Migrating to KubeJS 6

> This page is still being worked on, so if some info is missing, please check back later!

What's changed in the new KubeJS 6 (1.19.2+)?

onEvent()

`onEvent('event', e => {})` syntax was replaced by `SomeEventGroup.someEventName(e => {})`

```
// Before
onEvent('block.right_click', e => {
  if(e.block.id === 'minecraft:dirt') console.info('Hi!')
})

// After
BlockEvents.rightClicked(e => {
  if(e.block.id === 'minecraft:dirt') console.info('Hi!')
})
```

Not only that, but new events also support extra parameters for IDs and other things! You can now choose to make each id have its own event handler:

```
// Before
onEvent('block.right_click', e => {
  if(e.block.id === 'minecraft:dirt') console.info('Hi!')
  else if(e.block.id === 'minecraft:stone') console.info('Bye!')
})

// After
BlockEvents.rightClicked('minecraft:dirt', e => {
  console.info('Hi!')
})
```

```
BlockEvents.rightClicked('minecraft:stone', e => {
  console.info('Bye!')
})
```

Some events *require* ID, such as registry and tag events:

```
// Before
onEvent('item.registry', e => {})

// After
StartupEvents.registry('item', e => {})
```

```
// Before
onEvent('tags.items', e => {})

// After
ServerEvents.tags('item', e => {})
```

Using parameters is actually faster on the CPU than checking some `event.id == 'id'`

> You can find the full list of new events here.

onForgeEvent()

`onForgeEvent('package.ClassName', e => {})` has been replaced by `ForgeEvents.onEvent('package.ClassName', e => {})`

```
// Before
onForgeEvent('net.minecraftforge.event.level.BlockEvent$PortalSpawnEvent', e => {})

// After
ForgeEvents.onEvent('net.minecraftforge.event.level.BlockEvent$PortalSpawnEvent', e => {})
```

New! It now supports generic events:

```
ForgeEvents.onGenericEvent('net.minecraftforge.event.AttachCapabilitiesEvent',
'net.minecraft.world.entity.Entity', e => {})
```

Server settings

`settings.log...` properties have been removed from server scripts, and instead, moved to `local/kubejsdev.properties` file. By default, it won't be shipped with the pack, but you can change `saveDevPropertiesInConfig` to true to instead save the file in `kubejs/config/dev.properties`.

java()

`java('package.ClassName')` has been replaced by `Java.loadClass('package.ClassName')`

```
// Before
const CactusBlock = java('net.minecraft.world.level.block.CactusBlock')

// After
const CactusBlock = Java.loadClass('net.minecraft.world.level.block.CactusBlock')
```

There might be some more reflective helper methods later in the Java util class, such as listing all fields and methods in a class, etc.

Bye Bye Wrapper classes

None of the vanilla classes are wrappers anymore - `EntityJS`, `LevelJS`, `ItemStackJS`, `IngredientJS`, and others are gone. This may introduce some bugs, but in general, should make it significantly easier to interact with Minecraft and other mods. Almost all old methods are still supported by core-modding vanilla. This should also significantly boost performance, as it doesn't need to constantly wrap and unwrap classes.

KubeJS 6.1 Update

Other questions

If you have any other questions, feel free to ask them on my Discord Server.

Revision #26
Created 2022-10-14 14:21:44 UTC by Lat
Updated 2023-06-26 10:29:02 UTC by Lat