

# 000884285 Server-Side Assignment API

This API helps you order food from a selected group of restaurants, which meet a certain criteria of excellence. Create an account or login as a guest to order food. By creating an account, you can view your order history from when you first used the website with this account. There is no need to worry about delivery delays as well. Our collaboration with other top food delivery companies aids us to provide you with the best delivery services. Quicker than you could have ever imagined.  
More information: <https://helloverb.com>  
Contact Info: [hello@helloverb.com](mailto:hello@helloverb.com)  
Version: 1.0  
MIT  
<http://apache.org/licenses/LICENSE-2.0.html>

## Access

## Methods

[ [Jump to Models](#) ]

## Table of Contents

### Default

- [GET /addToCart](#)
- [GET /checkout/order](#)
- [GET /comment](#)
- [GET /CurrentUserInfo](#)
- [GET /list](#)
- [GET /userDatabase](#)
- [POST /addToCart/changeJSON](#)
- [POST /addToCart/newEntry](#)
- [POST /comment/add](#)
- [POST /register](#)
- [POST /updateOrderHistory](#)
- [POST /User](#)

## Default

GET /addToCart

Up

StoresFoodIDinCart (getAddToCart)

Returns the id of Food in Cart and the respective quantity in JSON

Return type

array[\[inline response 200 3\]](#)

Example data

Content-Type: application/json

[ {  
 "quantity" : 6.027456183070403,  
 "food" : 0.8008281904610115  
}, {  
 "quantity" : 6.027456183070403,  
 "food" : 0.8008281904610115  
} ]

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

OK

Example data

Content-Type: example-200

[ {"food":5,"quantity":2},{ "food":7,"quantity":2}]

GET /checkout/order

Up

UserCheckout (getCheckoutOrder)

This returns the complete details of the items the customer has placed in cart in JSON format

Return type

array[\[inline response 200 2\]](#)

Example data

Content-Type: application/json

```
[ {
  "image" : "image",
  "price" : "price",
  "restaurant" : "restaurant",
  "name" : "name",
  "optionId" : 0.8008281904610115
}, {
  "image" : "image",
  "price" : "price",
  "restaurant" : "restaurant",
  "name" : "name",
  "optionId" : 0.8008281904610115
} ]
```

#### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

#### Responses

200

OK

#### Example data

Content-Type: example-200

```
[{"optionId":21,"name":"Zizzi Special Pepperoni Pizza","restaurant":"Zizzi","price":"20","image":"images/food/pepperoni.
```

## GET /comment

ReturnCommentDatabase (**getComment**)

Returns a database which contains all the comments posted by uses of the website

#### Return type

array[[inline response 200 8](#)]

#### Example data

Content-Type: application/json

```
[ {
  "comment" : "comment",
  "email" : "email",
  "customer" : "customer"
}, {
  "comment" : "comment",
  "email" : "email",
  "customer" : "customer"
} ]
```

#### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

#### Responses

200

OK

#### Example data

Content-Type: example-200

```
[{"comment":" goooooood","customer":"vincent","email":"zhnehaozhao46@gmial.com"}]
```

## GET /CurrentUserInfo

ReturnNameOfCurrentUSer (**getCurrentUserInfo**)

Returns the Username of the current user. If no username is detected current user is logged in as a guest.

#### Return type

String

#### Example data

Content-Type: application/json

```
" "
```

#### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

#### Responses

200  
OK [String](#)  
Example data  
Content-Type: example-200

William
---------

GET /list

Up

ReturnFoodMenu ([getList](#))

Returns the entire food menu stored in JSON

Return type  
[inline\\_response\\_200\\_6](#)

