

Smart API

Manual de Operação

Manual de Operação **Smart API**

Versão do Manual: OM.SMARTAPI.2025.06.POR.V2

Padtec S/A provê aos clientes um completo suporte técnico e serviço. Por favor, sintase livre para contatar nossos escritórios ou sede da empresa.

Copyright © 2025 Padtec S/A

Todos os direitos reservados

Nenhuma parte deste documento pode ser reproduzida ou transmitida de alguma forma ou por terceiros sem autorização escrita da Padtec S/A.

Marcas Registradas

Padtec, LightPad, MultiPad, Pados, LightPad T-Era, LightPad Max, Padtec NMS Plus, Smart Site, Smart Workspot e Padtec Soluções são marcas registradas da Padtec S/A.

As demais marcas registradas e nomes comerciais mencionados neste manual são características de seus respectivos proprietários.

Aviso

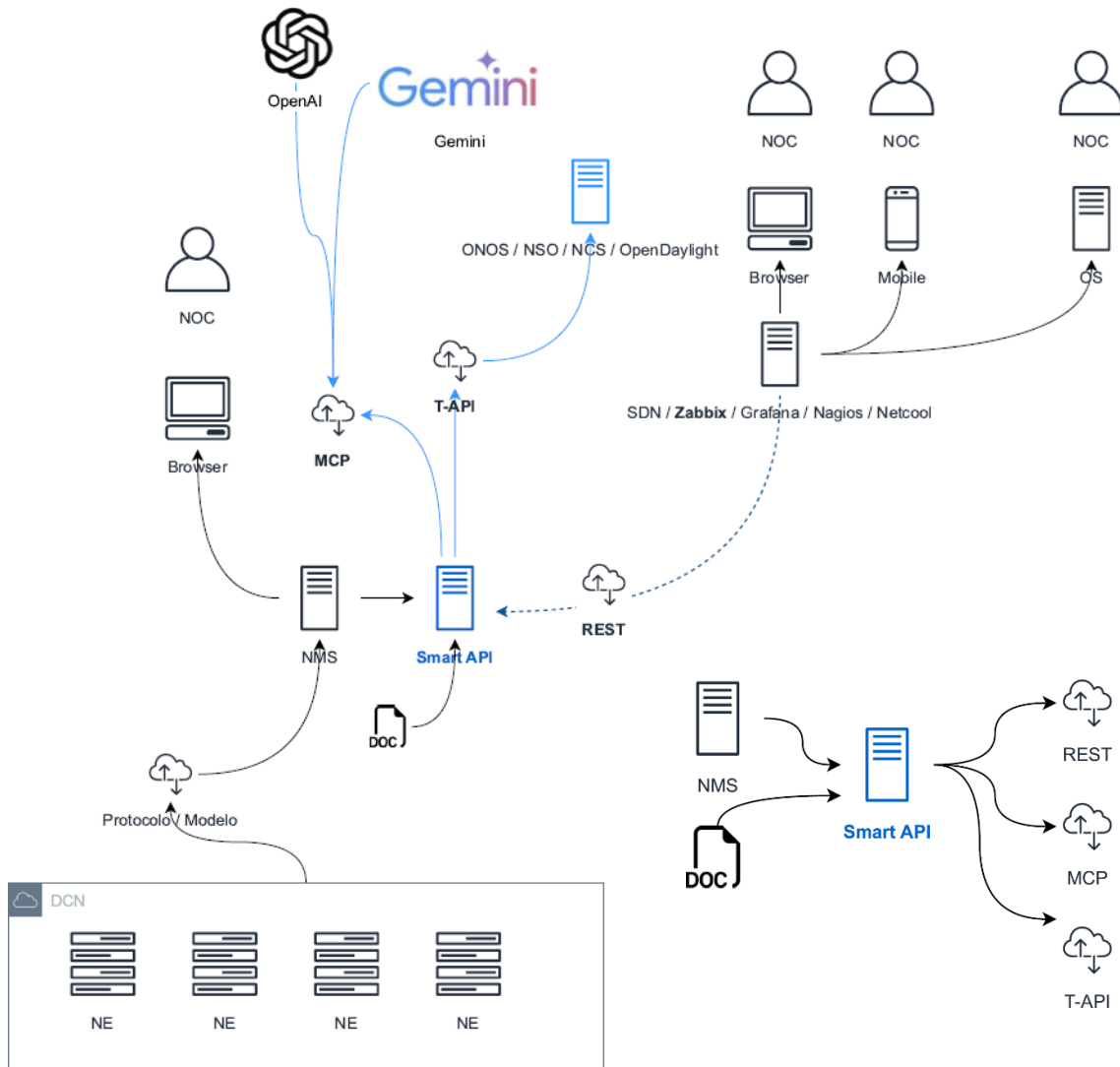
As informações contidas neste manual estão sujeitas a alterações sem aviso prévio. Todos os esforços foram utilizados na preparação deste documento para assegurar a exatidão do conteúdo, mas todas as declarações, informações e recomendações neste documento não constituem a garantia de alguma classe, expressa ou implicada.

