Sicredi
Engenharia de Software

API DESIGN

Referência Rápida

ARQUITETURA (REST/JSON)

APIs são fiéis ao estilo arquitetural REST baseadas nos protocolos HTTP definido pelos seguintes aspectos:

- uma URL, tal como `http://minhaempresa.com.br/negocio/dominio/v1/cervejas`
- Um *media type* que define os elementos de dados de transição de estado (JSON, XML, Atom). Preferencialmente o JSON
- Métodos HTTP (ex: GET, POST, PUT e DELETE)

RECURSOS

É uma abstração de uma informação (normalmente entidades de negócio) gerenciada por uma aplicação.
O recurso representa uma coleção onde cada elemento possui um identificador único.

Devem ser um substantivo no plural, não um verbo
- ✔ POST /cervejas
- ✔ POST /carros
- ✔ GET /produtos/{id}

Função não é domínio de negócio.
Nunca nomeie recursos desta forma:
- ✖ GET /obterCervejas
- ✖ PUT /cervejas/atualizar

Mapa correto de relações utilizando sub-recursos:
- ✔ POST /cervejas/{id}/avaliacoes
- ✔ GET /cervejas/{id}/avaliacoes/{id}

VERSIONAMENTO

Implemente o controle de versão na URI antes dos recursos e exponha as versões via path.

- `https://minhaempresa.com.br/bebidas/`**`v1`**`/cervejas`
- `https://minhaempresa.com.br/bebidas/`**`v2`**`/cervejas`

A versão deve ser atualizada quando quebrar o contrato com o cliente. Utilizar a estratégia do Semantic Versioning para o versionamento *major*, *minor* e *patch* da API

OPERAÇÕES

Uma operação é uma unidade de uma API que você pode chamar. É composta de um verbo HTTP e um caminho de URL que é subordinado ao contexto principal da API (URI). Criando uma operação você define como a API será exposta para utilização.

Temos as seguintes operações:

VERBO	PATH	CRUD	DESCRIÇÃO
GET	/cervejas	Lista (índice)	Usado para retornar uma coleção (lista)
GET	/cervejas/{id}	Read	Usado para retornar um determinado elemento da coleção
POST	/cervejas	Create	Usado para criar um elemento na coleção
PUT	/cervejas/{id}	Update	Usado para atualizar um elemento na coleção
PATCH	/cervejas/{id}	Partial Update	Usado para atualizar parcialmente um elemento na coleção
DELETE	/cervejas/{id}	Delete	Usado para excluir toda uma coleção

SUCESSFULL - 2xx

200	**OK**	Status genérico de sucesso	**GET** /cervejas/12345 **200 OK** { "tipo" : "Trapista" }
201	**Created**	Indica a criação de um recurso Normalmente utilizado para responder a requisição de POSTs	**POST** /cervejas **201 Created** Location: /cervejas/12345
202	**Accepted**	Indica que a requisição foi "aceita" para processamento Normalmente utilizada em chamadas assíncronas	**POST** /pedidos **202 Accepted** Location: /pedidos/123
204	**No Content**	Indica a execução com sucesso de uma operação que não retornará informação Normalmente utilizada como resposta dos verbos PUT, PATCH e DELETE	**DELETE** /cervejas/12345 **204 No Content**

REDIRECT - 3xx

301	Moved Permanently	Recurso requisitado foi movido para outro recurso, permanentemente	**GET** /ceva/ **301 Moved Permanently** Location: /cervejas
302	Found	Recurso requisitado foi movido para outro recurso, temporariamente	**GET** http://minhaempresa.com/bebidas **302 Found** Location: http://bebidas.minhaempresa.com

CLIENT ERROR - 4xx

400	**Bad Request**	Status genérico de erro para requisições que não puderam ser processadas devido a requisição não aderente ao contrato	**POST** /cervejas **400 Bad Request** { "mensagem": "Requisição mal formatada" }
401	**Unauthorized**	O servidor não reconheceu por falta de credenciais válidas para o recurso solicitado	**GET** /cervejas/12345 **401 Unauthorized**
403	**Forbidden**	Sua credencial não tem privilégios suficientes para acessar o recurso que está solicitando	**GET** /cervejas/12345 **403 Forbidden** { "mensagem": "Você não tem permissões suficientes" }
404	**Not Found**	A URI informada na requisição não existe	**GET** /cervejas/00001 404 Not Found
405	**Method not allowed**	O método HTTP utilizado no recurso não é permitido	**DELETE** /cervejas **405 Method Not Allowed**
412	**Precondition Fail**	Algo na requisição não está aderente ao esperado	**POST** /cervejas **412 Precondition Fail** { "mensagem": "Headers obrigatórios não informados" }
415	**Unsupported Media Type**	O payload está em um formato que o servidor não reconhece. As vezes é resolvido com atributos *Content-Type* ou *Content-Enconding*	**PUT** /cervejas/12345 **415 Unsupported Media Type**
422	**Unprocessable Entity**	Ocorreu algum erro de negócio com sua mensagem. Sintaticamente correto, semanticamente não.	**POST** /cervejas **422 Unprocessable Entity** { "mensagem": "Cerveja já existente" }

