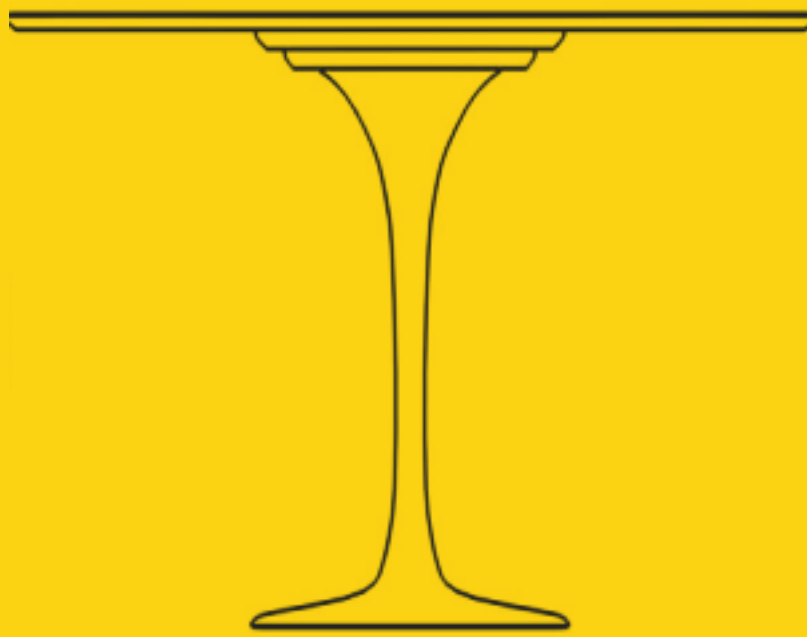
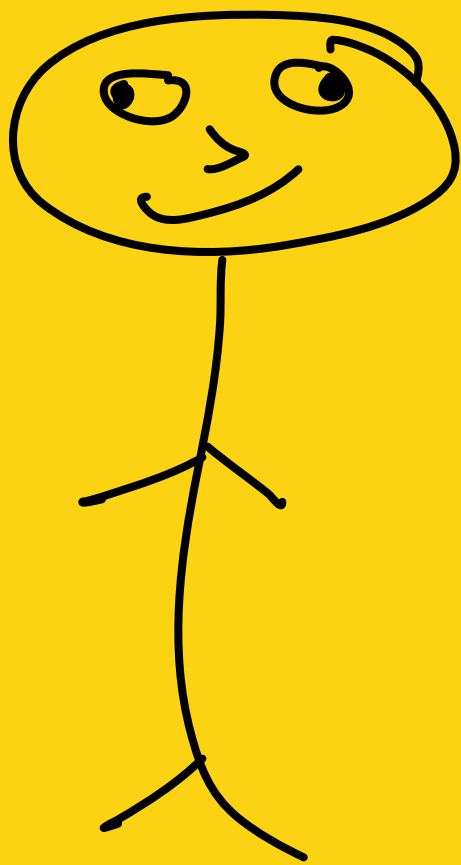


