



D-Guard Center

Documentação da interface HTTP/REST
Abertura do navegador na tela Ao vivo – Versão 1.00

[Histórico de alterações]

Revisão	Data	Autor	Observações
1.0	26.11.2018	John Jaraceski	Versão original

[Conteúdo]

Sobre este documento	4
Preparação	4
Formato das requisições	4
Formato das respostas	4
Conceitos preliminares	5
Contact-Id	5
Servidor, câmara e dispositivo	5
Mosaicos e layouts de câmeras	6
Abertura do navegador na tela “Ao vivo”	6
API REST para obter token de acesso (login)	7
API REST para obter layouts por número de receptora	8
API REST para obter layouts por número de conta	9
API REST para obter layouts por código de partição	10
Modelo de URL para abertura da tela “Ao vivo”	11
Notas sobre abertura da tela “Ao vivo”	12

1 Sobre este documento

Este documento fornece informações sobre a interface HTTP/REST dos softwares D-Guard Center (versão 6 ou superior) para a abertura do navegador (padrão) na tela "Ao vivo", incluindo informações sobre todas as requisições HTTP/GET necessárias para obtenção de token de acesso e de layout associado à determinado contact-id.

2 Preparação

Ative o serviço API REST do D-Guard Center, na tela "Configuração de Transmissão" para ativar a interface HTTP/REST. Um teste pode ser feito entrando com a seguinte URL em um navegador:

<http://servidor:porta/api/version>

Onde:

- servidor = domínio, nome na rede, ou IP do servidor D-Guard
- porta = número da porta em que o serviço API REST está ativo.

2.1 Formato das requisições

As requisições devem ser enviadas no padrão HTTP, usando os verbos padrões "GET", "POST", "PUT", entre outros. As linguagens de programação normalmente fornecem implementações padrão desse protocolo, como *HttpConnection*, *HttpClient* ou *XMLHttpRequest*.

A autenticação de usuário segue o formato "Authorization: Bearer + Token" do HTTP. Caso necessário, utilize a documentação do protocolo HTTP (rfc 2617) para maiores informações sobre esse método de autenticação.

2.2 Formato das respostas

As respostas para as requisições HTTP acompanham certos cabeçalhos indicando informações úteis sobre as mesmas. Seguem os principais:

- Content-Type: Indica o formato do conteúdo retornado. Exemplo: *text/plain* (para texto), *image/jpeg* (para imagem), etc.
- Content-Length: Indica o tamanho em bytes do conteúdo retornado.

3 Conceitos preliminares

Para um pleno entendimento das requisições necessárias para a abertura do navegador na tela “Ao vivo” é preciso entender alguns conceitos preliminares sobre as principais entidades envolvidas neste processo: contact-id; servidor, câmera e dispositivo; e mosaicos e layouts de câmeras.

3.1 Contact-Id

Os softwares D-Guard Center permitem a integração com centrais de monitoramento via uso de um protocolo chamado Contact-Id. Este protocolo, desenvolvido pela empresa Ademco, é utilizado em todo o mundo para efetuar a comunicação entre equipamentos de segurança.

O protocolo Contact-Id é composto por quatro grupos de informações, relacionados a seguir:

- Número de conta do cliente
- Qualificação do evento + Evento
- Partição
- Argumento

Dentro do escopo deste documento, somente será feito uso dos grupos de informações denominados “número de conta do cliente” e “partição”. Esta informação é utilizada nos softwares D-Guard para identificar uma central de monitoramento compatível com protocolo Contact-Id.

Além disso, o “número de conta do cliente” pode ser associado a um ou mais servidores em uma instância D-Guard Center. Dessa forma, é possível realizar uma associação indireta entre “número de conta do cliente” e layouts de câmeras cadastrados em uma instância D-Guard Center.

Também é possível o cadastro de várias receptoras de Contact-Id em uma instância D-Guard Center, permitindo assim que o “número de conta do cliente” se repita entre diferentes receptoras de Contact-Id.

3.2 Servidor, câmera e dispositivo

Os softwares D-Guard Center permitem o uso de diferentes tipos de dispositivos de vigilância eletrônica: NVRs, DVRs e câmeras. Todos estes diferentes tipos de dispositivos são cadastradas como uma única entidade chamada “servidores” na tela “Configuração de Servidores”.

A principal característica que distingue servidores associados a dispositivos dos tipos NVR/DVR e câmera é a quantidade de canais de vídeo associadas ao servidor.

