

03 Přidání itemu

V této lekci si ukážeme, jak do hry přidat vlastní item.

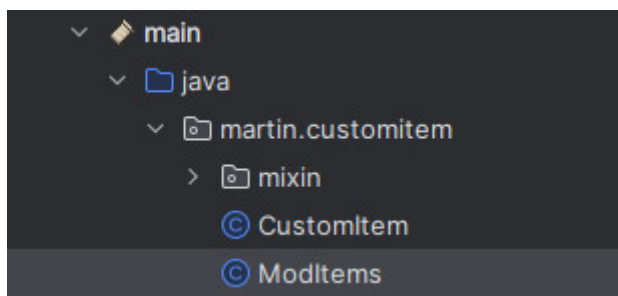
Abychom si usnadnili práci při přidávání itemů, tak si vytvoříme novou třídu `ModItems`, která bude obsahovat všechny itemy, které náš mód přidává.

Umístění třídy

Nové třídy budeme umisťovat do stejné složky, kde je naše hlavní třída `CustomItem`.
Složka by se měla jmenovat podle názvu balíčku, který jste si zvolili při vytváření módu - v mém případě tedy `martin.customitem`.

Třidu přidáme tak, že klikneme pravým tlačítkem myši na složku a zvolíme *New > Java Class*. Zobrazí se nám okno, kam zadáme název třídy - v našem případě tedy `ModItems`. Vytvoření třídy potvrdíme klávesou Enter.

Aktuálně by tedy struktura souborů měla vypadat následovně:



Připomenutí

Veškerý kód píšeme vždy mezi složené závorky `{}`

Jako první si přidáme metodu, která nám zajistí registraci itemu.

Ve třídě `ModItems` si tedy vytvoříme metodu `register`:

```
1  public static Item register(Item item, String id) {  
2  
3  }
```

- k této metodě můžeme přistupovat odkudkoliv `public static`
- jako hodnotu vrací registrovaný item `Item`

- a jako parametry budeme předávat item, který chceme registrovat a id, pod kterým chceme daný item registrovat
 - item bude typu `Item`
 - id bude textový řetězec - typ `String`

Identifikátor

Při registraci itemu si nejprve potřebujeme vytvořit identifikátor, pod kterým item registrujeme. Identifikátor je například `minecraft:stone`. Skládá se tedy ze dvou částí:

- `namespace`
 - to je část před `:` - používá se název módu
- `path`
 - to je část za `:` - mělo by to odpovídat anglickému názvu itemu

Vytvoříme si tedy proměnnou `itemID` typu `Identifier`, do které si uložíme identifikátor itemu. Ten vytvoříme pomocí metody `of` ze třídy `Identifier`. U této metody potřebujeme uvést právě `namespace` a `path`.

- `namespace` můžeme použít název módu, který máme uložený v proměnné `MOD_ID` ve třídě `CustomItem`.
- `path` bude parametr `id`, který jsme definovali v naší metodě `register`

```
1 Identifier itemID = Identifier.of(CustomItem.MOD_ID, id);
```

Přidání itemu do registrů

V Minecraftu se vše ukládá do tzv. *registrů*. Máme registry na itemy, bloky, kapaliny, enchanty, entity, dimenze, biomy atd. Všechny registry najdete na [Minecraft wiki](#). Do těchto registrů můžeme samozřejmě i přidávat nové věci. V této části si ukážeme, jak přidat nový item.

Vytvoříme proměnnou `registeredItem` typu `Item`, kam si uložíme registrovaný item. Samotná registrace se provádí pomocí metody `register` ze třídy `Registry`.

Jako první argument musíme specifikovat, do jakého registru chceme přidávat. Seznam všech registrů je ve třídě `Registries` - my přidáváme item, takže vybereme `ITEM`. Dále musíme uvést identifikátor itemu - proměnná `itemID`. A jako poslední samotný item, který chceme registrovat - parametr `item`.

```
1 Item registeredItem = Registry.register(Registries.ITEM, itemID, item);
```

Vrácení hodnoty

Jako poslední věc musíme vrátit registrovaný item, abychom ho po registraci mohli dále používat. To uděláme pomocí slova `return` a za to uvedeme hodnotu, kterou chceme vrátit. V našem případě proměnná `registeredItem`.

