

DOCUMENTACIÓN

API

Grupo
Otters

Proyecto
Reserved

Asignatura
Servicios Multimedia Basados en Internet

Universidad de Alicante
Curso 2017-2018

Integrantes del grupo

Carlos Chico Martínez

Miguel Ángel Sánchez Hernández

Sergio Lázaro Frías

Alejandro Tomás Martínez Martínez

Daniel Belmonte Amorós

Índice

<i>Métodos de Usuarios</i>	2
<i>Métodos de Reservas</i>	7
<i>Métodos de Control de Usuarios</i>	11
<i>Métodos de Denuncias</i>	13
<i>Métodos de Restaurantes</i>	16
<i>Métodos de Comentarios</i>	24
<i>Métodos de Imágenes</i>	28
<i>Métodos de Visita</i>	32
<i>Métodos de Pedidos</i>	36
<i>Métodos de Productos</i>	40

Usuarios

Método HTTP	GET
URI	/users
Parámetros de entrada	-
Descripción	Muestra todos los usuarios de la aplicación
Parámetros	

The screenshot shows the Postman application interface. At the top, there's a header bar with 'reserved.ovh:3000/us' and environment dropdowns. Below it, a search bar has 'GET' selected and 'reserved.ovh:3000/users/' entered. To the right are 'Send', 'Save', and 'Code' buttons. Underneath, tabs for 'Authorization', 'Headers', 'Body', 'Pre-request Script', and 'Tests' are visible, with 'Authorization' being the active tab. The 'Body' tab is selected, showing 'Pretty' (selected), 'Raw', 'Preview', and 'JSON' options. The JSON response is displayed as:

```
1 [ { 2   "nombre": "A13X", 3   "nick": "A13X", 4   "email": "alexdemorider@gmail.com" 5 } 6 ] 7
```

At the bottom, status information 'Status: 200 OK' and 'Time: 169 ms' is shown.

Método HTTP	GET
URI	/users/id
Parámetros de entrada	Id del usuario

Descripción	Muestra todos un usuario específico pasándole el id.
Parámetros	

The screenshot shows the Postman interface with a GET request to `reserved.ovh:3000/users/1`. The response status is 200 OK with a time of 178 ms. The response body is a JSON object:

```

1 [ 
2   {
3     "nombre": "Al3x",
4     "nick": "Al3x",
5     "email": "alexdemorider@gmail.com"
6   }
7 ]
  
```

Método HTTP	POST
URI	/users
Parámetros de entrada	-
Descripción	Crea un nuevo usuario
Parámetros	Nick, Password, Email

The screenshot shows the Postman interface with a POST request to `reserved.ovh:3000/users/`. The Body tab is selected and set to `form-data`. The form fields are:

Key	Value
email	jose@gmail.com
password	poaskdpoak
nick	jose

The screenshot shows a Postman request to `reserved.ovh:3000/users/`. The response is a JSON array with two elements:

```

1 [ 
2   {
3     "nombre": "A13x",
4     "nick": "A13x",
5     "email": "alexdemorider@gmail.com"
6   },
7   {
8     "nombre": "",
9     "nick": "jose",
10    "email": "jose@gmail.com"
11  }
12 ]

```

Método HTTP	PUT
URI	/users/id
Parámetros de entrada	El id del usuario que se va a modificar
Descripción	Modifica los datos de un usuario, uno o varios.
Parámetros	Un usuario solo puede modificar su email o su contraseña. El nick es único.

Método HTTP	DELETE
URI	/users/id
Parámetros de entrada	El id del usuario que se va a eliminar
Descripción	Elimina un usuario de la aplicación
Parámetros	Campo que quieras modificar

The screenshot shows the Postman interface with a successful DELETE request. The URL is `reserved.ovh:3000/users/5`. The response body is a single line: `"Usuario eliminado"`.

The screenshot shows the Postman interface with a successful GET request. The URL is `reserved.ovh:3000/users/`. The response body is an empty JSON array: `[]`.

Reservas

Método HTTP	GET
URI	/users/:id/reservations
Parámetros de entrada	id del usuario
Descripción	Muestra todas las reservas de un usuario
Parámetros	

The screenshot shows the Postman application interface. At the top, there's a header bar with 'localhost:3000/users/' and various icons. Below it is a search bar with 'localhost:3000/users/1/reservations'. To the right are buttons for 'Send' (highlighted in blue), 'Save', and 'Code'. Underneath the search bar, there are tabs for 'Authorization', 'Headers (1)', 'Body', 'Pre-request Script', and 'Tests'. The 'Body' tab is selected and has a dropdown 'Type' set to 'No Auth'. The 'JSON' tab is also selected. The response section shows 'Status: 200 OK' and 'Time: 37 ms'. The JSON response body is displayed as:

```

1 [ [
2   {
3     "dia": "1999-04-30T22:00:00.000Z",
4     "hora": "00:00:05",
5     "nombre": "Nou"
6   }
7 ]
  
```

Método HTTP	POST
URI	/users/:id/reservations
Parámetros de entrada	id del usuario que quiere crear la reserva

Descripción	Crea una reserva, en el momento que la creamos miramos en la tabla “AforoLibre” si quedan sitios disponibles en el restaurante. Si los comensales son mayores que el aforo libre no se puede hacer una reserva.
Parámetros	Dia,Turno,Hora,Comensales

The screenshot shows two requests in the Postman interface:

- Request 1 (GET):** URL: http://reserved.ovh:3000/users/1/reservations. Response status: 200 OK. Body (Pretty JSON):

```

1 [
2   {
3     "dia": "2017-05-04T22:00:00.000Z",
4     "hora": "21:00:00",
5     "nombre": "CasaPepe"
6   }
7 ]
  
```

- Request 2 (POST):** URL: http://reserved.ovh:3000/users/1/reservations. Response status: 200 OK. Body (Pretty JSON):

```

1
  
```

Método HTTP	DELETE
URI	/users/:id/reservations/:idreserva

Parámetros de entrada	id del usuario que quiere crear la reserva e id de la reserva que se quiere borrar
Descripción	Borra una reserva
Parámetros	

The screenshot shows the Postman application interface with two requests:

- GET Request:**
 - URL: `http://localhost:3000/users/1/reservations/`
 - Method: GET
 - Headers: Authorization (selected), Headers (1)
 - Body: JSON (Pretty, Raw, Preview)
 - Test Results: Status: 200 OK, Time: 22 ms
- DELETE Request:**
 - URL: `http://localhost:3000/users/1/reservations/1`
 - Method: DELETE
 - Headers: Authorization (selected), Headers (1)
 - Body: form-data (x-www-form-urlencoded selected)

Key	Value
dia	2017-5-5
turno	comida
hora	21:00
comensales	4
restaurante	14
 - Test Results: Status: 200 OK, Time: 122 ms

Método HTTP	PUT
URI	/users/:id/reservations/:idreserva
Parámetros de entrada	id del usuario que quiere modificar la reserva e id de la reserva que se quiere modificar
Descripción	Modifica una reserva
Parámetros	El campo o campos que se deseen modificar

The screenshot shows the Postman application interface with two requests defined:

- PUT Request:**
 - URL: http://localhost:3000/users/1/reservation/1
 - Body (x-www-form-urlencoded):

Key	Value
IdAforo	1
dia	2017-5-5
hora	21:25
turno	Comida
idrestaurante	1
comensales	17
- GET Request:**
 - URL: http://localhost:3000/users/1/reservations/
 - Type: No Auth

The results section shows the response for the GET request:

```

1 [
2   {
3     "dia": "2017-02-04T23:00:00.000Z",
4     "hora": "21:25:00",
5     "nombre": "Nou"
6   }
7 ]
  
```

Status: 200 OK Time: 65 ms

Control Usuarios

Método HTTP	GET
URI	/users/:id/usercontrol
Parámetros de entrada	id del usuario
Descripción	Muestra la última conexión de un usuario
Parámetros	

The screenshot shows the Postman application interface. At the top, there are tabs for 'Authorization', 'Headers', 'Body', 'Pre-request Script', and 'Tests'. The 'Body' tab is selected. Below it, there are buttons for 'Pretty', 'Raw', 'Preview', and 'JSON'. The JSON preview shows the following data:

```

1 [ [
2   {
3     "fecha": "1999-05-01T22:00:00.000Z"
4   }
5 ]

