JOAO PESSOA TRIBUNAL DE JUSTICA DO ESTADO DA PARAIBA
CNPJ: 09.283.185/0001-63
Ref.: PREGÃO ELETRÔNICO N º 90033/2024

Em atenção à solicitação de diligência referente à proposta apresentada pela INORPEL COMÉRCIO E SERVIÇOS LTDA no PREGÃO ELETRÔNICO Nº 90033/2024 (PROCESSO ADMINISTRATIVO Nº 004761-06.2024.8.15, Edital nº 033/2024), esclarecemos os seguintes pontos:

1. Integração com o Sistema VISIT:

   O hardware fornecido pela nossa solução possui API aberta, possibilitando a comunicação com o sistema VISIT do Tribunal de Justiça da Paraíba seguindo o modelo de arquitetura CGI. Essa abordagem permite a delegação de autorização ou negação de acesso, utilizando as imagens da base de dados do sistema VISIT para validação e reconhecimento de acesso. Ressaltamos que nossa solução é robusta e altamente eficiente, garantindo segurança, confiabilidade e flexibilidade na integração com o sistema do Tribunal.

2. Documentação de Interoperabilidade:

   Para garantir total transparência e facilidade na integração, disponibilizamos abaixo o link de acesso à documentação da API da linha de controle de acesso da Intelbras, fabricante do hardware proposto:

Documentação da API - Intelbras

3. Declaração do Fabricante:

   Anexamos a este documento a declaração emitida pelo fabricante Intelbras, atestando a compatibilidade da solução com o serviço de integração via modelo de arquitetura CGI e assegurando a funcionalidade requerida pelo Tribunal de Justiça da Paraíba.

Reiteramos nosso compromisso em fornecer uma solução que atenda integralmente às exigências do edital e ficamos à disposição para quaisquer esclarecimentos adicionais.

Atenciosamente,

Cabedelo-PB, 19 de março de 2025.

INORPEL COMERCIO E SERVIÇOS LTDA

Rodrigo Agra de Brito
RG:1831124 SSP PB
CPF:007.388.144-19
Representante Legal
Email: rodrigo.brito@inorpelcybersecurity.com.br

JOAO PESSOA TRIBUNAL DE JUSTICA DO ESTADO DA PARAIBA
CNPJ: 09.283.185/0001-63

Ref.: PREGÃO ELETRÔNICO N º 90033/2024

DECLARAÇÃO

A INTELBRAS S/A INDÚSTRIA DE TELECOMUNICAÇÃO ELETRÔNICA BRASILEIRA, privado, inscrita no CNPJ sob o nº 82.901.000/0001-27, sediada na Rodovia BR 101, Km 210, Área Industrial, São José/ SC, CEP 88.104-800, DECLARA, para os devidos fins, e a quem possa interessar, por intermédio de seu procurador signatário, que os produtos ora ofertados pela empresa **INORPEL COMERCIO E SERVICOS LTDA,** inscrita no CNPJ sob o nº 10.920.030/0001-70, para o referido processo, dispõe da seguinte especificação técnica:

PRODUTO
CONTROLADOR ACESSO RECONHECIMENTO FACIAL SS 5532 MF W

- A Intelbras disponibiliza a documentação API do produto SS 5532 MF W seguindo o modelo de arquitetura CGI.

São José/SC, 14 de fevereiro de 2025.

PATRICIA SCHERER

Assinado de forma digital por PATRICIA SCHERER
Dados: 2025.02.14 11:51:48 -03'00'