Celý kód bude vypadat následovně:

```
1 public static Item register(Item item, String id) {
2     Identifier itemID = Identifier.of(FabricDocsReference.MOD_ID, id);
3
4     Item registeredItem = Registry.register(Registries.ITEM, itemID,
5         item);
6     return registeredItem;
7 }
```

Tip

Pokud chcete, aby váš kód vypadal o něco lépe (správné odsazení textu, mezery, atd.), tak můžete použít klávesovou zkratku Ctrl+Alt+L

Registrování itemu

Nyní můžeme použít naši vytvořenou metodu k registraci itemu.

Ve třídě `ModItems` si vytvoříme proměnnou `MAGIC_DUST` typu `Item`

- hodnotu proměnné nebudeme chtít po vytvoření měnit a zároveň k ní chceme přistupovat odkudkoliv, proto proměnnou nastavíme jako `public static final`

Do proměnné nyní uložíme výsledek metody `register`, kde jako argument uvedeme item a id. Item musíme vytvořit nový. To uděláme pomocí slova `new` a názvu třídy (`Item`). Třída item ale potřebuje jako argument nastavení itemu. Opět tedy vytvoříme nový objekt, ale tentokrát to bude třída `Settings`. Třída je obsažena v balíčku `Item`, takže musíme použít `Item.Settings`. Item se základním nastavením tedy vytvoříme pomocí:

```
1 new Item(new Item.Settings())
```

Dále potřebujeme uvést ještě id itemu, která chceme registrovat. Můžeme použít stejný název jako jsme použili u názvu proměnné - `suspicious_substance`.

Celý kód registrace itemu a uložení do proměnné bude vypadat následovně:

```
1 public static final Item MAGIC_DUST = register(
```

```

2         new Item(new Item.Settings()),
3         "magic_dust"
4     );

```

Pokud bychom nyní spustili hru, tak se item ještě nepřidá, protože musíme v hlavní třídě našeho módu říct, že má použít naši třídu `ModItems`.

Na to si ve třídě `ModItems` vytvoříme metodu `initialize`, která se používá na provedení nastavení třídy. Tato metoda bude opět `public static`, abychom k ní mohli přistupovat odkudkoliv. A jako návratový typ nebude vracet nic, což se v programování označuje slovem `void`. Metoda zatím nebude obsahovat žádný kód a bude tedy vypadat následovně:

```

1 public static void initialize() {
2 }

```

Celá třída `ModItems` pak bude vypadat následovně:

```

1 public class ModItems {
2     public static final Item MAGIC_DUST = register(
3         new Item(new Item.Settings()),
4         "magic_dust"
5     );
6
7     public static void initialize() {
8     }
9
10    public static Item register(Item item, String id) {
11        Identifier itemID = Identifier.of(CustomItem.MOD_ID, id);
12
13        Item registeredItem = Registry.register(Registries.ITEM, itemID,
14        item);
15
16        return registeredItem;
17    }
18 }

```

V hlavní třídě našeho módu (`CustomItem`) pak naši třídu s itemy použijeme tak, že do metody `onInitialize` přidáme volání metody `initialize` ze třídy `ModItems`:

```

1 public class CustomItem implements ModInitializer {
2     public static final String MOD_ID = "custom-item";
3
4     public static final Logger LOGGER =
5     LoggerFactory.getLogger(MOD_ID);
6
7     @Override
8     public void onInitialize() {
9         LOGGER.info("Hello Fabric world!");
10    }
11 }

```

```
9      ModItems.initialize();
10    }
11  }
```

Když nyní mód spustíme, tak bychom item měli mít ve hře. Item zatím nemá žádnou texturu ani název a jediný způsob, jako ho nyní můžeme získat je přes příkaz:

```
1  /give @s custom-item:magic_dust
```