Evoluindo o CRUD da aplicação

Nesta aula mergulharemos no aprimoramento de aplicações através do "Evoluindo o CRUD da aplicação". Esta etapa é dedicada à crucial verificação de funcionalidades recém-integradas, utilizando a ferramenta Postman para simular e testar requisições HTTP. Nossa atenção se voltará especificamente para a funcionalidade de remarcação de consultas, exemplificando como novos recursos podem ser incorporados e validados no contexto de um sistema para consultórios médicos. Este processo não apenas testa a funcionalidade em si, mas reforça o ciclo de desenvolvimento ágil, enfatizando a importância da verificação contínua e da adaptação às necessidades emergentes. Ao fim desta etapa, estaremos preparados para explorar ainda mais o universo do desenvolvimento de APIs e Node.js, armados com o conhecimento prático sobre a evolução constante de software.

Apresentação de validadores

A validação é um aspecto fundamental de qualquer aplicação, assegurando que os dados inseridos sejam corretos, autênticos e adequados aos requisitos do sistema. Até agora, vocês já tiveram uma prévia de como as operações básicas do CRUD são estruturadas e algumas validações primárias foram implementadas. No entanto, neste módulo, exploraremos em detalhes os validadores, entendendo sua importância, como funcionam e como podem ser aplicados para fortalecer a integridade dos dados em nossa aplicação.

Começaremos explorando o conceito de validadores, fundamentais para proteger nossas aplicações de dados incorretos ou fraudulentos. A

aplicação prática de validadores é vasta, desde a verificação de e-mails válidos e números de CPF/CNPJ até a conformidade de formatos de arquivo em uploads. Além disso, discutiremos como esses validadores não apenas previnem a entrada de dados inválidos, mas também como podem ser usados para formatar ou corrigir dados antes de sua inserção no sistema, garantindo assim a qualidade e a confiabilidade da informação.

O processo de validação pode ser dividido em três etapas principais: entrada, validação e retorno. A entrada refere-se ao dado que será submetido à validação. A etapa de validação determina se o dado é verdadeiro ou falso, baseando-se em critérios preestabelecidos. E, dependendo do resultado dessa validação, o sistema providenciará um retorno correspondente - sucesso para dados válidos ou tratamento de erro/exceção para dados inválidos.

Esta aula visa equipá-los com o conhecimento teórico necessário para implementar validações eficazes em suas aplicações. Posteriormente, mergulharemos na prática, aplicando os conceitos discutidos em nosso mini projeto. Exploraremos a validação dos campos, a inclusão de novas funcionalidades e a verificação dessas implementações via Postman.

Entender e aplicar validadores é essencial para o desenvolvimento de software, transcendendo o ambiente acadêmico e aplicando-se diretamente ao desenvolvimento de aplicações reais. Preparem-se para explorar como essas técnicas podem melhorar significativamente a qualidade e a segurança das aplicações que desenvolvemos. Vamos juntos dar esse passo importante no nosso aprendizado sobre a construção de aplicações robustas e confiáveis. Nos vemos na próxima aula, prontos para transformar a teoria em prática!

Validação de campos

Inicialmente, focaremos na implementação da validação de campos em nossa aplicação. Esse processo é essencial para assegurar que os dados inseridos nas entidades da aplicação sejam adequados e válidos para o

contexto específico em que estão sendo utilizados.

A validação de campos é um passo crucial que segue as operações básicas do CRUD que já exploramos anteriormente. Ela se destina a verificar a autenticidade e a correção dos dados antes de permitir que entrem no sistema, utilizando uma lógica de validação que, em sua forma mais básica, se resume a determinar a veracidade (verdadeiro ou falso) de cada dado inserido. Este processo nos permite lidar com os dados de maneira individual, garantindo que cada campo seja validado conforme os requisitos específicos da aplicação.

Antes de iniciar a validação, é importante organizar e identificar quais são os campos cruciais que requerem validação em sua aplicação, determinando quais são essenciais e não podem conter erros. Cada campo será validado individualmente, o que significa que mesmo ao receber um conjunto de dados, é necessário analisar cada elemento separadamente para garantir a integridade e a validade dos dados no momento do armazenamento ou da execução de operações no banco de dados.

No nosso mini projeto, colocaremos em prática algumas validações fundamentais, começando pela validação de números de telefone, um dado essencial para a comunicação dentro do nosso sistema. Utilizaremos expressões regulares (regex) para garantir que os números de telefone sigam um formato específico, validando, por exemplo, o padrão brasileiro de números celulares, que inclui um código de área de dois dígitos, seguido pelo dígito 9, quatro dígitos, um traço e mais quatro dígitos.

