



Comunicação Servidor

Prof. Victor Farias

V 1.4

JSON

JSON

- JSON é um estrutura de dados derivada da notação de objetos do JS
- Sintaxe:
 - Os dados estão dispostos em pares nome/valor
 - Os dados são separados por vírgula
 - Os objetos são colocados em chaves
 - Listas são colocadas entre colchetes

JSON Exemplos

- Exemplo par nome/valor: "firstName":"John"
 - Nomes JSON requerem aspas duplas
 - Valores JSON podem ser números, strings, booleanos, listas, objetos ou null
- Exemplo objetos: {"firstName":"John", "lastName":"Doe"}
- Exemplo lista:
- "employees":[
 - {"firstName":"John", "lastName":"Doe"},
 - {"firstName":"Anna", "lastName":"Smith"},
 - {"firstName":"Peter","lastName":"Jones"}
-]

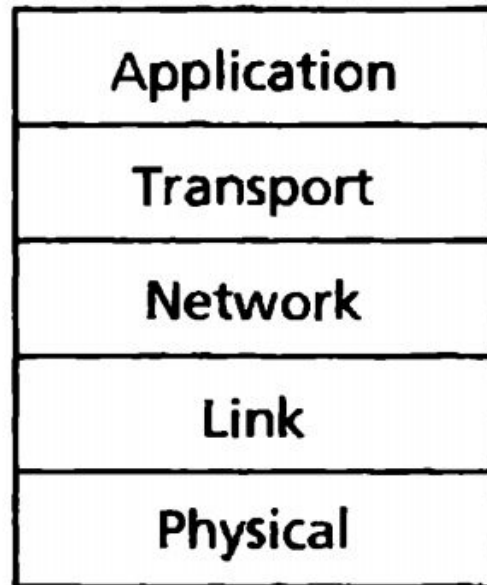
JSON - Como usar

- JSON é muito usado para ler dados do servidor e mostrar na página web
- Texto JSON para JS Object:
 - `var text = '{ "employees" : [' +`
 - `'{ "firstName":"John" , "lastName":"Doe" },' +`
 - `'{ "firstName":"Anna" , "lastName":"Smith" },' +`
 - `'{ "firstName":"Peter" , "lastName":"Jones" }]}';`
 - `var obj = JSON.parse(text); // Parsing JSON texto to JavaScript Object`
- JS Object para Texto JSON:
 - `var obj = [{a:1,b:1,c:1}, {a:2,b:2,c:3}]`
 - `var json = JSON.stringify(obj);`

Protocollo HTTP



Pilha de Protocolos na Internet



5 camadas

Camada	Protocolo
5.Aplicação	HTTP, SMTP, FTP, SSH, RTP, Telnet, SIP, RDP, IRC, SNMP, NNTP, POP3, IMAP, BitTorrent, DNS, Ping ...
4.Transporte	TCP, UDP, SCTP, DCCP ...
3.Redes	IP (IPv4, IPv6) , ARP, RARP, ICMP, IPsec ...
2.Enlace	Ethernet, 802.11 WiFi, IEEE 802.1Q, 802.11g, HDLC, Token ring, FDDI, PPP, Frame Relay,
1.Física	Modem, RDIS, RS-232, EIA-422, RS-449, Bluetooth, USB, ...

Endereço IP e Portas

- **Endereço IP**

- Identifica um host na rede
- Cada interface de rede tem um IP
- ex: 200.21.32.43
- URLs são traduzidos em IP usando DNS (globo.com.br -> 186.192.90.5)

- **Portas**

- Identificam os processos de origem e fim
- Permite a comunicação de diversas aplicações na mesma máquina
- Cada aplicação recebe e envia requisições por uma porta
- ex: Servidor Web, por padrão, recebem requisições na porta 80

Arquitetura Cliente Servidor

- **Servidor**

- Aplicação que fornece serviço
- Aceita requisições através da rede, em um porta conhecida, e retorna o resultado

