Le fichier package.json

1. Objectifs

- Comprendre ce qu'est réellement le fichier **package.json** et ses caractéristiques

2. Qu'est-ce que package.json

- Ce fichier est un guichet unique pour vous permettre de connaître toutes les informations sur le paquet que vous traitez actuellement. C'est un fichier JSON avec un tas de propriétés et chaque propriété nous dit quelque chose sur le package.
- Ce fichier JSON est présent à la racine du package et contient des métadonnées sur le package. Node utilise en fait ce fichier pour gérer la gestion des versions de notre projet, gérer les dépendances. Il contient également des informations pour identifier le projet lui-même, comme le nom, la description et la version du package. Parlons de ces propriétés en détail.
- Un **package.json** est un fichier **JSON** qui existe à la racine d'un projet Javascript/Node. Il contient des métadonnées pertinentes pour le projet et il est utilisé pour gérer les dépendances du projet, les scripts, la version et bien plus encore.
- Le fichier **package.json** est le centre de tout projet **Node.js** ou package npm. Il stocke des informations sur votre projet, de la même manière que la section >head< d'un document HTML décrit le contenu d'une page Web. Il se compose d'un seul objet JSON où les informations sont stockées dans des paires clé-valeur. Il n'y a que deux champs obligatoires; "**nom**" et "**version**", mais il est recommandé de fournir des informations supplémentaires sur votre projet qui pourraient être utiles aux futurs utilisateurs ou mainteneurs.

3. Créer un fichier package.json

- Il existe deux façons de créer/générer un fichier package.json.

- **Npm ou Yarn**

1. **Npm**

   - Pour créer le fichier package.json avec `npm` exécuter la commande `npm init` sur votre terminal

apcpedagogie.com

- Pour utiliser cette méthode, vous devez avoir Node installé sur votre système.

2. **Yarn**

   - Pour créer le fichier package.json avec `yarn`, exécuter-le `yarn init` sur votre terminal.
   - Pour utiliser la méthode **yarn**, vous devez avoir Node and **Yarn** installés sur votre système.

- **Manuellement**

  - Pour créer le fichier manuellement, créez un package.json fichier manuellement à la racine du projet et ajoutez un objet vide avec les propriétés requises, à savoir le **nom** et la **version**.

```
{
  "name": "apcpedagogie.com",
  "version": "1.0.0"
}
```

4. Propriétés de package.json

1. **Nom**

   - La propriété `name` est un **champ obligatoire** dans le package.jsonfichier, elle représente le nom du projet. Il existe des règles que vous devez suivre lorsque vous spécifiez un nom pour votre projet dans le fichier package.json.
   - Il existe quelques règles concernant la dénomination de votre package npm:
     - Cela devrait être unique.
     - Il ne doit pas être identique aux modules principaux de npm.
     - Il doit être inférieur ou égal à 214 caractères.
     - Le nom ne peut pas commencer par n trait de soulignement (_) ou un point (**.**).
     - Les nouveaux packages ne doivent pas comporter de lettres majuscules dans le nom.
     - Le nom finit par faire partie d'une URL, d'un argument sur la ligne de commande et d'un nom de dossier. Par conséquent, le nom ne peut contenir aucun caractère non sécurisé pour les URL.

apcpedagogie.com

- Vous devez de préférence choisir un nom plus court et facile à rappeler afin que l'utilisateur trouve plus facile d'utiliser votre package.

