OPTIORYX
35 Lammerstraat
Ghent, BE 9000
hello@optioryx.com

ALLOTCODA DOCUMENTATION

Updated: 25th October 2023

DIMENSIONER (IOS APP)

How to set up

In order to set up the iOS app you need to:

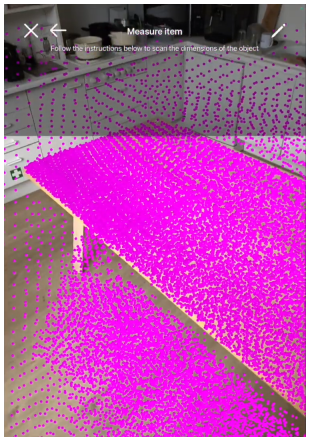
1. Navigate into its settings
2. Select the server configuration option
3. Enter the IP address you'll be hosting your Docker environment on

How to use

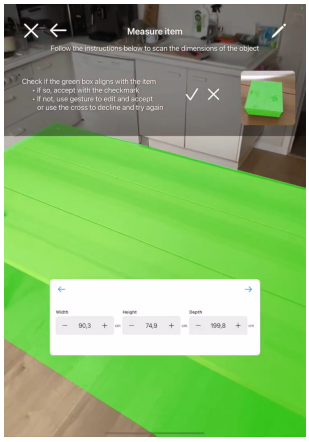
1. Select the "Scan" option to start collecting master data.

2. Line up your barcode to read it.

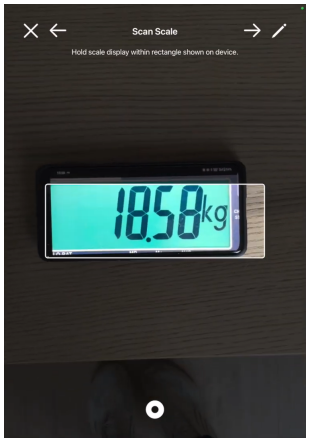
3. Tap the object you wish to measure to start dimensioning it. Tap again to stop.

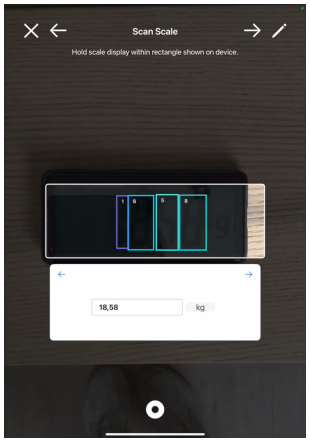


4. Verify that the green bounding box covers the object correctly, and adjust if necessary. You can always modify the dimensions manually.

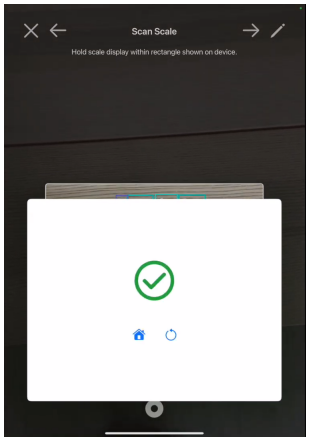


5. Record the weight of an object by lining up with the scale's seven-segment display.





6. Finally, you can choose to go back to the home screen or start a new scan.


Scan Scale
Hold scale display within rectangle shown on device.

MASTER DATA STORE (API)

The Master Data Store is an API written with Python FastAPI that uses MongoDB for storing most of its data and an S3 bucket for storing images and other binaries.

How to set up

Prerequisites

- mongo:4.0
- fiware/orion-ld:latest
- minio/minio:RELEASE.2020-08-18T19-41-00Z

Configure the service

```
master-data-store:
    build:
      context: ./master_data_store
    environment:
      - ORION_URL=http://orion:1026
      - MONGODB_URL=mongodb://mongo:27017
      - MINIO_URL=http://minio1:9000
      - MINIO_ACCESS_KEY=admin
      - MINIO_SECRET_KEY=password
      - API_LOCATION=http://master-data-store
    ports:
      - "80:80"
    depends_on:
      - orion
      - mongo
      - minio1
      - minio2
      - minio3
      - minio4
    restart: on-failure
```

Environment variables

- ORION_URL*: the URL to your Orion service

- MONGODB_URL*: the connection string to your MongoDB
- MINIO_URL*: the URL to your Minio service
- MINIO_ACCESS_KEY*: your Minio service access key
- MINIO_SECRET_KEY*: your Minio service secret key
- API_LOCATION*: the external URL on which this service is exposed

