**API Specification Doc**

**(*Food Fantasy App*)**

## NOTE: Please don’t request edit access to this demo document. Instead, click [here](https://docs.google.com/document/d/1HSQ3Fe77hnthw8hizqvXJU-qGEPHavMkctvCCadkVbY/template/preview?usp=drive_web) to make a copy of this template for your use. Thank you!

## Index

[NOTE: Please don’t request edit access to this demo document. Instead, click here to make a copy of this template for your use. Thank you!](#_Toc498232568)

[1. login](#_Toc498232569)

[Request](#_Toc498232570)

[Response](#_Toc498232571)

[2. Sign up](#_Toc498232572)

[Request](#_Toc498232573)

[Response](#_Toc498232574)

[3. Contact us](#_Toc498232575)

[Request](#_Toc498232576)

[Response](#_Toc498232577)

[4. get updates](#_Toc498232578)

[Request](#_Toc498232579)

[Response](#_Toc498232580)

[3. deletions](#_Toc498232581)

[Request](#_Toc498232582)

[Response](#_Toc498232583)

[4. get recipe image](#_Toc498232584)

[Request](#_Toc498232585)

[Response](#_Toc498232586)

5. create order

Request

Response

6. search

Request

Response

7. Delivery time slot

Request

Response

[Conventions](#_Toc498232587)

[Status Codes](#_Toc498232588)

## 

## 

## 

Methods

## 1. Login

Authenticate the user with the system and obtain the auth\_token

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/login/ |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| HEAD  POST  POST | api\_key  username  password | string  string  string |

**api\_key**

Api\_key must be sent with all client requests. The api\_key helps the server to validate the request source.

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | {  "auth\_key": <auth\_key>  }  auth\_key(**string**) - all further API calls must have this key in header |
| 403 | {"error":"API key is missing."} |
| 400 | {"error":"Please provide username."} |
| 400 | {"error":"Please provide password."} |
| 401 | {"error":"Invalid API key."} |
| 401 | {"error":"Incorrect username or password."} |
| 500 | {"error":"Something went wrong. Please try again later."} |

## 2. Sign up

Authenticate the user with the system and obtain the auth\_token

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/signup/ |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| HEAD  POST  POST  POST  POST | api\_key  username  password  contact no  email address | string  string  string  string  string |

**api\_key**

Api\_key must be sent with all client requests. The api\_key helps the server to validate the request source.

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | {  "auth\_key": <auth\_key>  }  auth\_key(**string**) - all further API calls must have this key in header |
| 403 | {"error":"API key is missing."} |
| 400 | {"error":"Please provide username."} |
| 400 | {"error":"Please provide password."} |
| 400 | {"error":"please provide contact no."} |
| 400 | {"error":"please provide email address."} |
| 401 | {"error":"Invalid API key."} |
| 401 | {"error":"Incorrect username or password."} |
| 500 | {"error":"Something went wrong. Please try again later."} |

## 3. Contact us

Authenticate the user with the system and obtain the auth\_token

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/contact/ |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| HEAD  POST  POST  POST | api\_key  username  contact no  email address | string  string  string  string |

**api\_key**

Api\_key must be sent with all client requests. The api\_key helps the server to validate the request source

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | {  "auth\_key": <auth\_key>  }  auth\_key(**string**) - all further API calls must have this key in header |
| 403 | {"error":"API key is missing."} |
| 400 | {"error":"Please provide username."} |
| 400 | {"error":"Please provide password."} |
| 400 | {"error":"please provide contact no."} |
| 400 | {"error":"please provide email address."} |
| 401 | {"error":"Invalid API key."} |
| 401 | {"error":"Incorrect username or password."} |
| 500 | {"error":"Something went wrong. Please try again later."} |

## 4. get updates

Get the new updates

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/updates/ |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| HEAD  POST | auth\_key  version | string  number |

**auth\_key**

The auth\_key that was given in response to /api/login

**version**

The current version of internal recipe database. Each time when updates are pulled from the server through the web service, the internal database version is incremented.

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | **Response will be an object containing the list of recipes (array)**  **as well as the updated recipe database. Each item in the recipe array has the following structure.**  {  "recipe\_id": 10,  "title": "Green Chilly Salad",  "category": 1,  "ingredients": {  "Green Chilly": "1 kg",  "Salt": "0.5 tbsp"  },  "steps": [  "First clean and cut the chillies",  "Now you can eat."  ],  "remarks": "serves 2 people"  }  **An example response is:-**  {  "recipes": [  {  "recipe\_id": 10,  "title": "Green Leaf Curry",  "category": 1,  "ingredients": {  "Green leaf": "1 kg",  "Salt": "0.5 tbsp"  },  "steps": [  "First clean and cut the leaves",  "Now you can eat."  ],  "remarks": "serves 2 people"  }  ],  "version": "4"  } |
| 400 | {"error":"Please specify database version."} |
| 400 | {"error":"Invalid database version."} |
| 401 | {"error":"Invalid API key."} |
| 500 | {"error":"Something went wrong. Please try again later."} |

## 3. deletions

Get the recipes that were deleted from the web interface, so that they can be deleted from the internal database also.