Índice

1.	Apresentação Smart API	5
2.	Acessando Swagger	7
2.1.	Gerando Token de Autorização	8
3.	Organização da API.....	9
4.	Alarm.....	11
4.1.	Alarm - Capability.....	11
4.2.	Alarm - Count	12
4.3.	Alarm - State.....	12
5.	Inventory	14
5.1.	Inventory - Capability	14
5.2.	Inventory - Count.....	15
5.3.	Inventory - State	15
6.	Measures.....	17
6.1.	Measures - Capability	17
6.2.	Measures - Count.....	18
6.3.	Measures - State	19

1. Apresentação Smart API

O **Smart API** é um software que disponibiliza uma interface **REST** corporativa, projetada para centralizar, em um único ponto de acesso, informações essenciais sobre os dispositivos, como alarmes, inventário e medições. Integrando-se de forma transparente a múltiplos sistemas de gerenciamento **NMS+**, a API consolida dados operacionais em tempo real e disponibiliza documentação **DOC** detalhada dos equipamentos.



REST: Representational State Transfer (REST) é uma arquitetura amplamente utilizada para o desenvolvimento de APIs em sistemas de telecomunicações, permitindo a comunicação entre diferentes plataformas, sistemas de gerenciamento de rede e aplicações operacionais.

Por meio de uma API REST, é possível realizar a troca de dados via protocolo HTTP, utilizando métodos padronizados como:

No contexto de telecomunicações, as APIs REST são fundamentais para automatizar operações, integrar sistemas com novas plataformas, e viabilizar soluções como gerenciamento centralizado, orquestração de serviços e monitoramento inteligente.

MCP: Model Context Protocol (MCP) é um padrão aberto e estrutura de código aberto que permite que modelos de IA, como LLMs, interajam com ferramentas, sistemas e dados externos de forma padronizada e segura. Ele segue uma arquitetura cliente-servidor, em que aplicações de IA (clientes) se conectam a servidores MCP que expõem recursos como APIs ou arquivos locais.

O protocolo simplifica a criação de agentes inteligentes e fluxos de trabalho, oferecendo flexibilidade na integração com diferentes ferramentas e provedores.

Nos ambientes de telecom, a IA pode utilizar o MCP para interpretar informações provenientes de alarmes, indicadores de desempenho e registros de rede, auxiliando no diagnóstico de possíveis falhas ou comportamentos anômalos. Além disso, o protocolo pode ser usado para realizar consultas em grandes volumes de dados, apoiando decisões operacionais com base em informações históricas ou em tempo real.

O MCP pode servir como interface entre plataformas de IA como ChatGPT ou Gemini e os sistemas de telecomunicações, permitindo que essas ferramentas sejam utilizadas como assistentes inteligentes no Network Operations Center (NOC). Nesse contexto, a IA não controla diretamente a rede, mas atua como apoio analítico e informacional, contribuindo para uma operação mais eficiente e informada.

T-API: A Transport API (T-API), desenvolvida pela Open Networking Foundation (ONF), é uma interface padronizada usada para o gerenciamento de redes ópticas e de transporte, como SDN e redes ópticas de camada inferior. Ela permite o controle de conexões, topologias, serviços e o monitoramento de desempenho da rede, além de facilitar a integração entre diferentes camadas e fornecedores.

