Arukereso Orders API Technical Documentation

This document outlines the Arukereso Orders API, a RESTful API built using the Laravel framework and adhering to OpenAPI specifications. The API provides endpoints for retrieving a list of orders, filtering orders based on specific parameters, creating new orders and updating order statuses.

Swagger documentation of the API is available:

http://127.0.0.1:8000/api/doc

API Endpoints and JSON Input Structure

The API includes endpoints such as retrieving a **list of orders**, filtering orders by parameters, **creating new orders, updating order statuses**.

„List Orders with Filters” endpoint: allows users to filter orders based on parameters such as order ID, status, start date, and end date, all parameters are optional.

```
{
  "order_id": "integer",
  "status": {
    "name": "string"
  },
  "start_date": "date",
  "end_date": "date"
}
```

```
curl -X 'POST' \
  'http://127.0.0.1:8000/api/orders/list' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -H 'X-CSRF-TOKEN: ' \
  -d '{
  "order_id": 1,
  "status": {
    "name": "new"
  },
  "start_date": "2024-01-10",
  "end_date": "2024-01-14"
}'
```

„Creating New Order” endpoint: create a new order by the provided JSON structure. To validate the order request, it is important that both the shipping method and products are present and accessible in the database.

```
{
  "customer": {
    "name": "string",
    "email": "string"
  },
  "shipping_method": {
    "name": "string"
  },
  "billing_address": {
    "name": "string",
    "postal_code": "string",
    "city": "string",
    "street": "string"
  },
  "shipping_address": {
    "name": "string",
    "postal_code": "string",
    "city": "string",
    "street": "string"
  },
  "products": [
    {
      "name": "string",
      "quantity": "integer"
    }
  ]
}
```

```
curl -X POST \
  http://127.0.0.1:8000/api/orders \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -H 'X-CSRF-TOKEN: ' \
  -d '{
    "customer": {
      "name": "John Doe",
      "email": "john.doe@example.com"
    },
    "shipping_method": "home_delivery",
    "billing_address": {
      "name": "John Doe",
      "postal_code": "12345",
      "city": "Example City",
      "street": "123 Main Street"
    },
    "shipping_address": {
      "name": "John Doe",
      "postal_code": "54321",
      "city": "Shipping City",
      "street": "456 Shipping Street"
    },
    "products": [
      {
        "name": "eius",
        "quantity": 10
      },
      {
        "name": "perspiciatis",
        "quantity": 10
      }
    ]
  }'
```

„Update Order” endpoint is designed to modify the status of a particular order. The function will only execute if there is a change in the order status; otherwise, it will not proceed.

```
{
  "order_id": "integer",
  "status": {
    "name": "string"
  }
}
```

```
curl -X 'PUT' \
  'http://127.0.0.1:8000/api/orders/1' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -H 'X-CSRF-TOKEN: ' \
  -d '{
  "order_id": 1,
  "status": {
    "name": "new"
  }
}'
```

Alternatively, you have the option to update order statuses in bulk by providing an array of JSON objects. Each object within the array should represent a unique order with the desired status update. This allows for a more efficient mass update of order statuses.

```
[
  {
    "order_id":
"integer",
    "status": {
      "name": "string"
    }
  },
  {
    "order_id":
"integer",
    "status": {
      "name": "string"
    }
  }
]
```

```
curl -X 'PUT' \
  'http://127.0.0.1:8000/api/orders/1' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -H 'X-CSRF-TOKEN: ' \
  -d '[{
    "order_id": "1",
    "status": {
      "name": "completed"
    },
      "order_id": "2",
    "status": {
      "name": "completed"
    },
      ]
  }'
```