ADPQ Installation Instructions

1. Install Visual Studio

2. Install Docker

3. Run Docker Commands to install ADPQ

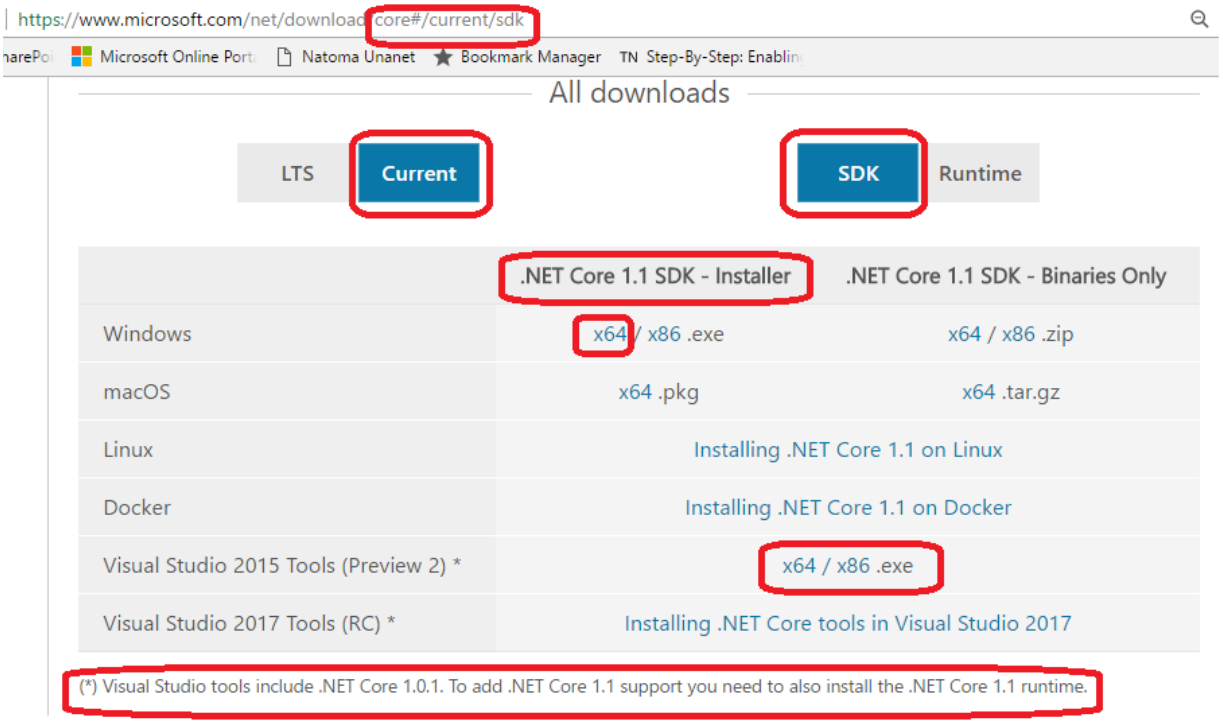
Visual Studio Installation

This document assumes that Visual Studio 2015 Professional Update 3 has already been installed.