```

Método HTTP	POST
URI	/users/:id/usercontrol
Parámetros de entrada	id del usuario
Descripción	Con este método se guarda la última

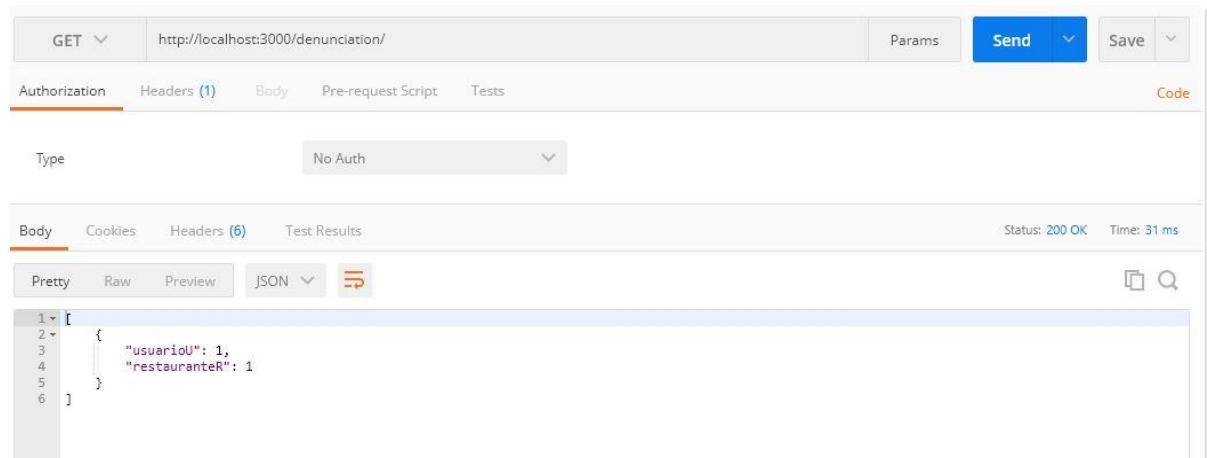
	conexión de un usuario, si es la primera vez que se conecta un usuario se le asigna un <i>IdControlUsuarios</i> y si ya se ha conectado más de una vez se modifica la columna “Fecha”
Parámetros	Fecha

The screenshot shows the Postman application interface. A POST request is being prepared to the URL `reserved.ovh:3000/users/1/usercontrol`. The 'Body' tab is active, indicating the use of x-www-form-urlencoded data. A single key-value pair is present: 'fecha' with the value '2017-05-02'.

The screenshot shows the Postman application interface after sending the request. The status bar indicates a successful response (Status: 200 OK) and a response time of 198 ms. The response body is displayed as a JSON object containing the string "Login Hecho".

Denuncia Usuarios

Método HTTP	GET
URI	/denunciation
Parámetros de entrada	-
Descripción	Muestra todos las denuncias en la base de datos de la aplicación
Parámetros	



The screenshot shows the Postman interface with a GET request to `http://localhost:3000/denunciation/`. The response is a 200 OK status with a JSON body:

```
1 [  
2 {  
3   "usuarioU": 1,  
4   "restauranteR": 1  
5 }  
6 ]
```

Método HTTP	GET
URI	/restaurants/id/denunciation
Parámetros de entrada	id del restaurante
Descripción	Muestra todos las denuncias que ha hecho un restaurante

Parámetros

HTTP Request: GET /restaurants/1/denunciation/

Response Body:

```

1 [ [
2   {
3     "usuarioU": 1,
4     "restauranteR": 1
5   }
6 ]

```

Método HTTP	POST
URI	/restaurants/:id/denunciation/:idusuario
Parámetros de entrada	id de restaurante e id de usuario a quien quieres hacerle la denuncia
Descripción	Muestra todos las denuncias que ha hecho un restaurante
Parámetros	

HTTP Request: POST /restaurants/1/denunciation/2

Response Body:

```

1 "Denuncia Hecha"

```

The screenshot shows the Postman application interface. At the top, there are several tabs with URLs: 'reserved.ovh:3000/re...', 'http://localhost:3000/...', 'http://localhost:3000/denur...', 'http://localhost:3000/...', '+', '***', and 'No Environment'. Below the tabs, the method is set to 'GET' and the URL is 'http://localhost:3000/restaurants/1/denunciation/'. There are buttons for 'Send' and 'Save'. Under the URL, there are tabs for 'Authorization', 'Headers (1)', 'Body', 'Pre-request Script', and 'Tests'. The 'Body' tab is selected and has a dropdown 'Type' set to 'No Auth'. The response section shows a status of 'Status: 200 OK' and 'Time: 26 ms'. The 'Body' tab contains tabs for 'Pretty', 'Raw', 'Preview', and 'JSON'. The JSON response is displayed as:

```
1 [  
2   {  
3     "usuarioU": 1,  
4     "restauranteR": 1  
5   },  
6   {  
7     "usuarioU": 2,  
8     "restauranteR": 1  
9   }  
10 ]
```

Restaurantes

Método HTTP	GET
URI	/restaurants
Parámetros de entrada	-
Descripción	Muestra todos los restaurantes de la aplicación
Parámetros	

The screenshot shows the Postman application interface. At the top, the URL is set to `localhost:3000/restaurants`. Below the URL bar, there are tabs for `Authorization`, `Headers (1)`, `Body`, `Pre-request Script`, and `Tests`. The `Body` tab is selected. The response status is `200 OK` and the time taken is `42 ms`. The response body is displayed in a JSON editor with the following content:

```

1  [
2   {
3     "nombre": "Nou",
4     "horario": "1-5 9-2",
5     "descripcion": "psoaajdpodjs",
6     "direccion": "calle prueba",
7     "telefono": "658413201",
8     "ciudad": "Alicante",
9     "tipoComida": "Italiano"
10    },
11    {
12      "nombre": "CasaPepe",
13      "horario": "Comida 13:00-17:00 Cena 21:00-00:00",
14      "descripcion": "La mejor comida del mundo",
15      "direccion": "C/Salamanca Número 15",
16      "telefono": "963213123",
17      "ciudad": "Alicante",
18      "tipoComida": "Marfileña"
19    },
20    {
21      "nombre": "Cacerolita",
22      "horario": "Comida 14:00-17:00 Cena 20:00-23:00".

```

Método HTTP	GET
URI	/restaurants/:id
Parámetros de entrada	Id del restaurante que se quiere mostrar

Descripción	Muestra un restaurante concreto
Parámetros	

The screenshot shows the Postman application interface. At the top, there's a header bar with the URL "localhost:3000/restau" and some environment dropdowns. Below that is a toolbar with "GET", "localhost:3000/restaurants/14", "Params", "Send", and "Save". Underneath is a navigation bar with "Authorization", "Headers (1)", "Body", "Pre-request Script", and "Tests". The main area has tabs for "Body", "Cookies", "Headers (6)", and "Test Results". The "Body" tab is selected and shows a JSON response:

```

1 [ 
2   {
3     "nombre": "CasaPepe",
4     "horario": "Comida 13:00-17:00 Cena 21:00-00:00",
5     "descripcion": "La mejor comida del mundo",
6     "direccion": "C/Salamanca Número 15",
7     "telefono": "963213123",
8     "ciudad": "Alicante",
9     "tipoComida": "Marfileña"
10    }
11 ]
  
```

The status bar at the bottom right indicates "Status: 200 OK" and "Time: 26 ms".