SERVER ERROR - 5xx

500	**Internal Server Error**	A requisição está certa, porém algum erro aconteceu no servidor. Não adianta mandar de novo agora!	**GET** /cervejas?sem-alcool=true **500 Internal Server Error** { "mensagem": "Alguma coisa deu errado" }
502	**Bad Gateway**	O API Gateway não conseguiu identificar exatamente o "erro" reportado pelo backend	**GET** /cervejas/12345 **502 Bad Gateway**
503	**Service Unavailable**	O servidor não consegue processar agora por sobrecarga ou manutenção	**GET** /cervejas/12345 **503 Service Unavailable**
504	**Gateway Timeout**	A requisição excedeu o tempo pré estabelecido de timeout	**GET** /cervejas **504 Gateway Timeout**

FILTROS, PAGINAÇÃO E ORDENAÇÃO

Os mecanismos de filtro , paginação e ordenação são utilizados em pesquisas (métodos GET) com o intuito de otimizar resultados com base em um conjunto de dados.

- GET /cervejas?_offset=5&_limit=3
- GET /cervejas?tipo="IPA"
- GET /cervejas?_sort="preco|asc"
- GET /cerveja?preco=greaterThan(10)

MODELO DE EXPOSIÇÃO DE URL

O Modelo de Exposição de URL interna tem como base o nome do componente, seguindo algumas diretivas. Ele é composto por:

https://**nome-componente**-**ambiente**-pass.sicredi.net

Ambiente	Exposição
Desenvolvimento	dev.rs-1
Homologação	hom.uat.rs-1
Pré-Produção	pre.uat.rs-1
Produção	prd.rs-1

Exemplo
- nome componente: geração-previa-sobras
- ambiente: homologação

https://geracao-previa-sobras-hom.uat.rs-1.paas.sicredi.net

CALLBACK

Utilizado em recursos assíncronos para informar ao consumidor o término de um procedimento.
Normalmente utilizado onde o processamento de uma determinada instrução é custoso e o tempo de resposta não é rápido.

Como sugestão: se o tempo de resposta for >30s considere criar um processamento assíncrono (batch)

SEGURANÇA

Mantenha a sua API segura. Avalie o nível de criticidade e adote a estratégia adequada como, por exemplo, autenticação, autorização, criptografia, etc... Sempre leve em consideração os seguintes atributos:
- Autenticação / Autorização
- Privacidade
- Auditoria
- Integridade
- Disponibilidade

E lembre-se, o elo mais fraco de toda a estratégia de segurança é sempre o ser humano!

ARQUITETURA (REST/JSON)

APIs são fiéis ao estilo arquitetural REST baseadas nos protocolos HTTP definido pelos seguintes aspectos:

- uma URL, tal como `http://minhaempresa.com.br/negocio/dominio/v1/cervejas`
- Um *media type* que define os elementos de dados de transição de estado (JSON, XML, Atom). Preferencialmente o JSON
- Métodos HTTP (ex: GET, POST, PUT e DELETE)

RECURSOS

É uma abstração de uma informação (normalmente entidades de negócio) gerenciada por uma aplicação.
O recurso representa uma coleção onde cada elemento possui um identificador único.

Devem ser um substantivo no plural, não um verbo
- ✔ POST /cervejas
- ✔ POST /carros
- ✔ GET /produtos/{id}

Função não é domínio de negócio.
Nunca nomeie recursos desta forma:
- ✖ GET /obterCervejas
- ✖ PUT /cervejas/atualizar

Mapa correto de relações utilizando sub-recursos:
- ✔ POST /cervejas/{id}/avaliacoes
- ✔ GET /cervejas/{id}/avaliacoes/{id}

VERSIONAMENTO

Implemente o controle de versão na URI antes dos recursos e exponha as versões via path.

- `https://minhaempresa.com.br/bebidas/`**`v1`**`/cervejas`
- `https://minhaempresa.com.br/bebidas/`**`v2`**`/cervejas`

A versão deve ser atualizada quando quebrar o contrato com o cliente. Utilizar a estratégia do Semantic Versioning para o versionamento *major*, *minor* e *patch* da API

OPERAÇÕES

Uma operação é uma unidade de uma API que você pode chamar. É composta de um verbo HTTP e um caminho de URL que é subordinado ao contexto principal da API (URI). Criando uma operação você define como a API será exposta para utilização.

