

# Test Cases para API `/assets`

## 1. POST `/assets` – Criação de Asset

### TC01 – Criar asset válido

Pré-condição: API disponível

Entrada: `{ "id": "1", "name": "Customer Database", "type": "DATABASE", "tags": ["customer", "pii"], "taxonomy": ["DATA", "CRM"], "status": "ACTIVE", "createdAt": "2026-01-17T12:00:00Z" }`

- Resultado esperado: **201 Created**; corpo da resposta contém asset criado com todos os campos obrigatórios

### TC02 – Criar asset sem nome

Pré-condição: API disponível

Entrada: `"name": ""`

- Resultado esperado: **400 Bad Request**; mensagem de erro `"name is required"`

### TC03 – Criar asset com tags malformadas

Pré-condição: API disponível

Entrada: `"tags": "finance, pii"` (string em vez de array)

- Resultado esperado: **400 Bad Request**; mensagem de erro `"tags must be array"`

### TC04 – Criar asset com tipo inválido

Pré-condição: API disponível

Entrada: `"type": "INVALID_TYPE"`

- Resultado esperado: **400 Bad Request**; mensagem de erro `"invalid type"`

## 2. GET `/assets` – Listagem de Assets

### TC05 – Listar todos os assets ativos

Pré-condição: Existem assets ativos

Entrada: `GET /assets`

- Resultado esperado: **200 OK**; todos os itens retornados possuem `"status": "ACTIVE"`

### TC06 – Listar assets inativos

Pré-condição: Existem assets inativos

Entrada: `GET /assets?status=INACTIVE`

- Resultado esperado: **200 OK**; todos os itens retornados possuem `"status": "INACTIVE"`

#### TC07 – Listar assets por tag

Pré-condição: Existem assets com tag `pii`

Entrada: `GET /assets?tag=pii`

- Resultado esperado: **200 OK**; todos os itens retornados possuem `"tags": ["pii", ...]`

#### TC08 – Listar assets por taxonomy

Pré-condição: Existem assets com taxonomy `FINANCE`

Entrada: `GET /assets?taxonomy=finance`

- Resultado esperado: **200 OK**; todos os itens retornados possuem `"taxonomy": ["FINANCE"]`

### 3. GET /assets/:id – Consulta de Asset por ID

#### TC09 – Consultar asset válido

Pré-condição: Asset com `id=2` existe

Entrada: `GET /assets/2`

- Resultado esperado: **200 OK**; corpo da resposta contém todos os campos obrigatórios; `"status": "ACTIVE"` ou `"INACTIVE"`

#### TC10 – Consultar asset inexistente

Pré-condição: Asset com `id=9999` não existe

Entrada: `GET /assets/9999`

- Resultado esperado: **404 Not Found**; corpo da resposta contém `"error": "Asset not found"`

### 4. Validações de Performance

#### TC11 – Tempo de resposta POST

Pré-condição: API disponível

Entrada: `POST /assets` válido

- Resultado esperado: Tempo de resposta < 500ms

#### TC12 – Tempo de resposta GET

Pré-condição: API disponível

Entrada: `GET /assets`

- Resultado esperado: Tempo de resposta < 500ms