

# TagEventJS

This event is fired when a tag collection is loaded, to modify it with script. You can add and remove tags for items, blocks, fluids and entity types.

This goes into server scripts.

Tags are per item/block/fluid/entity type and as such cannot be added based on things like NBT data!

## Parent class

[EventJS](#)

## Can be cancelled

No

## Variables and Functions

Name	Type	Info
<code>type</code>	<a href="#">String</a>	Tag collection type.
<code>get(<a href="#">String</a> tag)</code>	TagWrapper	Returns specific tag container which you can use to add or remove objects to. tag parameter can be something like 'forge:ingots/copper'. If tag doesn't exist, it will create a new one.
<code>add(<a href="#">String</a> tag, <a href="#">String</a>[/Regex ids])</code>	TagWrapper	Shortcut method for <code>event.get(tag).add(ids)</code> .
<code>remove(<a href="#">String</a> tag, <a href="#">String</a>[/Regex ids])</code>	TagWrapper	Shortcut method for <code>event.get(tag).remove(ids)</code> .
<code>removeAll(<a href="#">String</a> tag)</code>	TagWrapper	Shortcut method for <code>event.get(tag).removeAll()</code> .
<code>removeAllTagsFrom(String[] ids)</code>	void	Removes all tags from object

## TagWrapper class

# Variables and Functions

Name	Type	Info
add( <a href="#">String</a> [/Regex ids])	TagWrapper (itself)	Adds an object to this tag. If string starts with # then it will add all objects from the second tag. It can be either single string, regex (/regex/flags) or array of either.
remove( <a href="#">String</a> [/Regex ids])	TagWrapper (itself)	Removes an object from tag, works the same as add().
removeAll()	TagWrapper (itself)	Removes all entries from tag.
getObjectIds()	Collection<ResourceLocation>	Returns a list of all entries in a tag. Will resolve any sub-tags.

## Examples

```
// Listen to item tag event
onEvent('item.tags', event => {
  // Get the #forge:cobblestone tag collection and add Diamond Ore to it
  event.add('forge:cobblestone', 'minecraft:diamond_ore')

  // Get the #forge:cobblestone tag collection and remove Mossy Cobblestone from it
  event.remove('forge:cobblestone', 'minecraft:mossy_cobblestone')

  // Get #forge:ingots/copper tag and remove all entries from it
  event.removeAll('forge:ingots/copper')

  // Required for FTB Quests to check item NBT
  event.add('itemfilters:check_nbt', 'some_item:that_has_nbt_types')

  // You can create new tags the same way you add to existing, just give it a name
  event.add('forge:completely_new_tag', 'minecraft:clay_ball')

  // Removes all tags from this entry
  event.removeAllTagsFrom('minecraft:stick')

  // Add all items from the forge:stone tag to the c:stone tag, unless the id contains diorite
  const stones = event.get('forge:stone').getObjectIds()
  const blacklist = Ingredient.of(/.*diorite.*/)
  stones.forEach(stone => {
```

```
if (!blacklist.test(stone)) {  
    event.add('c:stone', stone)  
}  
})  
})
```

Recipes use item tags, not block or fluid tags, even if items representing those are blocks. Like `minecraft:cobblestone` even if it's a block, it will still be an item tag for recipes.

`tags.blocks` and `tags.fluids` are for adding tags to block and fluid types, they work the same way. You can find existing block and fluid tags if you look at a block with F3 mode enabled, on side. These are mostly only used for technical reasons, and like mentioned above, if its for recipes/inventory, you will want to use `tags.items` even for blocks.

---

Revision #30

Created 20 July 2020 10:46:15 by Lat

Updated 27 January 2023 21:18:26 by Lexxie