



D-Guard

Documentação da interface HTTP/REST
Exibição ao vivo e gravações – Versão 1.4

[Histórico de alterações]

Revisão	Data	Autor	Observações
1.0	29.05.2019	John Jaraceski	Versão original
1.1	16.08.2019	John Jaraceski	API exportação de gravações (MP4)
1.2	04.03.2019	Guilherme Aguiar	Cabeçalho Frame-Timestamp com precisão em milisegundos
1.3	21.12.2021	Winicius Moreira	API para obter URL RTSP de gravações no dispositivo
1.4	25.01.2022	Winicius Moreira	API para obter URL RTSP de vídeo ao vivo do dispositivo
1.5	03.02.2022	Wagner Cunha	API para controlar matriz virtual

[Conteúdo]

Sobre este documento	5
Preparação	5
Formato das requisições	5
Formato das respostas	5
Conceitos preliminares	6
Servidor, câmera e dispositivo	6
Transmissão ao vivo e gravações	6
Autenticação	7
API REST para autenticar (login)	7
Transmissão ao vivo	8
Fluxo de transmissão ao vivo	8
API REST para obter servidores	10
API REST para obter câmeras de um servidor	11
API REST para obter streams de uma câmera	12
API REST para obter imagem ao vivo	13
API REST para obter vídeo nativo ao vivo	14
URL RTSP para obter vídeo nativo ao vivo	16
API REST para obter URL RTSP de vídeo ao vivo do dispositivo	17
Transmissão de gravações	18
Fluxo de transmissão de gravações	18
API REST para obter servidores	20
API REST para obter câmeras de um servidor	21
API REST para requisitar gravações	22
API REST para obter imagem de gravações	24
API REST para obter URL RTSP de gravações no dispositivo	25
Exportação de gravações em vídeo MP4	26
Fluxo de exportação de gravações	26
API REST para obter servidores	28
API REST para obter câmeras de um servidor	29
API REST para iniciar exportação de gravações	30
API REST para checar status de processamento	31
API REST para requisitar arquivo de vídeo	32
Matriz virtual	33
Fluxo para listar estações e monitores	33
Fluxo para listar layouts	33
Fluxo para controlar o layout aberto em uma estação	34

1 Sobre este documento

Este documento fornece informações sobre a interface HTTP/REST dos softwares D-Guard (versão 7 ou superior) para a configuração do monitoramento de eventos de desconexão de servidores e perda de sinal de câmeras com execução de requisição HTTP/GET no momento do disparo dos eventos.

2 Preparação

Ative o serviço API REST do D-Guard, 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 exibição de conteúdo ao vivo ou gravações de câmeras, é preciso entender alguns conceitos preliminares sobre as principais entidades envolvidas neste processo: servidor, câmera e dispositivo; transmissão ao vivo e gravações.

3.1 Servidor, câmera e dispositivo

Os softwares D-Guard 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 normalmente apenas um canal de vídeo porque representa uma única câmera. Enquanto que um servidor do tipo NVR/DVR pode conter múltiplos canais de vídeo.

Além disso, um servidor também pode estar vinculado a uma outra instância D-Guard. 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.

3.2 Transmissão ao vivo e gravações

Os softwares D-Guard permitem a transmissão ao vivo de streams de câmeras tanto em formato Jpeg como nativo. A definição dos parâmetros de transmissão¹ dos streams pode ser configurada na aba "Configuração de Vídeo Multi-Stream" da tela "Configuração de Câmeras".

Além disso, o software D-Guard também permite a gravação ao vivo de streams de câmeras. Sendo que o conteúdo gravado é armazenado no formato nativo transmitido pela câmera. Uma vez armazenada, uma gravação pode ser pesquisada e exibida diretamente no software D-Guard com o uso do aplicativo Player.

¹ Algumas câmeras somente permitem a configuração dos parâmetros de transmissão diretamente pela interface (web) do próprio dispositivo que é fornecida pelo fabricante.

4 Autenticação

Os softwares D-Guard dispõem de uma API REST que permite a aplicações externas o acesso a inúmeras funcionalidades tanto para consulta como para administração. Para tanto, é necessário que as aplicações externas realizem um processo de autenticação com uso do formato "Authorization: Bearer + Token" do HTTP. Caso necessário, utilize a documentação do protocolo HTTP (rfc 2617) para maiores informações sobre esse método de autenticação.

4.1 API REST para autenticar (login)

A API "/login" permite a autenticação na instância do software D-Guard e retorna o *token* de autenticação que deverá ser utilizado em todas as demais requisições HTTP à API REST, exceto pela API "/version".

```
[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"
  }
}
```

5 Transmissão ao vivo

A transmissão ao vivo de streams de câmeras pode ser recuperada em formato Jpeg ou nativo. Sendo que o formato Jpeg é o mais compatível com quaisquer tipos de aplicações cliente, seja um navegador web, uma aplicação desktop ou mobile.

A exibição em formato Jpeg é feita com uso de uma API REST específica do software D-Guard que efetua a conversão do formato nativo utilizado pela câmera para o formato Jpeg. Enquanto que o formato nativo faz uso de outra API REST que retransmite o conteúdo original recuperado da câmera.

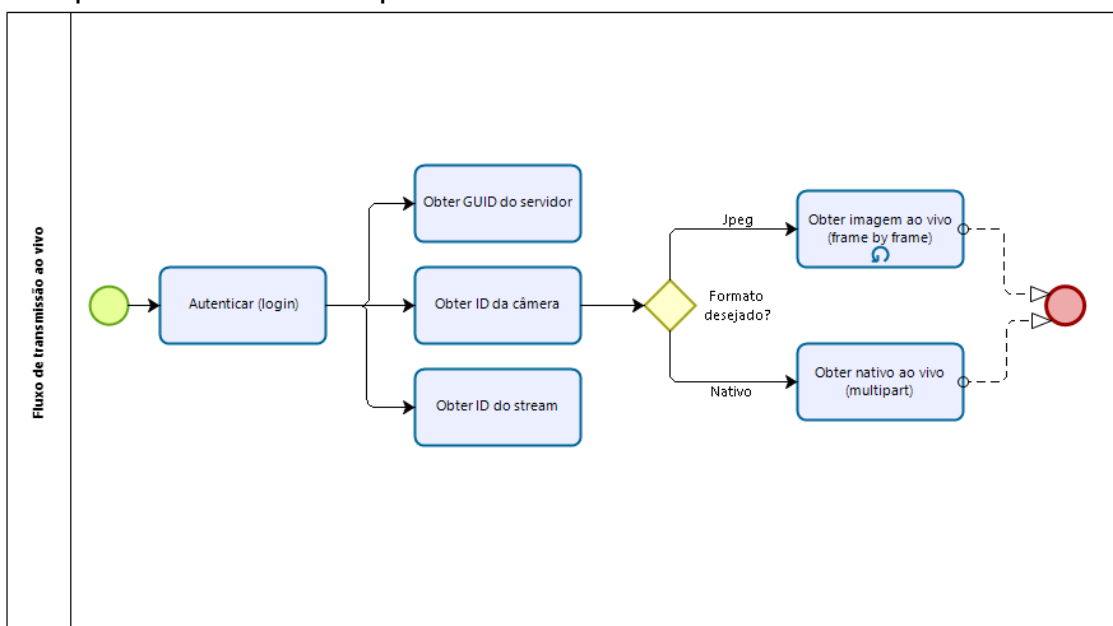
Uma grande diferença entre estas duas opções de formato – além do formato em si – está na maneira como os dados são recebidos pelo cliente da API REST.

