Desenvolvimento WEB

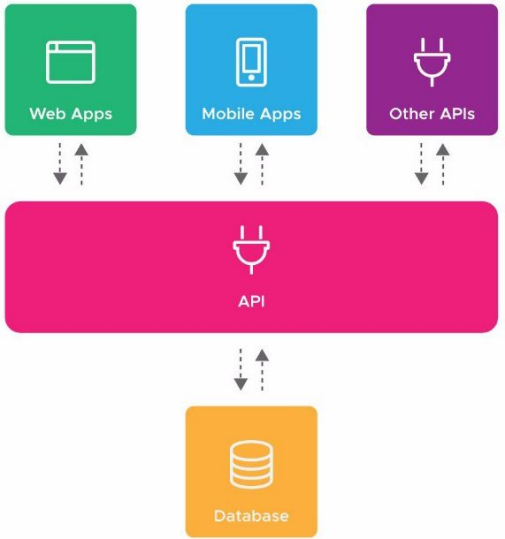
-

Full Stack Completo: Java + React

Prof. Roberto Alencar

{ REST:API }

APIs REST



Conceitos de Sistemas WEB

APIs

API ou *Application Programming Interface*, que em português quer dizer Interface de Programação de Aplicações, é um conjunto de funções e procedimentos que permitem a integração de sistemas, permitindo a reutilização das suas funcionalidades por outras aplicações ou software.

- Uma API é uma espécie de ponte que liga diferentes tipos de software ou aplicações e pode ser criada em várias linguagens de programação. Para além de um bom desenvolvimento, uma API deve ter uma documentação clara e objetiva para facilitar a sua implementação.

- As APIs são os blocos de construção que permitem a interoperabilidade para as principais plataformas de negócios na web. As APIs podem ser pensadas como colas digitais entre os backends das organizações e interfaces diversas (móveis, desktop e até mesmo outros fornecedores).

Conceitos de Sistemas WEB

APIs

- Existem basicamente quatro tipos de API considerando suas políticas de compartilhamento:

- **(1) APIs públicas ou abertas**
    - As APIs públicas são também conhecidas como APIs abertas e estão disponíveis para outros usuários ou programadores utilizarem com restrições mínimas ou, em alguns casos, de forma totalmente acessível.

- **(2) APIs privadas ou internas**
    - As APIs privadas ou internas são ocultadas aos usuários externos e são expostas apenas para os sistemas internos de uma organização. São utilizadas para o desenvolvimento interno da empresa, otimizando a produtividade e a reutilização dos serviços.

Conceitos de Sistemas WEB

APIs

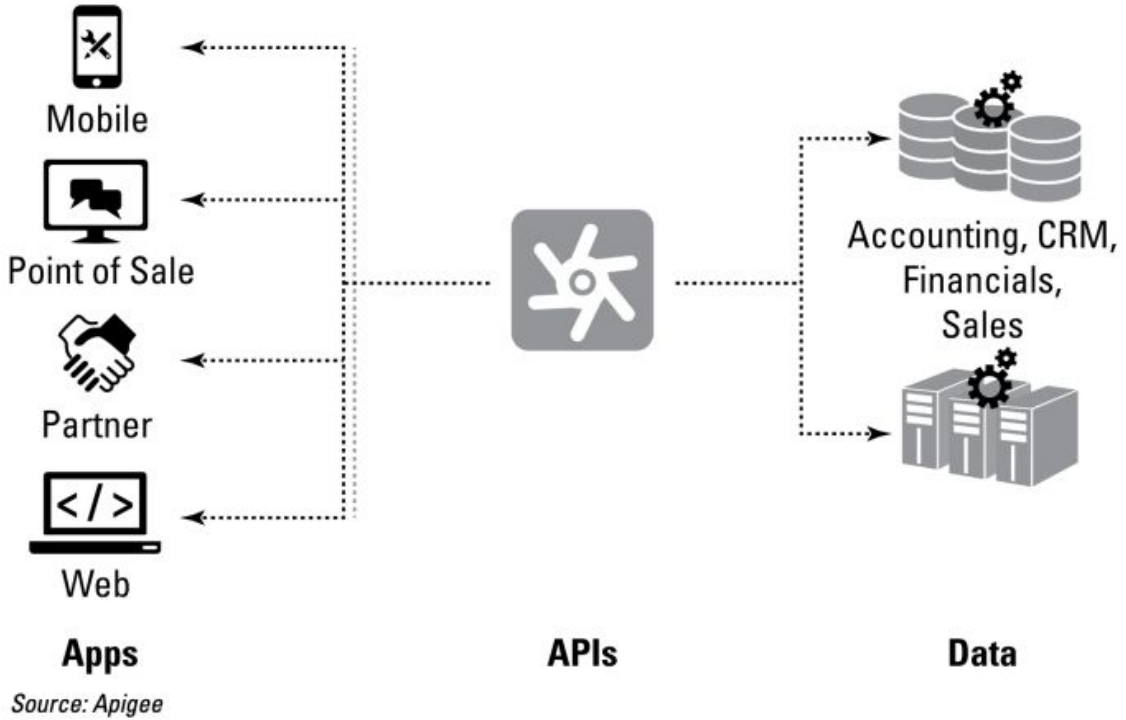
- Existem basicamente quatro tipos de API considerando suas políticas de compartilhamento:

- **(3) APIs de parceiros de negócio**
  - As APIs de parceiros comerciais são aquelas que são expostas entre os membros de um acordo comercial. Uma vez que não estão disponíveis para todos, é necessária uma autorização especial para sua utilização.

- **(4) APIs compostas**
  - Como o nome sugere, as APIs compostas combinam dois ou mais tipos de APIs, com o objetivo de criar uma sequência de operações relacionadas ou interdependentes.

Conceitos de Sistemas WEB

APIs



Conceitos de Sistemas WEB

Protocolos de API

- Os protocolos de API **permitem padronizar a troca de dados entre diferentes serviços web**. Isto proporciona a oportunidade de acessar capacidades em diferentes sistemas, por meio de diferentes linguagens de programação e em diferentes sistemas operacionais.

- Os mais importantes são:
  - Remote Procedure Call (RPC)
  - Service Object Access Protocol (SOAP)
  - Representational State Transfer (REST)

Conceitos de Sistemas WEB

Protocolos de API

- Remote Procedure Call (**RPC**)
  - O chamado de procedimento remoto ou RPC permite que as APIs web possam se aderir aos princípios de intercâmbio de recursos. O objetivo deste protocolo é definir a interação entre aplicações com base num programa que solicita dados - cliente - e outro que os fornece - servidor - remotamente.

- Service Object Access Protocol (**SOAP**)
  - É um protocolo verdadeiramente leve para o intercâmbio de informação estruturada num ambiente descentralizado e distribuído. As suas especificações contêm as regras de sintaxe dos pedidos e respostas enviadas pelos pedidos.
  - As aplicações que cumprem estes princípios permitem o envio de mensagens XML entre o sistema via HTTP (*Hypertext Transfer Protocol*) ou SMTP (*Simple Mail Transfer Protocol*).

Conceitos de Sistemas WEB

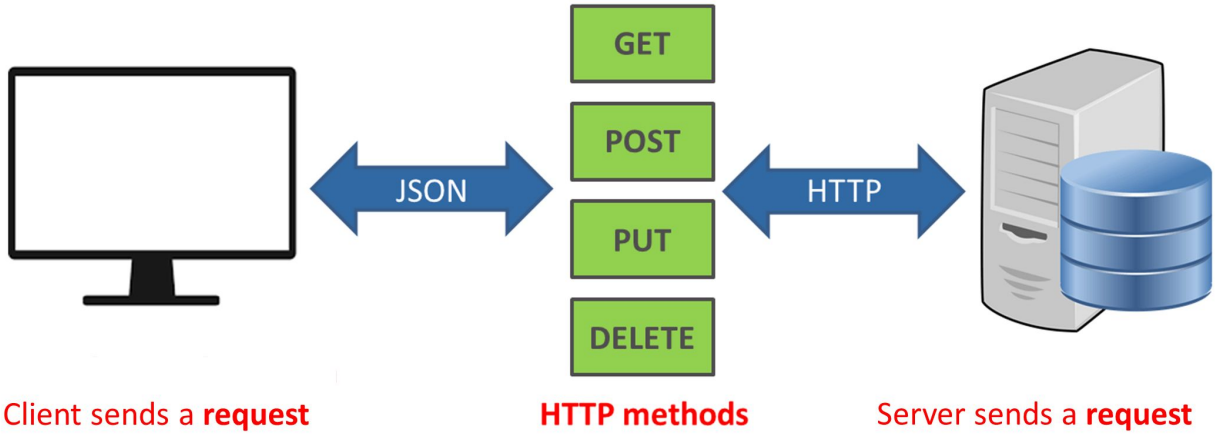
Protocolos de API

- Representational State Transfer (**REST**)
    - REST é um estilo de arquitetura de software com seis restrições para a criação de aplicações que correm sobre HTTP, especialmente serviços web.
    - É considerada uma alternativa SOAP, uma vez que muitos programadores têm dificuldade em utilizá-la porque têm de escrever grandes quantidades de código para realizar uma tarefa. Por outro lado, a REST segue uma lógica diferente, uma vez que facilita a disponibilidade de dados como recursos.

