04 Přidání itemu část 2

Přidání itemu do kreativního inventáře

Zatím item můžeme získat jen pomocí příkazu `/give`. Nyní si item přidáme do inventáře v kreativním módu.

Item musíme přidat vždy do nějaké záložky v inventáři. Můžeme si také vytvořit vlastní záložku, ale zatím si vystačíme se základními, co už jsou ve hře. Přidání itemu funguje tak, že když se hra spouští, tak "oznámí" všem módům "vytvářím záložky s itemy" a každý mód může říct, co chce do jaké záložky přidat. Těmto oznámením se říká *události*.

Přidávání itemů do inventáře je jedna z věcí, kterou nám zjednodušuje *Fabric* pomocí třídy `ItemGroupEvents`.

Kód pro přidání itemu budeme psát do metody `initialize()` ve třídě s itemy (`ModItems`) a bude vypadat následovně:

```
1  ItemGroupEvents.modifyEntriesEvent(ItemGroups.INGREDIENTS)
2          .register((itemGroup) -> itemGroup.add(ModItems.MAGIC_DUST));
```

Pomocí metody `modifyEntriesEvent` ze třídy `ItemGroupEvents` si získáme požadovanou skupinu itemů. V tomto případě skupinu `INGREDIENTS` (pomocí nápovědy si můžete zjistit další skupiny). A do této skupiny pomocí metody `register` přidáme náš item. Seznam všech skupin je uložen ve třídě `ItemGroups`.

Části kódu `(itemGroup) -> itemGroup.add(ModItems.MAGIC_DUST)` se říká **lambda výraz** a funguje podobně jako metoda, ale nemá žádné jméno. Lambda výraz má tvar `parametr -> výraz` případně můžeme použít tvar `parametr -> { blok kódu }`. Funguje stejně jako metoda, takže přijímá parametr, poté se provede nějaká akce a na konec může vrátit nějakou hodnotu. Název parametru si můžeme zvolit vlastní, ale měl by dávat smysl pro to, k čemu jej používáme.
V tomto případě máme parametr `itemGroup`, který má typ `FabricItemGroupEntries` (tato třída nám umožňuje upravovat skupinu itemů). Máme na výběr několik metod:

- `add(item)`
  - přidá item na konec seznamu
- `addAfter(afterItem, item)`
  - přidá item za `afterItem`
- `addBefore(beforeItem, item)`
  - přidá item před `beforeItem`

Konkrétní item získáme ze třídy `Items`, která obsahuje jako vlastnosti seznam všech itemů v Minecraftu. Jednotlivé itemy získáme pomocí jejich `ID` v Minecraftu, které musí být napsané velkými písmeny, protože se jedná o konstantní (neměnné) hodnoty (`static final`).
Například následujícím kódem přidáme náš `CUSTOM_ITEM` za item `PIGLIN_BANNER_PATTERN`.

```
1  ItemGroupEvents.modifyEntriesEvent(ItemGroups.INGREDIENTS).register(group
   ->
2          group.addAfter(Items.PIGLIN_BANNER_PATTERN, ModItems.MAGIC_DUST));
```

Při použití metod `addAfter` a `addBefore` je důležité si ověřit, že vybraný item, za nebo před který chceme přidávat v této skupině opravdu je.

Název itemu

Aktuálně se náš item jmenuje `item.custom-item.magic_dust`, což asi není úplně název, který by hráč chtěl vidět. Proto si ukážeme, jak item pojmenovat.

Názvy itemů se ukládají do souborů typu `json`. Do stejných souborů se ukládají i další věci v Minecraftu např. crafting recepty.

JSON soubor

Java**S**cript **O**bject **N**otation soubory se používají na ukládání a výměnu dat v mnoha programovacích jazycích. Jsou jednoduché nejen na vytvoření, ale díky jejich struktuře je také jednoduché je číst.

Soubory se skládají z dvojic **vlastností** a **hodnot**. Jak vlastnosti, tak i hodnoty se píší do uvozovek a mezi vlastnost a hodnotu se píše dvojtečka `:`. Samotné vlastnosti se oddělují čárkou `,`.
Jak už z názvu vyplývá, tak JSON se používá k popisu objektů. Samotné objekty se píší do složených závorek `{}`. Celý JSON soubor je také objekt, takže se celý obsah píše také do složených závorek.

> ⓘ **Složené závorky** `{}`
>
> `{` `Ctrl` + `Alt` + `B`
> `}` `Ctrl` + `Alt` + `N`
>
> - místo `Ctrl` + `Alt` můžeme použít pravý `Alt` (někdy označený také jako `AltGr`)

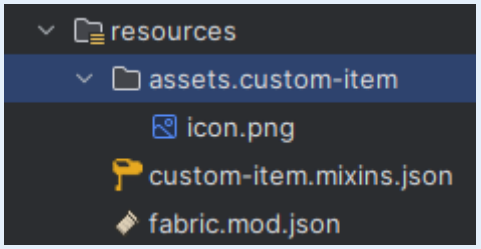
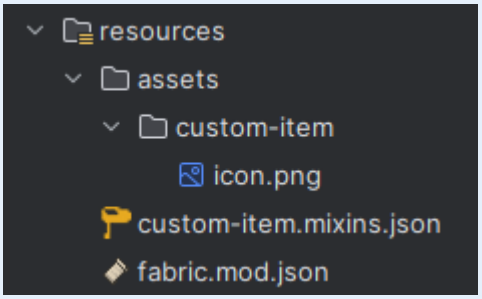
- Příklad:

```
{
        "vlastnost1":"hodnota1",
        "vlastnost2":"hodnota2"
}
```

Soubory, které nejsou kód ukládáme do složky `resources`. Soubory s překladem pak ukládáme konkrétně do složky `resources/assets/custom-item/lang` (`custom-item` je název módu).