No caso do formato Jpeg, a cada requisição HTTP enviada a API REST, é retornado um e apenas um único frame em formato Jpeg. Então, para se obter o próximo frame, é necessário o envio de uma nova requisição HTTP. Sendo que o cliente da API REST tem total controle sobre o fluxo de recebimento dos frames.

Já no caso do formato nativo, uma única requisição HTTP deve ser enviada a API REST, pois é retornado um fluxo contínuo contendo o conteúdo original da câmera em formato HTTP Multipart.

5.1 Fluxo de transmissão ao vivo

A imagem abaixo apresenta o fluxo geral de transmissão ao vivo para um stream específico de uma câmera.



A etapa inicial ou preliminar se refere a autenticação ou login e sem ela não é possível realizar nenhuma das etapas posteriores. Nessa etapa é feita a autenticação com uso de credenciais de acesso para se obter o *token* de autenticação. Este *token* de autenticação será necessário em todas as requisições feitas nas etapas posteriores e deverá ser enviado no cabeçalho da requisição HTTP.

As três etapas seguintes, após a autenticação, se referem a obtenção de dados básicos para realizar a recuperação de vídeo ao vivo.

Onde:

- “Obter GUID do servidor” se refere a obtenção do GUID do servidor que se deseja recuperar a transmissão ao vivo. Uma vez obtida a identificação do servidor, essa será a mesma independente de interrupções na execução do software D-Guard e pode até mesmo ser mantida como uma constante no lado cliente.
- “Obter ID da câmera” se refere a obtenção da identificação da câmera que se deseja recuperação a transmissão ao vivo. A identificação da câmera consiste em um número sequencial que identifica a câmera na lista de câmeras do servidor. Normalmente, essa se mantém constante enquanto não houver troca de modelo do servidor.
- “Obter ID do stream” se refere a obtenção da identificação do stream que se deseja recuperar a transmissão ao vivo. A identificação do stream consiste em um número sequencial que identifica o stream na lista de streams da câmera. Normalmente, essa se mantém constante enquanto não houver troca de modelo do servidor ou mudança nos parâmetros de transmissão da câmera. Essa etapa é opcional no fluxo e caso não seja realizada, será utilizado automaticamente o stream padrão de transmissão.

O ponto de decisão “Formato desejado?”, se refere a escolha do formato de recuperação: Jpeg ou nativo. A escolha desse formato normalmente é feita durante a etapa de análise / construção do processo de desenvolvimento da aplicação cliente.

É importante ressaltar que dependendo do formato de recuperação escolhido, a aplicação cliente fará uso de diferentes APIs REST para obtenção do conteúdo. Além disso, o próprio conteúdo em si será diferente e, no caso do conteúdo em formato nativo, este poderá ser alterado dinamicamente devido a modificações nos parâmetros de transmissão da câmera.

Caso seja escolhido o formato Jpeg, a etapa seguinte, "Obter imagem ao vivo (frame by frame)", se refere a recuperação de imagens em formato Jpeg para cada um dos frames transmitidos pela câmera. E a recuperação de cada frame deve ser feita manualmente pela aplicação cliente com o envio contínuo e sequencial de requisições HTTP a API REST.

Dessa forma, para realizar a exibição de um segundo de vídeo ao vivo de uma câmera configurada com 24 fps (*frames per second*), será necessário o envio contínuo e sequencial de 24 requisições HTTP a API REST.

Caso seja escolhido o formato nativo, a etapa seguinte, "Obter nativo ao vivo (multipart)", se refere a recuperação de um fluxo contínuo do conteúdo original recuperado da câmera com o envio de uma única requisição HTTP a API REST.

É importante ressaltar que no caso do formato nativo, não será necessário o envio de requisições HTTP adicionais porque os frames serão retornados em um fluxo contínuo.

5.2 API REST para obter servidores

A API "/servers" permite o gerenciamento dos servidores de uma instância do software D-Guard.

Neste momento, abordaremos apenas o método [GET] responsável por retornar a lista dos servidores previamente cadastrados.

[GET] <http://servidor:porta/api/servers>

Exemplo de uso com aplicativo cURL

Requisição:

```
curl -X GET "http://127.0.0.1:8080/api/servers" -H "Accept: application/json" -H "Authorization: Bearer eyJ1...XfnQ"
```

Retorno:

```
{
  "servers": [
    {
      "name": "Servidor DVR",
      "guid": "{182C4DA9-7CA7-416D-9F72-F3527941FA2D}",
      "parentGuid": null,
      "active": true,
      "connected": true
    },
    {
      "name": "Servidor Câmera",
      "guid": "{38A9F3AC-3522-42D2-818B-183004858406}",
      "parentGuid": null,
    }
  ]
}
```

```

        "active": true,
        "connected": true
    }
]
}

```

Onde:

- O atributo "guid" contém a identificação do servidor e trata-se do atributo mais relevante para as demais etapas do processo de recuperação de vídeo ao vivo.
- O atributo "active" indica se o servidor está ativo / habilitado.
- O atributo "connected" indica se o servidor está conectado neste momento. Então, é possível que um servidor esteja ativo e desconectado devido a uma falha de rede ou qualquer outro tipo de problema de transmissão.
- O atributo "parentGuid" contém a identificação do servidor raiz ao qual o servidor está vinculado. Neste caso, o servidor raiz se trata de uma outra instância do software D-Guard.

5.3 API REST para obter câmeras de um servidor

A API "/servers/{guid}/cameras" permite o gerenciamento das câmeras vinculadas a um servidor de uma instância do software D-Guard.

Neste momento, abordaremos apenas o método [GET] responsável por retornar a lista das câmeras de um servidor previamente cadastrado.

[GET] <http://servidor:porta/api/servers/{guid}/cameras>

Onde:

- O parâmetro "guid" contém a identificação de um servidor previamente cadastrado.

Exemplo de uso com aplicativo cURL

Requisição:

```

curl -X GET
"http://127.0.0.1:8080/api/servers/%7B182C4DA9-7CA7-416D-9F72-F35
27941FA2D%7D/cameras" -H "Accept: application/json" -H
"Authorization: Bearer eyJ1...XfnQ"

```

Retorno:

```

{
  "cameras": [
    {
      "id": 0,
      "name": "Câmera 1"
    }
  ]
}

```

```

    },
    {
      "id": 1,
      "name": "Câmera 2"
    },
    {
      "id": 2,
      "name": "Câmera 3"
    },
    {
      "id": 3,
      "name": "Câmera 4"
    }
  ]
}

```

Onde:

- O atributo "id" contém a identificação da câmera e trata-se do atributo mais relevante para as demais etapas do processo de recuperação de vídeo ao vivo. É importante ressaltar que a identificação de uma câmera é única para um determinado servidor, mas pode se repetir entre diferentes servidores.
- O atributo "name" contém o nome da câmera e normalmente é gerado automaticamente. Entretanto, pode ser modificado pelo usuário para conter um valor mais significativo.