Um servidor do tipo câmera IP contém apenas um canal de vídeo porque está conectado diretamente a uma única câmera. Enquanto que um servidor do tipo NVR/DVR pode conter múltiplos canais de vídeo, sendo que o limite máximo de canais é determinado pelo número e tipo de licença adquiridos pelo cliente para o software D-Guard.

Além disso, um servidor também pode estar vinculado a uma outra instância D-Guard Center. Neste caso, este servidor é visto como um agrupador de servidores e cada dispositivo configurado na instância remota é visto como um novo servidor na instância local do D-Guard Center.

3.3 Mosaicos e layouts de câmeras

Os softwares D-Guard Center permitem o uso de mosaicos pré-definidos e a montagem de mosaicos personalizados com até 100 (cem) câmeras, diagramados em células com dimensões de 1x1 até 10x10 e contendo mesclagem de células para possibilitar o destaque de determinadas câmeras.

Os layouts de câmeras, em contrapartida, podem ser vistos como instâncias de mosaicos onde cada célula ou mesclagem de células é configurada para realizar a exibição de uma determinada câmera. Além disso, é possível configurar a exibição de uma mesma câmera em diferentes layouts e, até mesmo, em um mesmo layout.

4 Abertura do navegador na tela “Ao vivo”

O processo de abertura do navegador na tela “Ao vivo” inicia pela obtenção do token de acesso e termina pela montagem da URL de abertura do navegador na tela “Ao vivo”. Além disso, também é necessária a obtenção do layout desejado para exibição na tela “Ao vivo”.

A etapa de obtenção do token de acesso consiste em efetuar uma requisição à API “/login” para autenticar-se em uma instância do software D-Guard Center e obter um token de acesso válido.

A obtenção do layout desejado para exibição na tela “Ao vivo” pode ser realizada efetuando-se requisições à diferentes API, entre elas: “/contact-id/layouts” e “/layouts”. A primeira API, “/contact-id/layouts” permite a obtenção dos layouts associados a um

determinado “número de conta do cliente”, enquanto a segunda API permite a obtenção de todos os layouts existentes na instância de software D-Guard.

Entretanto, é importante ressaltar que dependendo do cenário de integração desejado, é possível a abertura da tela “Ao vivo” no primeiro layout disponível. Nesse caso, a etapa de obtenção do layout desejado se torna opcional e apenas a etapa de obtenção do token é obrigatória.

A última etapa se refere à montagem da URL de abertura do navegador na tela “Ao vivo” e consiste somente em preencher parâmetros específicos em um modelo de URL predefinido.

É importante ressaltar que a abertura propriamente dita do navegador depende da linguagem de programação utilizada para integração com a instância de software D-Guard Center.

4.1 API REST para obter token de acesso (login)

A API “/login” permite a autenticação na instância do software D-Guard e retorna o *token* de acesso que deverá ser utilizado em todas as demais requisições HTTP à API REST, exceto pela API “/version”. Bem como, para montagem da URL de abertura do navegador na tela “Ao vivo”.

[POST] <http://servidor:porta/api/login>

```
{
  "username": [string],
  "password": [string]
}
```

Onde:

- O atributo “username” contém o nome do usuário previamente cadastrado.
- O atributo “password” contém a senha do usuário em texto claro.

Exemplo de uso com aplicativo cURL

Requisição:

```
curl -X POST "http://127.0.0.1:8080/api/login" -H "Accept: application/json" -H "Content-Type: application/json" -d '{"username": "admin", "password": "123456"}'
```

Retorno:

```
{
  "login": {
    "username": "admin",
    "userGuid": "{4FC0C44C-2789-4902-B322-A2927B14E6A8}",
    "userGroupGuid": "{491693D5-3F25-493E-9234-DAE05A7D8458}",
  }
}
```

```

        "userToken": "eyJ1...XfnQ",
        "serverDate": "2018-08-07",
        "serverTime": "14:00:00"
    }
}

```

4.2 API REST para obter layouts por número de receptora

A API `"/contact-id/layouts/{receiver-code}"` retorna a lista de todos os layouts associados a uma determinada receptora de Contact-Id em uma instância do software D-Guard Center.