Conceitos de Sistemas WEB

REST

- *Representational State Transfer*, abreviado como REST, não é uma tecnologia, uma biblioteca, e nem tampouco uma arquitetura, mas sim um modelo a ser utilizado para se projetar arquiteturas de software distribuído, baseadas em comunicação via rede.



Conceitos de Sistemas WEB

API Web RESTful

- Muitas empresas foram desafiadas a expandir a sua presença Web em outros meios. O movimento de aplicações móveis é uma dessas forças.

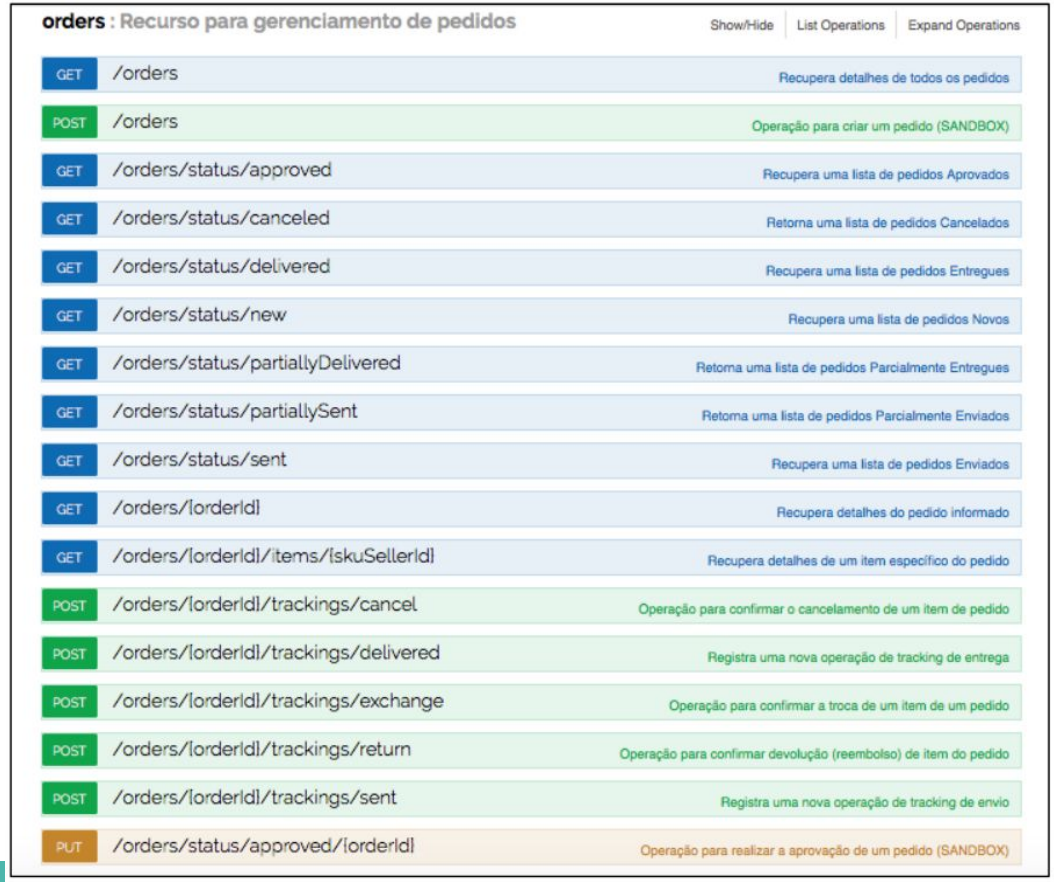
- Com o movimento da internet das coisas (IoT), essa pressão continuará a aumentar nos próximos anos.

- Nesse sentido, algumas empresas têm realizado uma migração digital de portais Web para uma presença mais ampla. Isso é acompanhado através da criação de uma API de serviços Web.

Conceitos de Sistemas WEB

API Web RESTful

Esta API apresenta funcionalidades expostas através de métodos HTTP tais como GET, POST ou PUT e pode ser consumida em qualquer linguagem moderna, seja por aplicativos Web, dispositivos móveis ou dispositivos da Internet das coisas.