Configure the Orion integration

```yaml
orion-mds-bridge:
    build:
      context: ./orion_mds_bridge
    environment:
      - ORION_URL=http://orion:1026
      - MDS_URL=http://master-data-store
    ports:
      - "1029:1028"
    depends_on:
      - orion
      - master-data-store
    restart: on-failure
```

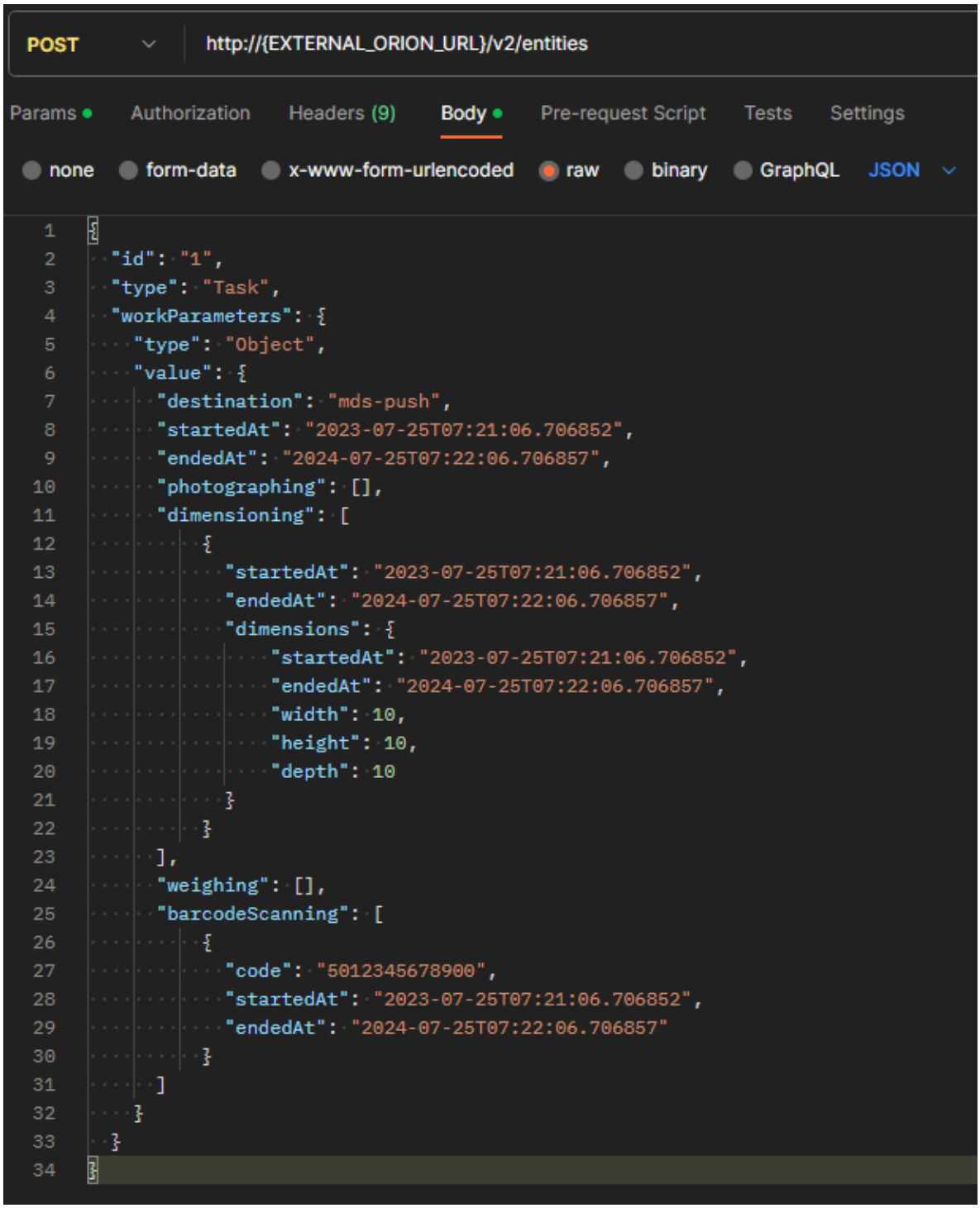
Environment variables

- ORION_URL*: the URL to your Orion service
- MDS_URL*: the URL to your Master Data Store service

