

**Client  
(Customer)**



**API Request/  
Response**



**Server  
(Kitchen)**

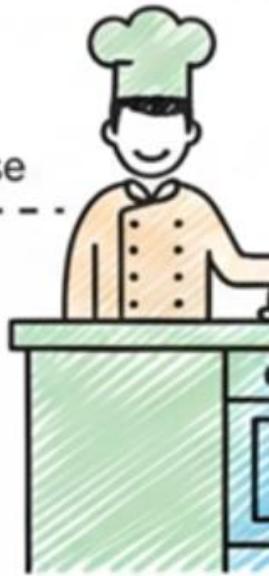


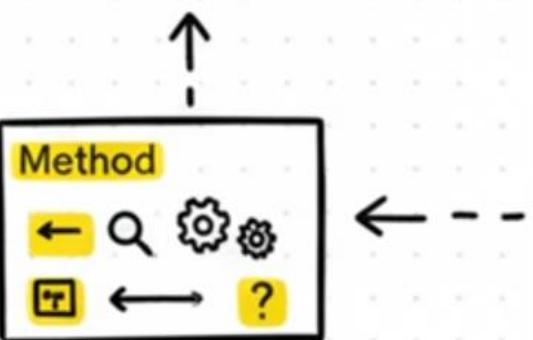
request

# REST (Representational State Transfer)

An architectural style for designing APIs. Data is represented by URLs and accessed via HTTP methods.

response





Component	What it is
Method	The action you want to take (e.g., GET, POST)
URL	The specific item from the menu (e.g., /users/5)
Headers	Special instructions (e.g., Content-Type)
Body	The specific details of your order (e.g., user info)





## Component

## What it is

### Status Code

A quick update on your order (e.g., Success!)



### Headers

Extra info about the delivery (e.g., Content-Type)

### Body



The actual 'food' you ordered (e.g., user data)

Server





Method	Action
GET	Read or retrieve data.
POST	Create new data.
PUT / PATCH	Update existing data.
DELETE	Remove data.



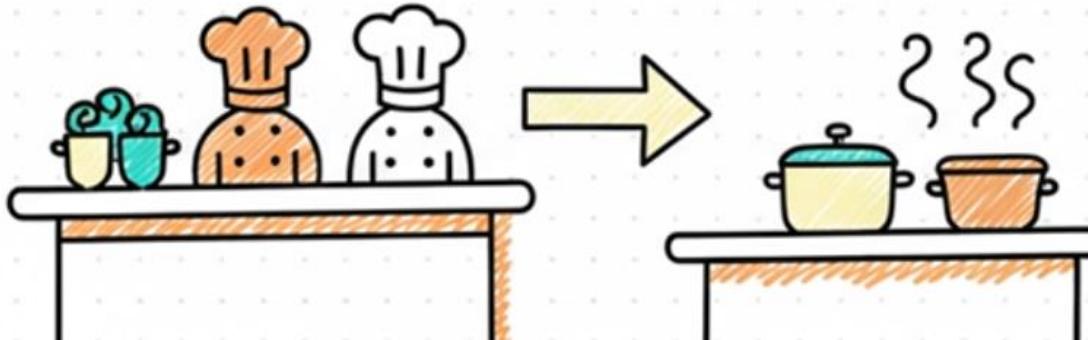
# Status Codes

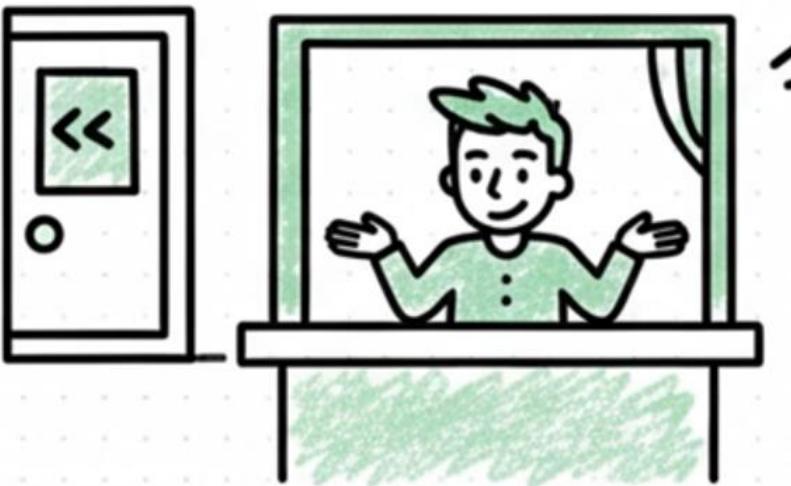
- 2xx: Success (50%)
- 4xx: Client Error (30%)
- 5xx: Server Error (10%)
- 3xx: Redirect (5%)
- 1xx: Informational (5%)

Codes are grouped into **categories** to quickly understand the result of a request.

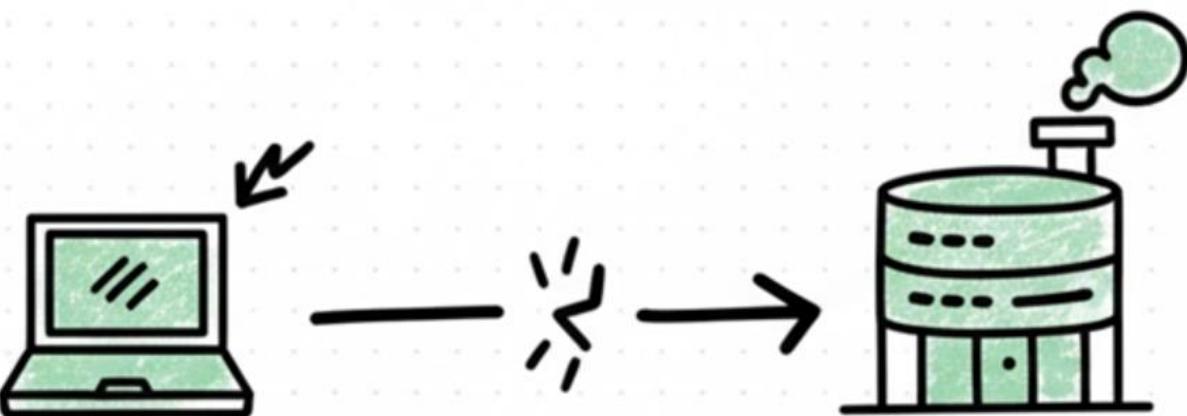


2XXX





# 4XXX





# 5XX





## Concept

## In a Nutshell

REST

The rules of communication.

Request

The client's order.

Response

The server's reply.

Methods

The actions (Create, Read, Update, Delete).

Status Codes

The outcome (Success, Error, etc.).

FastAPI



The framework that automates it all.

