

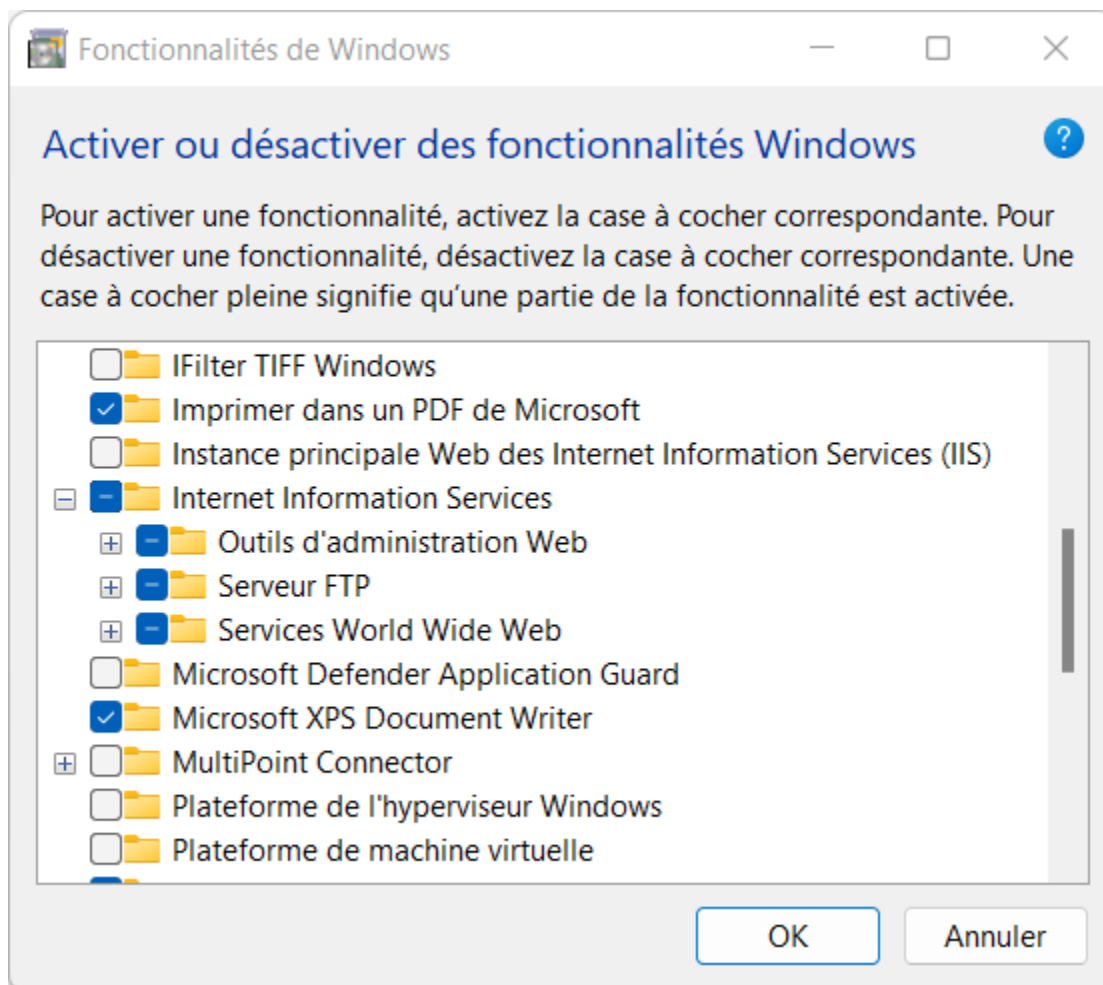
LAB11 - Deploy Angular app to an IIS Web server

If you have an Angular application and you need to deploy it to an IIS server this lab will show you how you can do it.

Enabling IIS

The IIS web server is not enabled by default, so we need to do it manually.

1. In Windows, navigate to **Control Panel > Programs > Programs and Features > Turn Windows features on or off** (left side of the screen).
2. Select the **Internet Information Services** check box. Select OK.



> The IIS installation may require a system restart.

Building the app

1. We will be serving our Angular application is served under `localhost/task-manager` . By default Angular has build our application assuming that the files will be loaded from the main path, `/` . To fix this we need to specify the `base-href` values when building the application.

Go to `index.html` file and update the `base` tag like the following:

```
<base href="./">
```

The HTML `<base href="..." />` specifies a base path for resolving relative URLs to assets such as images, scripts, and style sheets. For example, given the `<base href="/my/app/">` , the browser resolves a URL such as `some/place/foo.jpg` into a server request for `my/app/some/place/foo.jpg` . During navigation, the Angular router uses the *base href* as the base path to component, template, and module files.

