



EBook Gratuito

APPENDIMENTO phaser-framework

Free unaffiliated eBook created from
Stack Overflow contributors.

#phaser-
framework

Sommario

Di.....	1
Capitolo 1: Iniziare con il framework phaser.....	2
Osservazioni.....	2
Versioni.....	2
Phaser 2.....	2
Examples.....	2
Guida introduttiva Phaser.....	2
Iniziare con Phaser usando Node.js.....	3
Capitolo 2: Aggiungi un'immagine in Phaser.....	5
introduzione.....	5
Sintassi.....	5
Osservazioni.....	5
Examples.....	5
Crea e aggiungi allo schermo.....	5
Capitolo 3: Lavorare con TypeScript.....	7
Examples.....	7
Impostazione dell'ambiente Phaser (Asp.Net MVC5 - Typescript - Visual Studio 2015).....	7
Titoli di coda.....	18

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [phaser-framework](#)

It is an unofficial and free phaser-framework ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official phaser-framework.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capitolo 1: Iniziare con il framework phaser

Osservazioni

Phaser è principalmente un framework di gioco HTML5 desktop e mobile open source.

Include un robusto set di documentazione, funzionalità ed esempi per farti muovere velocemente verso un gioco di lavoro. Supporta WebGL, tramite il motore di rendering [Pixi.js](#), e include un fallback Canvas per il supporto su dispositivi meno recenti.

Sebbene il motore sia costruito in JavaScript, include anche le definizioni [TypeScript](#)

C'è una nuova visione del progetto che è conforme a ES6 chiamato [Lazer](#).

Versioni

Phaser 2

Versione	Data di rilascio
2.6.2 Kore Springs	2016/08/25
2.6.1 Caemlyn	2016/07/11
2.6.0 Fal Moran	2016/07/08
2.5.0 Cinque re	2016/06/17
2.4.9 Quattro Re	2016/06/16
2.4.8 Watch Hill	2016/05/19

Examples

Guida introduttiva Phaser

1. Crea una cartella
2. Crea un *index.html* nella nuova directory. Aprilo nell'editor Bracket
3. Scarica il repository Phaser da [github](#), quindi prendi il file *phaser.js* dalla cartella build. Inserisci il file nella directory del tuo progetto.
4. Apri *index.html* e collega **phaser.js** all'interno del tag dell'intestazione.

```
<!doctype html>
<html lang="en">
<head>
```

```
<meta charset="UTF-8" />
<title>My Gamer</title>

<script type="text/javascript" src="lib/phaser.js"></script>

<style type="text/css">
  body {
    margin: 0;
  }
</style>

</head>
<body>
  <div id="gameContainer"></div>
</body>
</html>
```

5. Crea un altro file js all'interno della directory denominata *game.js*

6. Apri il file *game.js* nell'editor e scrivi il seguente codice:

```
// Phaser instance, width 800px, height 600px render as CANVAS.
// Method signature - preload, create and update

var game = new Phaser.Game(800, 600, Phaser.CANVAS, 'gameContainer', { preload: preload,
create: create, update: update });

function preload() {
// this method used to load your game assets
}

function create() {
// this method run only once used to create to game world
}

function update() {
// this method loop 60 times in a seconds, used to handle gameplay.
}
```

8. Salva tutti i file e apri *index.html* usando il life server Bracket (icona in alto a destra).

9. L'ambiente di sviluppo Phaser è ora creato. Nel browser deve essere visualizzata una schermata della console per la verifica degli errori.

Iniziare con Phaser usando Node.js

1. Crea una cartella in cui ti piacerebbe vivere la tua partita e trasferiscila

```
mkdir my-new-game
cd my-new-game
```

2. Inizializza la directory usando npm.

```
npm init -y
```

3. Installa phaser come pacchetto di nodi.

```
npm install phaser
```

4. Installa http-server come modulo globale, da utilizzare sulla riga di comando.

```
npm install -g http-server
```

5. Creare un file index.html e fare riferimento all'eseguibile di Phaser e incollarvi il seguente codice.

```
<!doctype html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <title>My Gamer</title>
  <script type="text/javascript" src="node_modules/phaser/build/phaser.js"></script>
  <style type="text/css">
    body {
      margin: 0;
    }
  </style>
</head>

<body>
  <div id="helloWorld"></div>
</body>
<script>
var game = new Phaser.Game(800, 600, Phaser.CANVAS, 'helloWorld', {
  create: create
});

function create() {

  var text = "Hello World!";
  var style = {
    font: "65px Arial",
    fill: "#ff0044",
    align: "center"
  };

  var t = game.add.text(game.world.centerX, 300, text, style);
  t.anchor.set(0.5);

}
</script>

</html>
```

6. Avvia il server e carica <http://localhost:8080> nel tuo browser!

```
hs
```

Leggi Iniziare con il framework phaser online: <https://riptutorial.com/it/phaser-framework/topic/6168/iniziare-con-il-framework-phaser>

Capitolo 2: Aggiungi un'immagine in Phaser

introduzione

Aggiunta di un file immagine (preferibile a png) al gioco come oggetto "Immagine" in Phaser.

