Segue uma **documentação detalhada** para que o desenvolvedor do frontend consiga consumir o FornecedorController via fetch sem enfrentar problemas de **400 Bad Request** ou erros de validação.  
O foco é **clareza dos parâmetros**, **formato esperado do JSON** e **regras de validação** que precisam ser respeitadas.

**📄 Documentação da API — FornecedorController**

**Base URL**:

/api/fornecedores

**Formato de dados:**

* **Request Body:** JSON (Content-Type: application/json)
* **Response Body:** JSON (application/json)
* **Charset recomendado:** UTF-8

**Autenticação:**

* Se o sistema estiver protegido por Spring Security, incluir credentials: "include" no fetch para enviar cookies de sessão e CSRF token.

**CSRF Protection:**

* Enviar o header X-XSRF-TOKEN (obtido do cookie XSRF-TOKEN) em requisições POST, PUT e DELETE.

**1️⃣ Listar nomes filtrados**

**GET** /nomes

**Parâmetros de Query**

| **Nome** | **Tipo** | **Obrigatório** | **Descrição** |
| --- | --- | --- | --- |
| nome | String | Não | Parte do nome do fornecedor para filtrar. Se não informado, retorna todos. |

**Resposta (200 OK)**

[

{

"nomeFornecedor": "Fornecedor Exemplo"

},

{

"nomeFornecedor": "Outro Fornecedor"

}

]

📌 **Observações:**

* Apenas o campo nomeFornecedor é retornado.
* Útil para autocomplete no frontend.

**2️⃣ Listar fornecedores paginados**

**GET** /

**Parâmetros de Query**

| **Nome** | **Tipo** | **Obrigatório** | **Valor Padrão** | **Descrição** |
| --- | --- | --- | --- | --- |
| page | int | Não | 0 | Número da página (0-indexado). |
| size | int | Não | 20 | Quantidade de itens por página. |

**Resposta (200 OK)**

{

"content": [

{

"nomeFornecedor": "Fornecedor A",

"cnpj": "12345678000199"

}

],

"pageNumber": 0,

"pageSize": 20,

"totalElements": 1,

"totalPages": 1,

"last": true

}

📌 **Observações:**

* A ordenação é sempre **decrescente pelo nome** (nomeFornecedor).
* Campo content contém a lista da página atual.

**3️⃣ Adicionar fornecedor**

**POST** /

**Corpo da Requisição (JSON)**

{

"nomeFornecedor": "Fornecedor Teste",

"cnpj": "12345678000199"

}

**Validações**

* nomeFornecedor: obrigatório, máx. 255 caracteres.
* cnpj: obrigatório, formato válido (CNPJ).

**Resposta (201 Created)**

{

"nomeFornecedor": "Fornecedor Teste",

"cnpj": "12345678000199"

}

📌 **Possíveis erros:**

* **400 Bad Request**: Campos inválidos ou ausentes.
* **409 Conflict**: CNPJ já existente (dependendo da implementação do validarInsercao).

**4️⃣ Buscar fornecedor por CNPJ**

**GET** /{cnpj}

**Exemplo de Requisição**

GET /api/fornecedores/12345678000199

**Resposta (200 OK)**

{

"nomeFornecedor": "Fornecedor Teste",

"cnpj": "12345678000199"

}

📌 **Possíveis erros:**

* **404 Not Found**: CNPJ não encontrado.

**5️⃣ Deletar fornecedor**

**DELETE** /{cnpj}

**Exemplo de Requisição**

DELETE /api/fornecedores/12345678000199

**Resposta (204 No Content)**

Sem corpo na resposta.

📌 **Possíveis erros:**

* **404 Not Found**: Fornecedor não encontrado.
* **409 Conflict**: Se existir relação com outros registros e não puder excluir.

**6️⃣ Atualizar fornecedor**

**PUT** /{cnpj}

**Corpo da Requisição (JSON)**

{

"nomeFornecedor": "Fornecedor Atualizado",

"cnpj": "98765432000188"

}

**Validações**

* nomeFornecedor: obrigatório, máx. 255 caracteres.
* cnpj: obrigatório, formato válido.
* O CNPJ de path (/{cnpj}) identifica o registro a atualizar.
* O cnpj no body pode ser diferente, caso queira alterá-lo.

**Resposta (200 OK)**

{

"nomeFornecedor": "Fornecedor Atualizado",

"cnpj": "98765432000188"

}

📌 **Possíveis erros:**

* **400 Bad Request**: JSON malformado ou campos inválidos.
* **404 Not Found**: CNPJ original não encontrado.
* **409 Conflict**: CNPJ novo já existente.