

Manufacturing

This guide explains the concepts behind the manufacturing module and the relevant API calls to build apps around the manufacturing functionality.

In this guide

Resources

Before you create manufacturing apps, it's helpful to understand some of the different manufacturing-related resources:

BOM - Bill of materials

A bill of materials is a list of the raw materials and the quantities of each needed to manufacture an end product. The BOM is also known as the formula, recipe, or ingredients list.

When a manufacturing order is created, the BOM input quantities are multiplied by the order quantity to calculate total material requirements.

Routing

A routing lists all the production operations or steps to be performed to manufacture an end product. Each line of the routing represents one manufacturing stage.

It is through routing that labor and overhead for particular Work Orders are tracked and recorded. The default routing to be used when a BOM is used can be set on the BOM.

Manufacturing/Production Order

A manufacturing order tracks the manufacturing of a certain quantity of a specific product. The manufacturing order could be a multi-quantity order for stock or created automatically from a sales order line for *made on order* items.

Work Order

A work order is a sub document of a production order for a specific step in manufacturing. Work orders are optional and are created automatically when a production order starts based on the routing associated with the production order.

Architecture



States of a production order



Draft

A manufacturing order that is currently being created/edited by a user.

Request

A manufacturing order that was created automatically by Fulfil as a request based on current demand/rules.

Waiting

When the order is waiting for inventory allocation.

Assigned

When the order is assigned inventory. At this point the production order is ready to start. Remember that all assigned order may not be ready to start as they could be future orders (see ``planned_date``).

Running

The production order is currently running. At this point all of the raw material inventory used as inputs are consumed.

Though it is possible to move back to a previous status, this is considered a point of no-return. If the production order is moved back to any previous statuses, the inventory remains consumed.

Done

The production order (and all work orders) are completed. The inventory of the finished product is now available in the destination warehouse.

Using the API to build manufacturing apps

The following API calls should give you a picture on how to build your own manufacturing interfaces on top of Fulfil. These are the same API endpoints and methods used by the primary fulfil app, but building custom apps gives you the ability to reuse the building blocks provided by Fulfil, but build your own friendly UI and terminology within your company.

Based on the scope of your app, you can select the API calls you will need and it is unlikely you will need all of these calls. Build as little as you need to make your custom workflow happen.

Production Orders

Get a list of production orders

The most common need is to fetch a list of production orders in a certain state.

```
GET /model/production?state=running
```

On the other hand, if you already know the ids of the production order, then you can fetch production orders by passing a list of ids separated by ```,``.

```
GET /model/production?ids=12,13,15
```

Moving orders through states

Another important task is to move production orders through different states. This requires sending a PUT request to the production order(s).

Move to waiting

```
PUT /model/production/{id}/wait
```

Move to draft

```
PUT /model/production/{id}/draft
```

Assign Inventory

You can either try to assign inventory or force assign inventory. To make the best effort to assign inventory, you can send a PUT request.

```
PUT /model/production/{id}/assign_try
```

This returns true or false depending on whether the assignment was successful or not. If items are partially available, Fulfil allocates whatever is available.

You can also force assign inventory, but this is not recommended.

Start Running

When the production order is ready to start running, you can trigger running by sending a PUT request.

```
PUT /model/production/{id}/run
```

Mark as done

When the production order is completely done, you can mark it as done by sending a PUT request.

```
PUT /model/production/{id}/done
```

Work Orders

A work order is a sub document of a production order for a specific step in manufacturing. Work orders are optional and are created automatically when a production order starts based on the routing associated with the production order.

States of a work order**Draft**

A work order that is currently being created/edited by a user.

Request

A work order that was created automatically by Fulfil as a request based on current demand/rules.

Waiting

When the work order is waiting to be started.

Running

The work order is now running. This usually makes the cycle clock start counting to track time.

Finished

All cycles associated with the work is completed. The job itself may not be completed as this may have been a partial stop of the cycles.

Done

This work is completed and no more work can be done on the work order.

Get a list of work orders

The most common need is to fetch a list of work orders in a certain state.

```
GET /model/production.work?state=waiting
```

Common search filters

Field	
ids	Retrieve certain orders, specified by a comma-separated list of order IDs.
limit	<i>"The maximum number of results to show on a page. (max: 500)"</i>
since_id	<i>"Show orders after the specified ID."</i>
created_at_min	<i>"Show orders created at or after date. (format: 2019-04-25T10:15:47)"</i>
created_at_max	<i>"Show orders created at or before date. (format: 2019-04-25T10:15:47)"</i>
updated_at_min	<i>"Show orders last updated at or after date. (format: 2019-04-25T10:15:47)"</i>
updated_at_max	<i>"Show orders last updated at or before date. (format: 2019-04-25T10:15:47)."</i>
state	<div><i>"State of the work order"</i><ul style="list-style-type: none"><i>request</i><i>draft</i><i>waiting</i><i>running</i><i>done</i></div>
production.id	<div>Filter by the id of a production number. You can also search by other attributes of production<ul style="list-style-type: none">production.numberproduction.planned_dateproduction.effective_date</div>
operation.id	<div>Filter by the id of an operation. You can also search by other attributes of operation<ul style="list-style-type: none">operation.name</div>

Moving work orders through states

Starting a work order

To start an existing work_order, make a PUT call to the start_new_cycle endpoint.

```
PUT /model/production.work/{id}/start_new_cycle
```

For this operation to be successful, you need an operator assigned. If the user making this API call (or in case of apps on behalf of a user) is not an operator, then an operator needs to be set manually.

Operators are "employees" in Fulfil. If the operator is not assigned prior to this starting, the best approach is to set the operator.

```
PUT /model/production.work/{id}
```

```
{"operator": 123}
```

Finish a work order

To finish a work order, you can call the `finish_all_cycles` endpoint.

```
PUT /model/production.work/{id}/finish_all_cycles
```