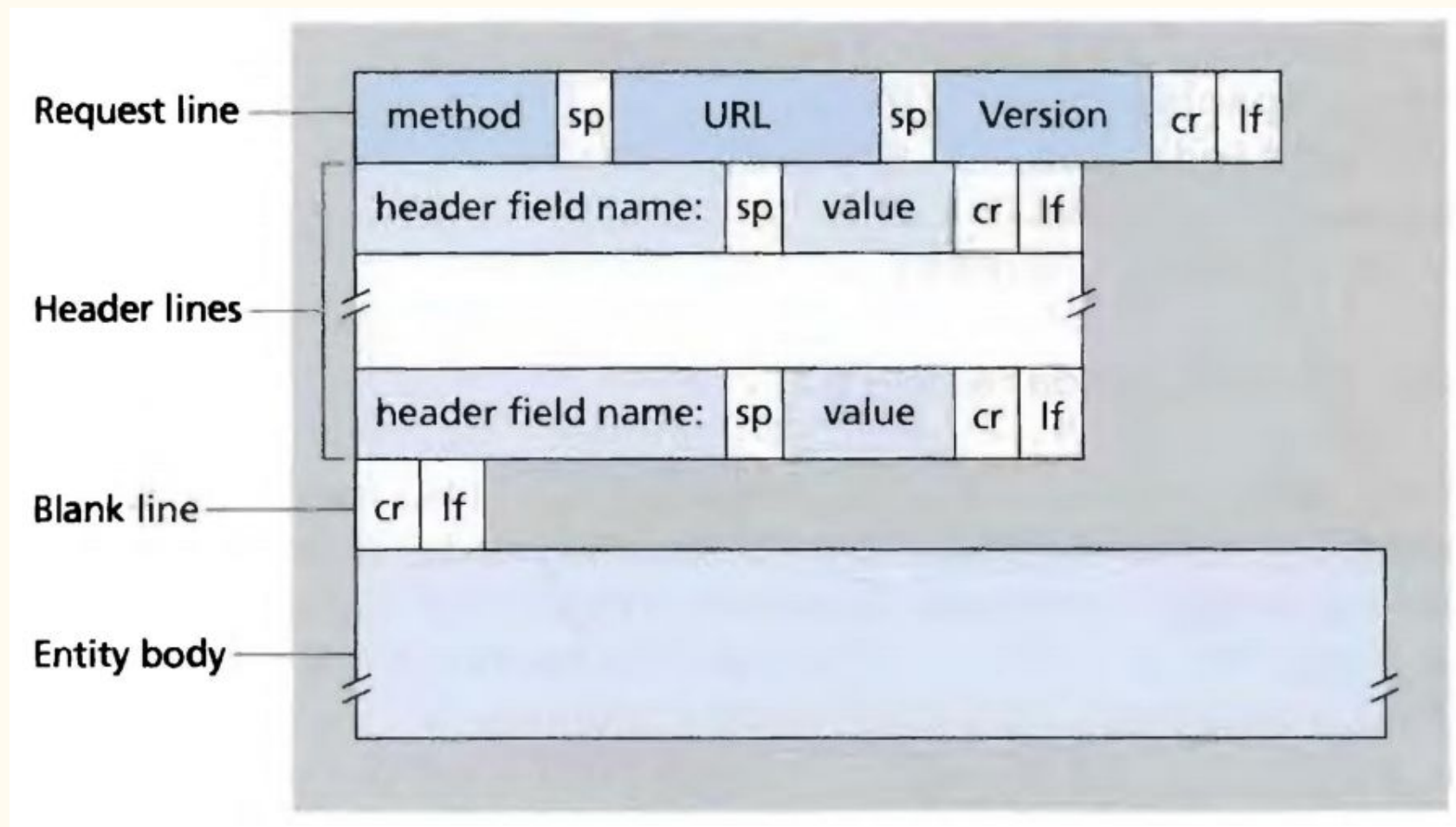
- **Cliente**

- Processo que requisita um serviço
- Para receber resposta, o cliente aloca um porta arbitrária

