



D-Guard

Documentação da interface HTTP/REST
Eventos com ações de requisição HTTP/GET – Versão 1.00

[Histórico de alterações]

Revisão	Data	Autor	Observações
1.0	07.08.2018	John Jaraceski	Versão original
1.1	14.03.2019	John Jaraceski	API HTTP genérica (diferentes métodos)
1.2	14.09.2022	Marcos Paulo Gandin	API para exportação de vídeos do evento

[Conteúdo]

1	Sobre este documento	6
2	Preparação	6
2.1.....	Formato das requisições	6
2.2.....	Formato das respostas	6
3	Conceitos preliminares	7
3.1.....	Servidor, câmera e dispositivo	7
3.2.....	Evento, fonte de eventos e ação	7
4	Monitoramento de eventos	8
4.1.....	Fluxo de configuração de monitoramento	9
4.2.....	API REST para autenticar (login)	10
4.3.....	API REST para obter nome dos eventos	11
4.4.....	API REST para obter períodos	12
4.5.....	API REST para obter servidores	13
4.6.....	API REST para obter câmeras de um servidor	14
4.7.....	API REST para obter fontes de eventos já configuradas	15
4.8.....	API REST para habilitar evento para uma fonte de eventos	16
4.9.....	API REST para obter ações definidas para um evento de uma fonte de eventos	18
4.10....	API REST para definir ação de envio de requisição HTTP/GET para um evento de uma fonte de eventos.....	20
4.11.....	API REST para configuração da ação de exportação de vídeos de eventos	22
5	Exemplos de uso da API REST	24

5.1.....	Obter a lista de nomes de servidores	24
5.2.....	Obter a lista de câmeras dos servidores	25
5.3.....	Monitorar evento de desconexão de um servidor	26
5.4.....	Verificar existência de monitoramento do evento de desconexão de um servidor	28
5.5.....	Monitorar evento de perda de sinal de uma câmera	31
5.6.....	Verificar existência de monitoramento do evento de perda de sinal de uma câmera	34

1 Sobre este documento

Este documento fornece informações sobre a interface HTTP/REST dos softwares D-Guard (versão 6 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 o registro de eventos, tais como desconexão de servidor ou perda de sinal de câmera, é preciso entender alguns conceitos preliminares sobre as principais entidades envolvidas neste processo: servidor, câmera e dispositivo; evento, fonte de eventos e ação.

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 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. 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 Evento, fonte de eventos e ação

Os softwares D-Guard permitem o monitoramento de inúmeros eventos, tais como : conexão e desconexão de servidores, restauração e perda de sinal de câmeras, abertura e fechamento de layouts, entre outros. O monitoramento de todos estes diferentes tipos de eventos pode ser configurado na tela "Configuração de Eventos".

A configuração do monitoramento de um evento é feita em três etapas:

1. Seleção do evento a ser monitorado
2. Seleção da fonte disparadora do evento
3. Definição da ação a ser realizada quando o evento selecionado for disparado pela fonte selecionada

Nesse processo de configuração é importante entender a diferença entre os conceitos “evento” e “fonte de eventos”. Eventos são fixos e pré-determinados dentro do software D-Guard, enquanto fontes de evento se referem aos dispositivos capazes de disparar um evento específico.

Dessa forma, um servidor ou uma câmera ou um layout são fontes de eventos capazes de disparar eventos específicos. Por exemplo, um servidor é capaz de disparar os eventos de conexão e desconexão, enquanto que uma câmera é capaz de disparar os eventos de perda e restauração de sinal, detecção e fim de detecção de movimento, entre outros.

Finalmente, a ação é aquilo que se deseja executar quando uma fonte de eventos disparar um evento específico. Sendo possível configurar múltiplas e variadas ações, tais como a reprodução de um sinal sonoro, o controle de um relé, o envio de um e-mail, a ativação de uma gravação de vídeo para uma câmera específica, o envio de uma requisição HTTP/GET, entre outras.

4 Monitoramento de eventos

O processo de monitoramento de eventos inicia pela configuração do monitoramento dos eventos desejados e termina pela verificação da execução das ações associadas.

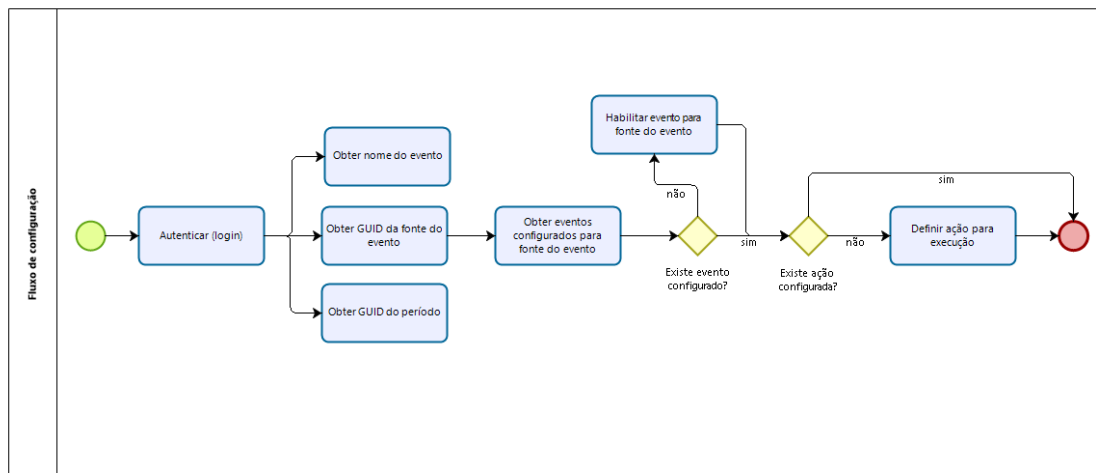
A etapa de configuração do monitoramento de eventos pode ser realizada na tela “Configuração de Eventos” ou através do uso da API REST. A configuração feita pela tela do software D-Guard é bastante simples e intuitiva, enquanto que o uso da API REST permite a automatização deste processo.

A verificação da execução das ações permite validar a etapa anterior e entender plenamente o funcionamento do motor de eventos do software. O tipo e a complexidade da verificação a ser realizada para validar a execução da ação pode variar bastante conforme o tipo de ação definida.

