Fluxmite Rules Guide

Create Local

Event: Something happened ….
Condition: Data comparsion: change-type = create
Action: *¹

1. **Entities**: Create new local
2. **Data**: Set data value
3. **Data**: Set data value: mite-id
4. Optional:
   1. **FE***²: fetch reference by remote id
   2. **Conditional**: is of type (fetched)
   3. **Data**: Set data value
5. Optional:
   1. **Mite**: set local hourly rates per service
6. **Entities**: Save local
7. **FE**: create local entity

Delete Local

Event: Something happened ….
Condition: Data comparsion: change-type = delete
Action:

1. **Entities**: Fetch local by id
2. **Entities**: Delete entity (fetched)

Update Local

Event: Something happened ….
Condition: Data comparsion: change-type = update
Action:

1. **Entities**: Fetch local by id
2. **Conditional**: is of bundle
3. **Data**: Set data value
4. Optional:
   1. **FE**: fetch reference by remote id
   2. **Conditional**: is of type (local)
   3. **Data**: Set data value
5. Optional:
   1. **Mite**: update local hourly rates per service
6. **FE**: update local

Create Remote

Event: After saving local
Condition:

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

Action:

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

Delete Remote

Event: After deleting local
Condition: -
   Action: FE: delete remote entity

Update Remote

Event: After updating local
Condition:

1. Entity is of bundle
2. Optional: is of bundle (reference)

Action:

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

All local enties <u>must have</u> a field called **field_mite_id** which contains the mite id of the connected remote entity.

*¹: action-category: action-name   *²: FE = Fluxservice Extension

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).

The dependency check can be disabled at the webservice endpoint configuration.

$*^1$: action-category: action-name $*^2$: FE = Fluxservice Extension