



BO - HUB

B-INN-000

Workshop Plugin Minecraft

Créer son premier plugin Minecraft





Workshop Plugin Minec

language: java
build tool: gradle



- The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.

WORKSHOP PLUGIN MINECRAFT

PRÉ-REQUIS

Si vous n'avez pas déjà installé les outils nécessaires à ce workshop suivez les instructions situées ici :

<https://github.com/alwyn974/workshop-mc-plugin/blob/master/REQUIREMENTS.md>

CONFIGURATION

CRÉATION D'UNE TASK GRADLE

Pour pouvoir tester votre plugin, vous devez créer une task gradle, pour déplacer le plugin dans le dossier `server/plugins/`

Cette task aura pour nom `buildAndMove` devra dépendre de `build` et respecter les prérequis suivants :

- Etre exécuté en dernier
- Utiliser la task `copy` pour copier le plugin généré par le build dans le dossier `./server/plugins/`



Task avec gradle `Script` basique, Doc de la classe `Task`



Utilisez `String.format` pour récupérer le nom du jar plus facilement, avec `archivesBaseName` et `version`



CRÉATION DE LA CLASSE PRINCIPALE

Créez votre propre package, moi, ça sera `re.alwyn974.plugin.workshop`, dans ce package créer une classe `PluginNamePlugin` (ex: `WorkshopPlugin`).

Cette classe devra hériter de `JavaPlugin` de `org.bukkit.plugin.java` et surcharger les méthodes `onEnable()` et `onDisable()`

Dans chacune des méthodes ajouter simplement un message dans la console.

CRÉATION DU FICHIER `plugin.yml`

Dans le dossier `src/resources` créez le fichier `plugin.yml` avec le contenu suivant :

```
name: Plugin Name
version: 1.0
author: Your Name
main: com.github.pluginname.Main
```

Remplacez le `main` par votre classe qui hérite de `JavaPlugin` et le `name` par le nom de votre plugin.



Plus d'informations sur le `plugin.yml`

BUILD

Pour tester votre plugin, vous devez exécuter la task gradle `buildAndMove` et ensuite lancer le docker-compose. Si votre plugin est reconnu dans la console vous aurez quelque chose comme ceci :

```
Terminal
~/B-INN-000> Affichage dans la console du serveur
[10:59:24 INFO]: [WorkshopPlugin] Loading WorkshopPlugin v1.0-SNAPSHOT
[10:59:24 INFO]: [WorkshopPlugin] Enabling WorkshopPlugin v1.0-SNAPSHOT
```

UTILISATION D'EVENTS

Comme avec Forge, les plugins utilisent aussi des events.

Créez une classe `Events` qui implementera `Listener` de `org.bukkit.event`.

Pour chaque méthode qui recevra un event il faudra ajouter une annotation spécifique.



Comment utiliser les events spigot [Documentation](#)

MODIFIER LE MESSAGE DE CONNEXION D'UN JOUEUR

En utilisant l'événement `PlayerJoinEvent` créez une méthode `onPlayerJoin` qui recevra un événement `PlayerJoinEvent`. Lorsqu'un joueur rejoindra le serveur il faudra afficher un nouveau message que `PlayerName` joined the game.

Exemple: `[+] alwyn974 - 1/20`

Si vous voulez ajouter des couleurs au message vous pouvez utiliser la classe `ChatColor` de `org.bukkit.ChatColor` ou utiliser les codes de couleur.



Color Codes. Javadoc de `ChatColor`

MODIFIER LE MESSAGE DE DÉCONNEXION D'UN JOUEUR

En utilisant l'événement `PlayerQuitEvent`

CRÉATION D'UNE COMMANDE SIMPLE
