# ND Gourmand API

*Author: Chao Luo*
*March 21, 2015*

## 1 Summary

All API calls documented here consist of an HTTP method and a relative path, and the path is relative to a configurable base URL.

## 2 Request

All API calls receive requests in the form of JSON. In general, the backend will refuse the request if it receives unexpected parameters.

## 3 Response

All responses, unless otherwise noted, are either formatted in JSON, whose MIME type is application/json, or HTML. JSON natively includes the types String, Number, Boolean, Array, Object as well as the singleton value null. In addition, this document uses Integer to refer to whole Numbers, Datetime String to refer to String values consisting of an absolute timestamp in the web-standard format ISO 8601, and Money String to refer to String values consisting of an unformatted representation of a floating point value. All JSON responses include the errors array, which is the APIs mechanism for reporting errors.

Error:

| Parameter | Description | Type |
|---|---|---|
| Errors | A list of errors. The error cannot be null or undefined. | Array of error objects |

Error object:

| Parameter | Description | Type |
|---|---|---|
| Code | The error code | Integer |
| Message | The error message | String |

## 4 Resource Overview

- /restaurants
- /restaurants/{restID}
- /restaurants/{restID}/menus
- /restaurants/{restID}/menus/{menuID}
- /restaurants/{restID}/menus/{menuID}/categoreis
- /restaurants/{restID}/menus/{menuID}/categories/{catID}
- /restaurants/{restID}/menus/{menuID}/categoreis/{catID}/items

- /restaurants/{restID}/menus/{menuID}/categoreis/{catID}/items/{itemID}
- /users
- /users/registration
- /users/{userID}
- /users/{userID}/currentorders
- /orders
- /orders/{orderID}
- /orders/{orderID}/items
- /orders/{orderID}/item/{orderItemID}

(1) /restaurants

| Action | Description | Paramenters |
|---|---|---|
| GET | Get the restaurant list (the main page). See below for JSON form. | None |
| POST | Add a new restaurant | **kwargs |

Data model:

| Field Name | Description | Type | Required |
|---|---|---|---|
| restID | Restaurant ID | varchar(20) | Yes |
| name | Restaurant name | varchar(45) | Yes |
| address | Restaurant address | varchar(100) | Yes |
| city | Location | varchar(45) | Yes |
| state | Location | varchar(20) | Yes |
| zip | Zip code | varchar(10) | Yes |
| phone | Phone | varchar(20) | Yes |
| lat | Latitude | decimal(10, 8) | Yes |
| lng | Longitude | decimal(11, 8) | Yes |
| url | Website url | varchar(100) | No |

JSON form: {restID, name, lat, lng, address, city, state, zip, url}
POST response: None
Error messages:

| Code | Text |
|---|---|
| 5000 | The restaurant already exists. |

(2) /restaurants/{restID}:

| Action | Description | Parameters |
|---|---|---|
| GET | return"GET /restaurants/{id=%s} ... RestaurantID.GET" | restID |
| PUT | Update a restaurant | restID, **kwargs |
| DELETE | Delete a restaurant | restID |

(3) /restaurants/{restID}/menus:

| Action | Description | Paramenters |
|---|---|---|
| GET | Return the list of menus for a restaurant with restID. See below for JSON form. | restID |
| POST | Add a new menu | restID, **kwargs |

Data model:

| Field Name | Description | Type | Required |
|---|---|---|---|
| restID | Restaurant ID | varchar(20) | Yes |
| menuID | Menu ID | int(11) | Yes |
| menuName | Menu name | varchar(30) | No |

JSON form:
{’href’ : ’restaurants/{restID}/menus/{menuID}/categories’, ’name’ : menuName}
Error message:

| Code | Text |
|---|---|
| 1000 | The menu already exists. |

(4) /restaurants/{restID}/menus/{menuID}:

| Action | Description | Paramenters |
|---|---|---|
| GET | return ”GET /restaurants/{restID=%s}/menus/{menuID=%s} ... MenuID.GET” % (restID,menuID) | restID, menuID |
| PUT | Update menu id for restaurant with restID | restID, menuID, **kwargs |
| DELETE | return ”DELETE /restaurants/{restID=%s}/menus/{menuID=%s} ... MenuID.DELETE” % (restID,menuID) | restID, menuID |

(5) /restaurants/{restID}/menus/{menuID}/categories:

| Action | Description | Paramenters |
|---|---|---|
| GET | return the list of categories from the five menuID. See below for JSON data. | restID, menuID |
| POST | Add a new category | restID, menuID, **kwargs |

Data model:

| Field Name | Description | Type | Required |
|---|---|---|---|
| menuID | Menu ID | int(11) | Yes |
| sectionID | Category ID | int(11) | Yes |
| sectionName | Category name | varchar(50) | No |

