## Step 1

## Web API source code

Source Repo : https://github.com/Dandekar-Ashish/PostCodeAPI

Instruction to create Deployable

- Open **PostCodeAPI.sln** file in Visual Studio.
- Build Project. It will download all dependant NuGet Packages. After successful build you can see deployable files in
- **~PostCodeAPI\bin\Debug\netcoreapp3.1\** Folder.
- Create Zip of all files. E.g. WebAPIPackage.zip
- This zip should contain direct files not folder.

## Step 2

## Web Application code

Source Repo https://github.com/Dandekar-Ashish/postcode-cli-app

Instruction to install mode modules and run application. We will Create production Deployment package once REST API service is live in Step 3. Because we need API URL to set in Config.json

- Open postcode-cli-app folder in VsCode
- Open Terminal
- Run Command npm i
- Use npm run start to run application

- Follow Below Steps after Step 3
  - Open Config.json
  - Replace PostCodeAPIUrl with your API URL
  - **{https://Your URL}**/Prod/api/postcode
  - You must append /Prod/api/postcode after your URL
  - Use npm run build to create production deployable package.
  - It will create deployable files in **build folder** in project root directly.

## Step 3

## AWS Deployment Steps

- Login to AWS Console
- Create new bucket for RestAPIService e.g **<PostCodeServiceBucket>**
- Upload Zip file which is created in Web API source code Deployment Step 1
- Upload PostCodeAWSFormation.yml in bucket
- Get URL (Object URL) of PostCodeAWSFormation.yml file from properties.
- Open CloudShell

Change ParameterValues in parameters.json

| Key | Value |
|---|---|
| **ReactArtifactBucketName** | Set Bucket to be created for React Application |
| **CoreArtifactBucketName** | Set bucket Created for RestAPIService in last step |
| **ZipFileNameForCoreAPIArtifact** | Set Zip File Name which is Created in Step 1. |

Run Following command (Keep parameters.json in root folder or give complete path for these files)

Change template Url of PostCodeAWSFormation.yml

```
aws cloudformation  create-stack --stack-name PostCodeStack --template-url "http://amazopn.path/PostCodeAWSFormation.yml" --parameters "file://parameters.json" --capabilities CAPABILITY_IAM
```

Another way to run formation without input file.  Just Change ParameterValues in the command , same as JSON file

```
aws cloudformation  create-stack --stack-name PostCodeStack --template-url "https://postcodeapibuckettest.s3.amazonaws.com/PostCodeAWSFormation.yml" --parameters ParameterKey=ReactArtifactBucketName,ParameterValue=reactbucketpostcodetest ParameterKey=CoreArtifactBucketName,ParameterValue=postcodeapibuckettest ParameterKey=ZipFileNameForCoreAPIArtifact,ParameterValue=PostCodeArtifacts.zip --capabilities CAPABILITY_IAM
```

- Open PostCodeStack Stack in AWS and check progress
- Once stack created Successfully Click on Stack Name and Go to Output Tab
- You will Get API URL
- Browse That URL . You will get Message in browser

  **Welcome to running ASP.NET Core on AWS Lambda**

- Now Let's deploy React Application.
- Now Complete the remaining steps of **Step 2.**
- Go to the S3Buckets, Refresh AWS Window.
- You can see New Bucket is created with publicly Access rights. (You have provided name for bucket in Json file => ReactArtifactBucketName)
- Open That bucket and Upload artifacts (all files and folder) of React App. (To create artifacts, follow remaining steps of Step 2.)
- Once upload complete Go to the Properties of that Bucket and scroll down to "Static website hosting" section.
- You will get **Bucket website endpoint**. Browse that URL.
- **Your React Application is Live.**

**Application is already Deployed in amazon**
**Application URI :** http://reactclipostcode.s3-website-us-east-1.amazonaws.com/

**Thank You.**