[1. Dotcover coverage on API project 2](#__RefHeading___Toc242_1801466266)

[Step 1: Run project 2](#__RefHeading___Toc275_1801466266)

[Step 2: Set Up dotCover 2](#__RefHeading___Toc277_1801466266)

[Step 3: Run dotCover.exe 2](#__RefHeading___Toc279_1801466266)

[Method 1: Using analyse command 2](#__RefHeading___Toc281_1801466266)

[Method 2: Using cover command 3](#__RefHeading___Toc283_1801466266)

[Method 3: Using dotnet run commands along with dotCover 3](#__RefHeading___Toc285_1801466266)

[Step 4: Perform Testing 4](#__RefHeading___Toc252_1801466266_Copy_1)

[Step 5: Stop the session 4](#__RefHeading___Toc254_1801466266_Copy_1)

[Step 6: Generating Reports and Merging Snapshots 5](#__RefHeading___Toc256_1801466266_Copy_1)

[Sample Report IMG references: 6](#__RefHeading___Toc358_1801466266)

[Sample XML report: 6](#__RefHeading___Toc360_1801466266)

[Sample HTML Report: 7](#__RefHeading___Toc362_1801466266)

[Report with dcvr extension 7](#__RefHeading___Toc364_1801466266)

[2. Dotcover coverage on Calculator app 9](#__RefHeading___Toc242_1801466266_Copy_4)

[Step 1: Set up Project 9](#__RefHeading___Toc277_1801466266_Copy_1)

[Step 2: Set Up dotCover 9](#__RefHeading___Toc409_1801466266)

[Step 3: Start app with dotCover.exe 9](#__RefHeading___Toc279_1801466266_Copy_1)

[Step 4: Perform Testing 10](#__RefHeading___Toc411_1801466266)

[Step 5: Stop the session 11](#__RefHeading___Toc254_1801466266_Copy_2)

[Step 6: Generating Reports and Merging Snapshots 12](#__RefHeading___Toc256_1801466266_Copy_2)

[3. Dotcover coverage on IIS 13](#__RefHeading___Toc242_1801466266_Copy_3)

[Step 1: Publish the project. 13](#__RefHeading___Toc244_1801466266)

[Step 2: Set up in IIS 13](#__RefHeading___Toc246_1801466266)

[Step 3: Check app running status by hitting api or UI. 14](#__RefHeading___Toc248_1801466266)

[Step 4: Start dotcover session with IIS 15](#__RefHeading___Toc250_1801466266)

[Step 5: Perform Testing 16](#__RefHeading___Toc252_1801466266)

[Step 6: Stop the session 16](#__RefHeading___Toc254_1801466266)

[Step 7: Generating Reports and Merging Snapshots 16](#__RefHeading___Toc256_1801466266)

# 1. Dotcover coverage on API project

## Step 1: Run project

On running below command, project will start building & run the respective project. Ensure the user is respective directory.

CMD: dotnet run

Sample Project: https://github.com/FabianGosebrink/ASPNETCore-WebAPI-Sample

Verify app is running successfully in the specified port. Then, stop the app.

## Step 2: Set Up dotCover

Download dependent dotcover tools.

Command line dotcover tool here: [https://www.jetbrains.com/dotcover/download/#section=commandline](https://www.jetbrains.com/dotcover/download/" \l "section=commandline)

Extract them into project folder where you want to run dotcover commands.

Or

store them in a folder & run from there by providing full path of app.exe or app.dll

Or

App path of the folder into System path and run the commands from anywhere

Run below two commands in the same dir:

*dotnet new tool-manifest*

*dotnet tool install JetBrains.dotCover.GlobalTool*

## Step 3: Run dotCover.exe

There are multiple commands to start the coverage session.

### Method 1: Using analyse command

On the properly built application, Run below command to start the application & coverage session along with it.

*dotcover analyse config.xml*

Contents of config.xml:

<AnalyseParams>

<TargetExecutable>path\to\.net\project\src\App\bin\Debug\net7.0\App.exe </TargetExecutable>

<TargetWorkingDir>bin\Debug\net7.0\</TargetWorkingDir>

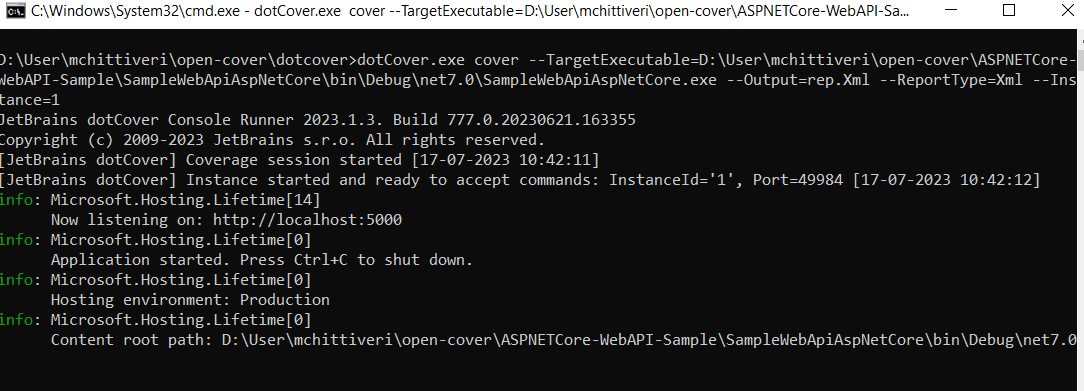
<Output>report.xml</Output>

<Instance>1</Instance>

</AnalyseParams>

### Method 2: Using cover command

dotCover.exe cover --TargetExecutable= path\to\.net\project\src\App\bin\Debug\net7.0\App.exe --Output=report.Xml --ReportType=Xml –Instance=1



The drawback w.r.t these methods is that it will start app is default port which is 5000.

### Method 3: Using dotnet run commands along with dotCover

**Problem statement:** App is launching in default port number 5000. It’s not picking ports from launchSettings.json file.

Reference: <https://dotnettools-support.jetbrains.com/hc/en-us/requests/5304948?page=1>

**Solution**: Using below command to send dotnet commands along with dotCover.exe command.

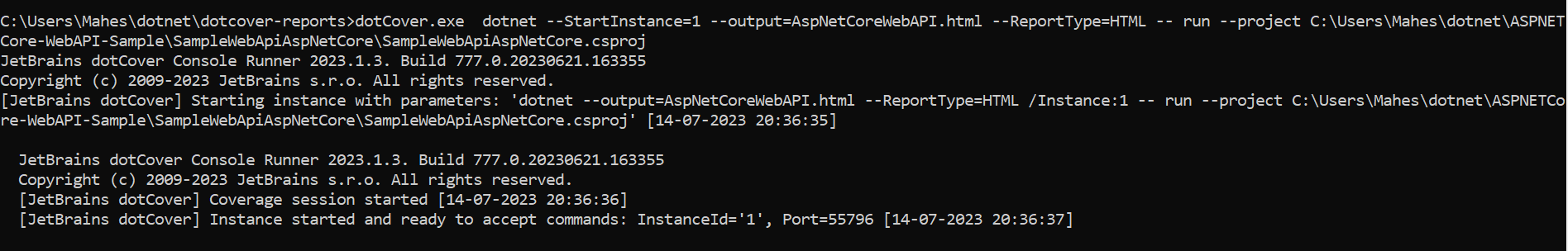
CMD:

.\dotCover.exe dotnet --StartInstance=1 --output=AspNetCoreWebAPI.html --ReportType=HTML -- run [dotnet commands here]

Command to run specific project will pick respective launch settings.