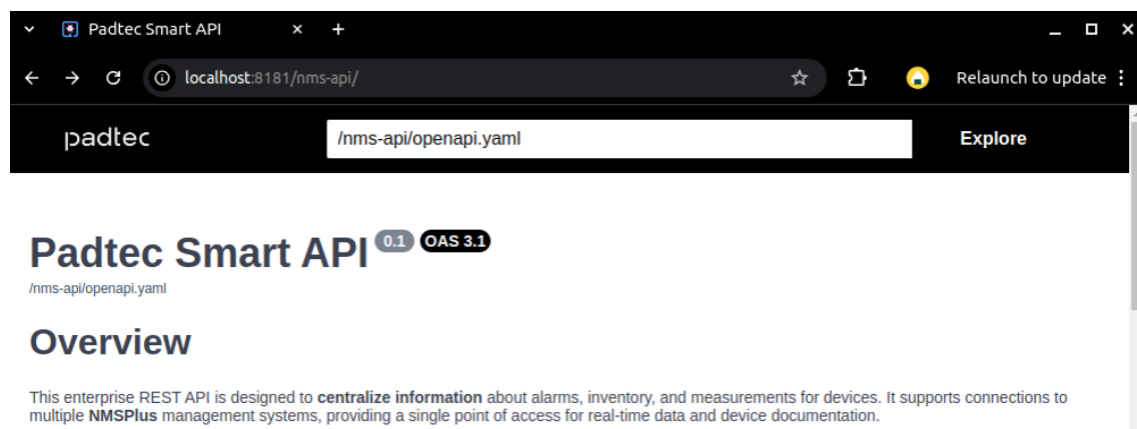
O Smart API tem o grande objetivo de centralizar a informação fornecida pelas gerências em um único ponto, e disponibilizar essa informação através de protocolos que podem ser usados em softwares de monitoramento ou de Inteligência.

2. Acessando Swagger

Swagger, atualmente conhecido como **OpenAPI**, é uma especificação amplamente adotada que oferece um conjunto de ferramentas para documentação, design, teste e consumo de APIs REST. Ele facilita a compreensão das funcionalidades de uma API, tanto por humanos quanto por sistemas, sem a necessidade de acesso ao código-fonte.

Em ambientes de telecomunicações, o Swagger é especialmente útil para interagir com APIs de gerenciamento de rede, como a **Smart API**. Por meio de uma interface gráfica acessível via navegador, ele permite explorar os endpoints, testar requisições e visualizar respostas de forma clara e intuitiva.

A Smart API pode ser acessada diretamente através do **Swagger**, utilizando o endereço `localhost:8181/nms-api/` ou, em casos de implantação em máquina virtual, pelo IP configurado, como `172.0.0.1:8181/nms-api/`. Essa abordagem proporciona uma experiência mais amigável e produtiva, otimizando o uso da API por equipes de operação, desenvolvimento e suporte técnico.



2.1. Gerando Token de Autorização

Acesse o campo "Admin Resource" e utilize o comando "POST" para criar registros de configuração, altere os parâmetros clicando no botão "Try it out":

No campo "clientId" adicione o nome de sua preferência para a geração do Token.

```
{  
  "username": "admin",  
  "password": "admin",  
  "clientId": "grafana"  
}
```

Para enviar o comando clique no botão "Execute".

The screenshot shows the "Admin Resource" section of an API client. The selected endpoint is `POST /nms-api/v1/admin/login` with the description "Login". The "Parameters" tab is active, showing "No parameters". The "Request body" tab is also active, showing a JSON object: `{ "username": "admin", "password": "admin", "clientId": "grafana" }`. The "Request body" is marked as "required" and the content type is set to "application/json". At the bottom, there are "Execute" and "Clear" buttons. The "Responses" section is visible at the bottom of the interface.


A resposta deve estar no campo "Server response" com o code 200, o token pode ser visto no campo "Response body".

200

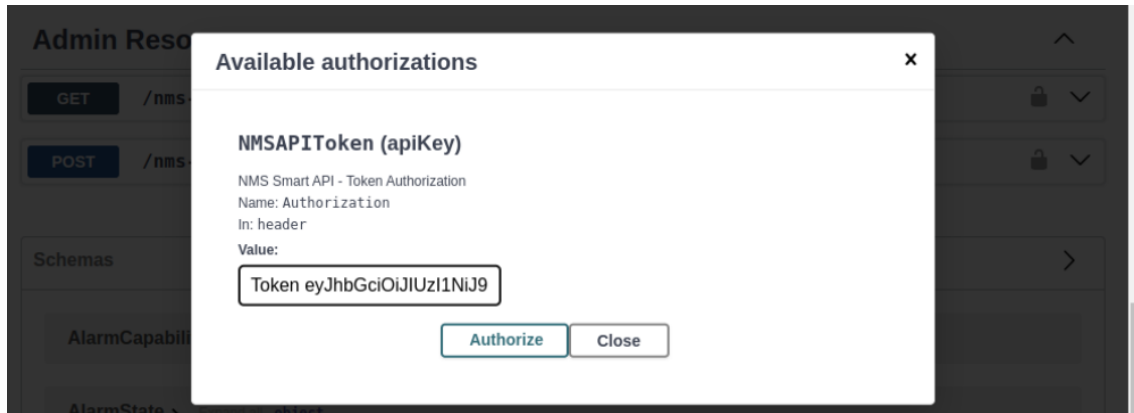
Response body

```
{  
  "token": "Token eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJncmFmYW5hIiwiaWF0IjoxNzQxODg0TUyYfQ.qXTajm-BA2Uu  
h5xiAVBUTLYbHFghIDpB6XGP0ryHZk"  
}
```