A recuperação de vídeo ao vivo está associada diretamente a uma câmera. Neste caso, além da identificação do servidor, é necessária também a identificação da câmera.

5.4 API REST para obter streams de uma câmera

A API `/servers/{guid}/cameras/{id}/streams` permite o gerenciamento dos streams vinculadas a uma câmera específica de um servidor de uma instância do software D-Guard.

Neste momento, abordaremos apenas o método [GET] responsável por retornar a lista dos streams de uma câmera de um servidor previamente cadastrado.

[GET]

<http://servidor:porta/api/servers/{guid}/cameras/{id}/streams>

Onde:

- O parâmetro "guid" contém a identificação de um servidor previamente cadastrado.
- O parâmetro "id" contém a identificação de uma câmera do servidor.

Exemplo de uso com aplicativo cURL

Requisição:

```
curl -X GET
"http://127.0.0.1:8080/api/servers/%7B182C4DA9-7CA7-416D-9F72-F35
27941FA2D%7D/cameras/0/streams" -H "Accept: application/json" -H
"Authorization: Bearer eyJ1...XfnQ"
```

Retorno:

```
{
  "streams": [
    {
      "id": 0,
      "enabled": true
    },
    {
      "id": 1,
      "enabled": false
    },
    {
      "id": 2,
      "enabled": true
    }
  ]
}
```

Onde:

- O atributo "id" contém a identificação do stream e trata-se do atributo mais relevante para as demais etapas do processo de recuperação de vídeo ao vivo. É importante ressaltar que a identificação de um stream é única para uma determinada câmera de um servidor, mas pode se repetir entre diferentes servidores.
- O atributo "enabled" indica se o stream está habilitado para transmissão. Sendo que um stream não habilitado estará indisponível para exibição ao vivo de transmissão.

A recuperação de vídeo ao vivo está associada diretamente a uma câmera e, opcionalmente, a um stream. Neste caso, além das identificações do servidor e da câmera, é necessária também a identificação do stream caso se deseja fazer a recuperação de um stream específico (diferente do stream padrão de visualização).

5.5 API REST para obter imagem ao vivo

A API `"/servers/{guid}/cameras/{id}/image.jpg"` permite a recuperação de um frame de vídeo ao vivo associado ao stream padrão de visualização de uma câmera de um servidor de uma instância do software D-Guard.

Também é possível a recuperação de frames de um stream específico usando-se uma API com a identificação do stream desejado: `"/servers/{guid}/cameras/{id}/streams/{id}/image.jpg"`

[GET]

<http://servidor:porta/api/servers/{guid}/cameras/{id}/image.jpg>

Exemplo de uso com aplicativo cURL

Requisição:

```
curl -X GET
"http://127.0.0.1:8080/api/servers/%7B182C4DA9-7CA7-416D-9F72-F35
27941FA2D%7D/cameras/0/image.jpg" -H "Accept: application/json"
-H "Authorization: Bearer eyJ1...XfnQ"
```

Retorno:

```
FF D8 FF E0 00 10 4A 46 49 46 00 01 01 00 00 01 00 01 00 00 FF
...
2A FA 1F 10 9A 6C 9F EA 9B E8 69 C6 98 E7 F7 4D F4 34 01 FF D9
```

O retorno desta API contém conteúdo binário associado a uma imagem em formato Jpeg que poderá ser renderizado ou manipulado pela aplicação cliente.

A resposta virá com um cabeçalho personalizado chamado Frame-Timestamp. Seu valor é preenchido com o horário associado ao frame, em formato UTC com a precisão em milissegundos. Exemplo: 2020-03-04T16.33.19.423.

O cabeçalho padrão Last-Modified retornará o mesmo horário, porém em formato GMT com a precisão em segundos. Exemplo: Wed, 04 Mar 2020 19:33:19 GMT

A recuperação de vídeo ao vivo precisa estar em algum tipo de *loop* contínuo para que a aplicação cliente faça a exibição com fluidez. Então, novas requisições HTTP precisam ser enviadas de forma contínua e sequencial para recuperar cada um dos próximos frames a serem exibidos.

É importante observar que essa conversão dos frames para Jpeg exige um uso de CPU no servidor considerável (devido ao processo de transcoding de vídeo), devendo ser usado o formato nativo sempre que possível.

5.6 API REST para obter vídeo nativo ao vivo

A API `"/servers/{guid}/cameras/{id}/native-video"` permite a recuperação do conteúdo original em formato HTTP Multipart associado ao stream padrão de visualização de uma câmera de um servidor de uma instância do software D-Guard.

Também é possível a recuperação do conteúdo original de um stream específico usando-se uma API com a identificação do stream desejado: `"/servers/{guid}/cameras/{id}/streams/{id}/native-video"`

[GET]

<http://servidor:porta/api/servers/{guid}/cameras/{id}/native-video>

Exemplo de uso com aplicativo cURL

Requisição:

```
curl -X GET
"http://127.0.0.1:8080/api/servers/%7B182C4DA9-7CA7-416D-9F72-F35
27941FA2D%7D/cameras/0/native-video" -H "Accept:
application/json" -H "Authorization: Bearer eyJ1...XfnQ"
```

Retorno:

```
HTTP/1.1 206 Partial Content
Content-Type: multipart/mixed; boundary=bhqgRKVEQFPtg7LeTTUc
Availability-Start-Time: Wed, 01 Jan 2019 00:00:00 GMT
Video-Codec: h264
Cache-Control: no-cache
Access-Control-Allow-Origin: *
```

```
Last-Modified: 2019-01-01T00:00:00
Content-Type: video/h264
Content-Length: 65357
```

```
FF D8 FF E0 00 10 4A 46 49 46 00 01 01 00 00 01 00 01 00 00 FF
...
2A FA 1F 10 9A 6C 9F EA 9B E8 69 C6 98 E7 F7 4D F4 34 01 FF D9
```

```
--bhqgRKVEQFPtg7LeTTUc
Last-Modified: 2019-01-01T00:00:00
Content-Type: video/h264
Content-Length: 26676
```

```
FF D8 FF E0 00 10 4A 46 49 46 00 01 01 00 00 01 00 01 00 00 FF
...
2A FA 1F 10 9A 6C 9F EA 9B E8 69 C6 98 E7 F7 4D F4 34 01 FF D9
```

```
--bhqgRKVEQFPtg7LeTTUc
Last-Modified: 2019-01-01T00:00:00
Content-Type: video/h264
Content-Length: 36676
```

```
FF D8 FF E0 00 10 4A 46 49 46 00 01 01 00 00 01 00 01 00 00 FF
...
2A FA 1F 10 9A 6C 9F EA 9B E8 69 C6 98 E7 F7 4D F4 34 01 FF D9
```

O retorno desta API contém conteúdo em formato HTTP Multipart e cada trecho retornado (*body part*) contém binário associado ao conteúdo original retornado pela câmera e retransmitido pela API REST que poderá ser renderizado ou manipulado pela aplicação cliente.