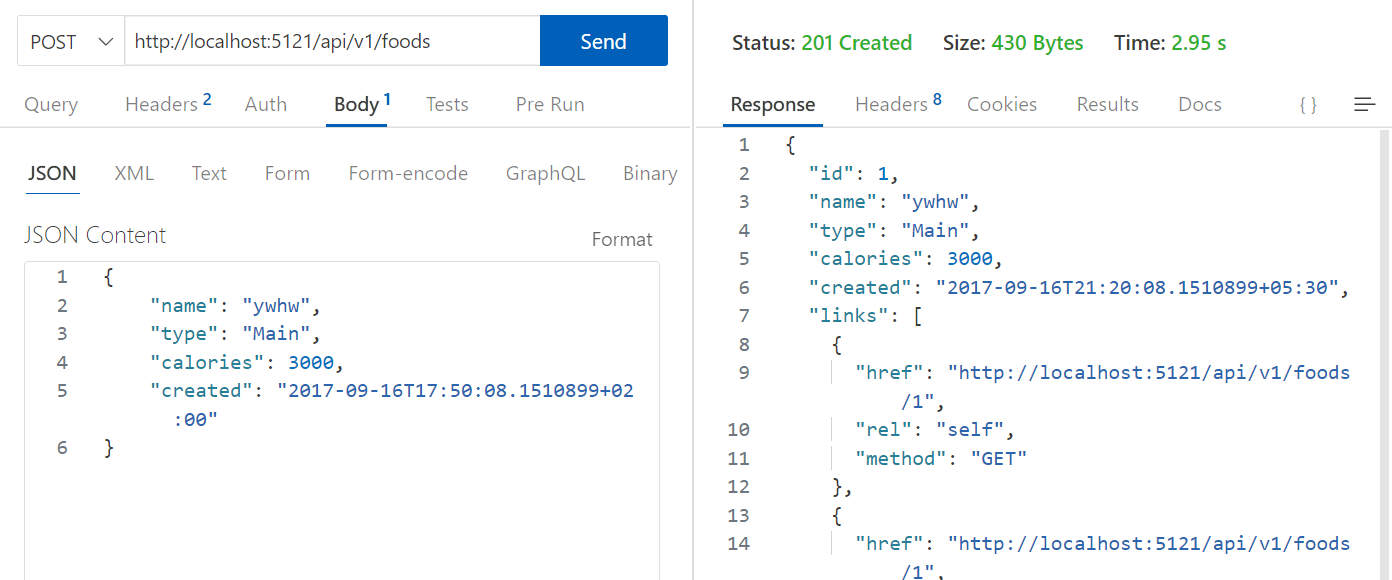
CMD:

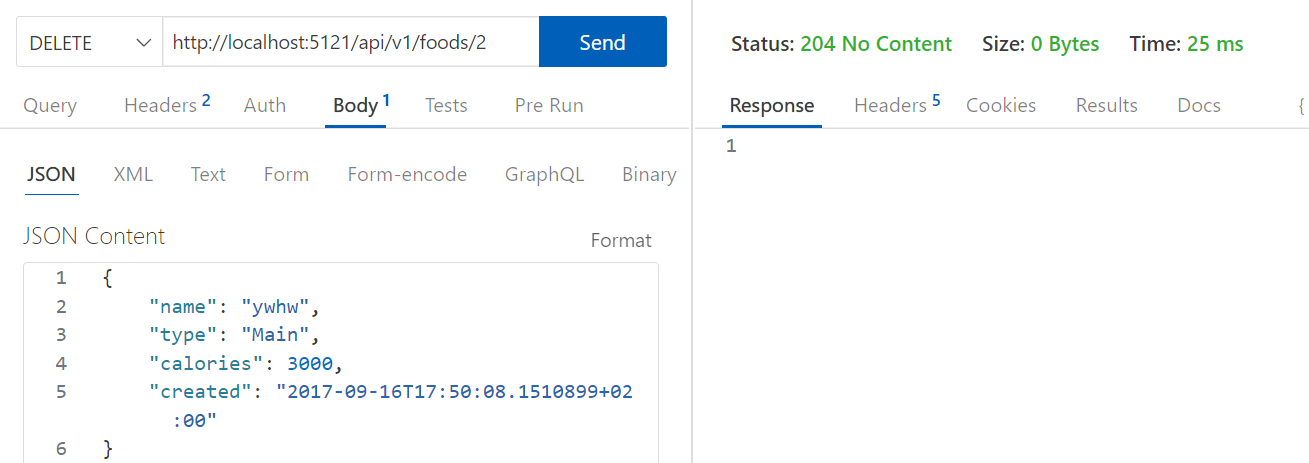
.\dotCover.exe dotnet --StartInstance=1 --output=AspNetCoreWebAPI.html --ReportType=HTML -- run --project path\to\csproject file\SampleWebApiAspNetCore.csproj



## Step 4: Perform Testing

Hit some api and check the responses

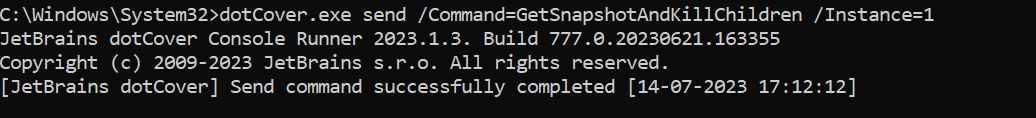


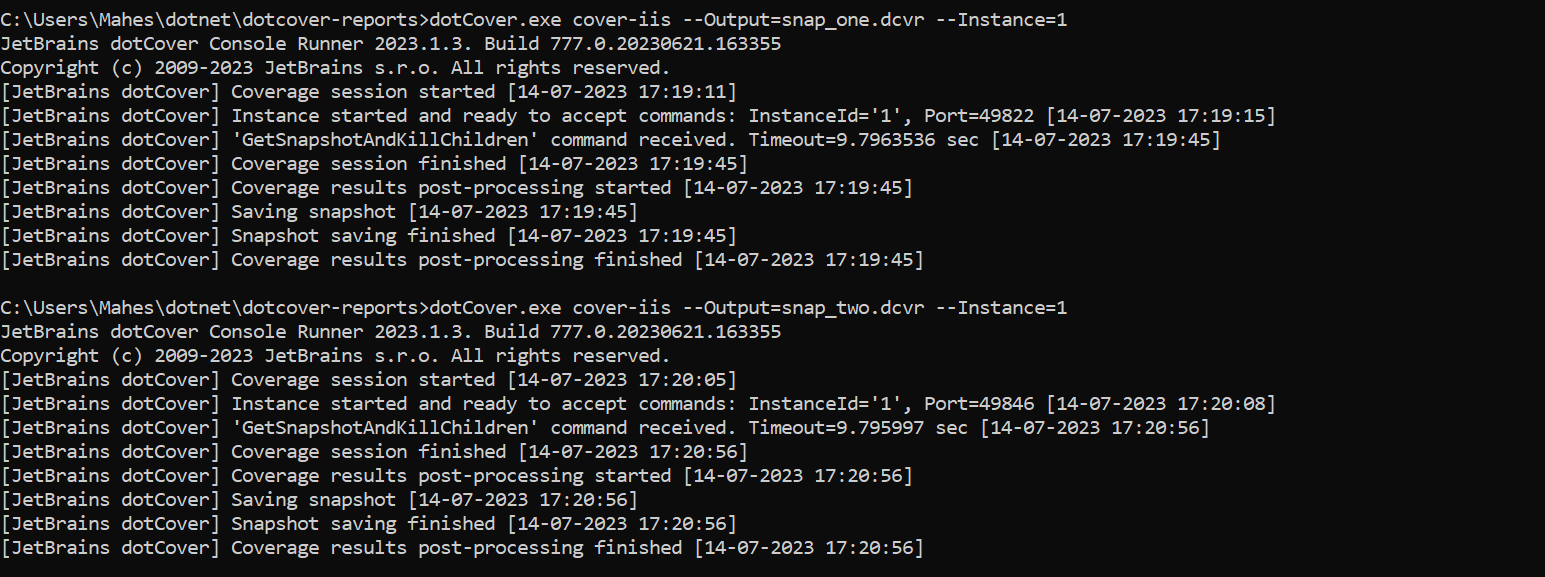


## Step 5: Stop the session

Stop the coverage session by using below command.

dotCover.exe send /Command=GetSnapshotAndKillChildren /Instance=1





## Step 6: Generating Reports and Merging Snapshots

Merge two snapshots using the below command.

