninadstudy / **repo2**  
Contains: No content | Last pushed: 6 minutes ago

ninadstudy / **repo1ma**  
Contains: Image | Last pushed: 27 minutes ago

**Practical No: 4**

**Date: 11/04/2023**

## **Aim: Working with Docker Desktop App**

### **Description:**

Give Overview of Docker Desktop App

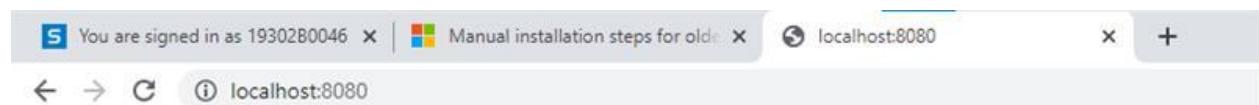
### **Code & Output:**

#### **Commands**

```
$ docker run -p 8080:8080 dotnetcoreservices/hello-world
```

```
PS C:\Users\admin> E:  
PS E:\> cd nk  
PS E:\nk> docker run -p 8080:8080 dotnetcoreservices/hello-world  
Unable to find image 'dotnetcoreservices/hello-world:latest' locally  
latest: Pulling from dotnetcoreservices/hello-world  
693502eb7dfb: Pull complete  
081cd4bfd521: Pull complete  
5d2dc01312f3: Pull complete  
585880aea240: Pull complete  
3905d0c644ea: Pull complete  
c59037c90022: Pull complete  
6eccd279e043: Pull complete  
Digest: sha256:f823a369fe3bb8af8155424c6cb6d95b666de8932ef4ad6651f602b536711d3d  
Status: Downloaded newer image for dotnetcoreservices/hello-world:latest  
Hosting environment: Production  
Content root path: /pipeline/source/app/publish  
Now listening on: http://0.0.0.0:8080  
Application started. Press Ctrl+C to shut down.  
PS E:\nk>
```

Run Localhost in browser



# Hello, world!

```
$ docker ps
```

```
PS C:\Users\admin> docker ps  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  
1e40dc3b3757 dotnetcoreservices/hello-world "/pipeline/source/ap..." 8 minutes ago Up 7 minutes 0.0.0.0:8080->8080/tcp vigilant_golick  
PS C:\Users\admin>
```

```
curl http://localhost:8080/will/itblend?
```

```
PS C:\Users\admin> docker ps
CONTAINER ID   IMAGE                  COMMAND               CREATED             STATUS              PORTS
1e40dc3b3757   dotnetcoreservices/hello-world   "/pipeline/source/ap..."   8 minutes ago    Up 7 minutes   0.0.0.0:8080
PS C:\Users\admin> curl http://localhost:8080/will/itblend?

StatusCode      : 200
StatusDescription : OK
Content          : {72, 101, 108, 108...}
RawContent       : HTTP/1.1 200 OK
                   Transfer-Encoding: chunked
                   Date: Tue, 11 Apr 2023 05:58:36 GMT
                   Server: Kestrel

                   Hello, world!

Headers          : {[Transfer-Encoding, chunked], [Date, Tue, 11 Apr 2023 05:58:36 GMT], [Server, Kestrel]}
RawContentLength : 14

PS C:\Users\admin> .
```

\$ docker kill PID (process id of application)

```
PS C:\Users\admin> docker kill 1e40dc3b3757
1e40dc3b3757
PS C:\Users\admin> .
```

Process Id terminated

```
PS C:\Users\admin> docker kill 1e40dc3b3757
1e40dc3b3757
PS C:\Users\admin> docker ps
CONTAINER ID   IMAGE      COMMAND   CREATED     STATUS      PORTS      NAMES
PS C:\Users\admin> .
```

```
To get more help with docker, check out our guides at https://docs.docker.com/go/guides/
PS C:\Users\admin> docker -v
Docker version 20.10.24, build 297e128
PS C:\Users\admin> docker image ls
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
dotnetcoreservices/hello-world   latest   8fb6660e238d  6 years ago   883MB
PS C:\Users\admin> docker container ls --all
CONTAINER ID   IMAGE      COMMAND   CREATED     STATUS      PORTS
1e40dc3b3757   dotnetcoreservices/hello-world   "/pipeline/source/ap..."   18 minutes ago   Exited (137) 7 minutes ago
PS C:\Users\admin> .
```

**Aim: Building ASP.Net core REST API**

**Description:**

Give Overview of ASP.Net core REST API

**Code & Output:**

**Create your web API**

1. Open two command prompts

Command prompt 1:

Command:

dotnet new webapi -o Glossary

output:

```
Processing post-creation actions...
Restoring D:\nk\MA\Glossary\Glossary.csproj:
  Determining projects to restore...
  Restored D:\nk\MA\Glossary\Glossary.csproj (in 3.4 sec).
Restore succeeded.
```

**Command:**

cd Glossary

dotnet run

```
D:\nk\MA>cd Glossary

D:\nk\MA\Glossary>dotnet run
Building...
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5160
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: D:\nk\MA\Glossary
warn: Microsoft.AspNetCore.HttpsPolicy.HttpsRedirectionMiddleware[3]
      Failed to determine the https port for redirect.
info: Microsoft.Hosting.Lifetime[0]
      Application is shutting down...
```

## 2. Command Prompt 2: (try running ready made weatherforecast class for testing)

Command:

```
curl --insecure http://localhost:5160/weatherforecast
```

output:

```
C:\Windows\System32>D:  
D:\>cd nk  
D:\nk>cd mA
```

```
D:\nk\MA>curl --insecure http://localhost:5160/weatherforecast  
[{"date": "2023-04-13", "temperatureC": 33, "temperatureF": 91, "summary": "Mild"}, {"date": "2023-04-14", "temperatureC": 23, "temperatureF": 73, "summary": "Freezing"}, {"date": "2023-04-15", "temperatureC": -9, "temperatureF": 16, "summary": "Cool"}, {"date": "2023-04-16", "temperatureC": 3, "temperatureF": 37, "summary": "Hot"}, {"date": "2023-04-17", "temperatureC": -7, "temperatureF": 20, "summary": "Scorching"}]  
D:\nk\MA>curl --insecure http://localhost:5001/api/glossary
```

## 3. Now Change the content:

To get started, remove the WeatherForecast.cs file from the root of the project and the WeatherForecastController.cs file from the Controllers folder.

Add Following two files

1) D:\Glossary\GlossaryItem.cs (type it in notepad and save as all files)

```
//GlossaryItem.cs  
namespace Glossary  
{  
    public class GlossaryItem  
    {  
        public string Term { get; set; }  
        public string Definition { get; set; }  
    }  
}
```

2) D:\Glossary\Controllers\ GlossaryController.cs (type it in notepad and save as all files)

```
//Controllers/GlossaryController.cs  
using System;  
using System.Collections.Generic;  
using Microsoft.AspNetCore.Mvc;  
using System.IO;  
namespace Glossary.Controllers  
{  
    [ApiController]  
    [Route("api/[controller]")]  
    public class GlossaryController: ControllerBase  
    {
```

```

private static List<GlossaryItem> Glossary = new List<GlossaryItem> {
    new GlossaryItem
    {
        Term= "HTML",
        Definition = "Hypertext Markup Language"
    },
    new GlossaryItem
    {
        Term= "MVC",
        Definition = "Model View Controller"
    },
    new GlossaryItem
    {
        Term= "OpenID",
        Definition = "An open standard for authentication"
    }
};

[HttpGet]
public ActionResult<List<GlossaryItem>> Get()
{
    return Ok(Glossary);
}

[HttpGet]
[Route("{term}")]
public ActionResult<GlossaryItem> Get(string term)
{
    var glossaryItem = Glossary.Find(item =>
        item.Term.Equals(term, StringComparison.InvariantCultureIgnoreCase));
    if (glossaryItem == null)
    { return NotFound();
    } else
    {
        return Ok(glossaryItem);
    }
}

[HttpPost]
public ActionResult Post(GlossaryItem glossaryItem)
{
    var existingGlossaryItem = Glossary.Find(item =>
        item.Term.Equals(glossaryItem.Term, StringComparison.InvariantCultureIgnoreCase));
    if (existingGlossaryItem != null)
    {
        return Conflict("Cannot create the term because it already exists.");
    }
    else
    {
        Glossary.Add(glossaryItem);
        var resourceUrl = Path.Combine(Request.Path.ToString(),
            Uri.EscapeUriString(glossaryItem.Term));
        return Created(resourceUrl, glossaryItem);
    }
}

```

```
}
```

```
[HttpPost]
```

```
public ActionResult Put(GlossaryItem glossaryItem)
```

```
{
```

```
var existingGlossaryItem = Glossary.Find(item =>
```

```
item.Term.Equals(glossaryItem.Term, StringComparison.InvariantCultureIgnoreCase));
```

```
if (existingGlossaryItem == null)
```

```
{
```

```
return BadRequest("Cannot update a nont existing term.");
```

```
} else
```

```
{
```

```
existingGlossaryItem.Definition = glossaryItem.Definition;
```

```
return Ok();
```

```
}
```

```
}
```

```
[HttpDelete]
```

```
[Route("{term}")]
```

```
public ActionResult Delete(string term)
```

```
{
```

```
var glossaryItem = Glossary.Find(item =>
```

```
item.Term.Equals(term, StringComparison.InvariantCultureIgnoreCase));
```

```
if (glossaryItem == null)
```

```
{ return NotFound();
```

```
}
```

```
else
```

```
{ Glossary.Remove(glossaryItem);
```

```
return NoContent();
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

3. Now stop running previous dotnet run on command prompt 1 using Ctrl+C. and Run it again for new code.

On Command prompt1:

### Command:

dotnet run

output:

```
D:\nk\MA\Glossary>dotnet run
Building...
D:\nk\MA\Glossary\GlossaryItem.cs(6,23): warning CS8618: Non-nullable p
n exiting constructor. Consider declaring the property as nullable. [D:\n
D:\nk\MA\Glossary\GlossaryItem.cs(7,23): warning CS8618: Non-nullable p
ue when exiting constructor. Consider declaring the property as nullable
D:\nk\MA\Glossary\Controllers\GlossaryController.cs(57,57): warning SYS
te: 'Uri.EscapeUriString can corrupt the Uri string in some cases. Cons
components instead.' [D:\nk\MA\Glossary\Glossary.csproj]
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5160
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: D:\nk\MA\Glossary
warn: Microsoft.AspNetCore.HttpsPolicy.HttpsRedirectionMiddleware[3]
      Failed to determine the https port for redirect.
```

On Command prompt2:

1) Getting a list of items:

Command:

```
curl --insecure https://localhost:5160/api/glossary
```

2) Getting a single item

Command:

```
curl --insecure https://localhost:5001/api/glossary/MVC
```

Output:

```
D:\nk\MA>curl --insecure http://localhost:5160/api/glossary
[{"term": "HTML", "definition": "Hypertext Markup Language"}, {"term": "MVC", "definition": "Model View Controller"}, {"term": "OpenID", "definition": "An open standard for authentication"}]
D:\nk\MA>curl --insecure http://localhost:5160/api/glossary/MVC
{"term": "MVC", "definition": "Model View Controller"}
```

2) Creating an item

Command:

```
curl --insecure -X POST -d "{\"term\": \"MFA\", \"definition\": \"An authentication process.\"}" -H "Content-
```

Type:application/json" https://localhost:5160/api/glossary

Output:

```
D:\nk\MA>curl --insecure -X POST -d "{\"term\": \"MFA\", \"definition\": \"An authentication process.\"}" -H "Content-Type:application/json" http://localhost:5160/api/glossary
{"term":"MFA","definition":"An authentication process."}
D:\nk\MA>curl --insecure -X PUT -d "{\"term\": \"MVC\", \"definition\": \"Modified record of Model View Controller.\"}" -H "Content-Type:application/json" http://localhost:5160/api/glossary
```

#### 4)Update Item

Command:

```
curl --insecure -X PUT -d "{\"term\": \"MVC\", \"definition\": \"Modified record of Model View Controller.\"}" -H "Content-Type:application/json" https://localhost:5001/api/glossary
```

Output:

```
D:\nk\MA>curl --insecure -X POST -d "{\"term\": \"MFA\", \"definition\": \"An authentication process.\"}" -H "Content-Type:application/json" http://localhost:5160/api/glossary
{"term":"MFA","definition":"An authentication process."}
D:\nk\MA>curl --insecure -X PUT -d "{\"term\": \"MVC\", \"definition\": \"Modified record of Model View Controller.\"}" -H "Content-Type:application/json" http://localhost:5160/api/glossary

D:\nk\MA>curl --insecure --request DELETE --url http://localhost:5160/api/glossary/openid
```

#### 5) Delete Item

Command:

```
curl --insecure --request DELETE --url https://localhost:5001/api/glossary/openid
```

Output:

```
D:\nk\MA>curl --insecure http://localhost:5160/api/glossary
[{"term": "HTML", "definition": "Hypertext Markup Language"}, {"term": "MVC", "definition": "Modified record of Model View Controller."}, {"term": "MFA", "definition": "An authentication process."}]
D:\nk\MA>
D:\nk\MA>dotnet --version
7.0.200

D:\nk\MA>
```

## Practical No: 6

Date: 13/04/2023

### Aim: Working with Members API of TeamService

#### Description:

Give Overview of Members API of TeamService

#### Code & Output:

#### Steps:

1. Create new project:

Command :

```
dotnet new webapi -o TeamService
```

output:

```
F:\tmp\MA\P6>dotnet new webapi -o TeamService

Welcome to .NET Core 3.1!
-----
SDK Version: 3.1.409

Telemetry
-----
The .NET Core tools collect usage data in order to help us improve your experience
t-out of telemetry by setting the DOTNET_CLI_TELEMETRY_OPTOUT environment variable.

Read more about .NET Core CLI Tools telemetry: https://aka.ms/dotnet-cli-telemetry

-----
Explore documentation: https://aka.ms/dotnet-docs
Report issues and find source on GitHub: https://github.com/dotnet/core
Find out what's new: https://aka.ms/dotnet-whats-new
Learn about the installed HTTPS developer cert: https://aka.ms/aspnet-core-https
Use 'dotnet --help' to see available commands or visit: https://aka.ms/dotnet-cli-commands
Write your first app: https://aka.ms/first-net-core-app

-----
Getting ready...
The template "ASP.NET Core Web API" was created successfully.

Processing post-creation actions...
Running 'dotnet restore' on TeamService\TeamService.csproj...
```

2. Remove existing weatherforecast files both model and controller files.

3. Add Member.cs to "D:\TeamService\Models" folder and add following code.

```
using System;
namespace TeamService.Models{
    public class Member {
        public Guid ID { get; set; }
        public string FirstName { get; set; }
        public string LastName { get; set; }
```

```

public Member() { }

public Member(Guid id) : this(){
    this.ID = id;
}

public Member(string firstName, string lastName, Guid id) : this(id){
    this.FirstName = firstName;
    this.LastName = lastName;
}

public override string ToString(){
    return this.LastName;
}
}
}

```

**4. Add Team.cs to "D:\TeamService\Models" folder and add following code.**

```

using System;
using System.Collections.Generic;

namespace TeamService.Models
{
    public class Team
    {
        public string Name { get; set; }
        public Guid ID { get; set; }
        public ICollection<Member> Members { get; set; }

        public Team()
        {
            this.Members = new List<Member>();
        }

        public Team(string name) : this()
        {
            this.Name = name;
        }

        public Team(string name, Guid id) : this(name)
        {
            this.ID = id;
        }

        public override string ToString()
        {
            return this.Name;
        }
    }
}

```

**5. Add TeamsController.cs to "D:\TeamService\Controllers" folder and add following code.**

```
using System;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Mvc;
using System.Collections.Generic;
using System.Linq;
using TeamService.Models;
using System.Threading.Tasks;
using TeamService.Persistence;

namespace TeamService
{
    [Route("[controller]")]
    public class TeamsController : Controller
    {
        ITeamRepository repository;

        public TeamsController(ITeamRepository repo)
        {
            repository = repo;
        }

        [HttpGet]
        public virtual IActionResult GetAllTeams()
        {
            return this.Ok(repository.List());
        }

        [HttpGet("{id}")]
        public IActionResult GetTeam(Guid id)
        {
            Team team = repository.Get(id);

            if (team != null)
            {
                return this.Ok(team);
            }
            else {
                return this.NotFound();
            }
        }

        [HttpPost]
        public virtual IActionResult CreateTeam([FromBody]Team newTeam)
        {
            repository.Add(newTeam);
            return this.Created($""/teams/{newTeam.ID}", newTeam);
        }
    }
}
```

```

[HttpPut("{id}")]
public virtual IActionResult UpdateTeam([FromBody]Team team, Guid id)
{
    team.ID = id;

    if(repository.Update(team) == null)
    {
        return this.NotFound();
    }
    else
    {
        return this.Ok(team);
    }
}

[HttpDelete("{id}")]
public virtual IActionResult DeleteTeam(Guid id)
{
    Team team = repository.Delete(id);

    if (team == null)
    {
        return this.NotFound();
    }
    else {
        return this.Ok(team.ID);
    }
}
}

```

6. Add MembersController.cs to "D:\TeamService\Controllers" folder and add following code.