Método HTTP	GET
URI	/restaurants/name/:nombre
Parámetros de entrada	Nombre del restaurante que se quiere mostrar
Descripción	Muestra un restaurante concreto buscandole el nombre
Parámetros	

The screenshot shows the Postman interface with the following details:

- URL:** localhost:3000/restau
- Method:** GET
- Response Status:** 200 OK
- Response Time:** 66 ms
- JSON Response:**

```

1 {
2   "nombre": "Nou",
3   "horario": "1-5 9-2",
4   "descripcion": "pscoajdpodjs",
5   "direccion": "calle prueba",
6   "telefono": "6584413201",
7   "ciudad": "Alicante",
8   "tipoComida": "Italiano"
9 }
10

```

Método HTTP	GET
URI	restaurants//type/:tipo
Parámetros de entrada	Tipo de comida del restaurante que se quiere mostrar
Descripción	Muestra un restaurante concreto buscandole el tipo de comida.
Parámetros	

localhost:3000/restaurante

GET localhost:3000/restaurants/type/Italiano

No Environment

Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Code

Type No Auth

Body Cookies Headers (6) Test Results Status: 200 OK Time: 85 ms

Pretty Raw Preview JSON

```

1 [ {
2   "nombre": "Nou",
3   "horario": "1-5 9-2",
4   "descripcion": "psoadjpodjs",
5   "direccion": "calle prueba",
6   "telefono": "658413201",
7   "ciudad": "Alicante",
8   "tipoComida": "Italiano"
9 }
10 ]
11

```

Método HTTP	GET
URI	restaurants/:id/capacity
Parámetros de entrada	Id del restaurante que queremos ver el aforo
Descripción	Muestra el aforo de un restaurante en un turno
Parámetros	

reserved.ovh:3000/re: http://localhost:3000/ http://localhost:3000/ http://localhost:3000/ http://localhost:3000/ + *** No Environment

GET http://localhost:3000/restaurants/1/capacity/ Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Code

Type No Auth

Body Cookies Headers (6) Test Results Status: 200 OK Time: 41 ms

Pretty Raw Preview JSON

```

1 {
2   "dia": "2017-05-04T22:00:00.000Z",
3   "turno": "Comida",
4   "aforo": 250
5 },
6

```

Método HTTP	POST
URI	/restaurants
Parámetros de entrada	-
Descripción	Crear un restaurante
Parámetros	Nombre, Password, email, horario, descripción, dirección, teléfono, ciudad, aforo, TipoComida, Coordenadas

The screenshot shows the Postman application interface. At the top, there are fields for 'reserved.ovh:3000/us' and 'http://localhost:3000/' with a dropdown menu between them, and a 'No Environment' dropdown. Below the address bar, there are buttons for 'Send', 'Save', and 'Code'. The main area shows a 'POST' method selected, and the URL 'http://localhost:3000/restaurants'. Under the 'Body' tab, which is selected, there are several parameters listed in a table:

Key	Value
nombre	prueba
password	pksoapkpod
email	oaps@gmail.com
horario	5-3
descripcion	oasskpdkopasdpdpo
direccion	aokdopak
telefono	695698745
ciudad	siodjiaoj
aforo	150
tipoComida	Italiano
coordenadas	asdadad
New key	Value

The screenshot shows the Postman interface with a GET request to `http://localhost:3000/restaurants`. The response status is 200 OK and the time taken is 182 ms. The response body is a JSON array containing two objects:

```

1 [ 2 {
  "nombre": "CasaPepe",
  "horario": "Comida 13:00-17:00 Cena 21:00-00:00",
  "descripcion": "La mejor comida del mundo",
  "direccion": "C/Salamanca Número 15",
  "telefono": "963213123",
  "ciudad": "Alicante",
  "tipoComida": "Marfileña"
},
{
  "nombre": "Cacerolita",
  "horario": "Comida 14:00-17:00 Cena 20:00-23:00",
  "descripcion": "La mejor comida del mundo",
  "direccion": "C/La Mancha Número 23",
  "telefono": "965213423",
  "ciudad": "Salamanca",
  "tipoComida": "Americana"
]
  
```

Método HTTP	PUT
URI	/restaurants/id
Parámetros de entrada	Id del restaurante que se quiere modificar
Descripción	Modifica los datos de un restaurante
Parámetros	Campos que se quieran modificar (uno o varios)

The screenshot shows the Postman interface with a PUT request to `http://localhost:3000/restaurants/1`. The Body tab is selected and set to `x-www-form-urlencoded`. The request contains the following key-value pairs:

Key	Value
horario	5-3
New key	Value

The screenshot shows the Postman application interface. At the top, there are two tabs: 'reserved.ovh:3000/restaurants/' and 'http://localhost:3000/'. The status bar indicates 'No Environment'. Below the tabs, the method is set to 'GET' and the URL is 'reserved.ovh:3000/restaurants/'. There are buttons for 'Params', 'Send', 'Save', and 'Code'. Under the 'Authorization' tab, it says 'Type: No Auth'. The 'Body' tab is selected, showing a JSON response with two objects. The JSON data is as follows:

```

1  [
2   {
3     "nombre": "CasaPepe",
4     "horario": "Comida 13:00-17:00 Cena 21:00-00:00",
5     "descripcion": "La mejor comida del mundo",
6     "direccion": "C/Salamanca Número 15",
7     "telefono": "963213123",
8     "ciudad": "Alicante",
9     "tipoComida": "Marfileña"
10    },
11   {
12     "nombre": "Cacerolita",
13     "horario": "Comida 14:00-17:00 Cena 20:00-23:00",
14     "descripcion": "La mejor comida del mundo",
15     "direccion": "C/La Mancha Número 23",
16     "telefono": "965213423",
17     "ciudad": "Salamanca",
18     "tipoComida": "Americana"
19   }
20 ]

```

The status bar at the bottom right shows 'Status: 200 OK' and 'Time: 130 ms'.

Método HTTP	DELETE
URI	/restaurants/:id
Parámetros de entrada	Id del restaurante que se quiere borrar
Descripción	Borra un restaurante de la aplicación
Parámetros	

The screenshot shows two separate API requests in the Postman application.

Request 1 (Top):

- Method: **DELETE**
- URL: **http://localhost:3000/restaurants/16**
- Authorization: No Auth
- Body: JSON response: "Restaurante eliminado"
- Status: 200 OK, Time: 347 ms

Request 2 (Bottom):

- Method: **GET**
- URL: **http://localhost:3000/restaurants/16**
- Authorization: No Auth
- Body: JSON response: []
- Status: 200 OK, Time: 133 ms

Comentarios

Método HTTP	GET
URI	/users/:id/comments
Parámetros de entrada	id del usuario
Descripción	Muestra los comentarios de un usuario
Parámetros	

The screenshot shows the Postman interface with the following details:

- Request URL: http://localhost:3000/users/1/comments
- Method: GET
- Response Status: 200 OK
- Response Time: 172 ms
- Response Body (Pretty):

```
1 [ ]  
2 {  
3   "contenido": "Me gusta mucho",  
4   "fecha": "2017-04-30T22:00:00.000Z",  
5   "nombre": "CasaPepe"  
6 }  
7 ]
```

Método HTTP	GET
URI	/restaurants/:id/comments
Parámetros de entrada	id del restaurante
Descripción	Muestra los comentarios de un restaurante
Parámetros	

Método HTTP	POST
-------------	------

URI	/comments
Parámetros de entrada	
Descripción	Un usuario crea un comentario para un restaurante
Parámetros	contenido, fecha, usuarioc, restaurante

The screenshot shows the Postman interface with a POST request to `http://localhost:3000/comment`. The 'Body' tab is selected, showing four parameters: `contenido` (value: La carne un poco seca), `fecha` (value: 2017-5-10), `usuarioc` (value: 1), and `restaurante` (value: 14). The 'Headers' tab shows one header: `x-www-form-urlencoded`.

Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> contenido	La carne un poco seca			
<input checked="" type="checkbox"/> fecha	2017-5-10			
<input checked="" type="checkbox"/> usuarioc	1			
<input checked="" type="checkbox"/> restaurante	14			
New key	Value	Description		

The screenshot shows the Postman interface with a GET request to `http://localhost:3000/users/1/comments`. The response status is 200 OK and the time is 149 ms. The response body is a JSON array of comments, with the third comment highlighted by a red oval.

```

1 [
2   {
3     "contenido": "Me gusta mucho",
4     "fecha": "2017-04-30T22:00:00.000Z",
5     "nombre": "CasaPepe"
6   },
7   {
8     "contenido": "dsafasdfads",
9     "fecha": "2017-04-30T22:00:00.000Z",
10    "nombre": "CasaPepe"
11  },
12  {
13    "contenido": "La carne un poco seca",
14    "fecha": "2017-05-09T22:00:00.000Z",
15    "nombre": "CasaPepe"
16  }
]

```

Método HTTP	DELETE
URI	/comments/:id
Parámetros de entrada	Id del comentario
Descripción	Eliminamos un comentario según el id pasado
Parámetros	

The screenshot shows the Postman interface with the following details:

- Request URL:** http://localhost:3000/comments/3
- Method:** DELETE
- Response Status:** 200 OK
- Response Time:** 149 ms
- Response Body (Pretty JSON):**

```

1 [ 
2   {
3     "contenido": "Me gusta mucho",
4     "fecha": "2017-04-30T22:00:00.000Z",
5     "nombre": "CasaPepe"
6   },
7   {
8     "contenido": "dsafasdfads",
9     "fecha": "2017-04-30T22:00:00.000Z",
10    "nombre": "CasaPepe"
11  },
12  {
13    "contenido": "La carne un poco seca",
14    "fecha": "2017-05-09T22:00:00.000Z",
15    "nombre": "CasaPepe"
16  },
17 ]

```

A red oval highlights the second comment object in the array.

