

RecipeEventJS

Examples

The most basic script to add a single recipe:

```
onEvent('recipes', event => {
  event.shaped('3x minecraft:stone', [
    'SAS',
    'S S',
    'SAS'
  ], {
    S: 'minecraft:sponge',
    A: 'minecraft:apple'
  })
})
```

The most basic script to remove a recipe:

```
onEvent('recipes', event => {
  event.remove({output: 'minecraft:stick'})
})
```

Example recipe script:

```
// kubejs/server_scripts/example.js
// This is just an example script to show off multiple types of recipes and removal methods
// Supports /reload

// Listen to server recipe event
onEvent('recipes', event => {
  // Remove broken recipes from vanilla and other mods
  // This is on by default, so you don't need this line
  //event.removeBrokenRecipes = true

  event.remove({}) // Removes all recipes (nuke option, usually not recommended)
  event.remove({output: 'minecraft:stone_pickaxe'}) // Removes all recipes where output is
```

```

stone pickaxe
    event.remove({output: '#minecraft:wool'}) // Removes all recipes where output is Wool tag
    event.remove({input: '#forge:dusts/redstone'}) // Removes all recipes where input is
Redstone Dust tag
    event.remove({mod: 'quartzchests'}) // Remove all recipes from Quartz Chests mod
    event.remove({type: 'minecraft:campfire_cooking'}) // Remove all campfire cooking recipes
    event.remove({id: 'minecraft:glowstone'}) // Removes recipe by ID. in this case,
data/minecraft/recipes/glowstone.json
    event.remove({output: 'minecraft:cooked_chicken', type: 'minecraft:campfire_cooking'}) //
You can combine filters, to create ANDk logic

// You can use 'mod:id' syntax for 1 sized items. For 2+ you need to use '2x mod:id' or
Item.of('mod:id', count) syntax. If you want NBT or chance, 2nd is required

// Add shaped recipe for 3 Stone from 8 Sponge in chest shape
// (Shortcut for event.recipes.minecraft.crafting_shaped)
// If you want to use Extended Crafting, replace event.shapeless with
event.recipes.extendedcrafting.shapeless_table
event.shaped('3x minecraft:stone', [
    'SAS',
    'S S',
    'SAS'
], {
    S: 'minecraft:sponge',
    A: 'minecraft:apple'
})

// Add shapeless recipe for 4 Cobblestone from 1 Stone and 1 Glowstone
// (Shortcut for event.recipes.minecraft.crafting_shapeless)
// If you want to use Extended Crafting, replace event.shapeless with
event.recipes.extendedcrafting.shaped_table
event.shapeless('4x minecraft:cobblestone', ['minecraft:stone', '#forge:dusts/glowstone'])

// Add Stonecutter recipe for Golden Apple to 4 Apples
event.stonecutting('4x minecraft:apple', 'minecraft:golden_apple')
// Add Stonecutter recipe for Golden Apple to 2 Carrots
event.stonecutting('2x minecraft:carrot', 'minecraft:golden_apple')

// Add Furnace recipe for Golden Apple to 3 Carrots
// (Shortcut for event.recipes.minecraft.smelting)