É importante ressaltar que a resposta da requisição HTTP segue o formato HTTP Multipart e que maiores detalhes sobre o mesmo podem ser obtidos na documentação oficial disponível em https://www.w3.org/Protocols/rfc1341/7_2_Multipart.html.

Além disso, o conteúdo de cada trecho retornado (*body part*) contém algumas informações encapsuladas em cabeçalhos específicos:

- A identificação do trecho (*body part*) é gerada aleatoriamente a cada requisição HTTP e deve ser obtida a partir do item de cabeçalho "Content-Type" da resposta de base HTTP.
- O item de cabeçalho "Last-Modified" contém a data/hora associada ao conteúdo original retornado pela câmera.
- O item de cabeçalho "Content-Type" contém o *codec* utilizado na codificação do conteúdo original retornado pela câmera.
- O item de cabeçalho "Content-Length" contém a quantidade de *bytes* do trecho retornado (*body part*) e que devem ser lidos pela aplicação cliente.

A recuperação de vídeo ao vivo precisa fazer a leitura contínua da resposta da requisição HTTP, extrair e processar cada trecho retornado (*body part*). O processamento de cada trecho é bastante dependente do *codec* utilizado pela câmera e, provavelmente, será repassado para alguma biblioteca externa de manipulação de imagem/vídeo como FFmpeg e similares.

5.7 URL RTSP para obter vídeo nativo ao vivo

A transmissão ao vivo de streams de câmera também pode ser realizada diretamente com uso do protocolo RTSP ao invés de requisições HTTP.

A URL RTSP para transmissão em tempo real de streams de câmera segue o seguinte formato:

<rtsp://usuario:senha@servidor:porta/servers/{guid}/cameras/{id}>

Onde:

- servidor = domínio, nome na rede, ou IP do servidor D-Guard
- porta = número da porta em que o serviço RTSP está ativo
- usuario = nome do usuário previamente cadastrado
- senha = senha do usuário previamente cadastrado
- guid = identificação do servidor que se deseja recuperar a transmissão ao vivo
- id = identificação da câmera que se deseja recuperar a transmissão ao vivo

Exemplo:

- `rtsp://admin:seventh@192.168.0.130:554/servers/C4C0867E-BE17-4C24-9828-9DBF2D73E169/cameras/0`

Também é possível recuperar a transmissão ao vivo de um stream específico de uma câmera. Neste caso, deve-se utilizar uma versão estendida da URL RTSP:

<rtsp://usuario:senha@servidor:porta/servers/{guid}/cameras/{id}/streams/{id}>

Exemplo:

- `rtsp://admin:seventh@192.168.0.130:554/servers/C4C0867E-BE17-4C24-9828-9DBF2D73E169/cameras/0/streams/2`

A informação relacionada a porta do serviço RTSP está disponível na tela “Configurações de Transmissão”. E também pode ser obtida usando-se a API “/transmission/rtsp-server” que permite a recuperação das configurações do servidor RTSP de uma instância do software D-Guard.

É importante ressaltar que serviço RTSP precisa estar ativo na instância do software D-Guard para que a transmissão em tempo real com protocolo RTSP esteja disponível para uso.

5.8 API REST para obter URL RTSP de vídeo ao vivo do dispositivo

A API “/servers/{guid}/cameras/{id}/streams/{id}/remote-url” permite a recuperação de um URL RTSP para visualização das imagens ao vivo diretamente do dispositivo escolhido, formatada com os parâmetros do stream solicitado.

[GET]
<http://servidor:porta/api/servers/{guid}/cameras/{id}/streams/{id}/remote-url>

Onde:

- O parâmetro “guid” contém a identificação de um servidor previamente cadastrado.
- O parâmetro “id” contém a identificação de uma câmera do servidor.
- O parâmetro “stream” contém a identificação de um stream da câmera.

Exemplo de uso com aplicativo cURL

Requisição:

```
curl -X GET  
"http://127.0.0.1:8080/api/servers/%7BDC416B0F-0044-4DD4-8357-762  
A53E3F2A0%7D/cameras/0/streams/0/remote-url" -H "accept:  
application/json" -H "Authorization: Bearer eyJl...XfnQ"
```

Retorno:

```
{  
  "remoteUrl": {  
    "url": string,  
    "username": string,  
    "password": string  
  }  
}
```

Para o funcionamento dessa API é necessário que o dispositivo selecionado tenha suporte ao protocolo RTSP e esteja devidamente homologado ao sistema.

6 Transmissão de gravações

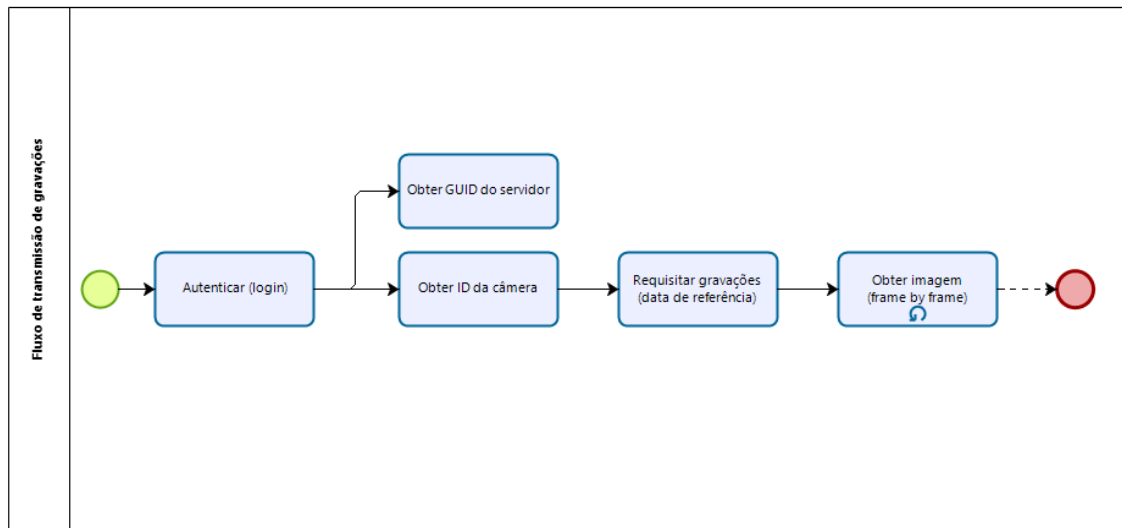
A transmissão de gravações de câmeras apenas é oferecida em formato Jpeg, embora sejam armazenadas internamente pelo software D-Guard no formato original retornado pela câmera.

A recuperação de gravações em formato Jpeg é feita com uso de uma API REST específica do software D-Guard que efetua a conversão do formato nativo utilizado pela câmera para o formato Jpeg.

