Ways and Means
Technology Pvt. Ltd.
CMMI DEV / 3
Exp. 2026-26-07 / Appraisal #0722

DISCOVER
Dedicated to discovery, committed to care
CREATIVE
Bringing imagination to reality
TRANSFROM
Transforming businesses across the globe
INSPIRE
We build to inspire
BRAINSTORM
Generating diverse ideas

SFA APPLICATION

Deployment Notes V1.0

Contents

1 Deployment Notes

Deployment notes are a set of instructions that provide detailed guidance on how to deploy an application to a specific environment or platform. These notes typically include steps to install and configure the required software and components, as well as any custom configurations or settings that are needed for the application to function properly.

2 Deployment Prerequisites

Before deploying the application, ensure that the following prerequisites are met:

1. Windows Server with IIS7.5 and above is installed.
2. Windows MS SQL Server 2016 or above is installed.
3. The Angular application is published and ready for deployment.
4. The API application folder is published and ready for deployment.
5. The web application folder is published and ready for deployment.
6. The website URLs for the API and web application are available.

Additionally, please ensure that the following tasks are completed prior to deployment:

1. Install any required SSL certificates on the server to ensure secure communication between the server and the client.
2. Configure firewall settings to allow traffic to and from the server on the required ports.
3. Configure DNS settings to point to the server's IP address.
4. Configure the database with the appropriate settings, such as authentication mode, users, and permissions.
5. Set any environment-specific configurations required by the application, such as API endpoints, database connection strings, and authentication settings.
6. Have a backup and disaster recovery plan in place to ensure data and application availability in the event of a system failure or outage.

3 SFA Web Application Deployment

Steps are given below:

1. Publish SFA Web Application using Visual Code IDE:

Publishing an Angular 14.0 code involves compiling the application into a set of static files that can be deployed to a web server. These static files include HTML, CSS, and JavaScript files, as well as any other assets that the application uses, such as images or fonts. The output of the publishing process is a folder containing all the necessary files to run the Angular application on a web server.

To publish an Angular 14.0 code, follow these steps:

i. Open the command prompt or terminal and navigate to the root folder of the Angular application.

ii. Run the command "ng build --prod" to build the application for production. This will compile the application and create the necessary static files for deployment.

iii. After the build process is complete, a "dist" folder will be created in the root directory of the application. This folder contains all the necessary files to run the application on a web server.

iv. Copy the contents of the "dist" folder to the physical path specified in the deployment notes document for the web application on the IIS server.

The published folder will contain the following files and folders:

- index.html: This is the entry point for the application and is the file that is served when a user accesses the website.
- main.js: This is the main JavaScript file that contains the compiled code for the application.
- styles.css: This file contains the compiled CSS styles for the application.
- assets/: This folder contains any assets used by the application, such as images or fonts.
- favicon.ico: This is the icon file that appears in the browser tab for the website.
- runtime.js: This file contains the runtime code for the application and is required for the application to run.
- polyfills.js: This file contains polyfills for features not supported by all browsers.
- vendor.js: This file contains code for third-party libraries and frameworks used by the application.

By publishing the Angular 14.0 code and copying the contents of the "dist" folder to the physical path specified in the deployment notes document, the application is ready to be deployed to the IIS server and made available for users to access.

2. Install IIS Server on the windows server:

    a. IIS (Internet Information Services) is a web server software that allows hosting of web applications on Windows machines.

    b. To install IIS Server on a Windows machine, follow these steps:

        i. Open the Control Panel and click on "Programs and Features".

        ii. Click on "Turn Windows features on or off".

        iii. Scroll down to find "Internet Information Services" and expand it.

        iv. Check the box next to "Web Management Tools" and "World Wide Web Services".

        v. Click "OK" and wait for the installation to complete.

3. Ensure that rewrite URL module is installed on the IIS:
    a. The rewrite URL module is required for configuring URL rewrite rules in IIS.
    b. If it is not installed on the IIS, you can install it directly by following these steps:
        i. Visit the following URL - https://www.iis.net/downloads/microsoft/url-rewrite
        ii. Click on the "Download" button and download the appropriate version for your system.
        iii. Run the installer and follow the prompts to complete the installation.
4. Create a website on IIS with the following values:
    a. Site Name: sfa.paperlessinstore.com
    b. ApplicationPool: A new application pool
    c. Physical Path: Give the physical path of the published files of Angular Project
    d. Host Name: sfa.paperlessinstore.com
    e. Leave the rest of the fields values default.
5. Ensure that the application pool is dedicated to this application with default settings.
    a. In IIS, click on the "Application Pools" node in the left-hand pane.
    b. Right-click on the application pool dedicated to the application and select "Advanced Settings".
    c. In the "Advanced Settings" window, ensure that the default settings are applied.
6. Go to the physical location of the web application published folder - ./assets/appconfig.production
    a. Edit the file "appconfig.production.json".
    b. Give the value to key "remoteServiceBaseUrl" as the URL to API - "https://sfaapi.paperlessinstore.com"
    c. Give the value to key "appBaseUrl" as the URL to the Web Application "https://sfa.paperlessinstore.com".
7. Create a web.config file and place in the root folder of the published web application code:

i. Open a text editor such as Notepad or Visual Studio Code.

ii. Create a new file and save it as "web.config" in the root directory of your Angular application.

iii. Copy and paste the following XML code into the web.config file:

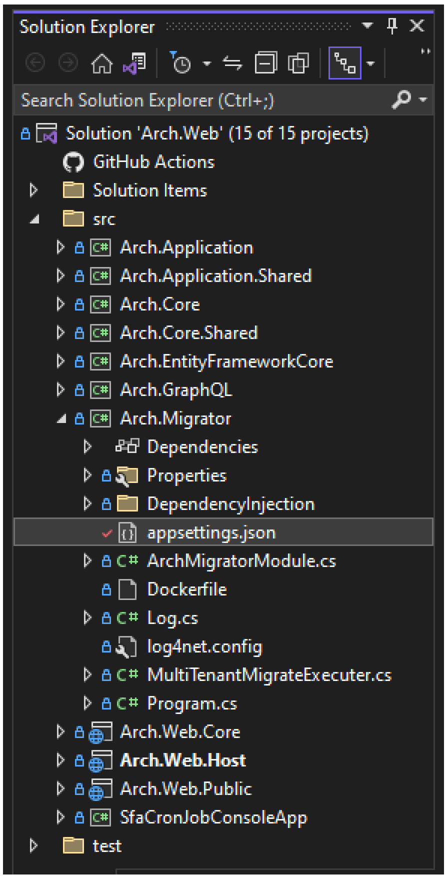
```
<?xml version="1.0" encoding="utf-8"?>
<configuration>

<system.webServer>
  <rewrite>
    <rules>
      <rule name="Angular Routes" stopProcessing="true">
        <match url=".*" />
        <conditions logicalGrouping="MatchAll">
          <add input="{REQUEST_FILENAME}" matchType="IsFile" negate="true" />
          <add input="{REQUEST_FILENAME}" matchType="IsDirectory" negate="true" />
        </conditions>
        <action type="Rewrite" url="./index.html" />
      </rule>
    </rules>
  </rewrite>
</system.webServer>

</configuration>
```

Note: Just create backup of the web application at the time of deployment, regular backups are not required unless any changes are done to the source code and redeployment is done. No programmatically generated files are stored in web application.

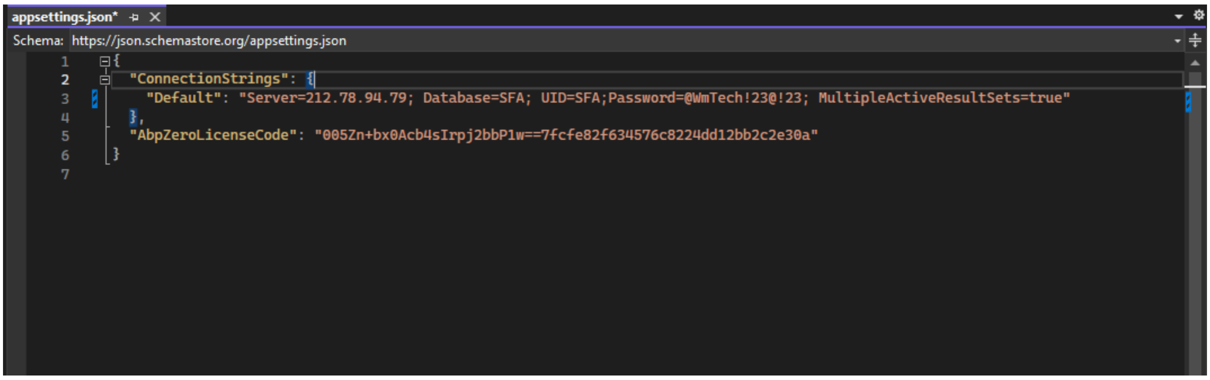
4 Web Application Server Deployment Notes

Publishing the code:

1. Open the web application solution in Visual Studio 2022.
2. Go to Arch.Migrator folder as shown in the image below:

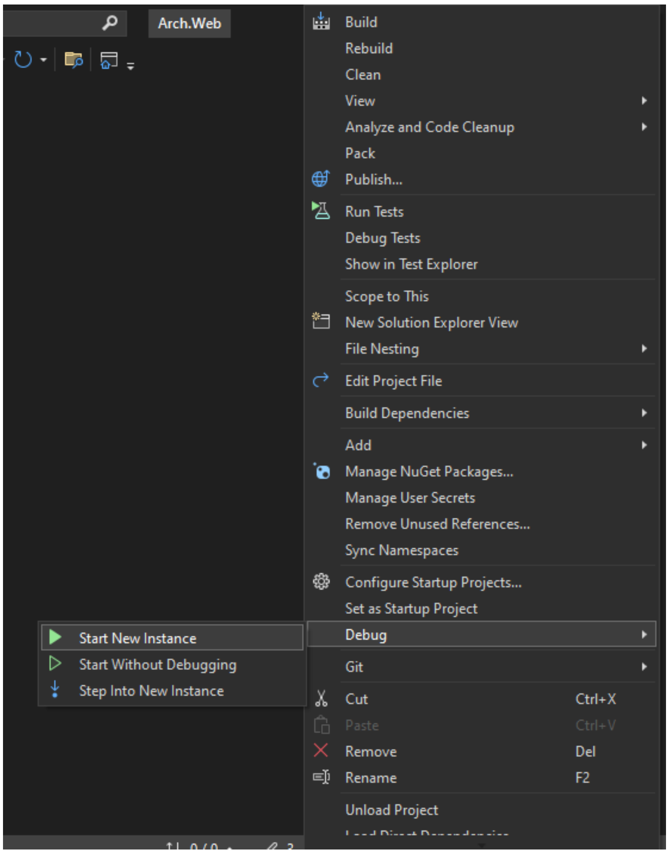


3. Update Connection string in the database setting in the appsettings.json file as shown in image

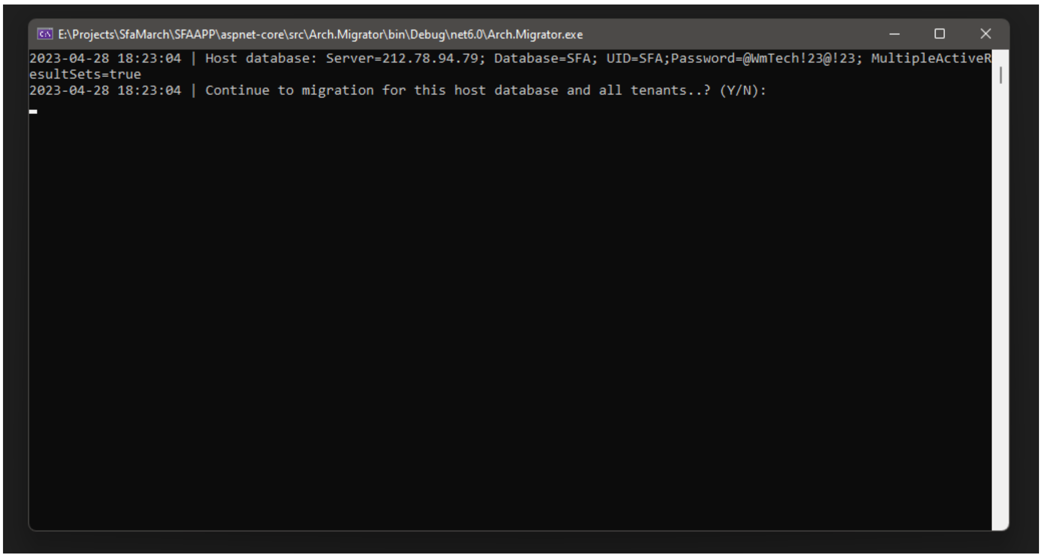


```
{
  "ConnectionStrings": {
    "Default": "Server=212.78.94.79; Database=SFA; UID=SFA;Password=@WmTech!23@!23; MultipleActiveResultSets=true"
  },
  "AbpZeroLicenseCode": "005Zn+bx0Acb4sIrpj2bbP1w==7fcfe82f634576c8224dd12bb2c2e30a"
}
```

4. Then right click on arch.migrator to generate the options shown in image below to start new instance

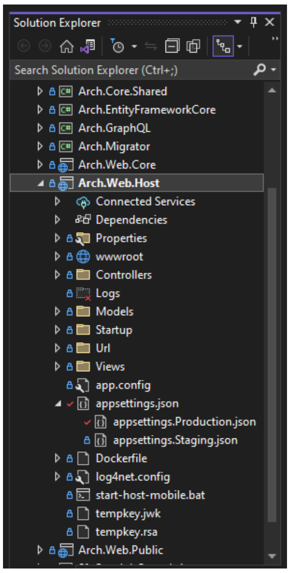


5. In Migrator window shown in the image below give “Y” as answer:

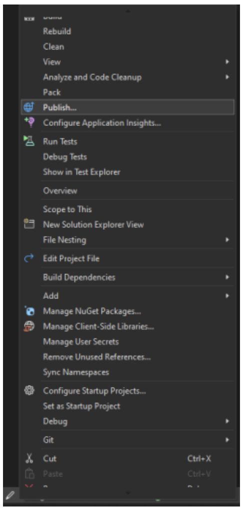

E:\Projects\SfaMarch\SFAAPP\aspnet-core\src\Arch.Migrator\bin\Debug\net6.0\Arch.Migrator.exe

```
2023-04-28 18:23:04 | Host database: Server=212.78.94.79; Database=SFA; UID=SFA;Password=@WmTech!23@!23; MultipleActiveR
esultSets=true
2023-04-28 18:23:04 | Continue to migration for this host database and all tenants..? (Y/N):
```

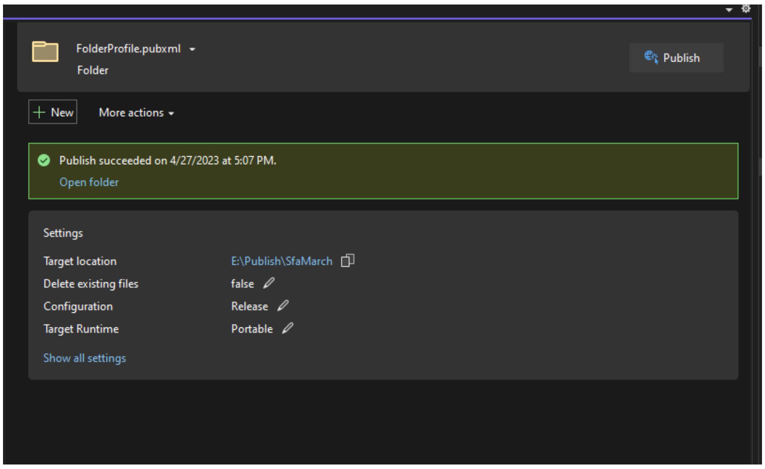
6. Now go to Arch.Web.Host and then open appsettings.json and appsettings.production.json and update the connection string.



7. Now right click on arch.web.host and publish



8. Go to New as shown in the image below:

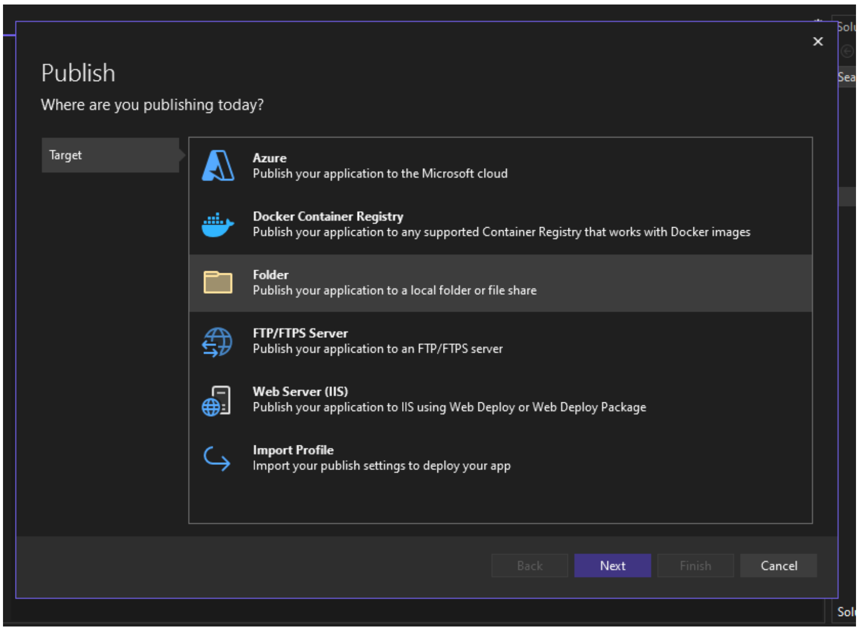


Publish succeeded on 4/27/2023 at 5:07 PM.

Settings

Target location E:\Publish\SfaMarch
Delete existing files false
Configuration Release
Target Runtime Portable

9. Preferred location for published files is to be selected.



Publish

Where are you publishing today?