Sintassi

- `game.load.image (nome: stringa, file: stringa,);`
- `game.add.image (x: numero, y: numero, nome: stringa);`

Osservazioni

- Un oggetto immagine è una buona scelta per le cose nel tuo gioco che non usano le animazioni dei fotogrammi e non devono essere necessariamente sprite.
- Di default il punto di ancoraggio per un'immagine nell'angolo in alto a sinistra, ma puoi cambiarlo in questo modo: `image.anchor.setTo(0.5, 0.5);`

Examples

Crea e aggiungi allo schermo

Devi prima creare un oggetto "Gioco" in Phaser.

```
var game = new Phaser.Game(800, 600, Phaser.AUTO, 'phaser-example', { preload: preload, create: create });
```

Nella funzione di richiamo del precaricamento caricare l'immagine.

```
function preload() {  
    game.load.image('thing', 'assets/thing-image.png');  
}
```

Parametro	Dettagli (Game.add.image)
nome	il nome usato per fare riferimento all'immagine nel metodo <code>game.add.image</code> .
file	percorso del file di asset (relativo alla directory root per il progetto).

Quindi nella funzione di creazione utilizzare il metodo "aggiungi" dell'oggetto di gioco per creare l'oggetto Immagine e farlo sullo schermo.

```
function create() {
```

```
var image = game.add.image(100, 100, 'thing');  
}
```

Parametro	Dettagli (Game.add.image)
X	le coordinate x dove l'immagine deve essere aggiunta.
y	la coordinata y in cui deve essere aggiunta l'immagine.
nome	il nome dell'immagine assegnata nel metodo game.load.image.

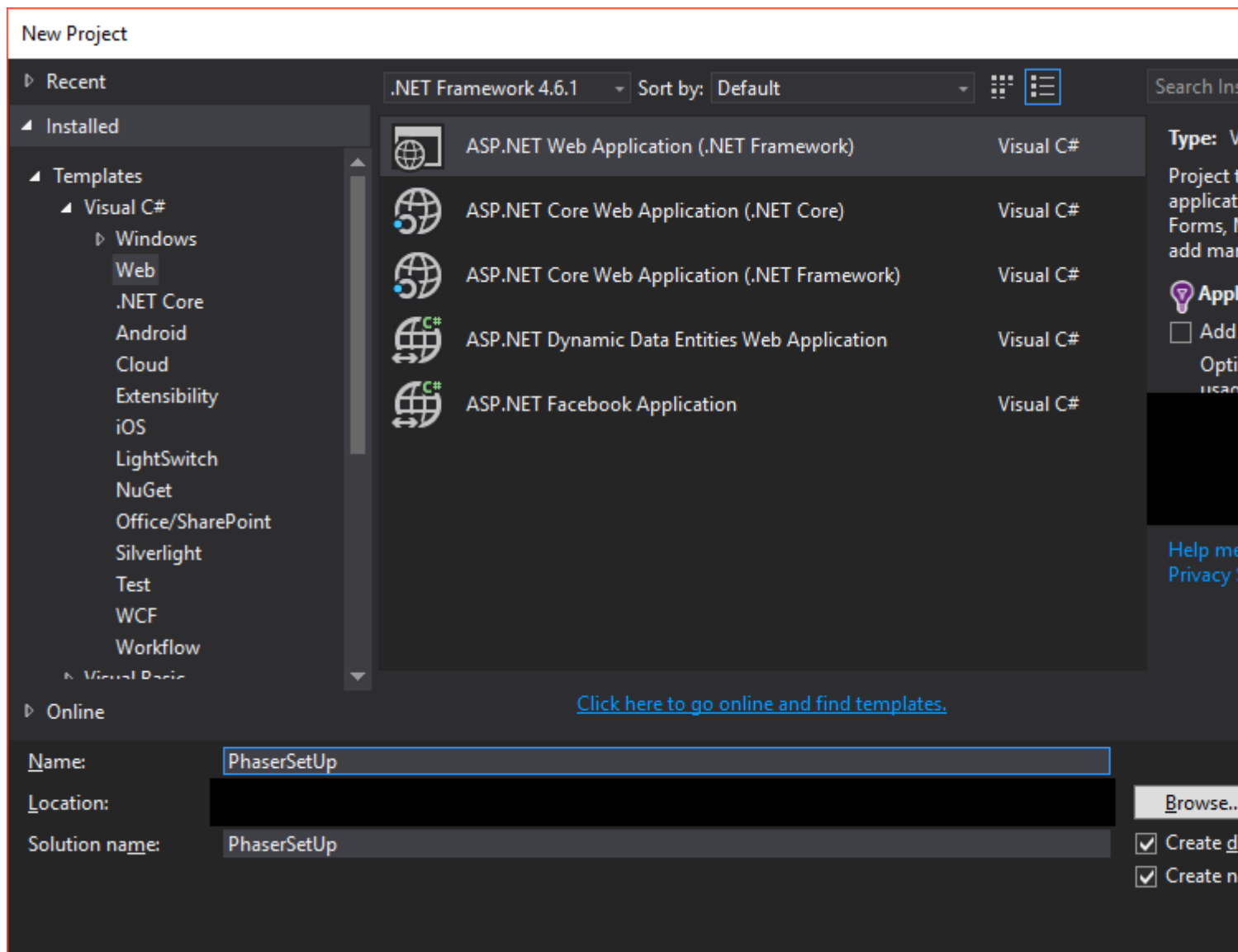
Leggi **Aggiungi un'immagine in Phaser online**: <https://riptutorial.com/it/phaser-framework/topic/9853/aggiungi-un-immagine-in-phaser>