Algumas ações podem ser verificadas diretamente no D-Guard, como a abertura de um layout ou o registro de uma gravação. Enquanto outras podem necessitar de uma verificação mais complexa e podem envolver a análise de arquivos de log ou o acesso a outros sistemas, como o envio de requisições HTTP/GET.

4.1 Fluxo de configuração de monitoramento

A imagem abaixo apresenta o fluxo geral de configuração de monitoramento de eventos para uma fonte de eventos.



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 propriamente dita a configuração do monitoramento de eventos.

Onde:

- “Obter nome do evento” se refere a obtenção do nome do evento que se deseja monitorar. Existe uma lista fixa de eventos que podem ser monitorados, então pode-se optar por manter uma lista de constantes com os nomes dos eventos.
- “Obter GUID da fonte do evento” se refere a obtenção do GUID associado ao servidor ou à câmera ou à qualquer outro tipo de entidade que se deseja monitorar o disparo de um evento.
- “Obter GUID do período” se refere a obtenção do GUID do período (horário de início e fim) de monitoramento de eventos. O período padrão compreende o horário de 00:00:00 às 23:59:59, mas outros períodos podem ser cadastrados com faixas de horários distintas.

A etapa seguinte, “Obter eventos configurados para fonte do evento”, se refere a obtenção da lista de eventos previamente configurados para a fonte do evento. Esta etapa precisa ser realizada

porque a API REST não permite o cadastro de um mesmo evento múltiplas vezes para uma mesma fonte de eventos.

Embora um evento específico somente possa ser cadastrado uma única vez para uma mesma fonte de eventos, é possível configurar múltiplas ações para ser executadas quando o evento for disparado. Por exemplo, o evento de desconexão de um servidor específico pode ser configurado para executar as ações de envio de e-mail, reprodução de sinal sonoro e envio de requisição HTTP/GET.

Caso nenhum evento esteja configurado para a fonte de eventos, então deve-se realizar a etapa “Habilitar evento para fonte de eventos”. Esta etapa se refere ao processo de cadastrar ou habilitar o monitoramento de um determinado evento para uma fonte de eventos específica. Por exemplo, é nesta etapa que o evento habilitamos o monitoramento do evento de desconexão para um determinado servidor, ou o evento de perda de sinal para uma determinada câmera de um servidor.

Com o evento já configurado para a fonte de eventos, deve-se fazer a configuração da ação (ou ações) que se deseja executar quando ocorrer o disparo do evento pela fonte de eventos. Da mesma forma como um mesmo evento não pode ser cadastrado múltiplas vezes para uma mesma fonte de eventos, uma mesma ação não pode ser cadastrada múltiplas vezes para um mesmo evento de uma fonte de eventos.

Dessa forma, por exemplo, ação de envio de requisição HTTP/GET somente pode ser cadastrada uma única vez para o evento de desconexão de um determinado servidor. Sendo assim, não é possível cadastrar dois ou mais envios de requisição HTTP/GET para o evento de desconexão de um determinado servidor.

4.2 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"
  }
}
```

4.3 API REST para obter nome dos eventos

A API "/event-actions/system-events" retorna a lista de todos os eventos monitorados por uma instância do software D-Guard.

[GET] <http://servidor:porta/api/event-actions/system-events>

Exemplo de uso com aplicativo cURL

Requisição:

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

Retorno:

```
{
  "systemEvents": [
    {
      "name": "connection",
      "type": 1,
      "allowsAny": true
    },
    {
      "name": "disconnection",
      "type": 1,
      "allowsAny": true
    },
    {
      "name": "signal-restore",
      "type": 3,
      "allowsAny": true
    }
  ]
}
```

```

        "name": "signal-lost",
        "type": 3,
        "allowsAny": true
    }
}
}

```

Onde:

- O atributo "name" contém o nome do evento e trata-se do atributo mais relevante para as demais etapas do processo de configuração de monitoramento de eventos.
- O atributo "type" indica o tipo de entidade associada ao evento, podendo ter os valores: 1 (servidores), 2 (câmeras), 3 (layouts), 4 (ios), 5 (falhas de gravação) e 6 (eventos personalizados).
- O atributo "allowAny" indica se o evento pode ser cadastrado para monitorar todas as entidades (servidor, câmera, layout, entre outras) de uma só vez sem a necessidade de ser cadastrado individualmente para cada uma das entidades.

4.4 API REST para obter períodos

A API "/event-actions/periods" permite o gerenciamento dos períodos de monitoramento de eventos em uma instância do software D-Guard.

Neste momento, abordaremos apenas o método [GET] responsável por retornar a lista dos períodos de monitoramento de eventos previamente cadastrados.

[GET] <http://servidor:porta/api/event-actions/periods>

Exemplo de uso com aplicativo cURL

Requisição:

```

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

```

Retorno:

```

{
  "periods": [
    {
      "guid": "{1A9EAB6A-531D-43BE-BF3F-A06693F71AB0}",
      "name": "Período 1",
      "startTime": "00:00:00",
      "endTime": "23:59:59",
      "daysOfWeek": [1, 2, 3, 4, 5, 6, 7]
    }
  ]
}

```

Onde:

- O atributo "guid" contém a identificação do período e trata-se do atributo mais relevante para as demais etapas do processo de configuração de monitoramento de eventos.
- O atributo "daysOfWeek" indica os dias da semana aos quais o período se aplica, podendo ter os valores: 1 (domingo), 2 (segunda-feira), 3 (terça-feira), 4 (quarta-feira), 5 (quinta-feira), 6 (sexta-feira) e 7 (sábado).

4.5 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 configuração de monitoramento de eventos.

- 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.

4.6 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-F3527941FA2D%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 configuração de monitoramento de eventos. 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 configuração do monitoramento de alguns eventos está associada com fontes de evento do tipo câmera. Nestes casos, além da identificação do servidor, é necessária também a identificação da câmera. Por exemplo, a configuração do monitoramento dos eventos de perda de sinal e detecção de movimento necessita das identificações do servidor e da câmera para ser efetuada.

4.7 API REST para obter fontes de eventos já configuradas

A API "/event-actions/sources" permite o gerenciamento da configuração de eventos para uma fonte de eventos de uma instância do software D-Guard.

Apenas para lembrar, uma fonte de eventos pode ser um servidor ou uma câmera ou qualquer outro tipo de entidade capaz de disparar eventos no software D-Guard.

Neste momento, abordaremos o método [GET] responsável por retornar a lista das fontes de eventos previamente configuradas.

[GET] <http://servidor:porta/api/event-actions/sources>

Exemplo de uso com aplicativo cURL

Requisição:

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

Retorno:

```
{
  "sources": [
    {
      "sourceGuid": "{92BFF008-9170-4501-8003-649A61D93CB8}",
      "periodGuid": "{1A9EAB6A-531D-43BE-BF3F-A06693F71AB0}"
    }
  ]
}
```

```

        "periodName": "Período 1",
        "eventName": "motion-detection",
        "eventKind": 2,
        "serverGuid": "{182C4DA9-7CA7-416D-9F72-F3527941FA2D}",
        "serverName": "Servidor DVR",
        "cameraId": 0,
        "cameraName": "Câmera 01"
    }
}
}

```

Onde:

- O atributo "sourceGuid" contém a identificação da configuração de monitoramento de um determinado evento para uma entidade específica e trata-se de um dos atributo mais relevante para as demais etapas do processo de configuração de monitoramento de eventos.
- O atributo "eventName" contém a identificação do evento monitorado.
- O atributo "eventKind" indica o tipo do evento ou da entidade associada ao evento, podendo ter os valores: 1 (servidores), 2 (câmeras), 3 (layouts), 4 (ios), 5 (falhas de gravação) e 6 (eventos personalizados).
- O atributo "serverGuid" contém a identificação do servidor monitorado e fonte do evento disparado. Então, o tipo do evento deve ser 1 (servidores), 2 (câmeras), 4 (ios) ou 5 (falhas gravação).
- O atributo "cameraId" contém a identificação da câmera monitorada e fonte do evento disparado. Então, o tipo do evento deve ser 2 (câmeras) ou 5 (falhas gravação).

O retorno desta API precisa ser analisado para identificar se um determinado evento precisa ou não ser habilitado para uma entidade específica.

Considerando o retorno apresentado no exemplo anterior, apenas não é possível habilitar o evento de detecção de movimento para a câmera "Câmera 01" do servidor "Servidor DVR". Entretanto, é possível habilitar este mesmo evento para outra câmera deste mesmo servidor ou de outros servidores. Bem como, é possível habilitar inúmeros outros eventos, tais como desconexão de servidor ou perda de sinal ou abertura de layout.

4.8 API REST para habilitar evento para uma fonte de eventos

A API "/event-actions/sources" permite o gerenciamento da configuração de eventos para uma fonte de eventos de uma instância do software D-Guard.

Apenas para lembrar, uma fonte de eventos pode ser um servidor ou uma câmera ou qualquer outro tipo de entidade capaz de disparar eventos no software D-Guard.

Neste momento, abordaremos o método [POST] responsável por habilitar o monitoramento de um determinado evento para uma entidade específica.

[POST] <http://servidor:porta/api/event-actions/sources>

```
{
  "periodGuid": [string],
  "eventName": [string],
  "serverGuid": [string],
  "cameraId": [int],
  "waitingTime": [int]
}
```

Onde:

- O atributo "periodGuid" contém a identificação do período de monitoramento do evento.
- O atributo "eventName" contém o nome do evento que se deseja monitorar.
- O atributo "serverGuid" contém a identificação do servidor responsável pelo disparado do evento que se deseja monitorar. Este atributo deve ser informado apenas se o tipo do evento for 1 (servidores), 2 (câmeras), 4 (ios) ou 5 (falhas de gravação).
- O atributo "cameraId" contém a identificação da câmera responsável pelo disparado do evento que se deseja monitorar. Este atributo deve ser informado apenas se o tipo do evento for 2 (câmeras), ou 5 (falhas de gravação).
- (opcional) O atributo "waitingTime" contém o intervalo de tempo de espera para disparo do evento. Este atributo é utilizado apenas para os eventos de desconexão de servidor e perda de sinal para evitar a geração de falsos-positivos.

Exemplo de uso com aplicativo cURL

Requisição:

```
curl -X POST "http://127.0.0.1:8080/api/event-actions/sources"
-H "Accept: application/json" -H "Content-Type:
application/json" -H "Authorization: Bearer eyJ1...XfnQ" -d
"{\"periodGuid\": \"1A9EAB6A-531D-43BE-BF3F-A06693F71AB0\",
\"eventName\": \"disconnection\",
\"serverGuid\": \"182C4DA9-7CA7-416D-9F72-F3527941FA2D\",
\"waitingTime\": 5}"
```

Retorno:

```
{
  "source": [
```

```

{
  "sourceGuid": "{7D7C9C46-67E7-4263-B714-2E5D1C0E74EE}",
  "periodGuid": "{1A9EAB6A-531D-43BE-BF3F-A06693F71AB0}",
  "periodName": "Período 1",
  "eventName": "disconnection",
  "eventKind": 1,
  "serverGuid": "{182C4DA9-7CA7-416D-9F72-F3527941FA2D}",
  "serverName": "Servidor DVR",
  "waitingTime": 5
}
]
}

```

Onde:

- O atributo "sourceGuid" contém a identificação da configuração de monitoramento de um determinado evento para uma entidade específica e trata-se de um dos atributo mais relevante para as demais etapas do processo de configuração de monitoramento de eventos.
- O atributo "eventName" contém a identificação do evento monitorado.
- O atributo "eventKind" indica o tipo do evento ou da entidade associada ao evento, podendo ter os valores: 1 (servidores), 2 (câmeras), 3 (layouts), 4 (ios), 5 (falhas de gravação) e 6 (eventos personalizados).
- O atributo "serverGuid" contém a identificação do servidor monitorado e fonte do evento disparado. Então, o tipo do evento deve ser 1 (servidores), 2 (câmeras), 4 (ios) ou 5 (falhas gravação).
- O atributo "waitingTime" contém o intervalo de tempo de espera para disparo do evento. Este atributo é utilizado apenas para os eventos de desconexão de servidor e perda de sinal para evitar a geração de falsos-positivos.

O retorno desta API contém os dados do monitoramento recém configurado. O atributo "sourceGuid" é essencial para a etapa de definição da ação (ou ações) a ser executada quando ocorrer o disparo do evento pela fonte de eventos. Neste exemplo, quando o servidor "Servidor DVR" disparar o evento de desconexão.

4.9 API REST para obter ações definidas para um evento de uma fonte de eventos

A API "/event-actions/sources/{guid}/actions" permite a recuperação das ações definidas para execução quando ocorrer o disparo de um determinado evento para uma fonte de eventos específica de uma instância do software D-Guard.

Apenas para lembrar, uma fonte de eventos pode ser um servidor ou uma câmera ou qualquer outro tipo de entidade capaz de disparar eventos no software D-Guard.

[GET] <http://servidor:porta/api/event-actions/sources/{guid}/actions>

Onde:

- O parâmetro "guid" contém a identificação da configuração de monitoramento de evento de um determinado evento para uma fonte de eventos específica.

Exemplo de uso com aplicativo cURL

Requisição:

```
curl -X GET "http://127.0.0.1:8080/api/event-actions/sources/%7B7D7C9C46-67E7-4263-B714-2E5D1C0E74EE%7D" -H "Accept: application/json" -H "Authorization: Bearer eyJ1...XfnQ"
```

Retorno:

```
{
  "actions": {
    "playSound": false,
    "relayControl": false,
    "sendEmail": false,
    "cameraSnapshot": false,
    "openLayout": false,
    "dynamicLayout": false,
    "activatePreset": false,
    "controlRecording": false,
    "switchPreviewStream": false,
    "switchRecordingStream": false,
    "changeProfile": false,
    "activateLPR": false,
    "controlAudio": false,
    "sendSMS": false,
    "monitorSnapshot": false,
    "sendHttpRequest": false,
    "sneepCall": false,
    "contactId": false,
    "situator": false,
    "recordBookmark": false
  }
}
```

Onde:

- Cada atributo identifica se uma determinada ação está ou não definida para ser executada quando ocorrer o disparo de um determinado evento para uma fonte de eventos específica.

O retorno desta API tem caráter apenas informativo porque a definição propriamente dita da ação é realizada com APIs específicas para cada tipo de ação.

4.10 API REST para definir ação de envio de requisição HTTP/GET para um evento de uma fonte de eventos

A API “/event-actions/sources/{guid}/actions/ send-http-request” permite a definição da ação de envio de requisição HTTP/GET quando ocorrer o disparo de um determinado evento para uma fonte de eventos específica de uma instância do software D-Guard.

Apenas para lembrar, uma fonte de eventos pode ser um servidor ou uma câmera ou qualquer outro tipo de entidade capaz de disparar eventos no software D-Guard.

[PUT] <http://servidor:porta/api/event-actions/sources/{guid}/actions/send-http-requests>

```
{
  "enabled": [boolean],
  "method": [integer],
  "url": [string],
  "authType": [integer],
  "user": [string],
  "password": [string],
  "headers": [array],
  [
    {
      "name": [string],
      "value": [string]
    }
  ]
  "body": [string]
}
```

Onde:

- O parâmetro “guid” contém a identificação da configuração de monitoramento de evento de um determinado evento para uma fonte de eventos específica.
- O atributo “enabled” indica se a ação está ativa / habilitada.
- O atributo “method” indica o tipo do método associado a requisição a ser enviada, podendo ter os valores: 0 (GET), 1 (PATCH), 2 (POST), 3 (PUT), e 4 (DELETE).
- O atributo “url” contém a *url* da requisição a ser enviada por HTTP/GET.
 - O valor definido para *url* pode fazer uso de variáveis internas que serão substituídas automaticamente no momento da requisição. Algumas das variáveis mais utilizadas são:
 - \$event.name\$: Nome do evento

- \$event.data\$: Data do evento
 - \$event.time\$: Hora do evento
 - \$server.name\$: Nome do servidor
 - \$camera.name\$: Identificação da câmera
- O atributo "authType" indica o tipo da autenticação associado a requisição a ser enviada, podendo ter os valores: 0 (none), 1 (basic), 2 (digest), e 3 (auto: basic ou digest).
- (opcional) O atributo "user" contém o nome do usuário a ser enviado no cabeçalho da requisição. A autenticação de usuário segue os formatos "Authorization: Basic" e/ou "Authorization: Digest" do HTTP. Caso necessário, utilize a documentação do protocolo HTTP (rfc 2617) para maiores informações sobre esses métodos de autenticação.
- (opcional) O atributo "password" contém a senha do usuário em texto claro a ser enviada no cabeçalho da requisição. A autenticação de usuário segue os formatos "Authorization: Basic" e/ou "Authorization: Digest" do HTTP. Caso necessário, utilize a documentação do protocolo HTTP (rfc 2617) para maiores informações sobre esses métodos de autenticação.
- (opcional) O atributo "headers" contém a lista de campos de cabeçalho a serem enviados na requisição.
- (opcional) O atributo "body" contém o corpo a ser enviado na requisição quando esta fizer uso dos métodos PATCH, POST, ou PUT.

Exemplo de uso com aplicativo cURL

Requisição:

```
curl -X PUT "http://127.0.0.1:8080/api/event-
actions/sources/%7B7D7C9C46-67E7-4263-B714-
2E5D1C0E74EE%7D/actions/send-http-request" -H "Accept:
application/json" -H "Content-Type: application/json" -H
"Authorization: Bearer eyJ1...XfnQ" -d "{\"enabled\": false,
\"url\": \"http://www.google.com?q=$server.name$\"}"
```

Retorno:

```
{
  "sendHttpRequest": {
    "enabled": true,
    "method": 0,
    "url": "http://www.google.com?q=$server.name$",
    "user": "",
    "password": ""
  }
}
```

Onde:

- O atributo "enabled" indica se a ação está ativa / habilitada.

- O atributo "method" contém o tipo do método associado a requisição a ser enviada.
- O atributo "url" contém a *url* da requisição a ser enviada por HTTP/GET.
- (opcional) O atributo "user" contém o nome do usuário a ser enviado no cabeçalho da requisição.
- (opcional) O atributo "password" contém a senha do usuário em texto claro a ser enviada no cabeçalho da requisição.

4.11 API REST para configuração da ação de exportação de vídeos de eventos

A API `/event-actions/sources/{sourceGuid}/actions/event-recording` permite a configuração da ação de exportação de vídeos de eventos para um determinado local (pasta no computador) quando ocorrer uma gravação de vídeo.

Apenas para lembrar, uma fonte de eventos pode ser um servidor ou uma câmera ou qualquer outro tipo de entidade capaz de disparar eventos no software D-Guard.

[PUT] <http://servidor:porta/api/event-actions/sources/{sourceGuid}/actions/event-recording>