Target

Azure
Publish your application to the Microsoft cloud

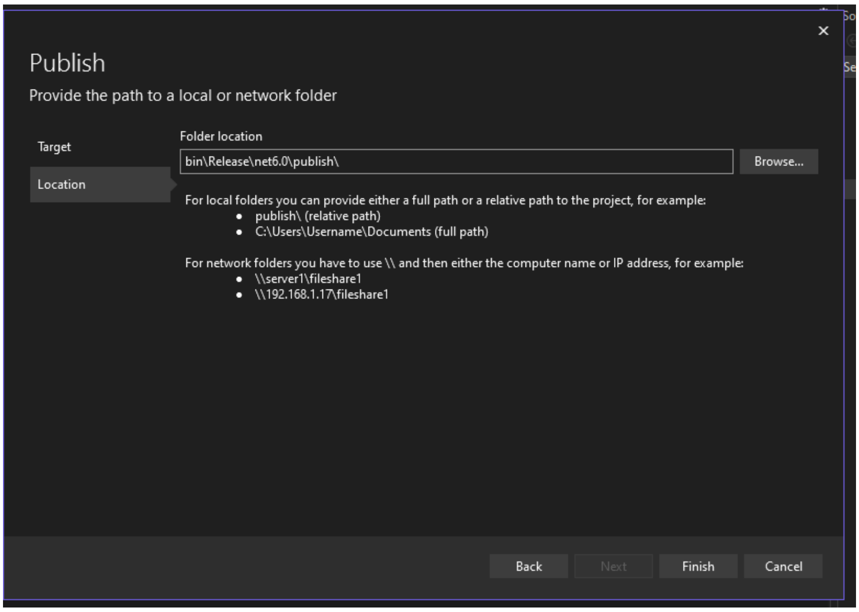
Docker Container Registry
Publish your application to any supported Container Registry that works with Docker images

Folder
Publish your application to a local folder or file share

FTP/FTPS Server
Publish your application to an FTP/FTPS server

Web Server (IIS)
Publish your application to IIS using Web Deploy or Web Deploy Package

Import Profile
Import your publish settings to deploy your app



Publish

Provide the path to a local or network folder

Target

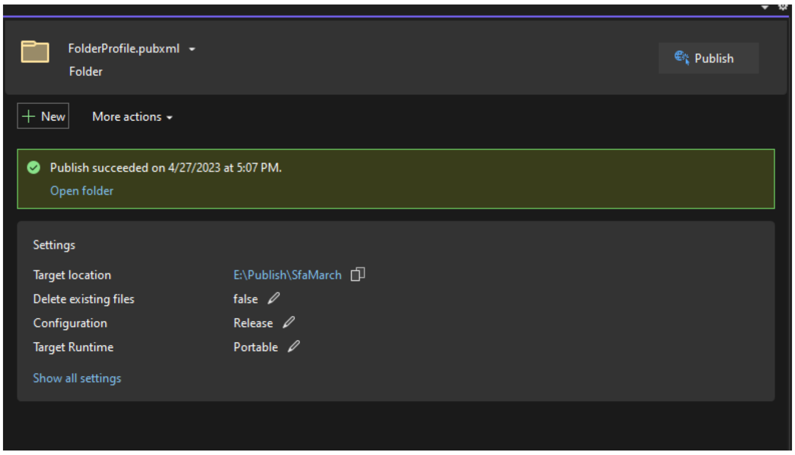
Location

Folder location: bin\Release\net6.0\publish\

For local folders you can provide either a full path or a relative path to the project, for example:
- publish\ (relative path)
- C:\Users\Username\Documents (full path)

For network folders you have to use \\ and then either the computer name or IP address, for example:
- \\server1\fileshare1
- \\192.168.1.17\fileshare1

10. Click on publish as shown in the image below, this will save the published files to the selected location in the process:



FolderProfile.pubxml
Folder

Publish succeeded on 4/27/2023 at 5:07 PM.
Open folder

Settings

Target location E:\Publish\SfaMarch
Delete existing files false
Configuration Release
Target Runtime Portable

After you have the published folder from the web application with database migrator executed at the time of publishing the folder.

The publish folder is ready to be deployed on the serve.

5 API Deployment

Following steps are to be followed for deploying the web application on the server:

1. Copy the published folder to the server.
2. Create IIS Website in same way as was done for the Web Application.
3. Copy the published folder on the physical path of the IIS website created.
4. Go to appsettings.json and appsettings.production.json to update the value of the key "ClientRootAddress" and provide it value https://sfaapi.paperlessinstore.com/.
5. Give IIS_USR read, write and execute access rights to wwwroot folder.
6. Application is deployed.

6 Cron Job Deployment

There are two cron jobs in the application. Both are console based application and follow standard deployment procedure for console based application in Visual Studio 2022. Before publishing both the cron jobs in the appsettings.json file in interface folder update the database credentials to the database server.

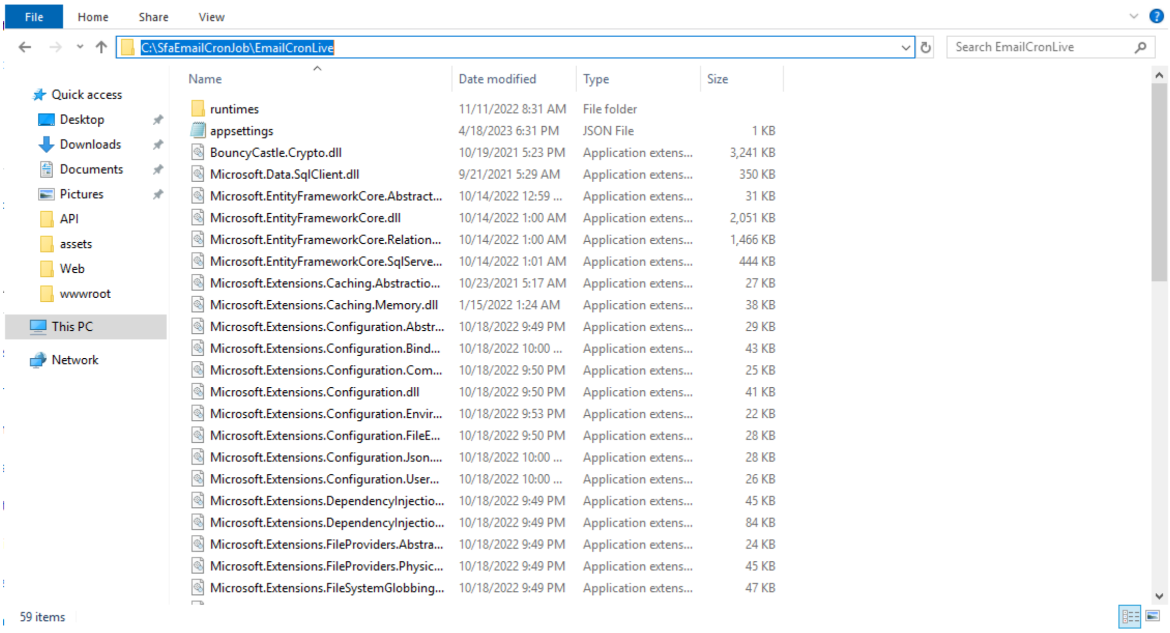
The published files will be .bat files which are ready for deployment.

1. Chronological Job for Email – SfaEmailCronJob
2. Chronological Job for Operational Jobs – SfaCronJob/

6.1 Chronological Job for Email – SfaEmailCronJob

Paste the published folder on the server on desired location. In the image below we have pasted it in C Drive.

1. In Windows Explorer we will go to the folder



C:\SfaEmailCronJob\EmailCronLive

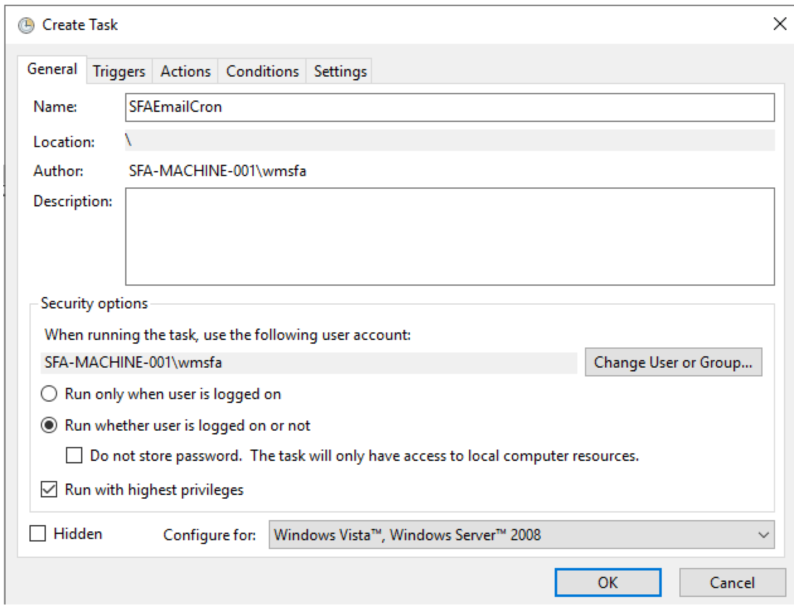
Name	Date modified	Type	Size
runtimes	11/11/2022 8:31 AM	File folder	
appsettings	4/18/2023 6:31 PM	JSON File	1 KB
BouncyCastle.Crypto.dll	10/19/2021 5:23 PM	Application extens...	3,241 KB
Microsoft.Data.SqlClient.dll	9/21/2021 5:29 AM	Application extens...	350 KB
Microsoft.EntityFrameworkCore.Abstract...	10/14/2022 12:59 ...	Application extens...	31 KB
Microsoft.EntityFrameworkCore.dll	10/14/2022 1:00 AM	Application extens...	2,051 KB
Microsoft.EntityFrameworkCore.Relation...	10/14/2022 1:00 AM	Application extens...	1,466 KB
Microsoft.EntityFrameworkCore.SqlServe...	10/14/2022 1:01 AM	Application extens...	444 KB
Microsoft.Extensions.Caching.Abstractio...	10/23/2021 5:17 AM	Application extens...	27 KB
Microsoft.Extensions.Caching.Memory.dll	1/15/2022 1:24 AM	Application extens...	38 KB
Microsoft.Extensions.Configuration.Abstr...	10/18/2022 9:49 PM	Application extens...	29 KB
Microsoft.Extensions.Configuration.Bind...	10/18/2022 10:00 ...	Application extens...	43 KB
Microsoft.Extensions.Configuration.Com...	10/18/2022 9:50 PM	Application extens...	25 KB
Microsoft.Extensions.Configuration.dll	10/18/2022 9:50 PM	Application extens...	41 KB
Microsoft.Extensions.Configuration.Envir...	10/18/2022 9:53 PM	Application extens...	22 KB
Microsoft.Extensions.Configuration.FileE...	10/18/2022 9:50 PM	Application extens...	28 KB
Microsoft.Extensions.Configuration.Json....	10/18/2022 10:00 ...	Application extens...	28 KB
Microsoft.Extensions.Configuration.User...	10/18/2022 10:00 ...	Application extens...	26 KB
Microsoft.Extensions.DependencyInjectio...	10/18/2022 9:49 PM	Application extens...	45 KB
Microsoft.Extensions.DependencyInjectio...	10/18/2022 9:49 PM	Application extens...	84 KB
Microsoft.Extensions.FileProviders.Abstra...	10/18/2022 9:49 PM	Application extens...	24 KB
Microsoft.Extensions.FileProviders.Physic...	10/18/2022 9:49 PM	Application extens...	45 KB
Microsoft.Extensions.FileSystemGlobbing...	10/18/2022 9:49 PM	Application extens...	47 KB