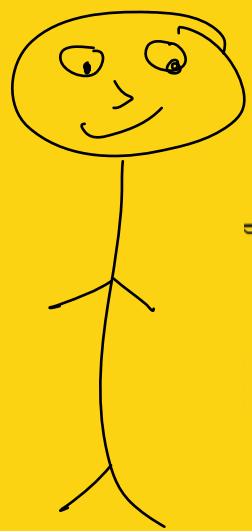
REST API



ANALOGY

You're in a restaurant, asking for a burger

I want a burger

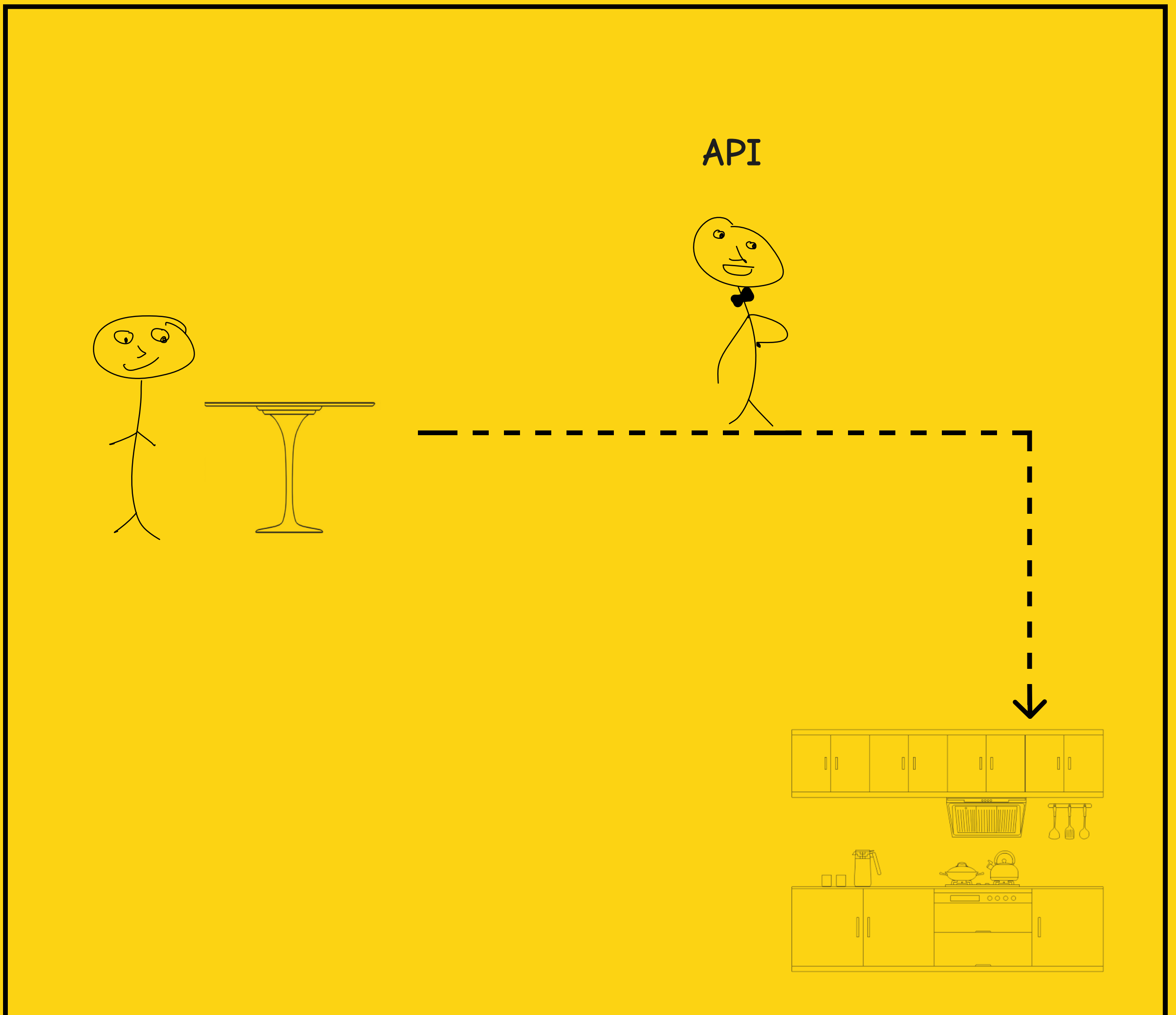


**You
(Client)**

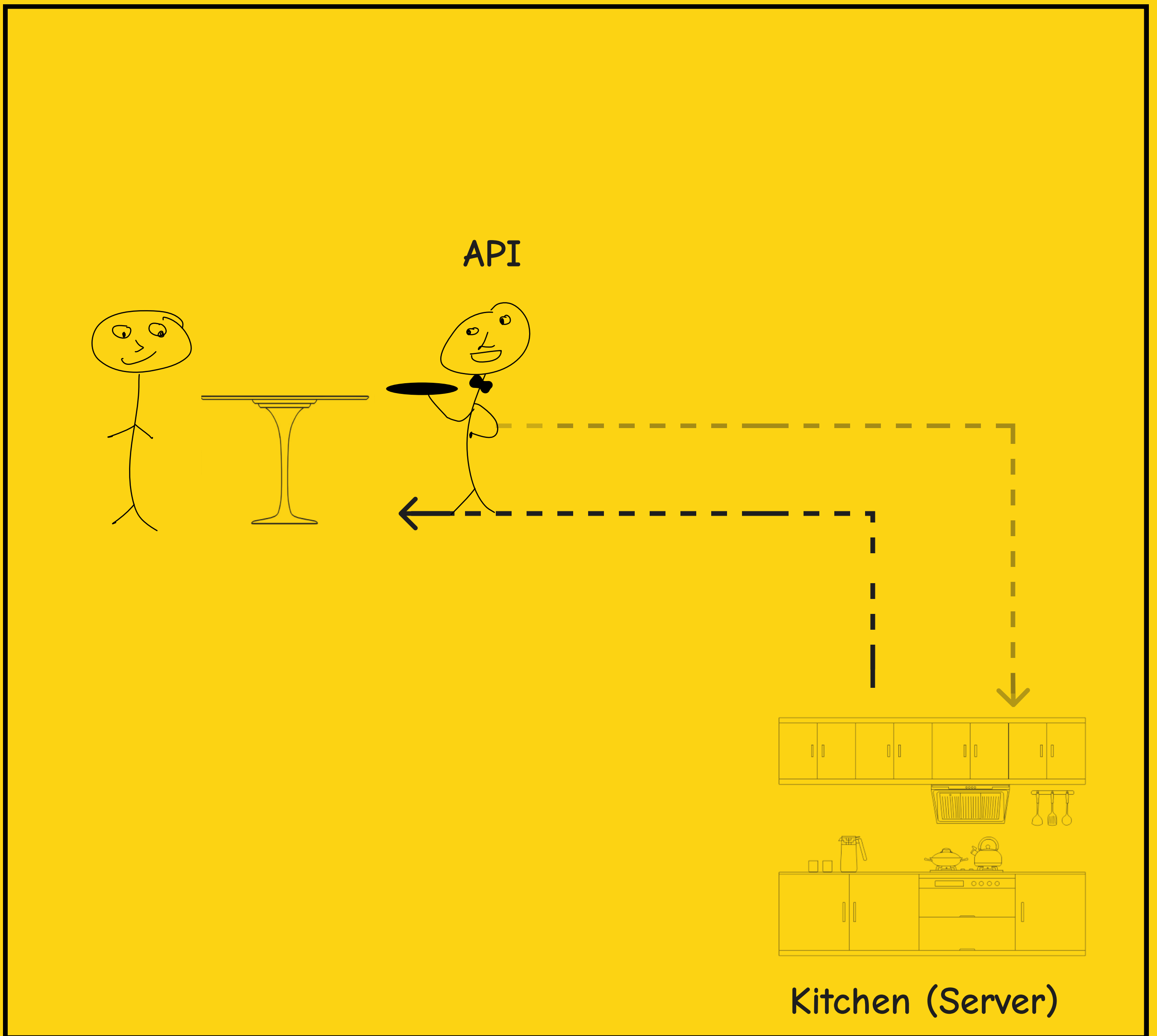
Welcome



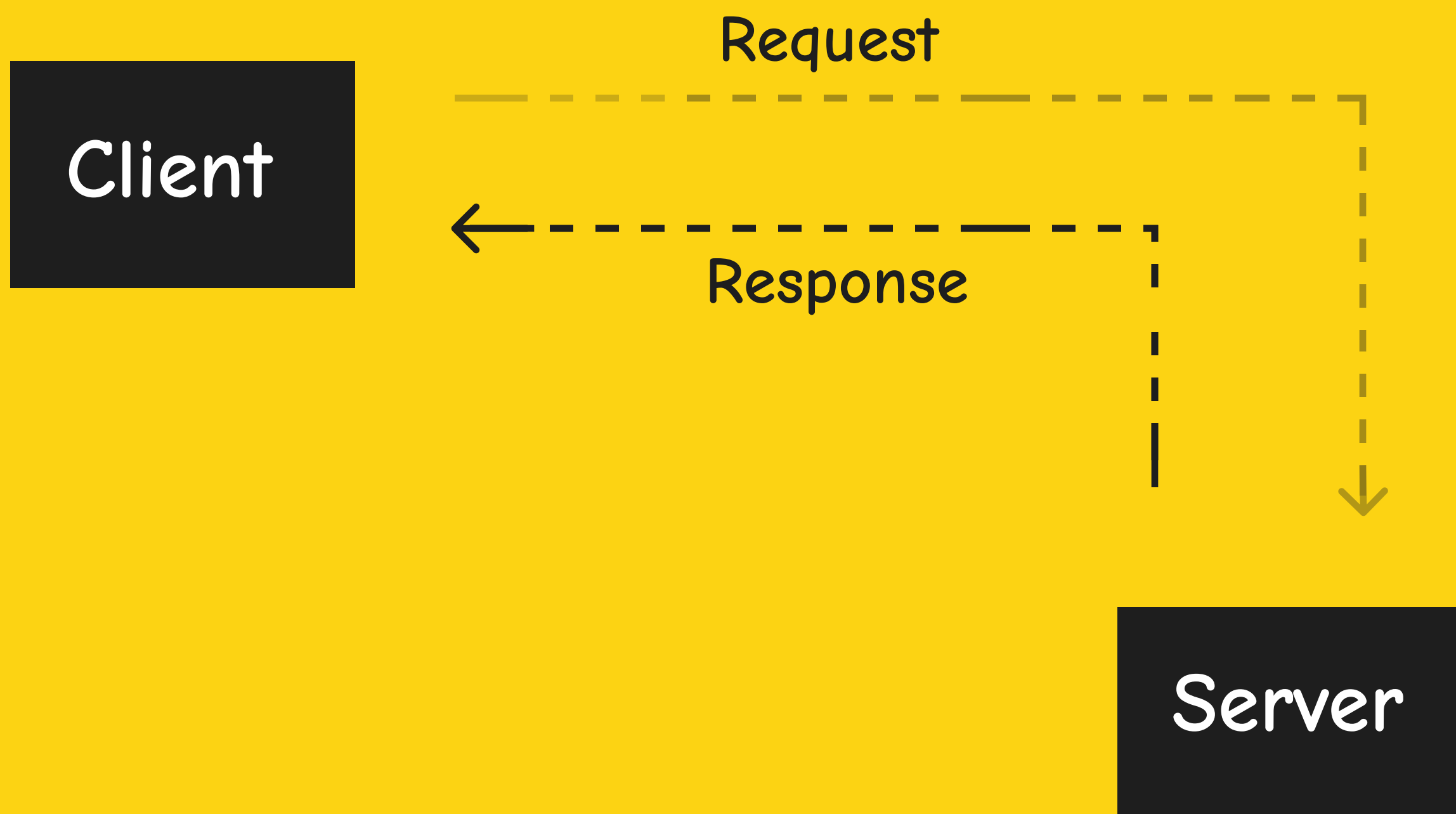
The **waiter** is like an **API** that takes your order and deliver the response



The **kitchen** act like a **server** that
prepare your order.