```

using System;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Mvc;
using System.Collections.Generic;
using System.Linq;
using TeamService.Models;
using System.Threading.Tasks;
using TeamService.Persistence;
namespace TeamService
{
    [Route("/teams/{teamId}/[controller]")]
    public class MembersController : Controller
    {
        ITeamRepository repository;

        public MembersController(ITeamRepository repo)
        {
            repository = repo;
        }
    }
}

```

```

        }

[HttpGet]
public virtual IActionResult GetMembers(Guid teamID)
{
    Team team = repository.Get(teamID);
    if (team == null)
    {
        return this.NotFound();
    }
    else
    {
        return this.Ok(team.Members);
    }
}

[HttpGet]
[Route("/teams/{teamId}/[controller]/{memberId}")]
public virtual IActionResult GetMember(Guid teamID, Guid memberId)
{
    Team team = repository.Get(teamID);
    if (team == null)
    {
        return this.NotFound();
    }
    else
    {
        var q = team.Members.Where(m => m.ID == memberId);
        if (q.Count() < 1)
        {
            return this.NotFound();
        }
        else
        {
            return this.Ok(q.First());
        }
    }
}

[HttpPut]
[Route("/teams/{teamId}/[controller]/{memberId}")]
public virtual IActionResult UpdateMember([FromBody]Member updatedMember, Guid teamID, Guid memberId)
{
    Team team = repository.Get(teamID);
    if (team == null)
    {
        return this.NotFound();
    }
    else
    {
        var q = team.Members.Where(m => m.ID == memberId);
        if (q.Count() < 1)
        {
            return this.NotFound();
        }
        else
    }
}

```

```

        {
            team.Members.Remove(q.First());
            team.Members.Add(updatedMember);
            return this.Ok();
        }
    }
}

[HttpPost]
public virtual IActionResult CreateMember([FromBody]Member newMember, Guid teamID)
{
    Team team = repository.Get(teamID);
    if (team == null)
    {
        return this.NotFound();
    }
    else
    {
        team.Members.Add(newMember);
        var teamMember = new { TeamID = team.ID, MemberID = newMember.ID };
        return
this.Created($"/teams/{teamMember.TeamID}/{controller}/{teamMember.MemberID}",
teamMember);
    }
}
[HttpGet]
[Route("/members/{memberId}/team")]
public IActionResult GetTeamForMember(Guid memberId)
{
    var teamId = GetTeamIdForMember(memberId);
    if (teamId != Guid.Empty)
    {
        return this.Ok(new { TeamID = teamId });
    }
    else
    {
        return this.NotFound();
    }
}
private Guid GetTeamIdForMember(Guid memberId)
{
    foreach (var team in repository.List())
    {
        var member = team.Members.FirstOrDefault(m => m.ID == memberId);
        if (member != null)
        {
            return team.ID;
        }
    }
    return Guid.Empty;
}
}

```

## 7. Create folder "D:\TeamService\Persistence":

**8. Add ITeamReposiroty.cs to "D:\TeamService\Persistence" folder and add following code.**

```
using System;
using System.Collections.Generic;
using TeamService.Models;
namespace TeamService.Persistence
{
    public interface ITeamRepository
    {
        IEnumerable<Team> List();
        Team Get(Guid id);
        Team Add(Team team);
        Team Update(Team team);
        Team Delete(Guid id);
    }
}
```

**9. Add MemoryTeamRepository.cs to "D:\TeamService\Persistence" folder and add following code.**

```
using System;
using System.Collections.Generic;
using System.Linq;
using TeamService;
using TeamService.Models;
namespace TeamService.Persistence
{
    public class MemoryTeamRepository : ITeamRepository
    {
        protected static ICollection<Team> teams;

        public MemoryTeamRepository()
        {
            if (teams == null)
            {
                teams = new List<Team>();
            }
        }

        public MemoryTeamRepository(ICollection<Team> teams)
        {
            MemoryTeamRepository.teams = teams;
        }

        public IEnumerable<Team> List()
        {
            return teams;
        }

        public Team Get(Guid id)
        {
            return teams.FirstOrDefault(t => t.ID == id);
        }

        public Team Update(Team t)
        {
            Team team = this.Delete(t.ID);
```

```

        if (team != null)
    {
        team = this.Add(t);
    }
    return team;
}
public Team Add(Team team)
{
    teams.Add(team);
    return team;
}
public Team Delete(Guid id)
{
    var q = teams.Where(t => t.ID == id);
    Team team = null;
    if (q.Count() > 0)
    {
        team = q.First();
        teams.Remove(team);
    }
    return team;
}
}
}

```

**10. Add following line to Startup.cs in public void ConfigureServices(IServiceCollection services) method**

```
services.AddScoped<ITeamRepository, MemoryTeamRepository>();
```

11. Now open two command prompts to run this project
12. On Command prompt 1: (go inside folder teamservice first)

Commands:

```
dotnet run
```

```
F:\tmp\MA\P6\TeamService>dotnet run
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: https://localhost:5001
tinfo: Microsoft.Hosting.Lifetime[0]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
linfo: Microsoft.Hosting.Lifetime[0]
Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
Content root path: F:\tmp\MA\P6\TeamService
```

13. On command prompt 2
  - a. To get all teams

Commands: To get all teams

```
curl --insecure https://localhost:5001/teams
```

## Output

```
F:\tmp\MA\P6>curl --insecure https://localhost:5001/teams
[]
```

- b. To create new team

### Commands

```
curl --insecure -H "Content-Type:application/json" -X POST -d
"{"id":"e52baa63-d511-417e-9e54-7aab04286281", "name":"KC"}"
https://localhost:5001/teams
```

### Output

```
F:\tmp\MA\P6>curl --insecure -H "Content-Type:application/json" -X POST -d
"{"id":"e52baa63-d511-417e-9e54-7aab04286281", "name":"KC"}"
https://localhost:5001/teams
{"name":"KC", "id": "e52baa63-d511-417e-9e54-7aab04286281", "members": []}
```

- c. To create one more new team

### Commands

```
curl --insecure -H "Content-Type:application/json" -X POST -d
"{"id":"e12baa63-d511-417e-9e54-7aab04286281", "name":"MSC
Part1"}"
https://localhost:5001/teams
```

### Output

```
F:\tmp\MA\P6>
F:\tmp\MA\P6>curl --insecure -H "Content-Type:application/json" -X POST -d
"{"id":"e12baa63-d511-417e-9e54-7aab04286281", "name":"MSC
Part1"}"
https://localhost:5001/teams
{"name": "MSC Part1", "id": "e12baa63-d511-417e-9e54-7aab04286281", "membe
rs": []}
F:\tmp\MA\P6>
```

- d. To get all teams

### Commands: To get all teams

```
curl --insecure https://localhost:5001/teams
```

### Output

```
F:\tmp\MA\P6>
F:\tmp\MA\P6>curl --insecure https://localhost:5001/teams
[{"name": "KC", "id": "e52baa63-d511-417e-9e54-7aab04286281", "members": []},
 {"name": "MSC Part1", "id": "e12baa63-d511-417e-9e54-7aab04286281", "mem
bers": []}]
F:\tmp\MA\P6>
```

- e. to get single team with team-id as parameter

### Commands

```
curl --insecure https://localhost:5001/teams/e52baa63-d511-417e-9e54-7aab04286281
```

### Output

```
F:\tmp\MA\P6>
F:\tmp\MA\P6>curl --insecure https://localhost:5001/teams/e52baa63-d511-417e-9e54-7aab04286281
>{"name": "KC", "id": "e52baa63-d511-417e-9e54-7aab04286281", "members": []}
```

- f. to update team details (change name of first team from "KC" to "KC IT DEPT")

**Commands**

```
curl --insecure -H "Content-Type:application/json" -X PUT -d
"{"id": "e52baa63-d511-417e-9e54-7aab04286281", "name": "KC IT DEPT"}" https://localhost:5001/teams/e52baa63-d511-417e-9e54-7aab04286281
```

Output

```
F:\tmp\MA\P6>
F:\tmp\MA\P6>curl --insecure -H "Content-Type:application/json" -X PUT -d
"{"id": "e52baa63-d511-417e-9e54-7aab04286281", "name": "KC IT DEPT"}" https://localhost:5001/teams/e52baa63-d511-417e-9e54-7aab04286281
>{"name": "KC IT DEPT", "id": "e52baa63-d511-417e-9e54-7aab04286281", "members": []}
```

- g. To delete team

**Commands**

```
curl --insecure -H "Content-Type:application/json" -X DELETE
https://localhost:5001/teams/e52baa63-d511-417e-9e54-7aab04286281
```

Output

```
F:\tmp\MA\P6>
F:\tmp\MA\P6>curl --insecure -H "Content-Type:application/json" -X DELETE
https://localhost:5001/teams/e52baa63-d511-417e-9e54-7aab04286281
"e52baa63-d511-417e-9e54-7aab04286281"
F:\tmp\MA\P6>
```

- h. With get all teams now it shows only one team (first one is deleted)

**Commands**

```
curl --insecure https://localhost:5001/teams
```

Output

```
F:\tmp\MA\P6>curl -insecure https://localhost:5001/teams
F:\tmp\MA\P6>curl --insecure https://localhost:5001/teams
[{"name": "MSC Part1", "id": "e12baa63-d511-417e-9e54-7aab04286281", "members": []}]
F:\tmp\MA\P6>
```

## **Practical No: 7**

**Date: 17/04/2023**

**Aim: Running Location Service**

**Description:**

**Give Overview of Location Service**

**7 a) Running Location Service Locally using .Net core**

**Code & Output:**

**We need to open 3 command prompts**

**On command prompt 1 start location service (go inside locationservices folder first)**

**Command :**

```
dotnet run --server.urls "http://*:5001"
F:\tmp\MA-NK\P7\StatlerWaldorfCorp.LocationService\StatlerWaldorfCorp.LocationService>dot
net run --server.urls "http://*:5001"
C:\Program Files\dotnet\sdk\3.1.409\Microsoft.Common.CurrentVersion.targets(2084,5): warn
ing MSB3088: Could not read state file "obj\Debug\netcoreapp3.1\StatlerWaldorfCorp.Locati
onService.csprojAssemblyReference.cache". Attempting to deserialize an empty stream. [F:\
tmp\MA-NK\P7\StatlerWaldorfCorp.LocationService\StatlerWaldorfCorp.LocationService\Statle
rWaldorfCorp.LocationService.csproj]
Hosting environment: Production
Content root path: F:\tmp\MA-NK\P7\StatlerWaldorfCorp.LocationService\StatlerWaldorfCorp.
LocationService\bin\Debug\netcoreapp3.1\
Now listening on: http://*:5001
Application started. Press Ctrl+C to shut down.
info: Microsoft.AspNetCore.Hosting.Internal.WebHost[1]
      Request starting HTTP/1.1 POST http://localhost:5001/locations/63e7acf8-8fae-42ce-9
349-3c8593ac8292 application/json 159
info: Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker[1]
```

**On command prompt 2 start location service (go inside teamservices folder first)**

**Command :**

