Reflection / Java access

Very limited reflection is possible, but is not recommended. Use it in cases when KubeJS doesnt support something.

In 1.18.2+ internal Minecraft classes are remapped to MojMaps at runtime, so you don't have to use obfuscated names if accessing internal Minecraft fields and methods.

An example of adding a block with a custom material, built using reflection to get the MaterialJS class, then make a new instance of that with amethyst sounds and material properties from internal Minecraft classes.

```
// Startup script, 1.18.2
const MaterialJS = java("dev.latvian.mods.kubejs.block.MaterialJS")
const Material = java('net.minecraft.world.level.material.Material')
const SoundType = java('net.minecraft.world.level.block.SoundType')

amethystMaterial = new MaterialJS('amethyst', Material.AMETHYST, SoundType.AMETHYST) // f_164531_ and f_154654_ are the respective obfuscated names of these fields, for older versions

//This builder uses 1.18.2 syntax, it will not work in 1.16 or 1.18.1
onEvent('block.registry', event => {
	event.create('amethyst_slab', 'slab')
		.material(amethystMaterial)// Use the new MaterialJS instance we created as the material
		.tagBlock('minecraft:crystal_sound_blocks')
		.tagBlock('minecraft:mineable/pickaxe')
		.requiresTool(true)
		.texture('texture', 'minecraft:block/amethyst_block')
})
```

This does come at a cost, it takes 1-2 seconds to load this script, instead of the normal milliseconds. You should import your classes at the top of the script, instead of in an event, especially if the event gets triggered more than once.

Revision #5
Created 28 June 2020 17:14:31 by Lat
Updated 14 October 2022 14:22:41 by ChiefArug