_____

INTELBRAS S/A
CNPJ: 82.901.000/0001-27

,

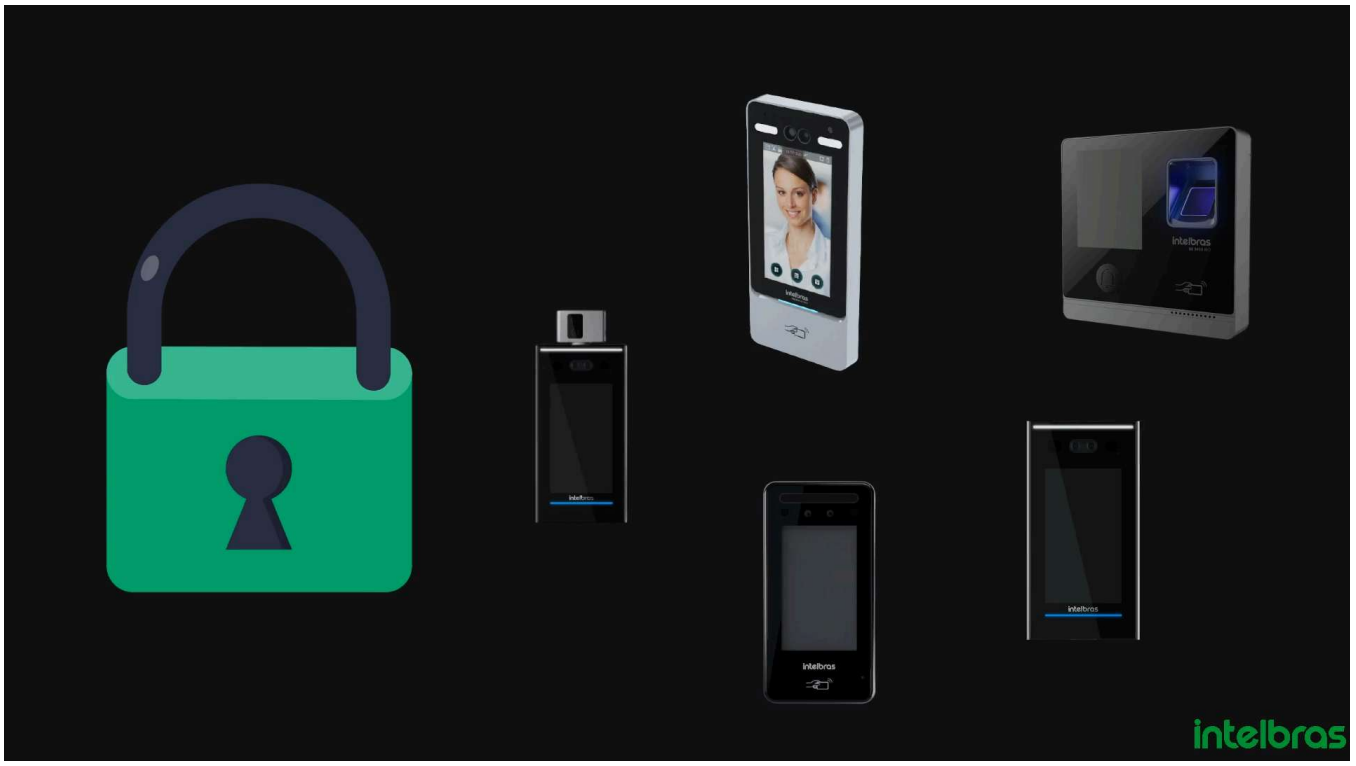
ID: 2025.072

Introdução

API - Dispositivos de Controle de Acesso Corporativo

Bem-vindo à documentação da API da linha de acesso da Intelbras. Esta API tem como objetivo fornecer informações e orientações sobre como integrar com os dispositivos de acesso da Intelbras.

Nossos equipamentos de acesso possuem uma interface de comunicação baseada em TCP/IP (Ethernet), tornando a integração simples e independente do sistema operacional e da linguagem de programação utilizados. Neste documento, você encontrará todos os detalhes necessários para utilizar a API e integrar facilmente com nossos dispositivos de acesso.



Dispositivos da Linha Bio-T que suportam a API:

Dispositivos	Suporta API?
SS 5520	✅
SS 5530 MF FACE	✅
SS 5530 MF FACE LITE	✅

Dispositivos	Suporta API?
SS 3530 MF FACE W	✅
SS 3430 BIO	✅
SS 3430 MF BIO	✅
SS 7520 FACE T	✅
SS 7530 FACE	✅
SS 3530 MF FACE	✅
SS 3540 MF FACE EX	✅
SS 1540 MF W	✅
SS 1530 MF W	✅
SS 3540 MF FACE BIO	✅
SS 3532 MF W	✅
SS 3542 MF W	✅
SS 5531 MF W	✅
SS 5541 MF W	✅
SS 5532 MF W	✅
SS 5542 MF W	✅
SS 3420 BIO	❌
SS 3420 MF BIO	❌
CT 3000 2PB	API
CT 3000 4PB	API

