Instructions for Loading the Code into the Visual Studio 2015 IDE

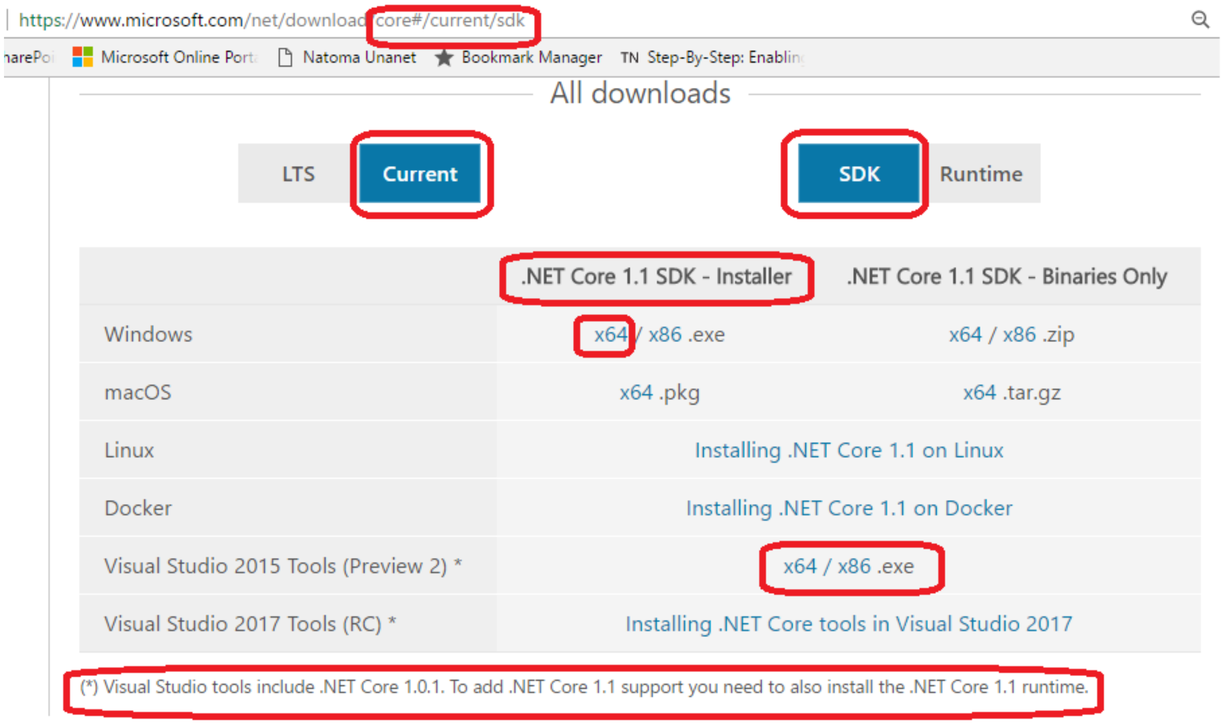
This document assumes that Visual Studio 2015 Community Update 3 has already been installed (Visual Studio 2015 Community Update 3).

Install the .NET Core 1.0.1 tools Preview 2 - https://go.microsoft.com/fwlink/?LinkID=827546

Install TypeScript Tools for Microsoft Visual Studio 2015 2.1.5.0. Installation file can be found here.

Install .NET Core SDK and Visual Studio 2015 Tools (https://www.microsoft.com/net/download/core). Notes on .NET Core SDK installation (see screenshot below):

