

Validação da implementação da aplicação

Sejam bem-vindos à nossa aula sobre a validação da implementação da aplicação através do Postman, vamos explorar como realizar operações CRUD - Create, Read, Update, Delete - essenciais para a interação entre usuários e os dados de uma aplicação. Iniciaremos com foco na criação de entidades, uma habilidade crítica que permite testar e assegurar a correta comunicação entre o front-end e o back-end de nossas aplicações. Este módulo não só reforça conceitos teóricos importantes como também nos imerge em práticas essenciais de desenvolvimento de software, preparando o terreno para as próximas fases do CRUD. Vamos garantir que todas as entidades necessárias estejam presentes e adequadamente implementadas, e revisar cuidadosamente nossos códigos e configurações para uma experiência de aprendizado completa e enriquecedora.

Criação de entidades via Postman

De início, vamos abordar um aspecto crucial do desenvolvimento de aplicações: a validação da implementação através do CRUD - Create, Read, Update, Delete. Essas operações representam a fundação para a interação entre usuários e dados armazenados, possibilitando a manipulação eficiente de registros em uma aplicação. Nosso foco estará na criação de entidades utilizando o Postman, uma ferramenta essencial para testar a comunicação entre o front-end e o back-end de nossas aplicações.

Para começar, é importante revisar o Diagrama de Entidade-Relacionamento (ER) do nosso mini projeto, garantindo que todas as entidades necessárias estejam presentes e corretamente implementadas.

Antes de prosseguirmos com a criação das entidades via Postman, há alguns pontos cruciais a serem lembrados. Primeiro, é essencial importar corretamente o arquivo de banco de dados no índice da sua aplicação, assegurando que a conexão com o banco seja estabelecida com sucesso. Além disso, ajustes no uso de imports e a correta exportação dos módulos em seus controladores são necessários para evitar erros comuns que podem ocorrer durante a implementação.

No que diz respeito ao Postman, essa ferramenta nos permite simular solicitações HTTP para testar as respostas da nossa API. Durante a criação de entidades, é fundamental definir claramente as entidades desejadas e os atributos correspondentes. Isso nos permite enviar solicitações POST bem-formadas, incluindo todos os dados necessários no corpo da requisição (body).

Para ilustrar o processo, utilizaremos o exemplo da criação de uma entidade 'Doctor', configurando corretamente o tipo de requisição como POST e especificando no corpo da requisição os atributos necessários, como nome, especialidade e contatos. Este procedimento será replicado para outras entidades do nosso projeto, como 'Appointments', seguindo a mesma lógica de implementação. Por exemplo, ao criar a entidade 'Doctor', o corpo da requisição (body) no Postman poderia ser:

```
{  
  "nome": "Dra. Ana Silva",  
  "especialidade": "Cardiologia",  
  "contatos": "ana.silva@hospital.com"  
}
```

Ao final desta etapa, teremos estabelecido uma base sólida para a criação de entidades através do Postman, preparando-nos para avançar nas próximas operações do CRUD. Essa prática é essencial não apenas para validar nossa implementação, mas também para familiarizar-nos com ferramentas fundamentais de teste de API, cruciais no desenvolvimento

moderno de software.

A seguir, continuaremos explorando as operações de CRUD, focando na leitura de entidades. Certifique-se de ter todas as configurações revisadas e sua aplicação pronta para a próxima fase de validação. Até lá, encorajo todos a revisarem seus códigos e a experimentarem com o Postman, consolidando o conhecimento adquirido até agora.

Leitura de entidades via Postman

Nesta etapa, focaremos na leitura de entidades utilizando o Postman, uma ferramenta essencial no processo de desenvolvimento e validação de aplicações. A leitura de entidades é crucial para assegurar que as informações possam ser recuperadas e exibidas aos usuários, permitindo-lhes interagir eficientemente com o sistema.

Prosseguindo, retomaremos o trabalho iniciado anteriormente, utilizando o Postman para realizar operações de CRUD. Já exploramos a criação de entidades e, agora, focaremos em como recuperar essas informações do banco de dados. Especificamente, vamos aprender a realizar solicitações GET para obter as entidades previamente inseridas, utilizando os IDs gerados automaticamente pelo banco de dados.

