REST - Boas práticas

Bootcamp Python
/código[S]
stone

Maicon Francisco de Carvalho
HOW Bootcamps

Boas práticas de uso de HTTP

- Uso adequado dos métodos HTTP;
- Uso adequado de URI’s;
- Uso de códigos de status padronizados para representação de sucessos ou falhas;
- Uso adequado de cabeçalhos HTTP.

URI’s legíveis

Utilize nomes legíveis por humanos, que sejam de fácil entendimento 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.

URI’s no singular ou no plural?

Não existem regras em relação a estas URI’s estarem no singular ou no plural - ou seja, não importa se você prefere utilizar */empresa* ou */empresas* . O ideal, no entanto, é manter um padrão - tenha todas as suas URI’s no singular ou todas no plural, mas não misturas.

Padronize as URI’s

Mantenha a consistência na definição das URI’s. Crie um padrão de nomenclatura para as URIs dos recursos e utilize sempre esse mesmo padrão.

Evite situações como:

/carro (Singular);

/produtos (Plural);

/politicasAdministrativas (Camel Case);

/processos_internos (Snake Case).

Distinguir operação e recurso

O método HTTP é utilizado para determinar a operação a ser realizada em um determinado recurso. Em geral, utiliza-se o GET para recuperar, POST para criar, PUT/PATH para alterar e DELETE para apagar.

Portanto, evite definir URI's que contenham a operação a ser realizada em um recurso, tais como:

/produtos/cadastrar;

/clientes/170/excluir;

/vendas/43/atualizar.

Uso correto do Content Negotiation

É comum que um serviço REST suporte múltiplos formatos para representar seus recursos, tais como XML, JSON e HTML. A informação sobre qual o formato desejado por um cliente ao consultar um serviço REST deve ser feita via Content Negotiation.

Portanto, evite definir URI's que contenham o formato desejado de um recurso, tais como:

/produtos/xml;

/clientes/112?formato=json.

Verbos HTTP para manipulação dos recursos

Quando um cliente dispara uma requisição HTTP para um serviço, além da URI que identifica quais recursos ele pretende manipular, é necessário que ele também informe o tipo de manipulação que deseja realizar no recurso.

GET	/clientes	Todos os clientes.
GET	/clientes/id	Um determinado cliente.
POST	/clientes	Criar um novo cliente.
PUT	/clientes/id	Atualizar o cliente.
DELETE	/clientes/id	Excluir um determinado cliente.

python

Obrigado!

stone + howtech