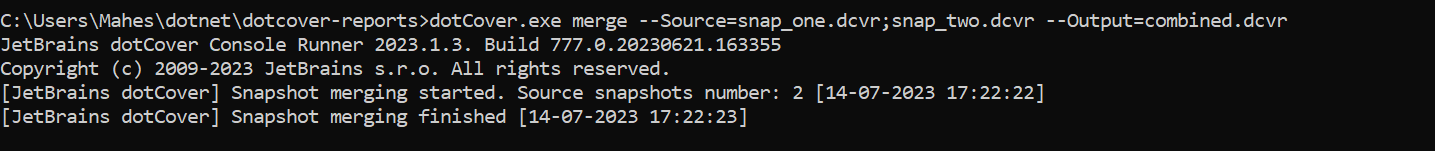
dotCover.exe merge --Source=snap\_one.dcvr;snap\_two.dcvr --Output=combined.dcvr

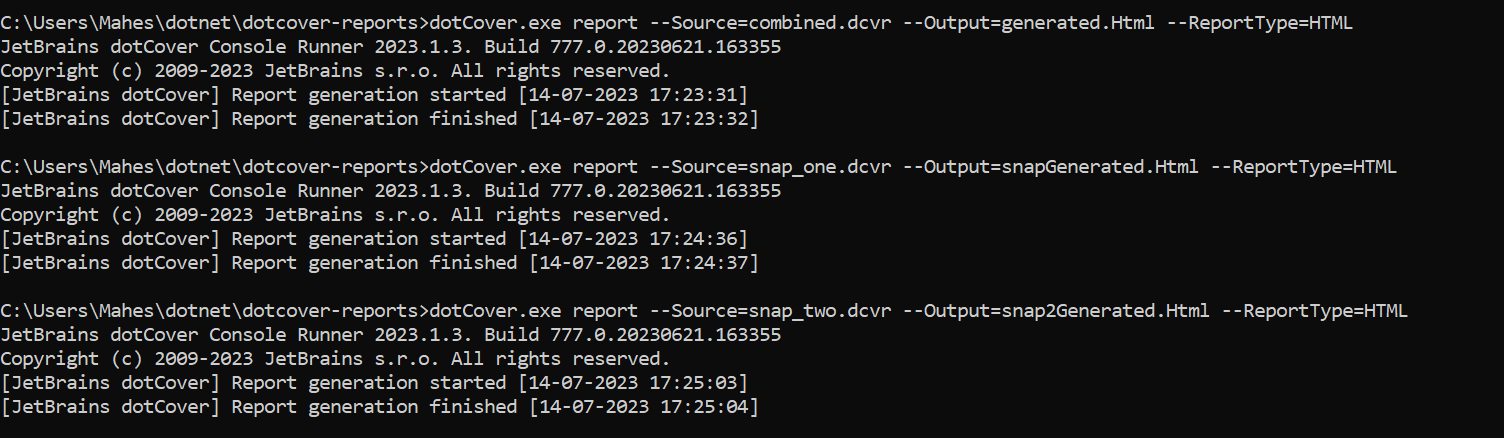
Generate HTML report from the snapshot using the below command.

dotCover.exe report --Source=combined.dcvr --Output=generated.Html --ReportType=HTML

dotCover.exe report --Source=snap\_one.dcvr --Output=snapGenerated.Html --ReportType=HTML

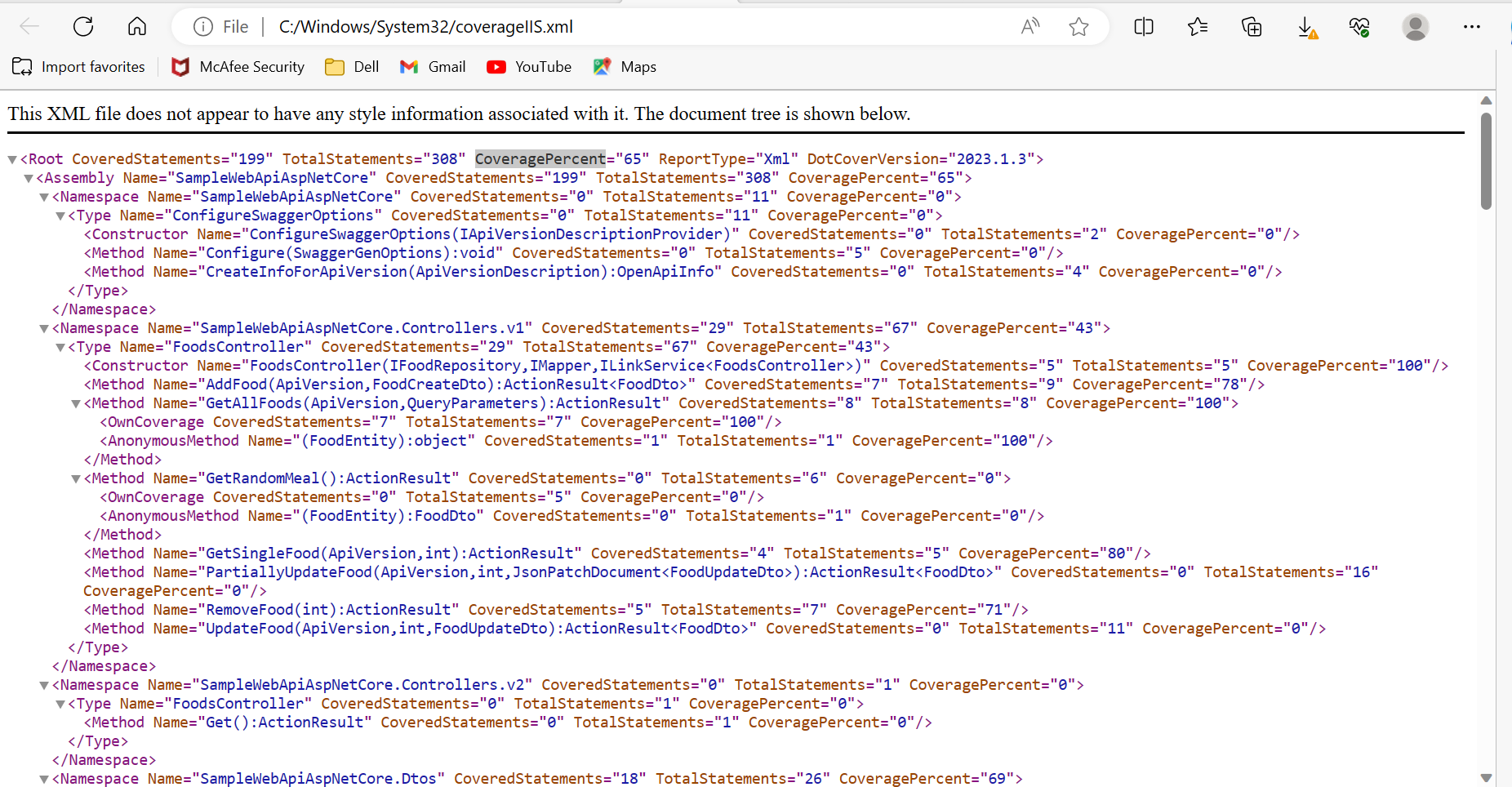
dotCover.exe report --Source=snap\_two.dcvr --Output=snap2Generated.Html --ReportType=HTML