How to use

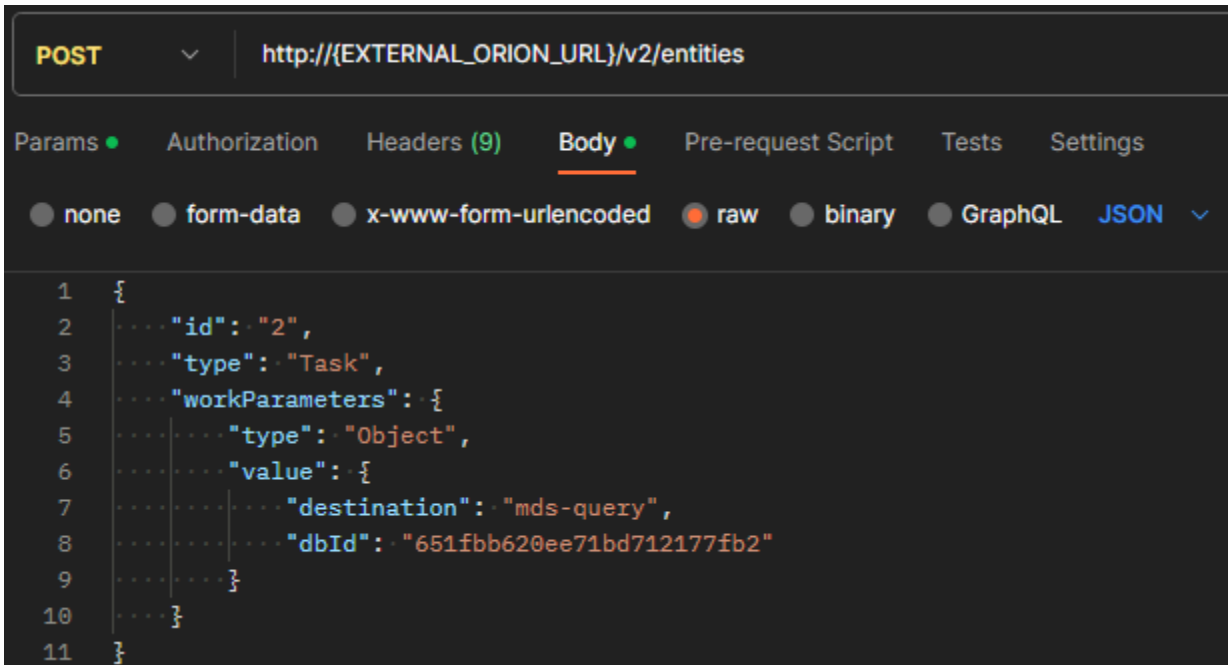
Adding data

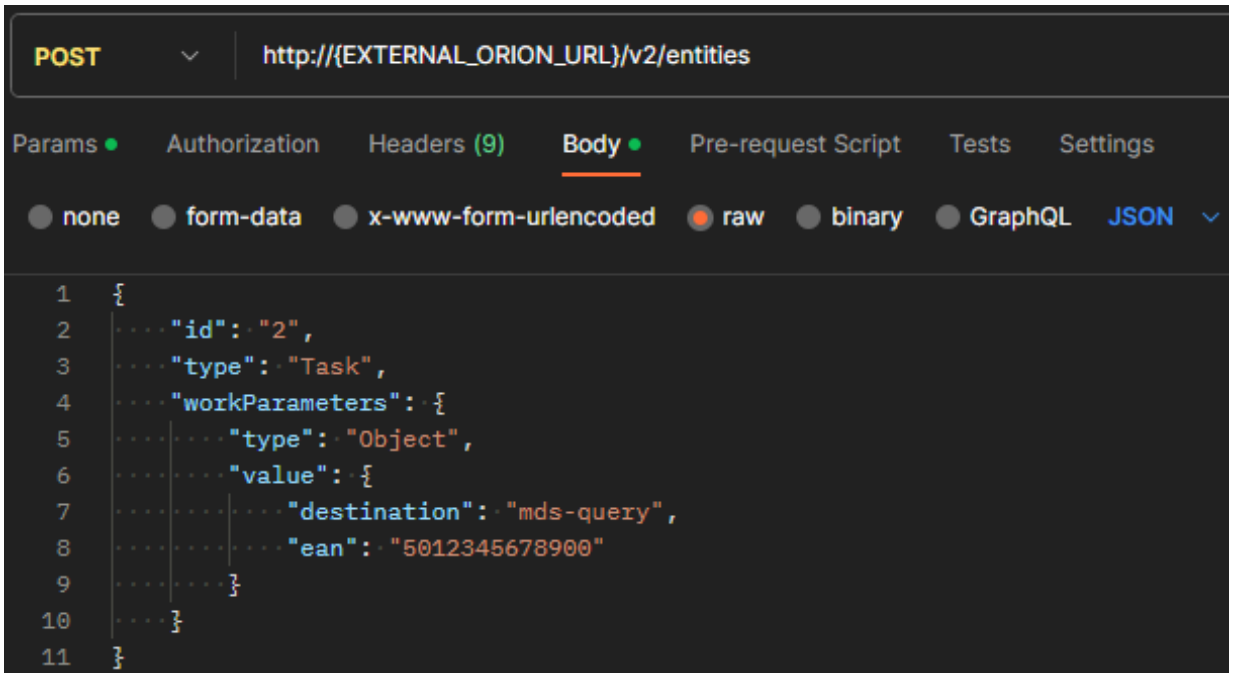
Use the “mds-push” destination, providing a workParameter with a value that fits the Item schema (see {API_LOCATION}/docs, /redoc, or /openapi.json for more information).