Para exemplificar, vamos revisar o processo de leitura da entidade 'Paciente' que criamos anteriormente, utilizando a requisição POST. Com o ID fornecido como resposta à criação da entidade, podemos utilizar este identificador para realizar uma solicitação GET e confirmar que a entidade foi efetivamente inserida no banco de dados. Este procedimento será replicado para outras entidades, como 'Prescrição', seguindo a mesma lógica de implementação.

Para a leitura da entidade 'Paciente', após sua criação, uma requisição GET no Postman poderia ser configurada como:

GET /pacientes/{pacienteld}

Substituindo {patientId} pelo ID real do paciente, essa solicitação nos permite verificar se a entidade foi armazenada corretamente.

Ao realizar solicitações GET, é crucial atentar-se a detalhes como a configuração da URL e a seleção do tipo de requisição adequado. Além disso, o Postman oferece funcionalidades adicionais que podem ser configuradas conforme a necessidade, como autorizações e cabeçalhos específicos, proporcionando uma simulação completa e eficiente para testar APIs.

Aqui, também exploraremos a leitura de todas as entidades de uma determinada categoria, por meio da solicitação GET ALL. Esse tipo de solicitação nos permite visualizar todas as instâncias de uma entidade armazenadas no banco de dados, facilitando a verificação e a gestão dos registros.

Por meio dessa prática, não apenas validamos a inserção correta das entidades no banco de dados, como também consolidamos nossa compreensão sobre a realização de solicitações GET através do Postman. Essa habilidade é fundamental para o desenvolvimento de aplicações robustas e interativas, permitindo uma interação eficaz com os dados.

Em seguida, avançaremos para o próximo passo no processo de validação do CRUD: a atualização de entidades. Essa etapa é essencial para garantir que as aplicações possam responder dinamicamente às necessidades dos usuários, permitindo a modificação de informações armazenadas conforme necessário. Prepare-se para mais uma sessão prática, onde aplicaremos os conhecimentos adquiridos para aprimorar ainda mais nossa aplicação. Incentive-se a experimentar e explorar o Postman, aplicando o que aprendeu sobre leitura de entidades.

Atualização de entidades via Postman

Agora, mergulharemos na atualização de entidades utilizando o Postman, um passo fundamental no ciclo de vida do CRUD (Create, Read, Update, Delete) em aplicações web. Após explorarmos a leitura de entidades

anteriormente, avançaremos para a prática de atualizar os dados previamente cadastrados e verificar essas alterações por meio de solicitações GET.

A atualização de entidades no Postman permite modificar informações inseridas no banco de dados. Utilizaremos os dados preliminares para exemplificar o processo de atualização e subsequente validação das mudanças efetuadas. Esse processo reforça a interdependência entre as diversas operações do CRUD, destacando a importância de uma abordagem prática para o aprendizado.

Iniciaremos acessando o Postman e localizando a coleção de testes que vem sendo construída ao longo das aulas. Vamos adicionar uma nova requisição PUT dentro da pasta correspondente à entidade que desejamos atualizar, por exemplo, "Doutores". Essa requisição será denominada "Update Doctor" e deve ser configurada para usar o método PUT, a fim de mapear corretamente a operação de atualização.

Com base nas informações retornadas por uma requisição READ anterior, utilizaremos o ID específico de um médico para exemplificar o processo de atualização. Por exemplo, se o Dr. João Mendes alterou seu nome, podemos refletir essa mudança na base de dados através de uma requisição PUT, atualizando o campo correspondente no corpo da solicitação.

Suponhamos que precisamos atualizar o e-mail do 'Doctor'. A requisição PUT no Postman seria algo como:

'PUT /doctors/{doctorId}'

com o corpo da requisição:

```
{  
  "email": "ana.silva.novo@hospital.com"  
}
```

Alterando {doctorId} pelo ID correspondente, essa solicitação atualiza o e-mail do doutor na base de dados.

É crucial lembrar de verificar a rota de atualização correta em sua aplicação, garantindo que a requisição PUT seja direcionada para o endpoint adequado, como "localhost:3000/doctors/[ID]", substituindo "[ID]" pelo identificador único do médico a ser atualizado.