Além da validação de telefone, outro aspecto crucial é a validação da existência de IDs de médicos e pacientes na base de dados, garantindo que as consultas possam ser registradas apenas se ambas as partes envolvidas estiverem corretamente cadastradas no sistema. Isso é realizado por meio da verificação da presença do ID no banco de dados, uma prática que reforça a integridade relacional das nossas informações.

As validações são implementadas diretamente nos modelos de dados, utilizando propriedades de validação específicas que incluem regras de validação e mensagens de erro personalizadas para guiar os usuários na

correção de dados inválidos. Essa abordagem não apenas eleva a qualidade dos dados inseridos no sistema como também contribui para a segurança e a usabilidade da aplicação.

A validação de campos é um pilar essencial no desenvolvimento de aplicações robustas e confiáveis. Ao implementar essas validações, estamos não apenas protegendo nossa aplicação contra dados incorretos, mas também melhorando a experiência do usuário ao garantir que a interação com o sistema seja o mais fluída e intuitiva possível.

Nova funcionalidade na aplicação

Agora, introduziremos uma nova funcionalidade à nossa aplicação. Este passo representa um marco importante no desenvolvimento de qualquer software, onde, após estabelecer as operações fundamentais do CRUD, busca-se expandir a aplicação para atender demandas adicionais ou melhorar a experiência do usuário.

A introdução de novas funcionalidades é um processo natural no ciclo de vida de uma aplicação. A evolução pode ocorrer de várias formas, como o acréscimo de novos recursos que ampliam a capacidade do sistema ou a modernização da interface e da experiência do usuário, sem necessariamente alterar a funcionalidade básica. Esse processo de expansão é essencial para manter a aplicação relevante e útil para seu público-alvo.

Em nosso contexto de um sistema para consultórios médicos, onde já implementamos entidades essenciais como médicos, pacientes e consultas, a adição de uma nova funcionalidade visa trazer uma camada adicional de realismo e utilidade ao sistema. Como exemplo prático, abordaremos a implementação de uma funcionalidade relativamente simples, mas crucial: a remarcação de consultas. Esta funcionalidade simula um cenário comum na realidade, onde pacientes ou médicos precisam alterar a data de uma consulta previamente agendada.

Para implementar esta nova funcionalidade, partiremos do nosso controller

de agendamentos (appointments) e introduziremos uma nova rota de remarcação. Utilizaremos o método HTTP PUT, similar ao empregado na atualização de informações, para modificar a data de uma consulta específica. Esse processo envolverá identificar a consulta por meio de seu ID, receber uma nova data através da requisição e, em seguida, atualizar este registro no banco de dados com a nova data solicitada.

Esse exemplo não apenas demonstra como expandir as funcionalidades de um sistema de maneira prática, mas também ilustra o potencial de evolução baseado na análise de necessidades reais dos usuários. Funcionalidades adicionais podem emergir de necessidades não atendidas ou erros recorrentes, sugerindo ajustes ou melhorias que tornem o sistema mais completo e eficiente.

Na prática, a adição dessa nova rota de remarcação envolve poucos passos no Visual Studio Code, mas representa um avanço significativo na usabilidade do sistema, refletindo um processo comum de realimentação e ajuste que ocorre no desenvolvimento de aplicações reais.

As novas funcionalidades são vitais para o crescimento e a adaptação da aplicação às mudanças de demandas e expectativas dos usuários. Ao incorporar essas evoluções, não apenas melhoramos a aplicação em si, mas também enriquecemos nossa compreensão sobre o desenvolvimento de software como um processo contínuo de aprendizado e adaptação.

Verificaremos a implementação desta nova funcionalidade através do Postman, consolidando nosso aprendizado e garantindo que nossa aplicação continue evoluindo de maneira coerente e funcional.

Verificação via Postman

Nesta última parte, o foco será na verificação via Postman da nova funcionalidade adicionada ao nosso projeto. Este passo é crucial para assegurar que as implementações recentes funcionam conforme o esperado, um processo essencial para manutenção da qualidade e da integridade do software.

O Postman, uma ferramenta amplamente reconhecida para teste de APIs, nos permite simular requisições HTTP e avaliar as respostas da aplicação, sendo fundamental para o teste de novas rotas e funcionalidades. Neste contexto, utilizaremos o Postman para verificar a funcionalidade de remarcação de consultas, recentemente incorporada ao nosso sistema para consultórios médicos.