```
dotnet run
```

```
F:\tmp\MA-NK\P7\StatlerWaldorfCorp.TeamService\StatlerWaldorfCorp.TeamService>dotnet run
info: Startup[0]
      Using http://localhost:5001 for location service URL.
Hosting environment: Production
Content root path: F:\tmp\MA-NK\P7\StatlerWaldorfCorp.TeamService\StatlerWaldorfCorp.TeamService\bin\Debug\netcoreapp3.1
Now listening on: http://localhost:5000
Application started. Press Ctrl+C to shut down.
info: Microsoft.AspNetCore.Hosting.Internal.WebHost[1]
```

## On command prompt 3 run all following commands

### Command to Add new team

#### Commands:

```
curl -H "Content-Type:application/json" -X POST -d "{\"id\":\"e52baa63-d511-417e-9e54-7aab04286281\", \"name\":\"KC\"}" http://localhost:5000/teams
```

```
F:\tmp\MA-NK\P7>curl -H "Content-Type:application/json" -X POST -d "{\"id\":\"e52baa63-d511-417e-9e54-7aab04286281\", \"name\":\"KC\"}" http://localhost:5000/teams
{"name": "KC", "id": "e52baa63-d511-417e-9e54-7aab04286281", "members": []}
```

### Command to add new member to team

```
curl -H "Content-Type:application/json" -X POST -d "{\"id\":\"63e7acf8-8fae-42ce-9349-3c8593ac8292\", \"firstName\":\"Ninad\", \"lastName\":\"Karlekar\"}"
http://localhost:5000/teams/e52baa63-d511-417e-9e54-7aab04286281/members
```

```
F:\tmp\MA-NK\P7>curl -H "Content-Type:application/json" -X POST -d "{\"id\":\"63e7acf8-8fae-42ce-9349-3c8593ac8292\", \"firstName\":\"Ninad\", \"lastName\":\"Karlekar\"}" http://localhost:5000/teams/e52baa63-d511-417e-9e54-7aab04286281/members
{"teamID": "e52baa63-d511-417e-9e54-7aab04286281", "memberID": "63e7acf8-8fae-42ce-9349-3c8593ac8292"}
```

### Command to add new location to member

```
curl -H "Content-Type:application/json" -X POST -d "{\"id\":\"64c3e69f-1580-4b2f-a9ff-2c5f3b8f0e1f\", \"latitude\":12.0, \"longitude\":12.0, \"altitude\":10.0, \"timestamp\":0, \"memberId\": \"63e7acf8-8fae-42ce-9349-3c8593ac8292\"}"
http://localhost:5001/locations/63e7acf8-8fae-42ce-9349-3c8593ac8292
```

```
F:\tmp\MA-NK\P7>curl -H "Content-Type:application/json" -X POST -d "{\"id\":\"64c3e69f-1580-4b2f-a9ff-2c5f3b8f0e1f\", \"latitude\":12.0, \"longitude\":12.0, \"altitude\":10.0, \"timestamp\":0, \"memberId\":\"63e7acf8-8fae-42ce-9349-3c8593ac8292\"}" http://localhost:5001/locations/63e7acf8-8fae-42ce-9349-3c8593ac8292
{"id":"64c3e69f-1580-4b2f-a9ff-2c5f3b8f0e1f", "latitude":12.0, "longitude":12.0, "altitude":10.0, "timestamp":0, "memberID":"63e7acf8-8fae-42ce-9349-3c8593ac8292"}
```

**Command To confirm it is accessible from teams (that is from port 5000) it shows last location**

```
curl http://localhost:5000/teams/e52baa63-d511-417e-9e54-7aab04286281/members/63e7acf8-8fae-42ce-9349-3c8593ac8292
```

```
F:\tmp\MA-NK\P7>curl http://localhost:5000/teams/e52baa63-d511-417e-9e54-7aab04286281/members/63e7acf8-8fae-42ce-9349-3c8593ac8292
{"lastLocation":{"id":"64c3e69f-1580-4b2f-a9ff-2c5f3b8f0e1f", "latitude":12.0, "longitude":12.0, "altitude":10.0, "timestamp":0, "memberID":"63e7acf8-8fae-42ce-9349-3c8593ac8292"}, "id":"63e7acf8-8fae-42ce-9349-3c8593ac8292", "firstName":"Ninad", "lastName":"Karlekar"}
F:\tmp\MA-NK\P7>
```

## 7 b) Running Location Service in Docker

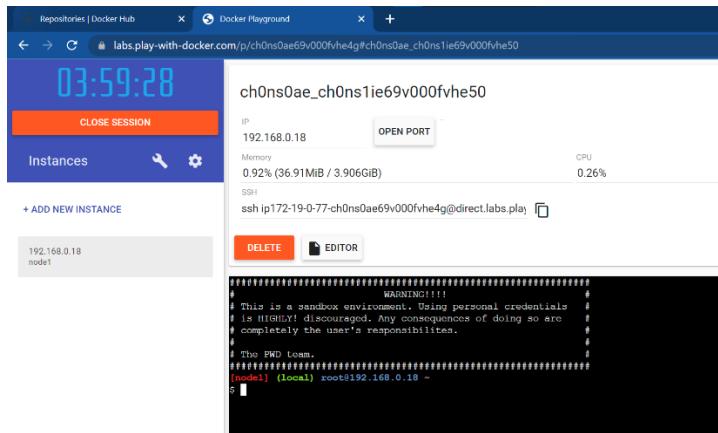
**Code & Output:**

(create docker hub login first to use it in play with docker)

Now login in to Play-With-Docker

Click on Start

Click on Add New Instance



## Start typing following commands

### Command : To run teamservice

```
docker run -d -p 5000:5000 -e PORT=5000 \
-e LOCATION__URL=http://localhost:5001 \
dotnetcoreservices/teamservice:location
```

output: (you can observe that it has started port 5000 on top)

```
[node1] (local) root@192.168.0.18 ~
$ docker run -d -p 5000:5000 -e PORT=5000 \
> -e LOCATION__URL=http://localhost:5001 \
> dotnetcoreservices/teamservice:location
Unable to find image 'dotnetcoreservices/teamservice:location' locally
location: Pulling from dotnetcoreservices/teamservice
693502eb7dfb: Pull complete
081cd4bfd521: Pull complete
5d2dc01312f3: Pull complete
585880aea240: Pull complete
3905d0c644ea: Pull complete
c59037c90022: Pull complete
5a7d450223d5: Pull complete
Digest: sha256:3e9355b72f0ba151d17a2dc9844331a5f590e3afa685d66789458525210346e1
Status: Downloaded newer image for dotnetcoreservices/teamservice:location
ac41cf7ab0989d3218d423e9db163356f7f48efb961e9486806a25e219b98773
[node1] (local) root@192.168.0.18 ~
$
```

### Command: to run location service

```
docker run -d -p 5001:5001 -e PORT=5001 \
dotnetcoreservices/locationservice:nodb
```

output: (now it has started one more port that is 5001 for location service)

```
[node1] (local) root@192.168.0.18 ~
$ docker run -d -p 5001:5001 -e PORT=5001 \
> dotnetcoreservices/locationservice:nodb
Unable to find image 'dotnetcoreservices/locationservice:nodb' locally
nodb: Pulling from dotnetcoreservices/locationservice
693502eb7dfb: Already exists
081cd4bfd521: Already exists
5d2dc01312f3: Already exists
585880aea240: Already exists
3905d0c644ea: Already exists
c59037c90022: Already exists
dbc03883a4ca: Pull complete
Digest: sha256:5f7aca33c5e2117e04f58a59e0cf96fd20d5cbf2cf66c3cd708118d573255168
Status: Downloaded newer image for dotnetcoreservices/locationservice:nodb
307e6560ff8ea80183faab2ef0d1bbabb1349436b99251b0674458743ff416e5
[node1] (local) root@192.168.0.18 ~
$
```

## Command : to check running images in docker

docker images

```
[node1] (local) root@192.168.0.18 ~
$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED     SIZE
dotnetcoreservices/teamservice    location   b27d0de8f2de  6 years ago  886MB
dotnetcoreservices/locationservice nodb      03339f0ea9dd  6 years ago  883MB
[node1] (local) root@192.168.0.18 ~
$
```

## Command: to create new team

```
curl -H "Content-Type:application/json" -X POST -d \
'{"id":"e52baa63-d511-417e-9e54-7aab04286281", "name":"KC"}'
http://localhost:5000/teams
```

```
$
[node1] (local) root@192.168.0.18 ~
$ curl -H "Content-Type:application/json" -X POST -d '{"id":"e52baa63-d511-417e-9e54-7aab04286281", "name":"KC"}' http://localhost:5000/teams
{"name":"KC", "id":"e52baa63-d511-417e-9e54-7aab04286281", "members":[]}
[node1] (local) root@192.168.0.18 ~
$
```

## Command :To confirm that team is added

curl http://localhost:5000/teams/e52baa63-d511-417e-9e54-7aab04286281

Output

```
$
[node1] (local) root@192.168.0.18 ~
$ curl http://localhost:5000/teams/e52baa63-d511-417e-9e54-7aab04286281
{"name":"KC", "id":"e52baa63-d511-417e-9e54-7aab04286281", "members":[]}
[node1] (local) root@192.168.0.18 ~
$
```

## Command : to add new member to teamab