Com o token é possível enviar o comando **GET** para consultar informações da rede, como topologias, alarmes ou status de equipamentos.

Acesse o campo "Admin Resource" e utilize o comando "GET" para consultar informações dataproviders, clique no botão , para adicionar o token com a autorização.

Adicione o valor de token no campo "Value" e clique no botão "Authorize" para adicionar a autorização.



Após esse processo todas as opções de consulta estão autorizadas.

3. Organização da API

1. **Capability:** Os endpoints do tipo Capability disponibilizam informações de referência, abrangendo:

Alarmes Suportados: todos os alarmes que o equipamento pode gerar.

Medidas Suportadas: todos os tipos de medições que o equipamento pode fornecer.

Inventário de Portas e Conexões Internas: informações detalhadas sobre as portas disponíveis e conexões internas.

2. **Count:** Os endpoints do tipo Count exibem a quantidade atual de dados coletados, fornecendo uma visão imediata do volume de informações processadas.
3. **State:** Os endpoints do tipo State fornecem informações sobre o estado operacional atual dos equipamentos, incluindo:

Alarmes Ativos: lista dos alarmes atualmente ativos.

Medidas Recentes: dados de desempenho coletados nos últimos segundos.

Inventário do Equipamento: detalhes atualizados da configuração em uso.

Lista de consultas disponibilizados no Smart API:

Alarm Endpoints for managing and monitoring system alarms		^
GET	/nms-api/v1/alarm/capability	Retrieve a list alarms available for device
GET	/nms-api/v1/alarm/count	Retrieve count alarms from device
GET	/nms-api/v1/alarm/state	Retrieve paginated alarm
GET	/nms-api/v1/alarm/summary	Retrieve count alarms from device
Inventory Endpoints for managing and monitoring system inventory		^
GET	/nms-api/v1/inventory/capability	Retrieve a list inventory available for device
GET	/nms-api/v1/inventory/count	Retrieve count inventory from device
GET	/nms-api/v1/inventory/state	Retrieve paginated inventory
Measures Endpoints for managing and monitoring system measures		^
GET	/nms-api/v1/measures/capability	Retrieve a list measures available for device
GET	/nms-api/v1/measures/count	Retrieve count measures from device
GET	/nms-api/v1/measures/state	Retrieve paginated measure

A Smart API oferece suporte ao método **GET** para consulta de informações e ao método **POST**, utilizado exclusivamente para a geração de tokens de autenticação. Atualmente, os métodos **PUT** e **DELETE** não são suportados.

Implantação:

- O software foi desenvolvido de forma independente do NMS+, utilizando interfaces proprietárias para obtenção de dados em tempo real, sem impacto no desempenho do sistema de gerência.
- A distribuição é realizada via Docker, facilitando o empacotamento e a implantação.

Coleta de Dados:

- O software realiza dois tipos de coleta:
 - **Polling:** coleta periódica de medidas e inventário (intervalo padrão de 30 segundos).
 - **Streaming:** coleta contínua de alarmes (intervalo padrão de 1 segundo).
- O intervalo de coleta via polling pode ser ajustado nos arquivos de configuração.

Escalabilidade:

- O software foi projetado para escalar horizontalmente, possibilitando a coleta de dados de múltiplas gerências simultaneamente.

4. Alarm

O **Smart API** de Alarmes tem como principal objetivo o monitoramento em tempo real dos alarmes gerados pelas plataformas. Além disso, disponibiliza endpoints que informam quais tipos de alarmes cada modelo de equipamento da Padtec é capaz de gerar.

