



eBook Gratuit

APPRENEZ

phaser-framework

eBook gratuit non affilié créé à partir des  
**contributeurs de Stack Overflow.**

#phaser-

framework

# Table des matières

À propos.....	1
<b>Chapitre 1: Démarrer avec phaser-framework.....</b>	<b>2</b>
Remarques.....	2
Versions.....	2
Phaser 2.....	2
Exemples.....	2
Mise en route de Phaser.....	2
Premiers pas avec Phaser avec Node.js.....	3
<b>Chapitre 2: Ajouter une image dans Phaser.....</b>	<b>6</b>
Introduction.....	6
Syntaxe.....	6
Remarques.....	6
Exemples.....	6
Créer et ajouter à l'écran.....	6
<b>Chapitre 3: Travailler avec TypeScript.....</b>	<b>8</b>
Exemples.....	8
Configuration de l'environnement Phaser (Asp.Net MVC5 - Typescript - Visual Studio 2015).....	8
<b>Crédits.....</b>	<b>19</b>

# À propos

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](mailto:info@zzzprojects.com)

# Chapitre 1: Démarrer avec phaser-framework

## Remarques

[Phaser](#) est principalement une infrastructure de jeu open source Desktop et Mobile HTML5.

Il comprend un ensemble robuste de documentation, de fonctionnalités et d'exemples pour vous permettre de progresser rapidement vers un jeu de travail. Il prend en charge WebGL, via le moteur de rendu [Pixi.js](#), et inclut une solution de secours Canvas pour prendre en charge les anciens périphériques.

Bien que le moteur soit construit en JavaScript, il inclut également des définitions [TypeScript](#)

Il y a une nouvelle vision du projet qui est conforme à la norme ES6 appelée [Lazer](#).

## Versions

### Phaser 2

Version	Date de sortie
<a href="#">2.6.2</a> Kore Springs	2016-08-25
<a href="#">2.6.1</a> Caemlyn	2016-07-11
<a href="#">2.6.0</a> Fal Moran	2016-07-08
<a href="#">2.5.0</a> Cinq rois	2016-06-17
<a href="#">2.4.9</a> Quatre rois	2016-06-16
<a href="#">2.4.8</a> Watch Hill	2016-05-19

## Exemples

### Mise en route de Phaser

1. Créer un dossier
2. Créez un `index.html` dans le nouveau répertoire. Ouvrez-le dans l'éditeur Bracket
3. Téléchargez le référentiel Phaser à partir de [github](#), puis récupérez le fichier `phaser.js` du dossier build. Placez le fichier dans votre répertoire de projet.
4. Ouvrez `index.html` et liez le **phaser.js** dans la balise header.

```
<!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. Créez un autre fichier js dans le répertoire nommé *game.js*

6. Ouvrez le fichier *game.js* dans l'éditeur et écrivez le code suivant:

```
// 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. Enregistrez tous les fichiers et ouvrez *index.html* en utilisant le serveur de vie Bracket (icône en haut à droite).

9. L'environnement de développement Phaser est maintenant créé. Un écran de console devrait apparaître dans le navigateur pour la vérification des erreurs.

## Premiers pas avec Phaser avec Node.js

1. Créez un dossier dans lequel vous souhaitez que votre jeu soit en direct et accédez à celui-ci.

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

2. Initialiser le répertoire en utilisant npm.

```
npm init -y
```

### 3. Installez phaser en tant que package de nœud.

```
npm install phaser
```

### 4. Installez http-server en tant que module global, à utiliser sur la ligne de commande.

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

### 5. Créez un fichier index.html et référez l'exécutable Phaser et collez-y le code suivant.

```
<!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. Démarrez le serveur et chargez <http://localhost:8080> dans votre navigateur!

```
hs
```

Lire Démarrer avec phaser-framework en ligne: <https://riptutorial.com/fr/ Phaser-framework/topic/6168/demarrer-avec-phaser-framework>

# Chapitre 2: Ajouter une image dans Phaser

## Introduction

Ajouter un fichier image (préférable à un png) à votre jeu en tant qu'objet "Image" dans Phaser.

## Syntaxe

- `game.load.image (nom: chaîne, fichier: chaîne,);`
- `game.add.image (x: numéro, y: numéro, nom: chaîne);`

## Remarques

- Un objet Image est un bon choix pour les objets de votre jeu qui n'utilisent pas d'animations d'images et ne nécessitent pas nécessairement un Sprite.
- Par défaut, le point d'ancrage d'une image dans le coin supérieur gauche, mais vous pouvez le modifier comme `image.anchor.setTo(0.5, 0.5);` ; `image.anchor.setTo(0.5, 0.5);`

## Exemples

### Créer et ajouter à l'écran

Vous devez d'abord créer un objet "Jeu" dans Phaser.

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

Dans la fonction de rappel de préchargement, chargez l'image.

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

Paramètre	Détails (Game.add.image)
prénom	le nom utilisé pour référencer l'image dans la méthode <code>game.add.image</code> .
fichier	chemin d'accès au fichier d'actif (relatif au répertoire racine du projet).

Ensuite, dans la fonction de création, utilisez la méthode "Ajouter" de l'objet du jeu pour créer l'objet Image et l'afficher à l'écran.

