



Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

HTTP

IIC2513 - Tecnologías y Aplicaciones Web

Sebastián Vicencio R.
2do Semestre 2020

Interacción cliente - servidor

CLIENTE



HTTP request



**INTERNET
(TCP/IP)**

HTTP response



SERVIDOR



HTTP

HyperText Transfer Protocol

**Protocolo que permite intercambio de
información en la Web**

Hipertexto

Acceso documentos mediante links

Link: texto que contiene referencia a otro documento



Hipertexto

Recursos

El motivo de la comunicación

“Cualquier cosa” que se quiera transmitir

Documento, foto, texto plano

Identificado por una **URI (o URL)**

HTTP: interfaz para interactuar con recursos

Recursos

Representación

Se transmite una representación del recurso



Recurso: **abstracto**
Representación: **concreto**

HTTP

Características principales

1. **Simple**: posible de ser leído por humanos
2. **Extensible**: información arbitraria por medio de headers
3. **Stateless**: dos requests consecutivos son independientes

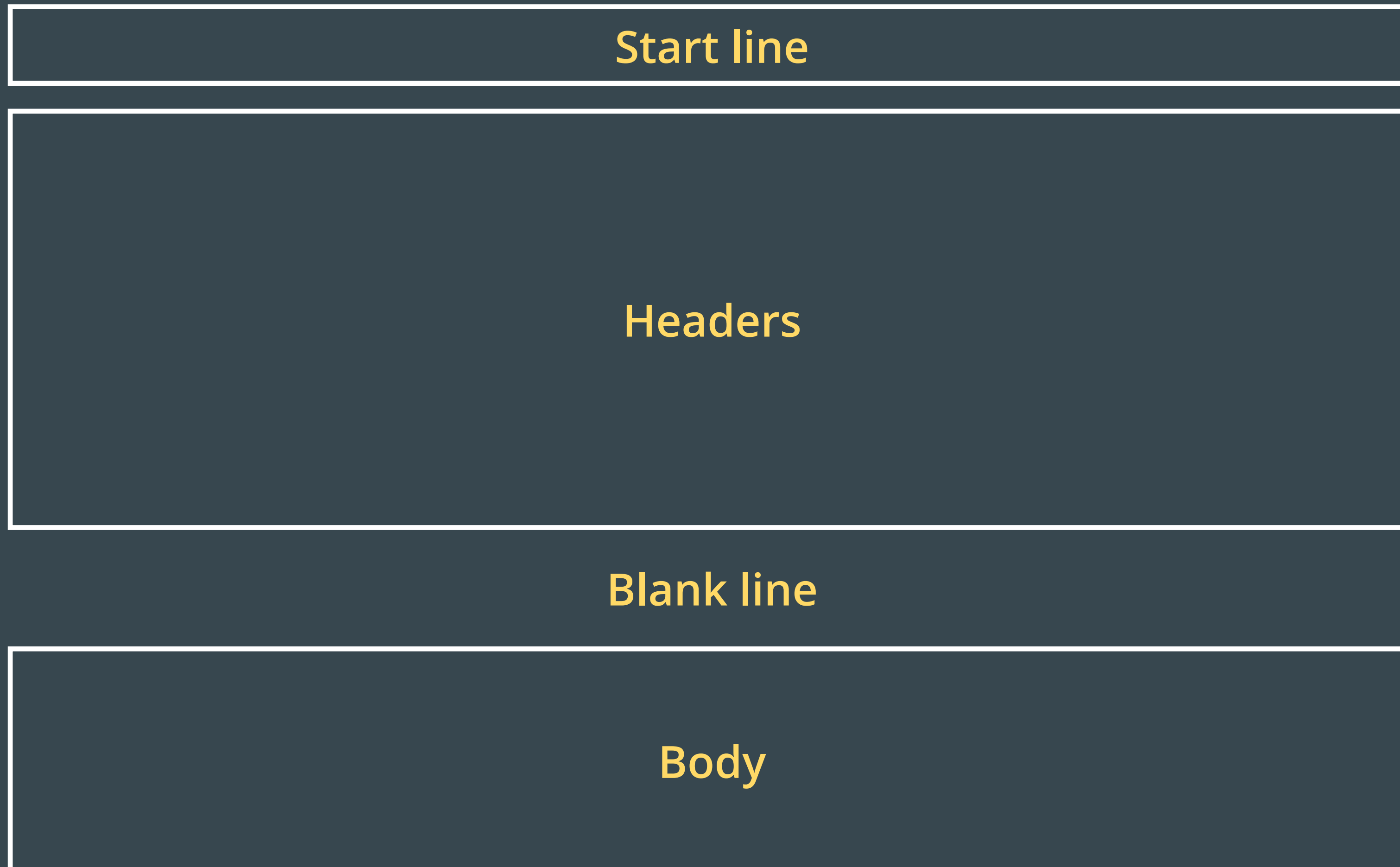
Mensajes HTTP

La base de la comunicación

Texto compuesto por múltiples líneas

Mensajes HTTP

Estructura general



Request HTTP

Mensaje del cliente

Request HTTP

Estructura

```
POST /users HTTP/1.1
Content-Length: 100
User-Agent: PostmanRuntime/7.25.0
Accept: */*
Host: jsonplaceholder.typicode.com
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Type: application/json
Cookie: __cfduid=d987500a78f9b8d599037b4

{
  "name": "Sebastián Vicencio",
  "username": "sivicencio",
  "email": "sivicencio@uc.cl"
}
```