```

```

event.smelting('2x minecraft:carrot', 'minecraft:golden_apple')
// Similar recipe to above but this time it has a custom, static ID - normally IDs are auto-
generated and will change. Useful for Patchouli
event.smelting('minecraft:golden_apple', 'minecraft:carrot').id('mymodpack:my_recipe_id')

// Add similar recipes for Blast Furnace, Smoker and Campfire
event.blasting('3x minecraft:apple', 'minecraft:golden_apple')
event.smoking('5x minecraft:apple', 'minecraft:golden_apple')
event.campfireCooking('8x minecraft:apple', 'minecraft:golden_apple')
// You can also add .xp(1.0) at end of any smelting recipe to change given XP

// Add a smithing recipe that combines 2 items into one (in this case apple and gold ingot
into golden apple)
event.smithing('minecraft:golden_apple', 'minecraft:apple', 'minecraft:gold_ingot')

// Create a function and use that to make things shorter. You can combine multiple actions
let multiSmelt = (output, input, includeBlasting) => {
  event.smelting(output, input)

  if (includeBlasting) {
    event.blasting(output, input)
  }
}

multiSmelt('minecraft:blue_dye', '#forge:gems/lapis', true)
multiSmelt('minecraft:black_dye', 'minecraft:ink_sac', true)
multiSmelt('minecraft:white_dye', 'minecraft:bone_meal', false)

// If you use custom({json}) it will be using vanilla Json/datapack syntax. Must include
"type": "mod:recipe_id"!
// You can add recipe to any recipe handler that uses vanilla recipe system or isn't
supported by KubeJS
// You can copy-paste the json directly, but you can also make more javascript-y by removing
quotation marks from keys
// You can replace {item: 'x', count: 4} in result fields with Item.of('x',
4).toResultJson()
// You can replace {item: 'x'} / {tag: 'x'} with Ingredient.of('x').toJson() or
Ingredient.of('#x').toJson()
// In this case, add Create's crushing recipe, Oak Sapling to Apple + 50% Carrot

```

```

// Important! Create has integration already, so you don't need to use this. This is just an
example for datapack recipes!
// Note that not all mods format their jsons the same, often the key names ('ingredients',
'results', ect) are different.
// You should check inside the mod jar (mod.jar/data/modid/recipes/) for examples
event.custom({
  type: 'create:crushing',
  ingredients: [
    Ingredient.of('minecraft:oak_sapling').toJson()
  ],
  results: [
    Item.of('minecraft:apple').toResultJson(),
    Item.of('minecraft:carrot').withChance(0.5).toResultJson()
  ],
  processingTime: 100
})

// Example of using items with NBT in a recipe
event.shaped('minecraft:book', [
  'CCC',
  'WGL',
  'CCC'
], {
  C: '#forge:cobblestone',
  // Item.of('id', '{key: value}'), it's recommended to use /kubejs hand
  // If you want to add a count its Item.of('id', count, '{key: value}'). This won't work
here though as crafting table recipes to do accept stacked items
  L: Item.of('minecraft:enchanted_book',
'{StoredEnchantments:[{lvl:1,id:"minecraft:sweeping"}]}'),
  // Same principle, but if its an enchantment, there's a helper method
  W: Item.of('minecraft:enchanted_book').enchant('minecraft:respiration', 2),
  G: '#forge:glass'
})

// In all shapeless crafting recipes, replace any planks with Gold Nugget in input items
event.replaceInput({type: 'minecraft:crafting_shapeless'}, '#minecraft:planks',
'minecraft:gold_nugget')

// In all recipes, replace Stick with Oak Sapling in output items
event.replaceOutput({}, 'minecraft:stick', 'minecraft:oak_sapling')

```

```

// By default KubeJS will mirror and shrink recipes, which makes things like UU-Matter
crafting (from ic2) harder to do as you have less shapes.
// You can use noMirror() and noShrink() to stop this behaviour.
event.shaped('9x minecraft:emerald', [
  ' D ',
  'D  ',
  '  '
], {
  D: 'minecraft:diamond'
}).noMirror().noShrink()
})

```

Possible settings you can change for recipes. It's recommended that you put this in it's own server scripts file, like `settings.js`

```

// priority: 5

// Enable recipe logging, off by default
settings.logAddedRecipes = true
settings.logRemovedRecipes = true
// Enable skipped recipe logging, off by default
settings.logSkippedRecipes = true
// Enable erroring recipe logging, on by default, recommended to be kept to true
settings.logErroringRecipes = false

```

As mentioned before, you can add any recipe from any mod with JSON syntax (see `event.custom({})`) but these mods are supported as addons with special syntax:

- [KubeJS Mekanism](#)
- [KubeJS Immersive Engineering](#)
- [KubeJS Thermal](#)
- [KubeJS Blood Magic](#)
- [KubeJS Create](#)

Ingredient Actions

Poorly documented things below!

