# Setup Angular2 Project in Visual Studio

The steps for setting up a project with an ASP.NET 4.x project in Visual Studio 2015 are as follows:

**Prerequisite:**

* Install Node.js
* Set up Visual Studio for TypeScript

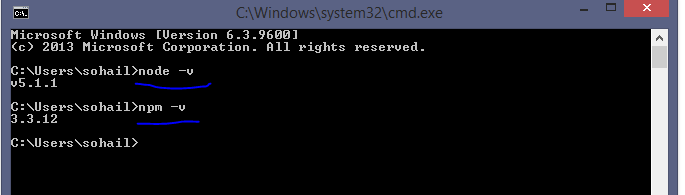
**Steps**

* Create the Visual Studio ASP.NET project
* Create configuration files in project root folder package.json, tsconfig.json, typings.json, system.config.js
* Install required packages
* Create your application (App module)
* Create a Component and add to your application
* Create a bootstrap for your application
* Create the web page that hosts the application
* Edit the TypeScript configuration file
* Build and run the app

# Install Node.js

Download and install node.js from (<https://nodejs.org/en/>) if not already present in your system.

**Verify that you are running node version** 4.4.x **-** 5.x.x**, and npm** 3.x.x by running node -v and npm -v in a terminal/console window. Older versions produce errors.



# Set up Visual Studio for TypeScript

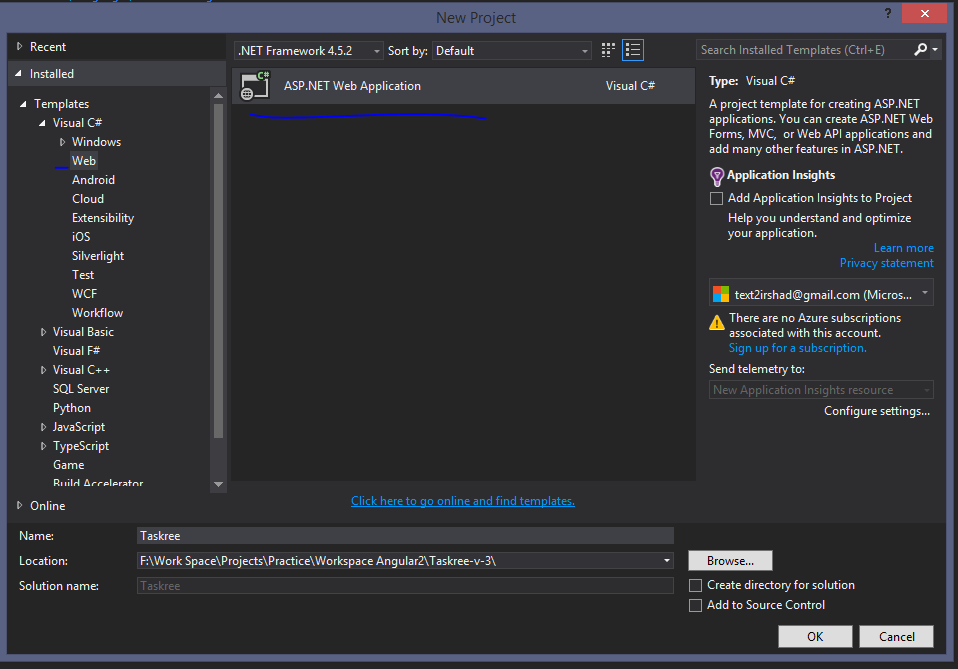
 Open Visual Studio and install the latest set of TypeScript tools as follows:

* Open Tools | Extensions and Updates.
* Select Online in the tree on the left.
* Search for TypeScript using the search box in the upper right.
* Select the most current available TypeScript version.
* Download and install the package.

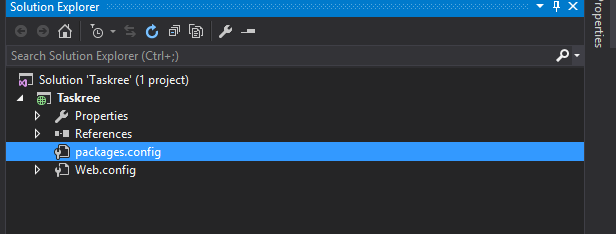
Now tell the visual studio to use external tool (such as npm) found in your path before using its own version of the external tools.

* Tools | Options to open the Options dialog.
* In the tree on the left, select Projects and Solutions | External Web Tools.
* On the right, move the $(PATH) entry above the $(DevEnvDir) entries. This tells Visual Studio to use the external tools (such as npm) found in your path before using its own version of the external tools.
* Click OK to close the dialog.
* Restart Visual Studio for this change to take effect.

# Create the Visual Studio ASP.NET project



The project structure should be like below:



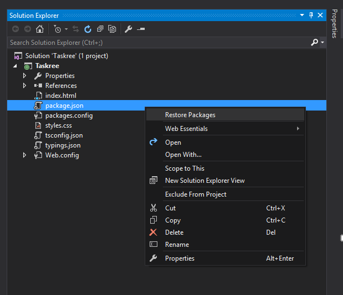
# Create configuration files in project root folder package.json, tsconfig.json, typings.json, system.config.js



# Install required packages

Restore the packages required for an Angular application as follows:

* Right-click on the package.json file in Solution Explorer and select Restore Packages.   
  This uses npm to install all of the packages defined in the package.json file. It may take some time.
* If desired, open the Output window (View | Output) to watch the npm commands execute.
* Ignore the warnings.
* When the restore is finished, a message should say: npm command completed with exit code 0.
* Click the Refresh icon in Solution Explorer.
* **Do not** include the node\_modules folder in the project. Let it be a hidden project folder.



# Create your application (App module)

You compose Angular applications into closely related blocks of functionality with [NgModules](https://angular.io/docs/ts/latest/guide/ngmodule.html). Angular itself is split into separate Angular Modules. This makes it possible for you to keep payload size small by only importing the parts of Angular that your application needs.

Every Angular application has at least one module: the root module, named AppModule here.

**Create an app subfolder** off the project root directory:



# Create a Component and add to your application

Every Angular application has at least one component: the root component, named AppComponent here.

Components are the basic building blocks of Angular applications. A component controls a portion of the screen—a view—through its associated template.

**Create the component file** app/app.component.ts with the following content:

# 

# Create a bootstrap for your application

Now you need to tell Angular to start up your application.

Create the file app/main.ts with the following content:



# Create the web page that hosts the application

In the project root folder, create an index.html file and paste the following lines into it:



# Build and run the app

To set index.html as start page, right-click index.html in Solution Explorer and select option Set as Startup Page.

Click the Run button or press F5 to build and run the application.

This launches the default browser and runs the QuickStart sample application.

Try editing any of the project files. Save and refresh the browser to see the changes.