Querying data

Use the “mds-query” destination, providing a workParameter with a value that contains either a “dbId” field (an ObjectId string) or a “ean”[1] field (the value provided with the “barcodeScanning.code”).





[1] ean: European Article Number

MASTER DATA STORE (WEB)

The web module of the Master Data Store is built in React and offers a complete overview of the collected data. It retrieves data from the Master Data Store.

How to set up

Prerequisites

- master-data-store

Configure the service

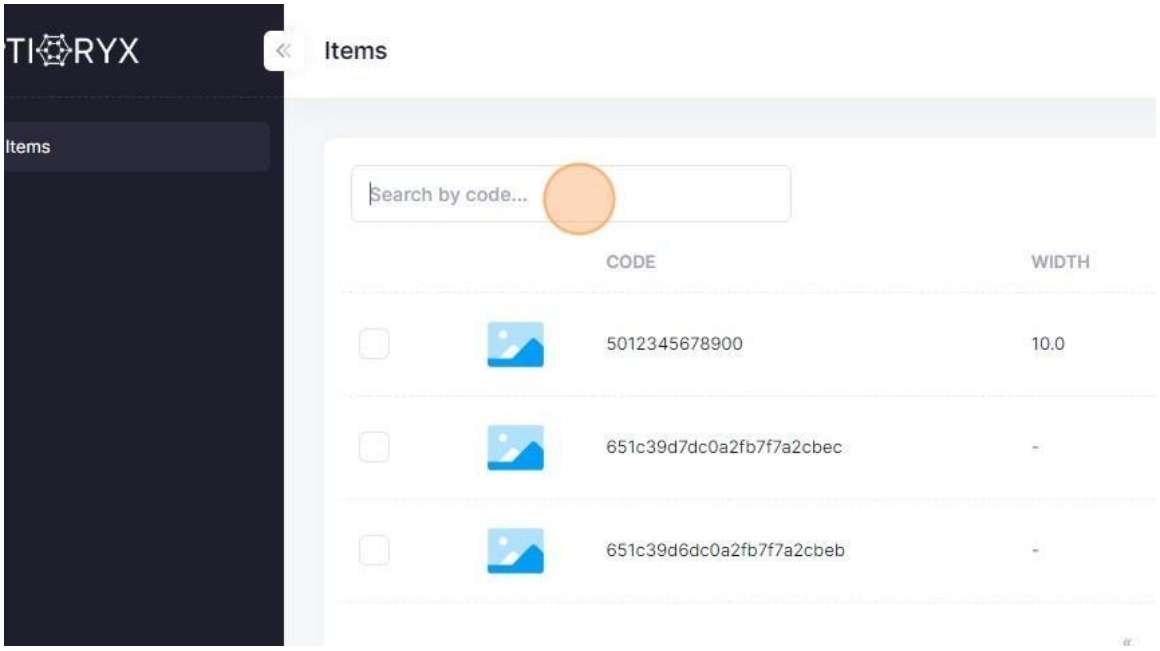
```yaml
master-data-store-web:
    build:
      context: ./master_data_store_web
    environment:
      - REACT_APP_OPENAPI_LOCATION=http://master-data-store
      - REACT_APP_API_LOCATION=http://127.0.0.1
    ports:
      - "3011:3011"
    depends_on:
      - master-data-store
    restart: on-failure
```

