



GUSTAVO CORONEL
DESARROLLA SOFTWARE



TRABAJANDO CON POSTMAN

TRABAJANDO CON RECURSOS

Eric Gustavo Coronel Castillo

I N S T R U C T O R

youtube.com/DesarrollaSoftware

gcoronelc@gmail.com

LOGRO ESPERADO

Al finalizar esta lección el participante tener claro cuales son los estándares para trabajar con recursos.



Recursos

Un recurso es un objeto o entidad que es identificable y manipulable a través de una URL (Uniform Resource Locator).

Los recursos son el concepto central en la arquitectura REST y representan información o datos que pueden ser accedidos y modificados mediante las operaciones HTTP estándar, como GET, POST, PUT y DELETE.

Ejemplo

<https://api.desarrollasoftware.com/users>

- El recurso es el usuario: **/users**
- La URL completa se conoce como **endpoint**
- De esta manera se expone el recurso usuario para que pueda ser manipulado

¿Cómo escribir un buen recurso?

- Se deben usar sustantivos, no usar verbos
- Se debe usar en plural

EJEMPLOS CORRECTOS

- `https://api.desarrollasoftware.com/orders`
- `https://api.desarrollasoftware.com/orders/1`

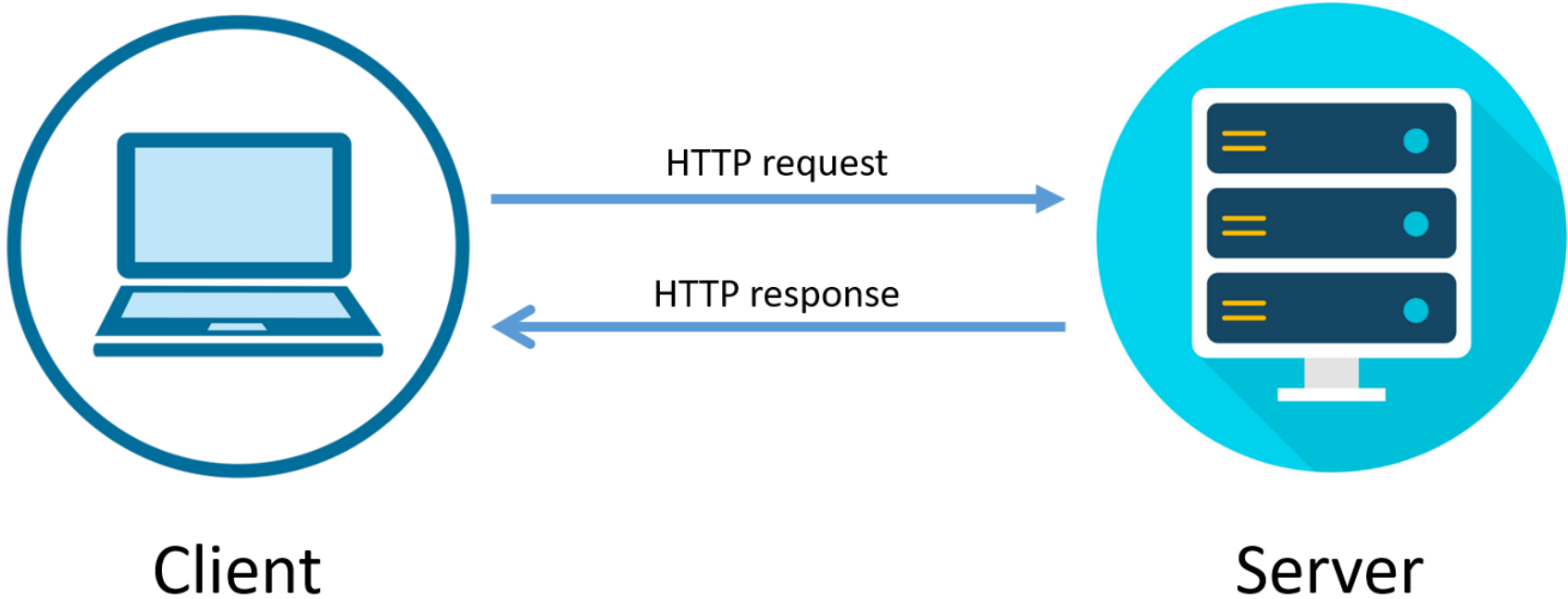
EJEMPLOS INCORRECTOS

- `https://api.desarrollasoftware.com/orders/getAll`
- `https://api.desarrollasoftware.com/orders/getById`

¿Cómo acceder a un recurso?

VERBO	EJEMPLO
GET	<code>https://api.desarrollasoftware.com/users</code>
GET	<code>https://api.desarrollasoftware.com/users/5</code>
POST	<code>https://api.desarrollasoftware.com/users</code>
PUT	<code>https://api.desarrollasoftware.com/users/5</code>
PATCH	<code>https://api.desarrollasoftware.com/users/5</code>
DELETE	<code>https://api.desarrollasoftware.com/users/5</code>

HTTP Request y HTTP Response



HTTP Request

- Un request es una **petición**
- Se quiere **solicitar** algo a la API
- Un **HTTP Request** es una petición que hace uso del protocolo **HTTP**

Partes de un request

Header	Body	QueryString
Información de metadata como la IP del usuario, User Agent, información del Token.	Información relevante para nuestra API ya que es lo que va a manipular y lo más común es mandarla en formato JSON . <pre>{ "username": "erodriguez", "password": "123456" }</pre>	Información que forma parte de la URL como parámetros. <code>api.kodoti.com/users?order By=creationDate</code>

