

Endpoints

Metadata

GET /flavors

body

```
{}
```

response

```
status 200
body: Array<Flavor>

interface Flavor{
  label: string;
  imgUrl: string;
}
```

GET /containers

body

```
{}
```

response

```
status 200
body: Array<ContainerType>

interface ContainerType{
  label: string;
  maxFlavors: number;
  variableQuantityOfFlavors: boolean;
}
```

GET /addins

body

```
{}
```

response

```
status 200
body: Array<Addin>

interface Addin{
  label: string;
```

```
imgURL: string;
}
```

GET /sauces

body

```
{}
```

response

```
status 200
body: Array<Sauce>

interface Sauce{
  label: string;
  imgURL: string;
}
```

Core

POST /session/:deviceId

open the session and identify the device with the server
TODO: implement a Logged user strategy

body

```
{}
```

response

```
status 200
body deviceId
```

Order

POST /order

Send new order to the server

body

```
interface Order {
  _id: number;
  containerType: ContainerType;
  flavors: Array<Flavor>;
  sauce: Sauce;
  addins: Array<Addin>;
  quantity: number;
  delivered: boolean; // false by default
}
```

response

```
status 200
body orderID
```

GET /orders/:deviceId

Get all the not delivered orders by the deviceId

body

```
{}
```

response

```
status 200
body Array<Order>

interface Order {
  _id: number;
  containerType: ContainerType;
  flavors: Array<Flavor>;
  sauce: Sauce;
  addins: Array<Addin>;
  quantity: number;
  delivered: boolean;
}
```

Location

GET /location/:orderId

Get the location of an order

body

```
{}
```

response

```
status 200
body LatLng

interface LatLng {
  latitude: number;
  longitude: number;
}
```

Review

POST /review

Send a review to the server

body

```
interface Review {  
  rating: Number;  
  img: String; // base64 image string  
  comment: String;  
}
```

response

```
status 200  
body reviewID
```