



# D1 - RESTaurant API

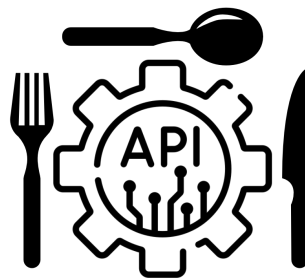
---

D-API-100

## RESTaurant API

---

Not just a pun on a good analogy



1.0.1

# RESTaurant API

delivery method: Github  
language: Javascript, SQL



- The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.

RESTaurant API provides **digital menu** and **online ordering solutions**, intuitive and ergonomic, to fit the customers demands.  
All of their services are optimized to provide gain of time.

As a member of their team, your job is to develop a secure, stable and efficient **back-end**, compliant with the specifications you are given.  
For security reasons, a **basic authentication** is required to use this API, with at least two different roles: **admin** and **client**.

A **mySQL** database is also required to store the data.



Feel free to organize and create the different tables you need in your mySQL database.  
Still, we highly recommend you to consider the following tables:  
item, category, formula, category-formula.



You are expected to deliver:

- a **backend server** in Express with a connection to a mySQL database;
- a clear and concise **README** that explains how to launch your API and its features;
- a **postman** collection exported in JSON format, that tests and covers all the routes of your API.



Do not limit yourself to the requirements.  
Go further and add extra features: OAuth2, full API documentation, new routes, pagination,...

## ENDPOINTS

To meet the requirements of final customers, you are given the **endpoints** of your API:

### /ITEMS

The **/items** route describes the items present in the restaurant, whether dishes, drinks, or any other product on the menu. For example, *pastis* and *bouillabaisse* are two valid items (unless the restaurant is in the Nord-Pas-de-Calais).

route	operation	description	credentials
/items	READ	get all items	client, admin
/items?parameters	READ	get all items matching filters	client, admin
/items	CREATE	add a new item	admin
/items/{id}	READ	get an item	client, admin
/items/{id}	UPDATE	change an item properties	admin
/items/{id}	DELETE	delete an item	admin

### /CATEGORIES

The **/categories** route is relative to the different categories of food. For example, the French triptych (starter, main course and dessert) are three of them. Terms like cocktails or wines are also categories.

route	operation	description	credentials
/categories	READ	get all item categories	client, admin
/categories	CREATE	add a new item category	admin
/categories/{id}	READ	get an item category	client, admin
/categories/{id}	UPDATE	change an item category properties	admin
/categories/{id}	DELETE	delete an item category	admin

### /FORMULAS

The **/formulas** route is relative to the different formulas of the menu.

route	operation	description	credentials
/formulas	READ	get all formulas	client, admin
/formulas?parameters	READ	get all formulas matching filters	client, admin
/formulas	CREATE	add a new formula	admin
/formulas/{id}	READ	get a formula	client, admin
/formulas/{id}	UPDATE	change a formula	admin
/formulas/{id}	DELETE	delete a formula	admin



You are given access to two documents:  
annotated menus and **OpenAPI specifications**.  
Take time to read, understand and follow them.

