



Programação Web III

Luigi Ciambarella Filho



Conhecendo a turma

<https://www.menti.com/al8rmdezniev>





Acordos

Conteúdo do módulo

O que vamos Aprender?

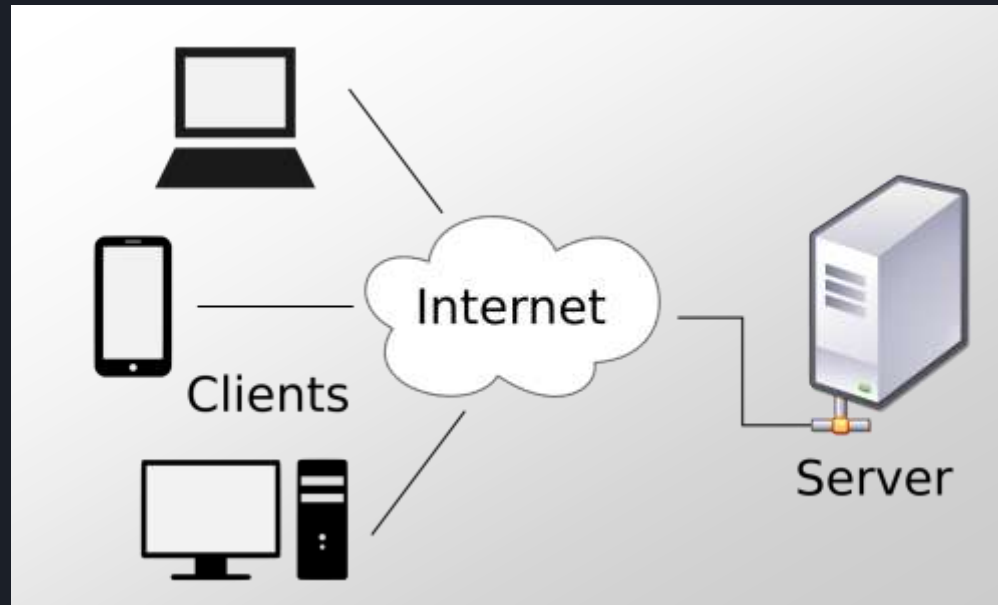
- REST
- API Controllers
- Filtros
- Controle de Exceções
- Autenticação e Autorização (JWT)

Aula 1 - Conteúdo da aula

Objetivos do Curso

- Dominar os conceitos básicos de APIs Web
- Entender o padrão REST e como aplicá-lo na construção de APIs RESTful
- Desenvolver a habilidade de tratar erros em APIs utilizando boas práticas
- Entender os padrões para Documentação de APIs (OAS - OpenApi Specification) e como adicioná-lo em uma API Web em ASP.NET Core
- Conhecer os principais riscos de segurança em APIs Web (OWASP - OWASP API Security Project) e dominar os princípios básicos de controle de acesso a APIs