ⓘ Zobrazení složek v IntelliJ IDEA

Editor nám složky zobrazuje zjednodušeně oproti tomu, jak reálně struktura složek vypadá. Pokud máme například složku `assets` a v ní jen další složku `custom-item`, tak se nám tyto dvě složky zobrazí jako `assets.custom-item`

Zjednodušené zobrazení	Reálná struktura složek
	

Samotné `json` soubory s překlady jsou pojmenované zkratkou jazyka, pro který je překlad určen. Například tedy soubor pro americkou angličtinu se bude jmenovat `en_us.json`. Soubor s češtinou pak `cs_cz.json`. Pro každý jazyk tedy musíte mít zvlášť soubor.

Kódy pro ostatní jazyky

Kódy pro všechny jazyky najdete na Minecraft Wiki. Sloupec `Locale Code - In-game`.

Vytvoříme si složku `lang` a v ní soubor `en_us.json`.

ⓘ Vytvoření složky

Složky se tvoří stejně jako soubory. Klikneme pravým tlačítkem myši na složku a zvolíme *New* > *Directory*

Pro náš item bude obsah souboru vypadat následovně:

```
{
  "item.custom-item.magic_dust": "Magic Dust"
}
```

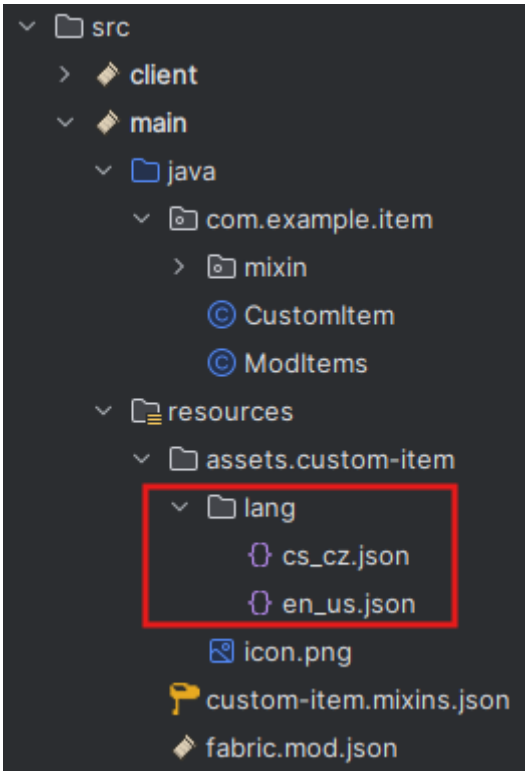
Vlastnosti uvádíme ve tvaru `<typ>.<namespace>.<path>`:

- `<typ>` je typ objektu např. `block`, `item`, `fluid`, `enchantment`
- `<namespace>` je namespace objektu (pro náš mód je to `custom-item`)
- `<path>` je název objektu (v našem případě `magic_dust`)

Ve složce `lang` si můžeme vytvořit soubor i pro češtinu `cs_cz.json` a vytvořit si název itemu v češtině.

```
{
  "item.custom-item.magic_dust": "Kouzelný prach"
}
```

Struktura souborů by měla vypadat následovně:



Když nyní hru spustíme, tak bychom měli místo `item.custom-item.magic_dust` vidět název itemu, který jsme si zvolili.

Základní text

Pokud se nenajde překlad názvu, tak se v základu používá hodnota ze souboru `en_us.json`, proto doporučuji **vždy nastavit hodnotu v tomto souboru** a až poté přidávat další jazyk.

Použití předmětu jako palivo

Jakýkoliv předmět, který přidáváme můžeme použít také jako palivo. K tomu se opět používají registry. Pro palivo jsou to konkrétně registry `FuelRegistry`. Do těchto registrů můžeme přidat jakýkoli předmět a ten poté budeme moci použít v peci jako palivo.

Kód opět píšeme do metody `initialize()` ve třídě `ModItems`, protože je to součástí nastavení našeho módu.

Kód na přidání předmětu do registrů paliv bude vypadat následovně:

```
1   FuelRegistry.INSTANCE.add(ModItems.MAGIC_DUST, 5 * 20);
```

Získáme si aktuální seznam předmětů z registru paliva (`INSTANCE`) a poté pomocí metody `add` přidáme vlastní item. Jako argumenty uvedeme item, který chceme přidat a čas, jak dlouho má item hořet.

> **ⓘ Čas v Minecraftu**
>
> Jednotka času v Minecraftu je *tick*. Proto se i veškerý čas v kódu uvádí v této jednotce.
> **1 sekunda = 20 ticků**

Díky tomu, že víme, že 1 sekunda je 20 ticků, tak si můžeme usnadnit výpočet doby, jak dlouho má item hořet a to tak, že vynásobíme počet sekund číslem 20. Například tedy náš item bude hořet 5 sekund (`5 * 20`)

Celá metoda `initialize` by aktuálně měla vypadat následovně:

```
1   public static void initialize() {
2     ItemGroupEvents.modifyEntriesEvent(ItemGroups.INGREDIENTS)
3                            .register((itemGroup) ->
    itemGroup.add(ModItems.MAGIC_DUST));
4
5     FuelRegistry.INSTANCE.add(ModItems.MAGIC_DUST, 5 * 20);
6   }
```