The screenshot shows the Postman application interface. At the top, there are two tabs: 'http://localhost:3000/' (active) and 'http://localhost:3000/'. Below the tabs, there are buttons for '+', '***', 'No Environment', and settings. The main area shows a 'GET' request to 'http://localhost:3000/users/1/comments'. The 'Params' button is visible. To the right are 'Send' and 'Save' buttons. Below the request, tabs include 'Authorization', 'Headers (1)', 'Body', 'Pre-request Script', and 'Tests'. The 'Code' tab is also present. Under 'Body', the 'Pretty' tab is selected, showing a JSON array with three elements. Each element has an 'id' field (1, 2, 3) and three properties: 'contenido', 'fecha', and 'nombre'. The 'contenido' values are 'Me gusta mucho', 'La carne un poco seca', and 'La carne un poco seca'. The 'fecha' values are '2017-04-30T22:00:00.000Z', '2017-05-09T22:00:00.000Z', and '2017-05-09T22:00:00.000Z'. The 'nombre' value is 'CasaPepe' for all three elements. The 'Status: 200 OK' and 'Time: 91 ms' are displayed at the top right of the body panel.

```
[{"id": 1, "contenido": "Me gusta mucho", "fecha": "2017-04-30T22:00:00.000Z", "nombre": "CasaPepe"}, {"id": 2, "contenido": "La carne un poco seca", "fecha": "2017-05-09T22:00:00.000Z", "nombre": "CasaPepe"}, {"id": 3, "contenido": "La carne un poco seca", "fecha": "2017-05-09T22:00:00.000Z", "nombre": "CasaPepe"}]
```

Imágenes

Método HTTP	GET
URI	/restaurants/:id/images
Parámetros de entrada	Id del restaurante
Descripción	Método para ver las imágenes de un restaurante
Parámetros	

The screenshot shows the Postman application interface. At the top, there are two environment dropdowns, both set to "No Environment". Below them are buttons for "Send" and "Save". The main area has tabs for "Authorization", "Headers (1)", "Body", "Pre-request Script", and "Tests", with "Authorization" currently selected. The "Body" tab is also active. The "Headers" section contains one entry: "Content-Type: application/json". The "Body" section is set to "JSON" and contains the following JSON response:

```
1 [  
2   {  
3     "url": "http://www.restaurantepaquito.org/imagenes/comedor"  
4   }  
5 ]
```

The status bar at the bottom right indicates "Status: 200 OK" and "Time: 123 ms".

Método HTTP	GET
URI	/images/:id
Parámetros de entrada	Id de la imagen
Descripción	Método para ver una imagen en concreto
Parámetros	

http://localhost:3000/ ● http://localhost:3000/ ● + *** No Environment Send Save

GET http://localhost:3000/restaurants/14/images/2 Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Code

Type: No Auth

Body Cookies Headers (6) Test Results Status: 200 OK Time: 219 ms

Pretty Raw Preview JSON = □ Q

```

1 [ 
2   {
3     "url": "http://www.restaurantepaquito.org/imagenes/barra"
4   }
5 ]

```

Método HTTP	POST
URI	/images
Parámetros de entrada	
Descripción	Método para crear una imagen
Parámetros	url, imagenesr

http://localhost:3000/ ● http://localhost:3000/ ● + *** No Environment Send Save

POST http://localhost:3000/images Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Code

form-data x-www-form-urlencoded raw binary

Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> url	http://www.restaurantepaquito.org/imagenes/me...			
<input checked="" type="checkbox"/> imagenesr	14			
New key	Value	Description		

Body Cookies Headers (6) Test Results Status: 404 Not Found Time: 76 ms

Drafts Draft Previous HTML = □ ○

POST <http://localhost:3000/restaurants/14/images>

Body (1)

Key	Value	Description
url	http://www.restaurantepaquito.org/imagenes/me...	
imagenesr	14	
New key	Value	Description

Status: 200 OK Time: 251 ms

GET <http://localhost:3000/restaurants/14/images>

Body (1)

```
[{"url": "http://www.restaurantepaquito.org/imagenes/comedor"}, {"url": "http://www.restaurantepaquito.org/imagenes/barra"}, {"url": "http://www.restaurantepaquito.org/imagenes/mesas\\n\\n"}]
```

Status: 200 OK Time: 152 ms

Método HTTP	DELETE
URI	/restaurants/:id/images/:id
Parámetros de entrada	Id del restaurante, id de la imagen
Descripción	Método para eliminar una imagen
Parámetros	

The screenshot shows two separate API requests in the Postman application.

Request 1: DELETE

- Method: **DELETE**
- URL: **http://localhost:3000/restaurants/14/images/3**
- Type: **No Auth**
- Body: **Pretty** view shows the response: **"Imagen eliminada"**
- Status: **200 OK**, Time: **67 ms**

Request 2: GET

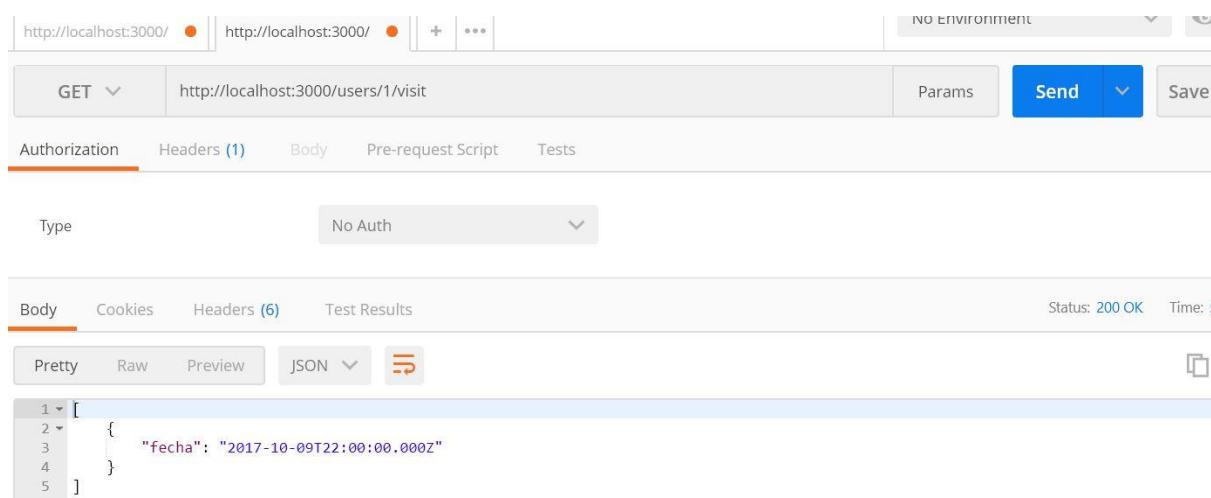
- Method: **GET**
- URL: **http://localhost:3000/restaurants/14/images**
- Type: **No Auth**
- Body: **Pretty** view shows the response:

```
1 [ {  
2   "url": "http://www.restaurantepaquito.org/imagenes/comedor"  
3 },  
4   {  
5     "url": "http://www.restaurantepaquito.org/imagenes/barra"  
6 }  
7 ]  
8 ]
```

- Status: **200 OK**, Time: **58 ms**

VisitaRestaurante

Método HTTP	GET
URI	/users/:id/visit
Parámetros de entrada	Id del usuario
Descripción	Muestra las visitas de un usuario
Parámetros	



The screenshot shows the Postman application interface. At the top, there are two tabs: 'http://localhost:3000/' and 'http://localhost:3000/'. Below them are buttons for 'GET', 'POST', 'PUT', 'DELETE', and 'Params'. To the right of the URL input field is a dropdown for 'Environment' set to 'NO Environment'. The main area shows a 'GET' request with the URL 'http://localhost:3000/users/1/visit'. On the right, there are 'Send' and 'Save' buttons. Below the request area, tabs for 'Authorization', 'Headers (1)', 'Body', 'Pre-request Script', and 'Tests' are visible, with 'Authorization' being the active tab. Under 'Body', the 'Pretty' view shows a JSON response:

```
1 [  
2   {  
3     "fecha": "2017-10-09T22:00:00.000Z"  
4   }  
5 ]
```

The status bar at the bottom indicates 'Status: 200 OK' and 'Time: 5ms'.