```
curl -H "Content-Type:application/json" -X POST -d \
'{"id":"63e7acf8-8fae-42ce-9349-3c8593ac8292", "firstName":"Ninad",
"lastName":"Karlekar"}'
http://localhost:5000/teams/e52baa63-d511-417e-9e54-7aab04286281/members
```

Output:

```
$
[node1] (local) root@192.168.0.18 ~
$ curl -H "Content-Type: application/json" -X POST -d '{"id":"63e7acf8-8fae-42ce-9349-3c8593ac8292", "firstName":"Ninad",
", "lastName":"Karlekar"}'
http://localhost:5000/teams/e52baa63-d511-417e-9e54-7aab04286281/members
{"teamID":"e52baa63-d511-417e-9e54-7aab04286281", "memberID":"63e7acf8-8fae-42ce-9349-3c8593ac8292"}
[node1] (local) root@192.168.0.18 ~
$
```

## Command :To confirm member added

```
curl http://localhost:5000/teams/e52baa63-d511-417e-9e54-7aab04286281
output:
```

```

$ [node1] (local) root@192.168.0.18 ~
$ curl http://localhost:5000/teams/e52baa63-d511-417e-9e54-7aab04286281
{"name": "KC", "id": "e52baa63-d511-417e-9e54-7aab04286281", "members": [{"id": "63e7acf8-8fae-42ce-9349-3c8593ac8292", "firstName": "Ninad", "lastName": "Karlekar"}]} [node1] (local) root@192.168.0.18 ~
$ [node1] (local) root@192.168.0.18 ~

```

### Command : To add location for member

```

curl -H "Content-Type:application/json" -X POST -d \
'{"id": "64c3e69f-1580-4b2f-a9ff-
2c5f3b8f0e1f", "latitude": 12.0, "longitude": 12.0, "altitude": 10.0,
"timestamp": 0, "memberId": "63e7acf8-8fae-42ce-9349-3c8593ac8292"}'
http://localhost:5001/locations/63e7acf8-8fae-42ce-9349-3c8593ac8292

```

```

[node1] (local) root@192.168.0.18 ~
$ curl -H "Content-Type:application/json" -X POST -d \
> '{"id": "64c3e69f-1580-4b2f-a9ff-2c5f3b8f0e1f", "latitude": 12.0, "longitude": 12.0, "altitude": 10.0,
"timestamp": 0, "memberId": "63e7acf8-8fae-42ce-9349-3c8593ac8292"}' http://localhost:5001/locations/
63e7acf8-8fae-42ce-9349-3c8593ac8292
{"id": "64c3e69f-1580-4b2f-a9ff-2c5f3b8f0e1f", "latitude": 12.0, "longitude": 12.0, "altitude": 10.0, "tim
estamp": 0, "memberID": "63e7acf8-8fae-42ce-9349-3c8593ac8292"} [node1] (local) root@192.168.0.18 ~
$ [node1] (local) root@192.168.0.18 ~

```

### Command : To confirm location is added in member

```
curl http://localhost:5001/locations/63e7acf8-8fae-42ce-9349-3c8593ac8292
```

output:

```

[node1] (local) root@192.168.0.18 ~
$ curl http://localhost:5001/locations/63e7acf8-8fae-42ce-9349-3c8593ac8292
[{"id": "64c3e69f-1580-4b2f-a9ff-2c5f3b8f0e1f", "latitude": 12.0, "longitude": 12.0, "altitude": 10.0, "ti
mestamp": 0, "memberID": "63e7acf8-8fae-42ce-9349-3c8593ac8292"}] [node1] (local) root@192.168.0.18 ~
$ [node1] (local) root@192.168.0.18 ~

```

## **Practical No: 8**

Date: 18/04/2023

### **Aim: Create a backing Service using ASP .NET**

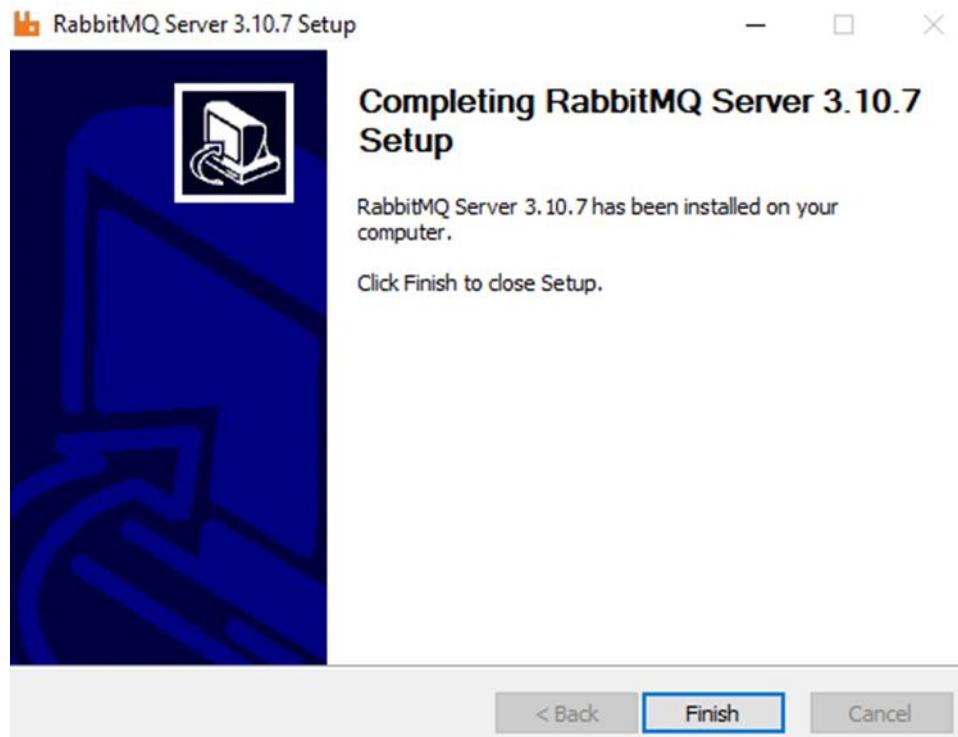
#### **Description:**

#### **Give Overview of Event Sourcing**

#### **Event Sourcing and CQRS Example Code & Output:**

1)install erland broker using OTP AND Radis cache

2)then install rabbitmq server



3)open folder "C:\Users\Admin\AppData\Roaming\RabbitMQ\db" and delete everything inside it.

4)now run RabbitMQ sbin command prompt as administrator

5) and type following commands:

In rabbitmq command prompt

rabbitmq-plugins enable rabbitmq\_management

```

F:\tmp\MA-NK\APK files\rabbit\rabbitmq_server-3.10.7\sbin>rabbitmq-plugins enable rabbitmq_management
Enabling plugins on node rabbit@DESKTOP-M2MC62A:
rabbitmq_management
The following plugins have been configured:
  rabbitmq_management
  rabbitmq_management_agent
  rabbitmq_web_dispatch
Applying plugin configuration to rabbit@DESKTOP-M2MC62A...
The following plugins have been enabled:
  rabbitmq_management
  rabbitmq_management_agent
  rabbitmq_web_dispatch

set 3 plugins.
Offline change; changes will take effect at broker restart.

```

rabbitmq-service remove  
rabitmq-service install  
rabitmq-server restart

```

F:\tmp\MA-NK\APK files\rabbit\rabbitmq_server-3.10.7\sbin>
F:\tmp\MA-NK\APK files\rabbit\rabbitmq_server-3.10.7\sbin>rabbitmq-server restart
2023-04-21 16:29:49.175000+05:30 [warning] <0.130.0> Using RABBITMQ_ADVANCED_CONFIG_FILE: c:/Users/
2023-04-21 16:29:51.442000+05:30 [info] <0.222.0> Feature flags: list of feature flags found:
2023-04-21 16:29:51.442000+05:30 [info] <0.222.0> Feature flags: [x] classic_mirrored_queue_vers
2023-04-21 16:29:51.442000+05:30 [info] <0.222.0> Feature flags: [x] implicit_default_bindings
2023-04-21 16:29:51.442000+05:30 [info] <0.222.0> Feature flags: [x] maintenance_mode_status
2023-04-21 16:29:51.442000+05:30 [info] <0.222.0> Feature flags: [x] quorum_queue
2023-04-21 16:29:51.442000+05:30 [info] <0.222.0> Feature flags: [x] stream_queue
2023-04-21 16:29:51.442000+05:30 [info] <0.222.0> Feature flags: [x] user_limits
2023-04-21 16:29:51.442000+05:30 [info] <0.222.0> Feature flags: [x] virtual_host_metadata
2023-04-21 16:29:51.442000+05:30 [info] <0.222.0> Feature flags: feature flag states written to di
2023-04-21 16:29:51.943000+05:30 [notice] <0.44.0> Application syslog exited with reason: stopped
2023-04-21 16:29:51.943000+05:30 [notice] <0.222.0> Logging: switching to configured handler(s); f

## ##      RabbitMQ 3.10.7
## ##
##### Copyright (c) 2007-2022 VMware, Inc. or its affiliates.
#####
##### Licensed under the MPL 2.0. Website: https://rabbitmq.com