Environment variables

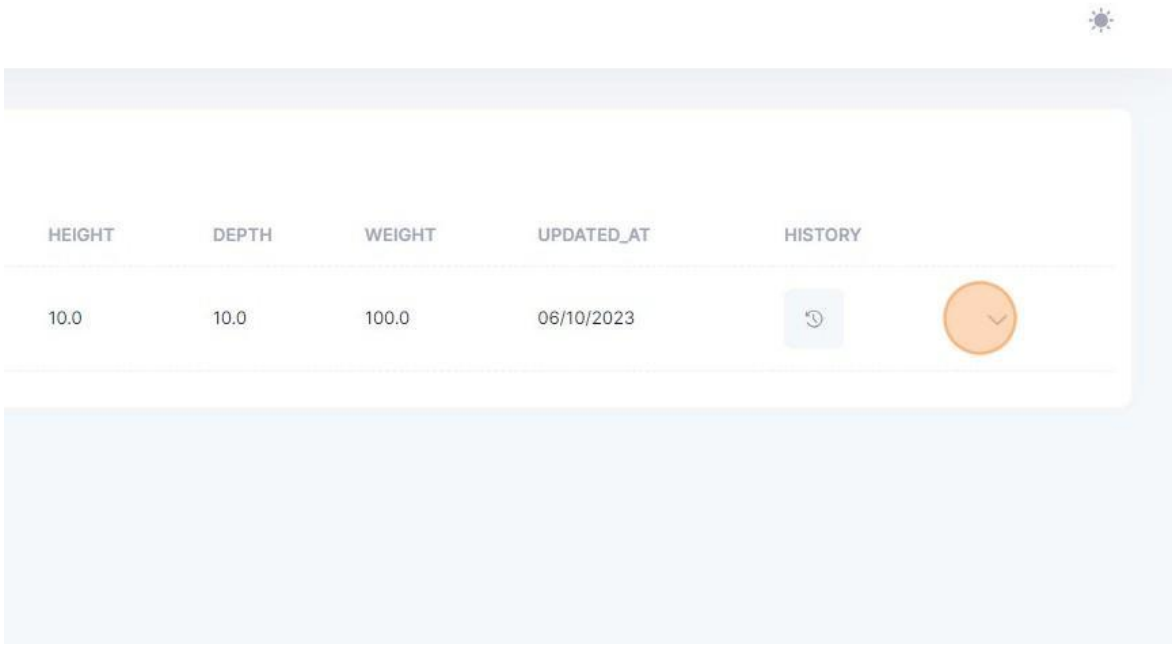
- REACT_APP_OPENAPI_LOCATION*: internal URL to the Master Data Store
- REACT_APP_API_LOCATION*: external URL to the Master Data Store

How to use

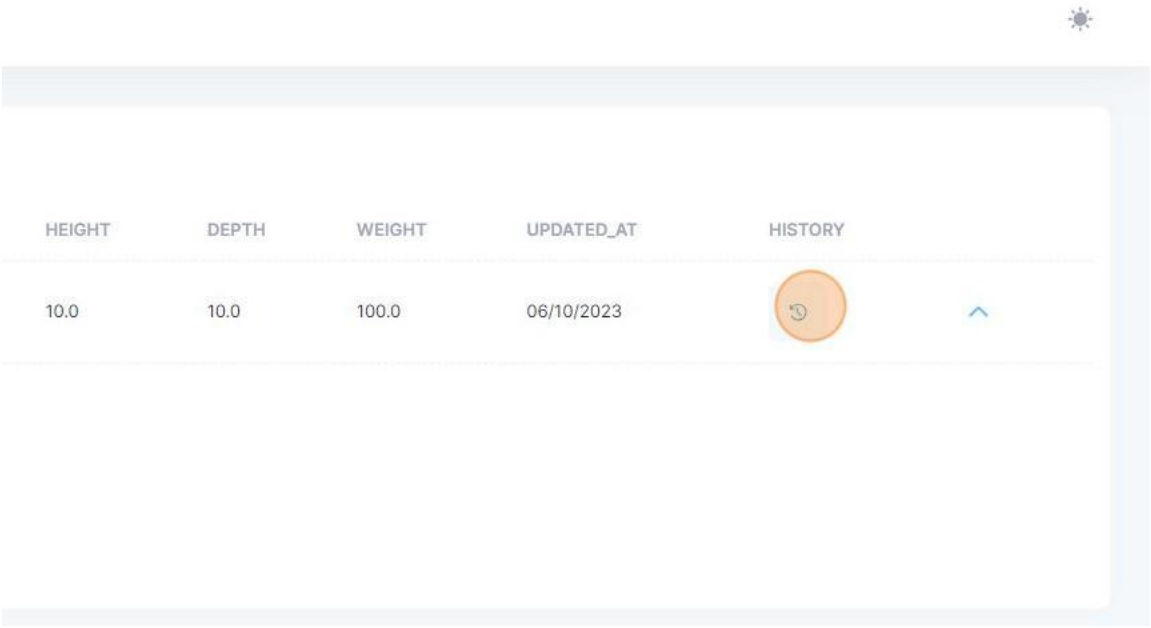
1. Click the "Search by code..." field.