```
{
  "enabled": [boolean],
  "serverGuid": [string],
  "cameraId": [integer],
  "streamId": [integer],
  "preDelayInSeconds": [integer],
  "postDelayInSeconds": [integer],
  "outputFilePath": [string],
  "sourceType": [integer]
}
```

Onde:

- O parâmetro "sourceGuid" contém a identificação da configuração de monitoramento de evento de um determinado evento para uma fonte de eventos específica.
- O atributo "enabled" indica se a ação está ativa / habilitada.
- O atributo "serverGuid" contém a identificação do servidor responsável pelo disparado do evento que se deseja exportar o vídeo.
- O atributo "cameraId" contém a identificação da câmera responsável pelo disparado do evento que se deseja exportar o vídeo.

- O atributo "streamId" contém a identificação do stream da câmera responsável pelo disparado do evento que se deseja exportar o vídeo.
- O atributo "preDelayInSeconds" contém o tempo em segundos antes do início de um evento eu quero ter um arquivo de gravação salvo.
- O atributo "postDelayInSeconds" contém o tempo em segundos depois do término de um evento eu quero ter um arquivo de gravação salvo.
- O atributo "outputFilePath" contém o endereço do local que serão gravados os vídeos exportados.
- O atributo "sourceType" contém a tipo de gravação sendo:
 - ⌘ 0 - Local
 - ⌘ 1 - Em Borda

Exemplo de uso com aplicativo cURL

Requisição:

```
curl -X PUT "http://localhost:8081/api/event-
actions/sources/2D1BC424-3826-4322-9182-
78EBC1165786/actions/event-recording" -H "accept:
application/json" -H "Authorization: bearer
eyJ1c2VyTmFtZSI6ImFkbWluIn0.a6Fy-vKXhGPLmjwpO1MJ2z--
oXXmhWSpQ3rPoKf6HS4" -H "Content-Type: application/json" -d
"{ \"EventRecording\": { \"enabled\": true, \"serverGuid\":
\"{FCF294E9-D5B5-4C38-9310-9AAD2678AF23}\" , \"cameraId\": 0,
\"streamId\": 0, \"preDelayInSeconds\": 5,
\"postDelayInSeconds\": 5, \"outputFilePath\":
\"C:\\\\Seventh\\\\DGuardCenter\\\\Dados\\\\EventVideoExport\\\\
\", \"sourceType\": 0 } }"
```

Retorno:

```
{
  "EventRecording": {
    "enabled": true,
    "serverGuid": "{FCF294E9-D5B5-4C38-9310-9AAD2678AF23}",
    "cameraId": 0,
    "streamId": 0,
    "preDelayInSeconds": 5,
    "postDelayInSeconds": 5,
    "outputFilePath":
"C:\\Seventh\\DGuardCenter\\Dados\\EventVideoExport\\",
    "sourceType": 0
  }
}
```

Onde:

- O parâmetro "sourceGuid" contém a identificação da configuração de monitoramento de evento de um determinado evento para uma fonte de eventos específica.
- O atributo "enabled" indica se a ação está ativa / habilitada.

- O atributo "serverGuid" contém a identificação do servidor responsável pelo disparado do evento que se deseja exportar o vídeo.
- O atributo "cameraId" contém a identificação da câmera responsável pelo disparado do evento que se deseja exportar o vídeo.
- O atributo "streamId" contém a identificação do stream da câmera responsável pelo disparado do evento que se deseja exportar o vídeo.
- O atributo "preDelayInSeconds" contém o tempo em segundos antes do início de um evento eu quero ter um arquivo de gravação salvo.
- O atributo "postDelayInSeconds" contém o tempo em segundos depois do término de um evento eu quero ter um arquivo de gravação salvo.
- O atributo "outputFilePath" contém o endereço do local que serão gravados os vídeos exportados.
- O atributo "sourceType" contém a tipo de gravação.

5 Exemplos de uso da API REST

Todos os exemplos apresentados fazem uso do aplicativo cURL para efetuar as requisições à API REST do software D-Guard.

5.1 Obter a lista de nomes de servidores

Para obter a lista de nomes de servidores cadastrados em uma instância do software D-Guard é necessário realizar os seguintes passos:

1. Efetuar requisição de autenticação

```
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\"}"
```

2. Extrair *token* de autenticação do retorno da requisição anterior

```
{
  "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"
  }
}
```

3. Adicionar *token* de autenticação como cabeçalho às demais requisições

```
-H "Authorization: Bearer eyJ1...XfnQ"
```


4. Efetuar requisição de obtenção da lista de servidores

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

5.2 Obter a lista de câmeras dos servidores

Para obter a lista de câmeras de cada um dos servidores cadastrados em uma instância do software D-Guard é necessário realizar os seguintes passos:

1. Efetuar requisição de autenticação

```
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\"}"
```

2. Extrair *token* de autenticação do retorno da requisição anterior

```
{
  "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"
  }
}
```

3. Adicionar *token* de autenticação ao cabeçalho das demais requisições

```
-H "Authorization: Bearer eyJ1...XfnQ"
```

4. Efetuar requisição de obtenção da lista de servidores

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

5. Extrair identificação de cada servidor do retorno da requisição anterior

```
{
  "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
    }
  ]
}
```

```
]
}
```

6. Efetuar requisições para cada um dos servidores para obtenção da lista de câmeras

```
curl -X GET "http://127.0.0.1:8080/api/servers/%7B182C4DA9-7CA7-416D-9F72-F3527941FA2D%7D" -H "Accept: application/json" -H "Authorization: Bearer eyJl...XfnQ"
```

```
curl -X GET "http://127.0.0.1:8080/api/servers/%7B38A9F3AC-3522-42D2-818B-183004858406%7D" -H "Accept: application/json" -H "Authorization: Bearer eyJl...XfnQ"
```

5.3 Monitorar evento de desconexão de um servidor

Para ativar o monitoramento do evento de desconexão de uma fonte de eventos do tipo servidor e definir a ação de envio de requisição HTTP/GET em uma instância do software D-Guard é necessário realizar os seguintes passos:

1. Efetuar requisição de autenticação