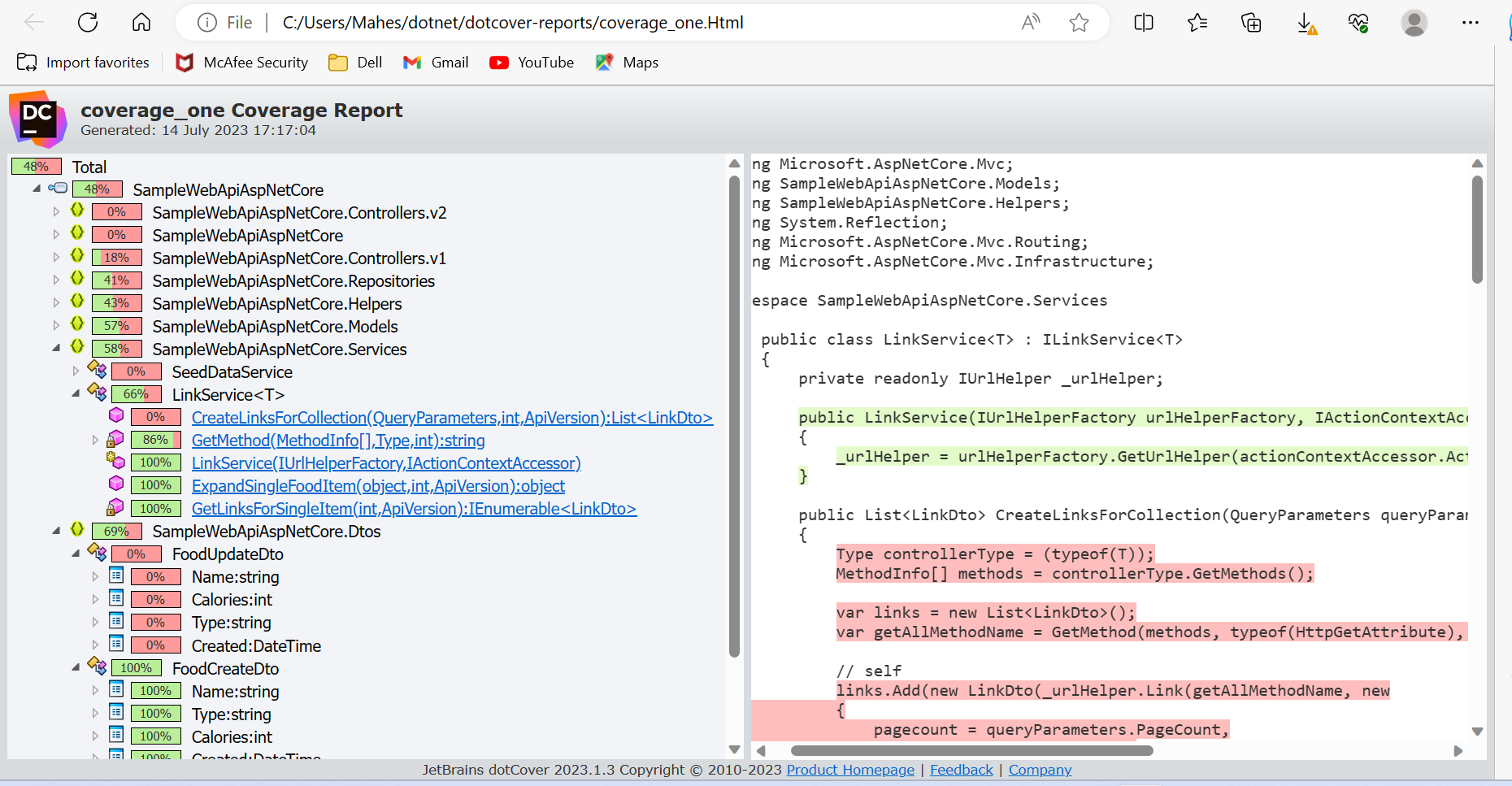


## Sample Report IMG references:

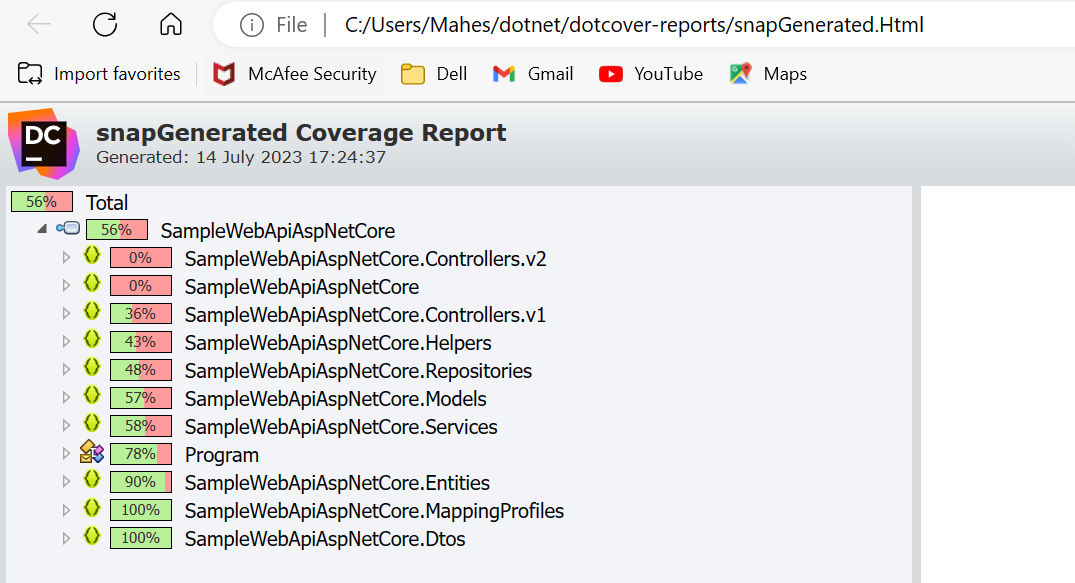
### Sample XML report:



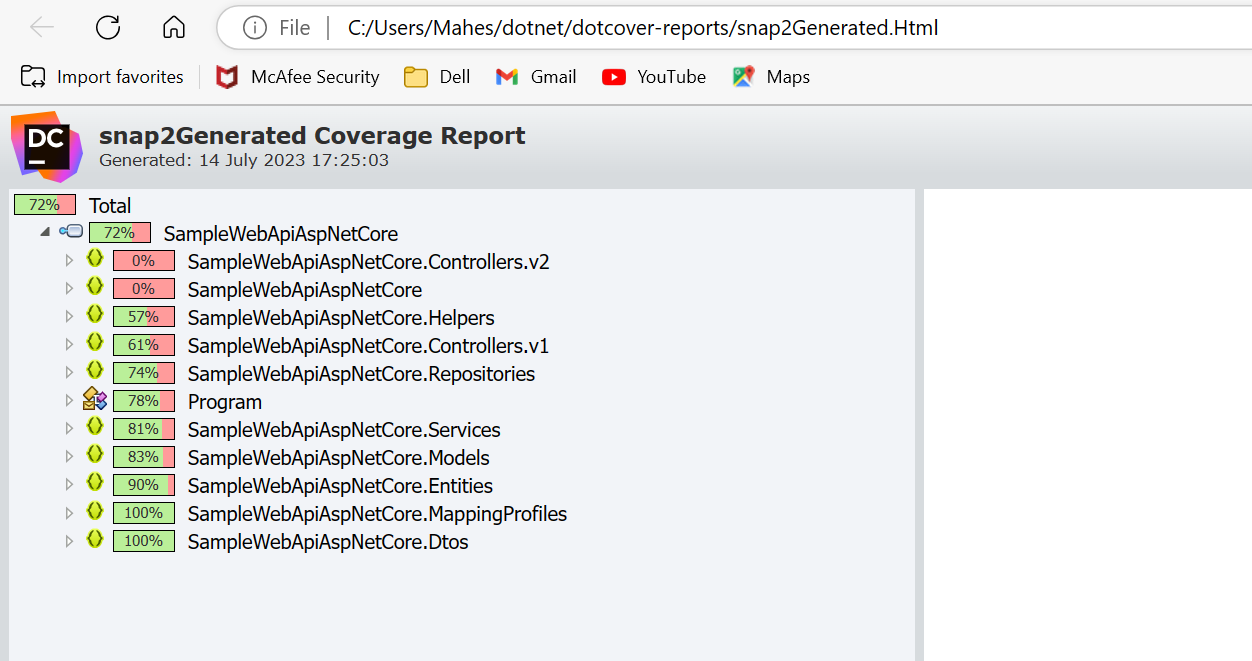
### Sample HTML Report:



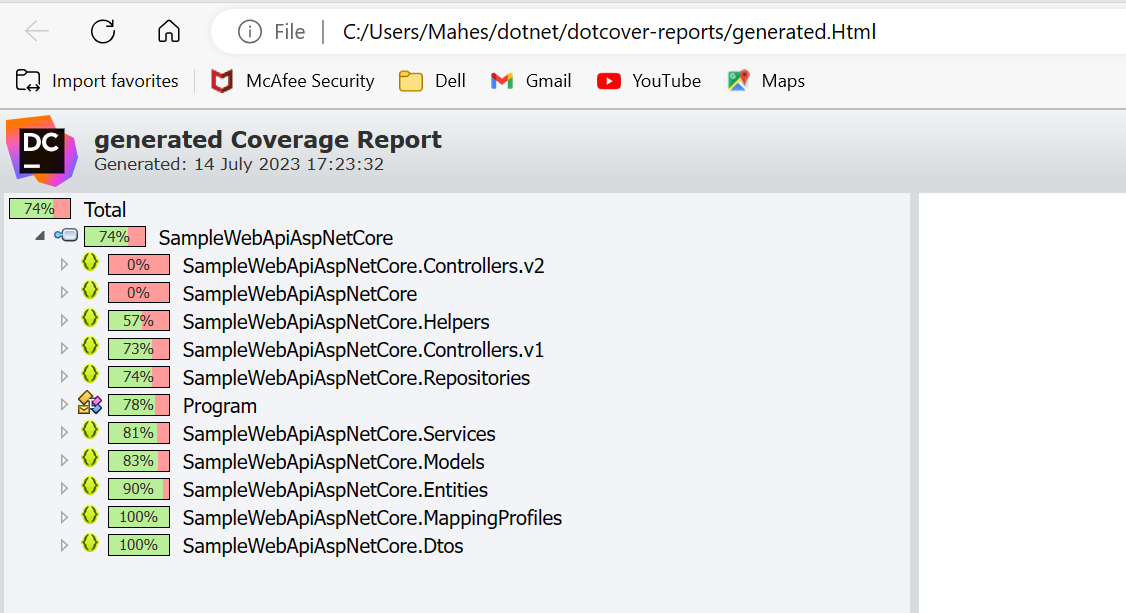
### Report with dcvr extension

Session 1: Generated HTML report from the dcvr snapshot file for session one 

Session 2: Generated HTML report from the dcvr snapshot file for session two



Combined report: Combined both dcvr files and generated HTML report from the combined.dcvr file



# 2. Dotcover coverage on Calculator app

## Step 1: Set up Project

Clone project from https://github.com/NuloSois/CalcMachine

Build the project with ‘dotnet build’ command.

## Step 2: Set Up dotCover

Download dependent dotcover tools.

Command line dotcover tool here: [https://www.jetbrains.com/dotcover/download/#section=commandline](https://www.jetbrains.com/dotcover/download/" \l "section=commandline)

Extract them into project folder where you want to run dotcover commands.

Or

store them in a folder & run from there by providing full path of app.exe or app.dll

Or

App path of the folder into System path and run the commands from anywhere

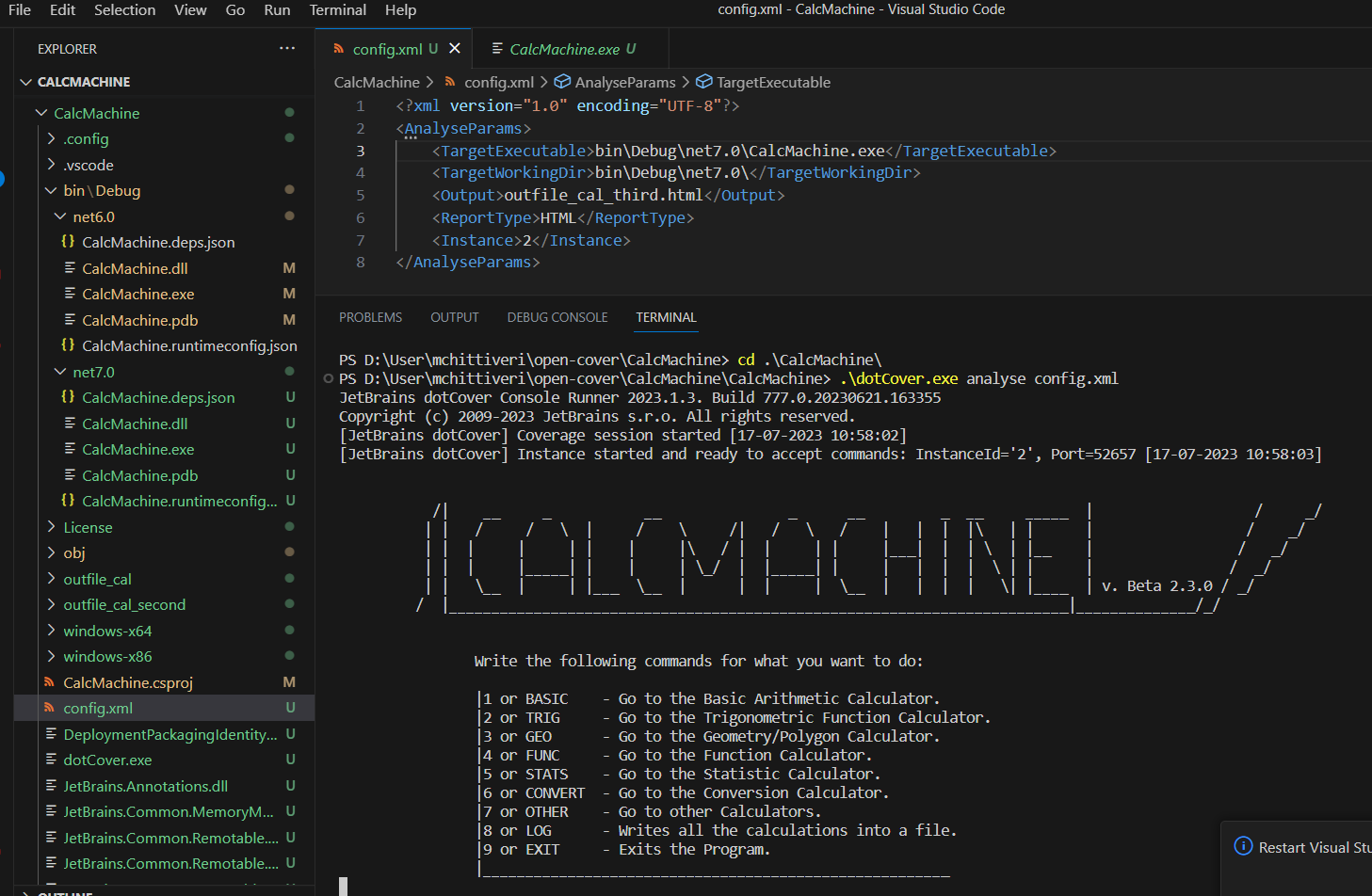
Run below two commands in the same dir:

*dotnet new tool-manifest*

*dotnet tool install JetBrains.dotCover.GlobalTool*

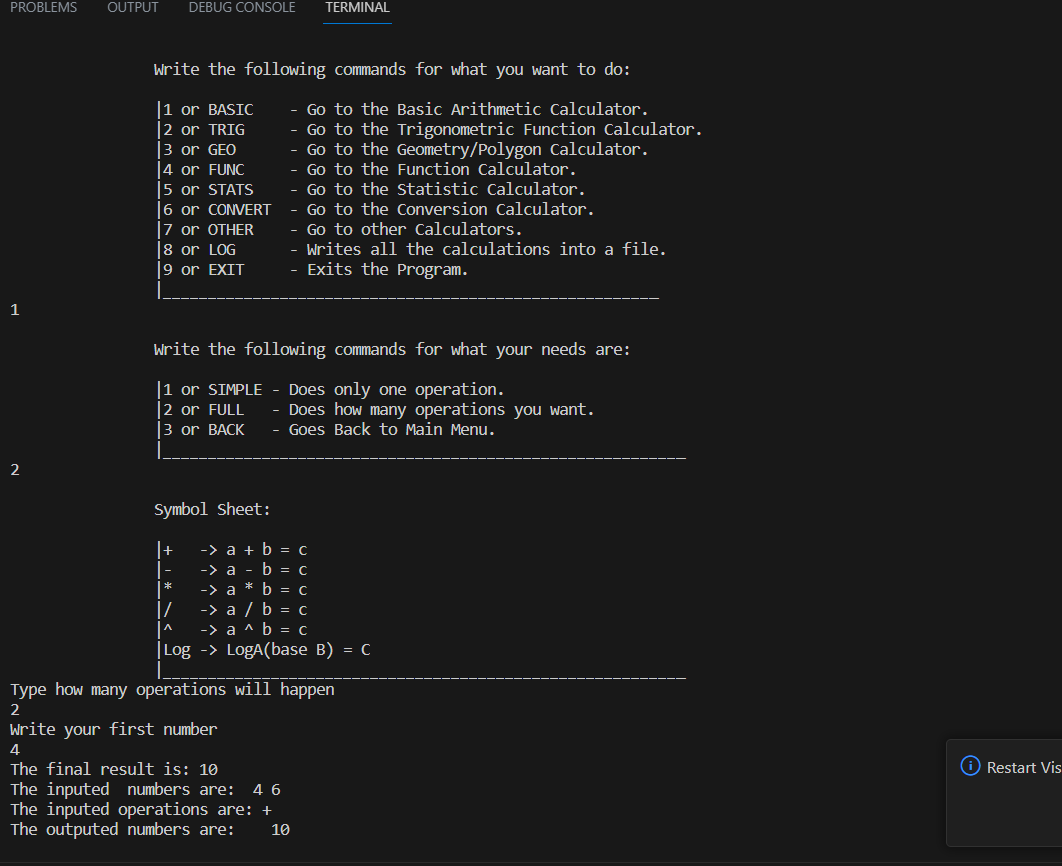
## Step 3: Start app with dotCover.exe

Using analyse command, Run app by integrating with dotCover.exe



## Step 4: Perform Testing

Choose from the list of options and provide necessary valid inputs to see the output.

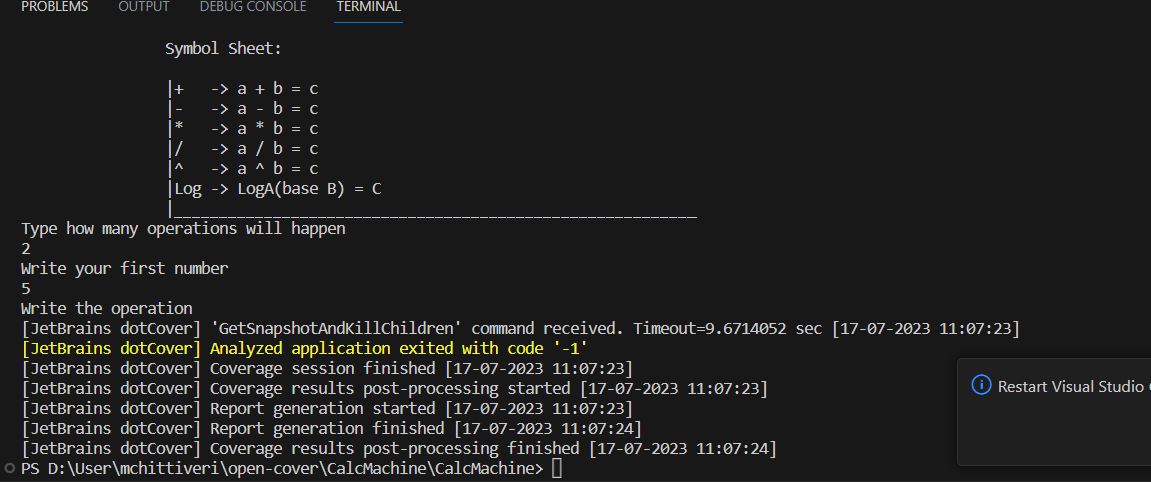


## Step 5: Stop the session

Stop the coverage session by using below command.

dotCover.exe send /Command=GetSnapshotAndKillChildren /Instance=1

Upon passing the command from a different terminal, user can observe the session being stopped.



## Step 6: Generating Reports and Merging Snapshots

Merge two snapshots using the below command.

dotCover.exe merge --Source=snap\_one.dcvr;snap\_two.dcvr --Output=combined.dcvr

Generate HTML report from the snapshot using the below command.

dotCover.exe report --Source=combined.dcvr --Output=generated.Html --ReportType=HTML

dotCover.exe report --Source=snap\_one.dcvr --Output=snapGenerated.Html --ReportType=HTML

dotCover.exe report --Source=snap\_two.dcvr --Output=snap2Generated.Html –ReportType=HTML

Refer first section for image references.

# 3. Dotcover coverage on IIS

## Step 1: Publish the project.

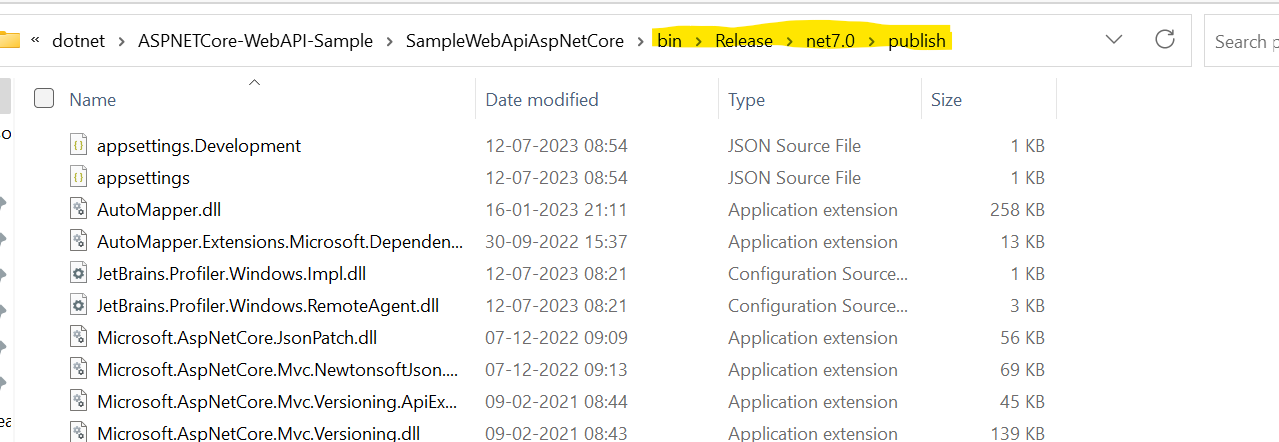
With the below command or from solution explorer, publish the project.

dotnet publish --configuration Release

Ref: <https://learn.microsoft.com/en-us/dotnet/core/tutorials/publishing-with-visual-studio-code?pivots=dotnet-7-0>

<https://www.gearhost.com/documentation/how-to-publish-your-app-from-visual-studio>