Capitolo 3: Lavorare con TypeScript

Examples

Impostazione dell'ambiente Phaser (Asp.Net MVC5 - Typescript - Visual Studio 2015)








Creare un nuovo progetto ASP.Net:




Seleziona il modello vuoto:

Select a template:

ASP.NET 4.6.1 Templates

 Empty	 Web Forms	 MVC	 Web API	 Single Page Application
 Azure API App	 Azure Mobile Service			

An empty project template for creating ASP.NET applications. This template does not have any content.

[Learn more](#)[Change Authentication](#)Authentication: **No Authentication** **Microsoft Azure** Host in the cloud

App Service

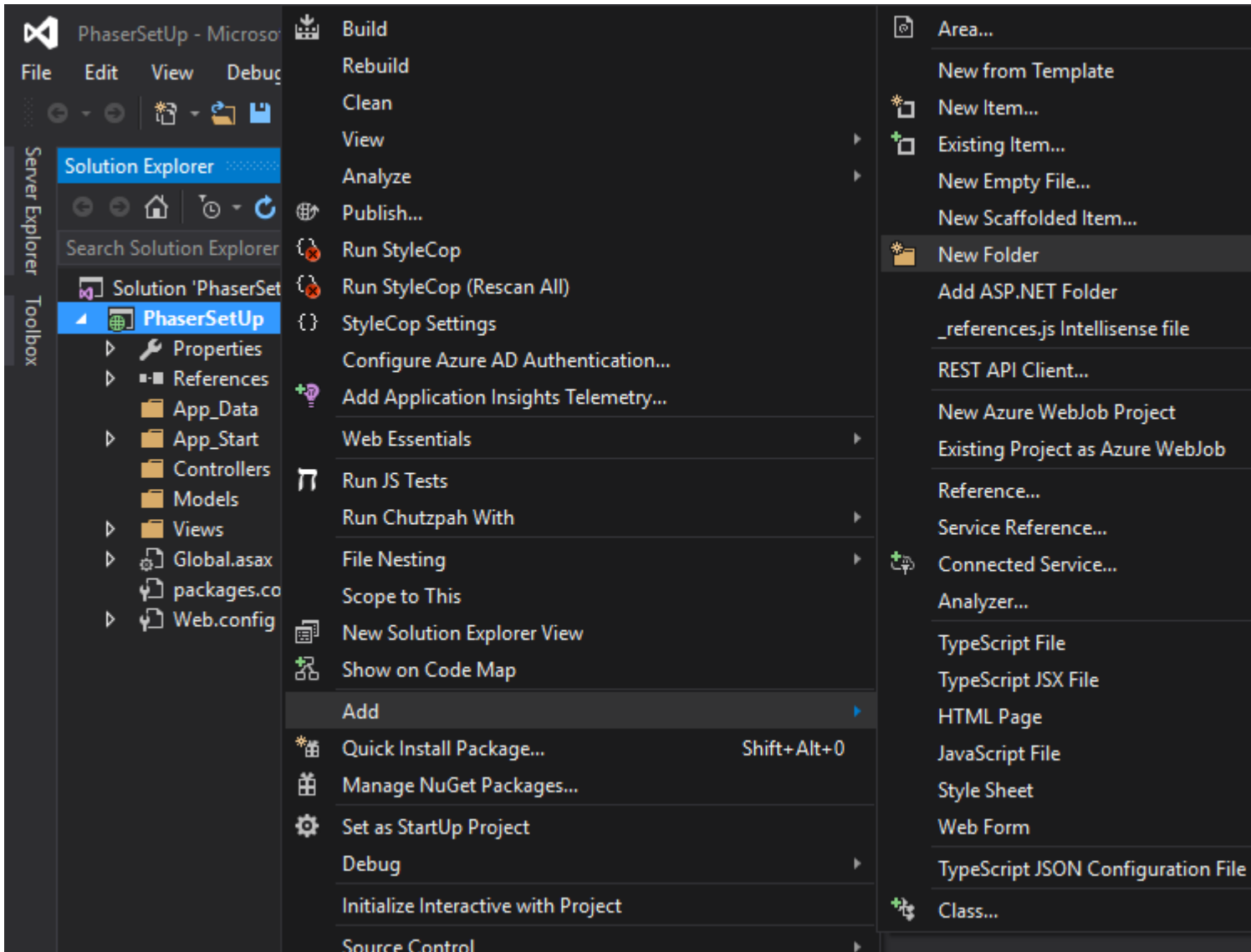
Add folders and core references for:

 Web Forms MVC Web API Add unit testsTest project name:

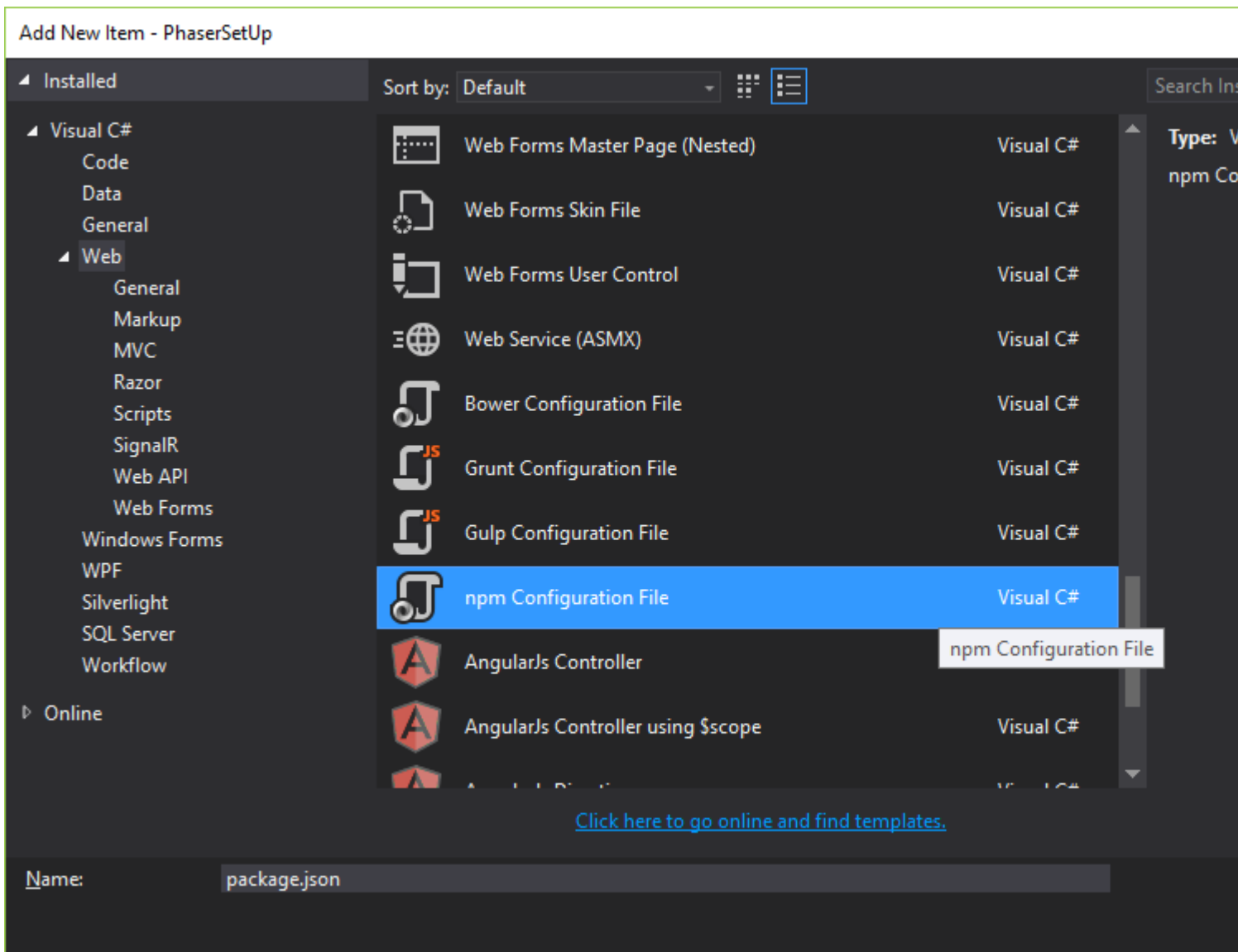
OK

Cancel

Aggiungi due nuove cartelle: `App` e `Scripts` nella cartella principale:

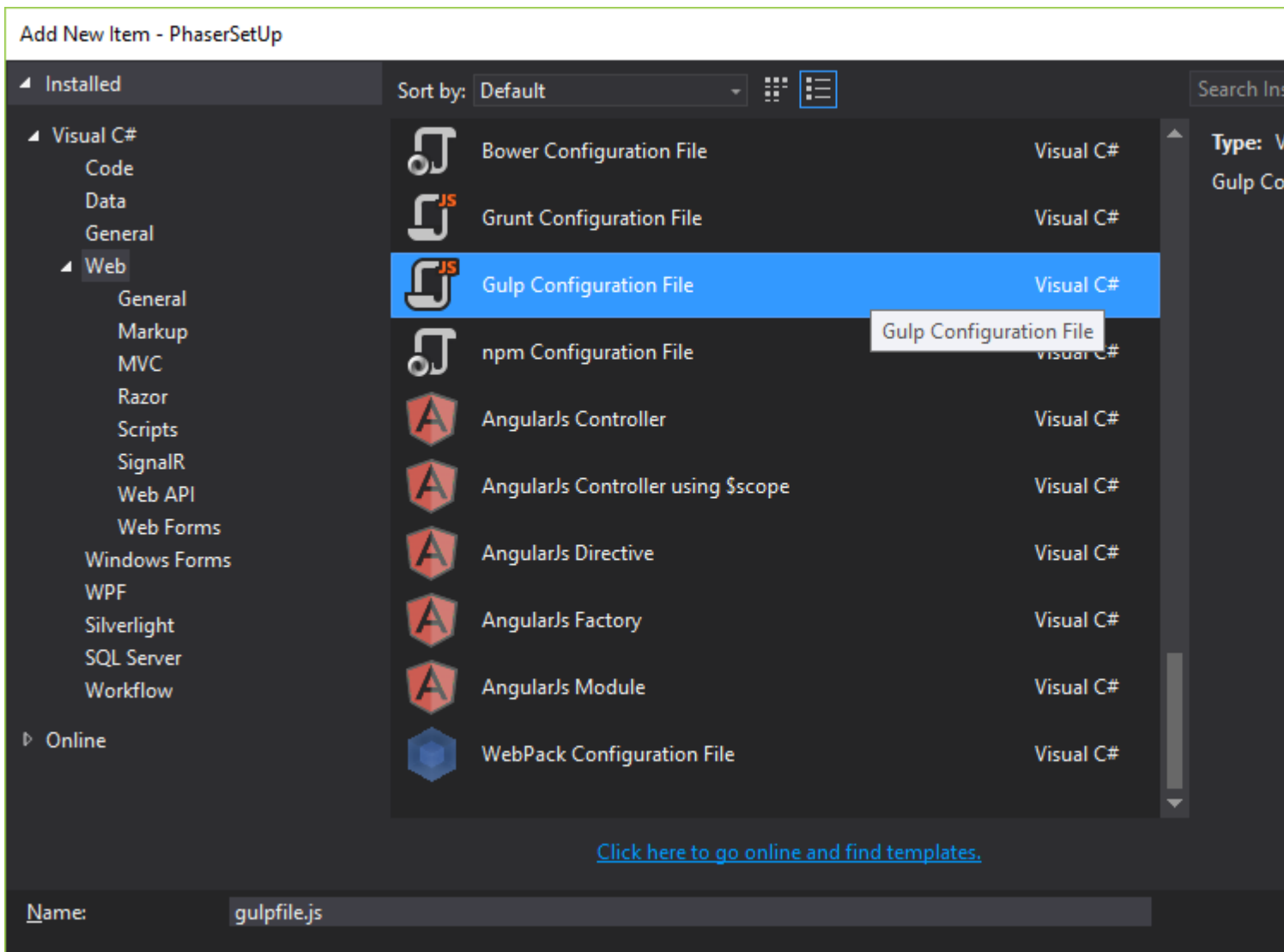


Aggiungi il file di configurazione `npm` nella cartella principale:

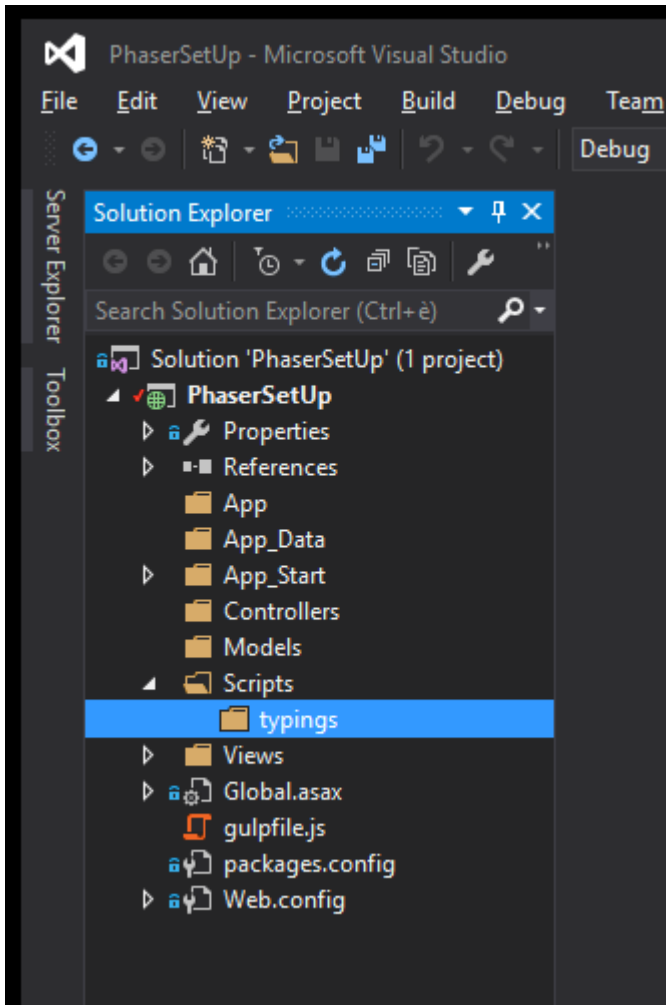


```
{
  "version": "1.0.0",
  "name": "phaser.js.environment.setup",
  "private": true,
  "devDependencies": {
    "gulp": "3.9.1",
    "phaser": "2.6.2"
  }
}
```

Aggiungi il file di configurazione `gulp` nella cartella principale:



Aggiungi la cartella `typings` nella cartella `Scripts` :