```
function create() {
```

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

Paramètre	Détails (Game.add.image)
X	les coordonnées x où l'image doit être ajoutée.
y	la coordonnée y où l'image doit être ajoutée.
prénom	le nom de l'image assignée dans la méthode game.load.image.

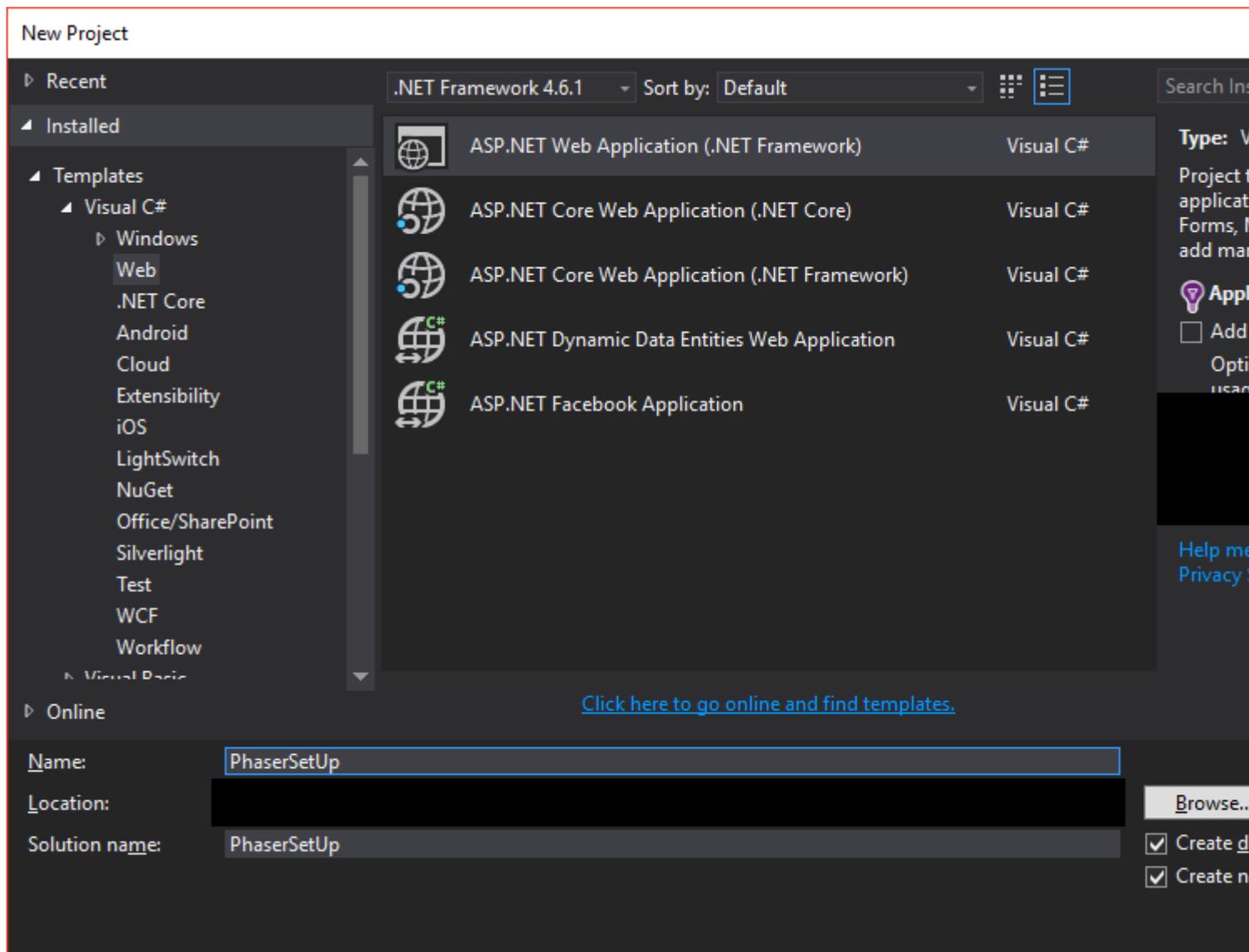
Lire Ajouter une image dans Phaser en ligne: <https://riptutorial.com/fr/phaser-framework/topic/9853/ajouter-une-image-dans-phaser>