Alarm Endpoints for managing and monitoring system alarms			⌵
GET	/api/v1/alarm/capability	Retrieve a list alarms available for device	🔒 ⌵
GET	/api/v1/alarm/count	Retrieve count alarms from device	🔒 ⌵
GET	/api/v1/alarm/count/{card}	Retrieve count alarms from device	🔒 ⌵
GET	/api/v1/alarm/state	Retrieve paginated alarm	🔒 ⌵

4.1. Alarm - Capability

Visualizar quais alarmes uma placa pode gerar.

GET /api/v1/alarm/capability Retrieve a list alarms available for device 🔒 ⌵

Returns a paginated list of alarm available with filters.

Parameters Cancel

Name	Description
alarmGroup	Example: LOS, FAIL, LOF, TEMPERATURE, LASER OFF, POWER, LOS Sync, SSF ...
string (query)	<input type="text" value="alarmGroup"/>
alarmName	Example: LOS - PORT 1
string (query)	<input type="text" value="alarmName"/>
alarmResource	Example: PORT 1
string (query)	<input type="text" value="alarmResource"/>
cardModel	Example: TM2400G-NK
string (query)	<input type="text" value="cardModel"/>
page	Page index
integer (query)	<input type="text" value="0"/>
port	Example: IN PORT 1
string (query)	<input type="text" value="port"/>
size	Number of items per page
integer (query)	<input type="text" value="100"/>

Execute Clear

Curl

```
curl -X 'GET' \
  'http://localhost:8181/api/v1/alarm/capability?cardModel=TM2400G-NK&page=0&size=100' \
  -H 'accept: application/json' \
  -H 'Authorization: Token eyJhbGciOiJIUzI1NiIsInR5cCI6IkpzZW50b3R0Ij09IiwiaWF0IjoxNzMSMTk2MTU0fQ.YbKgcoNPrwC0SRkCGhho0Lz3vzL17ytSz319bCodjic'
```

Request URL

http://localhost:8181/api/v1/alarm/capability?cardModel=TM2400G-NK&page=0&size=100

Server response

Code Details

200

Response body

```
{
  "cardModel": "TM2400G-NK",
  "alarmMid": "TM2400G-NK-9906--1-PORT 5-PORT 5",
  "alarmName": "LOS",
  "alarmGroup": "LOS",
  "alarmResource": "PORT 5",
  "port": "IN Port 5"
},
{
  "cardModel": "TM2400G-NK",
  "alarmMid": "TM2400G-NK-9916--1-LINE 1-LINE 1",
  "alarmName": "Pin below the minimum threshold",
  "alarmGroup": "THRESHOLD",
  "alarmResource": "LINE 1",
  "port": "IN Line 1"
},
{
  "cardModel": "TM2400G-NK",
  "alarmMid": "TM2400G-NK-9906--1-PORT 9-PORT 9",
  "alarmName": "LOS",
  "alarmGroup": "LOS",
  "alarmResource": "PORT 9",
  "port": "IN Port 9"
},
{
  "cardModel": "TM2400G-NK",
  "alarmMid": "TM2400G-NK-9906--1-PORT 7-PORT 7",
  "alarmName": "LOS"
}
```

Response headers

```
content-type: application/json;charset=UTF-8
transfer-encoding: chunked
```

Download

4.2. Alarm - Count

Contador da quantidade de alarmes agrupado por severidade.

GET

/api/v1/alarm/count

Retrieve count alarms from device

⛔

⬆

Returns count alarms from device

Parameters

Cancel

No parameters

Execute

Clear

Responses

Curl

curl -X 'GET' \\\n'http://localhost:8181/api/v1/alarm/count' \\\n-H 'accept: application/json' \\\n-H 'Authorization: Token eyJhbGciOiJIUzI1NiJ9.eyJ3dWUiOiJncmFmYW5hIiwiaWF0IjoxNzMSMTk2MTU0FQ.YbKgcNPwC05RkCGHhoolz5vzL17ytS2319bCodjJc'

Request URL

http://localhost:8181/api/v1/alarm/count

Server response

Code

Details

200

Response body

{\n "TOTAL": 672,\n "CRITICAL": 550,\n "MAJOR": 36,\n "MINOR": 50,\n "WARNING": 20,\n "UNKNOWN": 0,\n "CLEARED": 0\n}

⛏

Download

4.3. Alarm - State

Visualizar os alarmes ativos dos equipamentos.

