GRUNT WATCH

INSTALLATION GUIDE

IDEO-LAB - 2025

AGENDA

1. Installez Node.js et npm.
2. Installez Grunt CLI globalement.
3. Initialisez votre projet avec `npm init`.
4. Installez Grunt et `grunt-contrib-watch` localement.
5. Créez et configurez le fichier `Gruntfile.js`.
6. Exécutez Grunt avec la commande `grunt` pour surveiller vos fichiers et automatiser des tâches.

Étape 1 : Installer Node.js et NPM

Grunt repose sur Node.js. Il faut donc commencer par installer Node.js et npm (qui est le gestionnaire de packages pour Node.js).

1. Rendez-vous sur nodejs.org et téléchargez la dernière version stable (LTS) de Node.js.
2. Installez Node.js en suivant les instructions selon votre système d'exploitation (Windows, macOS ou Linux).
3. Une fois installé, vérifiez la version de Node.js et npm pour vous assurer qu'ils fonctionnent correctement :

bash Copier le code

```
node -v
npm -v
```

Cela devrait afficher les versions de Node.js et npm installées.

Étape 2 : Installer Grunt CLI

Grunt CLI (Command Line Interface) est l'interface en ligne de commande pour Grunt. Cela permet de lancer Grunt globalement dans tous vos projets sans avoir à l'installer à chaque fois localement.

Installez `grunt-cli` globalement :

bash Copier le code

```
npm install -g grunt-cli
```

Cela installe le CLI de Grunt, mais vous devrez toujours installer Grunt localement dans chaque projet.

Étape 3 : Initialiser votre projet avec NPM

Si vous n'avez pas encore de fichier `package.json`, initialisez votre projet en exécutant la commande suivante dans le répertoire de votre projet :

bash Copier le code

```
npm init
```

Cette commande vous posera une série de questions pour configurer le fichier `package.json` (le fichier qui contient les informations sur votre projet et ses dépendances). Si vous voulez utiliser les valeurs par défaut, vous pouvez simplement appuyer sur "Entrée" pour chaque question. Un fichier `package.json` sera généré à la fin du processus.

Étape 4 : Installer Grunt et grunt-contrib-watch localement

Ensuite, vous devez installer Grunt ainsi que le plugin `grunt-contrib-watch` dans votre projet.

1. Installez Grunt localement avec :

   ```bash
   npm install grunt --save-dev
   ```

   Cette commande installe Grunt comme une dépendance de développement (c'est pourquoi on utilise l'option `--save-dev`). Cela signifie que Grunt sera utilisé uniquement pour le développement et ne sera pas inclus dans la version de production du projet.

2. Installez ensuite le plugin `grunt-contrib-watch` qui est utilisé pour surveiller les changements dans les fichiers :

   ```bash
   npm install grunt-contrib-watch --save-dev
   ```

3. (Optionnel) Si vous souhaitez ajouter d'autres tâches (comme `jshint` pour analyser les fichiers JavaScript), installez-les également :

   ```bash
   npm install grunt-contrib-jshint --save-dev
   ```

Étape 5 : Créer le fichier Gruntfile.js

Le fichier `Gruntfile.js` est au cœur de la configuration de Grunt. Il contient toutes les instructions nécessaires pour que Grunt sache quoi faire, comme quelles tâches exécuter, quels fichiers surveiller, etc.

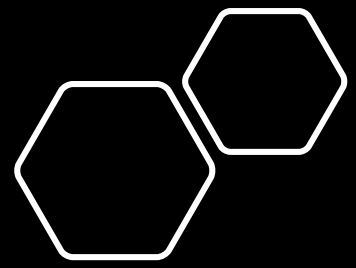
Créez un fichier `Gruntfile.js` à la racine de votre projet avec le contenu suivant :

js — Copier le code

```
module.exports = function(grunt) {
  // Configuration du projet
  grunt.initConfig({
    // Configuration de la tâche watch
    watch: {
      scripts: {
        files: ['js/**/*.js'],  // Spécifiez les fichiers à surveiller (dans ce cas, tous
        tasks: ['jshint'],  // Spécifiez les tâches à exécuter lorsqu'un fichier est modi
        options: {
          spawn: false,  // Rend la tâche watch plus rapide en évitant de redémarrer Grun
        },
      },
    },
    // Configuration de la tâche jshint pour analyser les fichiers JavaScript
    jshint: {
      all: ['Gruntfile.js', 'js/**/*.js']  // Analyser les fichiers .js dans le dossier j
    }
  });

  // Charger les plugins nécessaires
  grunt.loadNpmTasks('grunt-contrib-watch');
  grunt.loadNpmTasks('grunt-contrib-jshint');

  // Définir les tâches par défaut
  grunt.registerTask('default', ['watch']);
};
```



Explication du `Gruntfile.js` :

1. `grunt.initConfig()` : C'est ici que vous configurez vos tâches. Vous définissez quelle tâche doit surveiller quels fichiers et quelles actions doivent être effectuées lors de la modification de ces fichiers.
2. **Tâche `watch` :**
   - `files` : Cela spécifie quels fichiers doivent être surveillés. Ici, tous les fichiers `.js` dans le dossier `js/` sont surveillés.
   - `tasks` : Cela définit les tâches qui doivent être exécutées lorsqu'une modification est détectée. Ici, nous utilisons la tâche `jshint` pour analyser les fichiers JavaScript.
   - `options` : Des options supplémentaires peuvent être définies. L'option `spawn: false` permet d'accélérer le processus de surveillance.
3. **Tâche `jshint` :** Cette tâche analyse les fichiers JavaScript pour s'assurer qu'ils respectent certaines règles de style et ne contiennent pas d'erreurs.
4. **Chargement des plugins :** Ici, nous chargeons les plugins `grunt-contrib-watch` et `grunt-contrib-jshint` afin que Grunt puisse utiliser ces tâches.
5. **Tâche par défaut :** En définissant `grunt.registerTask('default', ['watch']);`, nous indiquons que lorsque vous exécutez la commande `grunt`, la tâche par défaut sera `watch`.

Étapes supplémentaires (facultatif) :

- **Ajouter d'autres tâches** : Vous pouvez ajouter d'autres tâches à Grunt pour automatiser plus de processus, comme la minification des fichiers, la compilation de fichiers Sass, etc.

  Par exemple, pour surveiller les fichiers Sass, vous pouvez installer `grunt-contrib-sass` et ajouter la configuration suivante au `Gruntfile.js` :

```bash
npm install grunt-contrib-sass --save-dev
```

  Puis, modifiez le `Gruntfile.js` :

```js
sass: {
  dist: {
    files: {
      'css/main.css': 'sass/main.scss'  // compile le fichier .scss en .css
    }
  }
},

watch: {
  scripts: {
    files: ['js/**/*.js'],
    tasks: ['jshint'],
    options: {
      spawn: false,
    },
  },
  sass: {
    files: ['sass/**/*.scss'],
    tasks: ['sass'],  // compile le SCSS à chaque modification
  },
}
```