Copy the path of the exe file. In this case it is C:\SfaEmailCronJob\EmailCronLive\SfaEmailCronJobApp.exe

2. Next step would be to create a batch file to run the exe. A file named “SFAEmailCronJob.bat” is created with following lines:

```
@echo off

setlocal enabledelayedexpansion

set count=0

set max_count=120



:loop

if !count! geq !max_count! goto end

start /B "" "C:\SfaEmailCronJob\SfaEmailCronJob\SfaEmailCronJobApp.exe"

echo count

set /a count+=1

timeout /T 5 /nobreak >nul

goto loop



:end
```

3. Now we have to setup the Task Scheduler to execute the batch file. The images below show the settings to be filled while creating the scheduler job.



Name: SFAEmailCron

Location: \

Author: SFA-MACHINE-001\wmsfa

Description:

Security options

When running the task, use the following user account:

SFA-MACHINE-001\wmsfa

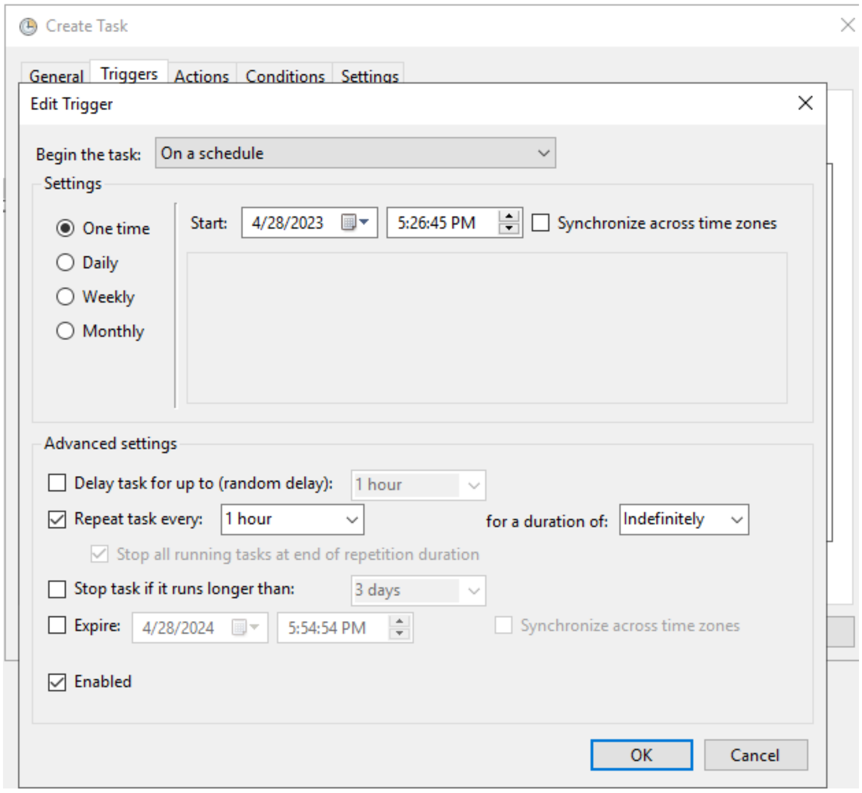
○ Run only when user is logged on

● Run whether user is logged on or not

□ Do not store password. The task will only have access to local computer resources.

☑ Run with highest privileges

□ Hidden Configure for: Windows Vista™, Windows Server™ 2008



Begin the task: On a schedule

Settings

● One time

○ Daily

○ Weekly

○ Monthly

Start: 4/28/2023 5:26:45 PM □ Synchronize across time zones

Advanced settings

□ Delay task for up to (random delay): 1 hour

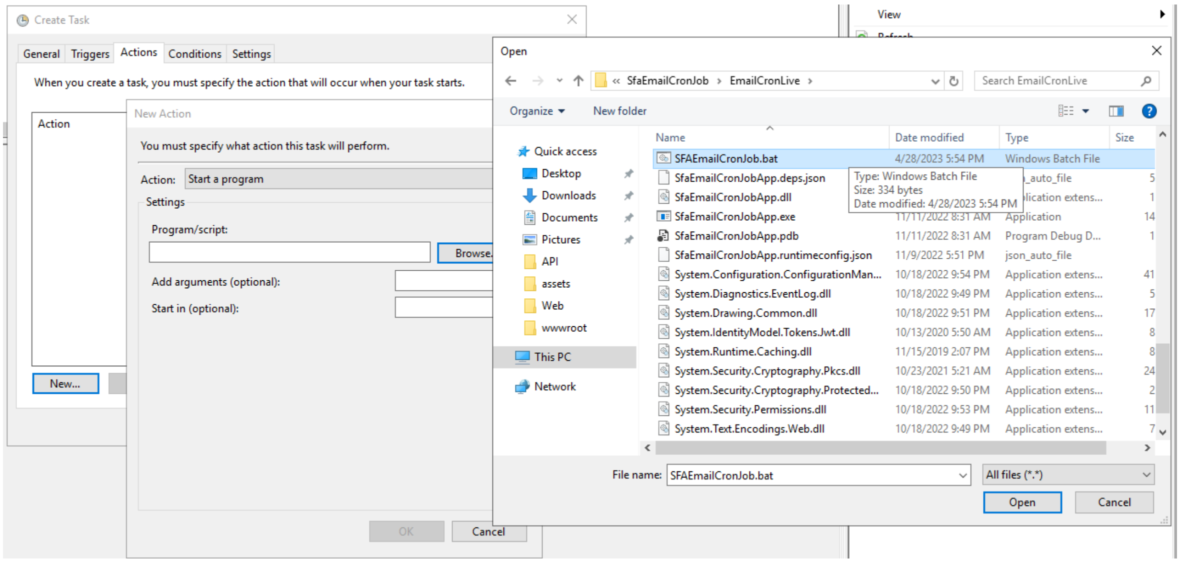
☑ Repeat task every: 1 hour for a duration of: Indefinitely

☑ Stop all running tasks at end of repetition duration

□ Stop task if it runs longer than: 3 days

□ Expire: 4/28/2024 5:54:54 PM □ Synchronize across time zones

☑ Enabled



When you create a task, you must specify the action that will occur when your task starts.

You must specify what action this task will perform.

Action: Start a program

Settings

Program/script:

Add arguments (optional):

Start in (optional):

« SfaEmailCronJob › EmailCronLive ›

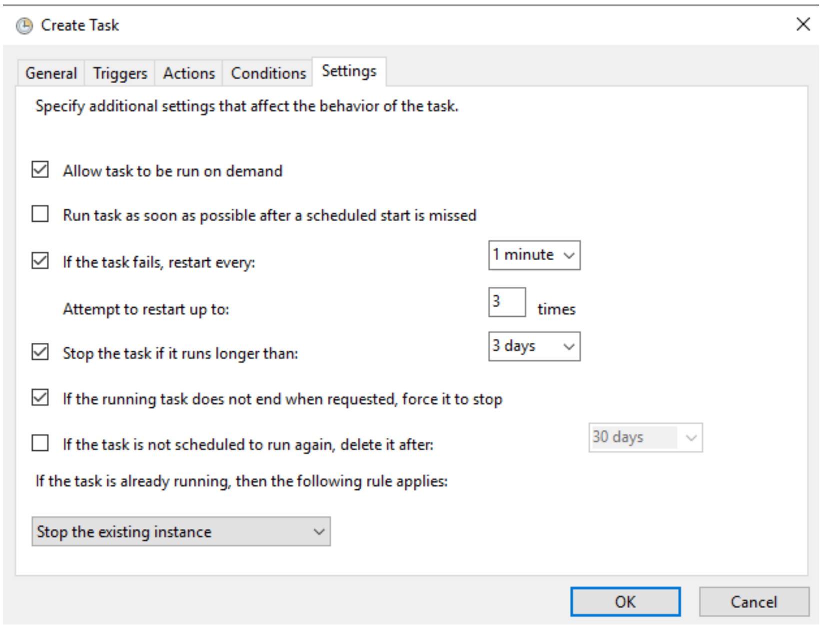
Search EmailCronLive

Name	Date modified	Type	Size
SFAEmailCronJob.bat	4/28/2023 5:54 PM	Windows Batch File	
SfaEmailCronJobApp.deps.json		_auto_file	5
SfaEmailCronJobApp.dll		lication extens...	1
SfaEmailCronJobApp.exe	11/11/2022 8:31 AM	Application	14
SfaEmailCronJobApp.pdb	11/11/2022 8:31 AM	Program Debug D...	1
SfaEmailCronJobApp.runtimeconfig.json	11/9/2022 5:51 PM	json_auto_file	
System.Configuration.ConfigurationMan...	10/18/2022 9:54 PM	Application extens...	41
System.Diagnostics.EventLog.dll	10/18/2022 9:49 PM	Application extens...	5
System.Drawing.Common.dll	10/18/2022 9:51 PM	Application extens...	17
System.IdentityModel.Tokens.Jwt.dll	10/13/2020 5:50 AM	Application extens...	8
System.Runtime.Caching.dll	11/15/2019 2:07 PM	Application extens...	8
System.Security.Cryptography.Pkcs.dll	10/23/2021 5:21 AM	Application extens...	24
System.Security.Cryptography.Protected...	10/18/2022 9:50 PM	Application extens...	2
System.Security.Permissions.dll	10/18/2022 9:53 PM	Application extens...	11
System.Text.Encodings.Web.dll	10/18/2022 9:49 PM	Application extens...	7

Type: Windows Batch File
Size: 334 bytes
Date modified: 4/28/2023 5:54 PM

File name: SFAEmailCronJob.bat

All files (*.*)



Specify additional settings that affect the behavior of the task.

