



Universidad
Nacional
de Quilmes

CONSTRUCCIÓN DE INTERFACES DE USUARIO

1do Cuatrimestre de 2019



Resumen

- Protocolo HTTP
- Aplicaciones Stateless vs Stateful

Protocolo HTTP

Hypertext Transfer Protocol

- Protocolo de Capa de aplicación (nivel 7 del Modelo OSI)
- Comunicación para la transferencia de información en la WWW.
- Sin estado, es decir, no almacena información sobre conexiones anteriores.
- Orientado al esquema Petición-Respuesta entre un cliente y un servidor

Protocolo HTTP



Protocolo HTTP

Algunos métodos de Petición

- GET: Solo para recuperar información, parámetros deben ser enviados por URL.
- POST: Envía datos para ser procesados, los datos deben ser enviados en el cuerpo (body) de la petición.
- PUT: Sube, carga o realizar un “upload” de un recurso específico (file)
- DELETE: Elimina un recurso específico

Protocolo HTTP

Códigos de Respuesta

- 200: OK (Petición correcta)
- 400: Bad Request (Petición incorrecta) el servidor no pudo interpretar la solicitud, por una sintaxis errónea.
- 404: Not Found (Recurso no encontrado)
- 500: internal Server Error (error no controlado) El servidor no puede controlar una excepción.

Protocolo HTTP

Códigos de Respuesta

1xx: Mensaje informativo.

3xx: Redirección 300
Multiple Choice

301 Moved Permanently
302 Found

304 Not Modified ?

2xx: Exito

200 OK

201 Created

202 Accepted

204 No Content

4xx: Error del cliente

400 Bad Request

401 Unauthorized

403 Forbidden

404 Not Found

5xx: Error del servidor

500 Internal Server Error

501 Not Implemented

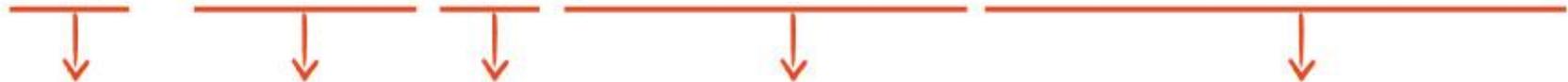
502 Bad Gateway

503 Service Unavailable

API REST

Estructura de URL

`http://localhost:4567/path/to/resource?att1=value1&att2=value2`



Esquema **Host** **Puerto** **Path** **Fragmento** **Query String**

`https://blog.makeitreal.camp/post1#titutlo2`



Protocolo HTTP - Estructura

Requests

```
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
```

```
-12656974
(more data)
```

Responses

```
HTTP/1.1 403 Forbidden
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
```

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

start-line

HTTP headers

empty line

body

Source: mozilla.org

API REST

Representational State Transfer

Cualquier interfaz entre sistemas que utilicen el protocolo HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON.

Stateless vs Stateful

Stateless - Sin estado

- Utilizado en Aplicaciones WEB
- El protocolo HTTP es sin estado.
- El servidor no almacena peticiones anteriores
- Tras responder una petición, se cierra la conexión
- El servidor toma cualquier petición como una nueva.
- El servidor no retiene información de sesión
- No hay necesidad de liberar recursos tomados
- Información adicional del lado del cliente para determinar información de usuario o sesión (Cookie o Token).

Aplicaciones Stateless vs Stateful

Stateful - CON estado

- Utilizado en aplicaciones Desktop o protocolo FTP.
- El servidor mantiene información de sesión, con el objetivo de representar flujos de trabajos y estados.
- NO se persiste el estado, no se mantienen al reiniciar los servidores. Información volátil
- Diferentes peticiones pueden mantener datos compartidos. Se mantiene información del cliente a lo largo de varias invocaciones.
- Es necesario predecir la capacidad de hardware
- Necesidad de liberar recursos ante el no uso prolongado.

Aplicaciones Stateless vs Stateful

Característica	Con Estado	Sin Estado
Mantiene información de sesión	SI	NO
Diferentes peticiones pueden compartir información	SI	NO
Mantiene información volátil	NO	SI
Contexto y tamaño predecible	NO	SI
Necesidad de liberación de recursos (garbage collector)	SI	NO
Necesidad de reconstrucción ante caídas	SI	NO

¿Preguntas?