2. To build your application for production, use the `build` command from the Angular CLI. By default, this command uses the `production` build configuration.

```
ng build
```

3. This command creates a `dist` folder in the application root directory with all the files that a hosting service needs for serving your application. It contains the compiled code of your angular application in the format of JavaScript and also the required HTML and CSS files.

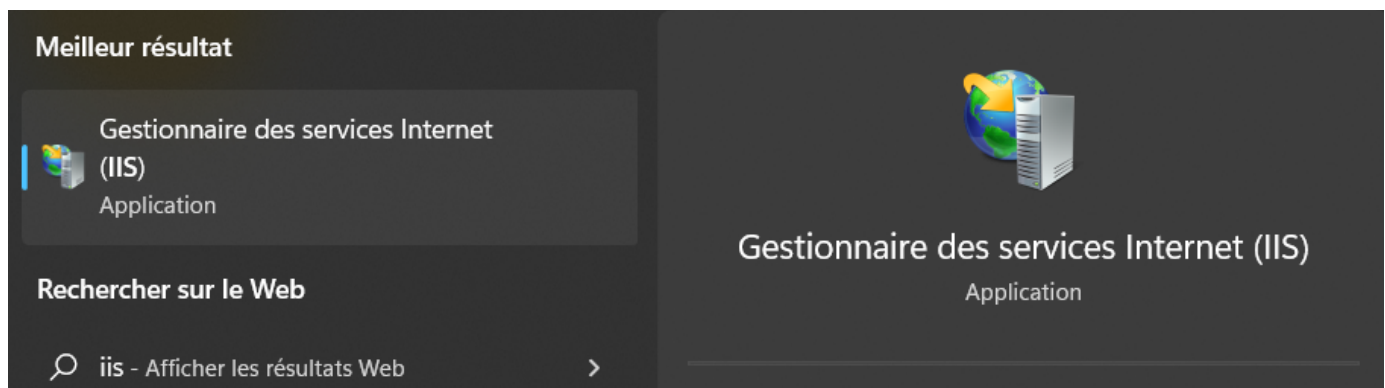
File	Description
assets	The folder contains resources copied from the Angular CLI assets configuration.
index.html	index.html file is the entrypoint for the application.
main.[hash].js	The file contain bundled application.
polyfill.[hash].bundle.js	It contains polyfill dependencies
runtime-[es-version].[hash].bundle.js	It contains webpack loader