☑ Allow task to be run on demand

☐ Run task as soon as possible after a scheduled start is missed

☑ If the task fails, restart every: 1 minute

Attempt to restart up to: 3 times

☑ Stop the task if it runs longer than: 3 days

☑ If the running task does not end when requested, force it to stop

☐ If the task is not scheduled to run again, delete it after: 30 days

If the task is already running, then the following rule applies:

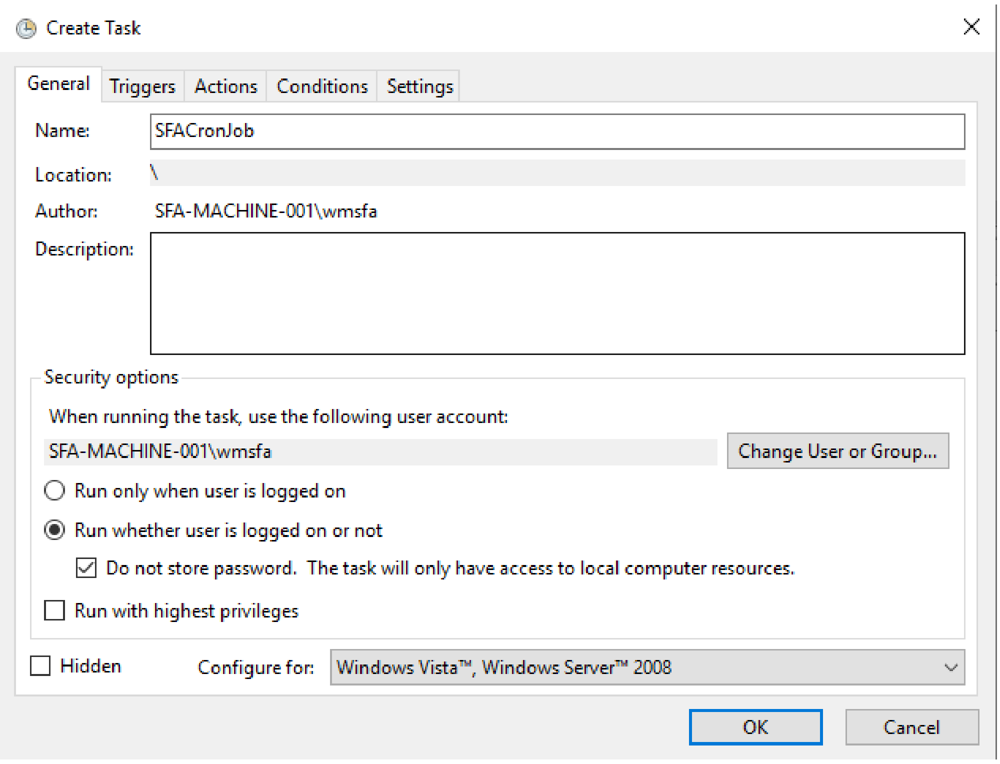
Stop the existing instance

6.2 Chronological Job for Operational Jobs – SfaCronJob/

1. The published folder to be placed on desired location on the server and copy the path of the exe. In this case the path to the exe is ” C:\SfaCronJob\SfaCronJob\SfaCronJobApp.exe”
2. Batch file to be created and placed in the folder of the SFACronJob

```
@echo off

start /B "" "C:\SfaCronJob\SfaCronJob\SfaCronJobApp.exe"
```

3. Create Task Scheduler Job to execute the batch file. The task job with settings shown in the image below:



Create Task

Name: SFACronJob

Location: \

Author: SFA-MACHINE-001\wmsfa

Description:

Security options

When running the task, use the following user account:

SFA-MACHINE-001\wmsfa

○ Run only when user is logged on

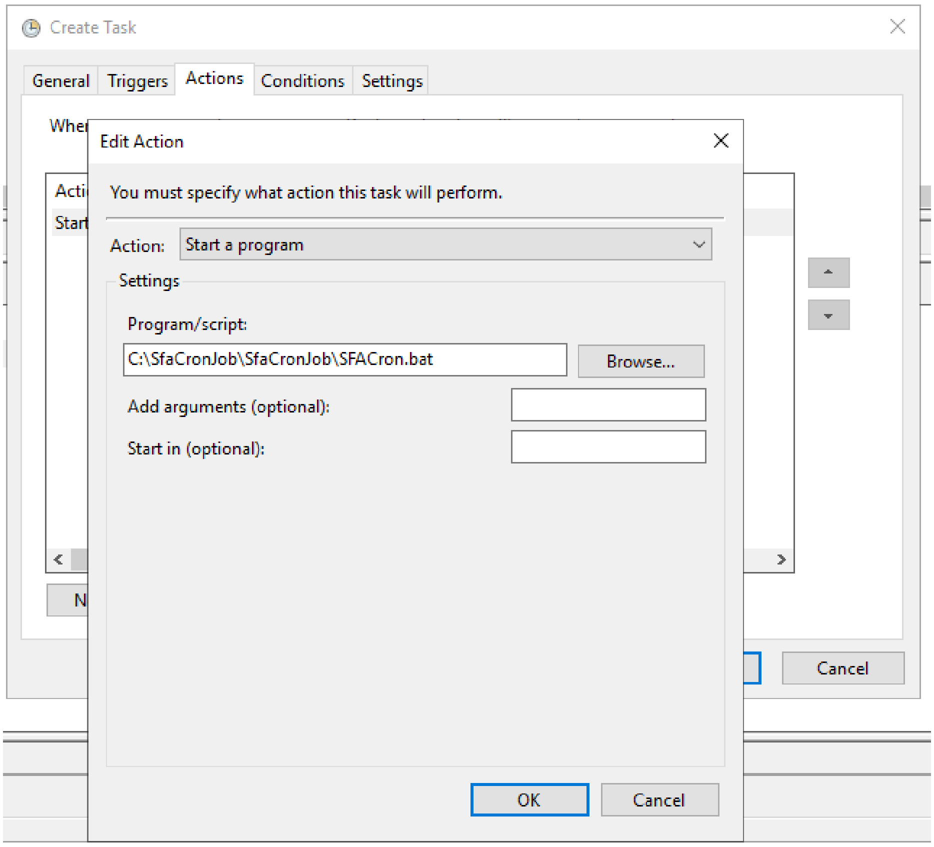
● Run whether user is logged on or not

☑ Do not store password. The task will only have access to local computer resources.

□ Run with highest privileges

□ Hidden

Configure for: Windows Vista™, Windows Server™ 2008



Create Task

Edit Action

You must specify what action this task will perform.

Action: Start a program

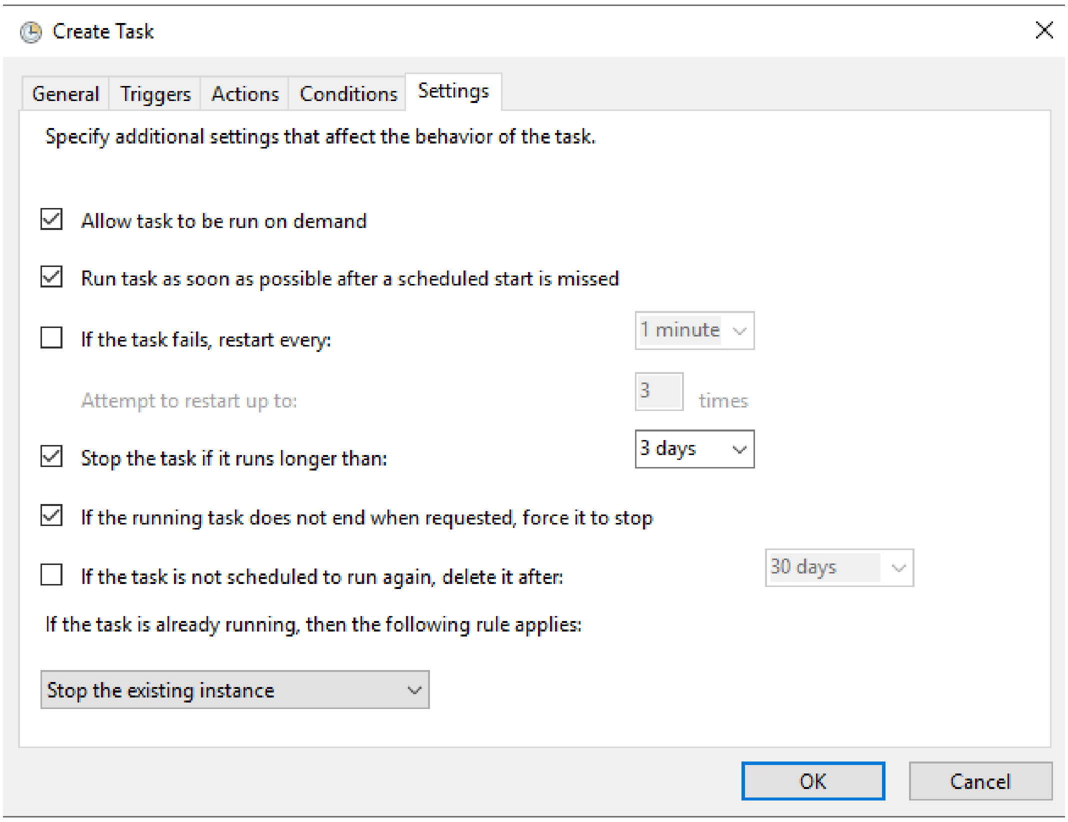
Settings

Program/script:

C:\SfaCronJob\SfaCronJob\SFACron.bat

Add arguments (optional):

Start in (optional):



Specify additional settings that affect the behavior of the task.