A cada requisição HTTP enviada a API REST, é retornado um e apenas um frame em formato Jpeg. Então, para se obter o próximo frame, é necessário o envio de uma nova requisição HTTP. Sendo que o cliente da API REST tem total controle sobre o fluxo de recebimento dos frames.

6.1 Fluxo de transmissão de gravações

A imagem abaixo apresenta o fluxo geral de transmissão de gravações de um stream específico de uma câmera.



A etapa inicial ou preliminar se refere a autenticação ou login e sem ela não é possível realizar nenhuma das etapas posteriores. Nessa etapa é feita a autenticação com uso de credenciais de acesso para se obter o *token* de autenticação. Este *token* de autenticação será necessário em todas as requisições feitas nas etapas posteriores e deverá ser enviado no cabeçalho da requisição HTTP.

As três etapas seguintes, após a autenticação, se referem a obtenção de dados básicos para realizar a recuperação de gravações.

Onde:

- “Obter GUID do servidor” se refere a obtenção do GUID do servidor que se deseja recuperar as gravações. Uma vez obtida a identificação do servidor, essa será a mesma independente de interrupções na execução do software D-Guard e pode até mesmo ser mantida como uma constante no lado cliente.
- “Obter ID da câmera” se refere a obtenção da identificação da câmera que se deseja recuperar as gravações. A identificação da câmera consiste em um número sequencial que identifica a câmera na lista de câmeras do servidor. Normalmente, essa se mantém constante enquanto não houver troca de modelo do servidor.

A etapa “Requisitar gravações (data de referência)”, se refere ao envio de uma requisição para definição da data de referência (inicial) para procura das gravações armazenadas. Internamente, o software D-Guard fará a procura da gravação mais próxima a partir da data de referência desejada. Caso não seja encontrada nenhuma gravação para a data de referência, será retornada a gravação mais próxima que poderá ser apenas alguns segundos após a data de referência ou até mesmo horas, dias ou meses depois.

Após obter essa identificação para recuperação de gravações, deve-se seguir para a próxima etapa, "Obter imagem (frame by frame)". Essa etapa se refere a recuperação de imagens em formato Jpeg para cada um dos frames das gravações armazenadas para a câmera. E a recuperação de cada frame deve ser feita manualmente pela aplicação cliente com o envio contínuo e sequencial de requisições HTTP a API REST.

Dessa forma, para realizar a exibição de um segundo de transmissão de gravações de uma câmera configurada com 24 fps (*frames per second*), será necessário o envio contínuo e sequencial de 24 requisições HTTP a API REST.

6.2 API REST para obter servidores

A API `/servers` permite o gerenciamento dos servidores de uma instância do software D-Guard.

Neste momento, abordaremos apenas o método [GET] responsável por retornar a lista dos servidores previamente cadastrados.

[GET] <http://servidor:porta/api/servers>

Exemplo de uso com aplicativo cURL

Requisição:

```
curl -X GET "http://127.0.0.1:8080/api/servers" -H "Accept: application/json" -H "Authorization: Bearer eyJl...XfnQ"
```

Retorno:

```
{
  "servers": [
    {
      "name": "Servidor DVR",
      "guid": "{182C4DA9-7CA7-416D-9F72-F3527941FA2D}",
      "parentGuid": null,
      "active": true,
      "connected": true
    },
    {
      "name": "Servidor Câmera",
      "guid": "{38A9F3AC-3522-42D2-818B-183004858406}",
      "parentGuid": null,
      "active": true,
      "connected": true
    }
  ]
}
```

Onde:

- O atributo "guid" contém a identificação do servidor e trata-se do atributo mais relevante para as demais etapas do processo de recuperação de gravações.
- O atributo "active" indica se o servidor está ativo / habilitado.
- O atributo "connected" indica se o servidor está conectado neste momento. Então, é possível que um servidor esteja ativo e desconectado devido a uma falha de rede ou qualquer outro tipo de problema de transmissão.
- O atributo "parentGuid" contém a identificação do servidor raiz ao qual o servidor está vinculado. Neste caso, o servidor raiz se trata de uma outra instância do software D-Guard.

6.3 API REST para obter câmeras de um servidor

A API "/servers/{guid}/cameras" permite o gerenciamento das câmeras vinculadas a um servidor de uma instância do software D-Guard.

Neste momento, abordaremos apenas o método [GET] responsável por retornar a lista das câmeras de um servidor previamente cadastrado.

[GET] <http://servidor:porta/api/servers/{guid}/cameras>

Onde:

- O parâmetro "guid" contém a identificação de um servidor previamente cadastrado.

Exemplo de uso com aplicativo cURL

Requisição:

```
curl -X GET
"http://127.0.0.1:8080/api/servers/%7B182C4DA9-7CA7-416D-9F72-F35
27941FA2D%7D/cameras" -H "Accept: application/json" -H
"Authorization: Bearer eyJ1...XfnQ"
```

Retorno:

```
{
  "cameras": [
    {
      "id": 0,
      "name": "Câmera 1"
    },
    {
      "id": 1,
      "name": "Câmera 2"
    },
    {
      "id": 2,
```

```

        "name": "Câmera 3"
    },
    {
        "id": 3,
        "name": "Câmera 4"
    }
]
}

```

Onde:

- O atributo "id" contém a identificação da câmera e trata-se do atributo mais relevante para as demais etapas do processo de recuperação de gravações. É importante ressaltar que a identificação de uma câmera é única para um determinado servidor, mas pode se repetir entre diferentes servidores.
- O atributo "name" contém o nome da câmera e normalmente é gerado automaticamente. Entretanto, pode ser modificado pelo usuário para conter um valor mais significativo.

A exibição de transmissão de gravações está associada diretamente a uma câmera. Neste caso, além da identificação do servidor, é necessária também a identificação da câmera.

6.4 API REST para requisitar gravações

A API `"/servers/{guid}/cameras/{id}/recordings/images"` permite a requisição de gravações armazenadas associado ao stream padrão de gravação de uma câmera de uma instância do software D-Guard.

[POST]

<http://servidor:porta/api/servers/{guid}/cameras/{id}/recordings/images>

```

{
    "searchDateTime": [string],
    "resize": [boolean],
    "width": [int],
    "height": [int],
    "secondsBetweenFrames": [int]
}

```

Onde:

- O parâmetro "guid" contém a identificação de um servidor previamente cadastrado.
- O parâmetro "id" contém a identificação de uma câmera do servidor.
- O atributo "searchDateTime" contém a data de referência para procura das gravações armazenadas. Este atributo deve ser informado no format "yyyy-MM-ddTHH:mm:ss".

- (opcional) O atributo "resize" indica se as imagens a serem retornadas devem ser redimensionadas ou mantidas no tamanho original.
- (opcional) O atributo "width" contém a largura a ser usada para redimensionamento das imagens a serem retornadas. Este atributo somente deve ser informado se o atributo "resize" for especificado com o valor "true".
- (opcional) O atributo "height" contém a altura a ser usada para redimensionamento das imagens a serem retornadas. Este atributo somente deve ser informado se o atributo "resize" for especificado com o valor "true".
- (opcional) O atributo "secondsBetweenFrames" contém o intervalo (em segundos) entre frames a ser usado para recuperação das imagens a serem retornadas. Este atributo permite acelerar a exibição das imagens ao fazer saltos (em segundos) entre os frames.