File	Description
style.[hash].bundle.css	It contains the style definitions

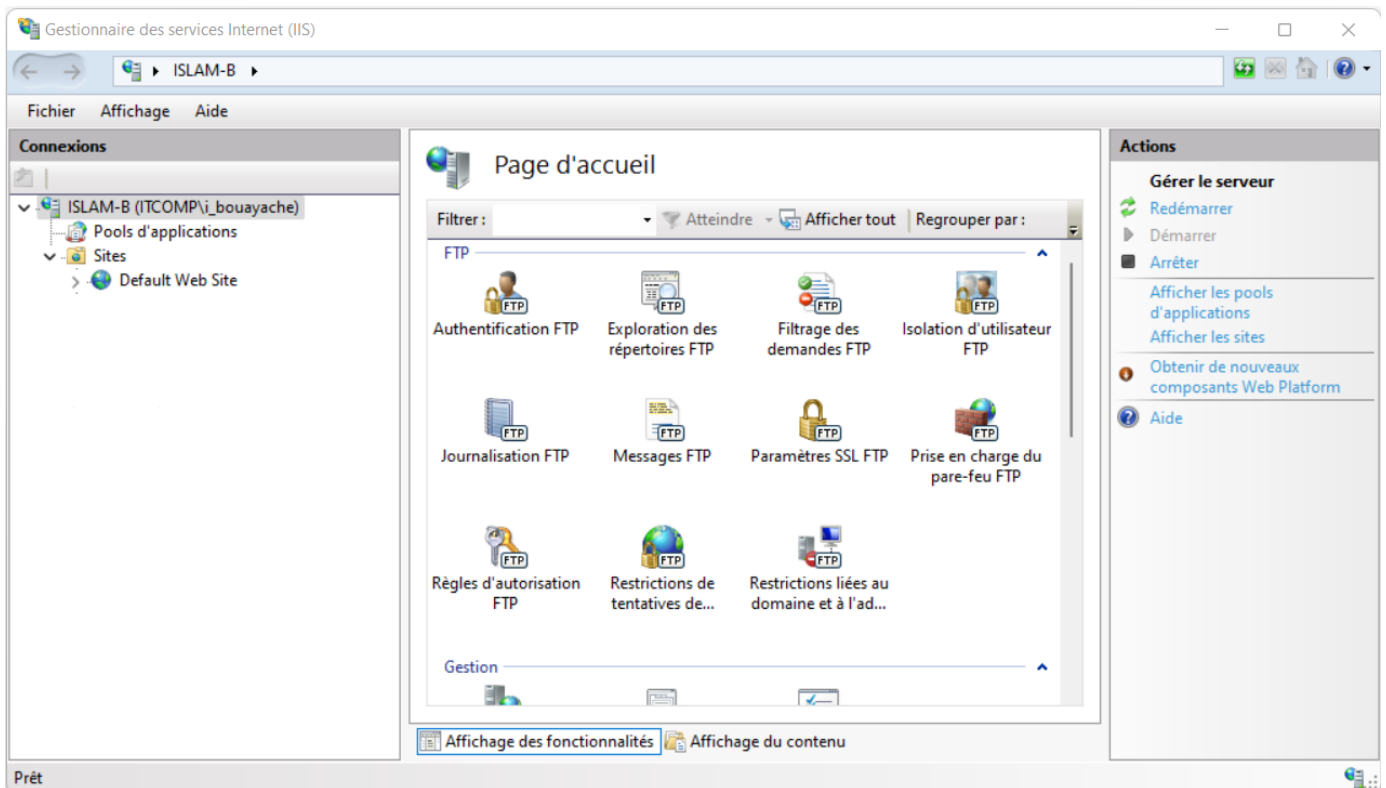
Serving the app

Now that we have configured Angular to build the files, we need to go in IIS Manager and add a new application.

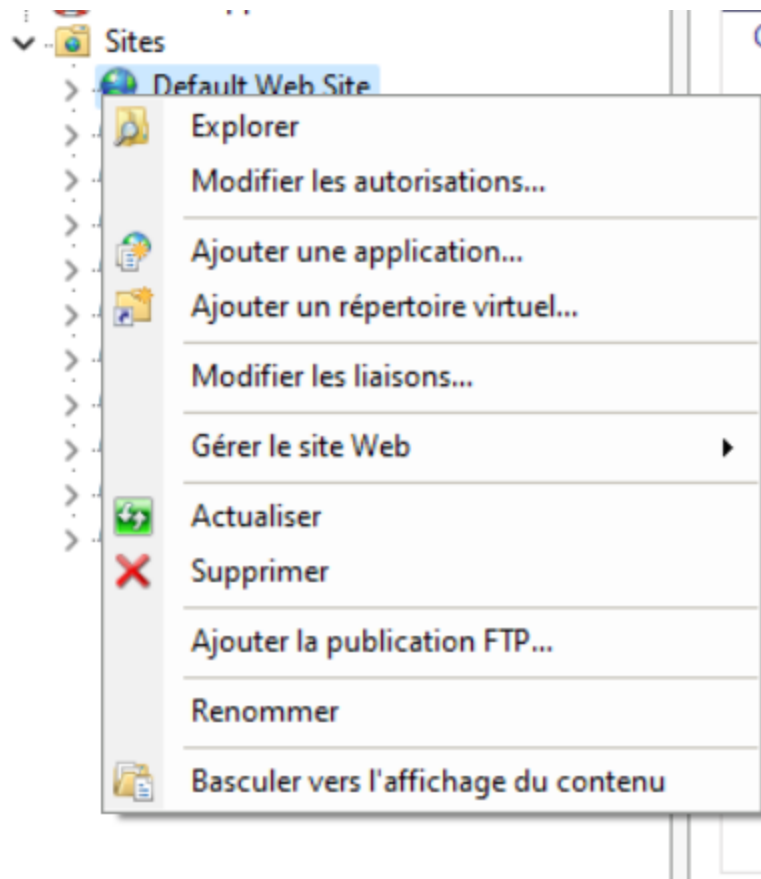
1. Copy the content of `dist/task-manager` folder to `C:/inetpub/wwwroot/task-manager`
2. Simply search for **IIS Manager** in the Windows search bar and open it.



3. When opened, expand the main menu (name of the local machine on the left site) then expand **sites**



4. Right Click on Default Web Site > Add Application



5. For the Alias field, type task-manager

6. For the **Physical path** choose the folder `C:/inetpub/task-manager`

Ajouter une application

Nom du site : Default Web Site
Chemin d'accès : /

Alias :
[Empty text box]

Pool d'applications : DefaultAppPool [Sélectionner...]

Exemple : ventes

Chemin d'accès physique :
[Empty text box] ...

Authentification directe
[Se connecter en tant que...] [Tester les paramètres...]

☐ Activer le préchargement

[OK] [Annuler]

7. Click on **Ok**

Now, the application is hosted and running, open the browser and navigate to <http://localhost/task-manager>.