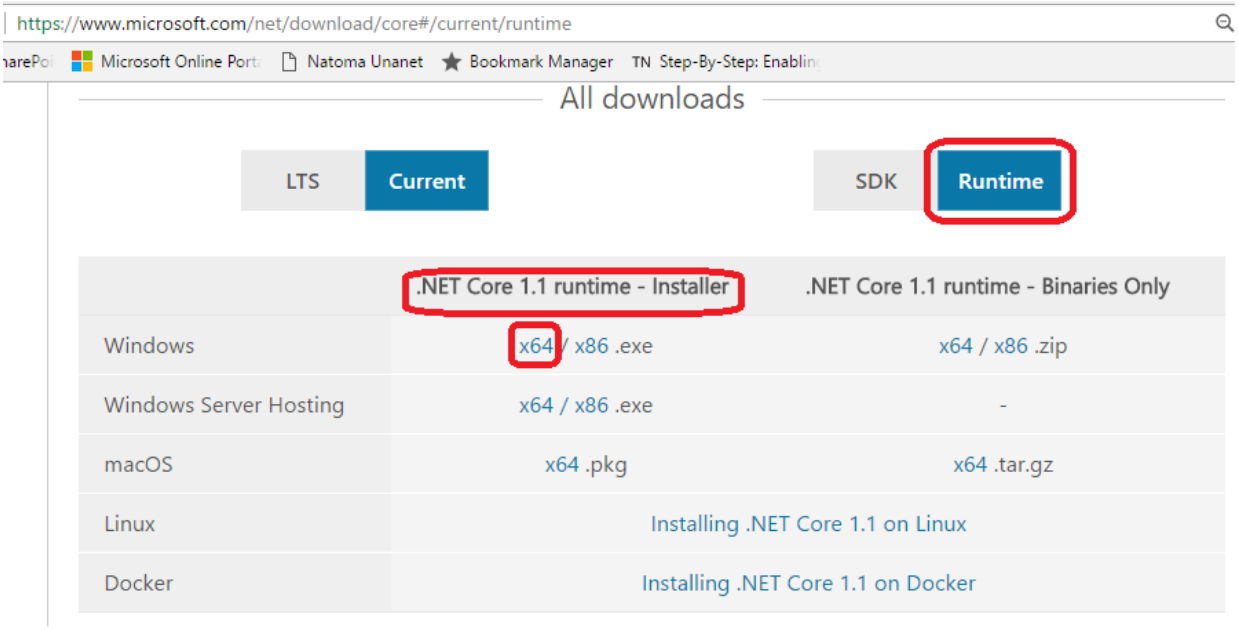
Install TypeScript Tools for Microsoft Visual Studio 2015 2.1.5.0. Installation file can be found here.

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.



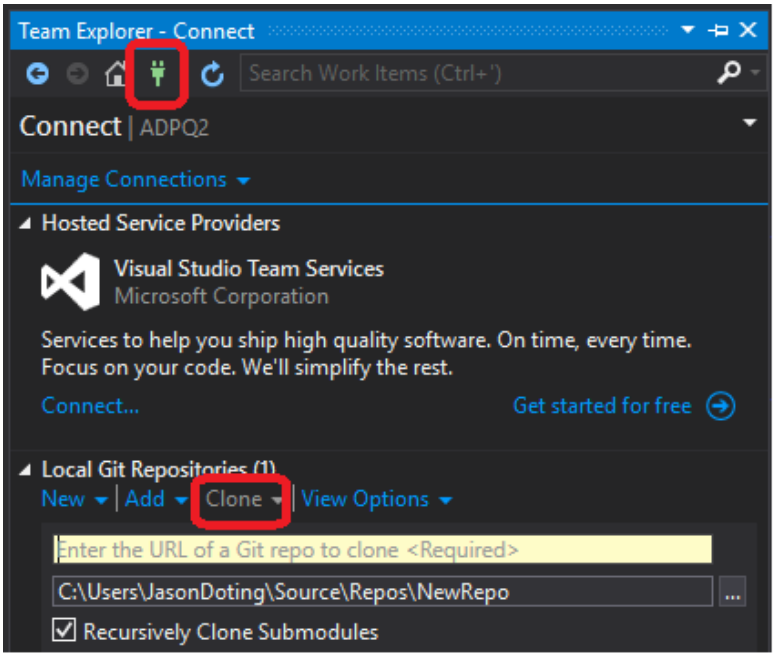
As noted above, installation of Visual Studio Tools does not guarantee that .NET Core 1.1 runtime is also installed. .NET Core 1.1 runtime can be explicitly installed now. See screenshot below.