Example data  
Content-Type: application/json

```
{
  "country" : [ {
    "country" : "country",
    "image" : "image",
    "price" : "price",
    "restaurant" : "restaurant",
    "name" : "name",
    "optionId" : 0.8008281904610115
  }, {
    "country" : "country",
    "image" : "image",
    "price" : "price",
    "restaurant" : "restaurant",
    "name" : "name",
    "optionId" : 0.8008281904610115
  } ],
  "chicken" : [ {
    "image" : "image",
    "price" : "price",
    "restaurant" : "restaurant",
    "name" : "name",
    "optionId" : 5.962133916683182
  }, {
    "image" : "image",
    "price" : "price",
    "restaurant" : "restaurant",
    "name" : "name",
    "optionId" : 5.962133916683182
  } ],
  "lunch" : [ null, null ],
  "pizza" : [ null, null ],
  "dessert" : [ {
    "image" : "image",
    "price" : "price",
    "restaurant" : "restaurant",
    "name" : "name",
    "optionId" : 1.4658129805029452
  }, {
    "image" : "image",
    "price" : "price",
    "restaurant" : "restaurant",
    "name" : "name",
    "optionId" : 1.4658129805029452
  } ],
  "sandwiches" : [ null, null ],
  "breakfast" : [ null, null ],
  "burgers" : [ {
    "image" : "image",
    "price" : "price",
    "restaurant" : "restaurant",
    "name" : "name",
    "optionId" : 6.027456183070403
  }, {
    "image" : "image",
    "price" : "price",
    "restaurant" : "restaurant",
    "name" : "name",
    "optionId" : 6.027456183070403
  } ],
  "wine" : [ {
    "image" : "image",
    "price" : "price",
    "restaurant" : "restaurant",
    "name" : "name",
    "optionId" : 5.637376656633329
  }, {
    "image" : "image",
    "price" : "price",
    "restaurant" : "restaurant",
    "name" : "name",
```

```
    "optionId" : 5.637376656633329
  } ]
}
```

#### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json
- application/xml

#### Responses

200

OK [inline\\_response\\_200\\_6](#)

#### Example data

Content-Type: example-200

```
{ "country": [ { "optionId": 1, "country": "Ghana", "name": "Spicy Jollof rice", "restaurant": "Mama Jays", "price": "20", "image": "im
```

## GET /userDatabase

### GetAccountDetails (getUserDatabase)

This request returns a database which contains the username, email, password and the previous orders per account created.

#### Return type

array[\[inline\\_response\\_200\]](#)

#### Example data

Content-Type: application/json

```
[ {
  "password" : "password",
  "userEmail" : "userEmail",
  "user" : "user",
  "prevOrders" : [ "prevOrders", "prevOrders" ]
}, {
  "password" : "password",
  "userEmail" : "userEmail",
  "user" : "user",
  "prevOrders" : [ "prevOrders", "prevOrders" ]
} ]
```

#### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

#### Responses

200

OK

#### Example data

Content-Type: example-200

```
[ { "user": "William", "userEmail": "william@example.com", "password": "123", "prevOrders": [ "Ham burger with Cheese and Patties"
```

## POST /addToCart/changeJSON

### UpdateOldDishesInCart (postAddToCartChangeJSON)

Whenever an old dish has already been added to cart and changes are detected this request can be triggered to make changes to the JSON storing this items.

#### Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

#### Request body

body [body\\_5](#) (optional)

*Body Parameter* – Updates items that have been added to cart already and returnn new updated JSON

example: {

```
  "value" : {
    "food" : 25,
    "quantity" : 3
  }
}
```

#### Request headers

#### Return type

array[\[body\\_5\]](#)

#### Example data

Content-Type: application/json

```
[ {
  "quantity" : 6.027456183070403,
  "food" : 0.8008281904610115
}, {
  "quantity" : 6.027456183070403,
  "food" : 0.8008281904610115
} ]
```

#### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

#### Responses

200

OK

#### Example data

Content-Type: example-200

```
[{"food":23,"quantity":5},{ "food":2,"quantity":2},{ "food":25,"quantity":3}]
```

## POST /addToCart/newEntry

### AddNewDishToCart (postAddToCartNewEntry)

Whenever a new dish is detected adds new dish to the cart and stores dish in an array. The ID number of the dish along with its quantity are the keys stored.

#### Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

#### Request body

body [body\\_1](#) (optional)

*Body Parameter* – Posts the ID number and default a quantity of 1 of the new dish.

example: {

```
  "value" : {
    "food" : 25,
    "quantity" : 1
  }
}
```

#### Request headers

#### Return type

array[[inline\\_response\\_200\\_4](#)]

#### Example data

Content-Type: application/json

```
[ {
  "quantity" : 6.027456183070403,
  "food" : 0.8008281904610115
}, {
  "quantity" : 6.027456183070403,
  "food" : 0.8008281904610115
} ]
```

#### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

#### Responses

200

OK

#### Example data

Content-Type: example-1

```
[{"food":25,"quantity":1}]
```

## POST /comment/add

### AddNewComment (postCommentAdd)

Posts new comments added by user and stores it in a JSON file. The update comments are then returned

#### Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

#### Request body

##### body [body\\_4](#) (optional)

*Body Parameter* – Posts new comments added by users and stores them in a JSON file. Returns an updated list of all comments made on website

example: {  
 "value" : {  
 "newComment" : "Great stuff, absolutely loved it",  
 "customerName" : "Mak",  
 "customerEmail" : "mak@example.com"  
 }  
}

#### Request headers

##### Return type

array[[inline\\_response\\_200\\_10](#)]

##### Example data

Content-Type: application/json

```
[ {  
  "comment" : "comment",  
  "email" : "email",  
  "customer" : "customer"  
}, {  
  "comment" : "comment",  
  "email" : "email",  
  "customer" : "customer"  
} ]
```

##### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

#### Responses

200

OK

##### Example data

Content-Type: example-200

```
[{"comment":" goooooood","customer":"vincent","email":"zhnehaozhao46@gmial.com"}, {"comment":"This is a Jest Comment test
```

## POST /register

#### StoreDetailsOfCurrentlyRegisteredUser (postRegister)

Sends registration details of user in JSON and stores in a json file

##### Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

#### Request body

##### body [body\\_3](#) (optional)

*Body Parameter* – JSON data containing registration details of new user. This includes the username, email, password and password confirmation.

example: {  
 "value" : {  
 "regUsername" : "Mark",  
 "regEmail" : "mark@example.com",  
 "regPassword" : "mak",  
 "regConfPassword" : "mak"  
 }  
}

#### Request headers

##### Return type

array[[inline\\_response\\_200\\_9](#)]

##### Example data

Content-Type: application/json

```
[ {  
  "password" : "password",  
  "userEmail" : "userEmail",  
  "user" : "user",  
  "prevOrders" : [ { }, { } ]  
}
```

```
}, {
  "password" : "password",
  "userEmail" : "userEmail",
  "user" : "user",
  "prevOrders" : [ { }, { } ]
} ]
```

#### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

#### Responses

200

OK

#### Example data

Content-Type: example-1

```
[{"user": "Mark", "userEmail": "mark@example.com", "password": "mak", "prevOrders": [{}]}]
```

## POST /updateOrderHistory

#### AddOrdersToUserOrderHistory (postUpdateOrderHistory)

Add new orders to users order history stored in JSON file. Read JSON file and writes to JSON file.

#### Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

#### Request body

body [body\\_2](#) (optional)

*Body Parameter* – Posts the new order to be stored in JSON file and returns the update list of previous orders

*example:* {  
 "value" : {  
 "newOrders" : "Nandos Grilled chicken and fries"  
 }  
}

#### Request headers

#### Return type

[inline\\_response\\_200\\_5](#)

#### Example data

Content-Type: application/json

```
{
  "password" : "password",
  "userEmail" : "userEmail",
  "user" : "user",
  "prevOrders" : [ "", "" ]
}
```

#### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json
- application/xml
- multipart/form-data

#### Responses

200

OK [inline\\_response\\_200\\_5](#)

#### Example data

Content-Type: example-200

```
{"user": "William", "userEmail": "william@example.com", "password": "123", "prevOrders": ["Ham burger with Cheese and Patties",
```

## POST /User

#### PostUserName (postUser)

Stores temporarily the name of the current user

#### Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

**body** [body](#) (optional)  
*Body Parameter* – The name of the user  
*example:* {  
  "value" : {  
    "User" : "vincent"  
  }  
}

Request headers

Return type  
[inline\\_response\\_200\\_1](#)

Example data  
Content-Type: application/json

```
{  
  "currentUser" : "currentUser"  
}
```

**Produces**  
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

**Responses**  
200  
OK [inline\\_response\\_200\\_1](#)  
**Example data**  
Content-Type: example-200

```
{ "currentUser": "Vincent" }
```

Models

[ [Jump to Methods](#) ]

