

Aulas 001a004 - Atividade Supervisionada

Rest Api

Rest é uma sigla de Representational State Transfer (Estado de transferência representacional), o conceito de Rest Api é referente a uma estrutura de consumo de informação. Essa estrutura segue um conjunto de princípios característicos e que podem ser usados para caracterizar diversas estruturas de troca de informação, presentes em projetos de Software, como Rest Api. Dentre estes princípios podemos citar:

Cliente Servidor

Separação do projeto em duas camadas, a camada de cliente que é responsável por receber e tratar os dados referentes a aplicação, e a camada do servidor que armazena e disponibiliza tais dados para o cliente.

Stateless

Cada requisição que o cliente faz para o servidor deve possuir toda a informação necessária e compreensível para que uma resposta seja formulada, assim não é de responsabilidade do servidor armazenar qualquer tipo de contexto. Exemplo: A seção do usuário deverá ser enviada em todas as requisições, para que as restrições da aplicação possam ser feitas baseando-se nas permissões associadas à seção do usuário que realiza uma determinada requisição.

Cache

A cache é utilizada para melhorar a performance de comunicação das aplicações, otimizando o tempo de resposta na comunicação entre cliente-servidor. É de responsabilidade do servidor controlar as diretivas de cache através do cabeçalho http (header).

Camadas

Esse modelo de arquitetura valoriza a divisão de responsabilidade dentre as diversas partes do projeto, Os princípios e as boas práticas de arquitetura e design de um projeto, sugerem a separação do sistema em camadas e a delegação de gerenciamento limitado a seu próprio escopo para cada uma das camadas do projeto. Caso algo seja alterado ou em alguma das camadas criadas as demais não sofrerem alterações. Nesse modelo, a cliente não se conecta diretamente ao servidor mas uma camada adicional de balanceamento de carga é adicionada para lidar com esse conexão

Código sob demanda

Essa característica faz referência a limitação da quantidade de código recebida pelo cliente, já que o cliente só terá acesso a partes específicas do código que fazem

sentido para que a sua navegação dentro do sistema flua normalmente, logo, ele não terá acesso a completude da base de código do sistema.

Interface Uniforme

Este ponto trata da padronização das interfaces utilizadas ao longo do sistema, um exemplo seria a padronização do formato dos arquivos de dados que são transportados ao longo do sistema, a maioria dos sistemas web da atualidade utiliza o formato .json para transferir dados ao longo da aplicação, e seguindo essa padronização as aplicações estão de acordo com o ponto de interface uniforme proposta pela estrutura de Rest Api.