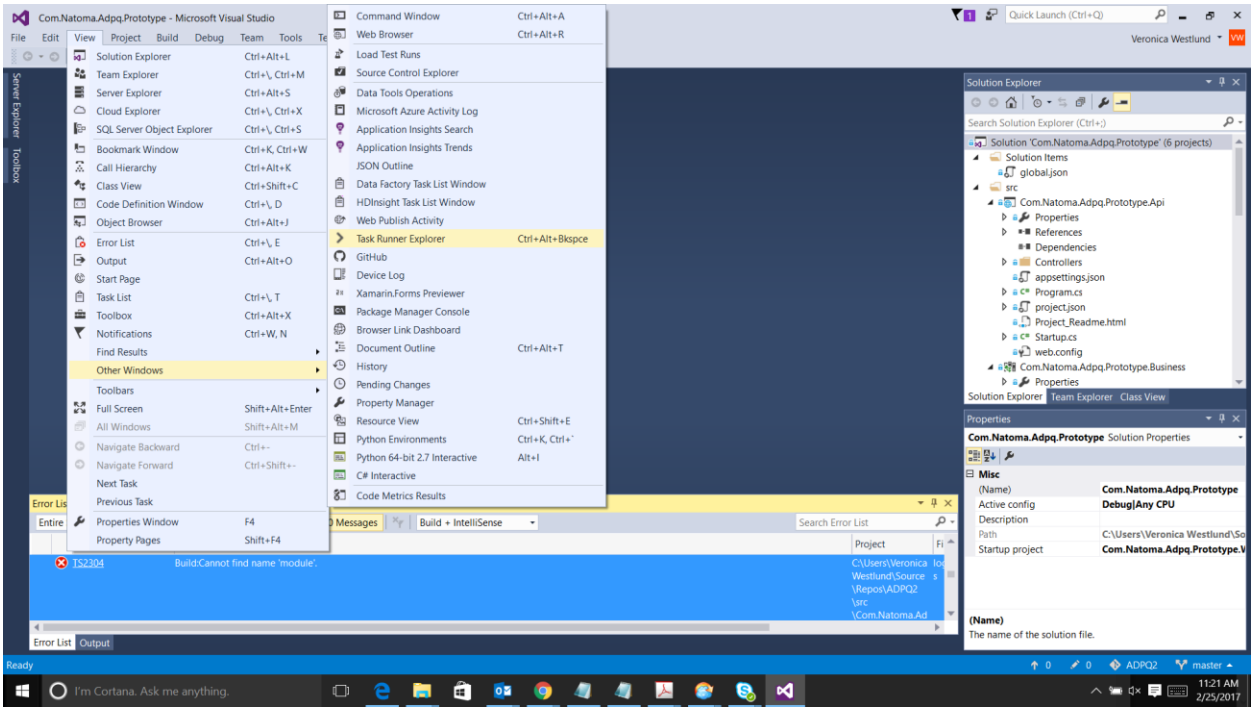
Open Visual Studio 2015 as Administrator.

Get code from GitHub.

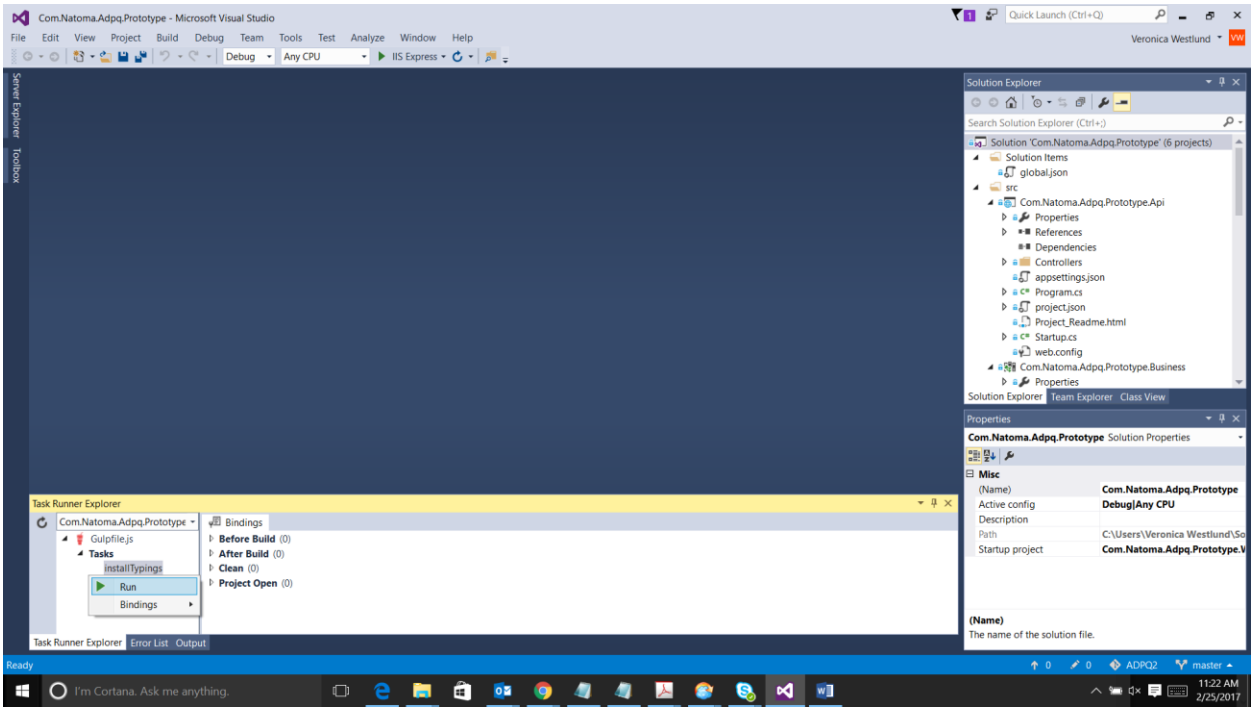
- Obtain the code repository URL (e.g. https://github.com/NatomaTechnologies/ADPQ2.git).
- Clone the Git repo (see screenshot below). Within Visual Studio, choose *View -> Team Explorer -> Connect (little green plug) -> Clone*. Copy the Git repo URL.
- Click the *Clone* button which appears after the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