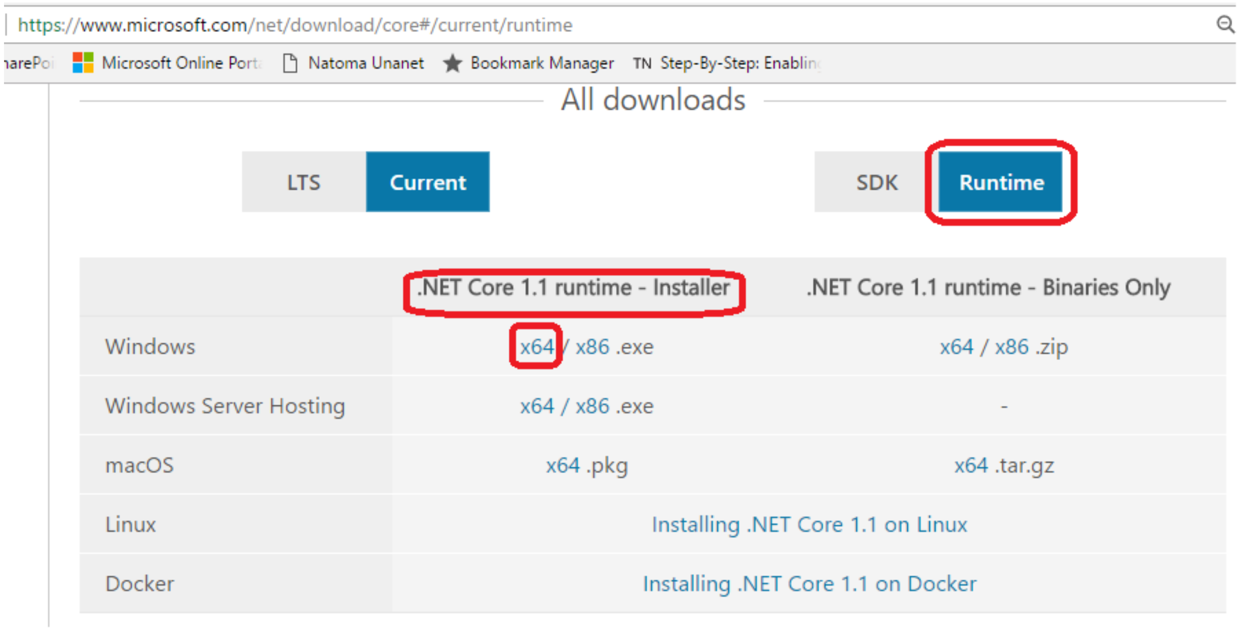
- Choose “Current” instead of the default “LTS”. This selects .NET Core 1.1 instead of 1.0.
- Confirm that “SDK” is chosen instead of “Runtime”.
- Install the SDK, choosing the x64 version of the SDK installer.
- Install Visual Studio 2015 Tools.



	.NET Core 1.1 SDK - Installer	.NET Core 1.1 SDK - Binaries Only
Windows	x64 / x86 .exe	x64 / x86 .zip
macOS	x64 .pkg	x64 .tar.gz
Linux	Installing .NET Core 1.1 on Linux	
Docker	Installing .NET Core 1.1 on Docker	
Visual Studio 2015 Tools (Preview 2) *	x64 / x86 .exe	
Visual Studio 2017 Tools (RC) *	Installing .NET Core tools in Visual Studio 2017	

(*) Visual Studio tools include .NET Core 1.0.1. To add .NET Core 1.1 support you need to also install the .NET Core 1.1 runtime.

As noted above, installation of Visual Studio Tools does not guarantee that .NET Core 1.1 runtime is also installed. .NET Core 1.1 runtime can be explicitly installed now. See screenshot below.