Repetiremos o processo de atualização para outras entidades, como consultas (appointments), pacientes (patients) e prescrições (prescriptions), modificando informações relevantes como data, nome, e-mail ou detalhes médicos, conforme necessário. Após cada atualização, uma requisição GET para o ID específico da entidade permitirá validar se as mudanças foram aplicadas corretamente.

Dessa forma, teremos consolidado nossa compreensão sobre a operação de atualização de entidades, um componente vital para a manutenção e gestão eficaz de aplicações dinâmicas. Encorajo todos a praticar esses conceitos no Postman, experimentando com diferentes entidades e campos de dados para fortalecer sua familiaridade com o processo de CRUD.

A seguir, avançaremos para o último componente do ciclo de CRUD: a deleção de entidades. Esse passo final permitirá completar nosso conjunto de operações fundamentais e garantir uma compreensão abrangente do desenvolvimento de aplicações web.

Deleção de entidades via Postman

Concluindo, dedicaremos esta última parte ao processo de deleção de entidades utilizando o Postman. A deleção, ou exclusão, de dados é um componente crítico do conjunto CRUD (Create, Read, Update, Delete), permitindo a remoção de registros indesejados ou obsoletos do banco de dados, um processo essencial para a manutenção de qualquer aplicação.

Neste módulo, utilizaremos o Postman, retomando o trabalho na coleção de testes previamente criada. O foco será a execução de requisições DELETE para diferentes entidades, começando com a recuperação do ID de um registro específico, como o do Dr. João Mendes. Inseriremos uma nova

requisição na pasta de médicos (doctors) dedicada exclusivamente à operação de deleção.

Para configurar corretamente a requisição DELETE, é crucial definir o caminho com base na configuração estabelecida no controller da aplicação. Portanto, a requisição será direcionada para “localhost:3000/doctors/[ID]”, onde “[ID]” é o identificador único do registro a ser deletado. Após a execução da requisição DELETE, um GET subsequente para o mesmo ID deve retornar uma resposta vazia, indicando a efetiva remoção do registro do banco de dados.

Para deletar uma entidade ‘Prescription’ que não é mais necessária, utilizamos:

```
DELETE /prescriptions/{prescriptionId}
```

Essa requisição remove a prescrição específica do banco de dados, onde {prescriptionId} é substituído pelo ID real da prescrição.

Essa prática de deleção não se limita a entidades médicas; será replicada para consultas (appointments), pacientes (patients) e prescrições (prescriptions), abrangendo o escopo completo das operações CRUD. Durante esse processo, também destacaremos funcionalidades adicionais do Postman, como a visualização de respostas em diferentes formatos e a análise de status code, tempo de resposta e tamanho dos dados retornados.

Ao implementar requisições DELETE, é essencial verificar o mapeamento correto e garantir que os IDs utilizados nas requisições correspondam exatamente aos registros desejados. Esse cuidado evita erros comuns e assegura a precisão do teste de validação.

Com a conclusão desta aula, teremos explorado todas as operações fundamentais do CRUD, equipando os alunos com as habilidades necessárias para desenvolver e validar aplicações dinâmicas e interativas. Encorajamos a continuidade da prática com o Postman, ampliando a coleção de testes para abranger novas requisições e garantir a manutenção eficaz da aplicação.

Nas próximas aulas, avançaremos para a inclusão de validações e testes

mais detalhados, consolidando o conhecimento adquirido e preparando os alunos para desafios mais complexos no desenvolvimento de software. A prática contínua e a revisão dos conceitos são fundamentais para o aprimoramento das habilidades em programação e desenvolvimento de aplicações web. Até a próxima aula, dedique-se à exploração e ao teste das funcionalidades de sua aplicação, fortalecendo sua base de conhecimento e preparando-se para os próximos passos em nossa jornada educativa.

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 o seu aprendizado após a nossa aula, gostaria de recomendar um conteúdo bônus, o vídeo “Personalizando erros de validação - Meu Primeiro CRUD em GoLang | AULA #8” do canal HunCoding no YouTube. Esse material é especialmente relevante para aqueles interessados em aprofundar seus conhecimentos em programação e entender melhor as nuances do CRUD, bem como personalização de erros de validação, que é um aspecto crucial para o desenvolvimento de aplicações robustas e confiáveis.

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