

Documentação - Módulo Identidade



Regras de Negócio - Módulo **pc-identidade**



Objetivo

Garantir o cadastro correto e consistente dos vendedores (sellers) no sistema, assegurando a validade dos dados fornecidos, unicidade do nome e formato correto do CNPJ.



Entrada de Dados

Atributos	Obrigatório	Tipo	Restrições/Validações	Mensagem de Erro (Exemplo)
seller_id	sim	string	Identificador único do seller no sistema. Deve ser exclusivo, conter apenas caracteres alfanuméricos minúsculos (lowercase).	"O seller_id é obrigatório."
nome_fantasia	sim	string	Nome de exibição único no sistema. Deve ser exclusivo.	"O nome_fantasia já está em uso por outro seller."
cnpj	sim	string	CNPJ deve conter 14 dígitos numéricos.	"O CNPJ é obrigatório e deve conter 14 dígitos válidos."

Regras de Negócio Específicas

REGRA-SELLER-001: Validação de CNPJ

- O CNPJ fornecido deve conter exatamente 14 dígitos numéricos.
- Não deve conter pontos, barras ou traços (formato puro).

Mensagem de erro exemplo:

"O CNPJ é inválido. Deve conter exatamente 14 dígitos numéricos."

REGRA-SELLER-002: Nome Fantasia Único

- O nome_fantasia deve ser exclusivo em todo o sistema.
- Tentativas de cadastro com nomes já existentes devem ser rejeitadas.

Mensagem de erro exemplo:

"O nome_fantasia informado já está cadastrado. Escolha outro."

REGRA-SELLER-003: Seller_id como Identificador Único

- O seller_id deve ser exclusivo em todo o sistema.
- Tentativas de cadastro com seller_id já existentes devem ser rejeitadas.
- O seller_id deve conter apenas **caracteres alfanuméricos minúsculos (lowercase)**.

Mensagem de erro exemplo:

"O seller_id informado já está cadastrado. Escolha outro."

"O seller_id deve conter apenas letras minúsculas e números, sem espaços ou caracteres especiais."

Modelagem da Coleção: **sellers**

Estrutura do Documento

Tt Campo	Tipo	Tt Obrigatório	Tt Descrição	Tt Restrições
_id	ObjectId ▾	Sim	Identificador único gerado automaticamente pelo MongoDB.	—
seller_id	String ▾	Sim	ID único do seller no sistema.	Alfanumérico, lowercase, sem espaços ou símbolos.
nome_fantasia	String ▾	Sim	Nome fantasia do seller. Deve ser único.	Mínimo 3 caracteres, não vazio.
cnpj	String ▾	Sim	CNPJ do seller (14 dígitos, sem pontuação).	Apenas dígitos, exatamente 14 caracteres.
created_at	DateTime ▾	Sim	Data de criação do documento.	Preenchido de forma automática.
updated_at	DateTime ▾	Sim	Data da última atualização.	Atualizado a cada modificação.

Índices e Restrições

- **seller_id**: índice único – utilizado como chave de identificação do seller no sistema;
- **cnpj**: sem restrição de unicidade, mas pode ter índice simples para acelerar buscas (opcional);
- **nome_fantasia**: índice simples opcional, se for utilizado em buscas frequentes.



Exemplo de Documento

```
{
  "_id": ObjectId("666a30d74e0f6c7123456789"),
  "seller_id": "loja123",
  "nome_fantasia": "Loja do João",
  "cnpj": "12345678000199",
  "created_at": "2025-06-04T14:00:00Z",
  "updated_at": "2025-06-04T14:00:00Z"
}
```



