

Categorias de produtos

Índice de endpoints · Modelos

API-015 — Listar

GET `/api/categorias-produto`

GET `/api/product-categories`

Finalidade: Listar.

Fonte: `src/main/java/com/wester/storage/controller/CategoriaProdutoController.java:39` — `CategoriaProdutoController.listar`.

Autenticação, autorização e perfis: `hasAnyAuthority('SCREEN_PRODUCTS', 'SCREEN_REPORTS')`
Consulte a correspondência entre autoridades e perfis em Autenticação.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `List<CategoriaProdutoDTO>`. Campos em Modelos.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
[ ]
```

json

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em Erros.

Regras de negócio e dependências: `CategoriaProdutoService.listar`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/categorias-produto" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-016 — Criar

POST `/api/categorias-produto`

POST `/api/product-categories`

Finalidade: Criar.

Fonte: `src/main/java/com/wester/storage/controller/CategoriaProdutoController.java:45` — `CategoriaProdutoController.criar`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_PRODUCTS')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em Autenticação.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo da requisição: `CategoriaProdutoRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em Modelos.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "codigo": "Exemplo",
  "descricao": "Exemplo"
}
```

json

Resposta de sucesso: `CategoriaProdutoDTO`. Campos em Modelos.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "codigo": "Exemplo",
  "descricao": "Exemplo",
  "totalProdutos": 1
}
```

json

Códigos HTTP: 201, 400, 401, 403. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em Erros.

Validações e erros identificados na cadeia local do serviço:

- Código de categoria já existe: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: CategoriaProdutoService.criar. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/categorias-produto" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/json' -d '{...}'
```

API-017 — Atualizar

PUT /api/categorias-produto/{id}

PUT /api/product-categories/{id}

Finalidade: Atualizar.

Fonte: src/main/java/com/wester/storage/controller/CategoriaProdutoController.java:55 — CategoriaProdutoController.atualizar.

Autenticação, autorização e perfis: hasAuthority('SCREEN_PRODUCTS') and hasRole('FULL_ACCESS') Consulte a correspondência entre autoridades e perfis em Autenticação.

Headers: Authorization: Bearer <TOKEN>. Accept: application/json recomendado. Content-Type: application/json para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
id	path	Long	Sim	—

Corpo da requisição: CategoriaProdutoRequestDTO; obrigatório. Validação Jakarta ativada por @Valid. Consulte os campos e restrições em Modelos.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

json

```
{
  "codigo": "Exemplo",
  "descricao": "Exemplo"
}
```

Resposta de sucesso: `CategoriaProdutoDTO`. Campos em Modelos.

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "id": 1,
  "codigo": "Exemplo",
  "descricao": "Exemplo",
  "totalProdutos": 1
}
```

Códigos HTTP: `200`, `400`, `401`, `403`, `404`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em Erros.

Validações e erros identificados na cadeia local do serviço:

- Código de categoria já existe: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: `CategoriaProdutoService.atualizar`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X PUT "${BASE_URL}/api/categorias-produto/1" -H "Authorization: Bearer ${TOKEN}" -H "
```

API-018 — Excluir

DELETE `/api/categorias-produto/{id}`

DELETE `/api/product-categories/{id}`

Finalidade: Excluir.

Fonte: `src/main/java/com/wester/storage/controller/CategoriaProdutoController.java:61` — `CategoriaProdutoController.excluir`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_PRODUCTS')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em Autenticação.

Headers: `Authorization: Bearer <TOKEN>` . `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: sem corpo quando 204.

Códigos HTTP: `204` , `400` , `401` , `403` , `404` . Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em Erros.

Validações e erros identificados na cadeia local do serviço:

- Não é possível excluir: a categoria possui

Regras de negócio e dependências: `CategoriaProdutoService.excluir` . Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X DELETE "${BASE_URL}/api/categorias-produto/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-019 — Vincular

`PUT` `/api/categorias-produto/{id}/produtos`

`PUT` `/api/product-categories/{id}/produtos`

Finalidade: Vincular.

Fonte: `src/main/java/com/wester/storage/controller/CategoriaProdutoController.java:70` — `CategoriaProdutoController.vincular` .

Autenticação, autorização e perfis: `hasAuthority('SCREEN_PRODUCTS')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em Autenticação.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo da requisição: `CategoriaProdutoVinculoRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em Modelos.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "produtoIds": [
    1
  ]
}
```

json

Resposta de sucesso: `CategoriaProdutoDTO`. Campos em Modelos.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "codigo": "Exemplo",
  "descricao": "Exemplo",
  "totalProdutos": 1
}
```

json

Códigos HTTP: `200`, `400`, `401`, `403`, `404`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em Erros.

Validações e erros identificados na cadeia local do serviço:

- Um ou mais produtos informados não foram encontrados.

Regras de negócio e dependências: `CategoriaProdutoService.vincularProdutos`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X PUT "${BASE_URL}/api/categorias-produto/1/produtos" -H "Authorization: Bearer ${TOKEN}"
```

API-020 — Desvincular

DELETE `/api/categorias-produto/{id}/produtos`

DELETE `/api/product-categories/{id}/produtos`

Finalidade: Desvincular.

Fonte: `src/main/java/com/wester/storage/controller/CategoriaProdutoController.java:79` — `CategoriaProdutoController.desvincular`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_PRODUCTS')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em Autenticação.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo da requisição: `CategoriaProdutoVinculoRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em Modelos.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "produtoIds": [
    1
  ]
}
```

Resposta de sucesso: `CategoriaProdutoDTO`. Campos em Modelos.

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "id": 1,
  "codigo": "Exemplo",
  "descricao": "Exemplo",
  "totalProdutos": 1
}
```

Códigos HTTP: 200, 400, 401, 403, 404. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em Erros.

Validações e erros identificados na cadeia local do serviço:

- Um ou mais produtos informados não foram encontrados.

Regras de negócio e dependências: `CategoriaProdutoService.desvincularProdutos`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X DELETE "${BASE_URL}/api/categorias-produto/1/produtos" -H "Authorization: Bearer $"
```