

# Aktoren (Actors) und Initiatoren (Initiators)

Um FabAccess erweiterbar zu halten, basiert die Steuerung externer Geräte wie Wifi-Schaltsteckdosen oder Türschlössern auf einem Aktoren- und Initiatorenkonzept.

## Aktoren (Actors)

Aktoren in FabAccess haben die Aufgabe, die digitalen Zustände von Ressourcen in reale Zustände umzusetzen. Vom Server aus werden die Übergänge der [Traits](#) (Zustände) an kleine Skripte oder Prozesse weitergegeben, die entsprechend darauf reagieren.

Aktoren erhalten vom Server Mitteilungen über Änderungen an Ressourcen, beispielsweise wenn eine Ressource ausgeliehen wird, und passen dann den realen Zustand der Ressource an. Dadurch wird die Ressource für Nutzer freigeschaltet / eingeschaltet. Darüber hinaus ermöglichen Aktoren die Steuerung zusätzlicher Anzeigen wie Maschinenampeln oder Displays.

Aktoren werden in der [Hauptkonfiguration](#) `bffh.dhall` definiert und an Ressourcen (Maschinen) gebunden (Mapping [actor\\_connections](#)).

## Initiatoren (Initiators)

Initiatoren sind STDOUT/Prozess-basierte Plugins. Mit Initiatoren werden reale Zustände auf die digitalen Zustände in FabAccess abgebildet. Initiatoren sind Plugins, die den Status einer Ressource aktiv verändern können, also zum Beispiel von `Free` auf `InUse` setzen. Sie fungieren sozusagen als Callbacks. Die Auslösevents (Trigger) sind in der Regel asynchron, von der App entkoppelt und lösen automatisiert aus.

Initiatoren werden in der [Hauptkonfiguration](#) `bffh.dhall` definiert und an Ressourcen (Maschinen) gebunden (Mapping [init\\_connections](#)).

## Beispiele

Mit Initiatoren können im einfachsten Fall fest definierte Initialzustände an Ressourcen übersendet werden, nachdem BFFH neugestartet wurde - zum Beispiel für einen täglichen "Werkstattreset". So können u.a. bestimmte Dinge automatisch ein-/ausgeschaltet bzw. in einen klar definierten Ausgangszustand versetzt werden. Ein Beispiel dafür findet sich in [Initiator: Generisches Python-Template für "Process"](#).

Initiatoren ermöglichen auch Zeitschaltungen für die automatische Rückgabe von Ressourcen, wenn sie nicht mehr verwendet werden (z.B. wenn Stromverbrauch für mehrere Minuten unterhalb eines Verbrauchsschwellwerts, dann MQTT-Nachricht versenden und Ressource freigeben). Eine Beispielimplementierung findet sich in [Initiator: Shelly Timeout](#).

Türkontakte, die den aktuellen Zustand der Tür übertragen können, können ebenso als Initiator geschrieben werden und BFFH kann dann entsprechend darauf reagieren.

## Sammlung von Aktoren und Initiatoren

Eine Sammlung von Initiatoren und Aktoren ist [hier](#) zu finden.

---

Version #17

Erstellt: 16 Oktober 2024 13:00:12 von Mario Voigt (Stadtfabrikanten e.V.)

Zuletzt aktualisiert: 8 März 2025 12:21:44 von Mario Voigt (Stadtfabrikanten e.V.)