JSON form: {
’href’: ’restaurants/{restID}/menus/{menuID}/categories/{catID}/items,
’name’: categoryName
}
(6) /restaurants/{restID}/menus/{menuID}/categories/{catID}:

| Action | Description | Paramenters |
|---|---|---|
| GET | return“GET /restaurants/{restID}/menus/{menuID}/categories/{catID} ... CategoryID.GET” | restID, menuID, catID |
| PUT | Update category id for restaurant id | restID, menuID, catID, **kwargs |
| DELETE | return “DELETE /restaurants/{restID}/menus/{menuID}/categories/{catID} ... CategoryID.DELETE” | restID, menuID, catID |

(7) /restaurants/{restID}/menus/{menuID}/categories/{catID}/items:

| Action | Description | Paramenters |
|---|---|---|
| GET | Return the list of items for the given category and restaurant. See below for the JSON data form. | restID, menuID, catID |
| POST | Add a new item to the category | restID, menuID, catID, **kwargs |

Data model:

| Field Name | Description | Type | Required |
|---|---|---|---|
| menuID | Menu ID | int(11) | Yes |
| sectionID | Category ID | int(11) | Yes |
| itemID | Item ID | int(11) | Yes |
| name | Item name | varchar(30) | Yes |
| price | Item price | decimal(6, 2) | Yes |
| description | Item description | varchar(100) | Yes |

JSON form: {
’href’: ’/restaurants/{restID}/menus/{menuID}/categories/{catID}/items/{itemID},
’name’: itemName,
’description’: desc,
’price’ : price
}

(8) /restaurants/{restID}/menus/{menuID}/categories/{catID}/items/{itemID}:

| Action | Description | Paramenters |
|---|---|---|
| GET | return “GET /restaurants/{restID}/menus/{menuID}/categories/{catID}/items/{itemID} ... ItemID.GET” | restID, menuID, catID, itemID |
| PUT | Update the item ID | restID, menuID, catID, itemID, **kwargs |
| DELETE | return “DELETE /restaurants/{restID}/menus/{menuID}/categories/{catID}/items/{itemID} ... ItemID.DELETE” | restID, menuID, catID, itemID |

(9) /users/registration:

| Action | Description | Paramenters |
|---|---|---|
| GET | Get the sign-up form. | None |
| POST | Create a new user | firstName, lastName, email, address, password, phone |

JSON form: {
firstName,
lastName,
email,
address,
password,
phone,
}
POST response: Return the user information if the insertion is successful or the error message if failure.
Error Messages:

| Code | Text |
|---|---|
| 8000 | User with the given email already exists |
| 8001 | Failed to add the user into database |
| 8002 | Some user information missing or invalid |

(10) /users/{userID}/currentorders:

| Action | Description | Paramenters |
|---|---|---|
| GET | Get the order list of userID. | userID |
| DELETE | Delete the orders of userID from the database | userID |

(11) /orders:

| Action | Description | Paramenters |
|---|---|---|
| GET | Return the list of orders | None |
| PUT | Add a new order or return the orderID for a given userID | userID |

Data model:

| Field Name | Description | Type | Required |
|---|---|---|---|
| orderID | Order ID | int | Yes |
| userID | User ID | int | Yes |
| paymentStatus | Payment | boolean | Yes |

JSON form: {
'href': 'orders/{orderID}/orderitems',
'userID': userID,
'payment': payment
}
(12) /orders/{orderID}:

| Action | Description | Paramenters |
|---|---|---|
| GET | Get orderItemID, quantity | orderID |
| DELETE | Delete this order | orderID |

Error Messages:

| Code | Text |
|---|---|
| 3000 | OrderID Does Not Exist |

(13) /orders/{orderID}/items/{orderitemID}:

| Action | Description | Paramenters |
|---|---|---|
| GET | Get orderItemID, quantity | orderID, itemID |
| PUT | Create or update an orderItem | orderID, itemID, **kwargs |
| DELETE | Delete an orderItem | orderID, itemID |

Data model:

| Field Name | Description | Type | Required |
|---|---|---|---|
| orderID | Order ID | varchar(20) | Yes |
| itemID | Item ID | int(11) | Yes |
| quantity | OrderItem quantity | int(11) | Yes |

JSON form: {
quantity: quantity
}
PUT response: orderItemID generated by the API during PUT.
Error messages:

| Code | Text |
|---|---|
| 9000 | Order with OrderID id Does Not Exist |
| 9001 | Item with ItemID id Does Not Exist |
| 9002 | Failed to add order item to shopping cart |
| 9003 | Expected integer “quantity” of items in order as JSON input |
| 9004<br>9005 | OrderItemID for OrderID id and ItemID Does Not Exist<br>Failed to delete order item from shopping cart |