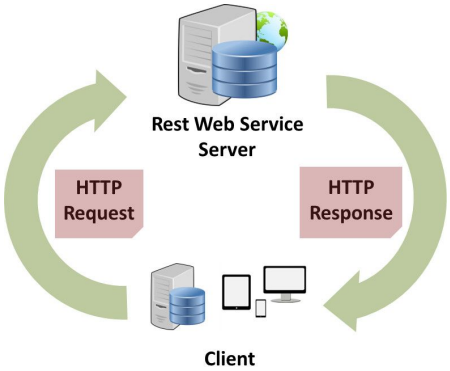


Conceitos de Sistemas WEB

REST

- O termo REST (*Representational State Transfer*) foi cunhado por Roy Fielding em 2000 na sua tese de doutorado.

- O autor, que foi um dos autores do protocolo HTTP e também cocriador do servidor Web Apache, criou um conjunto de princípios para orientar o bom desenho de sistemas distribuídos tais como performance, simplicidade, escalabilidade, portabilidade, confiabilidade e modificabilidade.



Conceitos de Sistemas WEB

REST

- A identificação do recurso deve ser feita utilizando-se o conceito de URI (*Uniform Resource Identifier*):
    - http://servicorest.com.br/produtos
    - http://servicorest.com.br/clientes
    - http://servicorest.com.br/clientes/57
    - http://servicorest.com.br/vendas

Conceitos de Sistemas WEB

REST

- Vejamos agora os principais métodos do protocolo HTTP e o cenário de utilização de cada um deles:

GET	Obter os dados de um recurso.
POST	Criar um novo recurso.
PUT	Substituir os dados de um determinado recurso.
PATCH	Atualizar parcialmente um determinado recurso.
DELETE	Excluir um determinado recurso.
HEAD	Similar ao GET, mas utilizado apenas para se obter os cabeçalhos de resposta, sem os dados em si.
OPTIONS	Obter quais manipulações podem ser realizadas em um determinado recurso.

Conceitos de Sistemas WEB

REST

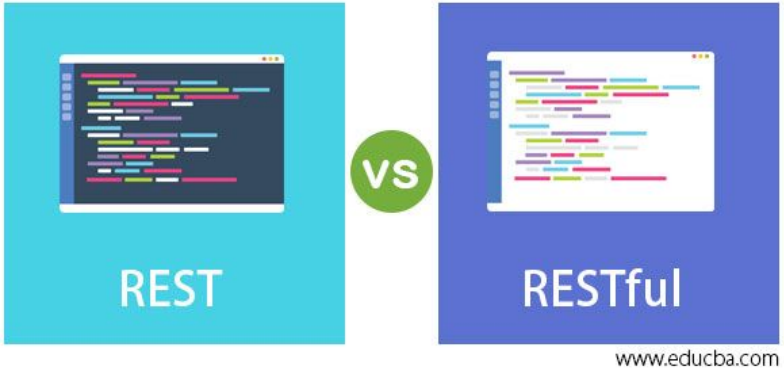
- Tente utilizar corretamente os códigos HTTP:

Categoria	Descrição	Exemplos
1xx	Mensagens Informativas	100, 102
2xx	Indicação de que a Requisição foi recebida com Sucesso	**200, 201**, 204
3xx	Indicação de Redirecionamento	307
4xx	Indicação de Erro por parte do *Client*	**400, 401, 403,415**
5xx	Indicação de Erro por parte do *Server*	**500**

Conceitos de Sistemas WEB

REST x RESTful

O protocolo HTTP e o padrão JSON podem ser usados para implementar os princípios REST. Quando estes princípios são observados, temos um sistema ou **API RESTful**.



8 Recomendações sobre REST API

{REST:API}

Recomendações REST API

REST

Recomendação 1 - Utilize URI's legíveis

- Ao definir uma URI, utilize nomes legíveis por humanos, que sejam de fácil dedução e que estejam relacionados com o domínio da aplicação.
- Isso facilita a vida dos clientes que utilizarão o serviço, além de reduzir a necessidade de documentações extensas.