Padrão de Sintaxe Utilizada

A sintaxe da API deve seguir o padrão de URI (RFC 3986 Uniform Resource Identifiers (URI) Generic Syntax).

```
<protocolo>://<servidor><abs_path>[?query]
```

protocolo: Valor padrão suportado pela API é o "http". Existem exceções apenas em algumas chamadas RTSP que usando o valor "RTSP".

servidor: O servidor é definido por "hostname:porta". O nome do host pode ser o endereço IP ou o nome de domínio de um dispositivo IP. A porta é o número da porta de conexão TCP configurada no dispositivo. Se a porta não for configurada, a porta **80** utilizada.

abs_path: O Request-URI para os recursos é abs_path. O abs_path nesta especificação é na maioria das vezes "/cgi-bin/*.cgi".

query: O campo query é uma string de informações a ser interpretada pelo dispositivo. Consiste em parâmetros relacionados ao recurso sendo requisitado. Deve ser informado seguindo a sintaxe nome=valor. Por exemplo: channel=1 (http://192.168.1.108/cgi-bin/snapshot.cgi?channel=1)

Formato de respostas

O servidor usa os códigos de status HTTP padrão. Com o seguinte código HTTP e significados:

HTTP Code	HTTP text	Descrição
200	OK	A solicitação foi bem-sucedida. O solicitado recurso será retornado no texto HTTP.
400	Bad Request	A solicitação tinha sintaxe incorreta ou era inerentemente impossível de ser satisfeito.
401	Unauthorized	A solicitação requer autenticação do usuário ou o a autorização foi recusada.
403	Forbidden	O usuário não tem o direito de acessar o serviço.
404	Not Found	O servidor não encontrou nada que corresponda ao solicitação.
500	Internal Server Error	O servidor encontrou uma condição inesperada que o impediu de atender à solicitação.
501	Not Implemented	O servidor não implementou o serviço.

Se o código HTTP for **200**, significa que a API foi executada com sucesso e os dados de resposta no corpo HTTP (multipart) pode ser uma multiline **key=value** de valor ou um objeto JSON ou apenas uma linha com uma palavra **"*OK*"**

Alguns dispositivos em suas versões mais recentes, podem retornar no corpo da resposta a requisição alguns códigos de erro verifique a página Códigos de Erros de Requisição

Autenticação

Os dispositivos possuem suporte digest authentication, consulte **RFC 2617** (RFC 2617 HTTP Authentication)

A autenticação Digest é um esquema de autenticação baseado em desafio-resposta que permite que um cliente se autentique com um servidor web protegido por senha.

Quando um cliente faz uma solicitação HTTP para um recurso protegido por autenticação Digest, o servidor web responde com um código de status HTTP 401, indicando que o acesso não é permitido sem autenticação. Junto com o código de status 401, o servidor também envia um cabeçalho "WWW-Authenticate" contendo informações sobre a proteção utilizada para a senha, bem como um desafio aleatório.

O cliente responde ao desafio enviando um cabeçalho "Authorization" contendo as credenciais de autenticação criptografadas, juntamente com informações sobre o desafio e outras informações necessárias. O servidor web verifica se as credenciais estão corretas e, se estiverem, retorna um código de status HTTP 200, indicando que o acesso ao recurso foi permitido.

O motivo pelo qual o servidor web envia um código de status HTTP 401 antes de enviar o código de status HTTP 200 é que o cliente precisa ser informado de que a autenticação é necessária antes de enviar as credenciais de autenticação criptografadas. O código de status HTTP 401 é usado para indicar que o acesso ao recurso é negado sem autenticação.

Em resumo, a sequência de códigos de status HTTP 401 seguido por um código de status HTTP 200 é um comportamento padrão na autenticação Digest para permitir que o cliente se autentique com o servidor web protegido por senha

> ⚠️ Se a solicitação HTTP enviada pelo cliente não fornecer informações de cabeçalho de "authorization" válidas, o dispositivo retorna o código de status **HTTP 401**.

