

INSTITUTO FEDERAL DE RONDÔNIA - CAMPUS JI-PARANÁ

Wallan Oliveira da Silva

Arquiterura REST

Pesquisa

Ji-Paraná
2023

Wallan Oliveira da Silva

Arquiterura REST

Pesquisa apresentada a Graduação em Análise e Desenvolvimento de Sitemas.

Professor: Joao Eujacio Teixeira Junior

Disciplina: Tópicos Especiais em Tecnologia
I

Turma: 3º Período

Ji-Paraná

2023

Resumo

A arquitetura REST (Representational State Transfer) é um estilo de design amplamente utilizado para desenvolver APIs que permitem a comunicação eficiente entre sistemas distribuídos. Proposto por Roy Fielding, esse paradigma segue princípios fundamentais, incluindo a separação entre clientes e servidores, ausência de estado entre requisições, uso de cache e uma interface uniforme através de métodos HTTP padrão, como GET, POST, PUT e DELETE.

Ao adotar REST, as organizações podem colher diversos benefícios. A simplicidade na criação de APIs, possibilitada pela interface uniforme e padrões HTTP, facilita o desenvolvimento e a integração. A arquitetura favorece a escalabilidade, tornando sistemas capazes de lidar com um grande volume de requisições. A visibilidade é aprimorada por meio de URLs descritivas e ações HTTP claras, contribuindo para a documentação e compreensão das APIs. A independência de plataforma permite que diferentes clientes consumam a mesma API, independentemente do dispositivo ou sistema operacional.

Melhores práticas ao desenvolver APIs REST incluem a escolha de nomes de recursos descritivos, o uso adequado de métodos HTTP para ações específicas, fornecer respostas claras com códigos de status apropriados e implementar segurança com autenticação e autorização robustas. O versionamento da API na URL e a consideração de casos mais complexos são igualmente importantes.

Apesar das vantagens, REST apresenta algumas desvantagens. Em operações complexas, a abordagem pode gerar complexidade. Além disso, a natureza stateless pode levar a uma maior carga de requisições e questões de controle em determinados cenários.

Referências confiáveis para o estudo da arquitetura REST incluem a tese original de Roy Fielding sobre o tema, o livro “RESTful Web APIs: Services for a Changing World” de Richardson e Amundsen, artigos acadêmicos como “RESTful Web Services” de Tilkov e Vinoski, bem como recursos online como a documentação oficial da API Java JAX-RS da Oracle. A seleção cuidadosa dessas referências proporcionará uma base sólida para entender, aplicar e avaliar a arquitetura REST no contexto do desenvolvimento de APIs.

Palavras-chave: Abstract. Resumo. ABNT.

Arquitetura REST: Princípios, Benefícios e Melhores Práticas para o Desenvolvimento de APIs

A arquitetura REST (Representational State Transfer) é um estilo arquitetural amplamente utilizado para projetar sistemas distribuídos e desenvolver APIs (Interfaces de Programação de Aplicativos) que permitam a comunicação entre diferentes sistemas. Ela foi apresentada pela primeira vez por Roy Fielding em sua tese de doutorado em 2000 e tem sido amplamente adotada para a construção de serviços web escaláveis e flexíveis. A seguir, são apresentados os princípios, benefícios e melhores práticas associados à arquitetura REST.

1 Princípios da Arquitetura REST:

1. Arquitetura Cliente-Servidor: Os sistemas são divididos em componentes independentes: clientes, que fazem requisições, e servidores, que fornecem recursos.

2. Stateless (Sem Estado): Cada requisição do cliente para o servidor deve conter todas as informações necessárias para que o servidor entenda e processe a requisição. O servidor não deve depender de estados anteriores.

3. Cache: Os servidores podem definir respostas como cacheados ou não-cacheados. Isso melhora a eficiência e a performance ao reduzir a necessidade de buscar dados do servidor repetidamente.

4. Interface Uniforme: Os recursos são identificados por URLs, e as ações são realizadas usando métodos HTTP padrão, como GET, POST, PUT e DELETE. Isso promove uma interface consistente e previsível.

5. Sistema em Camadas: Os componentes podem estar em diferentes camadas, permitindo a escalabilidade e a flexibilidade do sistema.

