

04_Packages et NPM

Les Packages et NPM

Introduction

Les packages en JavaScript sont des collections de fichiers de code réutilisables qui peuvent être facilement partagés et utilisés dans différents projets. NPM (Node Package Manager) est l'outil le plus utilisé pour gérer ces packages.

Table des Matières

1. [Qu'est-ce qu'un Package ?](#)
2. [Comment Fonctionne un Package ?](#)
3. [Importer un Package](#)
4. [Qu'est-ce que NPM ?](#)
5. [Utilisation de NPM](#)
6. [Ressources Utiles](#)

Qu'est-ce qu'un Package ?

Un package en JavaScript est un module ou une bibliothèque qui contient du code réutilisable dans des applications. Ces packages sont souvent publiés dans un registre, comme le registre NPM, où ils peuvent être facilement installés et utilisés.

Structure d'un Package

Un package typique contient :

- **package.json** : Un fichier de configuration qui décrit le package, ses dépendances, et ses scripts.
- **index.js** : Le point d'entrée du package.
- **README.md** : Un fichier de documentation.

Exemple de package.json

```
{
  "name": "mon-package",
  "version": "1.0.0",
  "description": "Un exemple de package",
  "main": "index.js",
  "scripts": {
    "start": "node index.js"
  },
  "dependencies": {
    "express": "^4.17.1"
  },
  "author": "Votre Nom",
  "license": "ISC"
}
```

Comment Fonctionne un Package ?

Un package fonctionne en encapsulant du code dans un module réutilisable. Ce module peut ensuite être importé et utilisé dans différentes parties d'une application ou dans d'autres projets.

Création d'un Package

1. Créer un nouveau dossier pour le package.
2. Initialiser un fichier `package.json` :

```
npm init -y
```

3. Ajouter le code dans un fichier `index.js` ou un autre fichier JavaScript.

Exemple de Code dans un Package

```
// index.js
module.exports = {
  saluer: function() {
    return "Bonjour, monde!";
  }
};
```

Importer un Package

Importation d'un Package Local

Pour importer un package local dans un projet, utiliser `require` :

```
const monPackage = require('./chemin/vers/mon-package');
console.log(monPackage.saluer());
```

Importation d'un Package depuis NPM

Pour importer un package depuis NPM, il faut d'abord l'installer puis l'utiliser dans le code.

Installation d'un Package NPM

```
npm install express
```

Utilisation du Package

```
const express = require('express');
const app = express();

app.get('/', (req, res) => {
  res.send('Bonjour, monde!');
});

app.listen(3000, () => {
  console.log('Serveur en cours d'exécution sur le port 3000');
});
```

Qu'est-ce que NPM ?

NPM (Node Package Manager) est le gestionnaire de packages par défaut pour Node.js. Il permet de gérer les dépendances d'un projet, d'installer des packages depuis le registre NPM, et de publier des packages.

Fonctionnalités de NPM

- **Installation de packages** : Installer des packages locaux ou globaux.
- **Gestion des dépendances** : Gérer les dépendances d'un projet via `package.json`.
- **Scripts** : Exécuter des scripts définis dans `package.json`.
- **Publication** : Publier des packages sur le registre NPM.

Utilisation de NPM

Installation de NPM

NPM est installé automatiquement avec Node.js. Vérifier si NPM est installé en exécutant :

```
npm -v
```

Commandes de Base de NPM

Initialiser un Projet

```
npm init
```

Installer un Package

```
npm install <nom-du-package>
```

Installer un Package en Tant que Dépendance de Développement

```
npm install <nom-du-package> --save-dev
```

Désinstaller un Package

```
npm uninstall <nom-du-package>
```

Mettre à Jour les Packages

```
npm update
```

Exécuter des Scripts Définis dans `package.json`

```
npm run <nom-du-script>
```

Ressources Utiles

- [NPM Documentation](#)
- [Node.js Documentation](#)
- [MDN Web Docs](#)