Protocolo HTTP

- **HTTP = Hypertext Transfer Protocol ou Protocolo de Transferência de Hipertexto**
- **Protocolo usado para transferir dados na WEB**
- **Funcionamento:**
 - O cliente envia uma requisição HTTP para o servidor
 - O servidor envia uma resposta HTTP ao cliente

Composição da Requisição HTTP



Métodos de Requisição

- **Métodos usados em requisições HTTP**

- **GET – Solicita algum recurso**

- Dados são anexados à URL, ficando visíveis ao usuário

- **POST – Envia dados referentes ao recurso especificado para serem processados**

- Dados são incluídos no corpo do comando

- **PUT – Envia certo recurso**

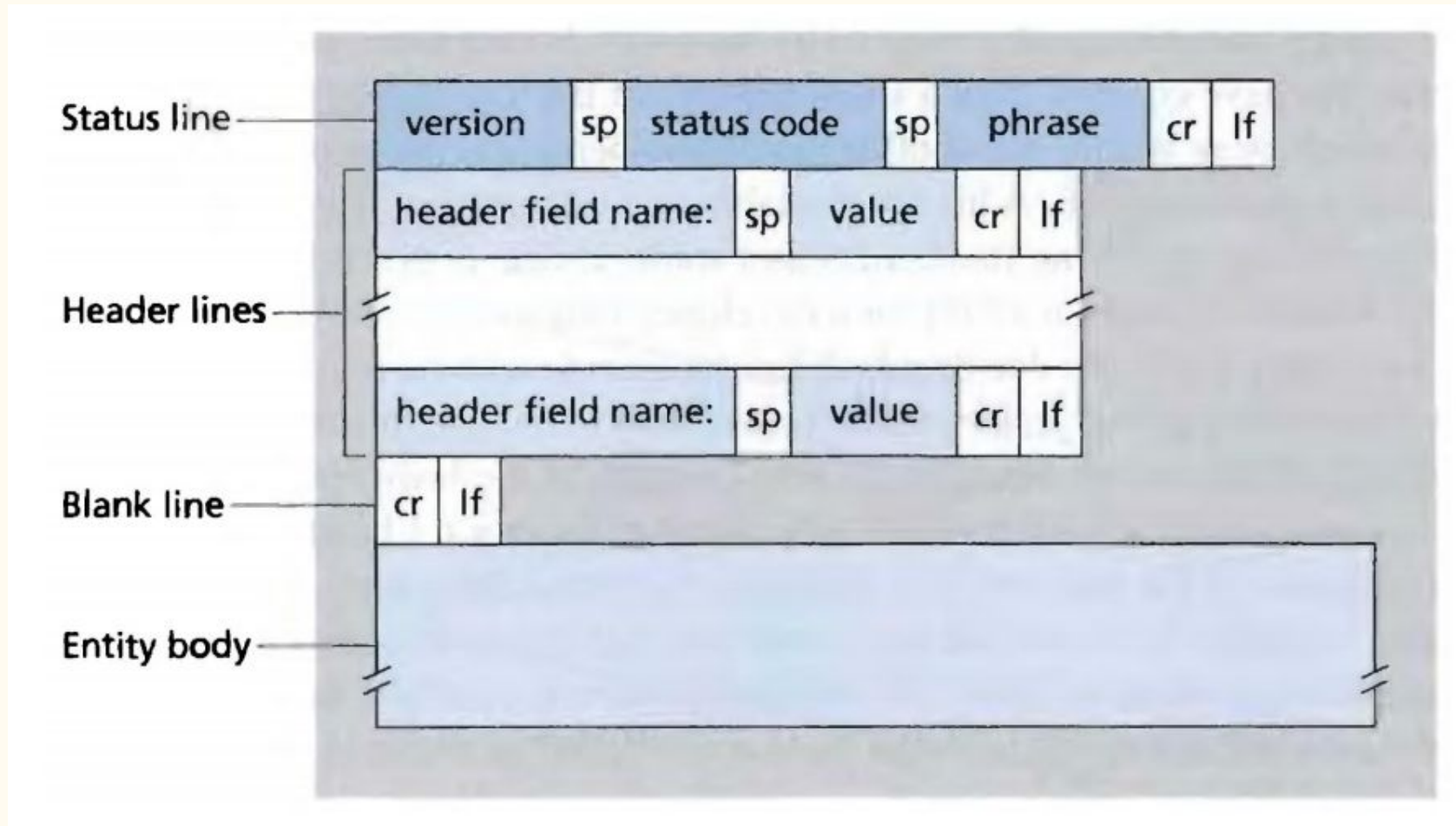
- Em geral, é usado para atualizar dados

