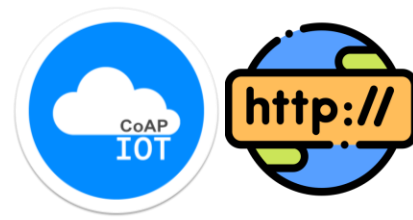


# Módulo 5:

## Aplicación de Protocolos de Capas para IoT

# Agenda



- Modulo 5:
  - Bases de HTTP
  - Bases de CoAP
- Laboratorio 5:
  - HTTP en Acción
  - CoAP en Acción
  - Inspección de Paquetes



# Maquina virtual

Link de descarga de la máquina virtual a ser usada en los laboratorios de la semana 2. Se necesitará también para el proyecto final

- [https://1drv.ms/u/c/389d243860c18a05/Eb8hTPPh6ZWJDlwrQegUiyZIBHM - 79SurwodKI10eUuJAA?e=9rg2VM](https://1drv.ms/u/c/389d243860c18a05/Eb8hTPPh6ZWJDlwrQegUiyZIBHM-79SurwodKI10eUuJAA?e=9rg2VM)  
(opcion 1)
- [https://drive.google.com/file/d/1oUB9clrBq\\_ORV7gx0ons8Tn6h1XZAZ4n/view?usp=drive link](https://drive.google.com/file/d/1oUB9clrBq_ORV7gx0ons8Tn6h1XZAZ4n/view?usp=drive_link)  
(opcion 2)

Esta máquina virtual contiene software como:

- Copper
- Node-red
- Wireshark

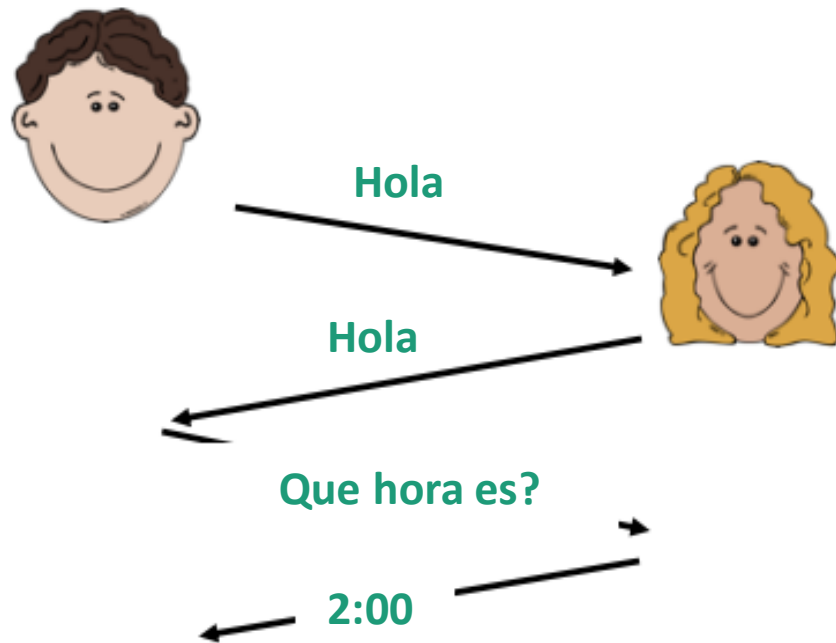


Notas:

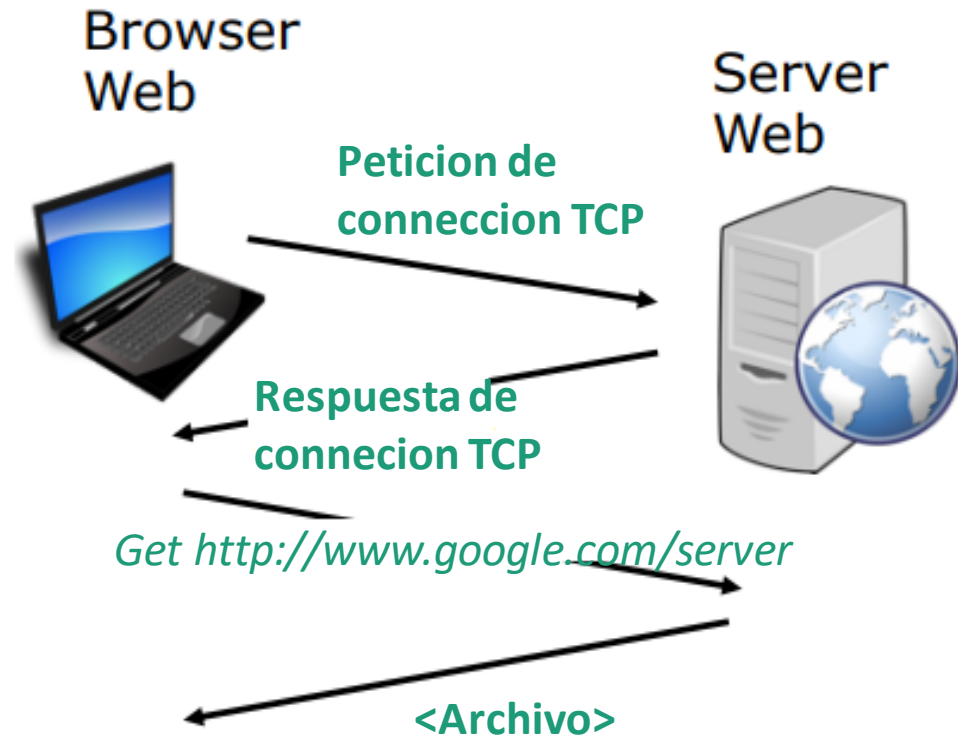
- la VM pesa 3GB por lo que tardara alrededor de 30min – 1h en instalarse
- Se necesita Oracle Virtual Box para abrir esta maquina virtual
  - <https://www.virtualbox.org/wiki/Downloads>

# Protocolos de Comunicación

## Protocolos de comunicación humana



## Protocolos de comunicación de redes



# Tipos de Capas de Aplicación

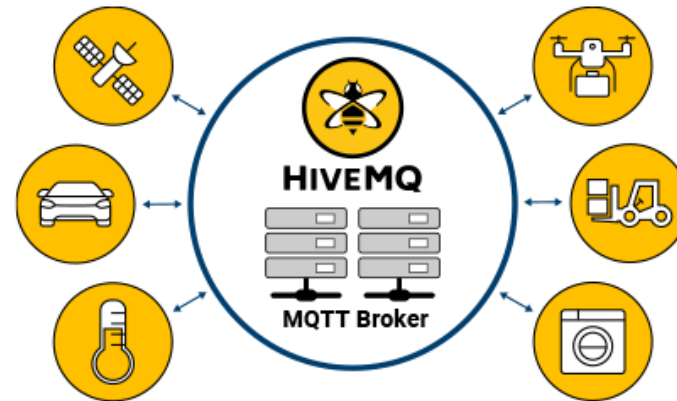
**Cliente/Servicio (navegador web):**

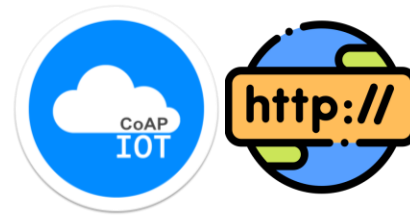
- HTTP
- CoAP



**Publicador/Subscriber (Twitter):**

- MQTT

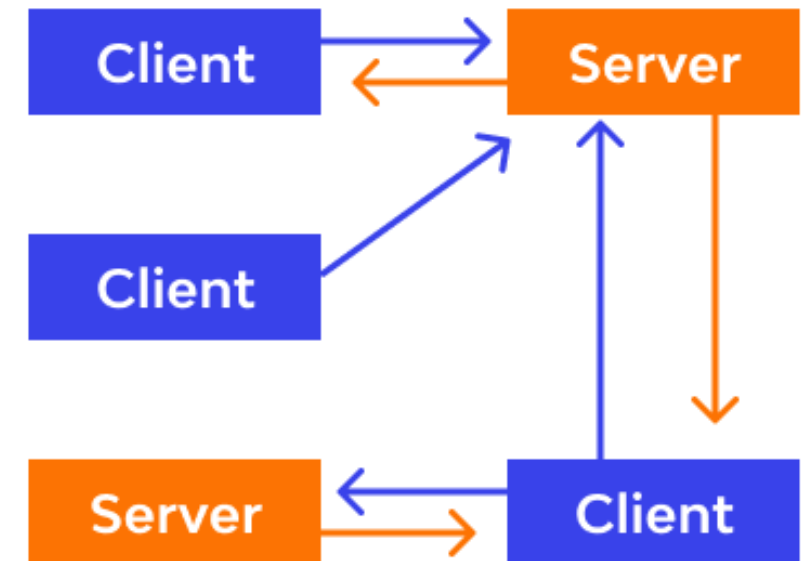




# ¿Como funciona el Internet?

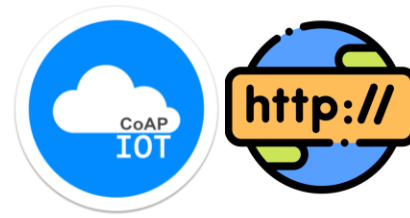
Recursos en la web:

- Servidores
- URIs (Uniform Resource Identifiers) Identificadores
- Clientes con requisito/respuesta

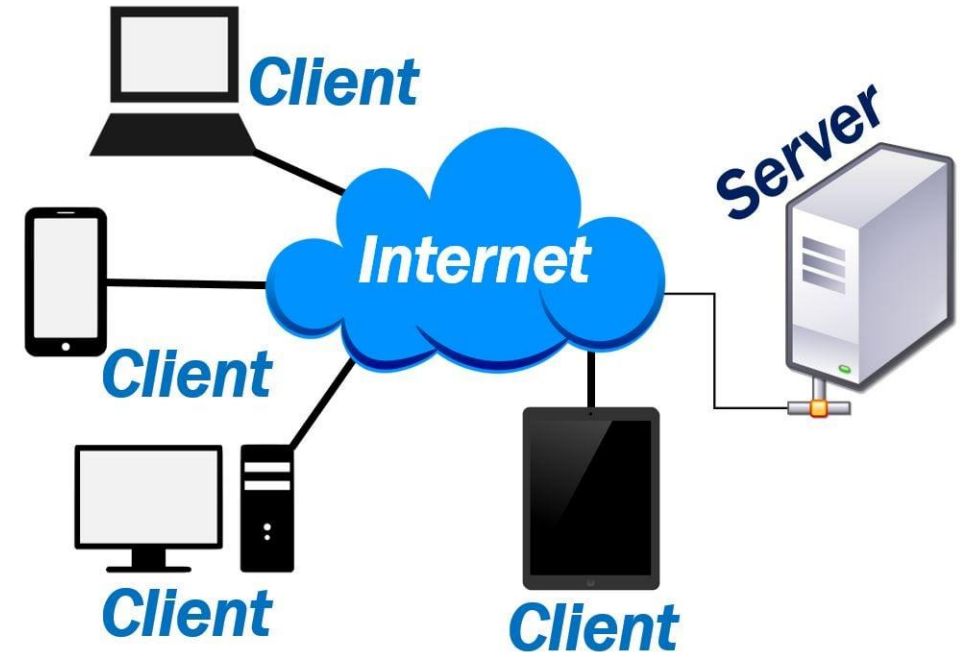


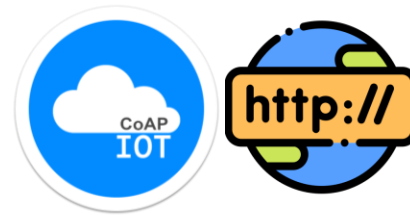
# Protocolo HTTP

## (Hypertext Transfer Protocol)



- Protocolo de comunicación de Cliente/Servidor:
  - **Cliente:** Envía un HTTP requisito de un recurso específico (URIs).
  - **Servidor:** Contesta con el recurso requerido.





# Protocolo HTTP: Petición

Método

Versión del Protocolo

POST / HTTP/1.1

Host: localhost:8000

User-Agent: Mozilla/5.0 (Macintosh;... )... Firefox/51.0

Accept: text/html,application/xhtml+xml,..., \*/\*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Connection: keep-alive

Upgrade-Insecure-Requests: 1

Content-Type: multipart/form-data; boundary=-12656974

Content-Length: 345

Primera  
línea

Encabezados  
de HTTP

Línea  
Vacía

Cuerpo

-12656974

(more data)

Métodos:

- GET: Lee Recursos
- POST: Crea Nuevos Recursos
- PUT: Actualiza Recursos
- DELETE: Elimina Recursos.