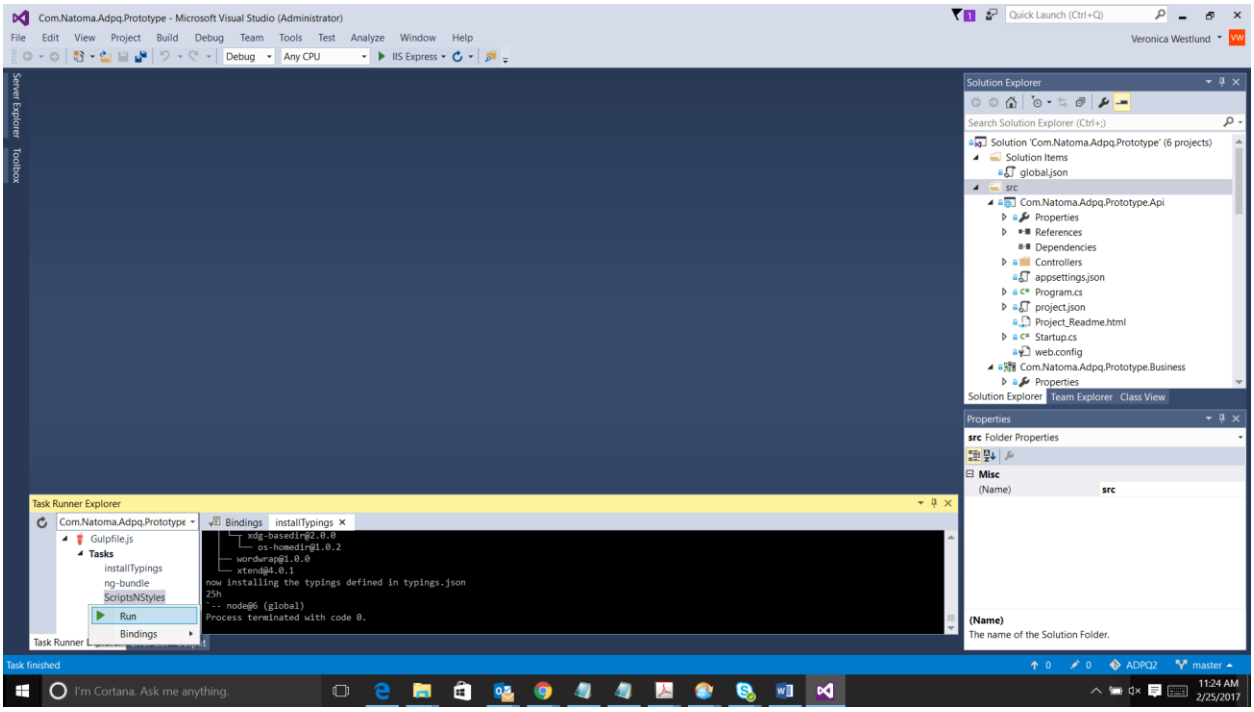
Within Visual Studio, Choose *View -> Other Windows -> Task Runner Explorer*.



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



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



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

Docker Installation

Install “Docker for Windows”. Link can be found here. Click on the “Install Docker for Windows” link (see screenshot below).

After clicking on the link above, choose the “Stable channel” (not the “Edge channel”). Then execute the downloaded InstallDocker.msi file.