Método HTTP	GET
URI	/restaurants/:id/visit
Parámetros de entrada	Id del restaurante
Descripción	Método para ver todas las visitas a un restaurante
Parámetros	

GET ▼ http://localhost:3000/restaurants/14/visit Params Send Save ↴

Authorization Headers (1) Body Pre-request Script Tests Code

Type No Auth

Body Cookies Headers (6) Test Results Status: 200 OK Time: 83 ms

Pretty Raw Preview JSON 🔍

```

1 [ 
2   {
3     "fecha": "2017-10-09T22:00:00.000Z"
4   }
5 ]

```

Método HTTP	GET
URI	/users/:id/visit/:id
Parámetros de entrada	Id del usuario, id del restaurante
Descripción	Método para ver las veces que un usuario en concreto ha visitado un restaurante en particular
Parámetros	

GET ▼ http://localhost:3000/users/1/visit/14 Params Send Save ↴

Authorization Headers (1) Body Pre-request Script Tests Code

Type No Auth

Body Cookies Headers (6) Test Results Status: 200 OK Time: 104 m

Pretty Raw Preview JSON 🔍

```

1 [ 
2   {
3     "fecha": "2017-10-09T22:00:00.000Z"
4   }
5 ]

```

Método HTTP	POST
URI	/users/:id/visit/:id
Parámetros de entrada	Id del usuario, id del restaurante

Descripción	Método para crear una visita de un usuario a un restaurante
Parámetros	fecha

The screenshot shows the Postman interface with two requests:

- POST Request:** URL: http://localhost:3000/users/1/visit/14. Body: Key 'fecha' Value '2018-01-01'. Response: Status: 200 OK, Time: 107 ms. Body content: "Visita Hecha".
- GET Request:** URL: http://localhost:3000/users/1/visit. Response: Status: 200 OK, Time: 280 ms. Body content: [{"fecha": "2017-10-09T22:00:00.000Z"}, {"fecha": "2017-12-31T23:00:00.000Z"}].

Método HTTP	DELETE
URI	/users/:id/visit/:id
Parámetros de entrada	Id del usuario, id de la visita
Descripción	Método para eliminar una visita
Parámetros	

The screenshot shows two API requests in Postman:

- DELETE Request:**
 - URL: `http://localhost:3000/users/1/visit/2`
 - Status: `200 OK`
 - Body: `"Visita eliminada"`
- GET Request:**
 - URL: `http://localhost:3000/users/1/visit`
 - Status: `200 OK`
 - Body: `[{ "fecha": "2017-10-09T22:00:00.000Z" }]`

Pedidos

Método HTTP	GET
URI	/orders
Parámetros de entrada	-
Descripción	Muestra todos los pedidos realizados en la aplicación
Parámetros	

The screenshot shows the Postman interface with the following details:

- Method:** GET
- URL:** localhost:3000/orders
- Authorization:** Inherit auth from parent
- Body:** JSON response (Pretty, Raw, Preview) showing:


```

1 [ 
2   {
3     "CuentaTotal": 12.21,
4     "Mesa": 3
5   }
6 ]
```
- Test Results:** Status: 200 OK, Time: 145 ms, Size: 370 B

Método HTTP	GET
URI	/orders/:id
Parámetros de entrada	Id del pedido
Descripción	Muestra los pedidos según su Id
Parámetros	CuentaTotal, Mesa

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** localhost:3000/orders/1
- Authorization:** Inherit auth from parent
- Description:** This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.
- Status:** 200 OK
- Time:** 7 ms
- Size:** 244 B
- Body:**

```

1 [ ]
2 {
3   "CuentaTotal": 12.21,
4   "Mesa": 3
5 }
6 []

```

Método HTTP	POST
URI	/orders
Parámetros de entrada	-
Descripción	Crea un pedido asociado a una reserva, si esa reserva no está creada, se crea previamente a crear el pedido
Parámetros	Un pedido se crea asignándole el Id de una reserva, Id de un empleado, la cuenta total (importe) y una mesa dentro del local

The screenshot shows two requests in Postman:

- POST Request:** Sent to `localhost:3000/orders`. The Body tab is selected, showing form-data fields: `reservap` (Value: 1), `asignare` (Value: 1), `cuentatotal` (Value: 17.50), and `mesa` (Value: 4). The response status is 200 OK, time 720 ms, and size 354 B. The response body is "Pedido creado".
- GET Request:** Sent to `localhost:3000/orders`. The Authorization tab is selected, showing "Inherit auth from parent". The response status is 200 OK, time 82 ms, and size 274 B. The response body is a JSON array of two objects representing orders.

```

1 ["Pedido creado"]
2 [
3   {
4     "CuentaTotal": 12.21,
5     "Mesa": 3
6   },
7   {
8     "CuentaTotal": 17.5,
9     "Mesa": 4
10 }
]
  
```

Método HTTP	DELETE
URI	/orders/:id
Parámetros de entrada	Id del pedido a eliminar

Descripción	Elimina un pedido
Parámetros	

DELETE localhost:3000/orders/4 Params Send Save

TYPE
Inherit auth from parent

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.

Body Cookies (1) Headers (6) Test Results Status: 200 OK Time: 148 ms Size: 230 B

Pretty Raw Preview JSON

```
1 "Pedido eliminado"
```

Método HTTP	PUT
URI	/orders/:id
Parámetros de entrada	Id del pedido a modificar
Descripción	Modifica un pedido
Parámetros	Se podrán modificar todos los campos de pedido

PUT Params Save

Authorization Headers (1) **Body** Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary

Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> reservap	1			
<input checked="" type="checkbox"/> asignare	1			
<input checked="" type="checkbox"/> cuentatotal	17.50			
<input checked="" type="checkbox"/> mesa	4			
New key	Value	Description		

Body Cookies (1) Headers (6) Test Results Status: 200 OK Time: 157 ms Size: 232 B

Pretty Raw Preview JSON

```
1 "Pedido actualizado"
```

GET Params Save

Authorization Headers (1) Body Pre-request Script Tests Cookies Code

TYPE

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.