Exemplo de uso com aplicativo cURL

Requisição:

```
curl -X POST
"http://127.0.0.1:8080/api/servers/%7B182C4DA9-7CA7-416D-9F72-F35
27941FA2D%7D/cameras/0/recordings/images" -H "Accept:
application/json" -H "Authorization: Bearer eyJl...XfnQ" -H
"Content-Type: application/json" -d "{ \"searchDateTime\":
\"2019-01-01T00:00:00\"}"
```

Retorno:

```
{
  "image": {
    "requestGuid": "{232C4ABC-8CC1-321A-F721-27BAD941239D}",
    "searchDateTime": "2019-01-01T00:00:00"
  }
}
```

Onde:

- O atributo "requestGuid" contém a identificação da requisição de recuperação de gravações e trata-se do atributo mais relevante para as demais etapas do processo de recuperação de gravações. É importante ressaltar que essa identificação é única e jamais será repetida, mesmo entre diferentes servidores.
- O atributo "searchDateTime" contém a data de referência utilizada para procura das gravações armazenadas.

A requisição de recuperação de gravações retornará uma resposta com erro HTTP/422 caso não existam gravações armazenadas entre a data de referência desejada e o momento atual.

6.5 API REST para obter imagem de gravações

A API `"/servers/{guid}/cameras/{id}/recordings/images"` permite a recuperação de um frame de vídeo gravado associado ao stream padrão de gravação de uma câmera de um servidor de uma instância do software D-Guard.

[GET]

<http://servidor:porta/api/servers/{guid}/cameras/{id}/recordings/images/{request}>

Onde:

- O parâmetro "guid" contém a identificação de um servidor previamente cadastrado.
- O parâmetro "id" contém a identificação de uma câmera do servidor.
- O atributo "request" contém a identificação da requisição de recuperação de gravações previamente realizada.

Exemplo de uso com aplicativo cURL

Requisição:

```
curl -X GET
"http://127.0.0.1:8080/api/servers/%7B182C4DA9-7CA7-416D-9F72-F35
27941FA2D%7D/cameras/0/recordings/images/%7B232C4ABC-8CC1-321A-F7
21-27BAD941239D%7D" -H "Accept: application/json" -H
"Authorization: Bearer eyJ1...XfnQ"
```

Retorno:

```
FF D8 FF E0 00 10 4A 46 49 46 00 01 01 00 00 01 00 01 00 00 FF
...
2A FA 1F 10 9A 6C 9F EA 9B E8 69 C6 98 E7 F7 4D F4 34 01 FF D9
```

O retorno desta API contém conteúdo binário associado a uma imagem em formato Jpeg que poderá ser renderizado ou manipulado pela aplicação cliente.

A recuperação de gravações precisa estar em algum tipo de *loop* contínuo para que a aplicação cliente faça a exibição com fluidez. Então, novas requisições HTTP precisam ser enviadas de forma contínua e sequencial para recuperar cada um dos próximos frames a serem exibidos.

A resposta virá com um cabeçalho personalizado chamado Frame-Timestamp. Seu valor é preenchido com o horário associado ao frame, em formato UTC com a precisão em milissegundos. Exemplo: 2020-03-04T16.33.19.423.

O cabeçalho padrão Last-Modified retornará o mesmo horário, porém em formato GMT com a precisão em segundos. Exemplo: Wed, 04 Mar 2020 19:33:19 GMT

6.6 API REST para obter URL RTSP de gravações no dispositivo

A API `"/servers/{guid}/cameras/{id}/recordings/remote-url/{start}/{end}"` permite a recuperação de um URL RTSP para reprodução das gravações diretamente no dispositivo escolhido, formatada com os parâmetros do período solicitado.

[GET]
<http://servidor:porta/api/servers/{guid}/cameras/{id}/recordings/remote-url/{start}/{end}>

Onde:

- O parâmetro "guid" contém a identificação de um servidor previamente cadastrado.
- O parâmetro "id" contém a identificação de uma câmera do servidor.
- O parâmetro "start" contém o horário inicial de reprodução.
- O parâmetro "end" contém o horário final de reprodução.

Exemplo de uso com aplicativo cURL

Requisição:

```
curl -X GET
"http://127.0.0.1:8080/api/servers/%7BDC416B0F-0044-4DD4-8357-762
A53E3F2A0%7D/cameras/0/recordings/remote-url/2021-12-21T08%3A00%3
A00/2021-12-21T08%3A30%3A00" -H "accept: application/json" -H
"Authorization: Bearer eyJ1...XfnQ"
```

Retorno:

```
{
  "remoteUrl": {
    "url": string,
    "username": string,
    "password": string
  }
}
```

Para o funcionamento dessa API é necessário que o dispositivo selecionado tenha suporte a reprodução de gravações através do protocolo RTSP e esteja devidamente homologado ao sistema.

7 Exportação de gravações em vídeo MP4

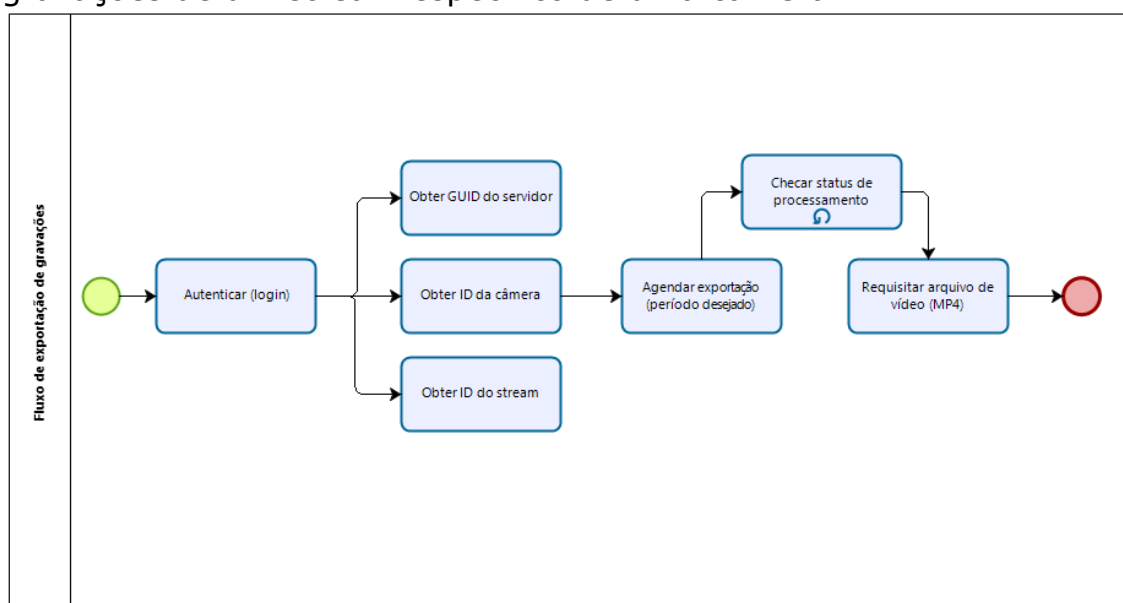
Embora a transmissão de gravações de câmeras seja oferecida apenas em formato Jpeg, é possível exportar as gravações de um período específico em arquivo de vídeo (formato MP4).