```
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\"}"
```

2. Extrair *token* de autenticação do retorno da requisição anterior

```
{
  "login": {
    "username": "admin",
    "userGuid": "{4FC0C44C-2789-4902-B322-A2927B14E6A8}",
    "userGroupGuid": "{491693D5-3F25-493E-9234-DAE05A7D8458}",
    "userToken": "eyJl...XfnQ",
    "serverDate": "2018-08-07",
    "serverTime": "14:00:00"
  }
}
```

3. Adicionar *token* de autenticação ao cabeçalho das demais requisições

```
-H "Authorization: Bearer eyJl...XfnQ"
```

4. Efetuar requisição de obtenção da lista de servidores

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

5. Extrair identificação do servidor desejado do retorno da requisição anterior

```
{
  "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
    }
]
}

```

6. Efetuar requisição de obtenção da lista de períodos de monitoramento

```

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

```

7. Extrair identificação do período de monitoramento desejado do retorno da requisição anterior

```

{
  "periods": [
    {
      "guid": "{1A9EAB6A-531D-43BE-BF3F-A06693F71AB0}",
      "name": "Período 1",
      "startTime": "00:00:00",
      "endTime": "23:59:59",
      "daysOfWeek": [1, 2, 3, 4, 5, 6, 7]
    }
  ]
}

```

8. Montar conteúdo da requisição de habilitação do monitoramento de desconexão da fonte de dados desejada

```

{
  "periodGuid": "{1A9EAB6A-531D-43BE-BF3F-A06693F71AB0}",
  "eventName": "disconnection",
  "serverGuid": "{182C4DA9-7CA7-416D-9F72-F3527941FA2D}",
  "waitingTime": 5
}

```

9. Efetuar requisição de habilitação do monitoramento do evento de desconexão da fonte de dados desejada

```

curl -X POST "http://127.0.0.1:8080/api/event-actions/sources"
-H "Accept: application/json" -H "Content-Type:
application/json" -H "Authorization: Bearer eyJl...XfnQ" -d
"{\"periodGuid\": \"{1A9EAB6A-531D-43BE-BF3F-A06693F71AB0}\",
\"eventName\": \"disconnection\",
\"serverGuid\": \"{182C4DA9-7CA7-416D-9F72-F3527941FA2D}\",
\"waitingTime\": 5}"

```

10. Extrair identificação da configuração de monitoramento do retorno da requisição anterior

```

{
  "source": [
    {
      "sourceGuid": "{7D7C9C46-67E7-4263-B714-2E5D1C0E74EE}",

```

```

        "periodGuid": "{1A9EAB6A-531D-43BE-BF3F-A06693F71AB0}",
        "periodName": "Período 1",
        "eventName": "disconnection",
        "eventKind": 1,
        "serverGuid": "{182C4DA9-7CA7-416D-9F72-F3527941FA2D}",
        "serverName": "Servidor DVR",
        "waitingTime": 5
    }
}
]
}

```

11. Montar conteúdo da requisição de definição da ação a ser executada para o monitoramento recém criado

```

{
  "enabled": false,
  "url": "http://www.google.com?q=$server.name$"
}

```

12. Efetuar requisição de definição da ação a ser executada para o monitoramento recém criado

```

curl -X PUT "http://127.0.0.1:8080/api/event-
actions/sources/%7B7D7C9C46-67E7-4263-B714-
2E5D1C0E74EE%7D/actions/send-http-request" -H "Accept:
application/json" -H "Content-Type: application/json" -H
"Authorization: Bearer eyJ1...XfnQ" -d "{\"enabled\": false,
\"url\": \"http://www.google.com?q=$server.name$\"}"

```

5.4 Verificar existência de monitoramento do evento de desconexão de um servidor

Para verificar a existência de monitoramento do evento de desconexão de uma fonte de eventos do tipo servidor em uma instância do software D-Guard é necessário realizar os seguintes passos:

1. Efetuar requisição de autenticação

```

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\"}"

```

2. Extrair *token* de autenticação do retorno da requisição anterior

```

{
  "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"
  }
}

```

3. Adicionar *token* de autenticação ao cabeçalho das demais requisições

```
-H "Authorization: Bearer eyJl...XfnQ"
```

4. Efetuar requisição de obtenção da lista de servidores

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

5. Extrair identificação do servidor desejado do retorno da requisição anterior

```
{
  "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
    }
  ]
}
```

6. Efetuar requisição de obtenção da lista de fontes de eventos já configuradas

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

7. Verificar se o evento de desconexão da fonte de eventos desejada está presente no retorno da requisição anterior

```
{
  "sources": [
    {
      "sourceGuid": "{92BFF008-9170-4501-8003-649A61D93CB8}",
      "periodGuid": "{1A9EAB6A-531D-43BE-BF3F-A06693F71AB0}",
      "periodName": "Período 1",
      "eventName": "motion-detection",
      "eventKind": 2,
      "serverGuid": "{182C4DA9-7CA7-416D-9F72-F3527941FA2D}",
      "serverName": "Servidor DVR",
      "cameraId": 0,
      "cameraName": "Câmera 01"
    },
    {
      "sourceGuid": "{7D7C9C46-67E7-4263-B714-2E5D1C0E74EE}",
      "periodGuid": "{1A9EAB6A-531D-43BE-BF3F-A06693F71AB0}",
      "periodName": "Período 1",
      "eventName": "disconnection",
      "eventKind": 1,
      "serverGuid": "{182C4DA9-7CA7-416D-9F72-F3527941FA2D}",
      "serverName": "Servidor DVR"
    }
  ]
}
```

```
}
```

8. Extrair identificação da configuração de monitoramento desejada do retorno da requisição anterior

```
{
  "sources": [
    {
      "sourceGuid": "{92BFF008-9170-4501-8003-649A61D93CB8}",
      "periodGuid": "{1A9EAB6A-531D-43BE-BF3F-A06693F71AB0}",
      "periodName": "Período 1",
      "eventName": "motion-detection",
      "eventKind": 2,
      "serverGuid": "{182C4DA9-7CA7-416D-9F72-F3527941FA2D}",
      "serverName": "Servidor DVR",
      "cameraId": 0,
      "cameraName": "Câmera 01"
    },
    {
      "sourceGuid": "{7D7C9C46-67E7-4263-B714-2E5D1C0E74EE}",
      "periodGuid": "{1A9EAB6A-531D-43BE-BF3F-A06693F71AB0}",
      "periodName": "Período 1",
      "eventName": "disconnection",
      "eventKind": 1,
      "serverGuid": "{182C4DA9-7CA7-416D-9F72-F3527941FA2D}",
      "serverName": "Servidor DVR",
      "waitingTime": 5
    }
  ]
}
```

9. Efetuar requisição de obtenção da lista de ações do evento de desconexão da fonte de eventos desejada

