

Documentação do Sistema CRUD com API em PHP

Introdução:

O termo **CRUD** é uma abreviação de quatro operações fundamentais de persistência de dados em sistemas de banco de dados: **Create** (Criar), **Read** (Ler), **Update** (Atualizar) e **Delete** (Excluir). Um sistema CRUD permite que os dados sejam inseridos, recuperados, modificados e apagados de um banco de dados. Neste documento, será detalhado cada uma dessas operações, como elas são implementadas dentro de uma API e sua relação com o banco de dados.

O que é CRUD?

CRUD representa as seguintes operações:

1. **Create** - Criar ou adicionar novos registros.
2. **Read** - Ler ou recuperar dados de um banco de dados.
3. **Update** - Atualizar ou modificar dados existentes.
4. **Delete** - Excluir dados de um banco de dados.

Essas quatro operações são básicas em qualquer aplicação que manipule dados de forma persistente e são fundamentais para o funcionamento de sistemas com banco de dados, como sistemas de gestão, e-commerce, sistemas de controle de estoque, etc.

Operações CRUD e sua Implementação na API

1. Create (POST) - Criar Registros

Função na API: A operação de **Create** é responsável por inserir novos dados no banco de dados. Neste sistema, ela é implementada usando o método HTTP **POST**. Quando uma requisição **POST** é recebida pela API, os dados fornecidos pelo usuário (como nome, descrição, preço, etc.) são lidos e um novo registro é criado na tabela **cliente**.

Funcionamento:

1. A API recebe uma requisição **POST** para a URL:
`http://localhost/api/testeApi.php/cliente`.
2. Os dados enviados no corpo da requisição (em formato JSON) são extraídos e armazenados nas variáveis PHP correspondentes.
3. Um comando SQL **INSERT** é montado para inserir os dados na tabela **cliente**.
4. A API responde com uma mensagem de sucesso ou falha, dependendo do resultado da inserção.

Exemplo de corpo na requisição "POST":

```
{  
  
  "nome": "Produto A",  
  
  "descricao": "Descrição do Produto A",  
  
  "preco": 100.50,  
  
  "custo": 50.00,  
  
  "lucro": 50.50,  
  
  "estoque": 20,  
  
  "categoria": "Eletrônicos"  
  
}
```

Exemplo de comando SQL gerado:

```
INSERT INTO cliente (nome, descricao, preco, custo, lucro, estoque, categoria)  
  
VALUES ('Produto A', 'Descrição do Produto A', 100.50, 50.00, 50.50, 20, 'Eletrônicos');
```

2. Read (GET) - Ler Registros

Função na API: A operação de **Read** é usada para consultar e obter dados do banco de dados. No sistema, ela é implementada usando o método HTTP **GET**. A API pode retornar uma lista de todos os registros de clientes ou um cliente específico baseado no ID.

Funcionamento:

1. A API recebe uma requisição **GET** para uma das seguintes URLs:
 - `http://localhost/api/testeApi.php/cliente/list` para listar todos os clientes.
 - `http://localhost/api/testeApi.php/cliente/{id}` para obter detalhes de um cliente específico.
2. Dependendo da URL, a API executa um comando SQL **SELECT**:
 - Para listar todos os clientes: `SELECT * FROM cliente.`
 - Para obter um cliente específico: `SELECT * FROM cliente WHERE idcli = {id}.`
3. A API retorna os dados em formato JSON.

Exemplo de resposta JSON para listar todos os clientes:

```
[
  {
    "id": 1,
    "nome": "Produto A",
    "descricao": "Descrição do Produto A",
    "preco": 100.50,
    "custo": 50.00,
    "lucro": 50.50,
    "estoque": 20,
    "categoria": "Eletrônicos"
  },
  {
    "id": 2,
    "nome": "Produto B",
    "descricao": "Descrição do Produto B",
    "preco": 200.00,
    "custo": 120.00,
    "lucro": 80.00,
    "estoque": 15,
    "categoria": "Móveis"
  }
]
```

3. Update (PUT) - Atualizar Registros

Função na API: A operação de **Update** permite modificar dados já existentes. No sistema, ela é implementada usando o método HTTP **PUT**. Um cliente existente pode ser atualizado fornecendo o ID e os novos valores para os campos que se deseja modificar.

Funcionamento:

1. A API recebe uma requisição **PUT** para a URL:
`http://localhost/api/testeApi.php/cliente/{id}`.
2. Os dados enviados no corpo da requisição (em formato JSON) são usados para atualizar o cliente com o ID especificado.
3. Um comando SQL **UPDATE** é montado e executado no banco de dados.
4. A API retorna uma mensagem indicando sucesso ou erro.

Exemplo de corpo da requisição **PUT**:

```
{
  "nome": "Produto A Atualizado",
  "descricao": "Nova descrição",
  "preco": 150.00,
  "custo": 80.00,
  "lucro": 70.00,
  "estoque": 30,
  "categoria": "Eletrônicos Atualizados"
}
```

Exemplo de comando SQL gerado:

```
UPDATE cliente SET nome = 'Produto A Atualizado', descricao = 'Nova descrição', preco = 150.00, custo = 80.00, lucro = 70.00, estoque = 30, categoria = 'Eletrônicos Atualizados' WHERE idcli = {id};
```

4. Delete (DELETE) - Excluir Registros

Função na API: A operação de **Delete** permite excluir um registro do banco de dados. Ela é implementada usando o método HTTP **DELETE**. Ao fornecer o ID de um cliente, o registro correspondente é excluído.

Funcionamento:

1. A API recebe uma requisição **DELETE** para a URL:
`http://localhost/api/testeApi.php/cliente/{id}`.
2. A API executa um comando SQL **DELETE** para remover o cliente com o ID especificado.
3. A API responde com uma mensagem de sucesso ou erro.

Exemplo de comando SQL gerado:

```
DELETE FROM cliente WHERE idcli = {id};
```

Relacionamento entre a API e o Banco de Dados

A API atua como uma camada intermediária entre o front-end (a interface de usuário) e o banco de dados. Cada operação CRUD está associada a um comando SQL que é executado no banco de dados MySQL. A API recebe as solicitações HTTP, processa os dados e interage com o banco de dados para executar as operações necessárias. Após a execução, a API envia uma resposta de volta ao cliente, geralmente em formato JSON, informando o resultado da operação.

Tabela de Operações CRUD e SQL Correspondentes:

Operação	Método HTTP	Comando SQL	Descrição
Create	POST	INSERT INTO	Insere um novo registro no banco
Read	GET	SELECT * FROM	Lê registros do banco de dados
Update	PUT	UPDATE ... SET	Atualiza um registro existente
Delete	DELETE	DELETE FROM	Remove um registro do banco de dados

Considerações Finais

Este sistema CRUD com API permite a criação, leitura, atualização e exclusão de registros de clientes, interagindo com um banco de dados MySQL. Ele segue a estrutura padrão de APIs RESTful, onde cada método HTTP é mapeado para uma operação de banco de dados.

Essa arquitetura é altamente flexível e pode ser estendida para diferentes tipos de dados ou entidades. As práticas de segurança, como a validação de dados e o tratamento de erros, devem ser sempre implementadas para garantir a robustez e integridade da API.