Table of Contents

- 1. [Comments](#)
- 2. [Food](#)
- 3. [User - User](#)
- 4. [body](#)
- 5. [body\\_1](#)
- 6. [body\\_2](#)
- 7. [body\\_3](#)
- 8. [body\\_4](#)
- 9. [body\\_5](#)
- 10. [inline\\_response\\_200](#)
- 11. [inline\\_response\\_200\\_1](#)
- 12. [inline\\_response\\_200\\_10](#)
- 13. [inline\\_response\\_200\\_2](#)
- 14. [inline\\_response\\_200\\_3](#)
- 15. [inline\\_response\\_200\\_4](#)
- 16. [inline\\_response\\_200\\_5](#)
- 17. [inline\\_response\\_200\\_6](#)
- 18. [inline\\_response\\_200\\_6\\_burgers](#)
- 19. [inline\\_response\\_200\\_6\\_chicken](#)
- 20. [inline\\_response\\_200\\_6\\_country](#)
- 21. [inline\\_response\\_200\\_6\\_dessert](#)
- 22. [inline\\_response\\_200\\_6\\_wine](#)
- 23. [inline\\_response\\_200\\_7](#)
- 24. [inline\\_response\\_200\\_7\\_burgers](#)
- 25. [inline\\_response\\_200\\_7\\_country](#)
- 26. [inline\\_response\\_200\\_7\\_offers](#)
- 27. [inline\\_response\\_200\\_8](#)
- 28. [inline\\_response\\_200\\_9](#)

Comments

**comment**  
[String](#) Users comment

**customer**  
[String](#) Name of user who commented

**email**  
[String](#) Email of user who commented

[Up](#)

Food

**country**

[Up](#)



[\*String\*](#)

## User - User

[Up](#)

The Users registered on website

**user**  
[\*String\*](#) Unique username of the customer

**userEmail**  
[\*String\*](#) Email of the customer

**password**  
[\*String\*](#) Password of the customer

**prevOrders**  
[\*array\[Object\]\*](#) List of customers previous orders

## body

[Up](#)

**User**  
[\*String\*](#) Name of Current User

## body\_1

[Up](#)

New Order in JSON

**food**  
[\*BigDecimal\*](#) ID of the dish

**quantity**  
[\*BigDecimal\*](#) Quantity of dish

## body\_2

[Up](#)

**newOrders**  
[\*String\*](#) New Order

## body\_3

[Up](#)

**regUsername**  
[\*String\*](#) Username of currently registered user

**regEmail**  
[\*String\*](#) Email of currently registered user

**regPassword**  
[\*String\*](#) Password of currently registered user

**regConfPassword**  
[\*String\*](#) Confirm password of currently registered user

## body\_4

[Up](#)

**newComment**  
[\*String\*](#) New comment posting

**customerName**  
[\*String\*](#) Name of customer posting new comment

**customerEmail**  
[\*String\*](#) Email of customer posting new comments

## body\_5

[Up](#)

**food**  
[\*BigDecimal\*](#)

**quantity**  
[\*BigDecimal\*](#)

## inline\_response\_200

[Up](#)

Account holder Details

**user**  
[\*String\*](#) Username of account holder

**userEmail**  
[\*String\*](#) Email address of account holder

**password**  
[\*String\*](#) Password of account holder for authentication

**prevOrders**  
[\*array\[String\]\*](#) Contain a list of previous orders by account holder

**inline\_response\_200\_1** [Up](#)

currentUser (optional)  
[String](#) Name of the Current User

**inline\_response\_200\_10** [Up](#)

comment  
[String](#) Comment of user

customer  
[String](#) Name of customer

email  
[String](#) Email of customer

**inline\_response\_200\_2** [Up](#)

optionId  
[BigDecimal](#) ID number of dish

name  
[String](#) Name of dish

restaurant  
[String](#) Name of restaurant that makes dish

price  
[String](#) Price of the dish

image  
[String](#) Image URL of dish

**inline\_response\_200\_3** [Up](#)

food  
[BigDecimal](#) The dish ID of the ordered food

quantity  
[BigDecimal](#) The order quantity of that specific dish

**inline\_response\_200\_4** [Up](#)

List of dishes

food  
[BigDecimal](#) ID of respective dishes in cart

quantity  
[BigDecimal](#) Quantity of the respective dishes

**inline\_response\_200\_5** [Up](#)

user  
[String](#)

userEmail  
[String](#)

password  
[String](#)

prevOrders  
[array\[null\]](#)