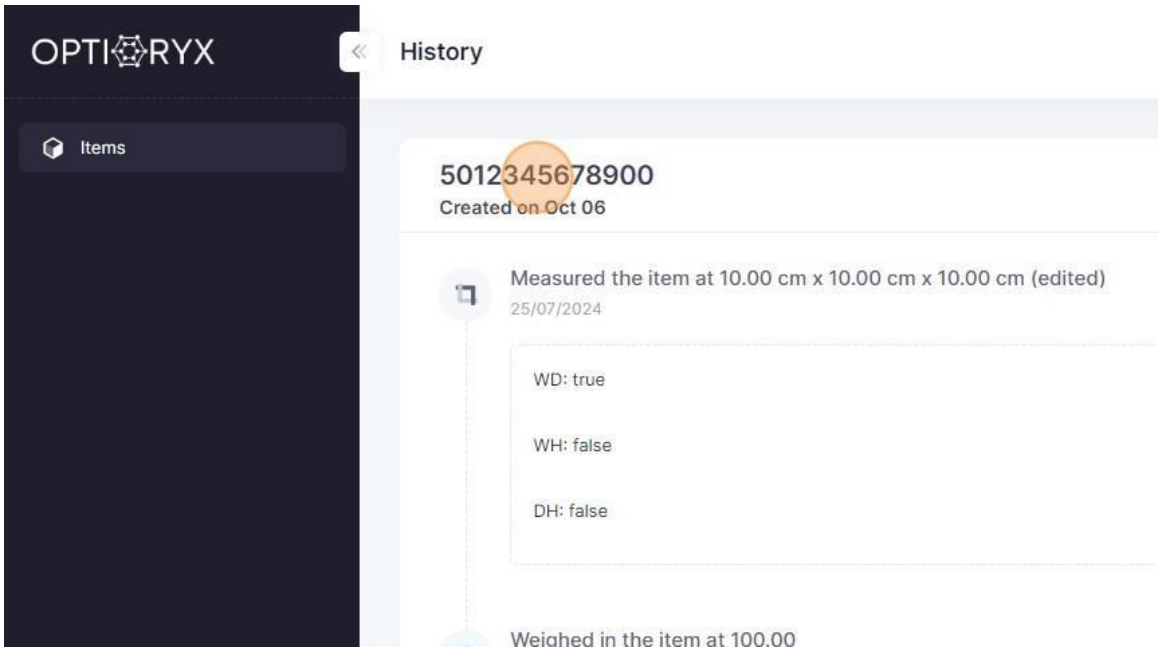
2. Type the barcode and press Enter
3. Open the collapsible to view details.



4. Click the history button to view the entire history for this item.



5. This will bring you to the history page.



BIN PACKING (ORION)

The Bin Packing Orion integration is a bin packing algorithm that uses data from the Master Data Store. Results from this algorithm are stored in a MongoDB collection, and can be viewed in the Truckload Building Web service. Additionally, step instructions can be generated optionally and viewed in the Adin webapp.