GET

/api/v1/alarm/state

Retrieve paginated alarm

⛔

⬆

Returns a paginated list of alarm states with filters.

Parameters

Cancel

Name	Description
alarmGroup string (query)	Example: LOS, FAIL, LOF, TEMPERATURE, LASER OFF, POWER, LOS Sync, SSF ... alarmGroup
alarmIsActive boolean (query)	Example: true, false --
alarmName string (query)	Example: LOS - PORT 1 alarmName
alarmOld string (query)	Example: 62879 alarmOld
alarmResource string (query)	Example: PORT 1 alarmResource
alarmSeverity string (query)	Example: CRITICAL, MAJOR, MINOR, WARNING, UNKNOWN, CLEARED alarmSeverity
cardFamily string (query)	Example: transponder, amplifier, roadm_wss, roadm_plic, others cardFamily
cardModel string (query)	Example: TM2400G-NK cardModel
cardPart string (query)	Example: 2795 cardPart

Execute

Clear

```
curl -X 'GET' \
'http://localhost:8181/api/v1/alarm/state?cardFamily=transponder&page=0&size=10' \
-H 'accept: application/json' \
-H 'Authorization: Token eyJhbGciOiJIUzI1NiIsInR5cGU6IjY9bnVlcnF1eWkiLCJpdjoxNDZSMThkZWU0fQ.YmkgOGNrcCvO5BRKCGHooL5zvL7t5Z3l9bcOdjjc'
```

```
http://localhost:8181/api/v1/alarm/state?cardFamily=transponder&page=0&size=10
```

```
content-length: 5079
content-type: application/json;charset=UTF-8
```

5. Inventory

O **Smart API** de inventário possui como premissa mostrar o status dos equipamentos, e algumas informações internas das conexões.

Inventory Endpoints for managing and monitoring system inventory			
GET	/api/v1/inventory/capability	Retrieve a list inventory available for device	🔒
GET	/api/v1/inventory/count	Retrieve count inventory from device	🔒
GET	/api/v1/inventory/count/{card}	Retrieve count inventory from device	🔒
GET	/api/v1/inventory/state	Retrieve paginated inventory	🔒

5.1. Inventory - Capability

Visualizar informações gerais sobre os modelos de equipamentos, portas e conexões internas.

Inventory

Endpoints for managing and monitoring system inventory

GET

/api/v1/inventory/capability

Retrieve a list inventory available for device

🔒

Returns a paginated list of inventory available with filters.

Parameters

Cancel

Name	Description
cardModel	Example: TM2400G-NK
string (query)	EOA2-HA-E21211-GW
cardPart	Example: 2795
string (query)	cardPart
page	Page index
integer (query)	0
size	Number of items per page
integer (query)	10

Execute

Clear

Responses

Curl

curl -X 'GET' \n'http://localhost:8181/api/v1/inventory/capability?cardModel=EOA2-HA-E21211-GW&page=0&size=10' \n-H 'accept: application/json' \n-H 'Authorization: Token eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJncmFmYXVhIiwiaWF0IjoxNzMSMTk2MTU0fQ.YbKgcoNPwC85RkCGHhoolz5vzL17yt5z319bCodjic'

Request URL

http://localhost:8181/api/v1/inventory/capability?cardModel=EOA2-HA-E21211-GW&page=0&size=10

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ { "cardModel": "EOA2-HA-E21211-GW", "cardPart": 2462, "ports": ["IN Data", "IN Line", "OUT Data", "OUT Line", "SC in", "SC out"], "crossconnection": ["IN Data->OUT Line", "IN Line->OUT Data", "SC in->SC out"], "descriptor": "adapters/models/model-amplifier-eoa2-ha-e21211-gw-amplifier45u--2462.md", "reload": "2025-02-13T16:56:59.444+00:00" } }</pre></div><div>Download</div></div> <div>Response headers</div> <div><pre>content-length: 324 content-type: application/json;charset=UTF-8</pre></div> <div>Request duration</div> <div>122 ms</div>

5.2. Inventory - Count

Visualizar quantidade total de equipamentos monitorados.