Body Cookies (1) Headers (6) Test Results Status: 200 OK Time: 71 ms Size: 243 B

Pretty Raw Preview JSON

```
1 [
2   {
3     "CuentaTotal": 17.5,
4     "Mesa": 4
5   }
6 ]
```

Productos

Método HTTP	GET
URI	/restaurants/:id/products
Parámetros de entrada	Id del restaurante
Descripción	Obtiene los productos de un restaurante
Parámetros	Id, nombre, precio y tipo de producto

The screenshot shows the Postman interface with a successful GET request to `localhost:3000/restaurants/1/products`. The response body is displayed in JSON format:

```

1 [
2   {
3     "idProducto": 1,
4     "nombre": "prueba",
5     "precio": 10,
6     "tipo": "Preparado"
7   }
8 ]

```

Método HTTP	DELETE
URI	/restaurants/:id/products/:idproducto
Parámetros de entrada	Id del restaurante. Id producto a eliminar

Descripción	Obtiene un producto de un restaurante
Parámetros	
<p>DELETE ▾ localhost:3000/restaurants/1/products/1</p> <p>Authorization Headers (2) Body ● Pre-request Script Tests Cookies Code</p> <p>TYPE Inherit auth from parent</p> <p>The authorization header will be automatically generated when you send the request. Learn more about authorization</p>	<p>Params Send ▾ Save ▾</p> <p>Status: 200 OK Time: 147 ms Size: 232 B</p> <p>Pretty Raw Preview JSON ▾  </p> <pre>1 "Producto eliminado"</pre>
<p>GET ▾ localhost:3000/restaurants/1/products/</p> <p>Authorization Headers (2) Body Pre-request Script Tests Cookies Code</p> <p>TYPE Inherit auth from parent</p> <p>The authorization header will be automatically generated when you send the request. Learn more about authorization</p>	<p>Params Send ▾ Save ▾</p> <p>Status: 200 OK Time: 56 ms Size: 212 B</p> <p>Pretty Raw Preview JSON ▾  </p> <pre>1 []</pre>
Método HTTP	POST
URI	/restaurants/:id/products/

Parámetros de entrada	Id del restaurante
Descripción	Crea un producto en un restaurante
Parámetros	Nombre, precio, tipo e información del producto

POST Params **Send** Save

Authorization Headers (2) Body **Pre-request Script** Tests Cookies Code

form-data x-www-form-urlencoded raw binary

Key	Value	Description	...	Bulk Edit
nombre	Ibéricos			
precio	5.75			
tipo	preparar			
informacion	surtido de ibéricos			
New key	Value	Description		

Body Cookies (1) Headers (7) Test Results Status: 200 OK Time: 196 ms Size: 353 B

Pretty Raw Preview JSON

```
1 "Producto creado"
```

GET Params **Send** Save

Authorization Headers (2) Body Pre-request Script Tests Cookies Code

TYPE
Inherit auth from parent

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.

Body Cookies (1) Headers (6) Test Results Status: 200 OK Time: 65 ms Size: 352 B

Pretty Raw Preview JSON

```
1 [
2   {
3     "idProducto": 2,
4     "nombre": "Patatas",
5     "precio": 3.75,
6     "tipo": "Preparar"
7   },
8   {
9     "idProducto": 3,
10    "nombre": "Ibéricos",
11    "precio": 5.75,
12    "tipo": "Preparar"
13  }
14]
```

Método HTTP	PUT
URI	/restaurants/:id/products/:idproducto
Parámetros de entrada	Id del restaurante. Id del producto a modificar
Descripción	Modificar un producto en un restaurante
Parámetros	Permite cambiar todos los campos del producto menos su identificador

The screenshot shows the Postman application interface. At the top, there is a header bar with 'PUT' selected, a URL field containing 'localhost:3000/restaurants/1/products/2', and several buttons: 'Params', 'Send', and 'Save'. Below the header, there are tabs for 'Authorization', 'Headers (2)', 'Body (●)', 'Pre-request Script', and 'Tests'. The 'Body' tab is active, showing a 'form-data' section with a single entry: 'nombre' with value 'Jamón'. There is also a 'raw' section where 'Value' is set to 'Jamón'. At the bottom of the body panel, there are tabs for 'Body', 'Cookies (1)', 'Headers (6)', and 'Test Results'. The 'Body' tab is active. In the 'Test Results' panel, the status is '200 OK', time is '204 ms', and size is '234 B'. Below the results, there are tabs for 'Pretty', 'Raw', 'Preview', and 'JSON' (which is currently selected). The JSON response is displayed in a code editor-like area: '1 "Producto actualizado"'.

The screenshot shows a POSTMAN interface with the following details:

- Method:** GET
- URL:** localhost:3000/restaurants/1/products
- Authorization:** Inherit auth from parent
- Body:** (Pretty, Raw, Preview, JSON) - The JSON response is displayed as follows:


```

1 [ [
2   {
3     "idProducto": 2,
4     "nombre": "Jamón",
5     "precio": 3.75,
6     "tipo": "Preparar"
7   },
8   {
9     "idProducto": 3,
10    "nombre": "Ibéricos",
11    "precio": 5.75,
12    "tipo": "Preparar"
13  }
14 ]
      
```
- Test Results:** Status: 200 OK, Time: 71 ms, Size: 351 B

Método HTTP	GET
URI	/restaurants/:id/products/:type
Parámetros de entrada	Id del restaurante. Tipo de producto
Descripción	Obtiene los productos de un restaurante según su tipo
Parámetros	Nombre y tipo de producto

The screenshot shows a POSTMAN interface with the following details:

- Method:** GET
- URL:** localhost:3000/restaurants/1/products/Preparar
- Authorization:** Inherit auth from parent
- Body:** (Empty)
- Headers:** (2)
- Pre-request Script:** (None)
- Tests:** (None)
- Cookies:** (None)
- Code:** (None)
- Params:** (None)
- Send** button
- Save** button
- Test Results:** Status: 200 OK, Time: 61 ms, Size: 292 B
- Body Content:**

```

1 [ [
2   {
3     "nombre": "Jamón",
4     "tipo": "Preparar"
5   },
6   {
7     "nombre": "Ibéricos",
8     "tipo": "Preparar"
9   }
10 ]

```

Productos de Pedido

Método HTTP	GET
URI	/orders/:id/orderproducts/
Parámetros de entrada	Id del pedido al que se va a asociar el producto
Descripción	Obtiene los productos de un pedido
Parámetros	Id del pedido, Id producto, Tipo de producto, nombre del producto

The screenshot shows a Postman request configuration and its response. The request method is GET, the URL is `localhost:3000/orders/1/orderproducts`, and the response status is 200 OK. The JSON response body contains the following data:

```

1 [ 
2   {
3     "PedidoP": 1,
4     "TipoProducto": "Preparado",
5     "IdProducto": 1,
6     "nombre": "prueba"
7   }
8 ]

```

Método HTTP	GET
URI	<code>/orders/:id/orderproducts/:tipo</code>
Parámetros de entrada	Id del pedido al que se va a asociar el producto. Tipo/Estado de producto (Preparar-preparando-preparado-servido)
Descripción	Obtiene los productos de un pedido según su estado
Parámetros	Id del pedido, Id producto, Tipo de producto, nombre del producto

The screenshot shows a POST request to `localhost:3000/orders/1/orderproducts/Preparado`. The response body is a JSON array with one element:

```

1 [ 
2   {
3     "PedidoP": 1,
4     "TipoProducto": "Preparado",
5     "IdProducto": 1,
6     "nombre": "prueba"
7   }
8 ]

```

Método HTTP	POST
URI	<code>/orders/:id/orderproducts/</code>
Parámetros de entrada	Id del pedido al que se va a asociar el producto
Descripción	Crea un producto asociado a un pedido. Realmente lo que hace es coger un producto ya creado y asociarlo a un pedido
Parámetros	Se asignará el Id de un pedido existente, el Id del producto existente, el tipo de producto (si es un plato que hay que preparar o si no, por ejemplo un refresco) y la hora en la que se ha solicitado ese producto

POST Params Save

Authorization Headers (1) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary

Key	Value	Description	...	Bulk Edit
pedidop	1			
productop	1			
tipoproducto	Preparado			
hora	2017-12-12 14:25			
New key	Value	Description		

Body Cookies (1) Headers (7) Test Results Status: 200 OK Time: 177 ms Size: 365 B

Pretty Raw Preview JSON

```
1 "Producto de pedido creado"
```

GET Params Save

Authorization Headers (2) Body Pre-request Script Tests Cookies Code

TYPE
Inherit auth from parent

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.

Body Cookies (1) Headers (7) Test Results Status: 200 OK Time: 279 ms Size: 417 B

Pretty Raw Preview JSON

```
1 [
2   {
3     "PedidoP": 1,
4     "TipoProducto": "Preparado",
5     "IdProducto": 1,
6     "nombre": "prueba"
7   }
8 ]
```

Método HTTP	DELETE
URI	/orders/:id/orderproducts/:idpp
Parámetros de entrada	Id del pedido al que se va a asociar el producto. Id del producto que se va a eliminar de ese pedido

Descripción	Elimina un producto de un pedido
Parámetros	

The screenshot shows the Postman interface with a DELETE request to `localhost:3000/orders/1/orderproducts/`. The request body is set to `x-www-form-urlencoded` and contains the following parameters:

Key	Value	Description
pedidop	1	
productop	1	
tipoproducto	Preparado	
hora	2017-12-12 14:25	
New key	Value	Description

The response status is `200 OK`, time `217 ms`, and size `242 B`. The response body is `"Producto de pedido eliminado"`.