How to set up

Prerequisites

- fiware/orion-ld:latest

Configure the Orion integration

```yaml
orion-bp-bridge:
    build:
      context: ./orion_bp_bridge
    environment:
      - ORION_URL=http://orion:1026
      - MDS_URL=http://master-data-store
      - API_KEY=012346789abcdef
      - MONGODB_URL=mongodb://mongo:27017
      - POSTGRES_USER=postgres
      - POSTGRES_PASSWORD=password
      - POSTGRES_DB=adin
      - DB_HOST=adin-db
      - MINIO_URL=http://minio1:9000
      - BUCKET_URL=http://127.0.0.1:9001/optipackimages/
      - MINIO_ACCESS_KEY=admin
      - MINIO_SECRET_KEY=password
    ports:
      - "1018:1018"
    depends_on:
      - orion
    restart: on-failure
```

Environment variables

- ORION_URL*: the URL to your Orion service

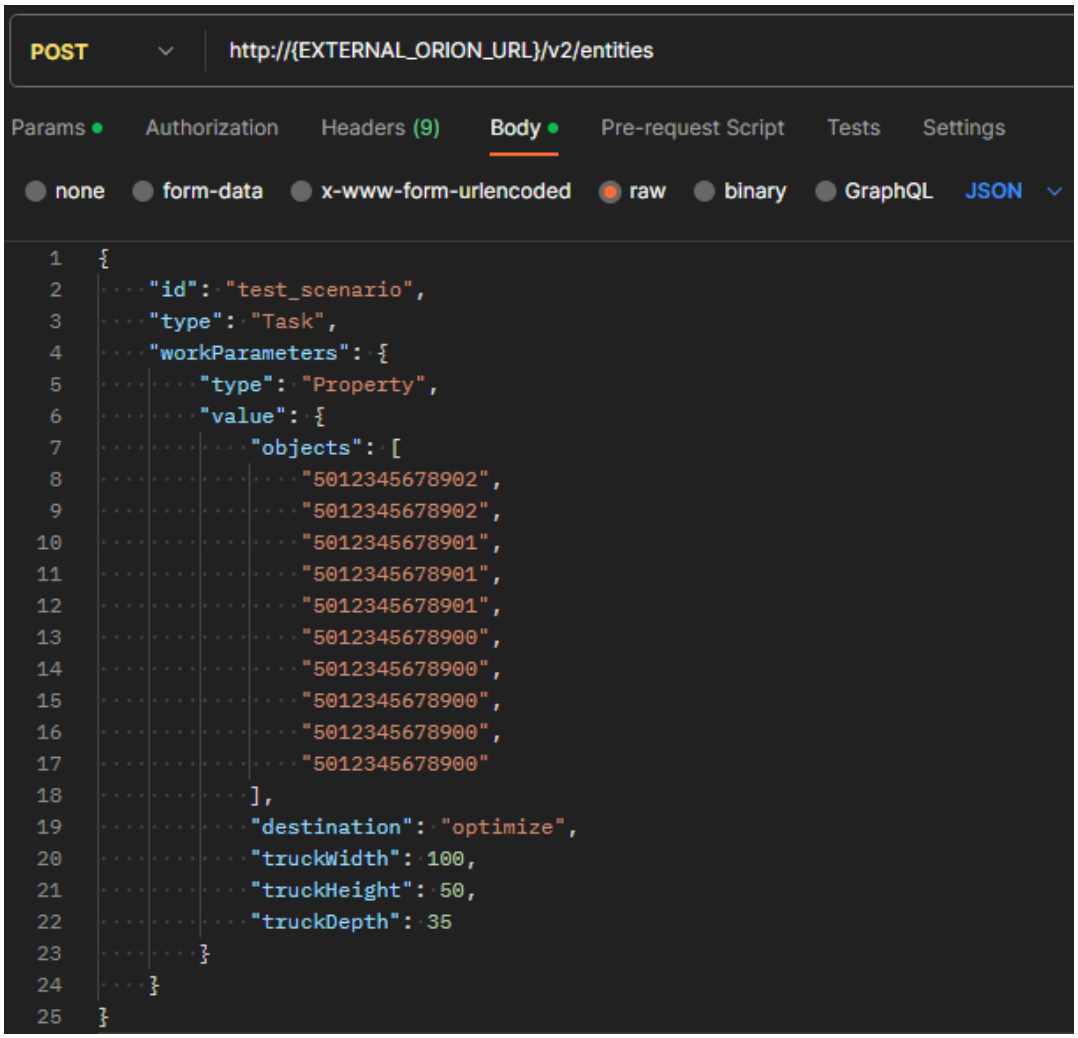
- MDS_URL*: the URL to your Master Data Store service
- API_KEY*: your API key for the Optioryx API[2]
- MONGODB_URL*: the connection string to your MongoDB
- POSTGRES_USER: the username for your Adin DB service
- POSTGRES_PASSWORD: the password for your Adin DB service
- DB_HOST: the name of your Adin DB service
- MINIO_URL: the URL for your Minio service
- BUCKET_URL: the URL to the Minio bucket on which to store the images
- MINIO_ACCESS_KEY: your Minio service access key
- MINIO_SECRET_KEY: your Minio service secret key