6. Code-On-Demand (Opcional): Os servidores podem fornecer código executável (como JavaScript) para ser executado no cliente. Essa opção não é frequentemente utilizada em APIs REST.



FASTFORMAT

Você precisa adquirir um plano para remover a marca d'água.

You need to subscribe to a plan to remove the watermark.

2 Benefícios da Arquitetura REST:

1. Simplicidade: A interface uniforme e os padrões HTTP simplificam o desenvolvimento e a integração de APIs.
2. Escalabilidade: A arquitetura REST permite que os sistemas sejam escalados horizontalmente para lidar com um grande número de requisições.
3. Visibilidade: URLs descritivas e ações HTTP claras facilitam a compreensão e a documentação das APIs.
4. Independência de Plataforma: Diferentes clientes (por exemplo, aplicativos web, dispositivos móveis) podem consumir a mesma API, independentemente da plataforma.
5. Flexibilidade: Os clientes podem consumir diferentes tipos de mídia (XML, JSON, HTML etc.) baseados em suas necessidades.



Você precisa adquirir um plano para remover a marca d'água.

You need to subscribe to a plan to remove the watermark.

3 Melhores Práticas para o Desenvolvimento de APIs REST:

1. Use Nomes de Recursos Descritivos: Escolha nomes de recursos que representem de maneira clara e intuitiva os objetos que estão sendo manipulados.
2. Utilize Métodos HTTP Adequados: Use os métodos HTTP apropriados (GET, POST, PUT, DELETE) para as ações correspondentes (recuperar, criar, atualizar, excluir).
3. Forneça Respostas Adequadas: Utilize códigos de status HTTP apropriados (200 OK, 201 Created, 400 Bad Request, 404 Not Found etc.) para indicar o resultado da requisição.
4. Versionamento: Utilize versionamento na URL da API para permitir atualizações sem quebrar a compatibilidade com versões anteriores.
5. Segurança: Implementa autenticação e autorização adequadas para proteger a API contra acesso não autorizado.



Você precisa adquirir um plano para remover a marca d'água.

You need to subscribe to a plan to remove the watermark.

4 Vantagens da Utilização da Arquitetura REST:

1. Flexibilidade: A interface uniforme e a independência de plataforma tornam as APIs REST flexíveis e adaptáveis.
2. Performance: A abordagem stateless e o uso eficiente de cache podem melhorar a performance da API.
3. Padrões Abertos: A utilização dos padrões HTTP e URLs facilita a integração com outras ferramentas e tecnologias.



Você precisa adquirir um plano para remover a marca d'água.

You need to subscribe to a plan to remove the watermark.

5 Desvantagens da Utilização da Arquitetura REST:

1. Complexidade para Operações Complexas: APIs REST podem se tornar complexas quando envolvem operações mais complicadas, como transações em várias etapas.
2. Sobrecarga de Requisições: A abordagem stateless pode resultar em um grande número de requisições, o que pode causar sobrecarga na rede.
3. Falta de Controle: Algumas características da arquitetura, como o uso de cache, podem limitar o controle em certos cenários.



Você precisa adquirir um plano para remover a marca d'água.

You need to subscribe to a plan to remove the watermark.

6 Exemplos:

1. Recuperar um Usuário (GET): GET /api/users/123
2. Criar um Novo Usuário (POST): POST /api/users
3. Atualizar Dados de um Usuário (PUT): PUT /api/users/123
4. Excluir um Usuário (DELETE): DELETE /api/users/123
5. Recuperar Lista de Produtos (GET): GET /api/products



Você precisa adquirir um plano para remover a marca d'água.

You need to subscribe to a plan to remove the watermark.

7 Referências:

Fielding, RT. (2000). **Architectural Styles and the Design of Network-Based Software Architectures**. University of California, Irvine. Disponível em: <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>

Richardson, L., & Amundsen, M(2013). **RESTful Web APIs: Services for a Changing World**. O'Reilly Media.

OracleJava API for RESTful Web Services (JAX-RS). Disponível em: <https://docs.oracle.com/javase/7/tutorial/jaxrs.htm>



Você precisa adquirir um plano para remover a marca d'água.

You need to subscribe to a plan to remove the watermark.