A exportação de gravações em vídeo (MP4) é feita com uso de uma API REST específica do software D-Guard que efetua a conversão do formato nativo utilizado pela câmera para o formato MP4.

Esta API REST realiza a geração do vídeo de maneira assíncrona devido ao tempo necessário para efetuar a conversão dos arquivos de gravação em formato nativo (criados pelo D-Guard) para o formato MP4.

7.1 Fluxo de exportação de gravações

A imagem abaixo apresenta o fluxo geral de exportação de gravações de um stream específico de uma câmera.



A etapa inicial ou preliminar se refere a autenticação ou login e sem ela não é possível realizar nenhuma das etapas posteriores. Nessa etapa é feita a autenticação com uso de credenciais de acesso para se obter o *token* de autenticação. Este *token* de autenticação será necessário em todas as requisições feitas nas etapas posteriores e deverá ser enviado no cabeçalho da requisição HTTP.

As três etapas seguintes, após a autenticação, se referem a obtenção de dados básicos para realizar a exportação de gravações.

Onde:

- “Obter GUID do servidor” se refere a obtenção do GUID do servidor que se deseja exportar as gravações. Uma vez obtida a identificação do servidor, essa será a mesma independente de interrupções na execução do software D-Guard e pode até mesmo ser mantida como uma constante no lado cliente.
- “Obter ID da câmera” se refere a obtenção da identificação da câmera que se deseja exportar as gravações. A identificação da câmera consiste em um número sequencial que identifica a câmera na lista de câmeras do servidor. Normalmente, essa se mantém constante enquanto não houver troca de modelo do servidor.

A etapa “Iniciar exportação (período desejado)”, se refere ao envio de uma requisição para iniciar a exportação de gravações em vídeo (MP4) de um período específico (data/hora inicial e final) para exportação das gravações armazenadas. Internamente, o software D-Guard fará a procura de todas as gravações armazenadas dentro do período especificado. Caso sejam encontradas gravações para o período desejado, será retornada uma chave de identificação da requisição de exportação de gravações. Caso contrário, quando nenhuma gravação for encontrada para o período especificado, será retornada uma mensagem de erro indicando que não existem gravações para o período especificado.

Após obter essa identificação, deve-se seguir para a próxima etapa, “Checar status de processamento”. Essa etapa se refere a recuperação do status de processamento da exportação iniciada na etapa anterior. E, também, nessa mesma etapa é possível acompanhar o andamento do processamento da exportação através da análise da informação sobre o progresso da exportação (valor percentual).

Finalmente, após o término do processamento da exportação, segue-se para a etapa final da exportação, “Requisitar arquivo de vídeo (MP4)”. Essa etapa se refere a recuperação do arquivo de vídeo em formato MP4 para o período especificado no início da exportação.

É importante ressaltar que na etapa de checagem do status de processamento é possível que a API REST retorne um status de erro indicando que não foi possível realizar a exportação de gravações para o período especificado. Problemas na exportação podem ocorrer por inúmeras razões, tais como arquivos corrompidos ou quantidade de frames insuficientes para geração do arquivo de vídeo (MP4).

7.2 API REST para obter servidores

A API `/servers` permite o gerenciamento dos servidores de uma instância do software D-Guard.

Neste momento, abordaremos apenas o método [GET] responsável por retornar a lista dos servidores previamente cadastrados.

[GET] <http://servidor:porta/api/servers>

Exemplo de uso com aplicativo cURL

Requisição:

```
curl -X GET "http://127.0.0.1:8080/api/servers" -H "Accept: application/json" -H "Authorization: Bearer eyJl...XfnQ"
```

Retorno:

```
{
  "servers": [
    {
      "name": "Servidor DVR",
      "guid": "{182C4DA9-7CA7-416D-9F72-F3527941FA2D}",
      "parentGuid": null,
      "active": true,
      "connected": true
    },
    {
      "name": "Servidor Câmera",
      "guid": "{38A9F3AC-3522-42D2-818B-183004858406}",
      "parentGuid": null,
      "active": true,
      "connected": true
    }
  ]
}
```

Onde:

- O atributo `"guid"` contém a identificação do servidor e trata-se do atributo mais relevante para as demais etapas do processo de exportação de gravações.
- O atributo `"active"` indica se o servidor está ativo / habilitado.
- O atributo `"connected"` indica se o servidor está conectado neste momento. Então, é possível que um servidor esteja ativo e desconectado devido a uma falha de rede ou qualquer outro tipo de problema de transmissão.
- O atributo `"parentGuid"` contém a identificação do servidor raiz ao qual o servidor está vinculado. Neste caso, o servidor raiz se trata de uma outra instância do software D-Guard.

7.3 API REST para obter câmeras de um servidor

A API `"/servers/{guid}/cameras"` permite o gerenciamento das câmeras vinculadas a um servidor de uma instância do software D-Guard.

Neste momento, abordaremos apenas o método [GET] responsável por retornar a lista das câmeras de um servidor previamente cadastrado.

[GET] <http://servidor:porta/api/servers/{guid}/cameras>

Onde:

- O parâmetro "guid" contém a identificação de um servidor previamente cadastrado.

Exemplo de uso com aplicativo cURL

Requisição:

```
curl -X GET
"http://127.0.0.1:8080/api/servers/%7B182C4DA9-7CA7-416D-9F72-F35
27941FA2D%7D/cameras" -H "Accept: application/json" -H
"Authorization: Bearer eyJ1...XfnQ"
```

Retorno:

```
{
  "cameras": [
    {
      "id": 0,
      "name": "Câmera 1"
    },
    {
      "id": 1,
      "name": "Câmera 2"
    },
    {
      "id": 2,
      "name": "Câmera 3"
    },
    {
      "id": 3,
      "name": "Câmera 4"
    }
  ]
}
```

Onde:

- O atributo "id" contém a identificação da câmera e trata-se do atributo mais relevante para as demais etapas do processo de exportação de gravações. É importante ressaltar que a identificação de uma câmera é única para um determinado servidor, mas pode se repetir entre diferentes servidores.

- O atributo "name" contém o nome da câmera e normalmente é gerado automaticamente. Entretanto, pode ser modificado pelo usuário para conter um valor mais significativo.

A exportação de gravações está associada diretamente a uma câmera. Neste caso, além da identificação do servidor, é necessária também a identificação da câmera.

7.4 API REST para iniciar exportação de gravações

A API `"/servers/{guid}/cameras/{id}/recordings/exports"` permite iniciar a exportação de gravações armazenadas pelo D-Guard.