Recomendações REST API

REST

Recomendação 2 - Utilize o mesmo padrão de URI na identificação dos recursos

- Mantenha a consistência na definição das URI's. Crie um padrão de nomenclatura para as URI's dos recursos e utilize sempre esse mesmo padrão.
- Evite situações como:
  - `http://servicorest.com.br/produto (Singular)`
  - `http://servicorest.com.br/clientes (Plural)`
  - `http://servicorest.com.br/processosAdministrativos (Camel Case)`
  - `http://servicorest.com.br/processos_judidicais (Snake Case)`

Recomendações REST API

REST

Recomendação 3 - Evite adicionar na URI a operação a ser realizada no recurso

- Evite definir URI's que contenham a operação a ser realizada em um recurso, tais como:
  - `http://servicorest.com.br/produtos/cadastrar`
  - `http://servicorest.com.br/clientes/10/excluir`
  - `http://servicorest.com.br/vendas/34/atualizar`

Recomendações REST API

REST

Recomendação 4 - Evite adicionar na URI o formato desejado da representação do recurso

- É comum que um serviço REST suporte múltiplos formatos para representar seus recursos, tais como XML e JSON.
- A informação sobre qual o formato desejado por um cliente ao consultar um serviço REST deve ser feita via Content Negotiation, conforme será mostrado mais adiante.
- Portanto, evite definir URI's que contenham o formato desejado de um recurso, tais como:
    - `http://servicorest.com.br/produtos/xml`
    - `http://servicorest.com.br/clientes/112?formato=json`

Recomendações REST API

REST

Recomendação 5 - Representações dos recursos

```xml
<cliente>
  <nome>Rodrigo</nome>
  <email>rodrigo@email.com.br</email>
  <sexo>Masculino</sexo>
  <endereco>
    <cidade>Brasilia</cidade>
    <uf>DF</uf>
  </endereco>
</cliente>
```

```json
{
    "nome": "Rodrigo",
    "email" : "rodrigo@email.com.br",
    "sexo": "Masculino",
    "endereco": {
        "cidade": "Brasilia",
        "uf": "DF"
    }
}
```

Recomendações REST API

REST

Recomendação 6 - Comunicação Stateless

- Requisições feitas por um cliente a um serviço REST devem conter todas as informações necessárias para que o servidor as interprete e as execute corretamente.
- Clientes não devem depender de dados previamente armazenados no servidor para processar uma requisição.
- Qualquer informação de estado deve ser mantida pelo cliente e não pelo servidor.
- Isso melhora a escalabilidade de um serviço REST.

Recomendações REST API

REST

Recomendação 7 - Evite manter dados de autenticação/autorização em sessão

- A principal dificuldade em criar um serviço REST totalmente Stateless ocorre quando precisamos lidar com os dados de autenticação/autorização dos clientes.

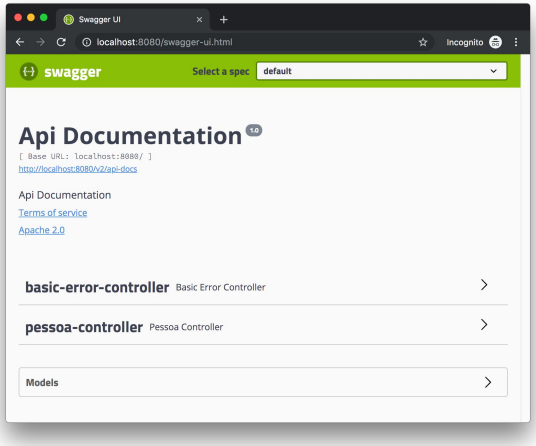
- A principal solução utilizada para resolver esse problema é a utilização de Tokens de acesso, que são gerados pelo serviço REST e devem ser armazenados pelos clientes, via cookies ou HTML 5 Web Storage.

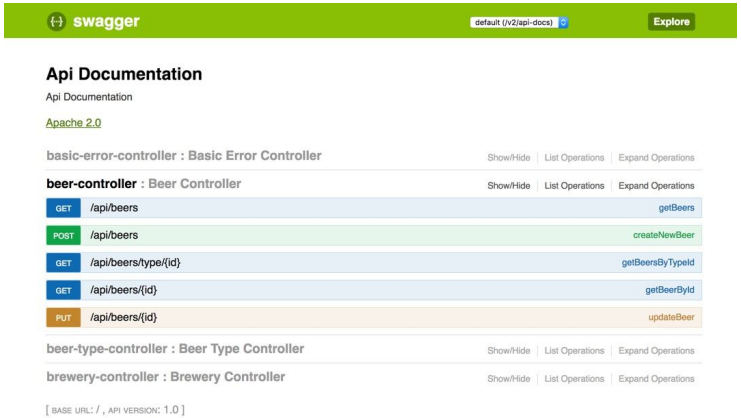
Recomendações REST API

REST

Recomendação 8 - Documente sua API

- (Sugestão) Ferramenta para documentar sua API:
  - https://swagger.io





Dúvidas

Obrigado !