# Protocolo HTTP: Respuesta

Versión del Protocolo    Códigos de Status

Primera línea → HTTP/1.1 403 Forbidden

Encabezados de HTTP

```
Server: Apache
Content-Type: text/html; charset=iso-8859-1
Date: Wed, 10 Aug 2016 09:23:25 GMT
Keep-Alive: timeout=5, max=1000
Connection: Keep-Alive
Age: 3464
Date: Wed, 10 Aug 2016 09:46:25 GMT
X-Cache-Info: caching
Content-Length: 220
```

Línea Vacía →

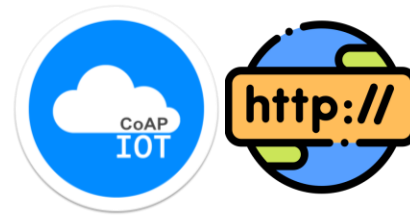
Cuerpo →

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML
2.0//EN">
  (more data)
```

Códigos de Status:

- 1XX: Información del Servidor.
- 2XX: Respuesta es un éxito.
- 3XX: Redirección de Comunicación.
- 4XX: Error del Cliente. (403)
- 5XX: Error del Servidor.

# CoAP (Constrained Application Protocol)



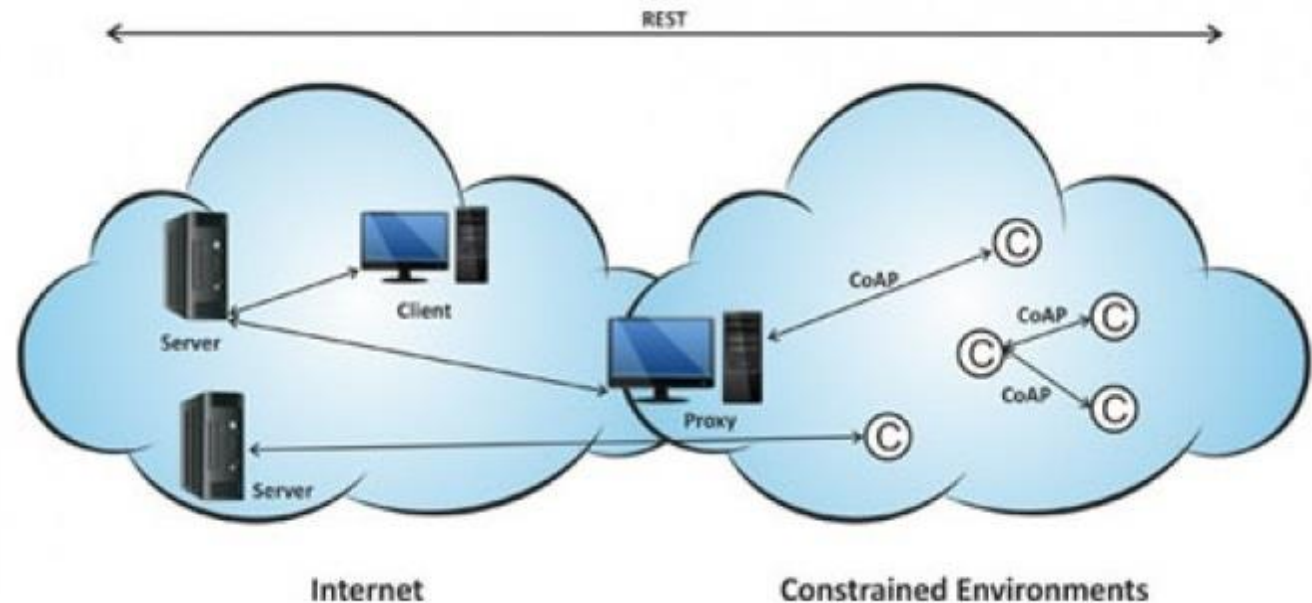
Protocolo de comunicación de Cliente/Servicio:

Cliente: Envía un HTTP requisito de un recurso específico (URIs).

Servidor: Contesta con el recurso requerido.

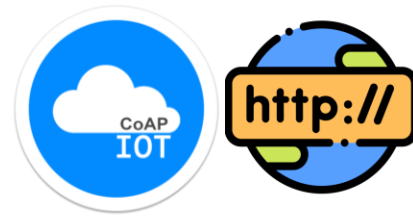
Proxy: Ruter o sistema que proporciona conexión a internet