Compito Gulp:

```
/// <binding ProjectOpened='install' />

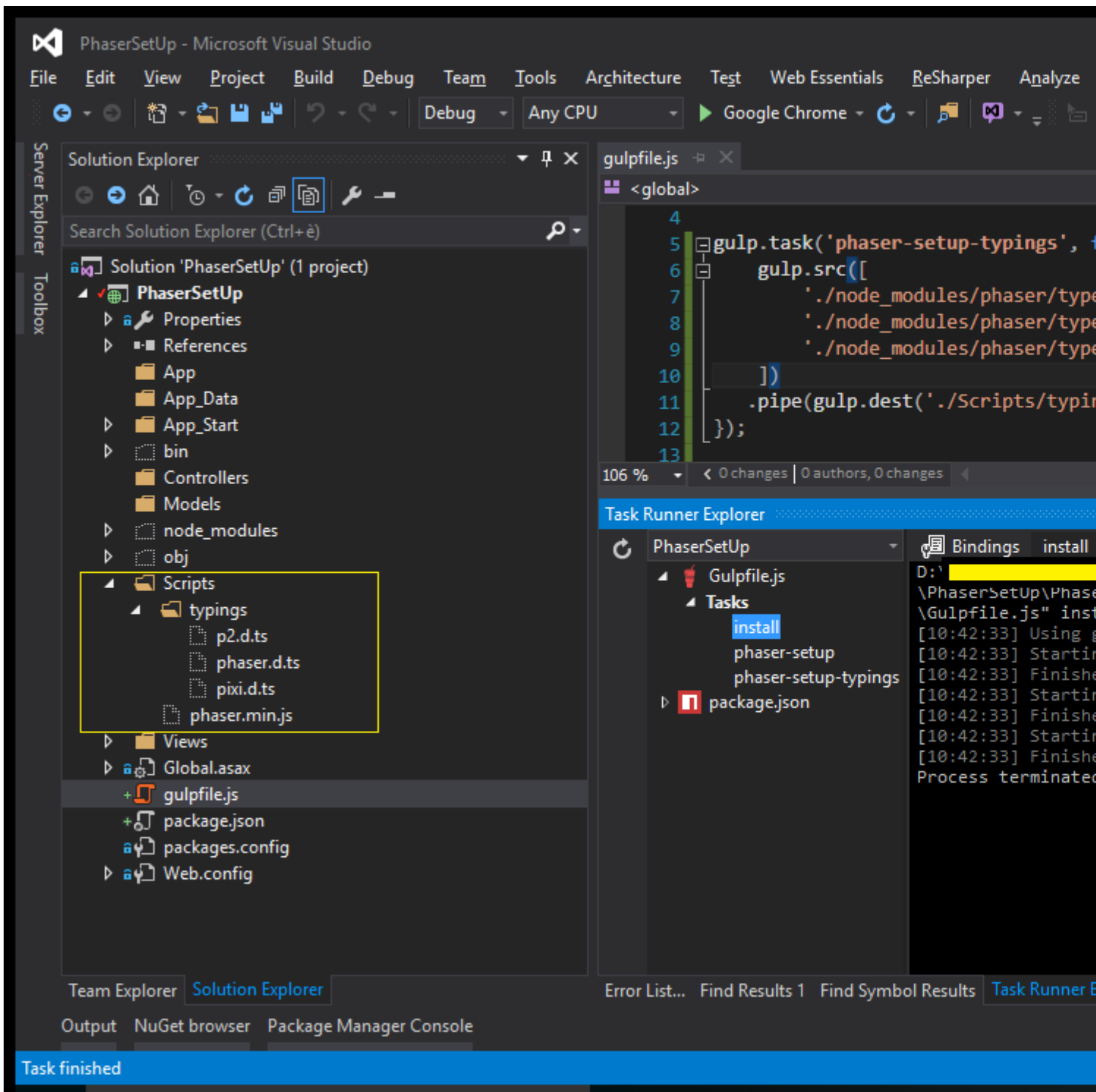
var gulp = require('gulp');

gulp.task('phaser-setup-typings', function () {
  gulp.src([
    './node_modules/phaser/typescript/pixi.d.ts',
    './node_modules/phaser/typescript/p2.d.ts',
    './node_modules/phaser/typescript/phaser.d.ts',
  ])
  .pipe(gulp.dest('./Scripts/typings'));
});

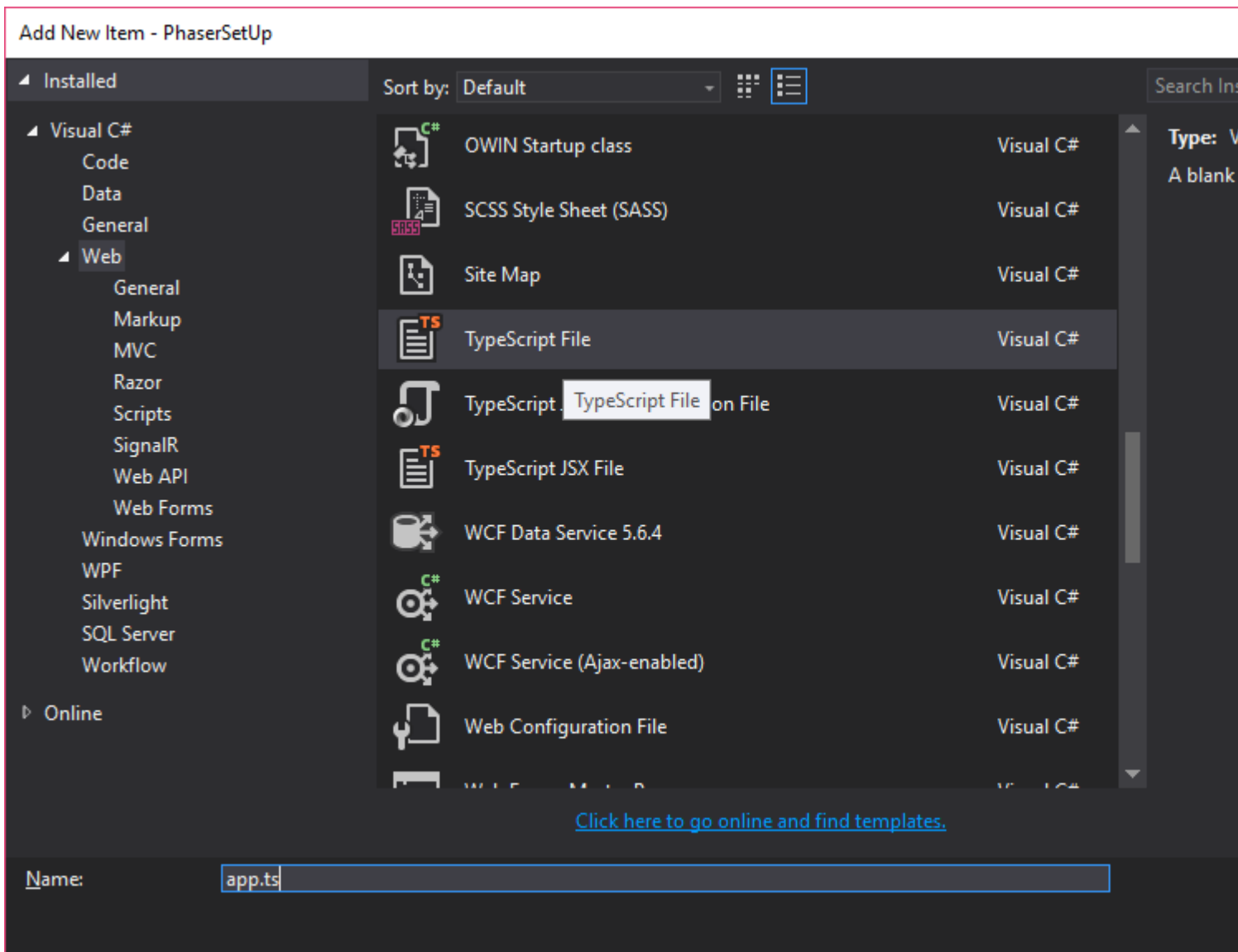
gulp.task('phaser-setup', function () {
  gulp.src([
    './node_modules/phaser/build/phaser.min.js',
  ])
  .pipe(gulp.dest('./Scripts/'));
});

gulp.task('install', ['phaser-setup-typings', 'phaser-setup']);
```