← Start line

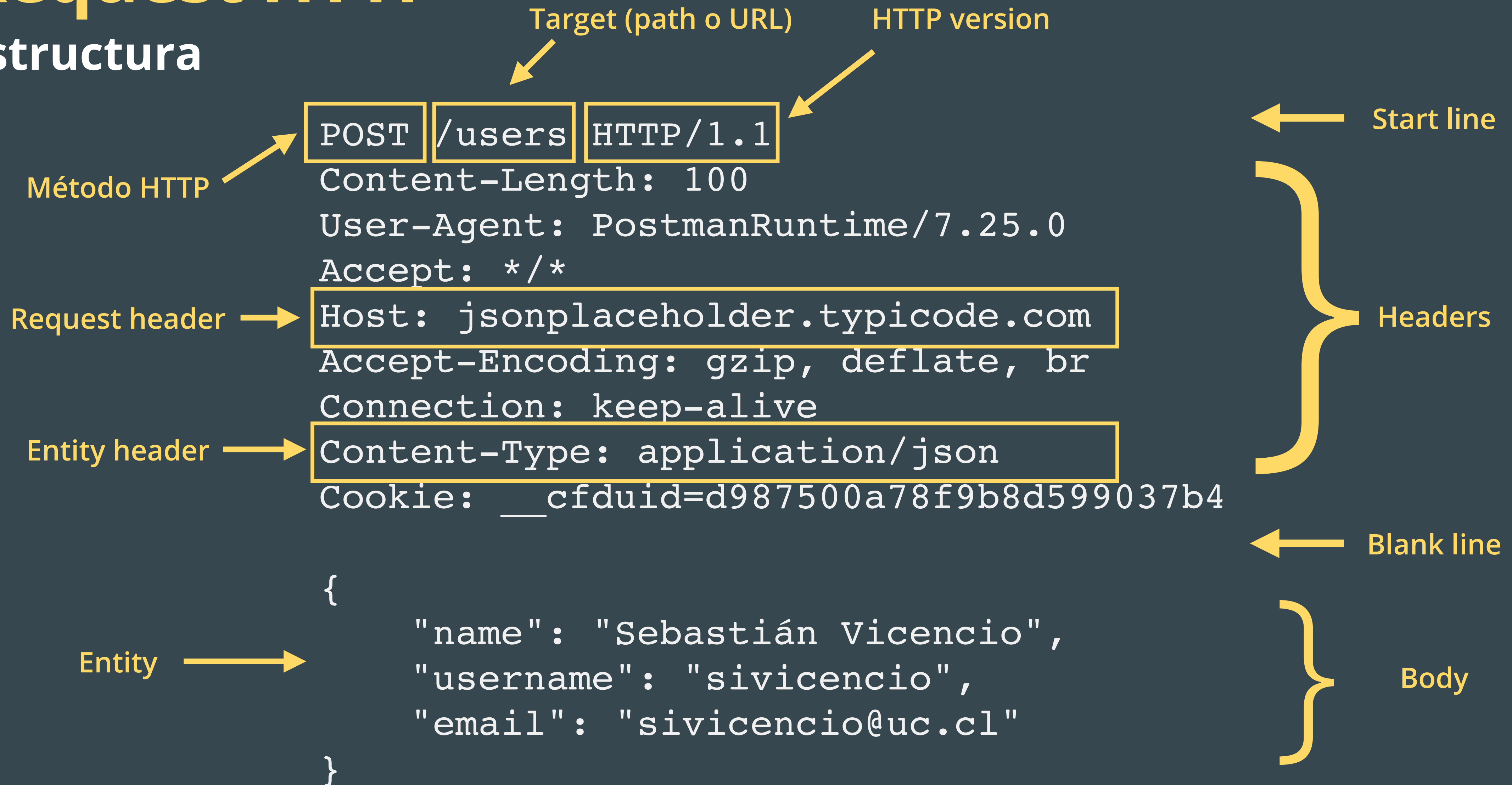
} Headers

← Blank line

} Body

Request HTTP

Estructura



Métodos HTTP

Propósito del request

Acción que cliente espera que
servidor realice

Métodos HTTP

- **GET**: obtener representación de un recurso específico
- **POST**: crear nuevo recurso en base a entidad adjunta
- **PUT**: reemplazar recurso específico con entidad adjunta
- **PATCH**: modificar recurso específico en base a entidad adjunta
- **DELETE**: eliminar recurso específico

Response HTTP

Mensaje del servidor

Response HTTP

Estructura

```
HTTP/1.1 201 Created
Date: Thu, 03 Sep 2020 06:13:27 GMT
Content-Length: 106
Content-Type: application/json; charset=utf-8
Connection: keep-alive
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Access-Control-Expose-Headers: Location
Location: http://jsonplaceholder.typicode.com/users/11
X-Content-Type-Options: nosniff
Etag: W/"6a-MIZeGYXbkV9UzaMzb/OuKeiqb00"
Via: 1.1 vegur
Server: cloudflare
```

```
{
  "name": "Sebastián Vicencio",
  "username": "sivicencio",
  "email": "sivicencio@uc.cl",
  "id": 11
}
```