CoAP: permite acceso a servicios web a dispositivos de limitada memoria, procesamiento



# TCP vs UDP

## (Transmission Control Protocol) vs (User Datagram Protocol)



TCP es un protocolo de conexión orientada.

Establece conexión entre dispositivos antes de transferir información.

HTTP usa TCP para asegurarse que toda la información sea recibida intacta.

HTTP envía mensajes largos y requiere mucho procesamiento de cpu y memoria que se traduce en consumo de energía.

UDP es un protocolo que solo envía paquetes de información sin establecer una conexión o asegurarse que la información haya llegado completa.

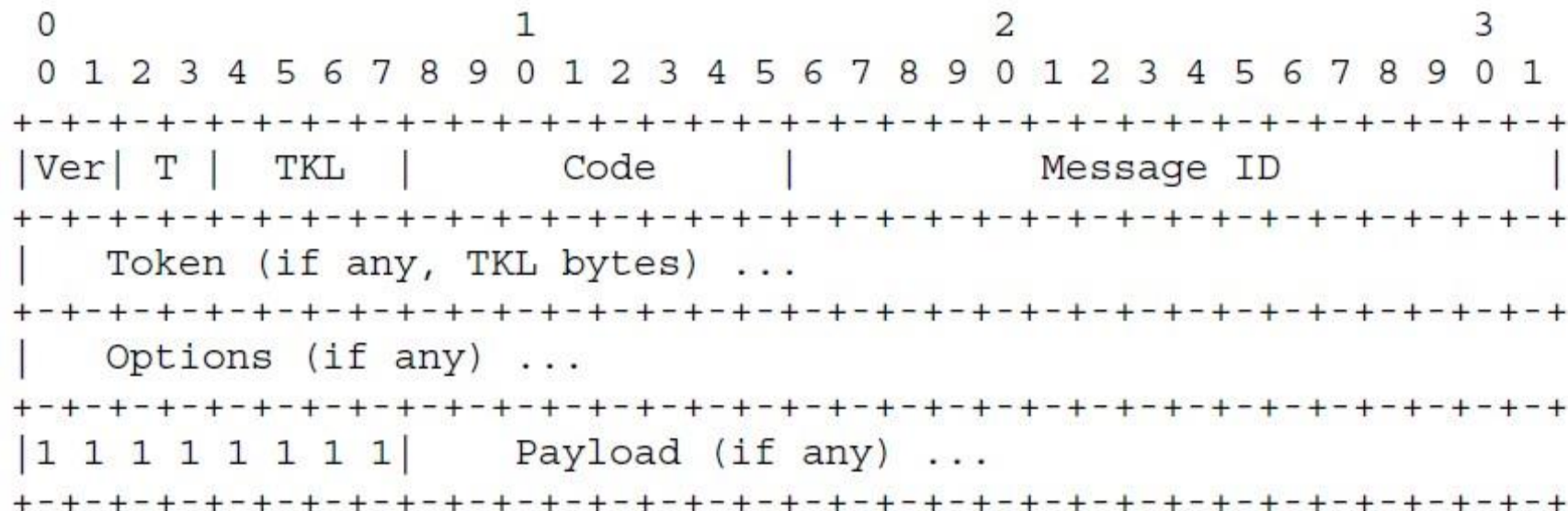
CoAP usa UDP en aplicaciones donde discontinuidad de la comunicación es aceptable.

Requiere poca memoria y procesamiento entonces ideal con dispositivos IoT.

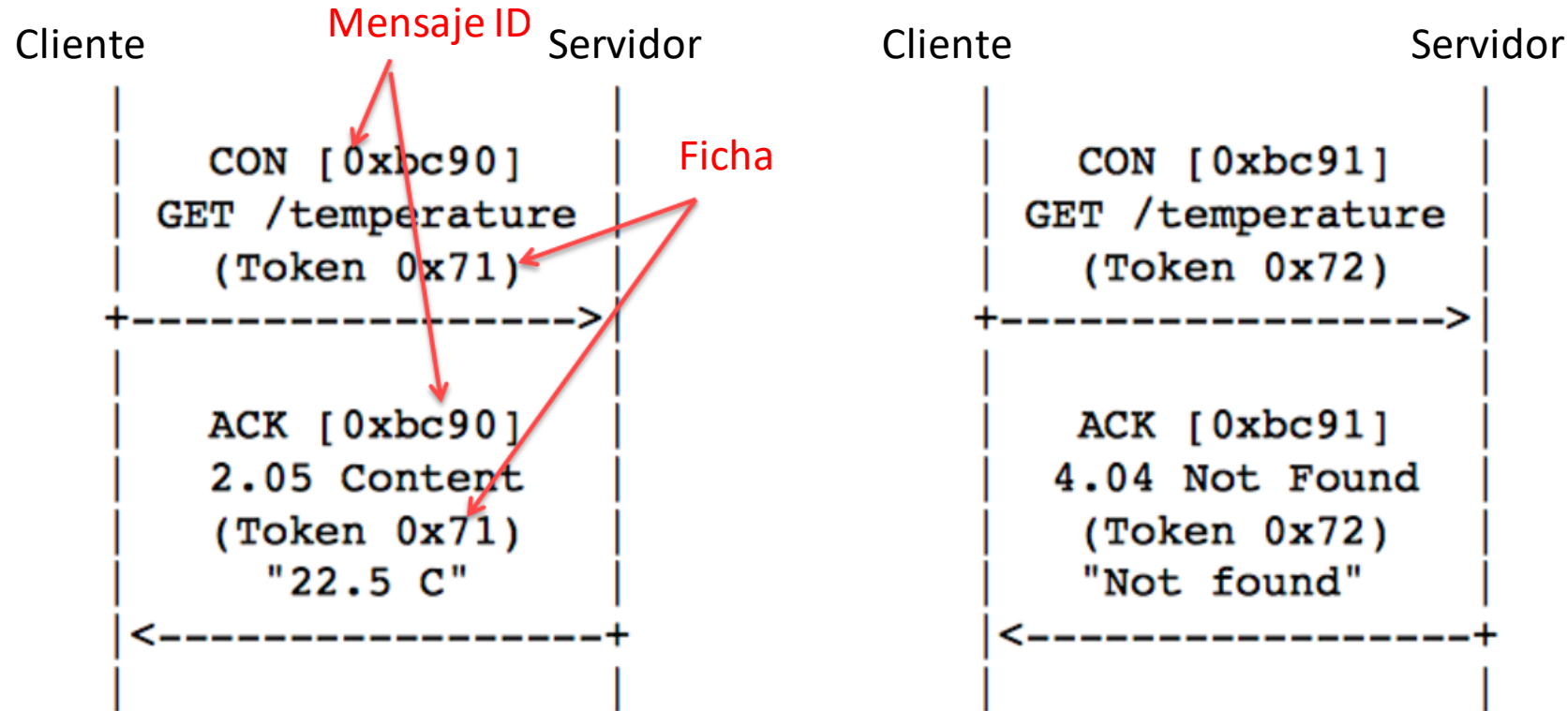
Métodos de comunicación similares a HTTP (GET, PUT, PUSH y DELET)



Token – Opcional: ficha de respuesta emparejadora



# CoAP Mensajes Requisito/Respuesta



# CoAP Código de Respuesta



	Code	Description	Reference