HTTP: Hypertext Transfer Protocol

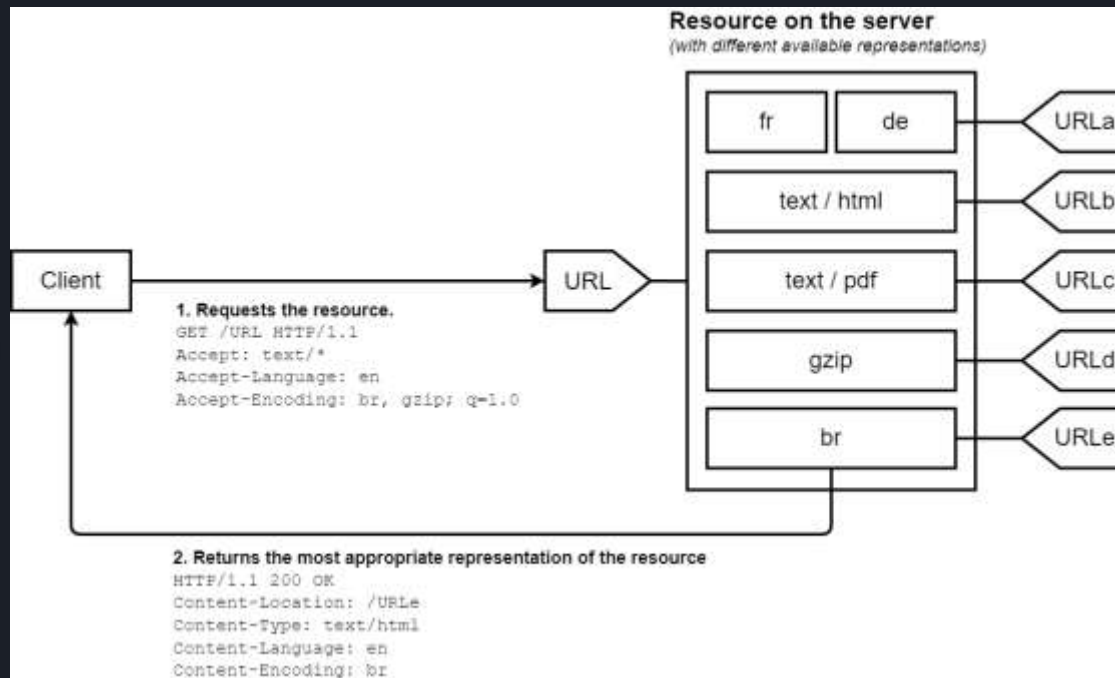


Revisão

URL: Uniform Resource Locator



HTTP - Negociação de Conteúdo



HTTP - Requisição HTTP

`POST /?id=1 HTTP/1.1` **Request line**

`Host: www.swingvy.com
Content-Type: application/json; charset=utf-8
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.12; rv:53.0)
Gecko/20100101 Firefox/53.0
Connection: close
Content-Length: 136`

Header

```
{  
  "status": "ok",  
  "extended": true,  
  "results": [  
    {"value": 0, "type": "int64"},  
    {"value": 1.0e+3, "type": "decimal"}  
  ]  
}
```