Método HTTP	PUT
URI	<code>/orders/:id/orderproducts/:idpp/preparando/:idp</code>
Parámetros de entrada	Id del pedido al que se va a asociar el producto. Id del producto de pedido que se va a actualizar de ese pedido. Id del registro del producto
Descripción	Cambia el estado de un producto de un pedido de preparar a preparando. Para cuando un producto entra en cocina para su preparación
Parámetros	Tipo producto (preparar-preparando-preparado-servido)

PUT ▾ | localhost:3000/orders/1/orderproducts/1/preparando/1 | Params | Send | Save

Authorization Headers (1) Body Pre-request Script Tests Cookies Code

TYPE

Inherit auth from parent

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.

Body Cookies (1) Headers (6) Test Results Status: 200 OK Time: 75 ms Size: 235 B

Pretty Raw Preview JSON

1 "Producto Preparandose"

Método HTTP	PUT
URI	/orders/:id/orderproducts/:idpp/preparado/:idp
Parámetros de entrada	Id del pedido al que se va a asociar el producto. Id del producto de pedido que se va a actualizar de ese pedido. Id del registro del producto
Descripción	Cambia el estado de un producto de un pedido de preparando a preparado. Para cuando un producto sale de cocina para que el camarero lo lleve a la mesa
Parámetros	Tipo producto (preparar-preparando-preparado-servido)

PUT ▾

localhost:3000/orders/1/orderproducts/1/preparando/1

Params

Send ▾

Save ▾

Authorization Headers (1) Body ● Pre-request Script Tests Cookies Code

TYPE

Inherit auth from parent

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.

Body Cookies (1) Headers (6) Test Results

Status: 200 OK Time: 60 ms Size: 232 B

Pretty Raw Preview JSON ▾

1 "Producto Preparado"

Método HTTP	PUT
URI	/orders/:id/orderproducts/:idpp/servido/:idp
Parámetros de entrada	Id del pedido al que se va a asociar el producto. Id del producto de pedido que se va a actualizar de ese pedido. Id del registro del producto
Descripción	Cambia el estado de un producto de un pedido de preparando a preparado. Para cuando un producto llega a la mesa y el camarero lo marca como entregado o servido al cliente
Parámetros	Tipo producto (preparar-preparando-preparado-servido)

The screenshot shows the Postman application interface. At the top, there's a header with 'PUT' (dropdown), 'localhost:3000/orders/1/orderproducts/1/servido/1' (URL input), 'Params' (button), 'Send' (blue button), and 'Save' (button). Below the header are tabs: 'Authorization' (selected), 'Headers (1)', 'Body' (radio button selected), 'Pre-request Script', 'Tests', 'Cookies' (button), and 'Code' (button). The 'Body' tab has a sub-section 'TYPE' with a dropdown set to 'Inherit auth from parent'. A note says: 'This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.' Below this, there are tabs: 'Body' (selected), 'Cookies (1)', 'Headers (6)', and 'Test Results'. On the right, status information is shown: 'Status: 200 OK', 'Time: 80 ms', and 'Size: 230 B'. Under the 'Body' tab, there are three options: 'Pretty' (selected), 'Raw', and 'Preview'. The 'Pretty' view shows a single item: '1 "Producto Servido"'. To the right of the preview area are icons for copy and search.

Facturas

Método HTTP	GET
URI	/orders/:idusuario/bill
Parámetros de entrada	Id de un usuario
Descripción	Obtiene las facturas de un usuario
Parámetros	Cuenta total

The screenshot shows the Postman interface with the following details:

- Method:** GET
- URL:** localhost:3000/orders/1/bill
- Authorization:** Inherit auth from parent
- Body:** JSON response (Pretty, Raw, Preview) showing:


```

1 [ ]
2 [
3   {
4     "cuentatotal": 17.5
5   },
6   {
7     "cuentatotal": 17.5
8 }
      
```
- Test Results:** Status: 200 OK, Time: 83 ms, Size: 255 B

Método HTTP	GET
URI	/orders/:idusuario/bill/name/:nombre
Parámetros de entrada	Id de un usuario, Nombre de un restaurante
Descripción	Obtiene las facturas de un usuario en un restaurante en concreto
Parámetros	Nombre reserva, Cuenta total

GET localhost:3000/orders/1/bill/name/faslñdfjalsfd

Authorization Headers (1) Body Pre-request Script Tests Cookies Code

TYPE
Inherit auth from parent

This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.

Body Cookies (1) Headers (6) Test Results Status: 200 OK Time: 56 ms Size: 307 B

Pretty Raw Preview JSON

```

1 [ ]
2 {
3   "nombre": "faslñdfjalsfd",
4   "cuentatotal": 17.5
5 },
6 {
7   "nombre": "faslñdfjalsfd",
8   "cuentatotal": 17.5
9 }
10 ]

```

Método HTTP	GET
URI	/orders/:idusuario/bill/filter/:cuentainicial/:cu entafinal
Parámetros de entrada	Id de un usuario, Precio mínimo, Precio máximo
Descripción	Obtiene las facturas de un usuario filtrado por precio
Parámetros	Cuenta total

The screenshot shows the Postman interface with the following details:

- Method:** GET
- URL:** localhost:3000/orders/1/bill/filter/10/18
- Authorization:** Inherit auth from parent
- Body:** JSON response (Pretty, Raw, Preview) showing:


```

1 [ ]
2 [
3   {
4     "cuentatotal": 17.5
5   },
6   {
7     "cuentatotal": 17.5
8 }
      
```
- Test Results:** Status: 200 OK, Time: 83 ms, Size: 255 B

Método HTTP	GET
URI	/orders/:idusuario/bill/min/:cuenta
Parámetros de entrada	Id de un usuario, Precio mínimo
Descripción	Obtiene las facturas de un usuario filtrado por precio mínimo
Parámetros	Cuenta total

The screenshot shows the Postman application interface. At the top, there's a header bar with 'GET', 'localhost:3000/orders/1/bill/min/10', 'Params', 'Send', 'Save', and tabs for 'Authorization', 'Headers (1)', 'Body', 'Pre-request Script', 'Tests', 'Cookies', and 'Code'. Below this, the 'Authorization' tab is selected, showing a dropdown menu 'Inherit auth from parent'. A note says: 'This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.' Under the 'Body' tab, the response is displayed as JSON:

```

1 [ ]
2   [
3     {
4       "cuentatotal": 17.5
5     },
6     {
7       "cuentatotal": 17.5
8     }
9   ]

```

At the bottom right, status information is shown: 'Status: 200 OK', 'Time: 73 ms', and 'Size: 255 B'. There are also 'Pretty', 'Raw', 'Preview', and 'JSON' buttons.

Método HTTP	GET
URI	/orders/:idusuario/bill/max/cuenta
Parámetros de entrada	Id de un usuario, Precio máximo
Descripción	Obtiene las facturas de un usuario filtrado por precio máximo
Parámetros	Cuenta total

The screenshot shows a Postman request configuration and its response. The request method is GET, the URL is `localhost:3000/orders/1/bill/max/20`, and the response status is 200 OK with a size of 255 B. The response body is a JSON array with two elements, each containing a key `cuentatotal` with a value of 17.5.

```

1 [ ]
2   {
3     "cuentatotal": 17.5
4   },
5   {
6     "cuentatotal": 17.5
7   }
8 ]
  
```

Empleados

Método HTTP	GET
URI	/restaurants/:id/employee/
Parámetros de entrada	Id de un restaurante
Descripción	Obtener todos los empleados de un restaurante
Parámetros	Nick y tipo de empleado

The screenshot shows the Postman application interface. At the top, there's a header bar with 'GET', 'localhost:3000/restaurants/1/employee', 'Params', 'Send', and 'Save'. Below the header are tabs: 'Authorization' (which is selected), 'Headers (2)', 'Body', 'Pre-request Script', 'Tests', 'Cookies', and 'Code'. The main area has a 'TYPE' dropdown set to 'Inherit auth from parent'. A note says: 'This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.' Under the 'Body' tab, there are tabs for 'Pretty', 'Raw', 'Preview', 'JSON', and a copy/paste icon. The 'Pretty' tab displays the following JSON:

```

1 [ [
2   {
3     "nick": "Luis Enrique",
4     "tipoempleado": "Camarero"
5   },
6   {
7     "nick": "Luis",
8     "tipoempleado": "Camarero"
9   }
10 ]

```

At the bottom right of the main area, there are status indicators: 'Status: 200 OK', 'Time: 104 ms', and 'Size: 433 B'. There are also icons for copy and search.

