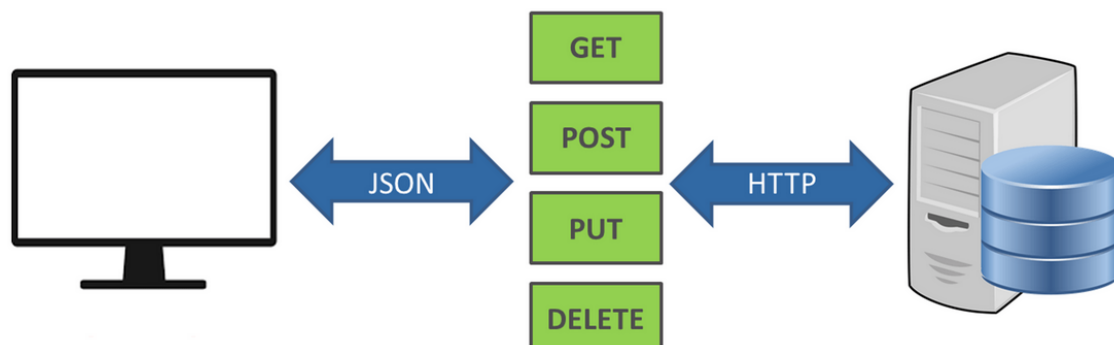


Padrões de Design para API'S REST

Este documento apresenta todas as informações necessárias para o desenvolvimento de serviços REST para os sistemas desenvolvidos pela GETIC.



Formato de Troca de Informações

JSON (JavaScript Object Notation) é o formato padrão troca de informações/dados entre sistemas desenvolvidos pela GETIC. Dentre as principais vantagens da adoção do formato JSON podemos citar:

- Leitura mais simples;
- Analisador(parsing) mais fácil;
- JSON suporta objetos;
- Velocidade maior na execução e transporte de dados;
- Arquivo com tamanho reduzido;

Recursos

Um recurso é uma abstração sobre um determinado tipo de informação que uma aplicação gerencia. Um dos princípios do REST diz que todo recurso

deve possuir uma identificação única. Essa identificação serve para que a aplicação consiga diferenciar qual dos recursos deve ser manipulado em uma determinada solicitação. A identificação do recurso deve ser feita através da URI (Uniform Resource Identifier).

```
http://{servidor}/{aplicacao}/api/{versao}/{recurso}
```

```
http://{servidor}/{aplicacao}/api/{versao}/{recurso}/{id}
```

```
http://{servidor}/{aplicacao}/api/{versao}/{recurso}/{id}/{subrec}
```

```
http://{servidor}/{aplicacao}/api/{versao}/{recurso}/{id}/{subrec}/{idsub}
```

Regras para Identificação de Recursos

- Utilizar os métodos HTTP de GET, POST, PUT, DELETE e PATCH;
- Não utilizar verbos;
- Usar sempre o plural, utilize ;

```
GET http://sistemas.int.cagece.com.br/api/v1/clientes
```

- Em caso de nomes compostos ou que exijam mais de uma palavra para identificação do recurso, crie subrecursos.

```
GET http://sistemas.int.cagece.com.br/api/v1/contracheques/anos
```

```
GET http://sistemas.int.cagece.com.br/api/v1/contracheques/anos/2019/meses
```

```
GET http://sistemas.int.cagece.com.br/api/v1/contracheques/anos/2019/meses/12
```

```
GET http://sistemas.int.cagece.com.br/api/v1/rendimentos/anos
```

```
GET http://sistemas.int.cagece.com.br/api/v1/rendimentos/anos/2019/meses
```

GET <http://sistemas.int.cagece.com.br/api/v1/rendimentos/anos/2019/meses/12>

GET <http://sistemas.int.cagece.com.br/api/v1/funcionarios/999999/rendimentos/anos>

GET <http://sistemas.int.cagece.com.br/api/v1/funcionarios/999999/rendimentos/anos/2019/meses>

GET <http://sistemas.int.cagece.com.br/api/v1/funcionarios/999999/rendimentos/anos/2019/meses/12>