```
curl -X GET "http://127.0.0.1:8080/api/event-
actions/sources/%7B7D7C9C46-67E7-4263-B714-
2E5D1C0E74EE%7D/actions" -H "Accept: application/json" -H
"Content-Type: application/json" -H "Authorization: Bearer
eyJ1...XfnQ"
```

10. Verificar se a ação de envio de requisição HTTP/GET para o evento de desconexão da fonte de eventos desejada está presente no retorno da requisição anterior

```
{
  "actions": {
    "playSound": false,
    "relayControl": false,
    "sendEmail": false,
    "cameraSnapshot": false,
    "openLayout": false,
    "dynamicLayout": false,
    "activatePreset": false,
    "controlRecording": false,
    "switchPreviewStream": false,
    "switchRecordingStream": false,
    "changeProfile": false,
    "activateLPR": false,
    "controlAudio": false,
    "sendSMS": false,
    "monitorSnapshot": false,

```

```

        "sendHttpRequest": true,
        "sneepCall": false,
        "contactId": false,
        "situator": false,
        "recordBookmark": false
    }
}

```

11. Efetuar requisição de obtenção dos detalhes da ação de envio de requisição HTTP/GET lista de ações da configuração de monitoramento desejada

```

curl -X GET "http://127.0.0.1:8080/api/event-
actions/sources/%7B7D7C9C46-67E7-4263-B714-
2E5D1C0E74EE%7D/actions/send-http-request" -H "Accept:
application/json" -H "Content-Type: application/json" -H
"Authorization: Bearer eyJ1...XfnQ"

```

5.5 Monitorar evento de perda de sinal de uma câmera

Para ativar o monitoramento do evento de perda de sinal de uma fonte de eventos do tipo câmera e definir a ação de envio de requisição HTTP/GET em uma instância do software D-Guard é necessário realizar os seguintes passos:

1. Efetuar requisição de autenticação

```

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\"}"

```

2. Extrair *token* de autenticação do retorno da requisição anterior

```

{
  "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"
  }
}

```

3. Adicionar *token* de autenticação ao cabeçalho das demais requisições

```

-H "Authorization: Bearer eyJ1...XfnQ"

```

4. Efetuar requisição de obtenção da lista de servidores

```

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

```

5. Extrair identificação do servidor desejado do retorno da requisição anterior

```

{
  "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
}
]
}

```

6. Efetuar requisição de obtenção da lista de câmeras do servidor desejado

```

curl -X GET "http://127.0.0.1:8080/api/servers/%7B182C4DA9-7CA7-416D-9F72-F3527941FA2D%7D/cameras" -H "Accept: application/json" -H "Authorization: Bearer eyJ1...XfnQ"

```

7. Extrair identificação da câmera desejada do retorno da requisição anterior

```

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

```

8. Efetuar requisição de obtenção da lista de períodos de monitoramento

```

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

```

9. Extrair identificação do período de monitoramento desejado do retorno da requisição anterior

```

{
  "periods": [
    {
      "guid": "{1A9EAB6A-531D-43BE-BF3F-A06693F71AB0}",

```



```

        "name": "Período 1",
        "startTime": "00:00:00",
        "endTime": "23:59:59",
        "daysOfWeek": [1, 2, 3, 4, 5, 6, 7]
    }
]
}

```

10. Montar conteúdo da requisição de habilitação do monitoramento de perda de sinal da fonte de dados desejada

```

{
  "periodGuid": "{1A9EAB6A-531D-43BE-BF3F-A06693F71AB0}",
  "eventName": "disconnection",
  "serverGuid": "{182C4DA9-7CA7-416D-9F72-F3527941FA2D}",
  "cameraId": 0,
  "waitingTime": 5
}

```

11. Efetuar requisição de habilitação do monitoramento do evento de perda de sinal da fonte de dados desejada

```

curl -X POST "http://127.0.0.1:8080/api/event-actions/sources"
-H "Accept: application/json" -H "Content-Type:
application/json" -H "Authorization: Bearer eyJ1...XfnQ" -d
"{\"periodGuid\": \"{1A9EAB6A-531D-43BE-BF3F-A06693F71AB0}\",
\"eventName\": \"disconnection\",
\"serverGuid\": \"{182C4DA9-7CA7-416D-9F72-F3527941FA2D}\",
\"cameraId\": 0, \"waitingTime\": 5}"

```

12. Extrair identificação da configuração de monitoramento do retorno da requisição anterior

```

{
  "source": [
    {
      "sourceGuid": "{3E1A7BA2-BC03-4150-8CF4-6B0E1D29B7B6}",
      "periodGuid": "{1A9EAB6A-531D-43BE-BF3F-A06693F71AB0}",
      "periodName": "Período 1",
      "eventName": "signal-lost",
      "eventKind": 3,
      "serverGuid": "{182C4DA9-7CA7-416D-9F72-F3527941FA2D}",
      "serverName": "Servidor DVR",
      "cameraId": 0,
      "cameraName": "Câmera 01",
      "waitingTime": 5
    }
  ]
}

```

13. Montar conteúdo da requisição de definição da ação a ser executada para o monitoramento recém criado

```

{
  "enabled": false,
  "url": "http://www.google.com?q=$server.name&$camera.id$"
}

```

14. Efetuar requisição de definição da ação a ser executada para o monitoramento recém criado

```
curl -X PUT "http://127.0.0.1:8080/api/event-
actions/sources/%7B3E1A7BA2-BC03-4150-8CF4-
6B0E1D29B7B6%7D/actions/send-http-request" -H "Accept:
application/json" -H "Content-Type: application/json" -H
"Authorization: Bearer eyJ1...XfnQ" -d "{\"enabled\": false,
\"url\":
\"http://www.google.com?q=$server.name$&$camera.id$\"}"
```

5.6 Verificar existência de monitoramento do evento de perda de sinal de uma câmera

Para verificar a existência de monitoramento do evento de perda de sinal de uma fonte de eventos do tipo câmera em uma instância do software D-Guard é necessário realizar os seguintes passos:

1. Efetuar requisição de autenticação

