

# 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. Optional:
  1. Mite: fetch referenz by mite id
  2. Conditional: is of type (fetched)
  3. Set data value
5. Optional:
  1. Mite: set local hourly rates per service
6. Save local
7. 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 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. Optional:
  1. Mite: fetch referenz by mite id
  2. Conditional: is of type (local)
  3. Set data value
5. Optional:
  1. Mite: update local hourly rates per service
6. Mite: update local

## Create Remote

**Event:** After saving local

**Condition:**

1. Is of bundle
2. Optional: Referenz is of bundle
3. 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. Optional:
  1. Mite: set remote hourly rates per service
4. Mite: create remote entity (returns the complete Mite entity)
5. Conditional if: is of type (entity created)
6. Set data value: mite-id (local)

## Delete Remote

**Event:** After deleting local

**Condition:** -

**Action:** Mite: delete remote entity

## Update Remote

**Event:** After updating local

**Condition:**

1. Entity is of bundle
2. Optional: Referenz is of bundle

**Action:**

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

All local enties must have a field called **field\_mite\_id** which contains the mite id of the connected remote entity.

# Fluxmite Rules Guide

---

## **Entity Dependencies:**

**Mite Customer:** Services are needed and must be handled before customers.

**Mite Project:** Customers and Services are needed and must be handled before projects.

**Mite Time entry:** Customers, Projects, Services, Users are needed and must be handled before time entries.

If the needed entities are not handled or have no database entries the handler (customer, project, time entry) won't run until (error log via watch\_dog).

If the order isn't correct the handler will correct it and run next time (error log via watch\_dog).