Action API

The Action API is not a plugin and does nothing by itself. It needs other plugins that use it and only provides classes and interfaces as well as a config loader for YAML Configuration files.

Table Of Contents

Installation

Simply install the Raid-Craft API and you are ready to go.

Configuration

The configuration takes place in the implementing plugins that use requirements, actions and trigger. Please take a look at the plugin you are using to know where you need to place the Action API configuration sections. The config that is described here is always inside a file from the implementing plugin, e.g. a quest or an achievement config file. The article of the respective plugin will tell you where you need to configure the requirements, trigger or actions.

You can get a list of all available actions, trigger and requirements with `/actionapi`. If they provide a ConfigGenerator they contain a description what they do. You can view the detailed description and information with `/actionapi help name`

All `Actions`, `Trigger` and `Requirements` are configured the same way. They are simply placed in different positions of the config file dependent on the plugin that is using the Action API.

The config section always looks like this:

Trigger

Without a trigger requirements will never be checked and actions will never be executed so a trigger is a must have in any situation. Trigger can be "requirements" by themselves if the config is build accordingly. A Trigger can also have its own requirements that will prevent the execution of actions until the requirements are met.

For example you could have a Trigger that executes everytime you press a button. You could then have a dummy requirement under the trigger that counts until 3. Add an action to the trigger that kills the player. This action will now be execute once the player clicked the button 3 times.

You can also set a trigger as `execute-once: true/(default: false)` and it will once execute once. This means it will not execute it actions and will not call processTrigger on implementing listeners, e.g. Quests or Achievements. In our example you could prevent the player from being killed more than one time.
Internally the `execute-once` trigger uses a persistant requirement that is set to false and persistant after it was checked once.

Overview