GET

/api/v1/inventory/count

Retrieve count inventory from device

⛔

⬆

Returns count inventory from device

Parameters

Cancel

No parameters

Execute

Clear

Responses

Curl

curl -X 'GET' \\\n'http://localhost:8181/api/v1/inventory/count' \\\n-H 'accept: application/json' \\\n-H 'Authorization: Token eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJncmFmYXV5IiwiaWF0IjoxNzMSMTk2MTU0fQ.YbKgcoNPwC85RkCGHhoolz5vzL17ytS2319bCodjJc'

Request URL

http://localhost:8181/api/v1/inventory/count

Server response

Code

Details

200

Response body

123

Download

Response headers

content-length: 3\\ncontent-type: application/json;charset=UTF-8

5.3. Inventory - State

Visualizar informações gerais dos equipamentos e localização.

GET

/api/v1/inventory/state

Retrieve paginated inventory

⛔

⬆

Returns a paginated list of inventory with filters.

Parameters

Cancel

Name	Description
cardFamily string (query)	Example: transponder, amplifier, roadm_wss, roadm_plc, others transponder
cardFirmwareVersion string (query)	Example: 1.0.0 cardFirmwareVersion
cardsActive boolean (query)	Example: true, false --
cardModel string (query)	Example: TM2400G-NK cardModel
cardName string (query)	Example: TM2400G-NK#11 cardName
cardPart string (query)	Example: 2795 cardPart
cardSerial string (query)	Example: 1 cardSerial
cardUid string (query)	Example: TM2400G-NK#11 cardUid
locationCollector string (query)	Example: 127.0.0.1 locationCollector
locationDataprovider string (query)	Example: 127.0.0.1 locationDataprovider

6. Measures

O **Smart API** de medidas possui como premissa coletar as medidas dos equipamentos com intervalo de *30 segundos, e normalizar os dados.

Measures Endpoints for managing and monitoring system measures

GET	/api/v1/measures/capability	Retrieve a list measures available for device	🔒	⌵
GET	/api/v1/measures/count	Retrieve count measures from device	🔒	⌵
GET	/api/v1/measures/count/{card}	Retrieve count measures from device	🔒	⌵
GET	/api/v1/measures/state	Retrieve paginated measure	🔒	⌵

6.1. Measures - Capability

Visualizar quais medidas uma placa pode gerar.

GET

/api/v1/measures/capability

Retrieve a list measures available for device

🔒

⌵

Returns a paginated list of measures from device

Parameters

Cancel

Name	Description
cardModel string (query)	Example: TM2400G-NK EOA2-HA-E21211-GW
measureGroup string (query)	Example: 2795 measureGroup
measureName string (query)	Example: 2795 measureName
measureResource string (query)	Example: 2795 measureResource
page integer (query)	Page index 0
port string (query)	Example: IN PORT 1 port
size integer (query)	Number of items per page 10

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8181/api/v1/measures/capability?cardModel=EOA2-HA-E21211-GW&page=0&size=10' \
  -H 'accept: application/json' \
  -H 'Authorization: Token eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXZWQ6OTk2MTU0fQ.YbKgcoNPrwC0SRkCGhoolz5vzL17ytS2319bCodjJc'
```

Request URL

http://localhost:8181/api/v1/measures/capability?cardModel=EOA2-HA-E21211-GW&page=0&size=10

Server response

Code

Details

200

Response body

```
{
  "measureGroup": "ORIN",
  "port": null,
  "updatedAt": "2025-02-13 16:57:02"
},
{
  "id": "EOA2-HA-E21211-GW-7255-0",
  "cardModel": "EOA2-HA-E21211-GW",
  "measureName": "IN Line IN Power",
  "measureResource": "IN Line Power",
  "measureGroup": "POWER",
  "port": null,
  "updatedAt": "2025-02-13 16:57:02"
},
{
  "id": "EOA2-HA-E21211-GW-7256-0",
  "cardModel": "EOA2-HA-E21211-GW",
  "measureName": "Target Gain Stage 2",
  "measureResource": "Target Gain Stage 2",
  "measureGroup": "GAIN",
  "port": null,
  "updatedAt": "2025-02-13 16:57:02"
},
{
  "id": "EOA2-HA-E21211-GW-7256-0",
  "cardModel": "EOA2-HA-E21211-GW",
  "measureName": "OUT Line Power",
  "measureResource": "OUT Line Power",
  "measureGroup": "POWER",
  "port": null,
  "updatedAt": "2025-02-13 16:57:02"
}
```

Response headers