# Chapitre 3: Travailler avec TypeScript

## Exemples

### Configuration de l'environnement Phaser (Asp.Net MVC5 - Typescript - Visual Studio 2015)

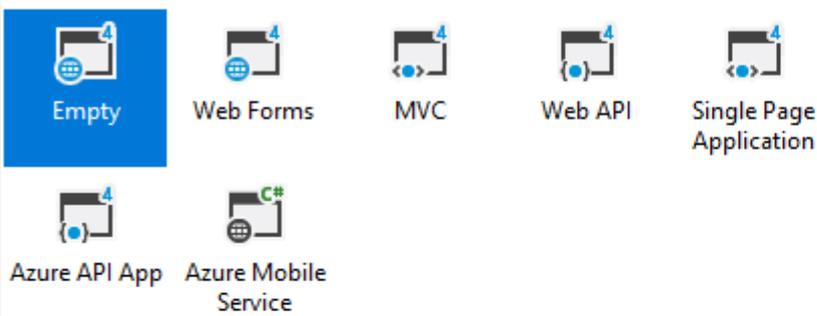
Créez un nouveau projet ASP.Net:



Sélectionnez le modèle vide:

Select a template:

## ASP.NET 4.6.1 Templates



The image shows a selection screen for ASP.NET 4.6.1 templates. The 'Empty' template is selected and highlighted in blue. Other templates include Web Forms, MVC, Web API, Single Page Application, Azure API App, and Azure Mobile Service.

Template Name	Selected
Empty	Yes
Web Forms	No
MVC	No
Web API	No
Single Page Application	No
Azure API App	No
Azure Mobile Service	No

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