Body message

HTTP - Resposta HTTP

protocol **status code**

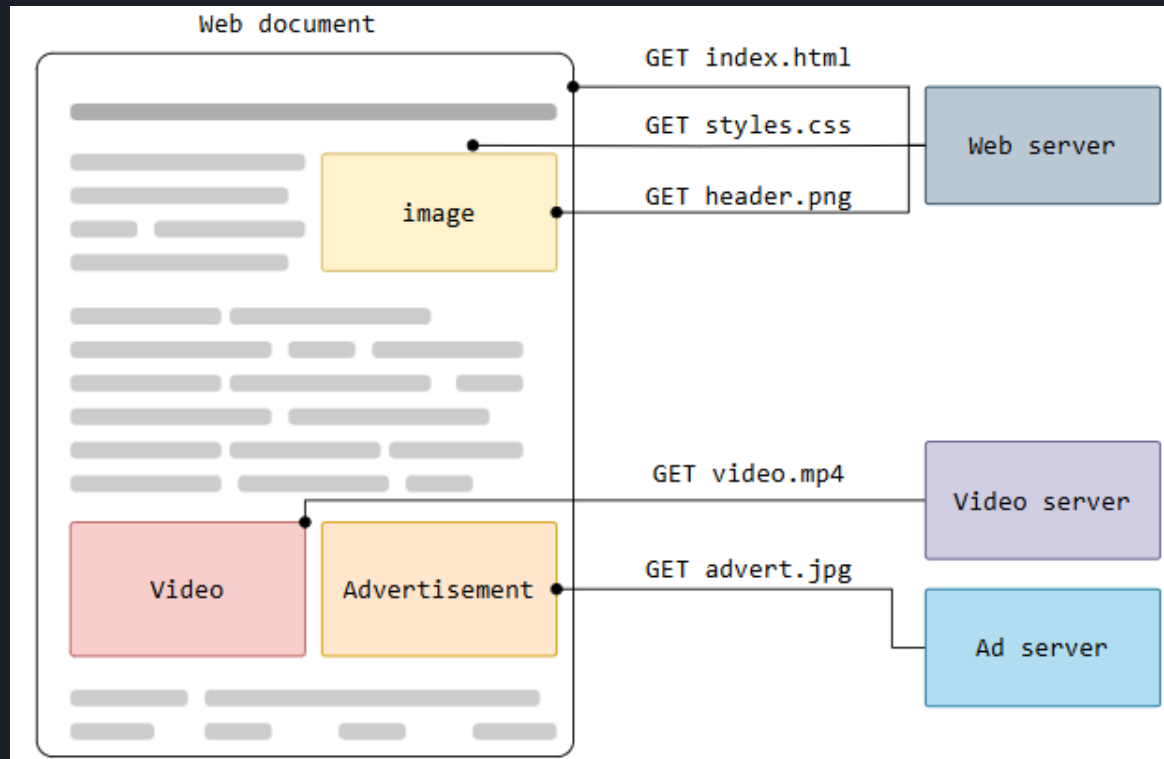
HTTP/1.x 200 OK

Transfer-Encoding: chunked
Date: Sat, 28 Nov 2009 04:36:25 GMT
Server: LiteSpeed
Connection: close
X-Powered-By: W3 Total Cache/0.8
Pragma: public
Expires: Sat, 28 Nov 2009 05:36:25 GMT
Etag: "pub1259380237;gz"
Cache-Control: max-age=3600, public
Content-Type: text/html; charset=UTF-8
Last-Modified: Sat, 28 Nov 2009 03:50:37 GMT
X-Pingback: <http://net.tutsplus.com/xmlrpc.php>
Content-Encoding: gzip
Vary: Accept-Encoding, Cookie, User-Agent

HTTP headers as Name: Value

Revisão

HTTP - Recursos



Revisão

Métodos de requisição HTTP:

Método	Corpo da requisição	Corpo da resposta
GET	Não Presente (Opcional)	Presente
HEAD	Não Presente (Opcional)	Ausente
POST	Presente	Presente
PUT	Presente	Presente
DELETE	Não Presente (Opcional)	Presente
CONNECT	Não Presente (Opcional)	Presente
OPTIONS	Não Presente (Opcional)	Presente
TRACE	Ausente	Presente
PATCH	Presente	Presente

Revisão

Código de estado da resposta HTTP:

Código	Utilidade
1xx	Informação
2xx	Sucesso
3xx	Redirecionamento
4xx	Erro do Cliente
5xx	Erro no Servidor

Revisão

API: Application Programming Interface (Web)

Tipos	
SOAP	S imple O bject A ccess P rotocol
REST	R epresentational S tate T ransfer
GraphQL	GraphQL
gRPC	G oogle R emote P rocedure C alls

REST - Representational State Transfer

O que é?

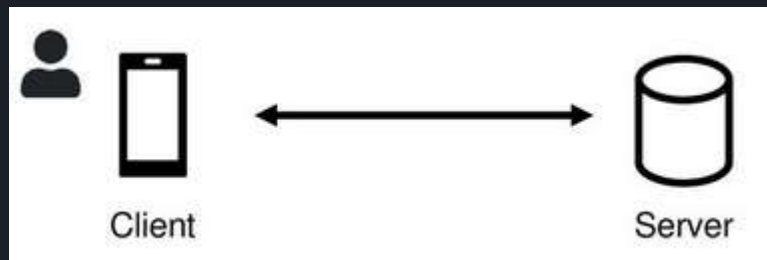
REST é um estilo de arquitetura de software que estabelece regras e convenções para criação de Web API

REST foi criado por Roy Fielding em sua tese de doutorado em 2000. Como toda idéia teórica existem muitas praticabilidades. Mas os princípios de uma API REST tem a intenção de criar API e sistemas que são: Eficiente, Escalável, Simples, Confiável, etc.