Exemplo de Autenticação:

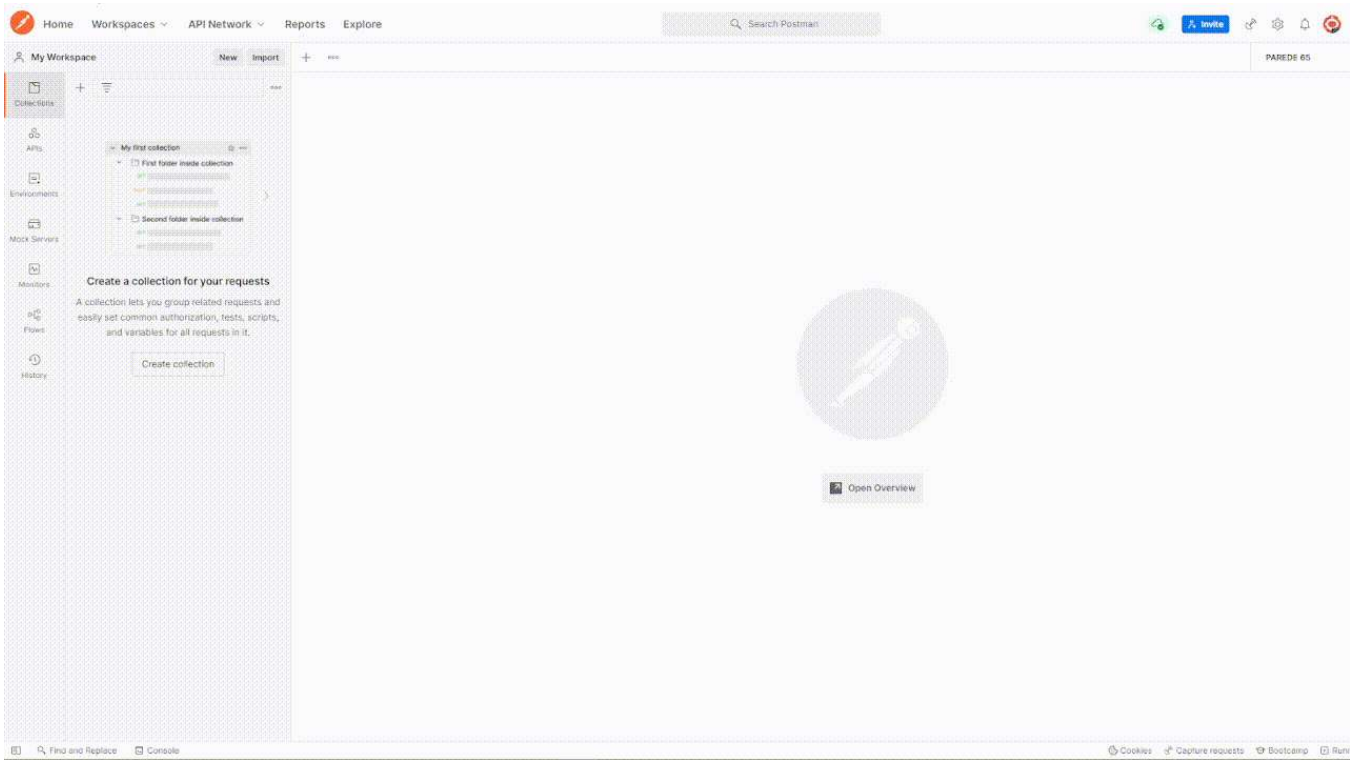
python Node

```python
import requests

device_ip = '192.168.1.201'
username = 'admin'
password = 'admin12345'

url = "http://{}/cgi-bin/global.cgi?action=getCurrentTime".format(
                                    str(ip)
                                )
digest_auth = requests.auth.HTTPDigestAuth(username, passwd)
rval = requests.get(url, auth=digest_auth, timeout=20, verify=False)
```

Biblioteca Postman

Recomendamos a execução das chamadas de teste da API utilizando um software client rest como Insomnia ou Postman antes de encapsular em seu software.

Para facilitar a integração foi realizado o mapeamento das principais chamadas na ferramenta Postman, possibilitando importar as chamadas e realizar os testes, disponível através do link: **DOWNLOAD POSTMAN COLLECTIONS**



Last updated on January 13, 2025