← Start line

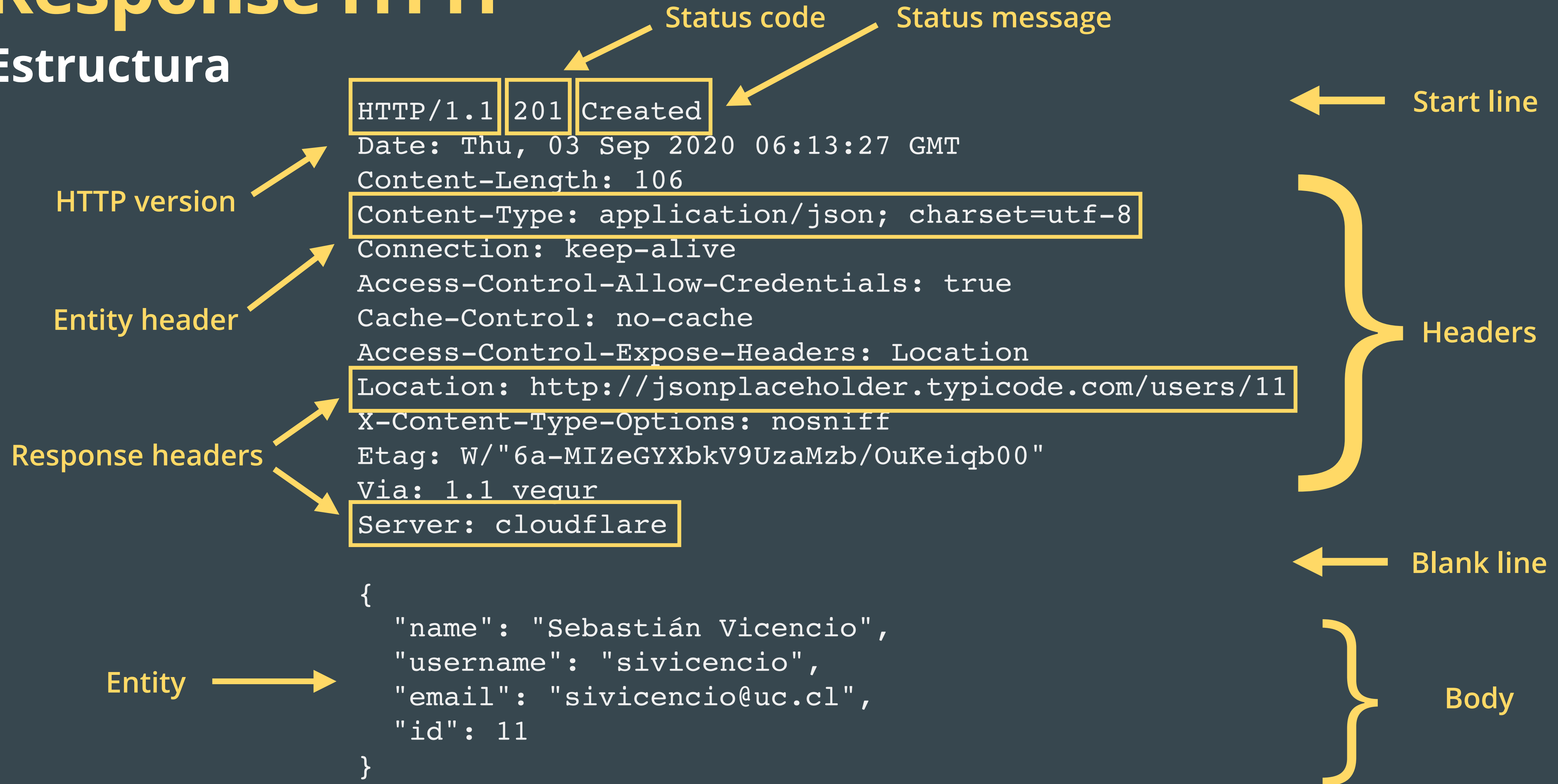
} Headers

← Blank line

} Body

Response HTTP

Estructura



Status codes

Tipo de respuesta del servidor

3 dígitos

Status codes

- **1xx: informativo** → Request recibido y continuando proceso
- **2xx: éxito** → Request recibido, entendido y aceptado
- **3xx: redirección** → Acciones necesarias para completar request
- **4xx: error de cliente** → Request con sintaxis errónea o no válido
- **5xx: error de servidor** → Servidor falló en completar request

URI

Texto que permite **identificar** de
forma única un **recurso web**

URL

Tipo de URI que incorpora
información de la **ubicación del**
recurso web

URI

Estructura

`http://example.org/general/specific?q=keyword&page=2#section`



SCHEME

Protocolo



AUTHORITY

Host



PATH

Secuencia de segmentos
(jerárquicos)



QUERY

Parámetros
key=value



FRAGMENT

Sección recurso

Paths

Convención para orden jerárquico

/users → Colección

/users/:id → Un elemento de la colección

/users/:id/posts → Colección como sub-recurso

/users/:userId/posts/:id → Sub-recurso

Path + método HTTP

Acciones sobre recursos

GET	/users
POST	/users
GET	/users/2
PUT	/users/2
PATCH	/users/2
DELETE	/users/2

Ejemplo

Request - response

Actividad

Hands-on: Requests en JS