<https://ics.uci.edu/~fielding/pubs/dissertation/top.htm>

6 Regras do REST

Regra 1: Arquitetura Servidor-Cliente

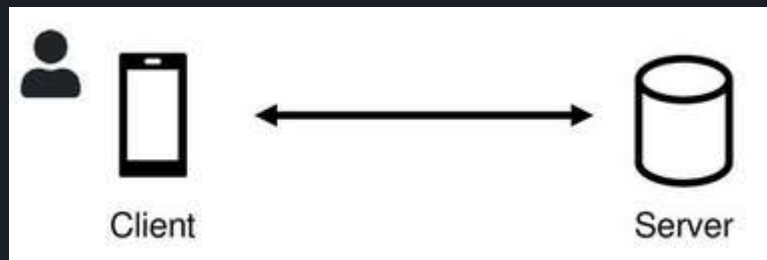


Um sistema RESTful separa as responsabilidades de armazenar e processar os dados (Servidor) e de requerer, consumir e apresentar os dados para o usuário.

Uma atualização no cliente não requer uma atualização no servidor e vice versa.

6 Regras do REST

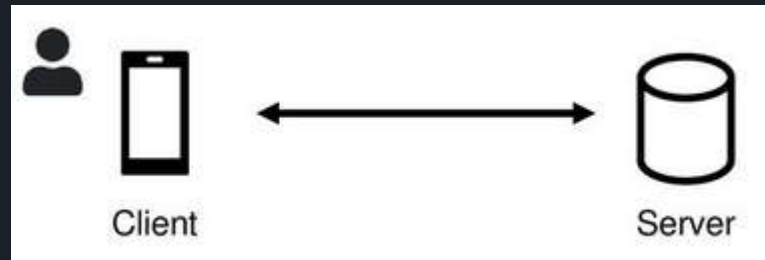
Regra 2: Comunicação sem estado



Cada solicitação de um cliente a um servidor deve conter todas as informações necessárias para que o servidor execute a operação solicitada, sem depender de informações armazenadas de interações anteriores.

6 Regras do REST

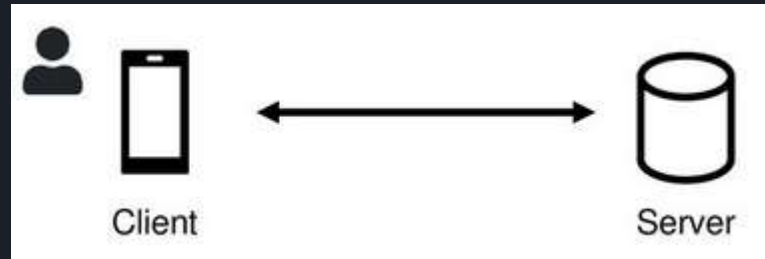
Regra 3: Capacidade de cache



As APIs REST devem facilitar o armazenamento em cache de respostas para melhorar o desempenho. Ao armazenar dados de resposta em cache, os clientes podem reduzir a latência de solicitações subsequentes, minimizar a carga nos servidores e diminuir o tráfego na rede.