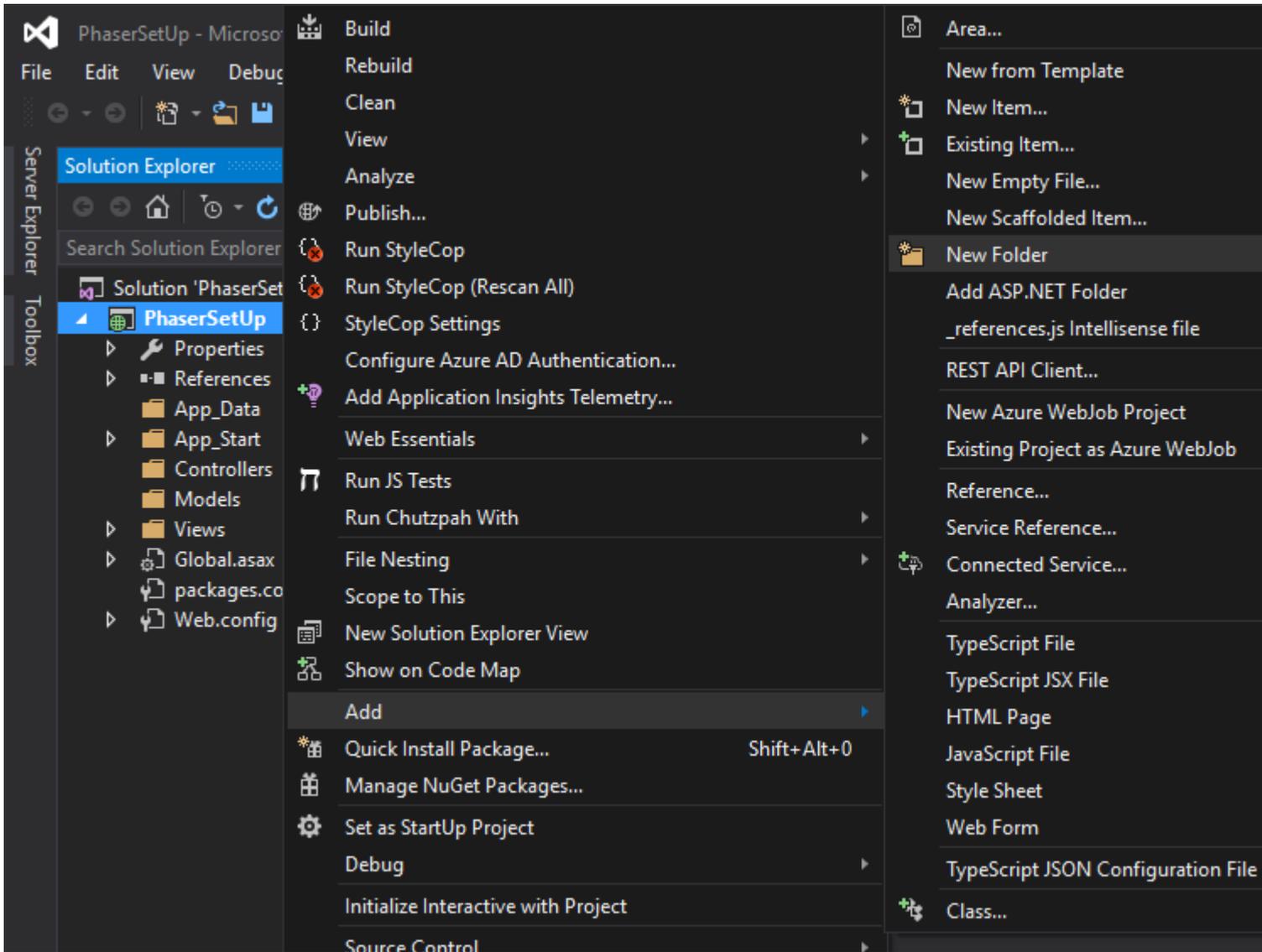
Add folders and core references for:

 Web Forms  MVC  Web API Add unit testsTest project name: 

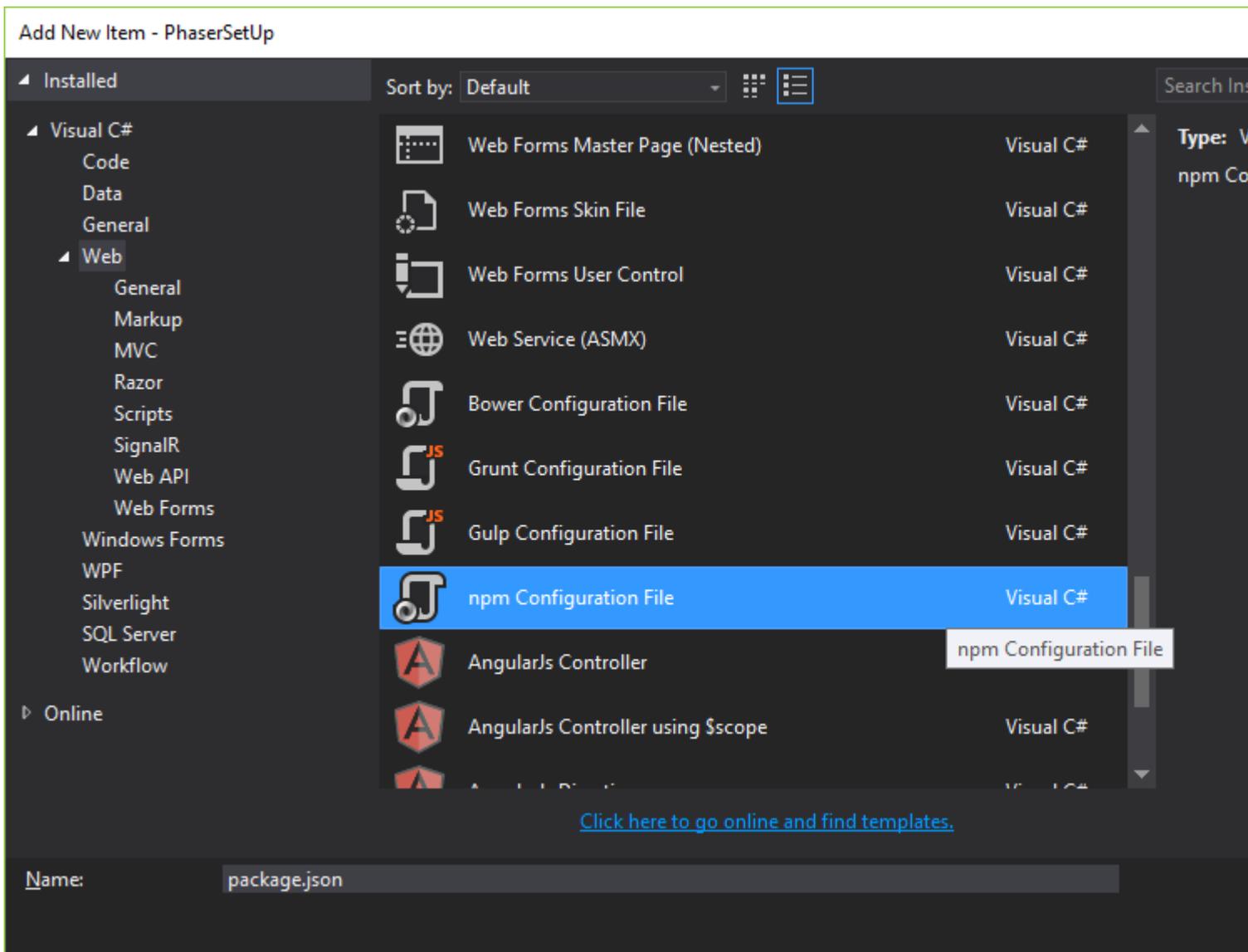
OK

Cancel

Ajoutez deux nouveaux dossiers: `App` et `Scripts` dans le dossier racine:

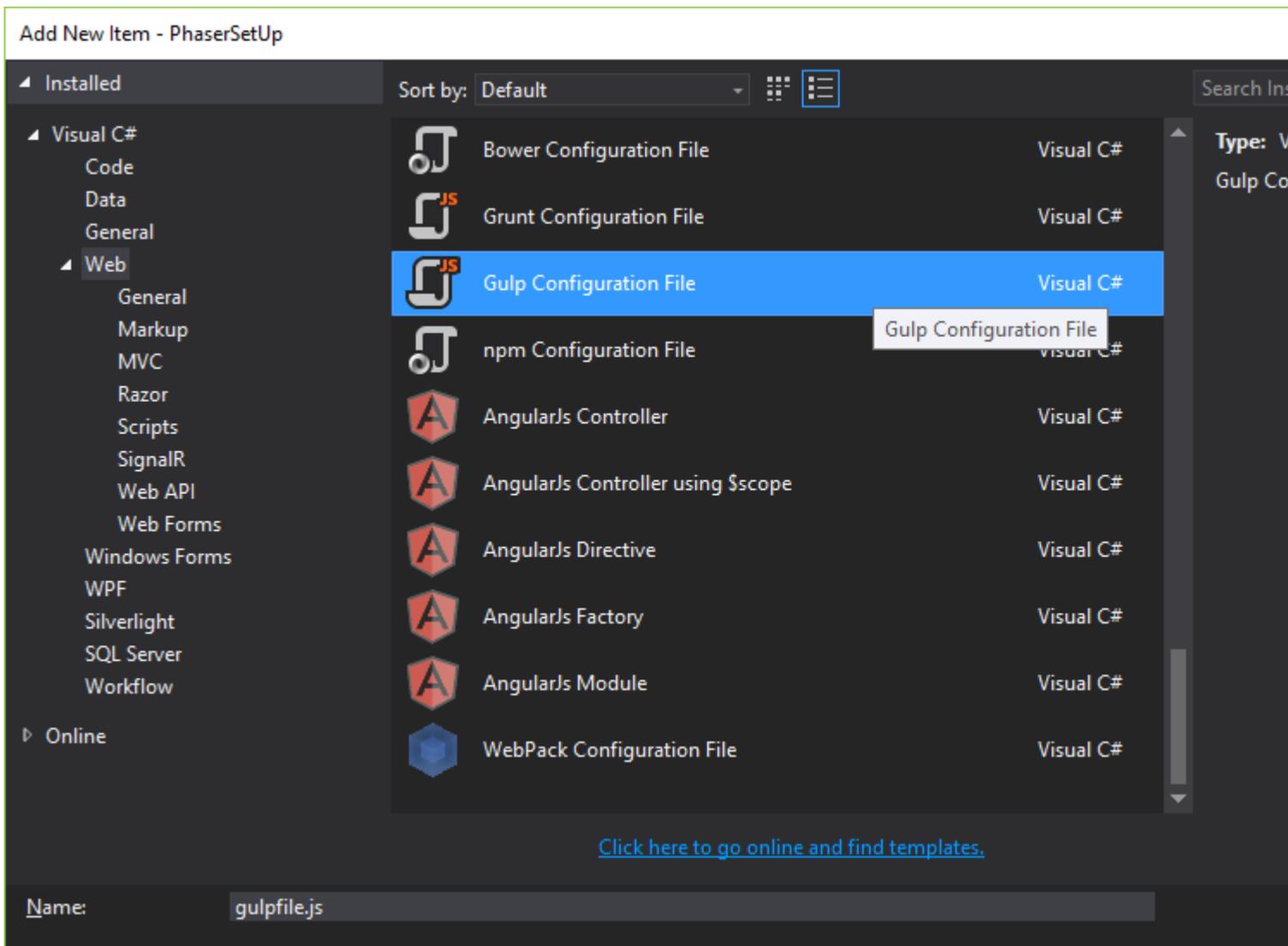


Ajoutez le fichier de configuration `npm` dans le dossier racine:

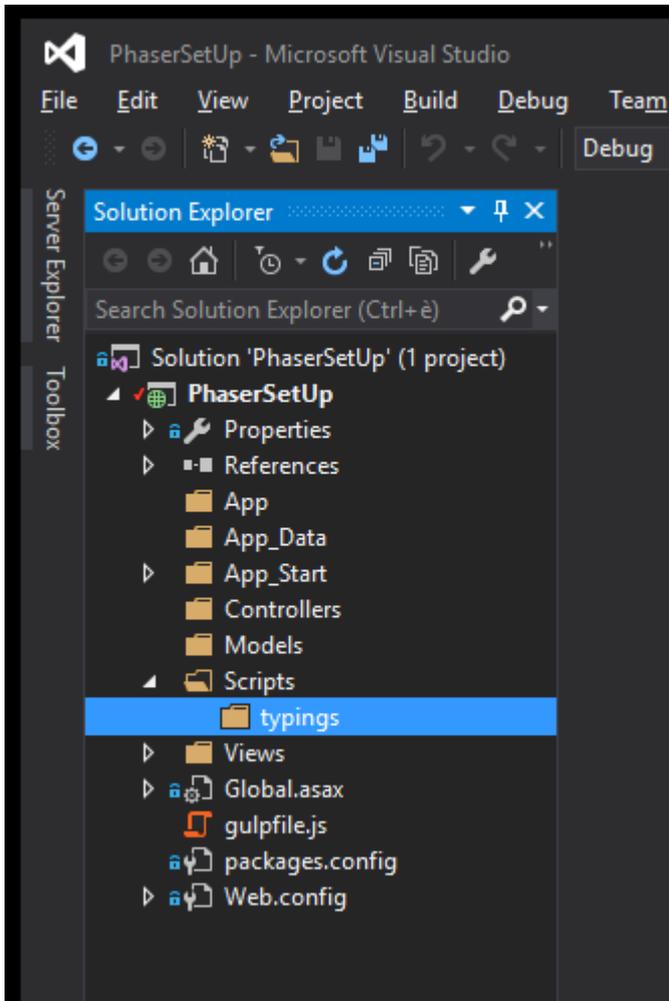


