ValkyrieLib 3 Wiki

This is a library of essential code used in many of ValkyrieofNight, and Team Kryptic Links mods. In this wiki you will find generic commonly used ValkyrieLib based Datapack conventions used across all of ValkyrieofNights mods.

Authors: ValkyrieofNight, Apmason13

CurseForge Page

Modrinth Page

- Recipe Item Types
- Recipe Fluid Types
- Overriding Recipes

Recipe Item Types

Item

For regular items it will look something like this:

```
  "raw:item": "minecraft:iron_ingot"
```

Items with NBT

You only need the `"match_nbt": true` if you are using this in an input slot. For output slots, the provided nbt will be added to the item regardless.

```
 "raw:item": "envirocore:fe_input", "nbt": "{tier:2,energy:{capacity:8192}}", "match_nbt": true
```

Item Tags

This would be the same syntax for both input slots and output slots.

```
 "raw:item_tag": "forge:ingots/iron"
```

For tags you can use various selectors to refine which items you want to target. Below is a list of possible selectors and what they do:

- all: All items that are in the tag.
- all_whitelisted: All items from the provided namespaces in the given tag.
- all_with_blacklist: All items from the tag as long as they don't belong to the blacklisted namespaces.
- first: First item value from the tag.
- first_whitelisted: First item from the tag that matches one of the provided namespaces.
- first_preferred: First tries to match an item from the provided namespaces but will take the first item in the tag if none match.
- first_with_blacklist: First item that does not match any of the blacklisted namespaces.

The following is an example of how to use these selectors with and item_tag:

```
{
"raw:item_tag": "minecraft:flowers",
    "selector": {
"type": "all_with_blacklist",
      "values": ["botania"]
}
```

```
}
```

Recipe Fluid Types

Fluid

For regular fluids it will look something like this:

```
  "raw:fluid": "minecraft:water"
```

Fluid Tags

This would be the same syntax for both input and output fluids.

```
 "raw:fluid_tag": "forge:milk"
```

For tags you can use various selectors to refine which fluids you want to target. Below is a list of possible selectors and what they do:

- all: All Fluids that are in the tag.
- all_whitelisted: All Fluids from the provided namespaces in the given tag.
- all_with_blacklist: All Fluids from the tag as long as they don't belong to the blacklisted namespaces.
- first: First Fluid value from the tag.
- first_whitelisted: First Fluid from the tag that matches one of the provided namespaces.
- first_preferred: First tries to match an Fluid from the provided namespaces but will take the first Fluid in the tag if none match.
- first_with_blacklist: First Fluid that does not match any of the blacklisted namespaces.

The following is an example of how to use these selectors with and fluid_tag:

```
{
"raw:fluid_tag": "forge:milk",
    "selector": {
"type": "all_with_blacklist",
      "values": ["simplegens"]
}
}
```

Overriding Recipes

With any recipe that uses ValkyrieLib's recipe system you can override or disable any recipe quite easily. In order to do that you'll need to place these overrides in a custom datapack or use KubeJS or another datapack loading mod.

Pre-Requisites

In order to override or disable a recipe you'll need to first figure out its exact directory and file name. To do that you can open the jar file of the corresponding mod(Once that uses ValkyrieLib as a dependency) as a zip and then navigate to the recipe that you want to override. The path inside that zip file from "data" and onwards is the path you need. For example, if I want to change/disable a recipe inside the `data/envirotech/envirocore/lens_grinder/colored/black.json` I need to create a `black.json` in the `data/envirotech/envirocore/lens_grinder/colored/` directory inside my custom datapack.

Disabling Recipes

Once you have found the directory of the recipe you want to disable is in, and created a json with the same exact name as that recipe you'll now need to put the following json inside of that json file to disable the recipe:

```
{
"override": "disable"
}
```

Replacing Recipes

Once you have found the directory of the recipe you want to replace is in, and created a json with the same name as that recipe you'll now need to put `"override": "replace"` in your new recipe. You could copy the existing recipe in that directory and insert that line inside that recipe and then change the values if you desire.

Example of `data/envirotech/envirocore/lens_grinder/colored/black.json` recipe with the override text inside the recipe:

```
{
  "override": "replace",
  "categories": ["envirotech:colored"],
  "focus": "envirotech:black",
  "r": 0, "g": 0, "b": 0,
  "input": {"raw:itemstack": "envirocore:lens"},
  "output": {"raw:itemstack": "envirocore:lens"},
```

```
  "duration": {"raw:int": 40}
}
```