[GET] <http://servidor:porta/api/contact-id/layouts/{receiver-code}>

Exemplo de uso com aplicativo cURL

Requisição:

```

curl -X GET "http://127.0.0.1:8080/api/contact-id/layouts/10001"
-H "Accept: application/json" -H "Authorization: Bearer
eyJ1...XfnQ"

```

Retorno:

```

{
  "layouts": [
    {
      "contactIdReceiverCode": "10001",
      "layoutGuid": "{91E696F8-98EB-4312-B75D-BD1755B85285}",
      "layoutName": "PRINCIPAL"
    },
    {
      "contactIdReceiverCode": "10001",
      "layoutGuid": "{A646D905-993C-4B54-A384-EFED8D595755}",
      "layoutName": "SECUNDÁRIO"
    }
  ]
}

```

Onde:

- O parâmetro "receiver-code" da URL de requisição deve conter apenas números e corresponde ao código da receptora (empresa).
- O atributo "contactIdReceiverCode" contém o código da receptora e trata-se exatamente do mesmo código de receptora utilizado na URL de requisição.
- O atributo "layoutGuid" contém a identificação do layout e trata-se do atributo mais relevante para as demais etapas do processo de abertura do navegador na tela "Ao vivo".
- O atributo "layoutName" contém o nome do layout e pode ser bastante útil para casos onde existam mais de um layout associado ao mesmo número de conta.

A quantidade de layouts associado a um mesmo código de receptora é dependente dos cadastrados efetuados na instância do software D-Guard Center.

Dessa forma, o retorno desta API precisa ser analisado para identificar exatamente qual layout se deseja utilizar para abertura do navegador na tela "Ao vivo".

4.3 API REST para obter layouts por número de conta

A API `/contact-id/layouts/{receiver-code}/{account-code}` retorna a lista de todos os layouts associados a uma determinada conta de uma receptora de Contact-Id em uma instância do software D-Guard Center.

[GET] <http://servidor:porta/api/contact-id/layouts/{receiver-code}/{account-code}>

Exemplo de uso com aplicativo cURL

Requisição:

```
curl -X GET
"http://127.0.0.1:8080/api/contact-id/layouts/10001/00000FF" -H
"Accept: application/json" -H "Authorization: Bearer eyJl...XfnQ"
```

Retorno:

```
{
  "layouts": [
    {
      "contactIdReceiverCode": "10001",
      "contactIdAccountCode": "00000FF",
      "layoutGuid": "{91E696F8-98EB-4312-B75D-BD1755B85285}",
      "layoutName": "PRINCIPAL"
    },
    {
      "contactIdReceiverCode": "10001",
      "contactIdAccountCode": "00000FF",
      "layoutGuid": "{A646D905-993C-4B54-A384-EFED8D595755}",
      "layoutName": "SECUNDÁRIO"
    }
  ]
}
```

Onde:

- O parâmetro "receiver-code" da URL de requisição deve conter apenas números de no máximo seis dígitos e corresponde ao código da receptora (empresa).
- O parâmetro "account-code" da URL de requisição deve conter numeração em formato hexadecimal de no máximo sete caracteres e corresponde ao número da conta.

- O atributo "contactIdReceiverCode" contém o código da receptora e trata-se exatamente do mesmo código de receptora utilizado na URL de requisição.
- O atributo "contactIdAccountCode" contém o número da conta e trata-se exatamente do mesmo número de conta utilizado na URL de requisição.
- O atributo "layoutGuid" contém a identificação do layout e trata-se do atributo mais relevante para as demais etapas do processo de abertura do navegador na tela "Ao vivo".
- O atributo "layoutName" contém o nome do layout e pode ser bastante útil para casos onde existam mais de um layout associado ao mesmo número de conta.

A quantidade de layouts associado a um mesmo código de receptora e número de conta é dependente dos cadastrados efetuados na instância do software D-Guard Center.

Dessa forma, o retorno desta API precisa ser analisado para identificar exatamente qual layout se deseja utilizar para abertura do navegador na tela "Ao vivo".

4.4 API REST para obter layouts por código de partição

A API "/contact-id/layouts/{receiver-code}/{account-code}/{partition-code}" retorna a lista de todos os layouts associados a uma determinada partição de uma conta de uma receptora de Contact-Id em uma instância do software D-Guard Center.