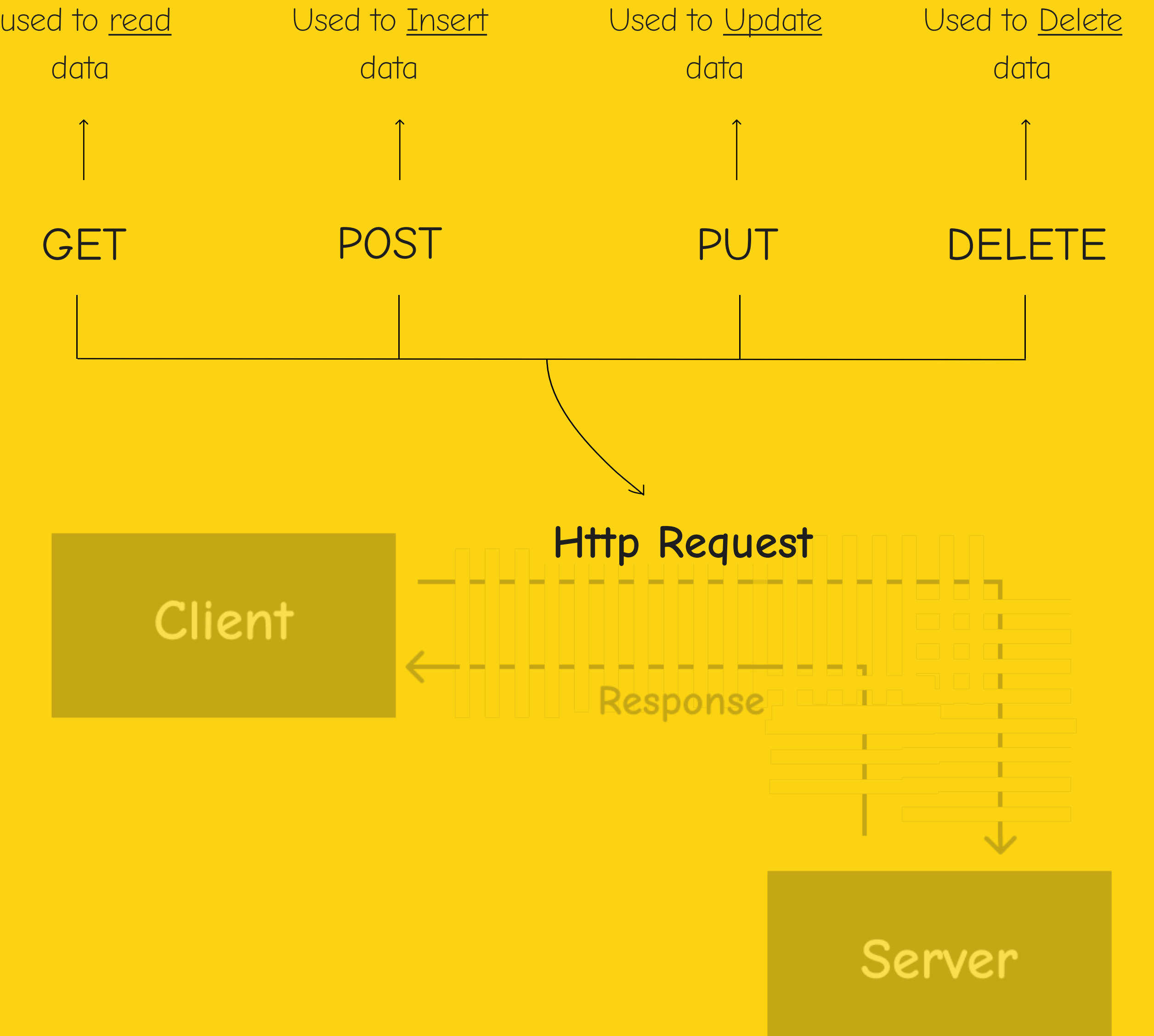


Client (Your app) communicate with the server using **HTTP** Protocol



HTTP METHODS

The client use 4 methods to communicate with the server

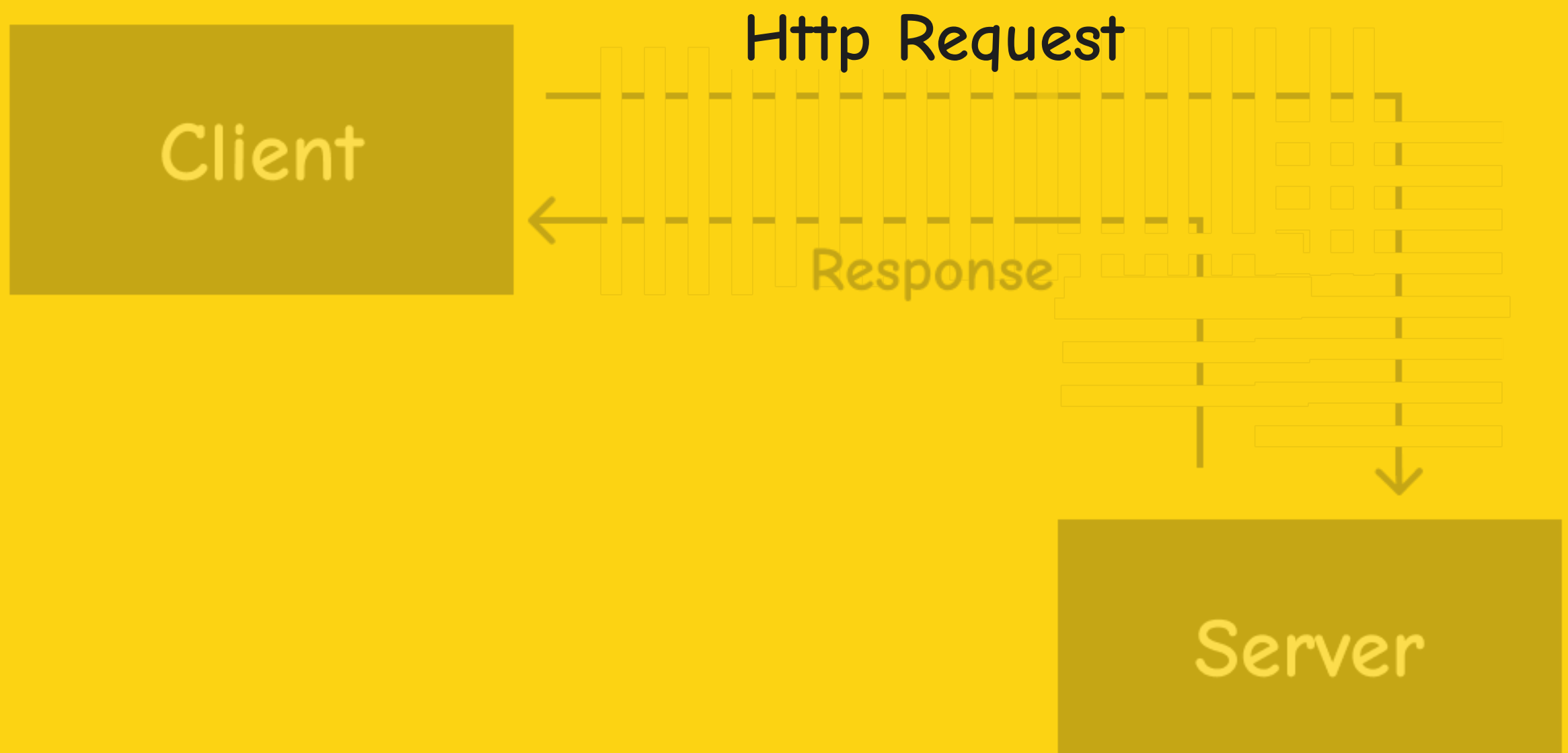


Example of Get Request

`http://restaurant.com/api/meals`



Called Resource



Best Practice for better naming

1) Do not use verbs

GET	/getAllCars	/cars
POST	/createNewCar	/cars
DELETE	/deleteAllRedCars	/cars

2) Use plural nouns

/cars instead of /car

/users instead of /user

3) Use sub-resources for relations

```
GET /cars/711/drivers/ #Returns a list of drivers for car 711
GET /cars/711/drivers/4 #Returns driver #4 for car 711
```


Best Practice for better naming

4) Filtering:

-> GET /cars?color=red Returns a list of red cars

5) Sorting:

Allow ascending and descending sorting over multiple fields.

-> GET /cars?sort=-manufacturer,+model

6) Version your API

-> /blog/api/v1