Para fornecer uma explicação detalhada sobre como utilizar as rotas no Postman, preciso revisar o arquivo server.ts e os arquivos de rotas (CheckinsRoutes.ts e ConfiguracoesRoutes.ts) para entender como as rotas estão configuradas. Isso me permitirá apresentar as URLs completas e explicar cada detalhe. Vou começar lendo o arquivo server.ts.

**blackbox read this file:**

Checkin\_Alunos\_Back/src/server.ts‎

**BLACKBOXAI**

O arquivo server.ts configura as rotas principais da aplicação. As rotas são montadas em caminhos específicos, que são:

* /usuarios para as rotas de usuários
* /alunos para as rotas de alunos
* /configuracoes para as rotas de configurações
* /checkins para as rotas de check-ins

Agora, vou revisar os arquivos de rotas (UsuariosRoutes.ts, AlunosRoutes.ts, ConfiguracoesRoutes.ts, e CheckinsRoutes.ts) para entender quais endpoints estão disponíveis em cada uma dessas rotas. Vou começar lendo o arquivo ConfiguracoesRoutes.ts.

**blackbox read this file:**

Checkin\_Alunos\_Back/src/routes/ConfiguracoesRoutes.ts‎

**BLACKBOXAI**

O arquivo ConfiguracoesRoutes.ts define os seguintes endpoints para a entidade Tb\_Configuracoes:

1. **POST /configuracoes**: Cria uma nova configuração.
2. **GET /configuracoes**: Retorna todas as configurações.
3. **GET /configuracoes/:id**: Retorna uma configuração específica pelo ID.
4. **PUT /configuracoes/:id**: Atualiza uma configuração específica pelo ID.
5. **DELETE /configuracoes/:id**: Deleta uma configuração específica pelo ID.

Agora, vou revisar o arquivo CheckinsRoutes.ts para entender os endpoints disponíveis para a entidade Tb\_Checkins.

**blackbox read this file:**

Checkin\_Alunos\_Back/src/routes/CheckinsRoutes.ts‎

**BLACKBOXAI**

O arquivo CheckinsRoutes.ts define os seguintes endpoints para a entidade Tb\_Checkins:

1. **POST /checkins**: Cria um novo check-in.
2. **GET /checkins**: Retorna todos os check-ins.
3. **GET /checkins/:id**: Retorna um check-in específico pelo ID.
4. **PUT /checkins/:id**: Atualiza um check-in específico pelo ID.
5. **DELETE /checkins/:id**: Deleta um check-in específico pelo ID.

Agora, vou revisar os arquivos de rotas para Usuarios e Alunos para completar a explicação. Vou começar com UsuariosRoutes.ts.

**blackbox read this file:**

Checkin\_Alunos\_Back/src/routes/UsuariosRoutes.ts‎

**BLACKBOXAI**

O arquivo UsuariosRoutes.ts define os seguintes endpoints para a entidade Tb\_Usuarios:

1. **POST /usuarios**: Cria um novo usuário.
2. **GET /usuarios**: Retorna todos os usuários.
3. **GET /usuarios/:id**: Retorna um usuário específico pelo ID.
4. **PUT /usuarios/:id**: Atualiza um usuário específico pelo ID.
5. **DELETE /usuarios/:id**: Deleta um usuário específico pelo ID.

Agora, vou revisar o arquivo AlunosRoutes.ts para entender os endpoints disponíveis para a entidade Tb\_Alunos.

**blackbox read this file:**

Checkin\_Alunos\_Back/src/routes/AlunosRoutes.ts‎

**BLACKBOXAI**

Agora que revisei todos os arquivos de rotas, aqui está um resumo detalhado de como utilizar as rotas no Postman, incluindo as URLs completas e explicações sobre cada detalhe.

**1. Rotas de Usuários**

* **Base URL**: http://localhost:3000/usuarios

**Endpoints:**

* **POST /usuarios**
  + **Descrição**: Cria um novo usuário.
  + **Exemplo de URL**: http://localhost:3000/usuarios
  + **Corpo da Requisição**: JSON com os dados do usuário.
* **GET /usuarios**
  + **Descrição**: Retorna todos os usuários.
  + **Exemplo de URL**: http://localhost:3000/usuarios
* **GET /usuarios/:id**
  + **Descrição**: Retorna um usuário específico pelo ID.
  + **Exemplo de URL**: http://localhost:3000/usuarios/1 (substitua 1 pelo ID desejado)
* **PUT /usuarios/:id**
  + **Descrição**: Atualiza um usuário específico pelo ID.
  + **Exemplo de URL**: http://localhost:3000/usuarios/1 (substitua 1 pelo ID desejado)
  + **Corpo da Requisição**: JSON com os dados atualizados do usuário.
* **DELETE /usuarios/:id**
  + **Descrição**: Deleta um usuário específico pelo ID.
  + **Exemplo de URL**: http://localhost:3000/usuarios/1 (substitua 1 pelo ID desejado)