Temos as seguintes operações:

VERBO	PATH	CRUD	DESCRIÇÃO
GET	/cervejas	Lista (índice)	Usado para retornar uma coleção (lista)
GET	/cervejas/{id}	Read	Usado para retornar um determinado elemento da coleção
POST	/cervejas	Create	Usado para criar um elemento na coleção
PUT	/cervejas/{id}	Update	Usado para atualizar um elemento na coleção
PATCH	/cervejas/{id}	Partial Update	Usado para atualizar parcialmente um elemento na coleção
DELETE	/cervejas/{id}	Delete	Usado para excluir toda uma coleção

SUCESSFULL - 2xx

200	**OK**	Status genérico de sucesso	**GET** /cervejas/12345 **200 OK** { "tipo" : "Trapista" }
201	**Created**	Indica a criação de um recurso Normalmente utilizado para responder a requisição de POSTs	**POST** /cervejas **201 Created** Location: /cervejas/12345
202	**Accepted**	Indica que a requisição foi "aceita" para processamento Normalmente utilizada em chamadas assíncronas	**POST** /pedidos **202 Accepted** Location: /pedidos/123
204	**No Content**	Indica a execução com sucesso de uma operação que não retornará informação Normalmente utilizada como resposta dos verbos PUT, PATCH e DELETE	**DELETE** /cervejas/12345 **204 No Content**

REDIRECT - 3xx

301	Moved Permanently	Recurso requisitado foi movido para outro recurso, permanentemente	**GET** /ceva/ **301 Moved Permanently** Location: /cervejas
302	Found	Recurso requisitado foi movido para outro recurso, temporariamente	**GET** http://minhaempresa.com/bebidas **302 Found** Location: http://bebidas.minhaempresa.com

CLIENT ERROR - 4xx

400	**Bad Request**	Status genérico de erro para requisições que não puderam ser processadas devido a requisição não aderente ao contrato	**POST** /cervejas **400 Bad Request** { "mensagem": "Requisição mal formatada" }
401	**Unauthorized**	O servidor não reconheceu por falta de credenciais válidas para o recurso solicitado	**GET** /cervejas/12345 **401 Unauthorized**
403	**Forbidden**	Sua credencial não tem privilégios suficientes para acessar o recurso que está solicitando	**GET** /cervejas/12345 **403 Forbidden** { "mensagem": "Você não tem permissões suficientes" }
404	**Not Found**	A URI informada na requisição não existe	**GET** /cervejas/00001 404 Not Found
405	**Method not allowed**	O método HTTP utilizado no recurso não é permitido	**DELETE** /cervejas **405 Method Not Allowed**
412	**Precondition Fail**	Algo na requisição não está aderente ao esperado	**POST** /cervejas **412 Precondition Fail** { "mensagem": "Headers obrigatórios não informados" }
415	**Unsupported Media Type**	O payload está em um formato que o servidor não reconhece. As vezes é resolvido com atributos *Content-Type* ou *Content-Enconding*	**PUT** /cervejas/12345 **415 Unsupported Media Type**
422	**Unprocessable Entity**	Ocorreu algum erro de negócio com sua mensagem. Sintaticamente correto, semanticamente não.	**POST** /cervejas **422 Unprocessable Entity** { "mensagem": "Cerveja já existente" }

SERVER ERROR - 5xx

500	**Internal Server Error**	A requisição está certa, porém algum erro aconteceu no servidor. Não adianta mandar de novo agora!	**GET** /cervejas?sem-alcool=true **500 Internal Server Error** { "mensagem": "Alguma coisa deu errado" }
502	**Bad Gateway**	O API Gateway não conseguiu identificar exatamente o "erro" reportado pelo backend	**GET** /cervejas/12345 **502 Bad Gateway**
503	**Service Unavailable**	O servidor não consegue processar agora por sobrecarga ou manutenção	**GET** /cervejas/12345 **503 Service Unavailable**
504	**Gateway Timeout**	A requisição excedeu o tempo pré estabelecido de timeout	**GET** /cervejas **504 Gateway Timeout**

Sicredi
Engenharia de Software

API DESIGN

Referência Rápida

FILTROS, PAGINAÇÃO E ORDENAÇÃO

Os mecanismos de filtro , paginação e ordenação são utilizados em pesquisas (métodos GET) com o intuito de otimizar resultados com base em um conjunto de dados.

- GET /cervejas?_offset=5&_limit=3
- GET /cervejas?tipo="IPA"
- GET /cervejas?_sort="preco|asc"
- GET /cerveja?preco=greaterThan(10)

CACHING

Esta estratégia possibilita reduzir o tráfego desnecessário, latência de rede e sobrecarga nos servidores que provém as APIs para as operações de GET. Ative a função de cache no seu Gateway ou implemente-o.

CALLBACK

Utilizado em recursos assíncronos para informar ao consumidor o término de um procedimento.
Normalmente utilizado onde o processamento de uma determinada instrução é custoso e o tempo de resposta não é rápido.

Como sugestão: se o tempo de resposta for >30s considere criar um processamento assíncrono (batch)

HYPERMEDIA

Conhecido como HATEOAS, provê hyperlink nas respostas de operações para possibilitar a navegação dinâmica em outras interfaces da API.

- GET /cervejas?tipo="IPA"
  200 OK
  { "id": 1234, "nome": "IPA",
  "links": [
  { "rel": "self", "type": "GET",
  "href": "/cervejas/12345/avaliações" }]
  }

SEGURANÇA

Mantenha a sua API segura. Avalie o nível de criticidade e adote a estratégia adequada como, por exemplo, autenticação, autorização, criptografia, etc... Sempre leve em consideração os seguintes atributos:

- Autenticação / Autorização
- Privacidade
- Auditoria
- Integridade
- Disponibilidade

E lembre-se, o elo mais fraco de toda a estratégia de segurança é sempre o ser humano!