Endpoints

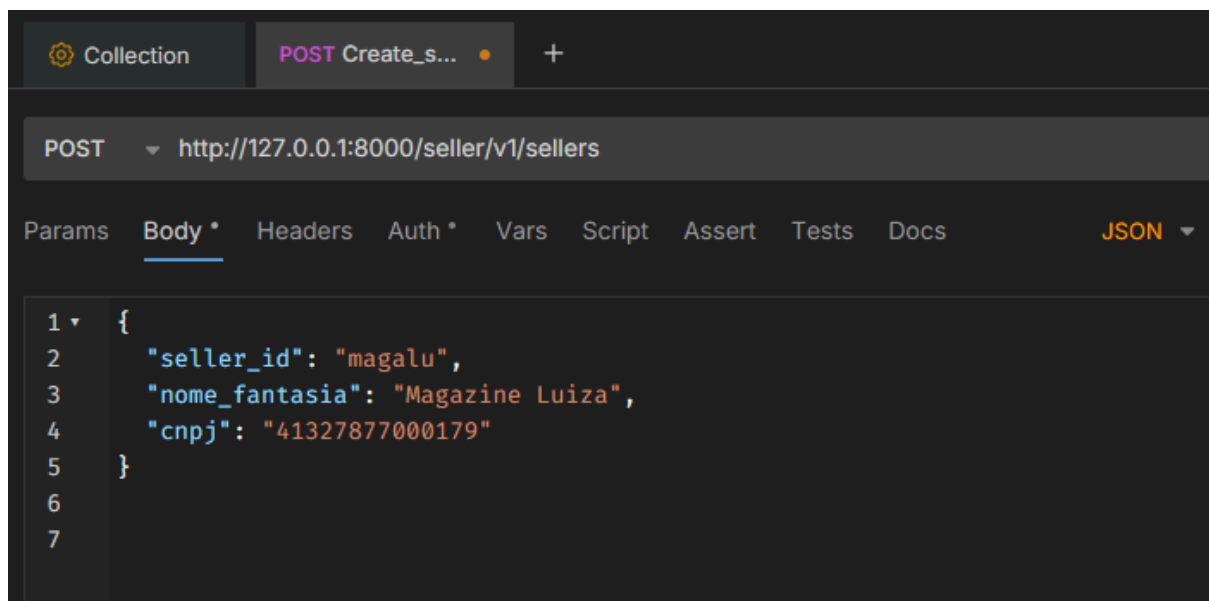
1. Criar Seller

POST /seller/v1/sellers

Descrição: Cria um novo seller no sistema.

Utilizar **METHOD** do tipo **POST**.

Abaixo exibiremos o método aberto através do **Bruno**. A chamada da API ocorre da seguinte forma:



Parâmetros de Entrada

Tipo Campo	Tipo	Descrição
seller_id	String	Identificador único do seller. Apenas caracteres alfanuméricos e todos em minúsculo (lowercase). Não pode se repetir.
nome_fantasia	String	Nome de exibição único no sistema. Deve ser exclusivo.
cnpj	String	Deve conter 14 dígitos numéricos.

Saída esperada (**Resposta** 201 Created):

```
Response Headers 5 Timeline Tests
1 {
2   "id": "06823386-ad70-7378-8000-ef849843c834",
3   "created_at": "2025-05-13T12:17:46.839000Z",
4   "updated_at": null,
5   "created_by": null,
6   "updated_by": null,
7   "seller_id": "teste",
8   "nome_fantasia": "teste",
9   "cnpj": "73781453000176"
10 }
```

2. Listar Todos os Sellers

GET /seller/v1/sellers

Descrição: Retorna todos os sellers cadastrados.

Utilizar **METHOD** do tipo **GET**.

Saída esperada (**Resposta** 200 OK):

```
Response Headers Timeline Tests
1 {
2   "meta": {
3     "page": {
4       "limit": 50,
5       "offset": 0,
6       "count": 1,
7       "max_limit": 100
8     },
9     "links": {
10      "previous": "/seller/v1/sellers?_offset=0&_limit=50",
11      "self": "/seller/v1/sellers?_offset=0&_limit=50",
12      "next": null
13    }
14  },
15  "results": [
16    {
17      "id": "06823ed7-7ca2-7946-8000-41c8f39aeb2d",
18      "created_at": "2025-05-14T01:10:15.789000Z",
19      "updated_at": null,
20      "created_by": null,
21      "updated_by": null,
22      "seller_id": "magalu",
23      "nome_fantasia": "Magazine Luiza",
24      "cnpj": "41327877000179"
25    }
26  ]
27 }
```

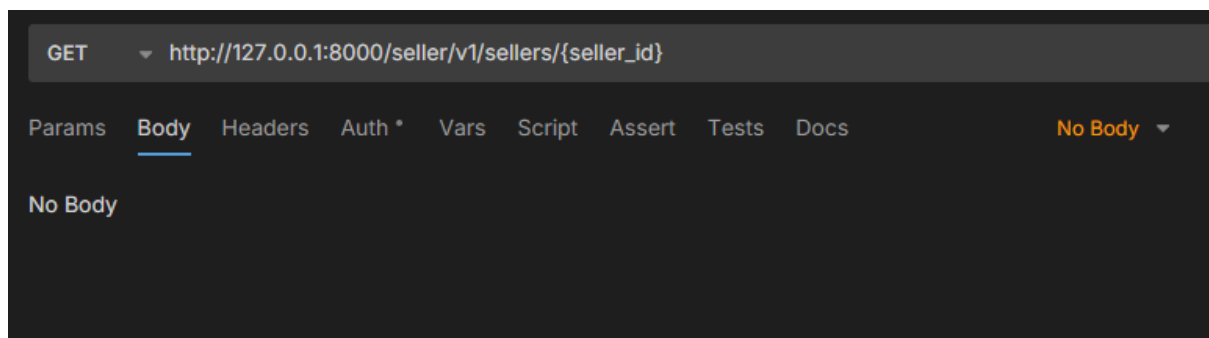
3. 🔍 Obter Seller por ID

GET /seller/v1/sellers/{seller_id}

Descrição: Busca um seller específico pelo seu **seller_id**.