- **DELETE – Exclui o recurso**

- **HEAD – Variação do GET em que o recurso não é retornado**

- Usado para obter metadados por meio do cabeçalho da resposta, sem ter que recuperar todo o conteúdo.

Resposta HTTP



Código de Status

- **O código de status é formado por três dígitos e o primeiro dígito representa a classe que pertence classificada em cinco tipos:**
 - **1xx: Informational (Informação)** – utilizada para enviar informações para o cliente de que sua requisição foi recebida e está sendo processada
 - **2xx: Success (Sucesso)** – indica que a requisição do cliente foi bem sucedida
 - **3xx: Redirection (Redirecionamento)** – informa a ação adicional que deve ser tomada para completar a requisição
 - **4xx: Client Error (Erro no cliente)** – avisa que o cliente fez uma requisição que não pode ser atendida
 - **5xx: Server Error (Erro no servidor)** – ocorreu um erro no servidor ao cumprir uma requisição válida
- O protocolo HTTP define somente alguns códigos em cada classe, mas cada servidor pode definir seus próprios códigos

REST



REST

- **REST = Representational State Transfer**
 - Abstração de arquitetura de um componente
 - Define forma de se consumir um dado recurso
 - Ignora implementação do componente
 - Foca na sintaxe de comunicação do componente
- **Usa protocolo HTTP**
 - Usa métodos/verbo (GET, POST, PUT ...) para indicar ação
 - URL define a estrutura do serviço
- **Usaremos JSON como formato de dados**

Padrão REST - Verbos

- Verbos/métodos:
 - **GET**
 - Recuperar itens
 - **POST**
 - Adicionar item
 - **PUT**
 - Atualizar item
 - **DELETE**
 - Deletar item

Padrão REST - URL

- Exemplo: <http://rest.learncode.academy/>
- **`http://rest.learncode.academy/api/ufc/alunos` (GET)**
 - Recuperar JSON com todos os alunos da ufc
- **`http://rest.learncode.academy/api/ufc/alunos/57846` (GET)**
 - Recuperar JSON do aluno com id 57846
- **`http://rest.learncode.academy/api/ufc/alunos` (POST)**
 - Inserir novo aluno na coleção de alunos
 - Cliente envia JSON com dados do novo aluno

Padrão REST - URL

- **`http://rest.learncode.academy/api/ufc/alunos/57846` (PUT)**
 - Atualizar dados do aluno com id 57846
 - Cliente envia JSON com os dados do aluno a ser modificado
- **`http://rest.learncode.academy/api/ufc/alunos/57846` (DELETE)**
 - Deleta aluno com id 57846

HTTP

React - Axios

Instalação

```
npm install axios --save
```

Sintaxe básica

```
axios({  
  method: 'POST',  
  url: "http://rest.learncode.academy/api/victor/alunos/",  
  data: {nome: "victor"}  
}).then((response) => {  
  // Tratar resposta  
}).catch((error) => {  
  // Tratar erro  
})
```

Método

URL

Payload

- Função do **then** é executada quando requisição terminar.
- Função recebe objeto **response** que representa a resposta da requisição
- Função é executada de forma assíncrona!

Formato do response:

{data: {...}, status: 200, statusText: "OK", headers: {...}, config: {...}, ...}

CORS



Abrir chrome desabilitando o cors

- windows + r
- `chrome.exe --user-data-dir="C:/Chrome dev session" --disable-web-security`

Perguntas?

Prof. Victor Farias