☑ Allow task to be run on demand

☑ Run task as soon as possible after a scheduled start is missed

☐ If the task fails, restart every: 1 minute

Attempt to restart up to: 3 times

☑ Stop the task if it runs longer than: 3 days

☑ If the running task does not end when requested, force it to stop

☐ If the task is not scheduled to run again, delete it after: 30 days

If the task is already running, then the following rule applies:

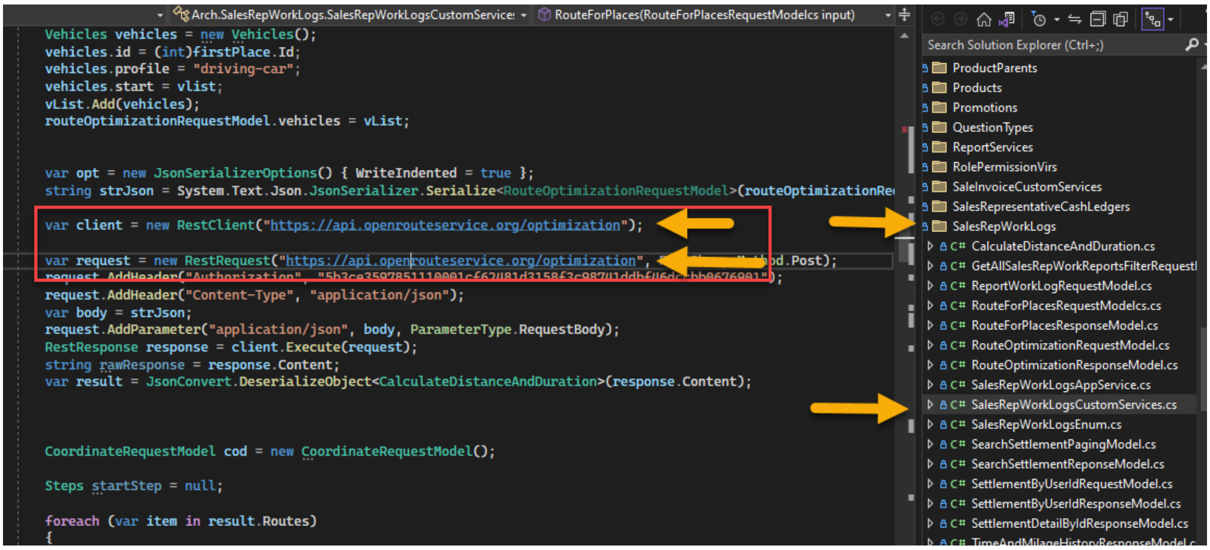
Stop the existing instance

7 Map Server Deployment

The Ubuntu server has to be installed with following:

1. ORS installation installation steps - https://giscience.github.io/openrouteservice/installation/Running-with-Docker.html
2. VROOM Installation steps- https://github.com/VROOM-Project/vroom-docker/blob/master/README.md

   Once both are setup and configured with the desired planet data file. Replace the API URL in the highlighted file at two locations as shown in the image below highlighted by yellow arrow.



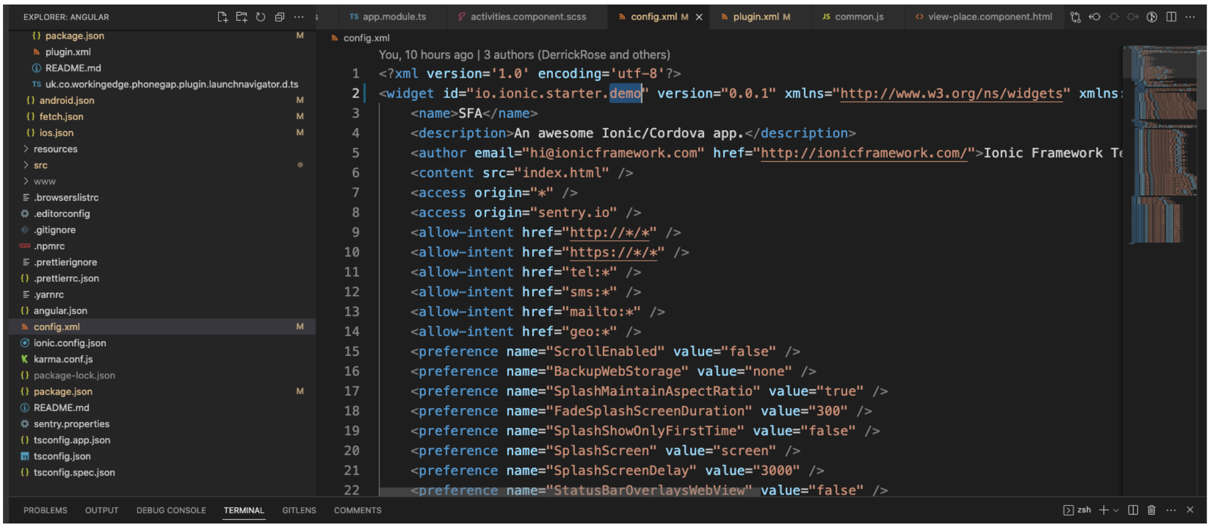
Arch.SalesRepWorkLogs.SalesRepWorkLogsCustomService:
RouteForPlaces(RouteForPlacesRequestModelcs input)
Search Solution Explorer (Ctrl+;)

```
Vehicles vehicles = new Vehicles();
vehicles.id = (int)firstPlace.Id;
vehicles.profile = "driving-car";
vehicles.start = vlist;
vList.Add(vehicles);
routeOptimizationRequestModel.vehicles = vList;


var opt = new JsonSerializerOptions() { WriteIndented = true };
string strJson = System.Text.Json.JsonSerializer.Serialize<RouteOptimizationRequestModel>(routeOptimizationRe

var client = new RestClient("https://api.openrouteservice.org/optimization");

var request = new RestRequest("https://api.openrouteservice.org/optimization", Method.Post);
request.AddHeader("Authorization", "5b3ce3597851110001cf62481d3158f3c98741ddbf46dccbb0676901");
request.AddHeader("Content-Type", "application/json");
var body = strJson;
request.AddParameter("application/json", body, ParameterType.RequestBody);
RestResponse response = client.Execute(request);
string rawResponse = response.Content;
var result = JsonConvert.DeserializeObject<CalculateDistanceAndDuration>(response.Content);


CoordinateRequestModel cod = new CoordinateRequestModel();

Steps startStep = null;

foreach (var item in result.Routes)
{
```

```
ProductParents
Products
Promotions
QuestionTypes
ReportServices
RolePermissionVirs
SaleInvoiceCustomServices
SalesRepresentativeCashLedgers
SalesRepWorkLogs
  CalculateDistanceAndDuration.cs
  GetAllSalesRepWorkReportsFilterRequestl
  ReportWorkLogRequestModel.cs
  RouteForPlacesRequestModelcs.cs
  RouteForPlacesResponseModel.cs
  RouteOptimizationRequestModel.cs
  RouteOptimizationResponseModel.cs
  SalesRepWorkLogsAppService.cs
  SalesRepWorkLogsCustomServices.cs
  SalesRepWorkLogsEnum.cs
  SearchSettlementPagingModel.cs
  SearchSettlementReponseModel.cs
  SettlementByUserIdRequestModel.cs
  SettlementByUserIdResponseModel.cs
  SettlementDetailByIdResponseModel.cs
  TimeAndMilageHistoryResponseModel.c
```