HTTP Response

- Es una **respuesta**
- Lo que se espera que **responda** la API
- Se acompaña de un **STATUS CODE**

Partes de un response

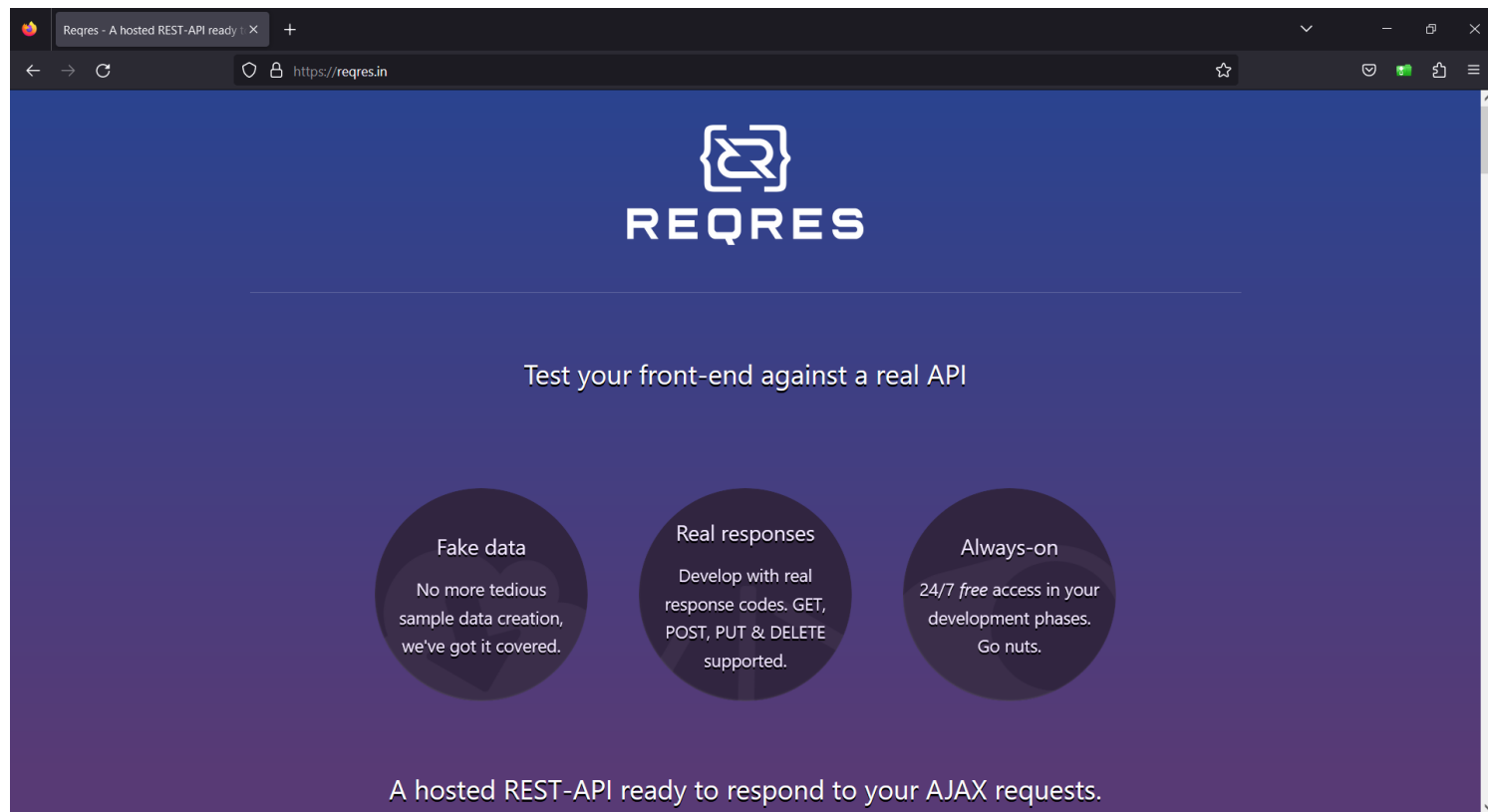
Header	Body
Información de metadata que manda el servidor al cliente.	La respuesta que adjunta el servidor al cliente para complacer su REQUEST . <pre>{ "id": 1, "name": "Eduardo", ... }</pre>

Status Code

- Las respuestas están acompañadas por un **código de estado**.
- El código de estado nos ofrece información de **como se procesó la información**.

- **404**: *el recurso no fue encontrado.*
- **400**: *bad request.*
- **500**: *Internal Server error.*
- **20x**: *Satisfactorios*

Ejemplos



GRACIAS
TOTALES



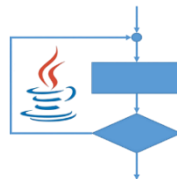
Gustavo Coronel
gcoronelc.github.io



gcoronelc.github.io

youtube.com/DesarrollaSoftware

facebook.com/groups/desarrollasoftware



FUNDAMENTOS DE PROGRAMACIÓN CON JAVA

Inicia tu aprendizaje, utilizando las mejores prácticas de programación



CURSO PROFESIONAL DE JAVA ORIENTADA A OBJETOS

Aprende programación en capas, patrones y buenas prácticas



PROGRAMACIÓN DE BASE DE DATOS ORACLE CON PL/SQL

Aprende a obtener el mejor rendimiento de tú base de datos



PROGRAMACIÓN DE BASE DE DATOS ORACLE CON JDBC

Aprende a programar correctamente con JDBC