Utilizar **METHOD** do tipo **GET**.

Abaixo exibiremos o método aberto através do **Bruno**. A chamada da API ocorre da seguinte forma:



Saída esperada (**Resposta** 200 OK):

```
Response Headers5 Timeline Tests 200 OK
1 {
2   "id": "06823ed7-7ca2-7946-8000-41c8f39aeb2d",
3   "created_at": "2025-05-14T01:10:15.789000Z",
4   "updated_at": null,
5   "created_by": null,
6   "updated_by": null,
7   "seller_id": "magalu",
8   "nome_fantasia": "Magazine Luiza",
9   "cnpj": "41327877000179"
10 }
```

4. ✎ Atualizar Seller

PATCH `/seller/v1/sellers/{seller_id}`

Descrição: Atualiza parcialmente os dados de um seller.

Utilizar **METHOD** do tipo **PATCH**.

Abaixo exibiremos o método aberto através do **Bruno**. A chamada da API ocorre da seguinte forma:

```
PATCH http://127.0.0.1:8000/seller/v1/sellers/{seller_id}
Params Body * Headers Auth * Vars Script Assert Tests Docs JSON Prettify
1 {
2   "nome_fantasia": "teste update",
3   "cnpj": "85151257000105"
4 }
5
6 |
```

Parâmetros de Entrada

Tipo Campo	Tipo	Descrição
seller_id	String	Deve ser informado no endpoint para identificar o seller a ser atualizado.
nome_fantasia	String	Nome de exibição único no sistema. Deve ser exclusivo.
cnpj	String	Deve conter 14 dígitos numéricos.

Saída esperada (**Resposta 200 OK**):

```
Response Headers Timeline Tests
1 {
2   "id": "06823386-ad70-7378-8000-ef849843c834",
3   "created_at": "2025-05-13T12:17:46.839000Z",
4   "updated_at": "2025-05-13T12:22:39.717000Z",
5   "created_by": null,
6   "updated_by": null,
7   "seller_id": "teste",
8   "nome_fantasia": "teste update",
9   "cnpj": "85151257000105"
10 }
```

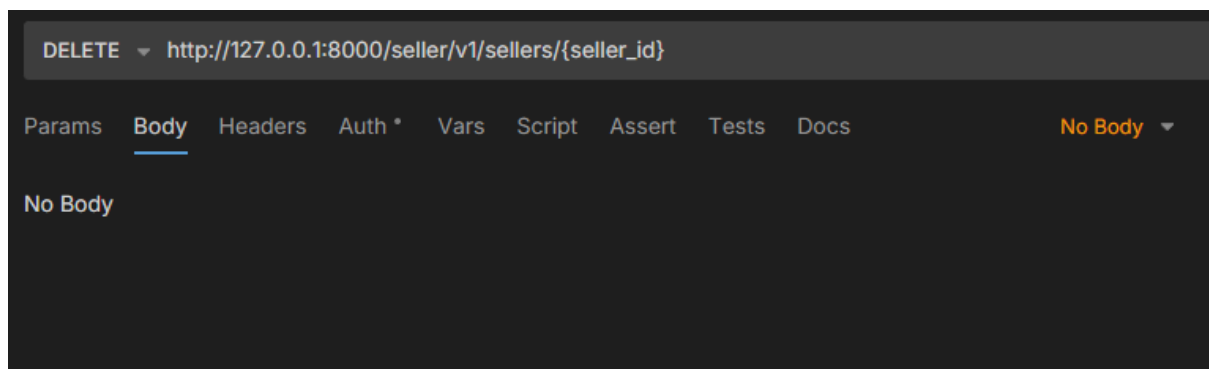
5. 🗑 Deletar Seller

DELETE /seller/v1/sellers/{seller_id}

Descrição: Remove um seller do sistema a partir do seu 'seller_id'.

Utilizar **METHOD** do tipo **DELETE**.

Abaixo exibiremos o método aberto através do **Bruno**. A chamada da API ocorre da seguinte forma:



Saída esperada (**Resposta 204 No Content**):

Response

Headers 4

Timeline

Tests

204 No Content

1