8 Mobile App Build

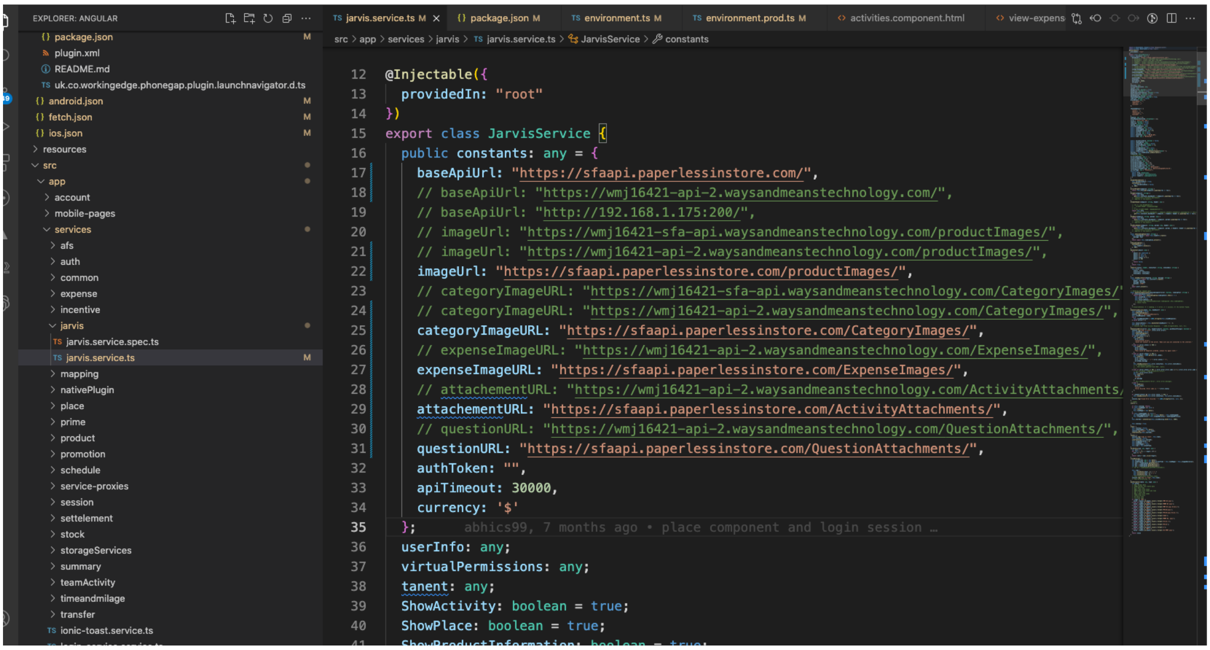
1. Take pull from the git and open the project in visual code IDE.
2. Prepare the environment using npm command for the project.
3. Building for android:
    a. As shown in the image below change the widget id in the project in config.xml file:



```xml
<?xml version='1.0' encoding='utf-8'?>
<widget id="io.ionic.starter.demo" version="0.0.1" xmlns="http://www.w3.org/ns/widgets" xmlns:
    <name>SFA</name>
    <description>An awesome Ionic/Cordova app.</description>
    <author email="hi@ionicframework.com" href="http://ionicframework.com/">Ionic Framework Te
    <content src="index.html" />
    <access origin="*" />
    <access origin="sentry.io" />
    <allow-intent href="http://*/*" />
    <allow-intent href="https://*/*" />
    <allow-intent href="tel:*" />
    <allow-intent href="sms:*" />
    <allow-intent href="mailto:*" />
    <allow-intent href="geo:*" />
    <preference name="ScrollEnabled" value="false" />
    <preference name="BackupWebStorage" value="none" />
    <preference name="SplashMaintainAspectRatio" value="true" />
    <preference name="FadeSplashScreenDuration" value="300" />
    <preference name="SplashShowOnlyFirstTime" value="false" />
    <preference name="SplashScreen" value="screen" />
    <preference name="SplashScreenDelay" value="3000" />
    <preference name="StatusBarOverlaysWebView" value="false" />
```

Ensure that the value of widget id will be “io.ionic.starter.demo”. Check the highlighted part in the image above.

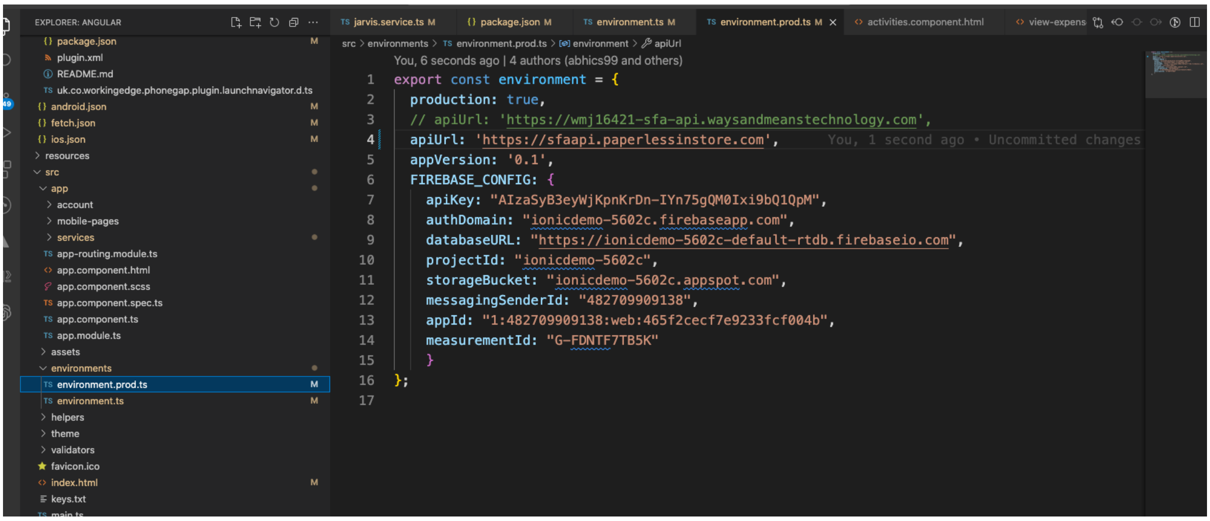
   b. Update API URL in the file Jarvis.service.ts as shown below in the image.



```typescript
@Injectable({
  providedIn: "root"
})
export class JarvisService {
  public constants: any = {
    baseApiUrl: "https://sfaapi.paperlessinstore.com/",
    // baseApiUrl: "https://wmj16421-api-2.waysandmeanstechnology.com/",
    // baseApiUrl: "http://192.168.1.175:200/",
    // imageUrl: "https://wmj16421-sfa-api.waysandmeanstechnology.com/productImages/",
    // imageUrl: "https://wmj16421-api-2.waysandmeanstechnology.com/productImages/",
    imageUrl: "https://sfaapi.paperlessinstore.com/productImages/",
    // categoryImageURL: "https://wmj16421-sfa-api.waysandmeanstechnology.com/CategoryImages/"
    // categoryImageURL: "https://wmj16421-api-2.waysandmeanstechnology.com/CategoryImages/",
    categoryImageURL: "https://sfaapi.paperlessinstore.com/CategoryImages/",
    // expenseImageURL: "https://wmj16421-api-2.waysandmeanstechnology.com/ExpenseImages/",
    expenseImageURL: "https://sfaapi.paperlessinstore.com/ExpenseImages/",
    // attachementURL: "https://wmj16421-api-2.waysandmeanstechnology.com/ActivityAttachments/
    attachementURL: "https://sfaapi.paperlessinstore.com/ActivityAttachments/",
    // questionURL: "https://wmj16421-api-2.waysandmeanstechnology.com/QuestionAttachments/",
    questionURL: "https://sfaapi.paperlessinstore.com/QuestionAttachments/",
    authToken: "",
    apiTimeout: 30000,
    currency: '$'
  };      abhics99, 7 months ago • place component and login session …
  userInfo: any;
  virtualPermissions: any;
  tanent: any;
  ShowActivity: boolean = true;
  ShowPlace: boolean = true;
```

API URL to be updated for baseAPIURL, imageURL, category image URL, expense image URL, attachment URL and questionURL as shown in the image above.

   c. Update API URL on environment.prod.ts



You, 6 seconds ago | 4 authors (abhics99 and others)

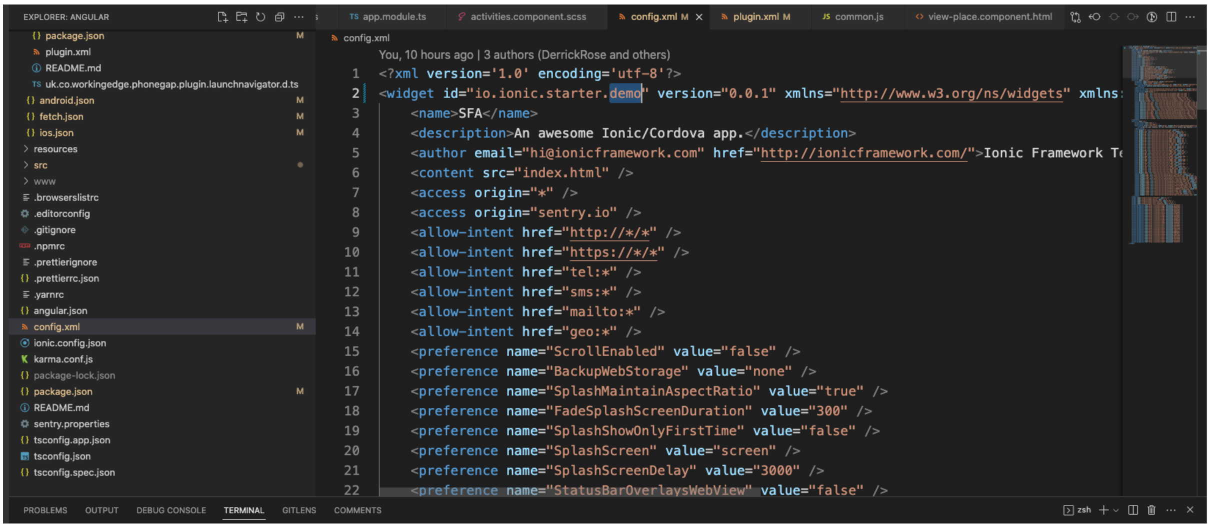
```
export const environment = {
  production: true,
  // apiUrl: 'https://wmj16421-sfa-api.waysandmeanstechnology.com',
  apiUrl: 'https://sfaapi.paperlessinstore.com',       You, 1 second ago • Uncommitted changes
  appVersion: '0.1',
  FIREBASE_CONFIG: {
    apiKey: "AIzaSyB3eyWjKpnKrDn-IYn75gQM0Ixi9bQ1QpM",
    authDomain: "ionicdemo-5602c.firebaseapp.com",
    databaseURL: "https://ionicdemo-5602c-default-rtdb.firebaseio.com",
    projectId: "ionicdemo-5602c",
    storageBucket: "ionicdemo-5602c.appspot.com",
    messagingSenderId: "482709909138",
    appId: "1:482709909138:web:465f2cecf7e9233fcf004b",
    measurementId: "G-FDNTF7TB5K"
  }
};
```

d. Prepare to build by executing the command
"ionic cordova prepare android"

e. Initiate building process by executing the command
"ionic cordova build android"

Build is created.

