



INWISE

Be ahead. Get connected!

Instrumentação do NEON via HTTP

Especificação dos serviços de instrumentação do NEON através de requisições HTTP.

Sumário

Instrumentação do NEON via HTTP	3
Executando comandos	4
Exemplo	4
Autenticação.....	5
Exemplo	5
Escutando eventos	6
Exemplo de recebimento de evento	6
Referência.....	7
Comandos.....	7
Eventos	8
Códigos de Erro.....	8
Maiores informações	9

Instrumentação do NEON via HTTP

Com o intuito de prover a integração com aplicativos, páginas web, módulos e componentes de software o NEON pode ser instrumentado de forma simples e segura através do protocolo HTTP, comportando-se como um aplicativo servidor apto a executar comandos e notificar eventos.

Através de requisições HTTP básicas é possível:

- Realizar, atender e cancelar ligações;
- Receber notificações de novas ligações, atendidas e finalizadas;
- Transferir ligações;
- Colocar chamada em espera;
- Enviar DTFM;

Logo após a inicialização, o NEON já está apto para receber requisições HTTP (na porta 15060) compatível com as recomendações da W3C para requisições web entre domínios (<http://www.w3.org/TR/cors/>).

Segue neste documento a descrição para autenticação, execução de comandos, recebimento de eventos e especificação dos formatos de dados. Todos os exemplos citados podem ser testados em qualquer navegador web.

Executando comandos

Para executar um comando no NEON basta enviar uma requisição HTTP para a URL local do NEON (Ex: <http://localhost:15060/Command>). Esta requisição deverá ser composta no formato submissão de formulário application/x-www-form-urlencoded (RFC1867) via GET ou POST. Os dados do formulário submetido são:

```
token=id_de_acesso
&action=nome_do_comando
&param1=dado1
```

Para qualquer requisição de comando realizada, o NEON retornará um XML no formato abaixo:

```
<ReturnData>
  <Success>(true | false)</Success>
  <Error>
    <Code/>
    <SystemCode/>
    <Description/>
  </Error>
  <Data>(XML data content)</Data>
</ReturnData>
```

Exemplo

Requisição – realiza uma ligação

<http://localhost:15060/Command?action=startcall&number=32&token=xxxxxxx>

Resposta

```
<?xml version="1.0" encoding="UTF-8"?>
<ReturnData>
  <Success>>true</Success>
  <Error>
    <Code>0</Code>
    <SystemCode/>
    <Description/>
  </Error>
  <Data/>
</ReturnData>
```

Caso o NEON não esteja com uma conta SIP conectada, os comandos que atuam sobre ligações retornarão o código de erro 601 (não registrado no SIP Proxy).

Através do Id da chamada é possível executar comandos para atender, finalizar e transferir. Este Id pode ser obtido quando o evento *OnNewCall* é disparado (consultar sessão [referência](#) deste documento).

Autenticação

Para acessar qualquer comando ou receber eventos, deve-se realizar uma autenticação identificando a aplicação que instrumentará o NEON com um nome de usuário e senha. O comando de autenticação retornará um id de acesso (token) que deverá ser utilizado para execução de qualquer comando ou recebimento de eventos.

O comando *authenticate* deve ser utilizado para autenticar a aplicação cliente. Este é o único comando que não necessita de um id de acesso.

Exemplo

Requisição
http://localhost:15060/Command?action=authenticate&user=xxxxx&password=xxxxxxx
Resposta
<pre><?xml version="1.0" encoding="UTF-8"?> <ReturnData> <Success>true</Success> <Error> <Code>0</Code> <SystemCode/> <Description/> </Error> <Data> <Token>497a0735</Token> </Data> </ReturnData></pre>

Escutando eventos

Através de requisições HTTP consecutivas a uma URL do NEON – metodologia conhecida como HTTP Polling (RFC 6202). Acessando o endereço <http://localhost:15060/Events?token=xxxxxxx> é possível receber os eventos de ligações ocorridos no NEON.

Exemplo de recebimento de evento

Requisição – realiza uma ligação
http://localhost:15060/Events?token=497a0735
Resposta
<pre><?xml version="1.0" encoding="UTF-8"?> <EventData> <Name>OnNewCall</Name> <Data> <CallId>43</CallId> <Type>outgoing</Type> <From>sip:31@sip.acme.com</From> <To>sip:32@sip.acme.com</To> </Data> </EventData></pre>

Imediatamente após o recebimento da resposta da requisição é necessário que uma nova requisição seja realizada. Caso uma nova requisição não seja realizada dentro de um prazo de 10 segundos, os eventos pendentes serão descartados.