Erlang:    25.0.4 [jit]
TLS Library: OpenSSL - OpenSSL 1.1.1d  10 Sep 2019

Doc guides:  https://rabbitmq.com/documentation.html
Support:    https://rabbitmq.com/contact.html
Tutorials:   https://rabbitmq.com/getstarted.html
Monitoring:  https://rabbitmq.com/monitoring.html

Logs: <stdout>
      c:/Users/User/AppData/Roaming/RabbitMQ/log/rabbit@DESKTOP-M2MC62A.log
      c:/Users/User/AppData/Roaming/RabbitMQ/log/rabbit@DESKTOP-M2MC62A_upgrade.log

Config file(s): c:/Users/User/AppData/Roaming/RabbitMQ/advanced.config

Starting broker... completed with 3 plugins.

```

- 6)Now open web browser (crome) and type following URL <http://localhost:15672>
- 7) It will show login page: here type username as “guest” and password as “guest” and login.
- 7)Now open 5 command prompts

**8) On command prompt 1 start team service for that go inside teamservice project folder and write command :**

dotnet run --server.urls=http://0.0.0.0:5001 output:

```
F:\tmp\MA-NK\P8\StatlerWaldorfCorp.Tea...
F:\tmp\MA-NK\P8\StatlerWaldorfCorp.Tea...
F:\tmp\MA-NK\P8\StatlerWaldorfCorp.Tea...
dotnet run --server.urls=http://0.0.0.0:5001
info: Startup[0]
      Using http://localhost:5001 for location service URL.
Hosting environment: Production
Content root path: F:\tmp\MA-NK\P8\StatlerWaldorfCorp.Tea...
Now listening on: http://0.0.0.0:5001
Application started. Press Ctrl+C to shut down.
```

**9) On command prompt 2 start LocationReports service for that go inside LocationReporter project folder and write command :**

dotnet run --server.urls=http://0.0.0.0:5002 output:

```
F:\tmp\MA-NK\P8\StatlerWaldorfCorp.LocationRe...
info: StatlerWaldorfCorp.LocationReporter.Servic...
      Team Service HTTP client using URL http://localhost:5001
info: StatlerWaldorfCorp.LocationReporter.Events.AMQPEventEmitter[0]
      AMQP Event Emitter configured with URI amqp://localhost:5672/
Hosting environment: Production
Content root path: F:\tmp\MA-NK\P8\StatlerWaldorfCorp.LocationRe...
Now listening on: http://0.0.0.0:5002
Application started. Press Ctrl+C to shut down.
```

**10) On command prompt 3 start EventProcessor service for that go inside EventProcessor project folder and write command :**

dotnet run --server.urls=http://0.0.0.0:5003

```
F:\tmp\MA-NK\P8\StatlerWaldorfCorp.EventProces...
info: StatlerWaldorfCorp.EventProcessor.Queues.AMQP.AMQPConnectionFactory[0]
      AMQP Connection configured for URI : amqp://localhost:5672/
info: StatlerWaldorfCorp.EventProcessor.Queues.AMQP.AMQPEventSubscriber[0]
      Initialized event subscriber for queue memberlocationrecorded
info: StatlerWaldorfCorp.EventProcessor.Queues.AMQP.AMQPConnectionFactory[0]
      AMQP Connection configured for URI : amqp://localhost:5672/
info: StatlerWaldorfCorp.EventProcessor.Queues.AMQP.AMQPEventEmitter[0]
      Emitting events on queue proximitydetected
info: StatlerWaldorfCorp.EventProcessor.Location.Redis.RedisLocationCache[0]
      Using redis location cache - 127.0.0.1:6379
info: StatlerWaldorfCorp.EventProcessor.Queues.AMQP.AMQPEventSubscriber[0]
      Subscribed to queue.
Hosting environment: Production
Content root path: F:\tmp\MA-NK\P8\StatlerWaldorfCorp.EventProcessor
Now listening on: http://0.0.0.0:5003
Application started. Press Ctrl+C to shut down.
```

**11) On command prompt 4 start Reality service for that go inside Reality project folder and write command :**

dotnet run --server.urls=http://0.0.0.0:5004

**output:**

```
F:\tmp\MA-NK\P8\StatlerWaldorfCorp.RealityService>dotnet run --server.urls=http://0.0.0.0:5004
Hosting environment: Production
Content root path: F:\tmp\MA-NK\P8\StatlerWaldorfCorp.RealityService
Now listening on: http://0.0.0.0:5004
Application started. Press Ctrl+C to shut down.
```

## 12) Now run all commands on command prompt 5

### Command: to add new team

```
curl -H "Content-Type:application/json" -X POST -d "{\"id\":\"e52baa63-d511-417e-9e54-7aab04286281\", \"name\":\"KC\"}" http://localhost:5001/teams
```

### output:

```
F:\tmp\MA-NK\P8>curl -H "Content-Type: application/json" -X POST -d "{\"id\":\"e52baa63-d511-417e-9e54-7aab04286281\", \"name\":\"KC\"}" http://localhost:5001/teams
{"name": "KC", "id": "e52baa63-d511-417e-9e54-7aab04286281", "members": []}
F:\tmp\MA-NK\P8>
```

### Command : to add new member

```
curl -H "Content-Type:application/json" -X POST -d "{\"id\":\"63e7acf8-8fae-42ce-9349-3c8593ac8292\", \"firstName\":\"Kirti\", \"lastName\":\"Bhatt\"}"
http://localhost:5001/teams/e52baa63-d511-417e-9e54-7aab04286281/members
```

### output:

```
F:\tmp\MA-NK\P8>curl -H "Content-Type:application/json" -X POST -d "{\"id\":\"63e7acf8-8fae-42ce-9349-3c8593ac8292\", \"firstName\":\"Ninad\", \"lastName\":\"Karlekar\"}"
http://localhost:5001/teams/e52baa63-d511-417e-9e54-7aab04286281/members
{"teamID": "e52baa63-d511-417e-9e54-7aab04286281", "memberID": "63e7acf8-8fae-42ce-9349-3c8593ac8292"}
F:\tmp\MA-NK\P8>
```

### Command: to add location of member

```
curl -H "Content-Type:application/json" -X POST -d "{\"reportid\":\"67c3e69f-1580-4b2f-a9ff-2c5f3b8f0e1f\",
\"origin\":\"1234\", \"latitude\":12.5, \"longitude\":12.5, \"memberId\":\"63e7acf8-8fae-42ce-9349-3c8593ac8292\"}"
http://localhost:5002/api/members/63e7acf8-8fae-42ce-9349-3c8593ac8292/locationreports
```

### output:

```
F:\tmp\MA-NK\P8>
F:\tmp\MA-NK\P8>curl -H "Content-Type:application/json" -X POST -d "{\"reportid\":\"67c3e69f-1580-4b2f-a9ff-2c5f3b8f0e1f\",\"origin\":\"1234\",\"latitude\":12.5,\"longitude\":12.5,\"memberID\":\"63e7acf8-8fae-42ce-9349-3c8593ac8292\"}" http://localhost:5002/api/members/63e7acf8-8fae-42ce-9349-3c8593ac8292/locationreports
{"reportID":"67c3e69f-1580-4b2f-a9ff-2c5f3b8f0e1f","origin":"1234","latitude":12.5,"longitude":12.5,"memberID":"63e7acf8-8fae-42ce-9349-3c8593ac8292"}
F:\tmp\MA-NK\P8>
```

## Output of event processor shows event received

```
F:\tmp\MA-NK\P8>StatlerWaldorfCorp.EventProcessor>dotnet run --server.urls=http://0.0.0.0:5003
info: StatlerWaldorfCorp.EventProcessor.Queues.AMQP.AMQPConnectionFactory[0]
      AMQP Connection configured for URI : amqp://localhost:5672/
info: StatlerWaldorfCorp.EventProcessor.Queues.AMQP.AMQPEventSubscriber[0]
      Initialized event subscriber for queue memberlocationrecorded
info: StatlerWaldorfCorp.EventProcessor.Queues.AMQP.AMQPConnectionFactory[0]
      AMQP Connection configured for URI : amqp://localhost:5672/
info: StatlerWaldorfCorp.EventProcessor.Queues.AMQP.AMQPEventEmitter[0]
      Emitting events on queue proximitydetected
info: StatlerWaldorfCorp.EventProcessor.Location.Redis.RedisLocationCache[0]
      Using redis location cache - 127.0.0.1:6379
info: StatlerWaldorfCorp.EventProcessor.Queues.AMQP.AMQPEventSubscriber[0]
      Subscribed to queue.
Hosting environment: Production
Content root path: F:\tmp\MA-NK\P8\StatlerWaldorfCorp.EventProcessor
Now listening on: http://0.0.0.0:5003
Application started. Press Ctrl+C to shut down.
info: StatlerWaldorfCorp.EventProcessor.Queues.AMQP.AMQPEventSubscriber[0]
      Received incoming event, 232 bytes.
```

## Command : To add another member to same Team

```
curl -H "Content-Type:application/json" -X POST -d "{\"id\":\"68e7acf8-8fae-42ce-9349-3c8593ac8292\", \"firstName\":\"Anushka\", \"lastName\":\"Bhatt\"}"
http://localhost:5001/teams/e52baa63-d511-417e-9e54-7aab04286281/members
F:\tmp\MA-NK\P8>
F:\tmp\MA-NK\P8>curl -H "Content-Type:application/json" -X POST -d "{\"id\":\"68e7acf8-8fae-42ce-9349-3c8593ac8292\", \"firstName\":\"Anushka\", \"lastName\":\"Bhatt\"}"
http://localhost:5001/teams/e52baa63-d511-417e-9e54-7aab04286281/members
{"teamID": "e52baa63-d511-417e-9e54-7aab04286281", "memberID": "68e7acf8-8fae-42ce-9349-3c8593ac8292"}
F:\tmp\MA-NK\P8>
```

## Command : To add location of this another member of same team