https://www.microsoft.com/net/download/core#/current/runtime

All downloads

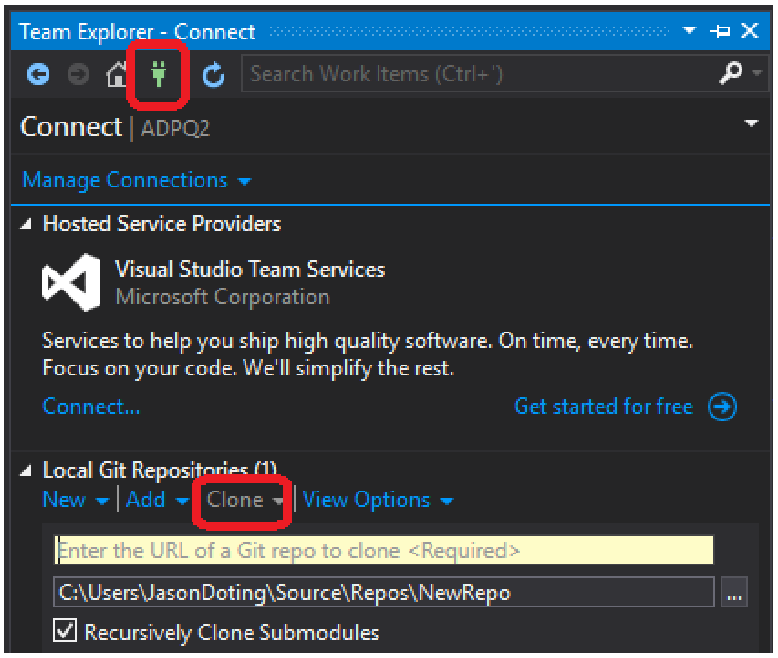
LTS | Current | SDK | Runtime

	.NET Core 1.1 runtime - Installer	.NET Core 1.1 runtime - Binaries Only
Windows	x64 / x86 .exe	x64 / x86 .zip
Windows Server Hosting	x64 / x86 .exe	-
macOS	x64 .pkg	x64 .tar.gz
Linux	Installing .NET Core 1.1 on Linux	
Docker	Installing .NET Core 1.1 on Docker	

Open Visual Studio 2015 as Administrator.

Get code from GitHub.

- Obtain the code repository URL (e.g. https://github.com/NatomaTechnologies/ADPQ2.git).
- Clone the Git repo (see screenshot below). Within Visual Studio, choose *View -> Team Explorer -> Connect (little green plug) -> Clone*. Copy the Git repo URL.
- Click the *Clone* button which appears after the URL is copied.



Team Explorer - Connect

Search Work Items (Ctrl+')

Connect | ADPQ2

Manage Connections

Hosted Service Providers

Visual Studio Team Services
Microsoft Corporation

Services to help you ship high quality software. On time, every time. Focus on your code. We'll simplify the rest.

Connect... Get started for free

Local Git Repositories (1)