```
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\"}"
```

2. Extrair *token* de autenticação do retorno da requisição anterior

```
{
  "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"
  }
}
```

3. Adicionar *token* de autenticação ao cabeçalho das demais requisições

```
-H "Authorization: Bearer eyJ1...XfnQ"
```

4. Efetuar requisição de obtenção da lista de servidores

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

5. Extrair identificação do servidor desejado do retorno da requisição anterior

```
{
  "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
    }
]
}

```

6. Efetuar requisição de obtenção da lista de câmeras do servidor desejado

```

curl -X GET "http://127.0.0.1:8080/api/servers/%7B182C4DA9-7CA7-416D-9F72-F3527941FA2D%7D/cameras" -H "Accept: application/json" -H "Authorization: Bearer eyJ1...XfnQ"

```

7. Extrair identificação da câmera desejada do retorno da requisição anterior

```

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

```

8. Efetuar requisição de obtenção da lista de fontes de eventos já configuradas

```

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

```

9. Verificar se o evento de perda de sinal da fonte de eventos desejada está presente no retorno da requisição anterior

```

{
  "sources": [
    {
      "sourceGuid": "{92BFF008-9170-4501-8003-649A61D93CB8}",
      "periodGuid": "{1A9EAB6A-531D-43BE-BF3F-A06693F71AB0}",
      "periodName": "Período 1",
      "eventName": "motion-detection",
      "eventKind": 2,
      "serverGuid": "{182C4DA9-7CA7-416D-9F72-F3527941FA2D}",
      "serverName": "Servidor DVR",
      "cameraId": 0,
      "cameraName": "Câmera 01"
    },
  ],
}

```

```

{
  "sourceGuid": "{7D7C9C46-67E7-4263-B714-2E5D1C0E74EE}",
  "periodGuid": "{1A9EAB6A-531D-43BE-BF3F-A06693F71AB0}",
  "periodName": "Período 1",
  "eventName": "disconnection",
  "eventKind": 1,
  "serverGuid": "{182C4DA9-7CA7-416D-9F72-F3527941FA2D}",
  "serverName": "Servidor DVR",
  "waitingTime": 5
},
{
  "sourceGuid": "{3E1A7BA2-BC03-4150-8CF4-6B0E1D29B7B6}",
  "periodGuid": "{1A9EAB6A-531D-43BE-BF3F-A06693F71AB0}",
  "periodName": "Período 1",
  "eventName": "signal-lost",
  "eventKind": 3,
  "serverGuid": "{182C4DA9-7CA7-416D-9F72-F3527941FA2D}",
  "serverName": "Servidor DVR",
  "cameraId": 0,
  "cameraName": "Câmera 01",
  "waitingTime": 5
}
]
}

```

10. Extrair identificação da configuração de monitoramento desejada do retorno da requisição anterior

```

{
  "sources": [
    {
      "sourceGuid": "{92BFF008-9170-4501-8003-649A61D93CB8}",
      "periodGuid": "{1A9EAB6A-531D-43BE-BF3F-A06693F71AB0}",
      "periodName": "Período 1",
      "eventName": "motion-detection",
      "eventKind": 2,
      "serverGuid": "{182C4DA9-7CA7-416D-9F72-F3527941FA2D}",
      "serverName": "Servidor DVR",
      "cameraId": 0,
      "cameraName": "Câmera 01"
    },
    {
      "sourceGuid": "{7D7C9C46-67E7-4263-B714-2E5D1C0E74EE}",
      "periodGuid": "{1A9EAB6A-531D-43BE-BF3F-A06693F71AB0}",
      "periodName": "Período 1",
      "eventName": "disconnection",
      "eventKind": 1,
      "serverGuid": "{182C4DA9-7CA7-416D-9F72-F3527941FA2D}",
      "serverName": "Servidor DVR",
      "waitingTime": 5
    },
    {
      "sourceGuid": "{3E1A7BA2-BC03-4150-8CF4-6B0E1D29B7B6}",
      "periodGuid": "{1A9EAB6A-531D-43BE-BF3F-A06693F71AB0}",
      "periodName": "Período 1",
      "eventName": "signal-lost",
      "eventKind": 3,
      "serverGuid": "{182C4DA9-7CA7-416D-9F72-F3527941FA2D}",
      "serverName": "Servidor DVR",
      "cameraId": 0,
      "cameraName": "Câmera 01",

```

```

        "waitingTime": 5
    }
]
}

```

11. Efetuar requisição de obtenção da lista de ações do evento de perda de sinal da fonte de eventos desejada

```

curl -X GET "http://127.0.0.1:8080/api/event-
actions/sources/%7B3E1A7BA2-BC03-4150-8CF4-
6B0E1D29B7B6%7D/actions" -H "Accept: application/json" -H
"Content-Type: application/json" -H "Authorization: Bearer
eyJ1...XfnQ"

```

10. Verificar se a ação de envio de requisição HTTP/GET para o evento de perda de sinal da fonte de eventos desejada está presente no retorno da requisição anterior

```

{
  "actions": {
    "playSound": false,
    "relayControl": false,
    "sendEmail": false,
    "cameraSnapshot": false,
    "openLayout": false,
    "dynamicLayout": false,
    "activatePreset": false,
    "controlRecording": false,
    "switchPreviewStream": false,
    "switchRecordingStream": false,
    "changeProfile": false,
    "activateLPR": false,
    "controlAudio": false,
    "sendSMS": false,
    "monitorSnapshot": false,
    "sendHttpRequest": true,
    "snepCall": false,
    "contactId": false,
    "situator": false,
    "recordBookmark": false
  }
}

```

11. Efetuar requisição de obtenção dos detalhes da ação de envio de requisição HTTP/GET lista de ações da configuração de monitoramento desejada

```

curl -X GET "http://127.0.0.1:8080/api/event-
actions/sources/%7B3E1A7BA2-BC03-4150-8CF4-
6B0E1D29B7B6%7D/actions/send-http-request" -H "Accept:
application/json" -H "Content-Type: application/json" -H
"Authorization: Bearer eyJ1...XfnQ"

```