

Nome: Bruno Soares dos Santos

E-mail: brunobsds8@hotmail.com

As seis regras de uma API REST (Representational State Transfer) são conhecidas como as "Restful Constraints". Estas regras são consideradas as melhores práticas para projetar e implementar uma API RESTful. As seis regras são:

1. Arquitetura Cliente-Servidor: A API deve ser construída seguindo o princípio de separação de responsabilidades. O cliente é responsável por fazer requisições e o servidor é responsável por armazenar os dados e processar as requisições.
2. Sem estado (Stateless): O servidor não deve manter nenhuma informação sobre o estado do cliente. Isso significa que cada requisição enviada pelo cliente deve conter todas as informações necessárias para que o servidor possa processá-la.
3. Cacheável: As respostas do servidor devem ser marcadas como cacheáveis ou não-cacheáveis. Isso permite que o cliente armazene as respostas em cache para que possam ser reutilizadas em futuras requisições.
4. Interface uniforme: A API deve ter uma interface uniforme, que define como o cliente deve interagir com o servidor. Isso inclui o uso de recursos, identificadores, métodos HTTP e formatos de mensagem.
5. Sistema em camadas: A API deve ser construída em camadas, onde cada camada pode se comunicar apenas com as camadas adjacentes. Isso ajuda a manter a escalabilidade e a segurança da API.
6. Manipulação de código sob demanda (optional): A API pode ser estendida através da transferência de código executável do servidor para o cliente. Isso é opcional e não é amplamente utilizado.

Seguir essas regras ajuda a garantir que a API RESTful seja escalável, flexível e fácil de usar. Além disso, essas regras ajudam a garantir a interoperabilidade entre diferentes sistemas e tecnologias.