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/deletions/ |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| HEAD  POST | auth\_key  version | string  number |

**version**

The current version of internal database. Each time when updates are pulled from the API, the internal database version increases.

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | **An array containing the ID’s of recipes to delete is given**  Example response:-  {"deletions":[10,11,40], "version":"5"} |
| 400 | {"error":"Please specify database version."} |
| 400 | {"error":"Invalid database version."} |
| 401 | {"error":"InvalidAuth key."} |
| 500 | {"error":"Something went wrong. Please try again later."} |

## 4. get recipe image

Get more information on a particular recipe

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | api/image/<recipe\_id>/ |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| HEAD  URL\_PARAM | auth\_key  <recipe\_id> | string  number |

**recipe\_id**

Id of the recipe you want the image of.

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | **An array containing the ID’s of recipes to delete is given**  Example response:-  {"image":"http:\/\/example.com\/recipe-5-image.jpg"} |
| 400 | {"error":"Please provide recipe\_id."} |
| 400 | {"error":"Invalidrecipe\_id."} |
| 401 | {"error":"InvalidAuth key."} |
| 500 | {"error":"Something went wrong. Please try again later."} |

5: Order operations

* [Creates a new order](http://docs.opendining.net/api/#!/orders.js/createOrder_post_0)

Request:

|  |  |
| --- | --- |
| METHOD | URL |
| Post | Api/v1/order |

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Parameter | values | Description |
| Head | Api-key | string |  |
| post | Res-id | string | **The ID of the restaurant to which the order will be submitted** |
| Post | email | string | If the user is logged in, send their email address when creating the order. |
| post | item | string | Items to add to the order to start with. |

Response:

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | {  "auth\_key": <auth\_key>  }  auth\_key(**string**) - all further API calls must have this key in header |
| 403 | {"error":"API key is missing."} |
| 400 | {"error":"One or more required parameters were missing"} |
| 401 | {"error":"Invalid API key."} |
| 500 | {"error":"Something went wrong. Please try again later."} |

SEARCH OPERATION

* [Perform a new search across restaurants and menu items](http://docs.opendining.net/api/#!/search.js/searchAllTypes_get_0). This returns both restaurants and menu items in a single call. Use the type property of each result to determine whether it's a restaurant or item

REQUEST:

RESPONSE:

Get menu

Request:

|  |  |
| --- | --- |
| Method | url |
| Post | Api/v1/get menu |

|  |  |  |
| --- | --- | --- |
| Type | parameter | values |
| Head | Api-key | string |
| Post | Rest-name | string |
| Post | cuisine | string |
| Post | location | string |
| Post | price | value |

Response:

|  |  |
| --- | --- |
| STATUS | RESPONSE |

Delivery time slot data

Request:

|  |  |
| --- | --- |
| method | url |
| Get | Get/timeslot |

|  |  |  |
| --- | --- | --- |
| TYPE | PARAMETER | VALUE |
| param | Order-id | value |
| date | date | date |

response:

|  |  |
| --- | --- |
| status | response |
| 500 | {"error":"Something went wrong. Please try again later."} |

Glossary

## Conventions

* **Client** - Client application.
* **Status** - HTTP status code of response.
* All the possible responses are listed under ‘Responses’ for each method. Only one of them is issued per request server.
* All response are in JSON format.
* All request parameters are mandatory unless explicitly marked as [optional]
* The type of values accepted for a *request* parameter are shown the the values column like this [**10**|<any number>] .The | symbol means *OR*. If the parameter is [optional], the default value is shown in blue bold text, as **10** is written in [**10**|<any number>].

## Status Codes

All status codes are standard HTTP status codes. The below ones are used in this API.

2XX -Success of some kind

4XX -Error occurred in client’s part

5XX -Error occurred in server’s part

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 200 | OK |
| 201 | Created |
| 202 | Accepted (Request accepted, and queued for execution) |
| 400 | Bad request |
| 401 | Authentication failure |
| 403 | Forbidden |
| 404 | Resource not found |
| 405 | Method Not Allowed |
| 409 | Conflict |
| 412 | Precondition Failed |
| 413 | Request Entity Too Large |
| 500 | Internal Server Error |
| 501 | Not Implemented |
| 503 | Service Unavailable |