2. **version**

   - La `version` est également un **champ obligatoire** dans le fichier **package.json**.
   - La propriété indique la version actuelle du module pour le projet.
   - La règle requise pour le champ de version est qu'il doit suivre les directives de gestion des versions sémantiques.

```
"version" : "1.0.0"
```

3. Le "`name`" et la "`version`" forment ensemble un identifiant supposé totalement unique. Vous devrez mettre à jour la version chaque fois qu'une modification significative est apportée au package.
4. Le versionnage sémantique suggère que la version de votre package doit commencer par **1.0.0**. Après la première version, vous pouvez suivre le principe ci-dessous lors de la mise à jour de la version du package:
   1. **Version du correctif**: les correctifs de bogues incrémentent le troisième chiffre à condition qu'il soit rétrocompatible et que vous n'introduisiez aucun changement de rupture. Exemple: **1.0.1** Version
   2. **mineure**: la nouvelle fonctionnalité incrémente le deuxième chiffre à condition qu'elle soit rétrocompatible et que vous n'introduisiez pas de changements de rupture. Exemple: **1.1.0** Version
   3. **majeure**: les modifications qui interrompent la rétrocompatibilité incrémentent le premier chiffre Exemple: **2.0.0**

5. **La description (description)**

   - La propriété description est utilisée pour décrire et fournir plus d'informations sur le projet.
   -
```
"description" : "Maîtrise du fichier package.json"
```

6. **La dépendances (dependencies)**

   - La propriété dependencies indique la liste des **modules/packages** requis pour que votre application fonctionne. Après avoir installé une dépendance, elle est ajoutée à la liste des dépendances.

apcpedagogie.com

```
"dépendances" : {
  "express": "^4.16.2",
 "package": "^1.0.1"
}
```

- Pour installer une dépendance, exécutez `npm i package` ou `npm install package` sur votre terminal. Où le package est le nom du package que vous essayez d'installer.
- Utiliser `--save` à la fin de la commande **install** pour ajouter une entrée de dépendance dans **package.json** de votre application.
- Par exemple, la commande suivante installera ExpressJS dans votre application et ajoutera également une entrée de dépendance dans **package.json**: `npm install express --save`.

7. **devDependencies**

- Pour spécifier les packages dont dépend votre projet, vous devez les lister en tant que "**dependencies**" ou "**devDependencies**" dans le fichier **package.json** de votre package .
- "dependencies": Packages requis par votre application en production.
- "devDependencies": Packages qui ne sont nécessaires que pour le développement et les tests locaux.
- Vous pouvez ajouter des dépendances à un fichier **package.json** à partir de la ligne de commande ou en modifiant manuellement le fichier `package.json`.
- Pour installer un **devDependency**, exécutez `npm i package --save-dev` ou `npm install package -D` sur votre terminal. Où le package est le nom du package que vous essayez d'installer.

8. **Les scripts (scripts)**

- La propriété **scripts** prend un objet JSON de paires **clé/valeur**. Chaque script peut être utilisé pour effectuer différents ensembles de tâches, telles que la création, le test, le linting de l'application.
- Vous pouvez exécuter les scripts en exécutant `npm run script name` ou `yarn script name` sur le terminal.

9. **main**

- La propriété `main` sert de point d'entrée à votre application et doit pointer vers le fichier qui sert de point d'entrée à votre application

apcpedagogie.com

- ```
  "main": "index.js",
  ```

10. **privée (private)**

    - La propriété `private` est **false** par défaut, mais peut être définie sur **true** pour empêcher la publication de l'**application/du package**.
    - ```
      "private": true,
      ```

11. **licence (license)**

    - Cette propriété indique le type de licence utilisé par le projet
    - Cela contient l'utilisation des informations sur vos packages. Il aide l'utilisateur à savoir si votre projet a des restrictions lors de son utilisation.
    - ```
      "licence" : "MIT"
      ```

12. **bugs**

    - La propriété `bugs` est utilisée pour pointer vers la page des problèmes du référentiel de l'application ou partout où les problèmes du projet peuvent être signalés.
    - Contient généralement une chaîne ou un objet pour signaler ou suivre les problèmes. Ceux-ci sont utiles pour les personnes qui rencontrent des problèmes avec votre package.
    - ```
      "bugs" : {
          "url" : "https://github.com/Easybuoy/package.json-mastery/issues"
        }
      ```

Sommaire du cours Node.js

Source : https://apcpedagogie.com/le-fichier-package-json · téléchargé le 05/11/2026 · © APC Pédagogie · apcpedagogie.com