É indispensável que seja enviado um token válido, caso contrário será retornado um código de erro. Qualquer evento notificado pelo NEON retornará um XML no formato abaixo:

```
<EventData>
  <Name />
  <Data>
    <Data1 />
    <DataN />
  </Data>
</EventData>
```

Referência

Comandos

Nome	Parâmetros
Authenticate: Realiza autenticação da aplicação cliente que instrumentará o NEON	Action: authenticate User: <i>nome da aplicação cliente</i> Password: <i>senha</i>
Ex: http://localhost:15060/Command?action=authenticate&user=xxx&password=xxxxxxx	
StartCall: Realiza uma nova ligação para um determinado número	Action: startcall Number: <i>Número a ser discado</i> Token: <i>Chave de acesso da aplicação do cliente</i>
Ex: http://localhost:15060/Command?action=startcall&number=32&token=xxxxxxx	
FinalizeCall: Finaliza uma ligação através do Id da chamada recebido em um dos eventos de ligação	Action: finalizcall Callid: <i>Id da ligação que deseja finalizar</i> Token: <i>Chave de acesso da aplicação do cliente</i>
Ex: http://localhost:15060/Command?action=finalizcall&callid=xxx&token=xxxxxxx	
AcceptCall: Atende uma ligação recebida através do Id da chamada	Action: acceptcall Callid: <i>Id da ligação que deseja atender</i> Token: <i>Chave de acesso da aplicação do cliente</i>
Ex: http://127.0.0.1:15060/Command?action=acceptcall&callid=xxx&token=xxxxxxx	
TransferCall: Transfere uma ligação em recebimento ou atendida para determinado número	Action: transfercall Callid: <i>Id da ligação que deseja transferir</i> To: <i>Número que deseja transferir a ligação</i> Token: <i>Chave de acesso da aplicação do cliente</i>
Ex: http://127.0.0.1:15060/Command?action=transfercall&to=35&callid=xxx&token=xxxxxx	
SendDTMF: Envia sinal de DTMF em uma ligação em andamento	Action: senddtmf Callid: <i>Id da ligação que deseja enviar DTMF</i> TDMF: <i>Número correspondente ao dígito DTMF</i> Token: <i>Chave de acesso da aplicação do cliente</i>
Ex: http://127.0.0.1:15060/Command?action=senddtmf&dtmf=5&callid=xxx&token=xxxxxxx	
SetHold: Coloca uma ligação em estado de espera	Action: sethold Callid: <i>Id da ligação</i> Hold: <i>Valor booleano indicando estado de espera</i> Token: <i>Chave de acesso da aplicação do cliente</i>
Ex: http://127.0.0.1:15060/Command?action=sethold&hold=true&callid=xxx&token=xxxxxxx	

Eventos

Nome	Dados
OnNewCall: Indica que uma nova ligação realizada ou recebida foi iniciada.	CallId: Id da ligação Type: <i>incoming</i> (recebida) <i>outgoing</i> (realizada) From: SIP URI do originário To: SIP URI do destinatário Duration: Duração da ligação em milissegundos SIPError: Código de erro SIP
OnFinalizedCall: Indica que uma ligação foi finalizada.	
OnAcceptCall: Indica que uma ligação foi atendida.	
OnTransferredCall: Indica que uma ligação foi transferida.	

Códigos de Erro

Código	Descrição
0	Nenhum erro
100	Erro no formato dos dados
102	Argumento inválido
500	Erro interno
600	Credenciais inválidas
601	Não registrado no SIP Proxy

Maiores informações

Salvador, 12 de dezembro de 2017.

Versão do documento: 1.2

Última atualização: 12 de dezembro de 2017

InWise Internet Company do Brasil S.A.

www.inwise.com.br

Tel: +55 (71) 3341.1525

Email: joselito@inwise.com.br

Atenção!

Este documento pode conter informação confidencial, legalmente protegida e para conhecimento exclusivo do destinatário. É estritamente proibido a leitura, exame, retransmissão, divulgação, distribuição, cópia ou outro uso da mesma ou a tomada de qualquer ação baseada nesta informação por pessoas ou entidades que não sejam o destinatário. Os conceitos, conclusões e outras informações neste documento que não relacionados aos negócios oficiais da minha empresa serão considerados como não fornecidos nem endossados por ela.