```
{
  "version": "1.0.0",
  "name": "phaser.js.environment.setup",
  "private": true,
  "devDependencies": {
    "gulp": "3.9.1",
    "phaser": "2.6.2"
  }
}
```

Ajoutez le fichier de configuration `gulp` dans le dossier racine:



Ajouter typings dossier de typings dans le dossier scripts :



## Tâche Gulp:

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

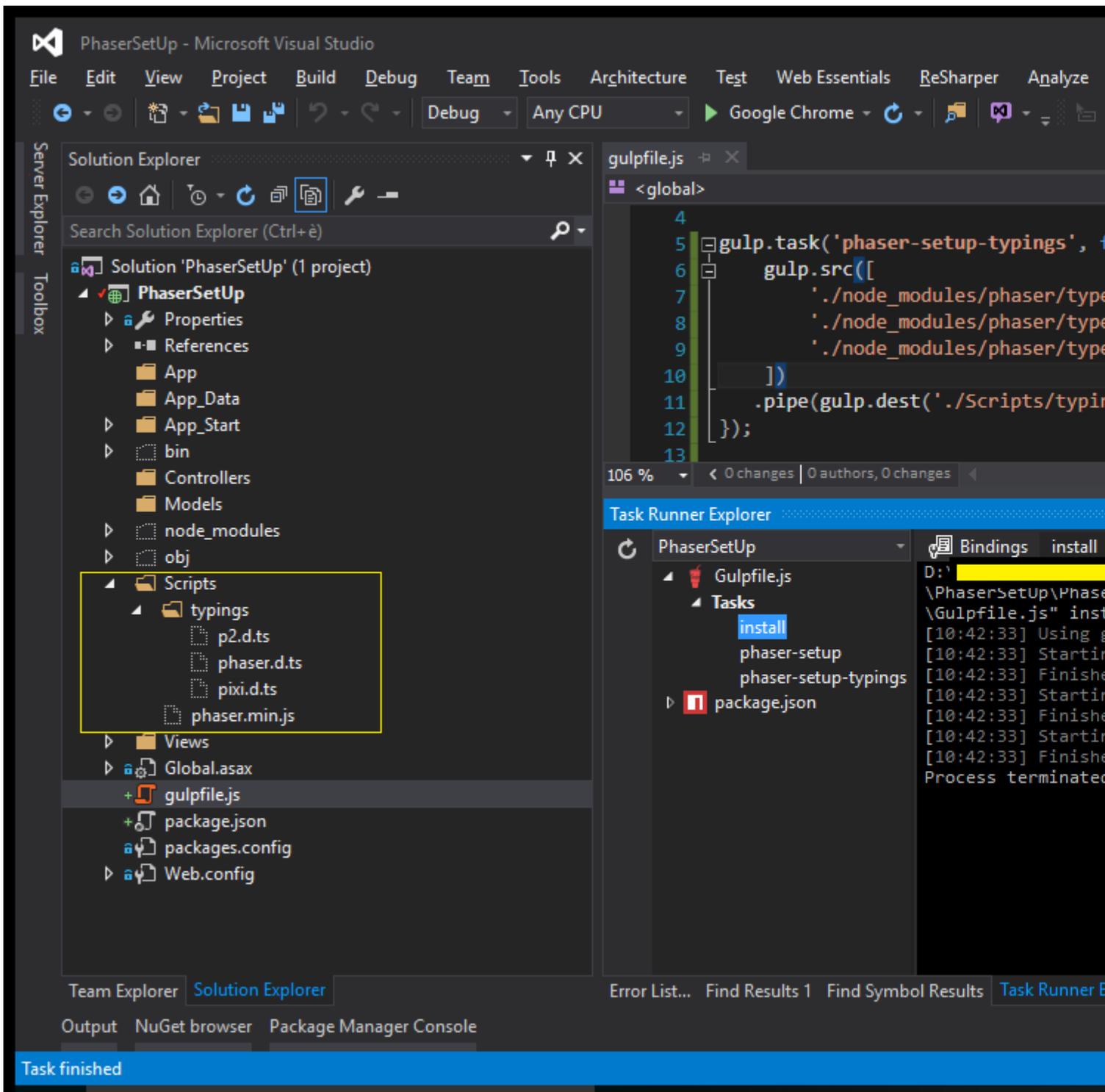
var gulp = require('gulp');