Para realizar a verificação, é imprescindível que a aplicação esteja em execução localmente, acessível via localhost na porta 3000 e com o banco de dados devidamente conectado. Esse cenário nos permite simular o ambiente de desenvolvimento, assegurando que as requisições feitas via Postman interajam corretamente com a aplicação.

A prática de verificação se inicia com a identificação da consulta a ser remarcada. Utilizamos uma requisição GET para recuperar os detalhes da consulta desejada, incluindo seu ID e a data agendada. Em seguida, procedemos com a criação de uma nova requisição PUT no Postman, destinada à rota de remarcação (`reschedule`), que atualizará a data da consulta com base no ID fornecido e na nova data especificada no corpo da requisição.

Para ilustrar este processo na prática, consideraremos um exemplo onde a nova funcionalidade de remarcação de consultas é testada. Após garantir que a aplicação esteja rodando localmente, acessamos o Postman e iniciamos com uma requisição GET para o endpoint /appointments, recuperando a lista de consultas disponíveis. Selecionamos uma consulta específica, notando seu ID e data atual.

A seguir, criamos uma nova requisição PUT direcionada ao endpoint /appointments/reschedule/{appointmentId}, onde {appointmentId} é substituído pelo ID da consulta selecionada. No corpo da requisição, especificamos um objeto JSON contendo a nova data para a consulta, como {“newDate”: “2024-03-15”}. Ao enviar esta requisição, esperamos que o Postman retorne uma resposta indicando sucesso, com a data da consulta atualizada refletida nos dados.

Finalmente, realizamos outra requisição GET para

/appointments/{appointmentId} para confirmar que a data foi atualizada conforme esperado. Este ciclo de verificação assegura que a funcionalidade de remarcação de consultas opera corretamente, validando nossa implementação e garantindo a integridade do nosso sistema.

Esse processo não apenas confirma a funcionalidade da rota de remarcação, mas também reforça a importância de realizar testes detalhados para cada nova funcionalidade implementada, utilizando o Postman como uma ferramenta essencial para esse fim. Além disso, a prática destaca a necessidade de verificar continuamente o ambiente no qual a aplicação está sendo testada, garantindo que os testes reflitam o comportamento esperado da aplicação em um contexto de desenvolvimento, teste ou produção.

Ao concluir esta etapa de verificação, consolidamos o aprendizado sobre a adição e teste de novas funcionalidades em uma aplicação, utilizando o Postman para assegurar a qualidade e a eficácia das implementações. Este encerramento nos prepara para futuras explorações e aprofundamentos em conceitos de desenvolvimento de APIs e Node.js, prometendo avanços ainda maiores na nossa jornada de aprendizado em Programação II. Aguardo todos nas próximas aulas, prontos para novos desafios e descobertas.

GitHub da Disciplina

Você pode acessar o repositório da disciplina no GitHub a partir do seguinte link: https://github.com/FaculdadeDescomplica/ProgramacaoII. Esse repositório tem como principal objetivo guardar os códigos das aulas práticas da disciplina para aprimorar suas habilidades em vários tópicos, incluindo a criação e consumo de APIs com controle de autenticação utilizando Node.js e utilizando boas práticas de programação e mercado.

Conteúdo Bônus

Para enriquecer ainda mais nosso aprendizado sobre a utilização do Postman na verificação e teste de APIs, recomendo o vídeo “Como usar o Postman para testar API pública/aberta” do canal LuizTools, disponível no YouTube. Esse material é ideal para alunos interessados em aprofundar seus conhecimentos práticos no uso do Postman, abordando técnicas avançadas e dicas valiosas para testar APIs públicas e abertas.

Referências Bibliográficas

Bibliografia Básica:

ELMASRI, R.; NAVATHE, S. B. Sistemas de banco de dados. 7.ed. Pearson: 2018.

MEDEIROS, L. F. de. Banco de dados: princípios e prática. Intersaberes: 2013.

VICCI, C. (Org.). Banco de dados. Pearson: 2014.

Bibliografia Complementar:

CARDOSO, L. da C. Design de aplicativos. Intersaberes: 2022.

JOÃO, B. do N. Usabilidade e interface homem-máquina. Pearson: 2017.

LEAL, G. C. L. Linguagem, programação e banco de dados: guia prático de aprendizagem.

Intersaberes: 2015.

PUGA, S.; FRANÇA, E.; GOYA, M. Banco de dados: implementação em SQL, PL/SQL e Oracle

11g. Pearson: 2013.

SETZER, V. W.; SILVA, F. S. C. Bancos de dados. Blucher: 2005.

Ir para exercício