

# Runtimes Java dans Visual Studio Code

Pour utiliser `java.configuration.runtimes` dans les paramètres de Visual Studio Code (VS Code), vous devez le configurer dans votre fichier `settings.json`. Ce paramètre vous permet de spécifier plusieurs versions du Java Development Kit (JDK) que VS Code peut utiliser pour vos projets Java. C'est particulièrement utile lorsque vous travaillez avec différents projets nécessitant des versions spécifiques de Java. Voici un guide étape par étape pour le configurer :

## Étapes pour configurer `java.configuration.runtimes`

### 1. Ouvrir les paramètres de VS Code :

- Sous Windows/Linux : Allez dans `File > Preferences > Settings` ou appuyez sur `Ctrl + ,`.
- Sous macOS : Allez dans `Code > Preferences > Settings` ou appuyez sur `Cmd + ,`.

### 2. Accéder au fichier de paramètres JSON :

- Dans l'interface des paramètres, recherchez `java.configuration.runtimes`.
- Vous verrez une option comme "Java: Configuration: Runtimes". Cliquez sur "Edit in settings.json" (généralement un lien sous la description du paramètre) pour ouvrir le fichier `settings.json`.

### 3. Modifier `settings.json` :

- Dans le fichier `settings.json`, ajoutez ou modifiez le tableau `java.configuration.runtimes`. Ce tableau contient des objets, chacun représentant une version du JDK que VS Code doit reconnaître.
- Chaque objet inclut généralement :
  - `name` : L'identifiant de la version Java (par exemple, `JavaSE-1.8`, `JavaSE-11`, `JavaSE-17`).
  - `path` : Le chemin absolu vers le répertoire d'installation du JDK sur votre système.
  - `default` (optionnel) : Définissez-le sur `true` pour en faire le JDK par défaut pour les dossiers non gérés (projets sans outils de construction comme Maven ou Gradle).

Voici un exemple de configuration :

```
{
  "java.configuration.runtimes": [
    {
      "name": "JavaSE-1.8",
      "path": "C:/Program Files/Java/jdk1.8.0_351",
      "default": true
    },
    {
      "name": "JavaSE-11",
```

```

        "path": "C:/Program Files/Java/jdk-11.0.15"
    },
    {
        "name": "JavaSE-17",
        "path": "C:/Program Files/Java/jdk-17.0.6"
    }
]
}

```

#### 4. Vérifier les chemins des JDK :

- Assurez-vous que le `path` pointe vers le répertoire racine de votre installation JDK (par exemple, là où se trouve le dossier `bin` contenant `java.exe` ou `java`).
- Sous Windows, utilisez des barres obliques (/) ou échappez les barres obliques inverses (\\) dans le chemin.
- Sous macOS/Linux, utilisez le chemin du système de fichiers approprié (par exemple, `/usr/lib/jvm/java-17-openjdk`).

#### 5. Enregistrer et recharger :

- Enregistrez le fichier `settings.json`.
- Redémarrez VS Code ou rechargez la fenêtre (Ctrl + R ou Cmd + R) pour appliquer les modifications.

#### 6. Vérifier la configuration :

- Ouvrez la palette de commandes (Ctrl + Shift + P ou Cmd + Shift + P) et exécutez la commande `Java: Configure Java Runtime`.
- Cela ouvre une vue montrant les JDK disponibles pour vos projets. Vérifiez que vos runtimes configurés apparaissent sous l'onglet "Project JDKs".

### Fonctionnement

- **Dossiers non gérés** : Pour les projets sans outils de construction (par exemple, fichiers Java simples), VS Code utilise le JDK par défaut spécifié dans `java.configuration.runtimes`.
- **Projets gérés (Maven/Gradle)** : Pour les projets avec des outils de construction, la version du JDK est déterminée par la configuration de construction (par exemple, `pom.xml` ou `build.gradle`), mais VS Code reconnaîtra toujours les runtimes listés ici pour la compatibilité.
- **Serveur de langage** : Le serveur de langage Java (partie des extensions Java) nécessite JDK 17+ pour fonctionner. Vous pouvez le spécifier séparément avec `java.jdt.ls.java.home` si nécessaire (par exemple, `"java.jdt.ls.java.home": "C:/Program Files/Java/jdk-17.0.6"`).

## Exemple avec plusieurs JDK

Si vous êtes sur un Mac et avez installé JDK 8 et 17 via Homebrew, votre configuration pourrait ressembler à ceci :

```
{
  "java.configuration.runtimes": [
    {
      "name": "JavaSE-1.8",
      "path": "/Library/Java/JavaVirtualMachines/jdk1.8.0_351.jdk/Contents/Home",
      "default": true
    },
    {
      "name": "JavaSE-17",
      "path": "/Library/Java/JavaVirtualMachines/jdk-17.0.6.jdk/Contents/Home"
    }
  ]
}
```

## Conseils

- Utilisez le bon format de `name` (JavaSE-<version>), correspondant à la version Java (par exemple, JavaSE-11 pour Java 11).
- Si vous n'avez qu'un seul JDK, vous pouvez toujours le lister pour vous assurer que VS Code l'utilise explicitement.
- Pour trouver votre chemin JDK, exécutez `java -version` dans un terminal et vérifiez le répertoire d'installation (par exemple, via `where java` sous Windows ou `which java` sous les systèmes Unix-like, puis remontez à la racine du JDK).

Cette configuration vous offre la flexibilité de travailler avec plusieurs versions de Java dans VS Code sans problème ! Faites-moi savoir si vous avez besoin d'aide pour trouver vos chemins JDK ou pour résoudre des problèmes.