

# Fluxmite Rules Guide

---

## Create Local

**Event:** Something happened ....

**Condition:** Data comparison: change-type = create

**Action:**

1. Create new local
2. Set data value
3. Set data value: mite-id
4. Save local
5. Mite: create local entity

## Delete Local

**Event:** Something happened ....

**Condition:** Data comparison: change-type = delete

**Action:**

1. Fetch local by id
2. Delete entity

## Update Remote

**Event:** After updating local

**Condition:** Entity is of bundle

**Action:**

1. Mite: fetch remote entity by local entity
2. Conditional if: is of type
3. Set data value
4. Mite: update remote entity (returns the complete Mite entity)
5. Conditional else: Mite: enqueue remote action

**Dependencies:** Rule Update Local will be triggered

## Update Local

**Event:** Something happened ....

**Condition:** Data comparison: change-type = update

**Action:**

1. Fetch local by id
2. Conditional: is of bundle
3. Set data value
4. Set data value: mite-id (used by create and update remote)
5. Mite: update local

## Create Remote

**Event:** After saving local

**Condition:**

1. Is of bundle
2. Data value is empty: mite-id (to avoid duplicate entries, because create local triggers this rules too)

**Action:**

1. Create new remote
2. Set data value
3. Mite: create remote entity (returns the complete Mite entity)

**Dependencies:** Rule Update Local will be triggered

## Delete Remote

**Event:** After deleting local

**Condition:** -

**Action:** Mite: delete remote entity

## Entity Dependencies:

**Mite Project -> Local Project:** Customers are needed and must be handled before projects

**Mite Time entry -> Local Time entry:** Customer, Projects, Services, Users are needed and must be handled before time entries

If the needed entities are not handled the handler (project, time entry) won't run until.

If the order isn't correct the handler will correct it and run next time.