Install ADPQ Through Docker Commands

After Docker is installed, open PowerShell and execute the following commands. These commands pull the latest Docker images, stop any running images, remove any existing images and then run the latest images. Finally, logs are pulled for the latest running images. If any of these commands time out, the command should be executed again.

```
docker pull natoma/adpq2:web.release

docker pull natoma/adpq2:api.release

docker pull natoma/adpq2:postgresdb.release
```

```
docker stop adpq2prodweb

docker stop adpq2prodapi

docker stop adpq2proddb

docker rm adpq2prodweb

docker rm adpq2prodapi

docker rm adpq2proddb
```

```
docker run -d -p 5432:5432 --name adpq2proddb natoma/adpq2:postgresdb.release

docker run -d -p 5050:5050 --name adpq2prodapi natoma/adpq2:api.release

docker run -d -p 5000:5000 --name adpq2prodweb natoma/adpq2:web.release
```

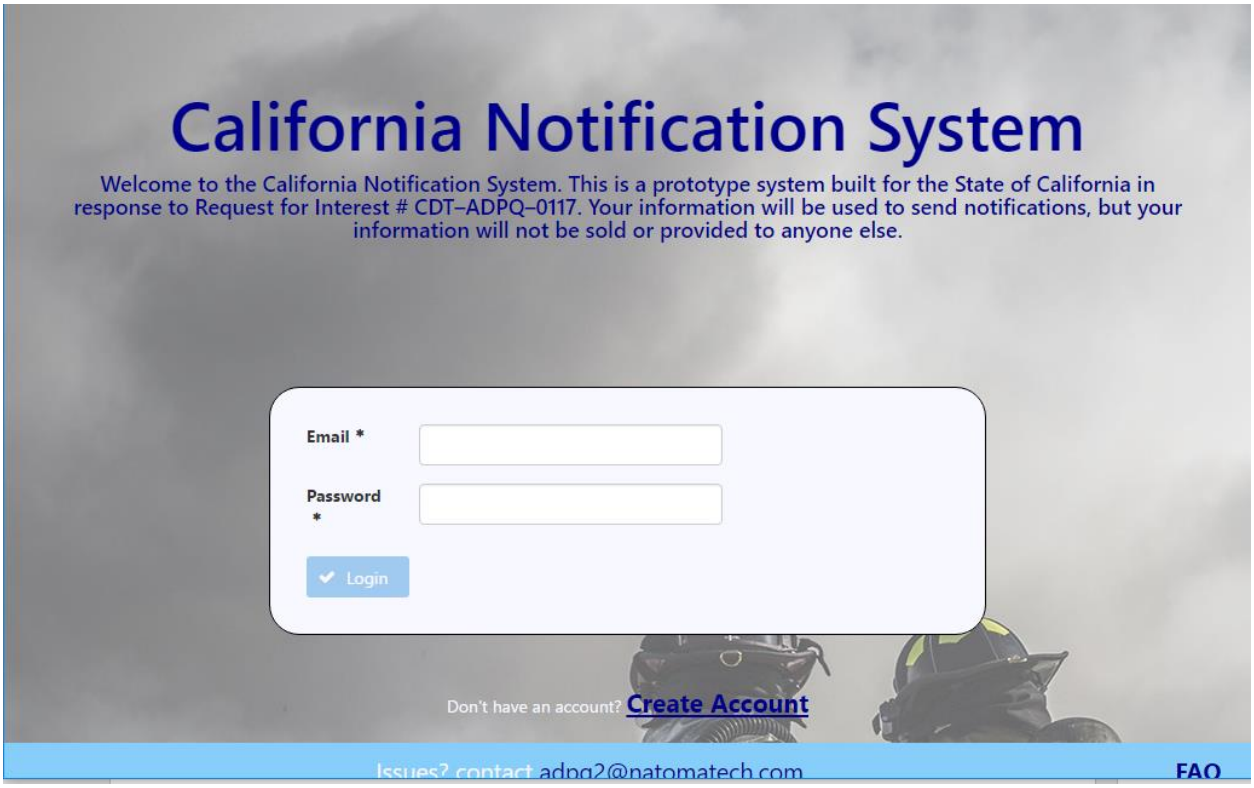
```
docker logs adpq2prodweb

docker logs adpq2prodapi

docker logs adpq2proddb
```

At this point, the Web server and the Web API server and the Database server should all be ready to test. To validate this, open a browser and key the following URL into the address bar.

http://localhost:5000. If the following is displayed, the Web server is responding.



Additional information:

The default setup assumes that the Web Server Docker container and the Web API server Docker container are both running on the same local machine (i.e. localhost) and on the default configured ports (i.e., 5000 and 5050). If this assumption is not true, config files must be changed on one or both Docker containers.