[POST]

<http://servidor:porta/api/servers/{guid}/cameras/{id}/recordings/exports>

```
{
  "start": [string],
  "end": [boolean]
}
```

Onde:

- O parâmetro "guid" contém a identificação de um servidor previamente cadastrado.
- O parâmetro "id" contém a identificação de uma câmera do servidor.
- O atributo "start" contém a data inicial para procura das gravações armazenadas. Este atributo deve ser informado no format "yyyy-MM-ddTHH:mm:ss".
- O atributo "end" contém a data inicial para procura das gravações armazenadas. Este atributo deve ser informado no format "yyyy-MM-ddTHH:mm:ss".

Exemplo de uso com aplicativo cURL

Requisição:

```
curl -X POST
"http://127.0.0.1:8080/api/servers/%7B182C4DA9-7CA7-416D-9F72-F35
27941FA2D%7D/cameras/0/recordings/exports" -H "Accept:
application/json" -H "Authorization: Bearer eyJl...XfnQ" -H
"Content-Type: application/json" -d "{ \"start\":
\"2019-01-01T00:00:00\", \"end\": \"2019-01-01T01:00:00\" }"
```

Retorno:

```
{
  "export": {
    "guid": "{36CCBEC3-C309-4140-9C1B-F30F6FAC586D}",
    "status": 0,
    "progress": 0
  }
}
```

}

Onde:

- O atributo "guid" contém a identificação da requisição de exportação de gravações e trata-se do atributo mais relevante para as demais etapas do processo de exportação de gravações. É importante ressaltar que essa identificação é única e jamais será repetida, mesmo entre diferentes servidores.
- O atributo "status" contém o status inicial de processamento da exportação. Onde o valor 0 (zero) indica "Em progresso" e o valor 1 (um) indica "Finalizada".
- O atributo "progress" contém o percentual de progresso de processamento da exportação.

A requisição de iniciar exportação de gravações retornará uma resposta com erro HTTP/422 caso não existam gravações armazenadas para o período especificado.

7.5 API REST para checar status de processamento

A API "/servers/{guid}/cameras/{id}/recordings/exports/{exportGuid}" permite a checagem do status e o acompanhamento do processamento de uma exportação de gravações previamente iniciada em uma instância do software D-Guard.

[GET]

<http://servidor:porta/api/servers/{guid}/cameras/{id}/recordings/exports/{request}>

Onde:

- O parâmetro "guid" contém a identificação de um servidor previamente cadastrado.
- O parâmetro "id" contém a identificação de uma câmera do servidor.
- O atributo "request" contém a identificação da exportação de gravações previamente iniciada.

Exemplo de uso com aplicativo cURL

Requisição:

```
curl -X GET
"http://127.0.0.1:8080/api/servers/%7B182C4DA9-7CA7-416D-9F72-F35
27941FA2D%7D/cameras/0/recordings/exports/%7B36CCBEC3-C309-4140-9
C1B-F30F6FAC586D%7D" -H "Accept: application/json" -H
"Authorization: Bearer eyJ1...XfnQ"
```

Retorno (em processamento):

```
{
  "export": {
```

```

    "guid": "{36CCBEC3-C309-4140-9C1B-F30F6FAC586D}",
    "status": 0,
    "progress": 50
  }
}

```

Retorno (em finalizada):

```

{
  "export": {
    "guid": "{36CCBEC3-C309-4140-9C1B-F30F6FAC586D}",
    "status": 1,
    "progress": 100
  }
}

```

Onde:

- O atributo "guid" contém a identificação da requisição de exportação de gravações previamente iniciada.
- O atributo "status" contém o status inicial de processamento da exportação. Onde o valor 0 (zero) indica "Em progresso" e o valor 1 (um) indica "Finalizada".
- O atributo "progress" contém o percentual de progresso de processamento da exportação.

A checagem de status do processamento da exportação de gravações precisa estar em algum tipo de *loop* contínuo para que a aplicação cliente possa acompanhar o andamento do processamento. Então, novas requisições HTTP precisam ser enviadas de forma contínua e sequencial para checar o status e, em caso de sucesso, efetuar a requisição do arquivo de vídeo (MP4).

Além disso, a requisição de checagem de status do processamento da exportação de gravações retornará uma resposta com erro HTTP/500 caso ocorra algum erro inesperado no momento do processamento da exportação de gravações.

7.6 API REST para requisitar arquivo de vídeo

A API `"/servers/{guid}/cameras/{id}/recordings/exports/{exportGuid}/file"` permite a recuperação do arquivo de vídeo (MP4) de uma exportação de gravações previamente iniciada e concluída com sucesso uma instância do software D-Guard.

[GET]

<http://servidor:porta/api/servers/{guid}/cameras/{id}/recordings/exports/{request}/file>

Onde:

- O parâmetro "guid" contém a identificação de um servidor previamente cadastrado.
- O parâmetro "id" contém a identificação de uma câmera do servidor.
- O atributo "request" contém a identificação da exportação de gravações previamente iniciada e concluída com sucesso.

Exemplo de uso com aplicativo cURL

Requisição:

```
curl -X GET
"http://127.0.0.1:8080/api/servers/%7B182C4DA9-7CA7-416D-9F72-F35
27941FA2D%7D/cameras/0/recordings/exports/%7B36CCBEC3-C309-4140-9
C1B-F30F6FAC586D%7D/file" -H "Accept: application/json" -H
"Authorization: Bearer eyJ1...XfnQ"
```

Retorno:

```
FF D8 FF E0 00 10 4A 46 49 46 00 01 01 00 00 01 00 01 00 00 FF
...
2A FA 1F 10 9A 6C 9F EA 9B E8 69 C6 98 E7 F7 4D F4 34 01 FF D9
```

O retorno desta API contém conteúdo binário associado ao vídeo em formato MP4 que poderá ser manipulado pela aplicação cliente.

8 Matriz virtual

Esse recurso do D-Guard permite controlar o que é exibido nos monitores das estações de trabalho (Desktop). Através de requisições feitas ao servidor D-Guard, pode-se informar qual *layout* deve estar aberto em cada monitor de cada estação conectada no servidor D-Guard.

Para utilizar essa funcionalidade, é necessário que no D-Guard estação, o usuário configure a estação para participar (inscrever-se) na matriz virtual do servidor. Na tela "Matriz virtual" da estação, deve-se ativar o recurso e configurar quais monitores estarão acessíveis para controle remoto.

8.1 Fluxo para listar estações e monitores

Através do endpoint */virtual-matrix/workstations*, requisitado no servidor D-Guard, pode-se obter a lista de estações participantes dessa funcionalidade, com seus respectivos monitores.

8.2 Fluxo para listar layouts

Através do endpoint */virtual-matrix/layouts*, requisitado no servidor D-Guard, pode-se obter a lista de layouts que estarão disponíveis para envio às estações.

8.3 Fluxo para controlar o layout aberto em uma estação

Através do endpoint abaixo, requisitado no servidor D-Guard, pode-se controlar qual layout deve ser aberto em um determinado monitor de uma estação do D-Guard.

Deve-se informar qual o *guid* da estação, o *guid* do monitor e o *guid* do layout.

```
/virtual-matrix/workstations/{workstationGuid}/monitors/{monitorGuid}/layout
```