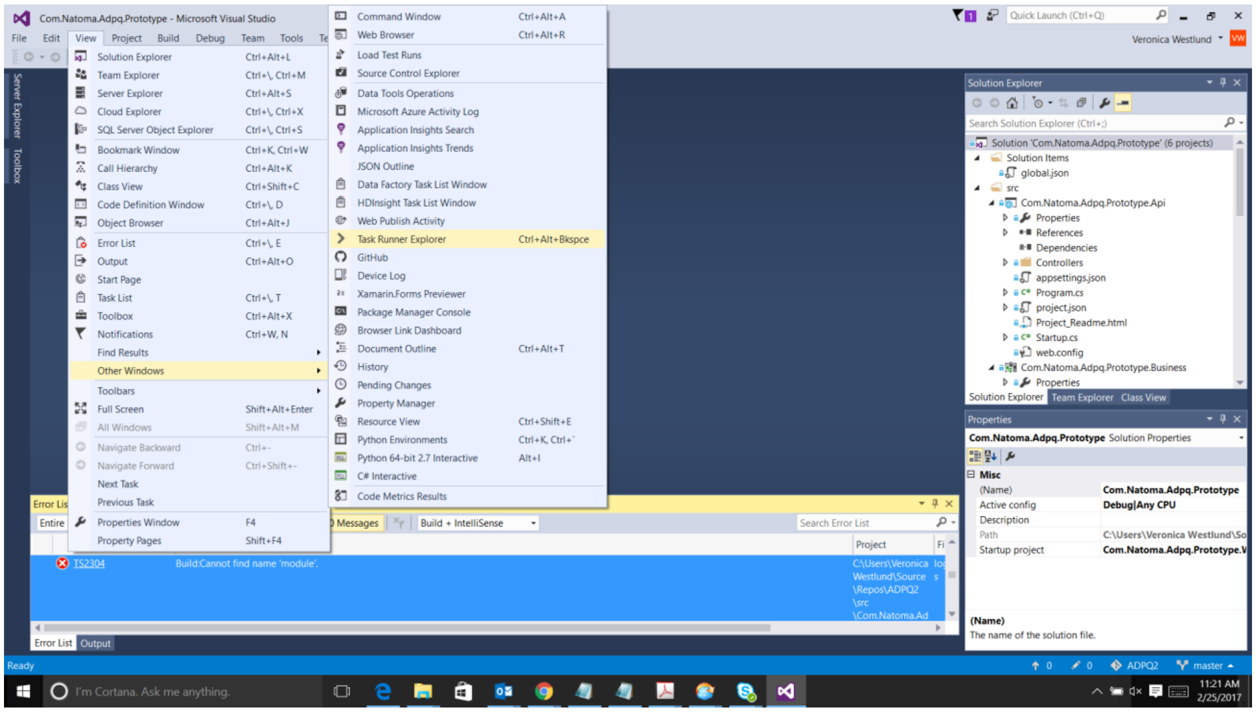
New | Add | Clone | View Options

Enter the URL of a Git repo to clone <Required>

C:\Users\JasonDoting\Source\Repos\NewRepo

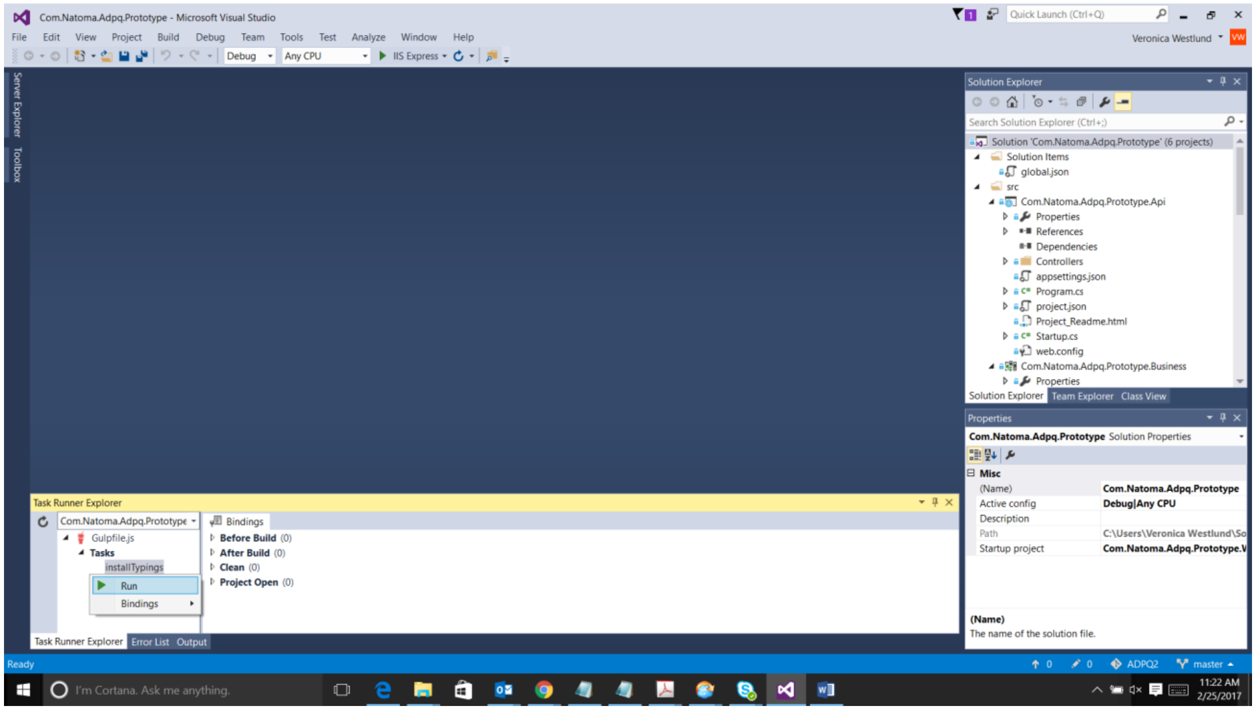
☑ Recursively Clone Submodules

Within Visual Studio, Choose *View -> Other Windows -> Task Runner Explorer*.

