Overall explanation of the State Manager

What is

This state manager machine is a fully automated way to create, process, or other process name you would like to create, all this following pre created modifiable workflow blueprints like a recipe

Ecommerce application:

An e-commerce can use this system to create, validate, and finish Orders, letting the e-commerce to create micros services to expose apis for various effects.
With this system you can use too micro services from third parties.

Why

The problematic this system comes to solve is to handle all the heavyweight processes that are needed, saying on a blueprint you can define what Api to use with which data and what data of the response to save to use later on the next Api or the final Api.

Even creating custom functions in code to handle data as required, for example, extract the first element of an array.

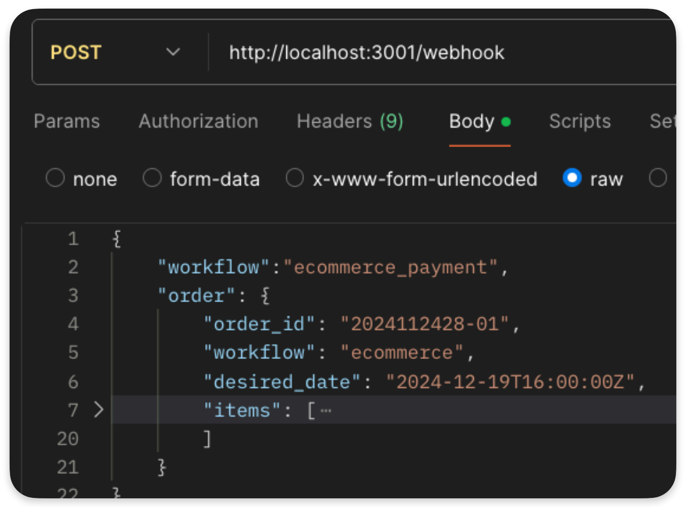
the state manager is like building a power ranger robot from Apis from x number of sources, you decide on the blueprint the parts and how to assemble them, and finally where to send him to fight

Flow chart

Runs recursively, every 60 seconds after finishing processing a batch

State Order Creation

Key points:
Order object
order_id - has to be unique
Workflow :previously declared on workflowBlueprint schema



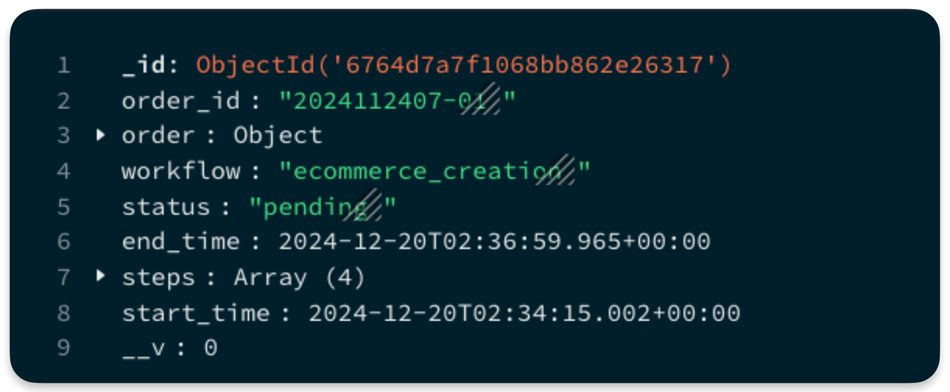
http://localhost:3001/webhook

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○

```
{
    "workflow":"ecommerce_payment",
    "order": {
        "order_id": "2024112428-01",
        "workflow": "ecommerce",
        "desired_date": "2024-12-19T16:00:00Z",
        "items": [⋯
        ]
    }
}
```