You can transform ingredients in shaped and shapeless recipes by adding these functions at end of it:

- `.damageIngredient(IngredientFilter filter, int damage?)` // Will damage item when you craft with it
- `.replaceIngredient(IngredientFilter filter, ItemStackJS item)` // Will replace item with another (like bucket)
- `.keepIngredient(IngredientFilter filter)` // Will keep item without doing anything to it
- `.customIngredientAction(IngredientFilter filter, String customId)` // Custom action that has to be registered in startup script

IngredientFilter can be either

- `ItemStackJS ('minecraft:dirt', Item.of('minecraft:diamond_sword').ignoreNBT(), etc)`
- Integer index of item in crafting table (0, 1, etc)
- Object with item and/or index (`{item: 'something', index: 0}`, etc)

Examples:

```
onEvent('recipes', event => {
  event.shapeless('9x minecraft:melon_slice', [ // Craft 9 watermelon slices
    Item.of('minecraft:diamond_sword').ignoreNBT(), // Diamond sword that ignores damage
    'minecraft:melon' // Watermelon block
  ]).damageIngredient(Item.of('minecraft:diamond_sword').ignoreNBT()) // Damage the sword (also
  has to ignore damage or only 0 damage will work)

  // Craft example block from 2 diamond swords and 2 dirt. After crafting first diamond
  sword is damaged (index 0) and 2nd sword is kept without changes.
  event.shaped('kubejs:example_block', [
    'SD ',
    'D S'
  ], {
    S: Item.of('minecraft:diamond_sword').ignoreNBT(),
    D: 'minecraft:dirt'
  }).damageIngredient(0).keepIngredient('minecraft:diamond_sword')

  // Craft example block from 2 diamond swords and 2 stones. After crafting, diamond sword
  is replaced with stone sword
  event.shapeless('kubejs:example_block', [
    Item.of('minecraft:diamond_sword').ignoreNBT(),
    'minecraft:stone',
```

```

    []Item.of('minecraft:diamond_sword').ignoreNBT(),
    []'minecraft:stone'
[]).replaceIngredient('minecraft:diamond_sword', 'minecraft:stone_sword')

    // Craft clay from sand, bone meal, dirt and water bottle. After crafting, glass bottle is
left in place of water bottle
[]event.shapeless('minecraft:clay', [
    []'minecraft:sand',
    []'minecraft:bone_meal',
    []'minecraft:dirt',
    []Item.of('minecraft:potion', {Potion: "minecraft:water"})
[]).replaceIngredient({item: Item.of('minecraft:potion', {Potion: "minecraft:water"})},
'minecraft:glass_bottle')

    []// Register a customIngredientAction, and recipe that uses it
    []// This one takes the nbt from an enchanted book and applies it to a tool in the crafting
table, for no cost.
    []// Thanks to Prunoideae for providing it!
    []Ingredient.registerCustomIngredientAction("apply_enchantment", (itemstack, index, inventory)
=> {
        let enchantment =
inventory.get(inventory.find(Item.of("minecraft:enchanted_book").ignoreNBT())).nbt;
        if (itemstack.nbt == null)
            itemstack.nbt = {}
        itemstack.nbt = itemstack.nbt.merge({ Enchantments:
enchantment.get("StoredEnchantments") })
        return itemstack;
    })

    []event.shapeless("minecraft:book", ["#forge:tools",
Item.of("minecraft:enchanted_book").ignoreNBT()])
        .customIngredientAction("#forge:tools", "apply_enchantment")
})

```

Revision #31

Created 2020-07-20 11:21:34 UTC by Lat

Updated 2022-12-03 06:25:45 UTC by ChiefArug