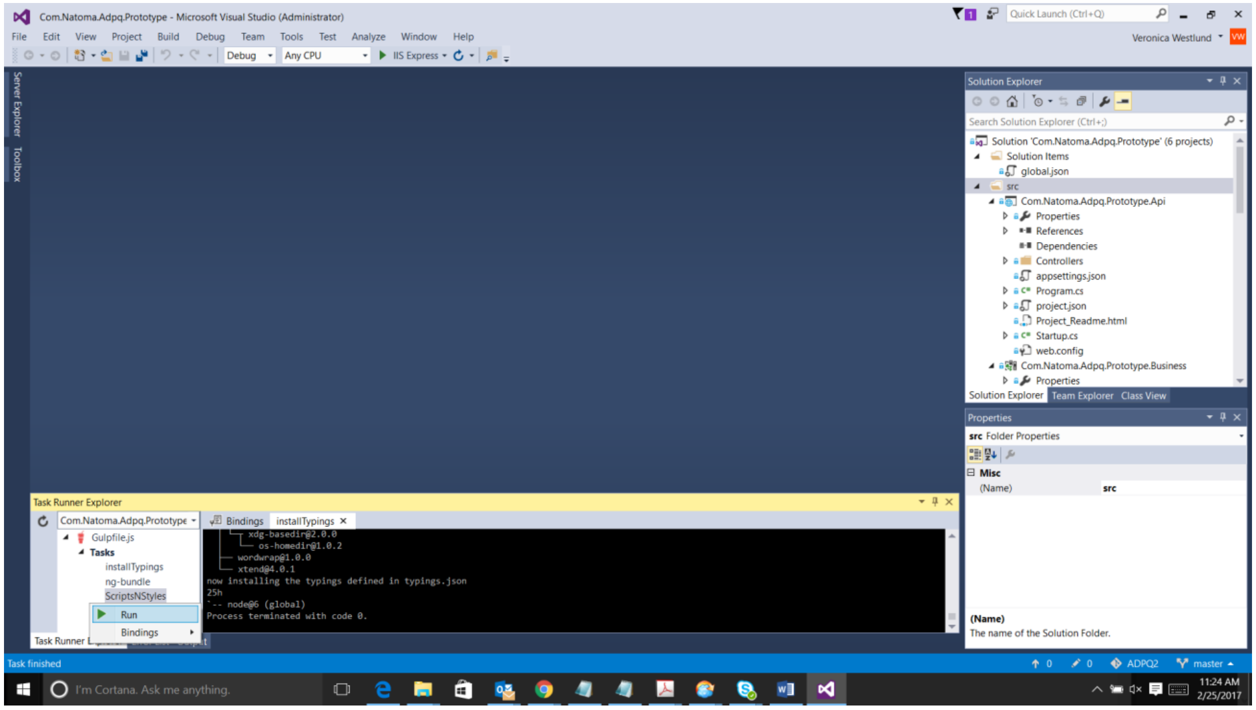

Com.Natoma.Adpq.Prototype - Microsoft Visual Studio
View
Other Windows
Task Runner Explorer Ctrl+Alt+Bkspce
Solution Explorer
Error List
Output

TS2304 Build:Cannot find name 'module'.

In *Task Runner Explorer*, Run *Install Typings* Gulp task.



In *Task Runner Explorer*, Run *ScriptNStyles* task.