6 Regras do REST

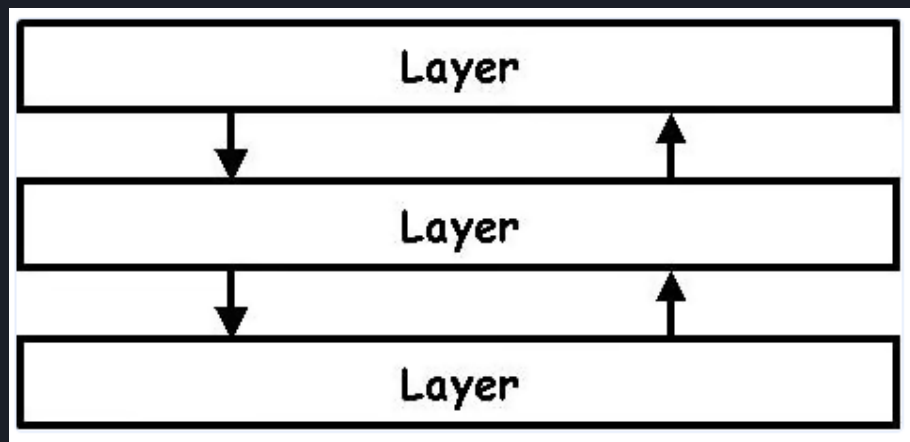
Regra 4: Uso de interface uniforme



Um dos aspectos cruciais do design da API RESTful é a adesão a uma interface uniforme. Isso envolve o uso de convenções consistentes e métodos HTTP padrão para processar solicitações de API

6 Regras do REST

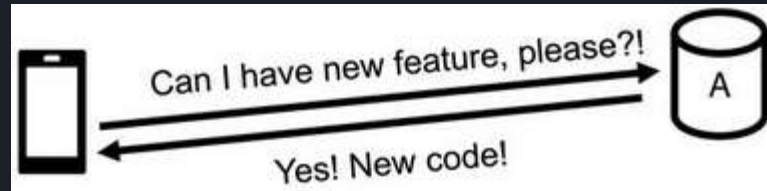
Regra 5: Sistema em camadas



A arquitetura de sistema em camadas separa as preocupações em diferentes camadas, como interface do usuário, lógica de negócios e camadas de acesso a dados em um aplicativo Web típico de n camadas.

6 Regras do REST

Regra 6: Suporte para código sob demanda



Code-on-Demand é uma restrição de APIs REST, permitindo que os servidores forneçam lógica de aplicativo para executar ações específicas em recursos. Embora nem sempre aplicável, permite maior flexibilidade e extensibilidade em determinados cenários.

REST

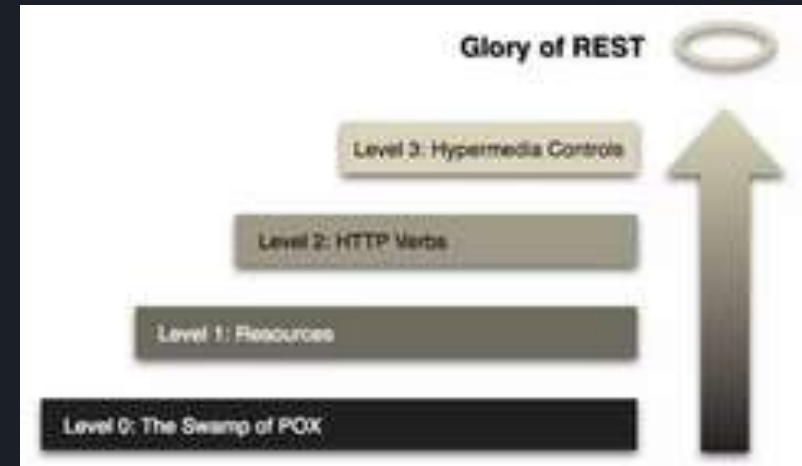
Modelo de Maturidade Richardson

Nível 0 - O Pântano da POX (XML Simples e Antigo)

Nível 1 – Recursos

Nível 2 - Verbos HTTP

Nível 3 - Controles de Hipermídia (HATEOAS (Hypermedia As The Engine Of Application State))



REST

Outros Modelos de Maturidade

Classification of HTTP-based APIs

(http://algermissen.io/classification_of_http_apis.html)

W S3 maturity model

<https://www.menti.com/al8rmdezniev>



Obrig.ada