```
curl -H "Content-Type:application/json" -X POST -d "{\"reportid\":\"70c3e69f-1580-4b2f-a9ff-2c5f3b8f0e1f\",
\"origin\":\"1234\", \"latitude\":12.7, \"longitude\":12.8, \"memberID\":\"68e7acf8-8fae-42ce-9349-3c8593ac8292\"}"
http://localhost:5002/api/members/68e7acf8-8fae-42ce-9349-3c8593ac8292/locationreports
```

```
F:\tmp\MA-NK\P8>
F:\tmp\MA-NK\P8>curl -H "Content-Type:application/json" -X POST -d "{\"reportid\":\"70c3e69f-1580-4b2f-a9ff-2c5f3b8f0e1f\", \"origin\":\"1234\", \"latitude\":12.7, \"longitude\":12.8, \"memberID\":\"68e7acf8-8fae-42ce-9349-3c8593ac8292\"}"
http://localhost:5002/api/members/68e7acf8-8fae-42ce-9349-3c8593ac8292/locationreports
{"reportID":"70c3e69f-1580-4b2f-a9ff-2c5f3b8f0e1f","origin":"1234","latitude":12.7,"longitude":12.8,"memberID":"68e7acf8-8fae-42ce-9349-3c8593ac8292"}
F:\tmp\MA-NK\P8>
```

RabbitMQ™ RabbitMQ 3.10.7 Erlang 25.0.4

Refreshed 2023-04-21 16:51:39 Refresh every 5 seconds

Virtual host: All Cluster rabbit@DESKTOP-M2MC62A User guest Log out

Overview Connections Channels Exchanges Queues Admin

### Queues

All queues (1)

Pagination: Page 1 of 1 - Filter:   Regex ? Displaying 1 item, page size up to:  100

Overview		Messages			Message rates			+/-
Name	Type	Features	State	Ready	Unacked	Total	Incoming	deliver / get ack
memberlocationrecorded	classic		Idle	0	0	0	0.00/s	0.00/s 0.00/s

Add a new queue

HTTP API Server Docs Tutorials Community Support Community Slack Commercial Support Plugins GitHub Changelog

RabbitMQ™ 3.8.3 Erlang 22.3

Overview Connections Channels Exchanges Queues Admin

Publish message

Get messages

Warning: getting messages from a queue is a destructive action. ?

Ack Mode:

Encoding:

Messages:

**Get Message(s)**

Message 1

The server reported 0 messages remaining.

Exchange	(AMQP default)
Routing Key	proximitydetected
Redelivered	0
Properties	
Payload	{"SourceMemberID": "63e7acf8-8fae-42ce-9349-3c8593ac8292", "TargetMemberID": "68e7acf8-8f}
333 bytes	
Encoding:	string

**Aim: Building an ASP.NET Core Web Application & Invoking REST APIs from JavaScript****Description:****Explain Invoking REST APIs from JavaScript****Code & Output:****Controller Folder****ApiController.cs**

```
using Microsoft.AspNetCore.Mvc;
using StatlerWaldorfCorp.WebApp.Models;
namespace StatlerWaldorfCorp.WebApp.Controllers
{
    [Route("api/test")]
    public class ApiController : Controller
    {
        [HttpGet]
        public IActionResult GetTest()
        {
            return this.Ok(new StockQuote { Symbol = "API", Price = 9999 });
        }
    }
}
```

**HomeController.cs**

```
using Microsoft.AspNetCore.Mvc;
using System.Threading.Tasks;
using StatlerWaldorfCorp.WebApp.Models;
namespace StatlerWaldorfCorp.WebApp.Controllers
{
    public class HomeController : Controller
    {
        public IActionResult Index()
        {
            var model = new StockQuote { Symbol = "HLLO", Price = 3200 };

            return View(model);
        }
    }
}
```

**Models Folder****StockQuote.cs**

```
namespace StatlerWaldorfCorp.WebApp.Models
{
```

```

public class StockQuote
{
    public string Symbol { get; set; }
    public int Price { get; set; }
}

```

## Views Folder

### Index.cshtml

```

<html>
<head>
    <title>Hello world</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
    <script src="/Scripts/hello.js"></script>
</head>
<body>
    <h1>Hello World</h1>
    <div>
        <h2>Stock Quote</h2>
        <div>
            Symbol: @Model.Symbol<br/>
            Price: $@Model.Price<br/>
        </div>
        </div>
        <br/>
        <div>
            <p class="quote-symbol">The Symbol is </p>
            <p class="quote-price">The price is $</p>
        </div>
    </body>
</html>

```

## Folder wwwroot\Scripts

### hello.js

```

$(document).ready(function () {
    $.ajax({
        url: "/api/test"
    }).then(function (data) {
        $('.quote-symbol').append(data.symbol);
        $('.quote-price').append(data.price);
    });
});

```

## Program.cs

```

using System;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.Builder;
using Microsoft.Extensions.Configuration;
using System.IO;
namespace StatlerWaldorfCorp.WebApp
{
    public class Program

```

```

    {
        public static void Main(string[] args)
        {
            var config = new ConfigurationBuilder()
                .AddCommandLine(args)
                .Build();

            var host = new WebHostBuilder()
                .UseContentRoot(Directory.GetCurrentDirectory())
                .UseKestrel()
                .UseStartup<Startup>()
                .UseConfiguration(config)
                .Build();
            host.Run();
        }
    }
}

```

## Startup.cs

```

using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.Extensions.Logging;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Configuration;
namespace StatlerWaldorfCorp.WebApp
{
    public class Startup
    {
        public Startup(IHostingEnvironment env)
        {
            var builder = new ConfigurationBuilder()
                .SetBasePath(env.ContentRootPath)
                .AddEnvironmentVariables();
            Configuration = builder.Build();
        }

        public IConfiguration Configuration { get; set; }

        public void ConfigureServices(IServiceCollection services)
        {
            services.AddMvc();
        }

        public void Configure(IApplicationBuilder app, IWebHostEnvironment env, ILoggerFactory loggerFactory)
        {
            //loggerFactory.AddConsole();
            //loggerFactory.AddDebug();
            app.UseDeveloperExceptionPage();
            app.UseRouting();
            app.UseEndpoints(endpoints =>
            {
                endpoints.MapControllerRoute(
                    name: "default",
                    pattern: "{controller=Home}/{action=Index}/{id?}");
            });
            app.UseStaticFiles();
        }
    }
}

```

}

## StatlerWaldorfCorp.WebApp.csproj

```
<Project Sdk="Microsoft.NET.Sdk.Web">
  <PropertyGroup>
    <TargetFramework>netcoreapp5.0</TargetFramework>
  </PropertyGroup>

  <ItemGroup>
    <PackageReference Include="Microsoft.AspNetCore" Version="1.1.1" />
    <PackageReference Include="Microsoft.AspNetCore.Mvc" Version="1.1.2" />
    <PackageReference Include="Microsoft.AspNetCore.StaticFiles" Version="1.1.1" />
    <PackageReference Include="Microsoft.Extensions.Logging.Debug" Version="1.1.1" />
    <PackageReference Include="Microsoft.VisualStudio.Web.BrowserLink" Version="1.1.0" />
    <PackageReference Include="Microsoft.Extensions.Configuration" Version="1.1.1"/>
    <PackageReference Include="Microsoft.Extensions.Options.ConfigurationExtensions"
Version="1.1.1"/>
    <PackageReference Include="Microsoft.Extensions.Configuration.Json" Version="1.1.1"/>
    <PackageReference Include="Microsoft.Extensions.Configuration.CommandLine"
Version="1.1.1"/>
  </ItemGroup>
</Project>
```

```
Configuring...
-----
1 command is running to initially populate your local package
it'll take up to a minute to complete and will only happen once
Decompressing 100% 11258 ms
Expanding 100% 36250 ms
Hosting environment: Production
Content root path: F:\tmp\MA-NK\P9\StatlerWaldorfCorp.WebApp
Now listening on: http://localhost:5000
Application started. Press Ctrl+C to shut down.
Info: Microsoft.AspNetCore.Hosting.InternalWebHost[1]
```

```
← → C ⌂ ⓘ localhost:5000
```

# Hello World

## Stock Quote

Symbol: HLLO

Price: \$3200

The Symbol is API

The price is \$9999

## Practical No: 10 Ninad Karlekar 22306A1012

Date: 20/04/2023

### Aim: Working with Docker Volumes and Networks

#### Description:

#### Explain the concept of Docker Volumes and Networks

#### Code & Output:

Perform Following Inside Play-With-Docker

- 1) Pull nginx image into docker

#### Command:

```
docker pull nginx
```

#### Output:

```
[node1] (local) root@192.168.0.13 ~
$ docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
26c5c85e47da: Pull complete
4f3256bdf66b: Pull complete
2019c71d5655: Pull complete
8c767bdbcb9ae: Pull complete
78e14bb05fd3: Pull complete
75576236abf5: Pull complete
Digest: sha256:63b44e8ddb83d5dd8020327c1f40436e37a6ffffd3ef2498a6204df23be6e7e94
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
[node1] (local) root@192.168.0.13 ~
```

- 2) Now run the pulled image in Container named “webApp”

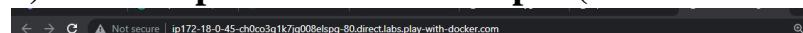
#### Command:

```
docker run -it --name=webApp -d -p 80:80 nginx
```

#### Output:

```
docker.io/library/nginx:latest
[node1] (local) root@192.168.0.13 ~
$ docker run -it --name=webApp -d -p 80:80 nginx
e6172b3412cdd484462bcde1016d688da559ef46d603662c5cf1dab50500bd62
[node1] (local) root@192.168.0.13 ~
$ █
```

- 3) Click on port 80 to check output (it shows welcome page)



## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

**4) We make changes into “index.html” file inside /usr/share/nginx/html folder**

**Commands:**

docker exec -it webApp bash //this command to execute bash shell

cd /usr/share/nginx/html // to go inside html folder

echo "Hello NK" > index.html // to change content of index.html file

**output:**

```
[node1] (local) root@192.168.0.13 ~
$ docker exec -it webApp bash
root@e6172b3412cd:/# cd /usr/share/nginx/html
root@e6172b3412cd:/usr/share/nginx/html# echo "Hello NK" > index.html
root@e6172b3412cd:/usr/share/nginx/html#
root@e6172b3412cd:/usr/share/nginx/html# exit
exit
[node1] (local) root@192.168.0.13 ~
```

**5) Type exit to return to docker prompt and check process status using ps option**

**Commands:**

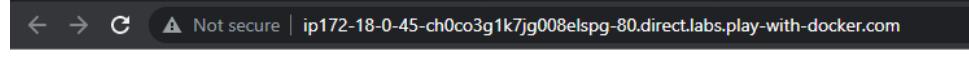
Exit

Docker ps

**Output:**

```
[node1] (local) root@192.168.0.13 ~
$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e6172b3412cd nginx "/docker-entrypoint...." 8 minutes ago Up 8 minutes 0.0.0.0:80->80/tcp webApp
[node1] (local) root@192.168.0.13 ~
$
```

**6) Now refresh on port 80 output (you should get modified output)**



Not secure | ip172-18-0-45-ch0co3g1k7jg008elsgp-80.direct.labs.play-with-docker.com

# Hello NK

**7) Now stop running container named “webApp”.**

**Command:**

docker stop webApp

**output:**

```
e6172b3412cd nginx "/docker-e
[node1] (local) root@192.168.0.13 ~
$ docker stop webApp
webApp
[node1] (local) root@192.168.0.13 ~
```

**8) Start nginx in new container named as “webApp1”.**

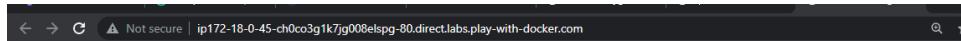
**Command:**

docker run -it --name=webApp1 -d -p 80:80 nginx

**output:**

```
[node1] (local) root@192.168.0.13 ~
$ docker run -it --name=webApp1 -d -p 80:80 nginx
a479ca15921cefadf6c5b9b84d1310e538a84466926adb2e352ec86d734b25b7
[node1] (local) root@192.168.0.13 ~
$
```

**9) Now Click on port 80 (you will see the welcome page again)**



## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

**10) To solve this issue we create new volume.**

**Command:**

docker volume create MyVolume

**output:**

```
#####
[node2] (local) root@192.168.0.12 ~
$ docker volume create MyVolume
MyVolume
[node2] (local) root@192.168.0.12 ~
$ docker volume ls
```

**11) Check volume is created**

**Command:**

docker volume ls

**output:**

```
MyVolume
[node2] (local) root@192.168.0.12 ~
$ docker volume ls
DRIVER      VOLUME NAME
local       MyVolume
[node2] (local) root@192.168.0.12 ~
```

**12) Check details of volume**

**Command:**

docker volume inspect MyVolume

**output:**

```
[node2] (local) root@192.168.0.12 ~
$ docker volume inspect MyVolume
[
    {
        "CreatedAt": "2023-04-20T05:59:25Z",
        "Driver": "local",
        "Labels": {},
        "Mountpoint": "/var/lib/docker/volumes/MyVolume/_data",
        "Name": "MyVolume",
        "Options": {},
        "Scope": "local"
    }
]
[node2] (local) root@192.168.0.12 ~
```

### 13) Mount this volume to nginx new container named “webApp4”

**Command:**

```
docker run -d --name=webApp4 --mount
source=MyVolume,destination=/usr/share/nginx/html -p 80:80
nginx
```

**output:**

```
[node2] (local) root@192.168.0.12 ~
$ docker run -d --name=webApp4 --mount source=MyVolume,destination=/usr/share/nginx/html -p 80:80 nginx
Unable to find image "nginx:latest" locally
latest: Pulling from library/nginx
26c5c85e47da: Full complete
4f3256df66b: Full complete
2019c71d5655: Full complete
8c767bdbe9ae: Full complete
7be14bb05fd3: Full complete
75576236abf5: Full complete
Digest: sha256:63b44e8db83d5dd8020327clf40436e37a6ffffd3ef2498a6204df23be6e7e94
Status: Downloaded newer image for nginx:latest
d991588d6db39e2aa2c790e0eb6b927921504d672a69435e72e105ff5692c5a5
[node2] (local) root@192.168.0.12 ~
$ ls /
```

### 14) Now keep on doing “ls” and “cd “ to go inside \_data folder of our volume “MyVolume”

**Commands:**

```
ls /
cd /var/lib/docker
ls
```

**output:**

```
[node2] (local) root@192.168.0.12 ~
$ ls /
bin      docker.log  lib      mnt      root      srv      usr
certs   etc        lib64    opt      run       sys      var
dev     home       media    proc     sbin      tmp
[node2] (local) root@192.168.0.12 ~
$ cd /var/lib/docker
[node2] (local) root@192.168.0.12 /var/lib/docker
$ ls
buildkit  containers  network   plugins   swarm      trust
containerd image      overlay2  runtimes  tmp       volumes
[node2] (local) root@192.168.0.12 /var/lib/docker
```

**Commands:**

```
cd volumes
ls
cd MyVolume
ls
```

**output:**

```
[node2] (local) root@192.168.0.12 /var/lib/docker
$ cd volumes
[node2] (local) root@192.168.0.12 /var/lib/docker/volumes
$ ls
MyVolume          backingFsBlockDev metadata.db
[node2] (local) root@192.168.0.12 /var/lib/docker/volumes
$ cd MyVolume
[node2] (local) root@192.168.0.12 /var/lib/docker/volumes/MyVolume
$ ls
data
[node2] (local) root@192.168.0.12 /var/lib/docker/volumes/MyVolume
```

### Commands:

cd \_data

ls

### Output:

```
[node2] (local) root@192.168.0.12 /var/lib/docker/volumes/MyVolume
$ cd _data
[node2] (local) root@192.168.0.12 /var/lib/docker/volumes/MyVolume/_data
$ ls
50x.html      index.html
[node2] (local) root@192.168.0.12 /var/lib/docker/volumes/MyVolume/_data
$ echo "From MyVolume Ninad 22306A1012" > index.html
[node2] (local) root@192.168.0.12 /var/lib/docker/volumes/MyVolume/_data
$
```

### 15) Modify contents of index.html file with “from MyVolume hello KB”

#### Command:

echo “from MyVolume Ninad 22306A1012” > index.html

#### Output:

```
[node2] (local) root@192.168.0.12 /var/lib/docker/volumes/MyVolume/_data
$ echo "From MyVolume Ninad 22306A1012" > index.html
[node2] (local) root@192.168.0.12 /var/lib/docker/volumes/MyVolume/_data
$
```

### 16) Now refresh port 80 (to get modified output)

From MyVolume Ninad 22306A1012

### 17) Now stop this running container named “webApp4”

#### Command:

docker stop webApp4

#### Output:

```
[node2] (local) root@192.168.0.12 /var/lib/docker/volumes/MyVolume/_data
$ docker stop webApp12
webApp12
[node2] (local) root@192.168.0.12 /var/lib/docker/volumes/MyVolume/_data
$ docker run -d --name=webApp4
```

## 18) Now run nginx in new container named “webApp6”

**Command:**

```
docker run -d --name=webApp6 --mount  
source=MyVolume,destination=/usr/share/nginx/html -p 80:80 nginx
```

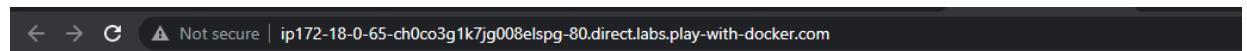
**output:**

```
[node2] (local) root@192.168.0.12 /var/lib/docker/volumes/MyVolume/ data  
$ docker run -d --name=webApp13 --mount source=MyVolume,destination=/usr/share/nginx/html -p 80:80 nginx  
c75b9993670e731270af739ecb516541776366ea9170242d6030e439863c64f9  
[node2] (local) root@192.168.0.12 /var/lib/docker/volumes/MyVolume/ data  
$ █
```

## 19) Click on port 80 and refresh the page you should get edited file as output.

We can load the page again localhost:80 and still see the html file that we edited in the volume. So, with the help of volumes, we can easily access the data even we stop the container and it's very easy to access data and import the data to anywhere.

**Output:**



From MyVolume Ninad 22306A1012