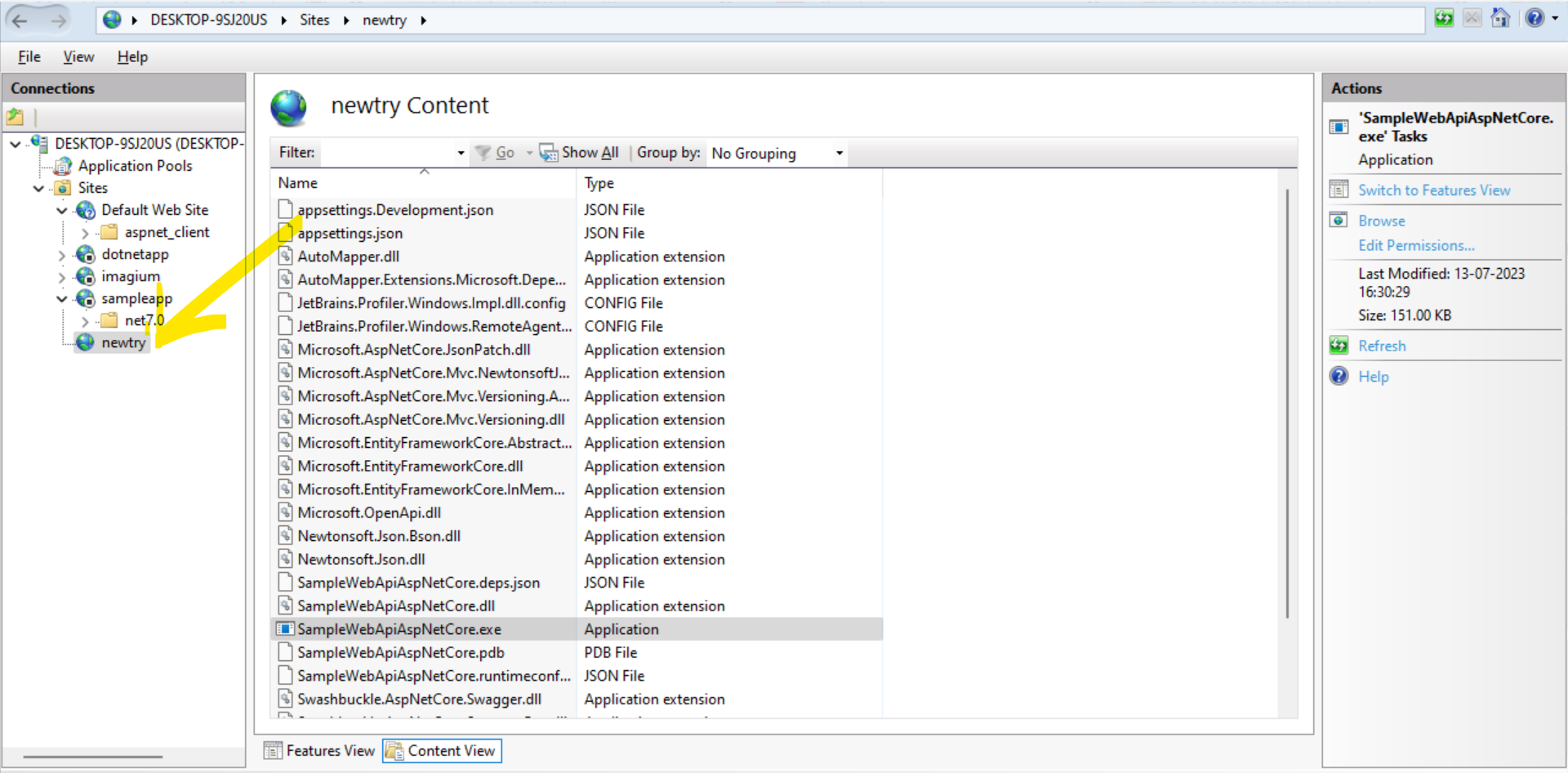
Sample Project: <https://github.com/FabianGosebrink/ASPNETCore-WebAPI-Sample>

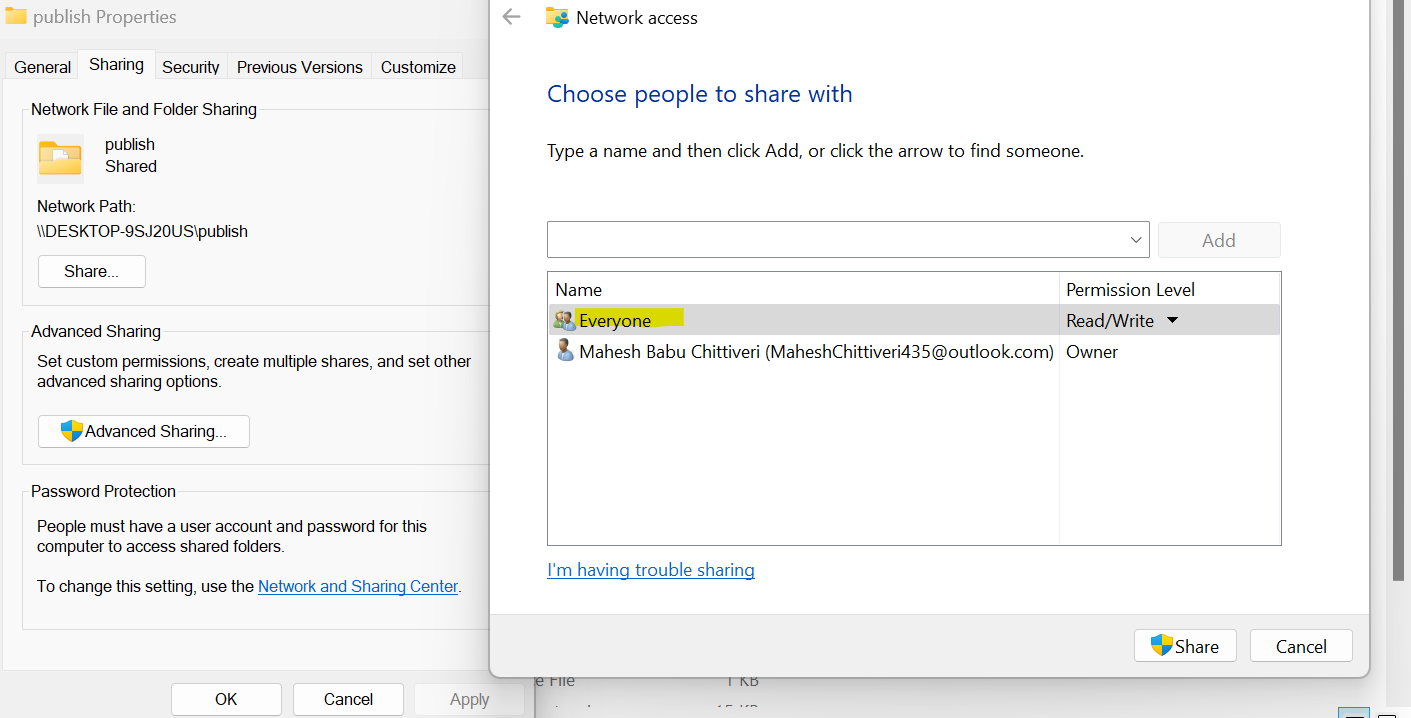


## Step 2: Set up in IIS

Copy the content to a different location to create a website in IIS or directly point the same folder.

Please make sure that folder has sharing rights.