Método HTTP	GET
URI	/restaurants/:id/employee/:idempleado
Parámetros de entrada	Id de un restaurante. Id de un empleado
Descripción	Obtener un empleado de un restaurante según su Id
Parámetros	Nick y tipo de empleado

The screenshot shows the Postman application interface. At the top, there is a header bar with buttons for 'GET' (selected), 'localhost:3000/restaurants/1/employee/1' (URL), 'Params' (button), 'Send' (button), and 'Save' (button). Below the header, there are tabs for 'Authorization' (selected), 'Headers (2)', 'Body', 'Pre-request Script', 'Tests', 'Cookies' (button), and 'Code' (button). The 'Authorization' tab has a dropdown menu set to 'Inherit auth from parent'. A note below it says: 'The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)'. The 'Body' tab is selected, showing the response content: a JSON array with one element. The JSON is:

```

1 [ 
2   { 
3     "nick": "danibeam",
4     "tipoempleado": "Cocinero"
5   }
6 ]

```

Below the body, status information is displayed: Status: 200 OK, Time: 54 ms, Size: 259 B. There are also buttons for 'Pretty', 'Raw', 'Preview', 'JSON', and a copy icon.

Método HTTP	DELETE
URI	/restaurants/:id/employee/:idempleado
Parámetros de entrada	Id de un restaurante. Id de un empleado
Descripción	Elimina un empleado de un restaurante
Parámetros	

The screenshot shows two requests in the Postman interface:

- DELETE Request:**
 - Method: **DELETE**
 - URL: **localhost:3000/restaurants/1/employee/1**
 - Body: **Inherit auth from parent**
 - Response Status: **200 OK**, Time: **258 ms**, Size: **232 B**
 - Body content: **"Empleado eliminado"**
- GET Request:**
 - Method: **GET**
 - URL: **localhost:3000/restaurants/1/employee/1**
 - Body: **Inherit auth from parent**
 - Response Status: **200 OK**, Time: **64 ms**, Size: **212 B**
 - Body content: **[]**

Método HTTP	POST
URI	/restaurants/:id/employee/:idempleado
Parámetros de entrada	Id de un restaurante

Descripción	Crear un empleado (dar de alta) en un restaurante determinado																														
Parámetros	Nick, contraseña, tipo de empleado (Camarero-Cocinero) y reserva																														
<p>POST <input type="button" value="localhost:3000/restaurants/1/employee"/> Params <input type="button" value="Send"/> Save</p> <p>Authorization Headers (2) Body <input checked="" type="radio"/> Pre-request Script Tests Cookies Code</p> <p><input type="radio"/> form-data <input checked="" type="radio"/> x-www-form-urlencoded <input type="radio"/> raw <input type="radio"/> binary</p> <table border="1"> <thead> <tr> <th>Key</th> <th>Value</th> <th>Description</th> <th>...</th> <th>Bulk Edit</th> </tr> </thead> <tbody> <tr> <td>nick</td> <td>Luis</td> <td></td> <td></td> <td></td> </tr> <tr> <td>password</td> <td>1234</td> <td></td> <td></td> <td></td> </tr> <tr> <td>tipoempleado</td> <td>Camarero</td> <td></td> <td></td> <td></td> </tr> <tr> <td>empleado</td> <td>1</td> <td></td> <td></td> <td></td> </tr> <tr> <td>New key</td> <td>Value</td> <td>Description</td> <td></td> <td></td> </tr> </tbody> </table> <p>Body Cookies (1) Headers (7) Test Results Status: 200 OK Time: 339 ms Size: 353 B</p> <p>Pretty Raw Preview JSON <input type="button"/></p> <pre>1 "Empleado creado"</pre>		Key	Value	Description	...	Bulk Edit	nick	Luis				password	1234				tipoempleado	Camarero				empleado	1				New key	Value	Description		
Key	Value	Description	...	Bulk Edit																											
nick	Luis																														
password	1234																														
tipoempleado	Camarero																														
empleado	1																														
New key	Value	Description																													
<p>GET <input type="button" value="localhost:3000/restaurants/1/employee/2"/> Params <input type="button" value="Send"/> Save</p> <p>Authorization Headers (2) Body Pre-request Script Tests Cookies Code</p> <p>TYPE Inherit auth from parent</p> <p>The authorization header will be automatically generated when you send the request. Learn more about authorization</p> <p>This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.</p> <p>Body Cookies (1) Headers (6) Test Results Status: 200 OK Time: 78 ms Size: 258 B</p> <p>Pretty Raw Preview JSON <input type="button"/></p> <pre>1 [2 { 3 "nick": "Antonio", 4 "tipoempleado": "Camarero" 5 } 6]</pre>																															
Método HTTP	PUT																														

URI	/restaurants/:id/employee/:idempleado
Parámetros de entrada	Id de un restaurante. Id de un empleado a modificar
Descripción	Modificar un empleado de un restaurante
Parámetros	Se pueden modificar todos los campos de empleado menos su identificador

The screenshot shows the Postman application interface. At the top, there is a header bar with 'PUT' selected, a URL field containing 'localhost:3000/restaurants/1/employee/2', and buttons for 'Params', 'Send', and 'Save'. Below the header, there are tabs for 'Authorization', 'Headers (2)', 'Body' (which is currently selected), 'Pre-request Script', and 'Tests'. Under the 'Body' tab, there are options for 'form-data', 'x-www-form-urlencoded' (which is selected), 'raw', and 'binary'. A table below shows a key-value pair: 'nick' with the value 'Luis Enrique'. There is also a row for 'New key' with a placeholder 'Value' and 'Description'. At the bottom of the interface, there are tabs for 'Body' (selected), 'Cookies (1)', 'Headers (6)', and 'Test Results'. The 'Test Results' tab shows a status of 'Status: 200 OK', 'Time: 106 ms', and 'Size: 234 B'. Below this, there are buttons for 'Pretty', 'Raw', 'Preview', and 'JSON'. The JSON preview pane displays the response: '1 "Empleado actualizado"'.

GET localhost:3000/restaurants/1/employee/2

Params Send Save

Authorization Headers (2) Body Pre-request Script Tests Cookies Code

TYPE
Inherit auth from parent

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.

Body Cookies (1) Headers (6) Test Results Status: 200 OK Time: 71 ms Size: 263 B

Pretty Raw Preview JSON

```

1 [ 
2   {
3     "nick": "Luis Enrique",
4     "tipoempleado": "Camarero"
5   }
6 ]

```

Método HTTP	GET
URI	/orders/:id/employee/:idempleado
Parámetros de entrada	Id de un pedido, Id del empleado al que va a llamar
Descripción	Avisar al camarero para que acuda a una mesa
Parámetros	Mesa

The screenshot shows the Postman application interface. At the top, there is a header bar with buttons for 'GET' (selected), 'localhost:3000/orders/1/employee/1' (the URL), 'Params' (button), 'Send' (button), and 'Save' (button). Below the header, there are tabs for 'Authorization' (selected), 'Headers (1)', 'Body', 'Pre-request Script', 'Tests', 'Cookies' (button), and 'Code' (button).

The 'Authorization' tab contains a section titled 'TYPE' with a dropdown menu set to 'Inherit auth from parent'. A note below states: 'The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)'.

The 'Body' tab is selected, showing a JSON response:

```
1 [ ]  
2 {  
3   "mesa": 4  
4 }  
5 [ ]
```


At the bottom right of the interface, there are status indicators: 'Status: 200 OK', 'Time: 85 ms', and 'Size: 223 B'. There are also buttons for 'Pretty', 'Raw', 'Preview', 'JSON' (selected), and a copy icon.