GET <http://sistemas.int.cagece.com.br/api/v1/colaboradores/999999/rendimentos/anos>

GET <http://sistemas.int.cagece.com.br/api/v1/colaboradores/999999/rendimentos/anos/2019/meses>

GET <http://sistemas.int.cagece.com.br/api/v1/colaboradores/999999/rendimentos/anos/2019/meses/12>

- Utilizar nomes legíveis que facilite o entendimento por outro desenvolvedor;
- Manter a compatibilidade entre versões do sistemas;

GET <http://sistemas.int.cagece.com.br/api/v1/clientes>

GET <http://sistemas.int.cagece.com.br/v2/clientes>

- Caso queira retornar um registro específico, utilize Path Parameters para passar o sequencial do recurso desejado

GET <http://sistemas.int.cagece.com.br/api/v1/clientes/280>

GET <http://sistemas.int.cagece.com.br/api/v1/clientes/280/faturas>

GET <http://sistemas.int.cagece.com.br/api/v1/clientes/280/faturas/1234>

- Se for utilizar um serviço com filtro você pode seguir duas alternativas. Para serviços com filtros limitados utilize o GET com Query Parameters

GET <http://localhost:8080/api/v1/empresas?nome=CAGECE>

- Se o filtro tiver paginação e/ou vários campos, utilize o método POST passando os campos no corpo da requisição.

```
POST http://localhost:8080/api/v1/clientes/filtrar
{
  "pagina": 1,
  "maxItens": 10, //qtd máxima de itens por página
  "cliNome": "MARIA",
  "ordenarPor": "cliNome",
  "campos": [
    "cliSeqCliente",
    "cliNome",
    "cliDatNascimento"
  ]
}
```

- O retorno para as consultas devem seguir o padrão, podendo ser adicionado mais campos de acordo com a necessidade.

```
{
  "pagina":1, //necessário quando houver paginação
  "codigo": 200, //caso julgue necessário
  "mensagem": "...", //caso julgue necessário
  "resultado": [
    {
      "cliSeqCliente": 001,
      "cliNome": "MARIA",
      "cliDatNascimento": "01/03/1980"
    },
    {
      "cliSeqCliente": 011,
      "cliNome": "MARIA",
      "cliDatNascimento": "01/01/1989"
    },
    ...
  ]
}
```

Métodos HTTP que devem ser utilizados

Método	Objetivo
GET	Obter dados de um recurso específico
POST	Criar um novo recurso
PUT	Atualizar os dados de um determinado recurso
PATCH	Atualizar dados parciais de um determinado recurso
DELETE	Excluir um recurso específico

Códigos HTTP

Código	Semântica
2xx	Indica que a requisição foi processada com sucesso
3xx	Indica ao cliente uma ação a ser tomada para que a requisição seja concluída
4xx	Indica erro(s) na requisição causado(s) pelo cliente
5xx	Indica que a requisição não foi concluída devido a erro(s) ocorrido(s) no servidor

Observação

Ao consumir um recurso e o registro não existir na base de dados, não retorne um 404, prefira retornar um 200 com uma mensagem informando que o o registro não está cadastrado. utilize 404 apenas quando o recurso não for encontrado.

RESUMO

URL	VERBO HTTP	POST BODY	RESULTADO
http://www.cagece.com.br/prax/api/v1/clientes	GET	Vazio	Consulta todos os clientes
http://www.cagece.com.br/prax/api/v1/clientes	POST	JSON	Cria um Registro
http://www.cagece.com.br/prax/api/v1/clientes/12345	GET	Vazio	Consulta um registro específico
http://www.cagece.com.br/prax/api/v1/clientes/filtrar	POST	JSON	Consulta Registros
http://www.cagece.com.br/prax/api/v1/clientes/12345	PUT	JSON	Altera registro
http://www.cagece.com.br/prax/api/v1/clientes/12345	PATCH	JSON	Altera registro parcialmente
http://www.cagece.com.br/prax/api/v1/clientes/12345	DELETE	JSON	Delete Registro