**Step to make Angular 2 Hello world application:**

**Step 1: create package.json**

{

    "name": "angular-quickstart",

    "version": "1.0.0",

    "scripts": {

        "start": "tsc&& concurrently \"tsc -w\" \"lite-server\" ",

        "lite": "lite-server",

        "postinstall": "typings install",

        "tsc": "tsc",

        "tsc:w": "tsc -w",

        "typings": "typings"

    },

    "licenses": [{

        "type": "MIT",

        "url": "https://github.com/angular/angular.io/blob/master/LICENSE"

    }],

    "dependencies": {

        "@angular/common": "~2.1.0",

        "@angular/compiler": "~2.1.0",

        "@angular/core": "~2.1.0",

        "@angular/forms": "~2.1.0",

        "@angular/http": "~2.1.0",

        "@angular/platform-browser": "~2.1.0",

        "@angular/platform-browser-dynamic": "~2.1.0",

        "@angular/router": "~3.1.0",

        "@angular/upgrade": "~2.1.0",

        "angular-in-memory-web-api": "~0.1.5",

        "bootstrap": "^3.3.7",

        "core-js": "^2.4.1",

        "reflect-metadata": "^0.1.8",

        "rxjs": "5.0.0-beta.12",

        "systemjs": "0.19.39",

        "zone.js": "^0.6.25"

    },

    "devDependencies": {

        "concurrently": "^3.0.0",

        "lite-server": "^2.2.2",

        "typescript": "^2.0.3",

        "typings": "^1.4.0"

    }

}

* [tsconfig.json](https://angular.io/docs/ts/latest/guide/typescript-configuration.html#tsconfig)—TypeScript compiler configuration.
* [typings](https://angular.io/docs/ts/latest/guide/typescript-configuration.html#typings)—TypesScript declaration files.

**Step 2: Create tsconfig.json**

{

    "compilerOptions": {

        "target": "es5",

        "module": "commonjs",

        "moduleResolution": "node",

        "sourceMap": true,

        "emitDecoratorMetadata": true,

        "experimentalDecorators": true,

        "removeComments": false,

        "noImplicitAny": false,

        "watch": true

    } ,

    "compileOnSave": true

}

**Step 3: Create typings.json**

{

    "globalDependencies": {

        "core-js": "registry:dt/core-js#0.0.0+20160725163759",

        "jasmine": "registry:dt/jasmine#2.2.0+20160621224255",

        "node": "registry:dt/node#6.0.0+20160909174046"

    }

}

**Now use npm install command to install dependency.**

**Step 4: Create app folder and inside app folder create app.main.ts**

import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

import { AppModule } from './app.module';

const platform = platformBrowserDynamic();

platform.bootstrapModule(AppModule);

**Step 5: Create app.component.ts**

import { Component } from '@angular/core';

@Component({

  selector: 'my-app',

  template:'<h1>Hello C# corner</h1>'

})

export class AppComponent { }

**Step 6: Create app.module.ts**

import {

    NgModule

} from '@angular/core';

import {

    BrowserModule

} from '@angular/platform-browser';

import {

    AppComponent

} from './app.component';

@NgModule({

    imports: [BrowserModule],

    declarations: [AppComponent],

    bootstrap: [AppComponent]

})

export class AppModule {}

**Step 7: Create index.html**

<html>

  <head>

    <title>My First AngularJs 2 Application</title>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!--<link rel="stylesheet" href="styles.css">-->

 <base href="/">

    <!-- 1. Load libraries -->

     <!-- Polyfill(s) for older browsers -->

    <script src="node\_modules/core-js/client/shim.min.js"></script>

    <script src="node\_modules/zone.js/dist/zone.js"></script>

    <script src="node\_modules/reflect-metadata/Reflect.js"></script>

    <script src="node\_modules/systemjs/dist/system.src.js"></script>

    <!-- 2. Configure SystemJS -->

    <script src="systemjs.config.js"></script>

    <script>

      System.import('app/app.main.js').catch(function(err){ console.error(err); });

    </script>

  </head>

  <!-- 3. Display the application -->

  <body>

    <my-app>Loading...</my-app>

  </body>

</html>

**Step 8: Create systemjs.config.js**

/\*\*

 \* System configuration for Angular samples

 \* Adjust as necessary for your application needs.

 \*/

(function(global) {

    System.config({

        paths: {

            // paths serve as alias

            'npm:': 'node\_modules/'

        },

        // map tells the System loader where to look for things

        map: {

            // our app is within the app folder

            app: 'app',

            // angular bundles

            '@angular/core': 'npm:@angular/core/bundles/core.umd.js',

            '@angular/common': 'npm:@angular/common/bundles/common.umd.js',

            '@angular/compiler': 'npm:@angular/compiler/bundles/compiler.umd.js',

            '@angular/platform-browser': 'npm:@angular/platform-browser/bundles/platform-browser.umd.js',

            '@angular/platform-browser-dynamic': 'npm:@angular/platform-browser-dynamic/bundles/platform-browser-dynamic.umd.js',

            '@angular/http': 'npm:@angular/http/bundles/http.umd.js',

            '@angular/router': 'npm:@angular/router/bundles/router.umd.js',

            '@angular/forms': 'npm:@angular/forms/bundles/forms.umd.js',

            // other libraries

            'rxjs': 'npm:rxjs',

            'angular-in-memory-web-api': 'npm:angular-in-memory-web-api',

        },

        // packages tells the System loader how to load when no filename and/or no extension

        packages: {

            app: {

                main: './main.js',

                defaultExtension: 'js'

            },

            rxjs: {

                defaultExtension: 'js'

            },

            'angular-in-memory-web-api': {

                main: './index.js',

                defaultExtension: 'js'

            }

        }

    });

})(this);

**Step 9: Launch.json and tasks.json file**

Visual Studio Code uses launch.json and tasks.json file to launch your Angular2 Application.  
  
**launch.json**  
Press F1 or CTRL+SHIFT+P in VS code and type launch in the address bar, select node.js file from the selection.

When you create launch.jsonfile, it will be created under .vscode folder, as shown below.

In order to launch the Angular2 Application, launch.json will use the lite-Server defined in node\_modules. Change the program property from “${workspaceRoot}/app.js” to “${workspaceRoot}/node\_modules/lite-server/bin/lite-server”.

**tasks.json**  
Press F1 or CTRL+SHIFT+P and type “task”, select “Configure Task runner” and then use “Typescript-Watch Mode” .This will create tasks.json, which will be created under .vscode folder.

Remove arguments from argsproperty, as we don’t require it now and add new property as “watch”:true.  
  
Now, it’s time to see your first Angular2 Application in the Browser. Run the Application, using CTRL+F5. It will run the Application and see the output in the Browser.