SUCCESS (2.xx)	2.01	Created	<a href="#">[RFC7252]</a>
	2.02	Deleted	<a href="#">[RFC7252]</a>
	2.03	Valid	<a href="#">[RFC7252]</a>
	2.04	Changed	<a href="#">[RFC7252]</a>
	2.05	Content	<a href="#">[RFC7252]</a>
CLIENT ERROR (4.xx)	4.00	Bad Request	<a href="#">[RFC7252]</a>
	4.01	Unauthorized	<a href="#">[RFC7252]</a>
	4.02	Bad Option	<a href="#">[RFC7252]</a>
	4.03	Forbidden	<a href="#">[RFC7252]</a>
	4.04	Not Found	<a href="#">[RFC7252]</a>
	4.05	Method Not Allowed	<a href="#">[RFC7252]</a>
	4.06	Not Acceptable	<a href="#">[RFC7252]</a>
	4.12	Precondition Failed	<a href="#">[RFC7252]</a>
	4.13	Request Entity Too Large	<a href="#">[RFC7252]</a>
	4.15	Unsupported Content-Format	<a href="#">[RFC7252]</a>
SERVER ERROR (5.xx)	5.00	Internal Server Error	<a href="#">[RFC7252]</a>
	5.01	Not Implemented	<a href="#">[RFC7252]</a>
	5.02	Bad Gateway	<a href="#">[RFC7252]</a>
	5.03	Service Unavailable	<a href="#">[RFC7252]</a>
	5.04	Gateway Timeout	<a href="#">[RFC7252]</a>
	5.05	Proxying Not Supported	<a href="#">[RFC7252]</a>