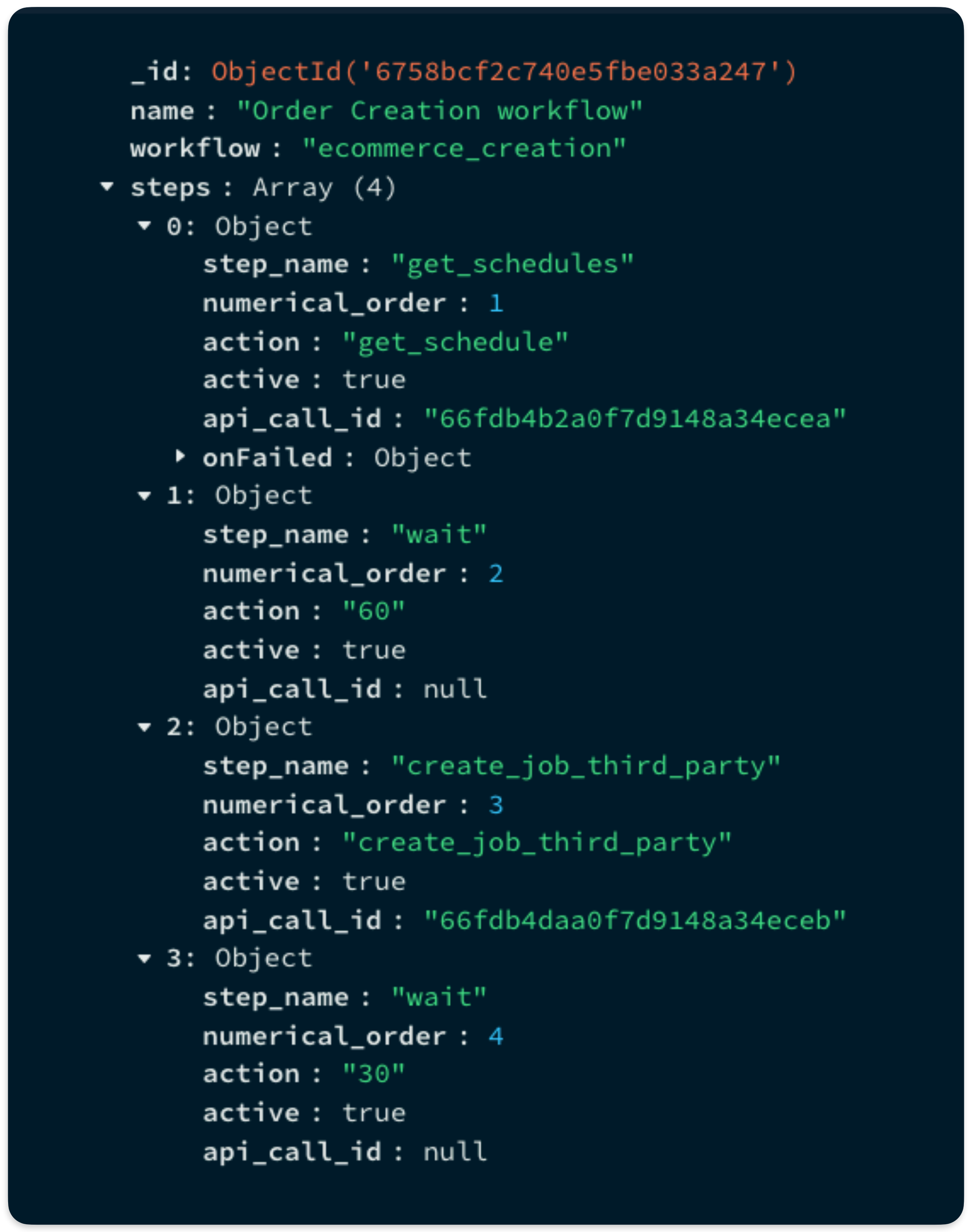
API POST Webhook in

Creates an State Order with pending state, to be processed on batch later



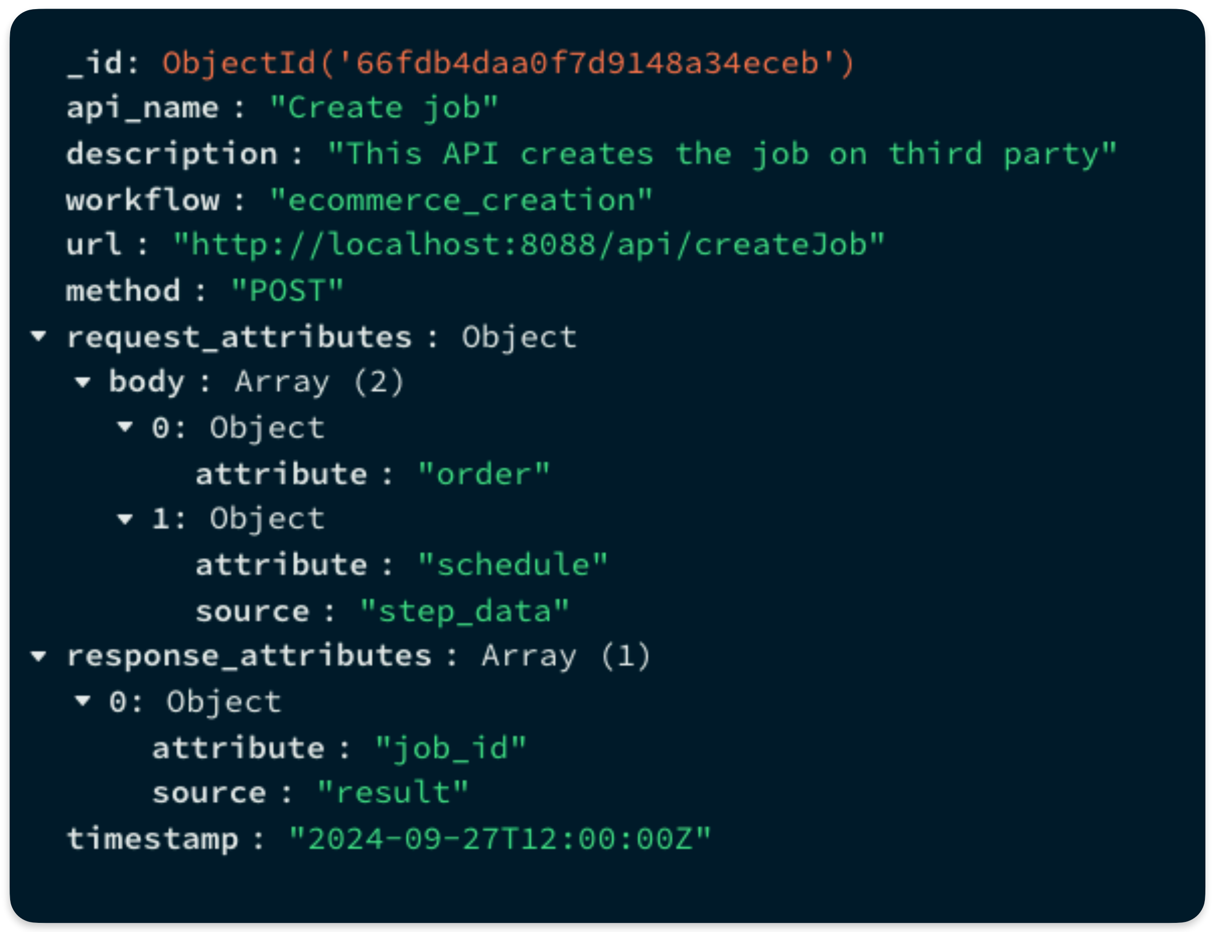
```
  _id: ObjectId('6764d7a7f1068bb862e26317')
  order_id : "2024112407-01"
▸ order : Object
  workflow : "ecommerce_creation"
  status : "pending"
  end_time : 2024-12-20T02:36:59.965+00:00
▸ steps : Array (4)
  start_time : 2024-12-20T02:34:15.002+00:00
  __v : 0
```

Workflow Blueprint



```
    _id: ObjectId('6758bcf2c740e5fbe033a247')
    name : "Order Creation workflow"
    workflow : "ecommerce_creation"
  ▾ steps : Array (4)
    ▾ 0: Object
        step_name : "get_schedules"
        numerical_order : 1
        action : "get_schedule"
        active : true
        api_call_id : "66fdb4b2a0f7d9148a34ecea"
      ▸ onFailed : Object
    ▾ 1: Object
        step_name : "wait"
        numerical_order : 2
        action : "60"
        active : true
        api_call_id : null
    ▾ 2: Object
        step_name : "create_job_third_party"
        numerical_order : 3
        action : "create_job_third_party"
        active : true
        api_call_id : "66fdb4daa0f7d9148a34eceb"
    ▾ 3: Object
        step_name : "wait"
        numerical_order : 4
        action : "30"
        active : true
        api_call_id : null
```

Apicall Schema



```
  _id: ObjectId('66fdb4daa0f7d9148a34eceb')
  api_name : "Create job"
  description : "This API creates the job on third party"
  workflow : "ecommerce_creation"
  url : "http://localhost:8088/api/createJob"
  method : "POST"
▾ request_attributes : Object
  ▾ body : Array (2)
    ▾ 0: Object
        attribute : "order"
    ▾ 1: Object
        attribute : "schedule"
        source : "step_data"
▾ response_attributes : Array (1)
  ▾ 0: Object
      attribute : "job_id"
      source : "result"
  timestamp : "2024-09-27T12:00:00Z"
```

State Order Processing

It runs every x time and process x state orders at the same time, but waits till all orders are processed before check pending orders(this to avoid memory overflow)

Function finds and start the process on the orders

The system will follow the "workflow blueprint" designated by the workflow at the moment of creation, following each step with api_call_id previously established on the api_call_id schema

Extras:
-onFailed is used as an alternate way on case of failed
-you can reprocess all or some orders