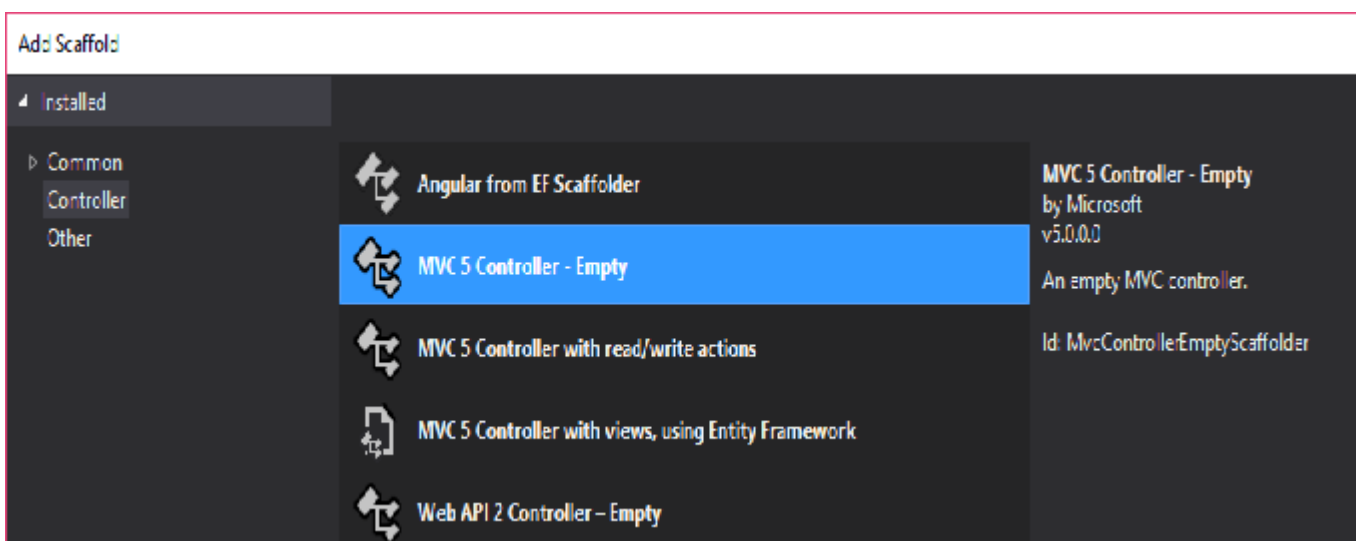
Esegui l'attività di installazione:



Aggiungi un file dattiloscritto nella cartella `App` :



Aggiungi un controller MVC :



```
using System.Web.Mvc;

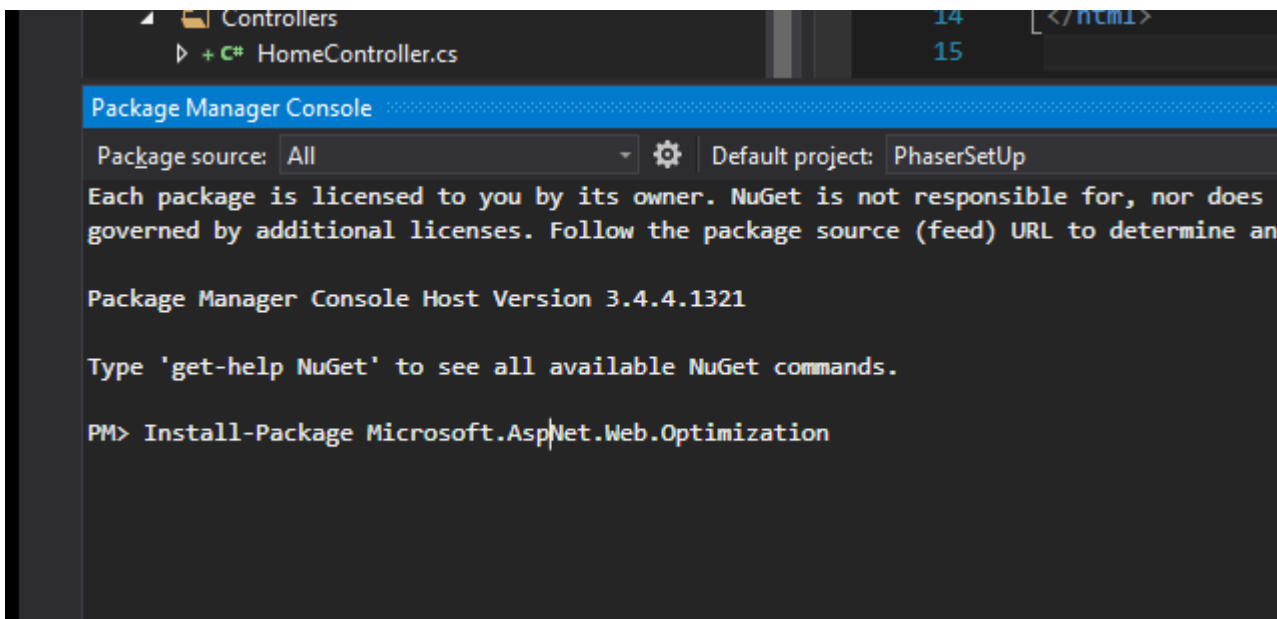
namespace PhaserSetUp.Controllers
{
```



```
public class HomeController : Controller
{
    // GET: Home
    public ActionResult Index()
    {
        return View();
    }
}
```

Aggiungi [web optimization pacchetto nuget web optimization](#) :

```
Install-Package Microsoft.AspNet.Web.Optimization
```



Aggiungi la classe `BundleConfig.cs` nella cartella `App_Start` :

```
using System.Web.Optimization;

namespace PhaserSetUp.App_Start
{
    public class BundleConfig
    {
        public static void RegisterBundles(BundleCollection bundles)
        {
            bundles.Add(new ScriptBundle("~/bundles/app").Include(
                "~/App/app.js"));
        }
    }
}
```

Modifica il `Global.asax`

```
using System;
using System.Web;
using System.Web.Mvc;
using System.Web.Routing;
using System.Web.Http;
```

```

namespace PhaserSetUp
{
    public class Global : HttpApplication
    {
        void Application_Start(object sender, EventArgs e)
        {
            // Code that runs on application startup
            AreaRegistration.RegisterAllAreas();
            GlobalConfiguration.Configure(WebApiConfig.Register);
            RouteConfig.RegisterRoutes(RouteTable.Routes);
            BundleConfig.RegisterBundles(BundleTable.Bundles);
        }
    }
}

```

Aggiungi una vista:

Visual Studio Add New Item dialog for 'PhaserSetUp'. The 'Visual C#' > 'Web' > 'MVC 5 Layout Page (Razor)' item is selected. The 'Name' field is set to 'Index.cshtml'.

```

@using System.Web.Optimization
<!DOCTYPE html>

<html>
<head>

```

```
<meta name="viewport" content="width=device-width" />
<title>@ViewBag.Title</title>
</head>
<body>
  <div>
    @RenderBody()
  </div>
  <script src="../../Scripts/phaser.min.js"></script>
  @Scripts.Render("~/bundles/app")
</body>
</html>
```

Leggi Lavorare con TypeScript online: <https://riptutorial.com/it/phaser-framework/topic/8054/lavorare-con-typescript>

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con il framework phaser	4444 , Bob_Gneu , Community , InferOn , Shohanur Rahaman
2	Aggiungi un'immagine in Phaser	Jim
3	Lavorare con TypeScript	Bob_Gneu , InferOn