4. Building for iOS
   a. As shown in the image below change the widget id in the project in config.xml file:

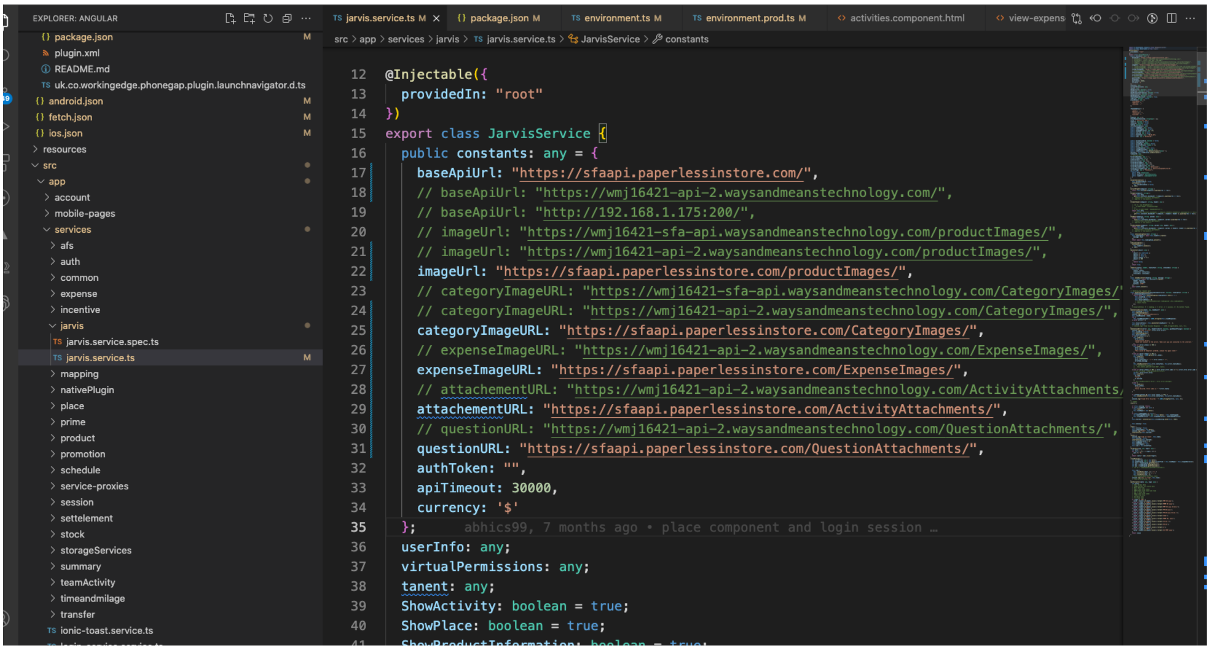


You, 10 hours ago | 3 authors (DerrickRose and others)

```
<?xml version='1.0' encoding='utf-8'?>
<widget id="io.ionic.starter.demo" version="0.0.1" xmlns="http://www.w3.org/ns/widgets" xmlns:
    <name>SFA</name>
    <description>An awesome Ionic/Cordova app.</description>
    <author email="hi@ionicframework.com" href="http://ionicframework.com/">Ionic Framework Te
    <content src="index.html" />
    <access origin="*" />
    <access origin="sentry.io" />
    <allow-intent href="http://*/*" />
    <allow-intent href="https://*/*" />
    <allow-intent href="tel:*" />
    <allow-intent href="sms:*" />
    <allow-intent href="mailto:*" />
    <allow-intent href="geo:*" />
    <preference name="ScrollEnabled" value="false" />
    <preference name="BackupWebStorage" value="none" />
    <preference name="SplashMaintainAspectRatio" value="true" />
    <preference name="FadeSplashScreenDuration" value="300" />
    <preference name="SplashShowOnlyFirstTime" value="false" />
    <preference name="SplashScreen" value="screen" />
    <preference name="SplashScreenDelay" value="3000" />
    <preference name="StatusBarOverlaysWebView" value="false" />
```

Ensure that the value of widget id will be "io.ionic.starter.sfa". Check the highlighted part in the image above.

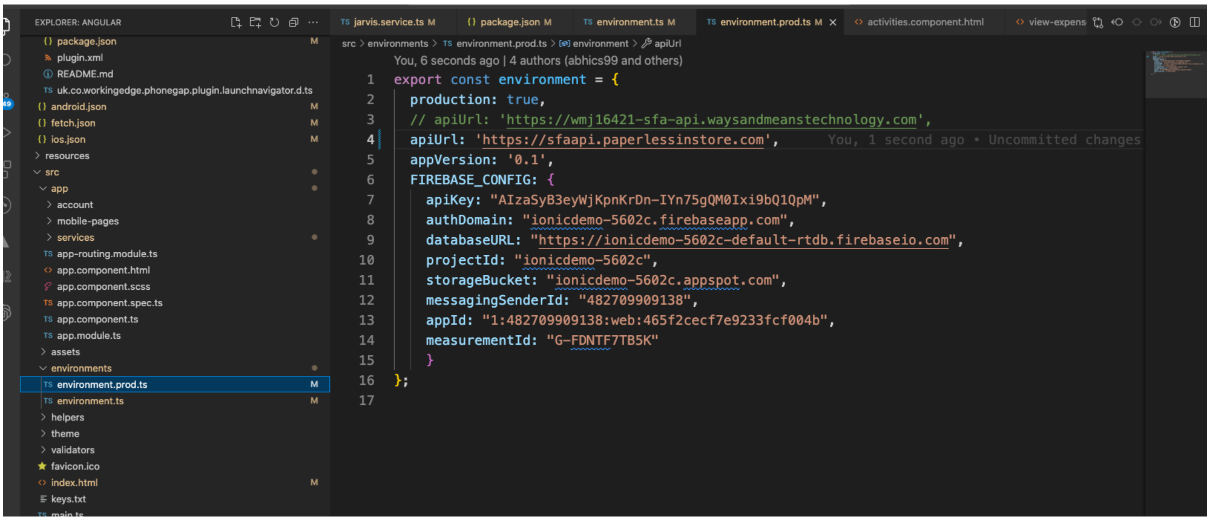
   b. Update API URL in the file Jarvis.service.ts as shown below in the image.



```
@Injectable({
  providedIn: "root"
})
export class JarvisService {
  public constants: any = {
    baseApiUrl: "https://sfaapi.paperlessinstore.com/",
    // baseApiUrl: "https://wmj16421-api-2.waysandmeanstechnology.com/",
    // baseApiUrl: "http://192.168.1.175:200/",
    // imageUrl: "https://wmj16421-sfa-api.waysandmeanstechnology.com/productImages/",
    // imageUrl: "https://wmj16421-api-2.waysandmeanstechnology.com/productImages/",
    imageUrl: "https://sfaapi.paperlessinstore.com/productImages/",
    // categoryImageURL: "https://wmj16421-sfa-api.waysandmeanstechnology.com/CategoryImages/"
    // categoryImageURL: "https://wmj16421-api-2.waysandmeanstechnology.com/CategoryImages/",
    categoryImageURL: "https://sfaapi.paperlessinstore.com/CategoryImages/",
    // expenseImageURL: "https://wmj16421-api-2.waysandmeanstechnology.com/ExpenseImages/",
    expenseImageURL: "https://sfaapi.paperlessinstore.com/ExpenseImages/",
    // attachementURL: "https://wmj16421-api-2.waysandmeanstechnology.com/ActivityAttachments/
    attachementURL: "https://sfaapi.paperlessinstore.com/ActivityAttachments/",
    // questionURL: "https://wmj16421-api-2.waysandmeanstechnology.com/QuestionAttachments/",
    questionURL: "https://sfaapi.paperlessinstore.com/QuestionAttachments/",
    authToken: "",
    apiTimeout: 30000,
    currency: '$'
  };      abhics99, 7 months ago • place component and login session …
  userInfo: any;
  virtualPermissions: any;
  tanent: any;
  ShowActivity: boolean = true;
  ShowPlace: boolean = true;
  ShowProductInformation: boolean = true;
```

API URL to be updated for baseAPIURL, imageURL, category image URL, expense image URL, attachment URL and questionURL as shown in the image above.

c. Update API URL on environment.prod.ts



```
export const environment = {
  production: true,
  // apiUrl: 'https://wmj16421-sfa-api.waysandmeanstechnology.com',
  apiUrl: 'https://sfaapi.paperlessinstore.com',      You, 1 second ago • Uncommitted changes
  appVersion: '0.1',
  FIREBASE_CONFIG: {
    apiKey: "AIzaSyB3eyWjKpnKrDn-IYn75gQM0Ixi9bQ1QpM",
    authDomain: "ionicdemo-5602c.firebaseapp.com",
    databaseURL: "https://ionicdemo-5602c-default-rtdb.firebaseio.com",
    projectId: "ionicdemo-5602c",
    storageBucket: "ionicdemo-5602c.appspot.com",
    messagingSenderId: "482709909138",
    appId: "1:482709909138:web:465f2cecf7e9233fcf004b",
    measurementId: "G-FDNTF7TB5K"
    }
};
```

d. Prepare to build by executing the command
“ionic cordova prepare iOS”

e. Initiate building process by executing the command
“ionic cordova build iOS”

Build is created.