[GET] <http://servidor:porta/api/contact-id/layouts/{receiver-code}/{account-code}/{partition-code}>

Exemplo de uso com aplicativo cURL

Requisição:

```
curl -X GET
"http://127.0.0.1:8080/api/contact-id/layouts/10001/00000FF/01"
-H "Accept: application/json" -H "Authorization: Bearer
eyJ1...XfnQ"
```

Retorno:

```
{
  "layouts": [
    {
      "contactIdReceiverCode": "10001",
      "contactIdAccountCode": "00000FF",
      "contactIdPartitionCode": "01",
      "layoutGuid": "{91E696F8-98EB-4312-B75D-BD1755B85285}",
      "layoutName": "PRINCIPAL"
    },
    {
      "contactIdReceiverCode": "10001",
      "contactIdAccountCode": "00000FF",
```

```

        "contactIdPartitionCode": "01",
        "layoutGuid": "{A646D905-993C-4B54-A384-EFED8D595755}",
        "layoutName": "SECUNDÁRIO"
    }
}
}

```

Onde:

- O parâmetro "receiver-code" da URL de requisição deve conter apenas números de no máximo seis dígitos e corresponde ao código da receptora (empresa).
- O parâmetro "account-code" da URL de requisição deve conter numeração em formato hexadecimal de no máximo sete caracteres e corresponde ao número da conta.
- O parâmetro "partition-code" da URL de requisição deve conter numeração em formato hexadecimal de no máximo dois caracteres e corresponde ao código da partição.
- O atributo "contactIdReceiverCode" contém o código da receptora e trata-se exatamente do mesmo código de receptora utilizado na URL de requisição.
- O atributo "contactIdAccountCode" contém o número da conta e trata-se exatamente do mesmo número de conta utilizado na URL de requisição.
- O atributo "contactIdPartitionCode" contém o código da partição e trata-se exatamente do mesmo código de partição utilizado na URL de requisição.
- O atributo "layoutGuid" contém a identificação do layout e trata-se do atributo mais relevante para as demais etapas do processo de abertura do navegador na tela "Ao vivo".
- O atributo "layoutName" contém o nome do layout e pode ser bastante útil para casos onde existam mais de um layout associado ao mesmo número de conta.

A quantidade de layouts associado a um mesmo código de receptora, número de conta e código de partição é dependente dos cadastrados efetuados na instância do software D-Guard Center.

Dessa forma, o retorno desta API precisa ser analisado para identificar exatamente qual layout se deseja utilizar para abertura do navegador na tela "Ao vivo".

4.5 Modelo de URL para abertura da tela "Ao vivo"

A abertura do navegador na tela "Ao vivo" depende da linguagem de programação utilizada para integração com a instância de software D-Guard Center. Entretanto, a montagem da URL de abertura da tela "Ao vivo" seja um modelo predefinido.

O formato da URL de abertura da tela “Ao vivo” é apresentado a seguir:

```
http://servidor:porta/#!/login?token=[token-acesso]&continue={"path":"live","search":{"%22layout%22:%22[layout-guid]%22}}
```

Exemplo de URL formatada

```
http://servidor:porta/#!/login?token=eyJ1...XfnQ&continue={"path":"live","search":{"%22layout%22:%22%7B91E696F8-98EB-4312-B75D-BD1755B85285%7D%22}}
```

Onde:

- O parâmetro “[token-acesso]” se refere ao token de acesso e deve ser substituído pelo valor de retorno da API “/login”.
- O parâmetro “[layout-guid]” se refere ao guid do layout desejado para abertura da tela “Ao vivo” e deve ser substituído pelo valor de retorno da API “/contact-id/layouts”.
 - A tela “Ao vivo” permite apenas a exibição de um único layout por vez, mas este layout pode conter várias câmeras.
 - O valor do guid deve ser formatado para substituir os caracteres “{” por “%7B” e “}” por “%7D”.

4.6 Notas sobre abertura da tela “Ao vivo”

A abertura do navegador na tela “Ao vivo” depende da linguagem de programação utilizada para integração com a instância de software D-Guard Center.

Independente da linguagem de programação utilizada, é sugerido que a abertura da tela “Ao vivo” seja feita sempre na mesma aba ou janela do navegador a fim de evitar a execução de múltiplas instâncias da aplicação D-Guard Web.

Caso a linguagem de programação escolhida seja JavaScript, sugere-se que a abertura da tela “Ao vivo” seja realizada com uso da função `Window.open()`¹ e do parâmetro `windowName`.

Este parâmetro permite especificar o nome de contexto (aba ou janela) no qual a tela “Ao vivo” será carregada. Caso este contexto ainda não esteja em uso, uma nova aba ou janela será abertura, caso contrário, será feito uso da mesma aba ou janela.

¹ A documentação completa da função `Window.open()` está disponível em <https://developer.mozilla.org/en-US/docs/Web/API/Window/open>.