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'));
});

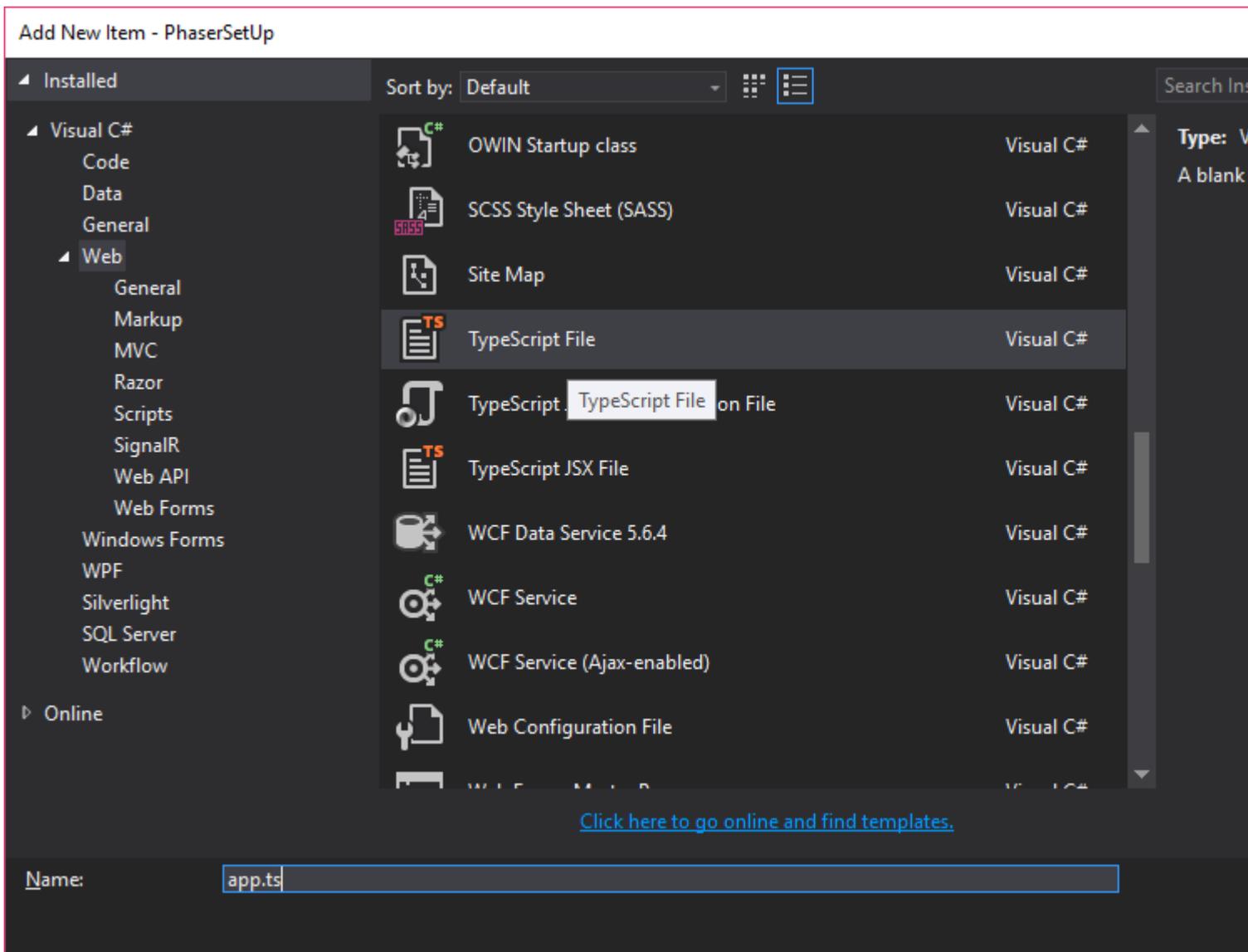
gulp.task('phaser-setup', function () {
  gulp.src([
    './node_modules/phaser/build/phaser.min.js',
  ])
  .pipe(gulp.dest('./Scripts/'));
});

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

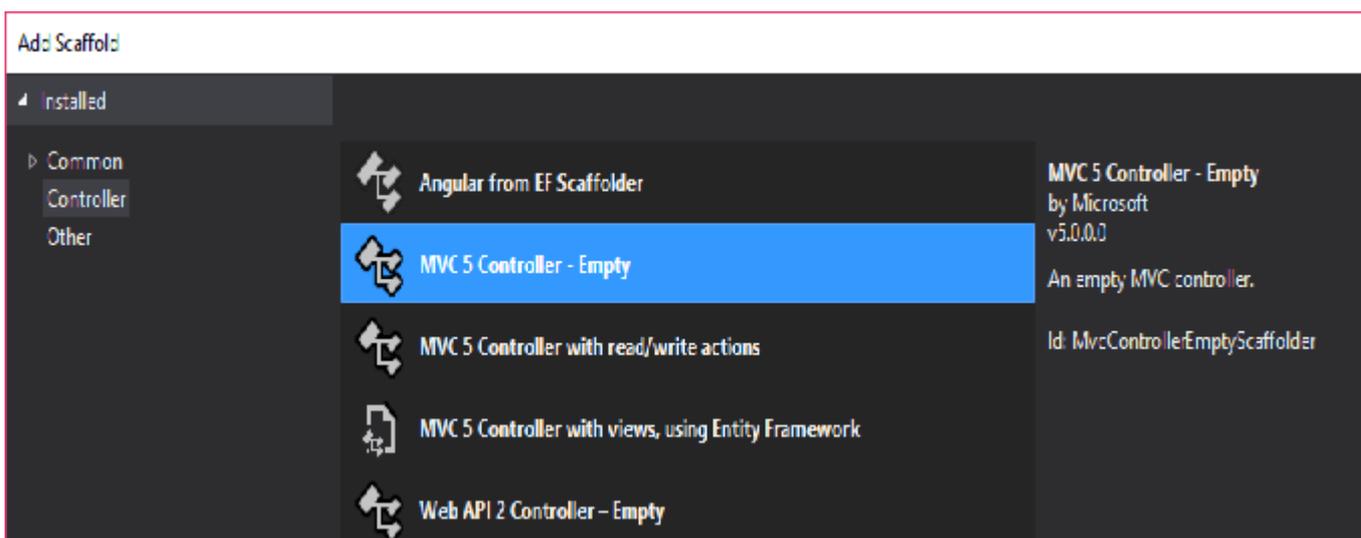
Exécutez la tâche d'installation:



Ajoutez un fichier texte dans le dossier App :



Ajouter un contrôleur MVC :