**inline\_response\_200\_6** [Up](#)

Food Menu Categorised into different food groups

country  
[array\[inline\\_response\\_200\\_6\\_country\]](#) Foods based on Country

burgers  
[array\[inline\\_response\\_200\\_6\\_burgers\]](#) Menu of burgers

dessert  
[array\[inline\\_response\\_200\\_6\\_dessert\]](#) Menu of dessert

chicken  
[array\[inline\\_response\\_200\\_6\\_chicken\]](#) Menu of chicken

wine  
[array\[inline\\_response\\_200\\_6\\_wine\]](#) Menu of wine

pizza  
[array\[inline\\_response\\_200\\_6\\_wine\]](#) Menu of pizza

sandwiches  
[array\[inline\\_response\\_200\\_6\\_wine\]](#) Menu of sandwiches

lunch  
[array\[inline\\_response\\_200\\_6\\_wine\]](#) Menu of lunch

breakfast  
[array\[inline\\_response\\_200\\_6\\_wine\]](#) Menu of breakfast

#### inline\_response\_200\_6\_burgers

[Up](#)

optionId  
[BigDecimal](#) Food ID

name  
[String](#) Name of food

restaurant  
[String](#) Restaurant food is cooked

price  
[String](#) Price of the food

image  
[String](#) Image URL of food.

#### inline\_response\_200\_6\_chicken

[Up](#)

Food ID

optionId  
[BigDecimal](#) Name of food

name  
[String](#) Restaurant food is cooked

restaurant  
[String](#) Price of the food

price  
[String](#) Price of the food

image  
[String](#) Image URL of food.

#### inline\_response\_200\_6\_country

[Up](#)

optionId  
[BigDecimal](#) Food ID

country  
[String](#) Country of Origin

name  
[String](#) Name of food

restaurant  
[String](#) Restaurant food is cooked

price  
[String](#) Price of the food

image  
[String](#) Image URL of food.

#### inline\_response\_200\_6\_dessert

[Up](#)

optionId  
[BigDecimal](#) Food ID

name  
[String](#) Name of food

restaurant  
[String](#) Restaurant food is cooked

price  
[String](#) Price of the food

image  
[String](#) Image URL of food.=

#### inline\_response\_200\_6\_wine

[Up](#)

optionId  
[BigDecimal](#) Food ID

name  
[String](#) Name of food

restaurant  
[String](#) Restaurant food is cooked

price  
[String](#) Price of the food

image  
[String](#) Image URL of food.

**inline\_response\_200\_7** [Up](#)

country  
[array\[inline\\_response\\_200\\_7\\_country\]](#)  
burgers  
[array\[inline\\_response\\_200\\_7\\_burgers\]](#)  
dessert  
[array\[inline\\_response\\_200\\_7\\_burgers\]](#)  
chicken  
[array\[inline\\_response\\_200\\_7\\_burgers\]](#)  
wine  
[array\[inline\\_response\\_200\\_7\\_burgers\]](#)  
pizza  
[array\[inline\\_response\\_200\\_7\\_burgers\]](#)  
sandwiches  
[array\[inline\\_response\\_200\\_7\\_burgers\]](#)  
lunch  
[array\[inline\\_response\\_200\\_7\\_burgers\]](#)  
breakfast  
[array\[inline\\_response\\_200\\_7\\_burgers\]](#)  
offers  
[array\[inline\\_response\\_200\\_7\\_offers\]](#)

**inline\_response\_200\_7\_burgers** [Up](#)

optionId  
[BigDecimal](#)  
name  
[String](#)  
restaurant  
[String](#)  
price  
[String](#)  
image  
[String](#)

**inline\_response\_200\_7\_country** [Up](#)

optionId  
[BigDecimal](#)  
country  
[String](#)  
name  
[String](#)  
restaurant  
[String](#)  
price  
[String](#)  
image  
[String](#)

**inline\_response\_200\_7\_offers** [Up](#)

name  
[String](#)  
restaurant  
[String](#)  
price  
[String](#)

**inline\_response\_200\_8** [Up](#)

Contains comment,name and email of user

comment  
[String](#) Comment of the user  
customer  
[String](#) Name of the user who posted comment  
email  
[String](#) Email of user who posted comment

**inline\_response\_200\_9** [Up](#)

**user**  
*[String](#)* Name of currently registered user

**userEmail**  
*[String](#)* Email of currently registered user

**password**  
*[String](#)* Password of currently registered user

**prevOrders**  
*[array\[Object\]](#)* List to store orders of currently registered user