The latter portion of these variables are only required if you wish to view the packing results through Adin.

[2] Optioryx API: request your key at hello@optioryx.com

How to use

Use the “optimize” destination, providing a workParameter with the truck dimensions (“truckWidth”, “truckHeight”, “truckDepth”) and a value containing an array “objects” that contains the barcodes of the items to pack.



TRUCKLOAD BUILDING (WEB)

The Truckload Building webapp is built in Next.js and offers an overview of the optimizations as well as a 3D visualization of the truckload builds. It retrieves optimizations from a MongoDB collection, populated with data processed by the Bin Packing Orion integration.

How to set up

Prerequisites

- orion-bp-bridge

Configure the service

```yaml
truckload-building-web:
    build:
      context: ./truckload_building_web
    environment:
      - DATABASE_URL=mongodb://mongo:27017/test
    ports:
      - "3012:3000"
    restart: on-failure
```

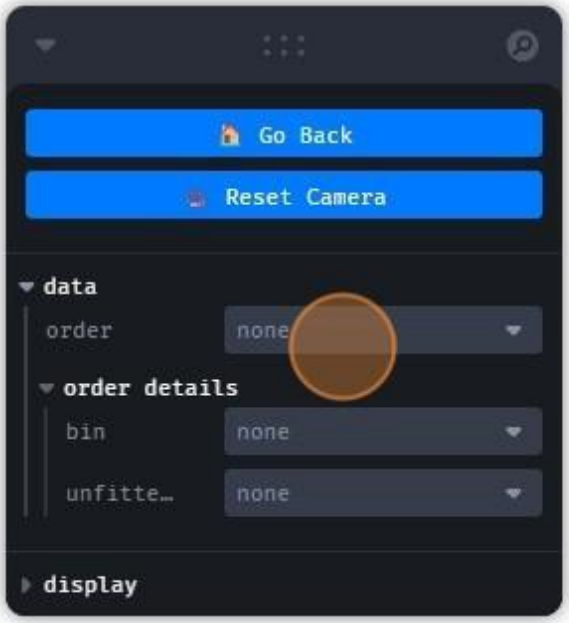
Environment variables

- DATABASE_URL: the connection string to your MongoDB

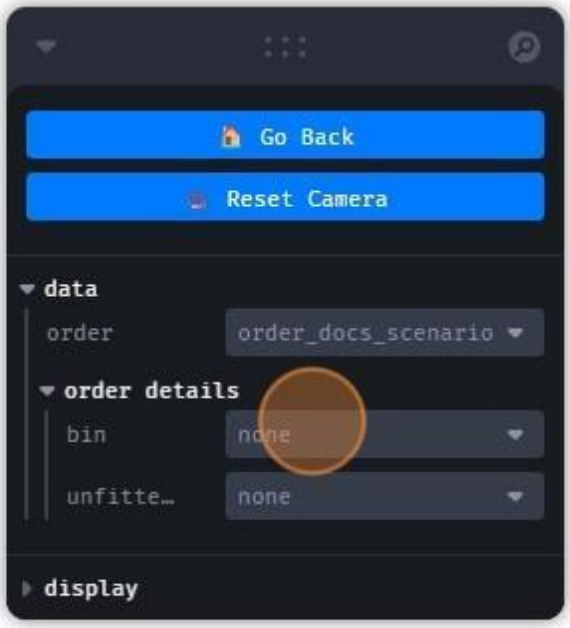
How to use

1. Select the optimization you wish to view.

2. Select an order.



3. Select a bin (truckload).



4. You can now view and interact with the visualization. Left-click while dragging to spin the truckload, right-click while dragging to pan the camera, scroll to zoom in and out.