```
using System.Web.Mvc;

namespace PhaserSetUp.Controllers
{
```

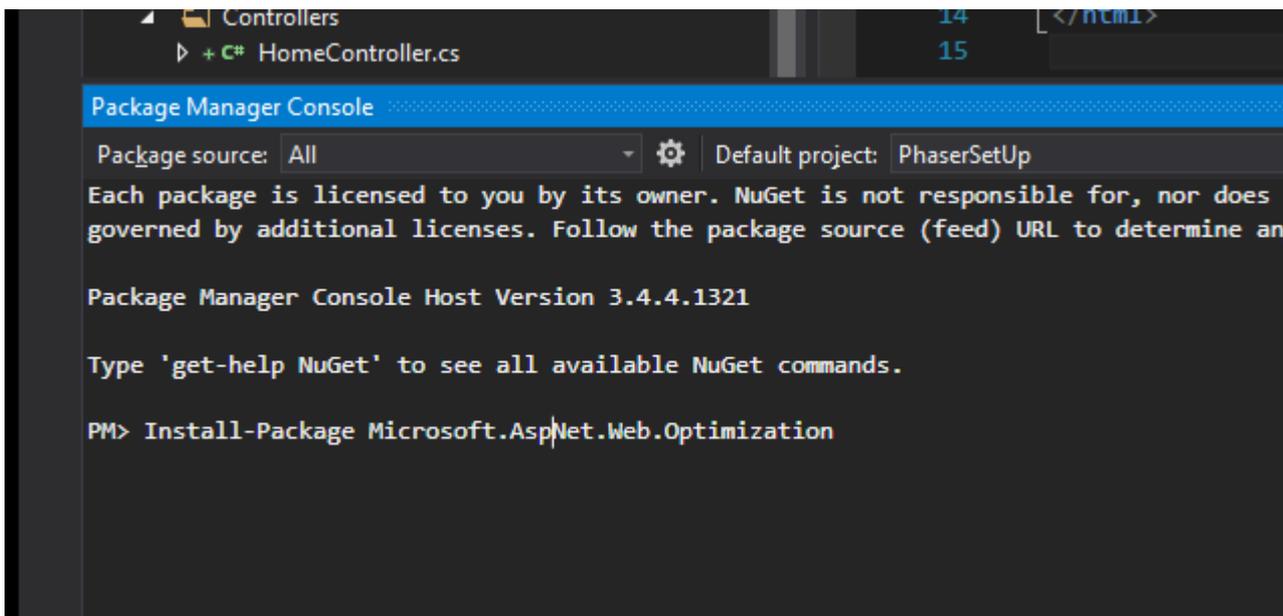
```

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

```

Ajouter [web optimization package nuget](#) [web optimization](#) :

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



Ajoutez la classe `BundleConfig.cs` dans le dossier `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"));
        }
    }
}

```

Modifier le `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);
        }
    }
}

```

Ajouter une vue:

Visual Studio Add New Item dialog for 'PhaserSetUp'. The 'MVC 5 Layout Page (Razor)' template is selected. The name 'Index.cshtml' is entered in the 'Name:' field.

```

@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>
```

Lire Travailler avec TypeScript en ligne: <https://riptutorial.com/fr/phaser-framework/topic/8054/travailler-avec-typescript>

# Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec phaser-framework	<a href="#">4444</a> , <a href="#">Bob_Gneu</a> , <a href="#">Community</a> , <a href="#">InferOn</a> , <a href="#">Shohanur Rahaman</a>
2	Ajouter une image dans Phaser	<a href="#">Jim</a>
3	Travailler avec TypeScript	<a href="#">Bob_Gneu</a> , <a href="#">InferOn</a>