```
content-length: 2028
content-type: application/json;charset=UTF-8
```

Request duration

67 ms

6.2. Measures - Count

Visualizar a quantidade de medidas coletadas.

GET /api/v1/measures/count Retrieve count measures from device

Returns count measures from device

Parameters

No parameters

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8181/api/v1/measures/count' \
  -H 'accept: application/json' \
  -H 'Authorization: Token eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXZWQ6OTk2MTU0fQ.YbKgcoNPrwC0SRkCGhoolz5vzL17ytS2319bCodjJc'
```

Request URL

http://localhost:8181/api/v1/measures/count

Server response

Code

Details

200

Response body

2285

Response headers

```
content-length: 4
content-type: application/json;charset=UTF-8
```

6.3. Measures - State

Visualizar quais medidas foram coletadas dos equipamentos.

GET

/api/v1/measures/state

Retrieve paginated measure

🔒 ⬆

Returns a paginated list of measures

Parameters

Cancel

Name	Description
cardFamily string (query)	Example: transponder, amplifier, roadm_wss, roadm_plc, others <input type="text" value="cardFamily"/>
cardModel string (query)	Example: TM2400G-NK <input type="text" value="cardModel"/>
cardPart string (query)	Example: 2795 <input type="text" value="cardPart"/>
cardSerial string (query)	Example: 1 <input type="text" value="cardSerial"/>
cardUid string (query)	Example: TM2400G-NK#11 <input type="text" value="cardUid"/>
locationCollector string (query)	Example: 127.0.0.1 <input type="text" value="locationCollector"/>
locationDataprovider string (query)	Example: 127.0.0.1 <input type="text" value="locationDataprovider"/>
locationSite string (query)	Example: NE 1 <input type="text" value="locationSite"/>
measureGroup string (query)	Example: 2795 <input type="text" value="measureGroup"/>
measureName string (query)	Example: 2795 <input type="text" value="measureName"/>
measureResource string (query)	Example: 2795 <input type="text" value="measureResource"/>
measureUid string (query)	Example: 2795 <input type="text" value="measureUid"/>
measureValue string (query)	Example: 2795 <input type="text" value="measureValue"/>
page integer (query)	Page index <input type="text" value="0"/>
port string (query)	Example: IN PORT 1 <input type="text" value="port"/>
size integer (query)	Number of items per page <input type="text" value="100"/>

Execute

Clear

Curl

```
curl -X 'GET' \
  'http://localhost:8181/api/v1/measures/state?page=0&size=100' \
  -H 'accept: application/json' \
  -H 'Authorization: Token eyJhbGciOiJIUzI1NiJ9.eyJ2dWUiOiJncmFwYXVhbnRlIiwiaWF0IjoxNzMSMTk2MTU0fQ.YbKgc0NPrwC05RkCGHhoolz5vzL17ytS2319bCodjJc'
```

Request URL

http://localhost:8181/api/v1/measures/state?page=0&size=100

Server response

Code

Details

200

Response body

```
{
  "locationDataProvider": "172.22.202.8",
  "locationCollector": null,
  "locationSite": "ME 1",
  "cardId": "EOA1-HA-E21001-GV#1002",
  "cardModel": "EOA1-HA-E21001-GV",
  "cardFamily": "amplifier",
  "cardPart": 2466,
  "cardSerial": 1002,
  "measureUid": "EOA1-HA-E21001-GV#1002-7265-0",
  "measureName": "Target Gain",
  "measureResource": null,
  "measureGroup": "GAIN",
  "measureValue": "0.057",
  "part": null,
  "update": "2025-02-13 17:48:38"
},
{
  "locationDataProvider": "172.22.202.8",
  "locationCollector": null,
  "locationSite": "ME 1",
  "cardId": "EOA1-HA-E21001-GV#1002",
  "cardModel": "EOA1-HA-E21001-GV",
  "cardFamily": "amplifier",
  "cardPart": 2466,
  "cardSerial": 1002,
  "measureUid": "EOA1-HA-E21001-GV#1002-7261-0",
  "measureName": "Power Temperature"
}

```

Response headers

```
content-type: application/json;charset=UTF-8
transfer-encoding: chunked

```

Request duration

106 ms

20

OM.SMARTAPI.2025.06.POR.V2



Padtec S/A

Rua Doutor Ricardo Benetton Martins, 1.000
Parque II do Polo de Alta Tecnologia
CEP: 13.086-510 • Campinas • SP • Brasil

padtec.com.br • **sac@padtec.com.br**