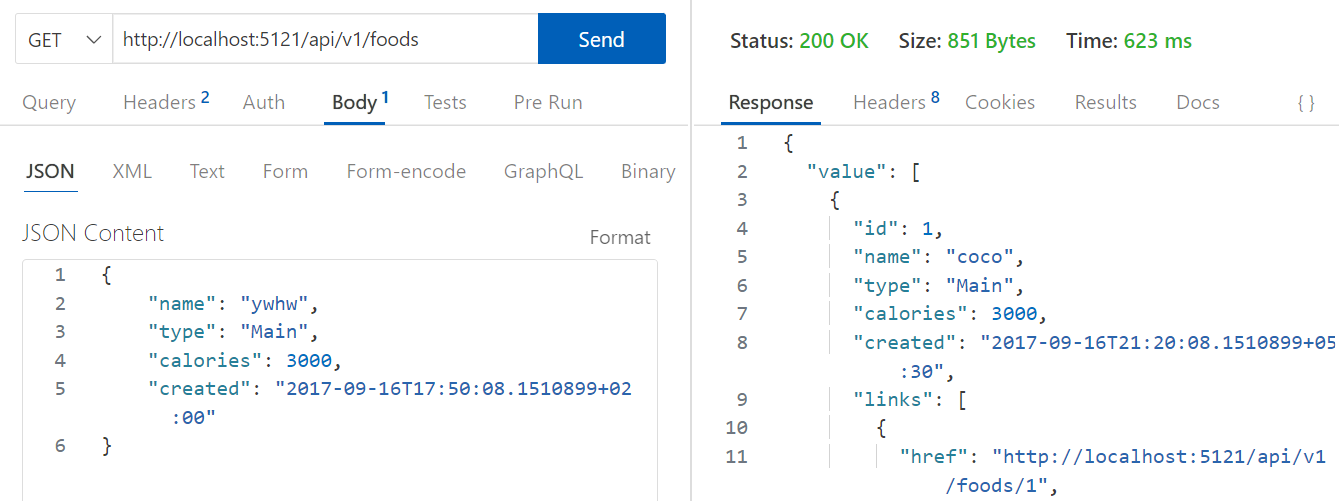


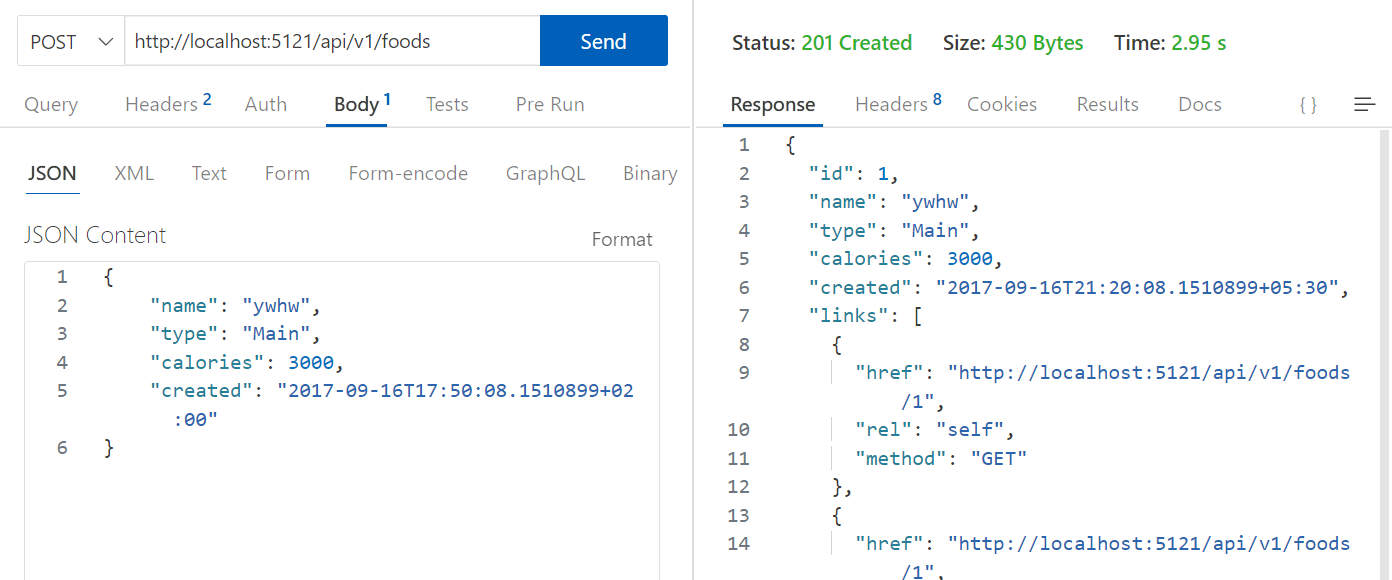


\*Without sharing, user will get 500 Internal server error.

## Step 3: Check app running status by hitting api or UI.

Hit the apis and observe the responses. Below are the sample screenshots of the same.

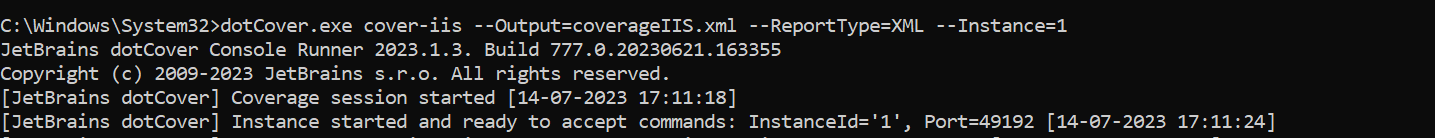




## Step 4: Start dotcover session with IIS

Once ensuring that app is running successfully, start dotcover coverage session with the below command. Inorder to run dotCover.exe commands from anywhere, user needs to add dotcover extracted directory to PATH.

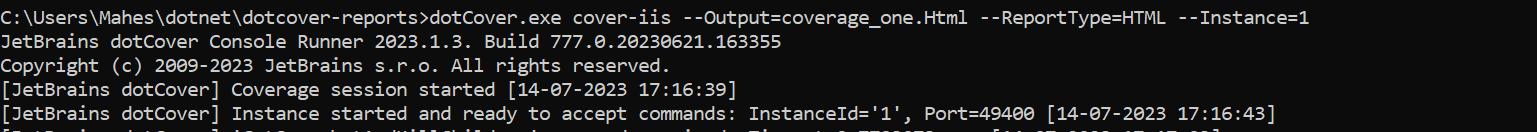
dotCover.exe cover-iis --Output=coverageIIS.xml --ReportType=XML --Instance=1



This will generate report in an xml file. Run below commands for HTML reports

dotCover.exe cover-iis --Output=coverage\_one.Html --ReportType=HTML --Instance=1

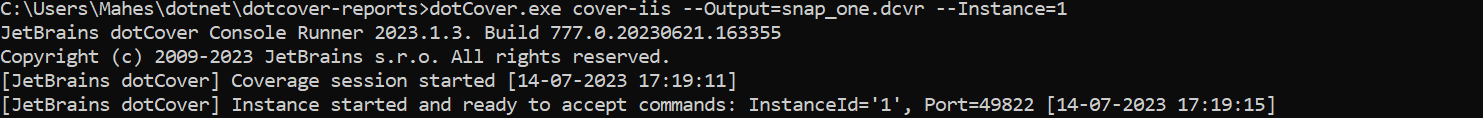
dotCover.exe cover-iis --Output=coverage\_two.Html --ReportType=HTML --Instance=1



Run below commands to generate coverage snapshots.

dotCover.exe cover-iis --Output=snap\_one.dcvr --Instance=1

dotCover.exe cover-iis --Output=snap\_two.dcvr --Instance=1



## Step 5: Perform Testing

Hit some api and check the responses. Refer first section for Image reference.

## Step 6: Stop the session

Stop the coverage session by using below command.

dotCover.exe send /Command=GetSnapshotAndKillChildren /Instance=1

Refer first section for image references.

## Step 7: Generating Reports and Merging Snapshots

Merge two snapshots using the below command.

dotCover.exe merge --Source=snap\_one.dcvr;snap\_two.dcvr --Output=combined.dcvr

Generate HTML report from the snapshot using the below command.

dotCover.exe report --Source=combined.dcvr --Output=generated.Html --ReportType=HTML

dotCover.exe report --Source=snap\_one.dcvr --Output=snapGenerated.Html --ReportType=HTML

dotCover.exe report --Source=snap\_two.dcvr --Output=snap2Generated.Html –ReportType=HTML

Refer first section for image references.