**2. Rotas de Alunos**

* **Base URL**: http://localhost:3000/alunos

**Endpoints:**

* **POST /alunos**
  + **Descrição**: Cria um novo aluno.
  + **Exemplo de URL**: http://localhost:3000/alunos
  + **Corpo da Requisição**: JSON com os dados do aluno.
* **GET /alunos**
  + **Descrição**: Retorna todos os alunos.
  + **Exemplo de URL**: http://localhost:3000/alunos
* **GET /alunos/:matricula**
  + **Descrição**: Retorna um aluno específico pela matrícula.
  + **Exemplo de URL**: http://localhost:3000/alunos/123 (substitua 123 pela matrícula desejada)
* **PUT /alunos/:matricula**
  + **Descrição**: Atualiza um aluno específico pela matrícula.
  + **Exemplo de URL**: http://localhost:3000/alunos/123 (substitua 123 pela matrícula desejada)
  + **Corpo da Requisição**: JSON com os dados atualizados do aluno.
* **DELETE /alunos/:matricula**
  + **Descrição**: Deleta um aluno específico pela matrícula.
  + **Exemplo de URL**: http://localhost:3000/alunos/123 (substitua 123 pela matrícula desejada)

**3. Rotas de Configurações**

* **Base URL**: http://localhost:3000/configuracoes

**Endpoints:**

* **POST /configuracoes**
  + **Descrição**: Cria uma nova configuração.
  + **Exemplo de URL**: http://localhost:3000/configuracoes
  + **Corpo da Requisição**: JSON com os dados da configuração.
* **GET /configuracoes**
  + **Descrição**: Retorna todas as configurações.
  + **Exemplo de URL**: http://localhost:3000/configuracoes
* **GET /configuracoes/:id**
  + **Descrição**: Retorna uma configuração específica pelo ID.
  + **Exemplo de URL**: http://localhost:3000/configuracoes/1 (substitua 1 pelo ID desejado)
* **PUT /configuracoes/:id**
  + **Descrição**: Atualiza uma configuração específica pelo ID.
  + **Exemplo de URL**: http://localhost:3000/configuracoes/1 (substitua 1 pelo ID desejado)
  + **Corpo da Requisição**: JSON com os dados atualizados da configuração.
* **DELETE /configuracoes/:id**
  + **Descrição**: Deleta uma configuração específica pelo ID.
  + **Exemplo de URL**: http://localhost:3000/configuracoes/1 (substitua 1 pelo ID desejado)

**4. Rotas de Check-ins**

* **Base URL**: http://localhost:3000/checkins

**Endpoints:**

* **POST /checkins**
  + **Descrição**: Cria um novo check-in.
  + **Exemplo de URL**: http://localhost:3000/checkins
  + **Corpo da Requisição**: JSON com os dados do check-in.
* **GET /checkins**
  + **Descrição**: Retorna todos os check-ins.
  + **Exemplo de URL**: http://localhost:3000/checkins
* **GET /checkins/:id**
  + **Descrição**: Retorna um check-in específico pelo ID.
  + **Exemplo de URL**: http://localhost:3000/checkins/1 (substitua 1 pelo ID desejado)
* **PUT /checkins/:id**
  + **Descrição**: Atualiza um check-in específico pelo ID.
  + **Exemplo de URL**: http://localhost:3000/checkins/1 (substitua 1 pelo ID desejado)
  + **Corpo da Requisição**: JSON com os dados atualizados do check-in.
* **DELETE /checkins/:id**
  + **Descrição**: Deleta um check-in específico pelo ID.
  + **Exemplo de URL**: http://localhost:3000/checkins/1 (substitua 1 pelo ID desejado)

**Considerações Finais**

* **URL Base**: A URL base para todas as requisições é http://localhost:3000, onde 3000 é a porta padrão configurada no servidor. Se você estiver usando uma porta diferente, substitua 3000 pela porta correta.
* **Formato do Corpo da Requisição**: Para as requisições que requerem um corpo (POST e PUT), certifique-se de definir o cabeçalho Content-Type como application/json no Postman.

Com essas informações, você deve ser capaz de interagir com a API usando o Postman de forma eficaz.