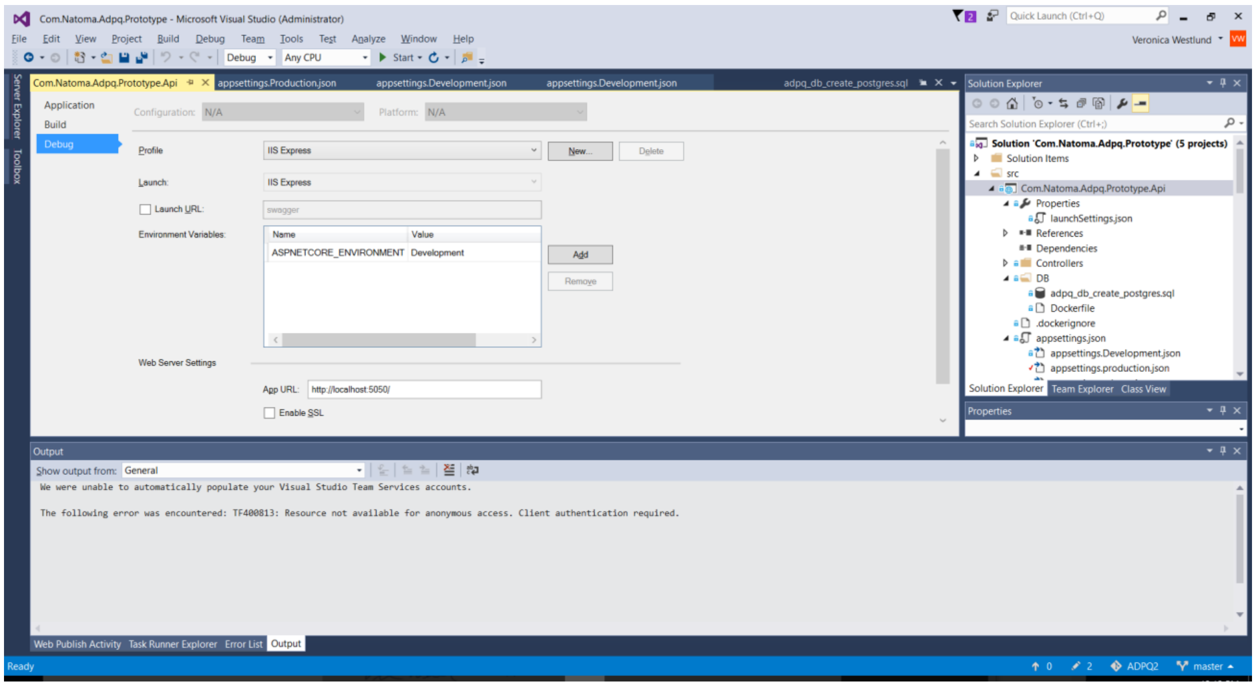

Com.Natoma.Adpq.Prototype - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug Any CPU IIS Express
Solution Explorer
Solution 'Com.Natoma.Adpq.Prototype' (6 projects)
Solution Items
global.json
src
Com.Natoma.Adpq.Prototype.Api
Properties
References
Dependencies
Controllers
appsettings.json
Program.cs
project.json
Project_Readme.html
Startup.cs
web.config
Com.Natoma.Adpq.Prototype.Business
Properties
Task Runner Explorer
Com.Natoma.Adpq.Prototype
Gulpfile.js
Tasks
installTypings
ng-bundle
ScriptNStyles
Run
Bindings
installTypings
Task finished

```
    ┌ xdg-basedir@2.0.0
     └─ os-homedir@1.0.2
  ├─ wordwrap@1.0.0
  └─ xtend@4.0.1
now installing the typings defined in typings.json
25h
`-- node@6 (global)
Process terminated with code 0.
```

Attempt to rebuild the solution. A successful rebuild indicates setup has been performed per expectations.

To run it locally you will need a Postgres database. You can build to a database image or pull and run the postgresdb.release docker image from the dockerhub respository (. https://hub.docker.com/r/natoma/adpq2) or you can run the adpq_db_create_postgres.sql script in the DB folder of the Com.Natoma.Adpq.Prototype.Api project on your local Postgres database. Change the appsettings.Development.json file to point to the correct database.

To run the application you can confirm that the Properties for Com.Natoma.Adpq.Prototype.Api are as follows, if not you can change them:


Com.Natoma.Adpq.Prototype - Microsoft Visual Studio (Administrator)
Com.Natoma.Adpq.Prototype.Api
appsettings.Production.json
appsettings.Development.json
adpq_db_create_postgres.sql
Application
Build
Debug
Solution Explorer
Solution 'Com.Natoma.Adpq.Prototype' (5 projects)
Output

Configuration: N/A Platform: N/A

Profile: IIS Express

Launch: IIS Express

☐ Launch URL: swagger

Environment Variables:

Name	Value
ASPNETCORE_ENVIRONMENT	Development

Web Server Settings

App URL: http://localhost:5050/

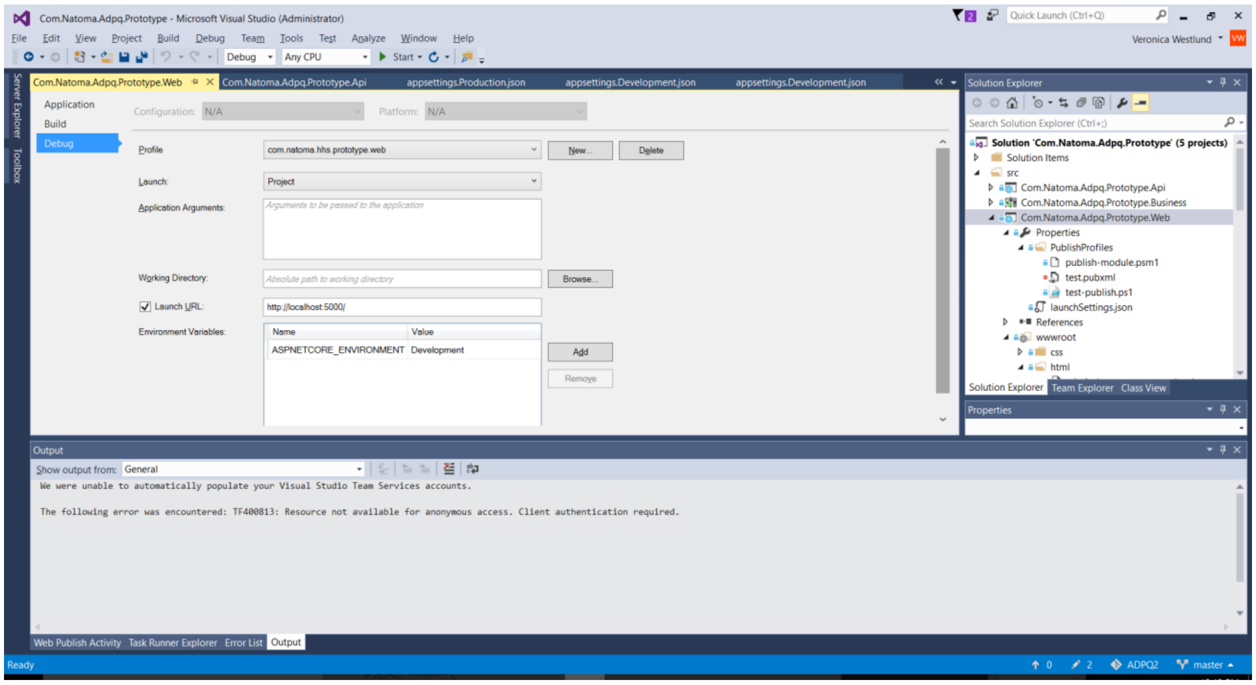
☐ Enable SSL

Show output from: General

```
We were unable to automatically populate your Visual Studio Team Services accounts.

The following error was encountered: TF400813: Resource not available for anonymous access. Client authentication required.
```

You can also confirm the properties of the Com.Natoma.Adpq.Prototype.Web project are as follows:


Com.Natoma.Adpq.Prototype - Microsoft Visual Studio (Administrator)
Com.Natoma.Adpq.Prototype.Web
Com.Natoma.Adpq.Prototype.Api
Application
Build
Debug
Solution Explorer
Solution 'Com.Natoma.Adpq.Prototype' (5 projects)
Com.Natoma.Adpq.Prototype.Business
Output

Configuration: N/A Platform: N/A

Profile: com.natoma.hhs.prototype.web

Launch: Project

Application Arguments: Arguments to be passed to the application

Working Directory: Absolute path to working directory

☑ Launch URL: http://localhost:5000/

Environment Variables:

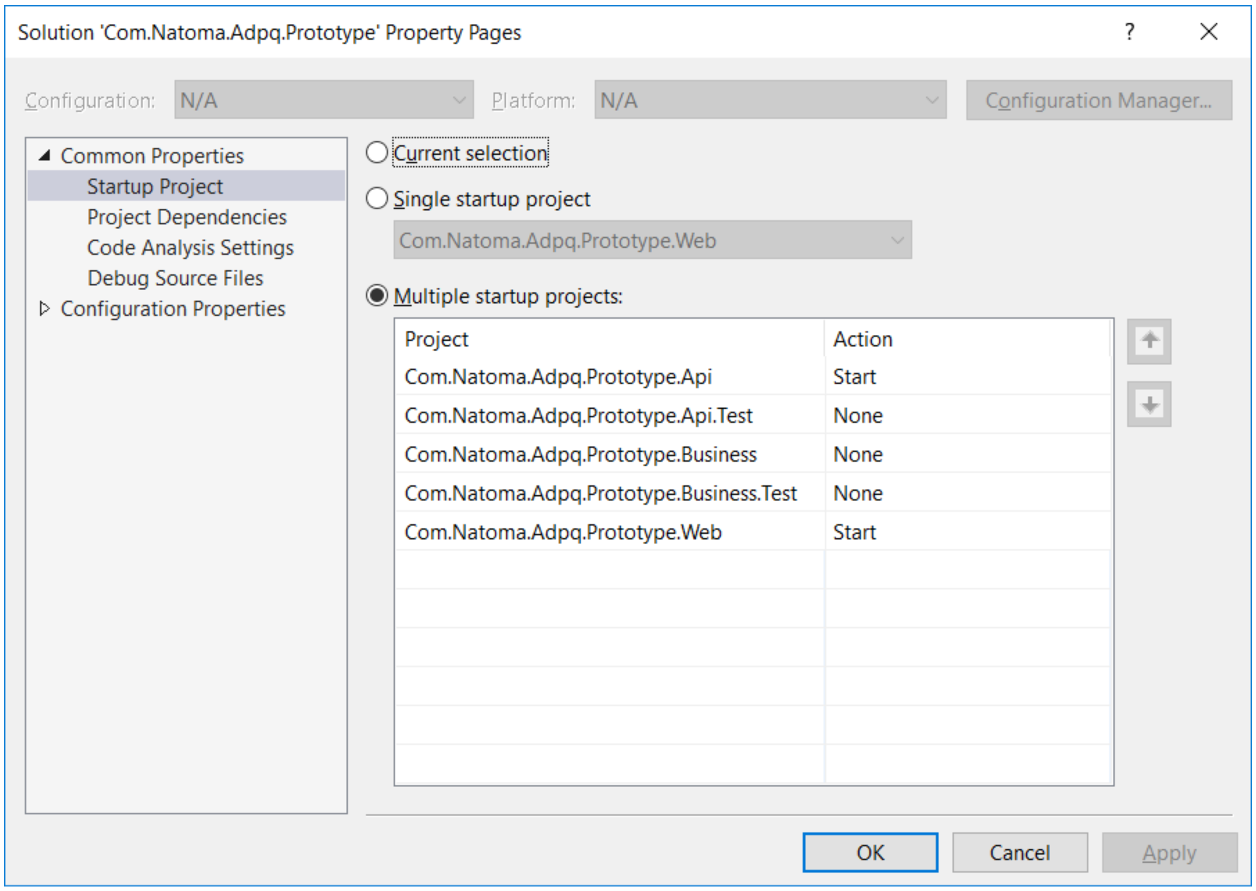
Name	Value
ASPNETCORE_ENVIRONMENT	Development

Show output from: General

```
We were unable to automatically populate your Visual Studio Team Services accounts.

The following error was encountered: TF400813: Resource not available for anonymous access. Client authentication required.
```

You must also make sure the solution is set up to start both projects (click properties on the solution). And set to Multiple startup projects.



Solution 'Com.Natoma.Adpq.Prototype' Property Pages

Configuration: N/A   Platform: N/A   Configuration Manager...

- Common Properties
  - Startup Project
  - Project Dependencies
  - Code Analysis Settings
  - Debug Source Files
- Configuration Properties

( ) Current selection
( ) Single startup project: Com.Natoma.Adpq.Prototype.Web
[● Multiple startup projects:]

Project	Action
Com.Natoma.Adpq.Prototype.Api	Start
Com.Natoma.Adpq.Prototype.Api.Test	None
Com.Natoma.Adpq.Prototype.Business	None
Com.Natoma.Adpq.Prototype.Business.Test	None
Com.Natoma.Adpq.Prototype.Web	Start

Click OK

You can now run the project locally.