

# Criar dispositivos de rede do ISE usando a API ERS

## Contents

---

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Configurar](#)

[Ativar ERS \(Porta 9060\)](#)

[Criar ERS Admin](#)

[Configurar Postman](#)

[SDK do ISE e autorização de carteiro básico](#)

[Criar NAD usando XML](#)

[Criar NAD usando JSON](#)

[Verificar](#)

[Troubleshooting](#)

---

## Introdução

Este documento descreve o processo para criar dispositivos de acesso à rede (NADs) no ISE através da API ERS usando PostMan como o cliente REST.

## Pré-requisitos

### Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- ISE (Identity Services Engine, mecanismo de serviços de identidade)
- ERS (External RESTful Services, Serviços RESTful externos)
- Clientes do REST como Postman, RESTED, Insomnia, etc.

### Componentes Utilizados

As informações neste documento são baseadas nestas versões de software:

- Cisco ISE (Identity Services Engine) 3.1 patch 6
- Postman REST client v10.17.4



Observação: o procedimento é semelhante ou idêntico para outras versões do ISE e Clientes REST. Você pode usar essas etapas em todas as versões 2.x e 3.x do software ISE, a menos que declarado o contrário.

---

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

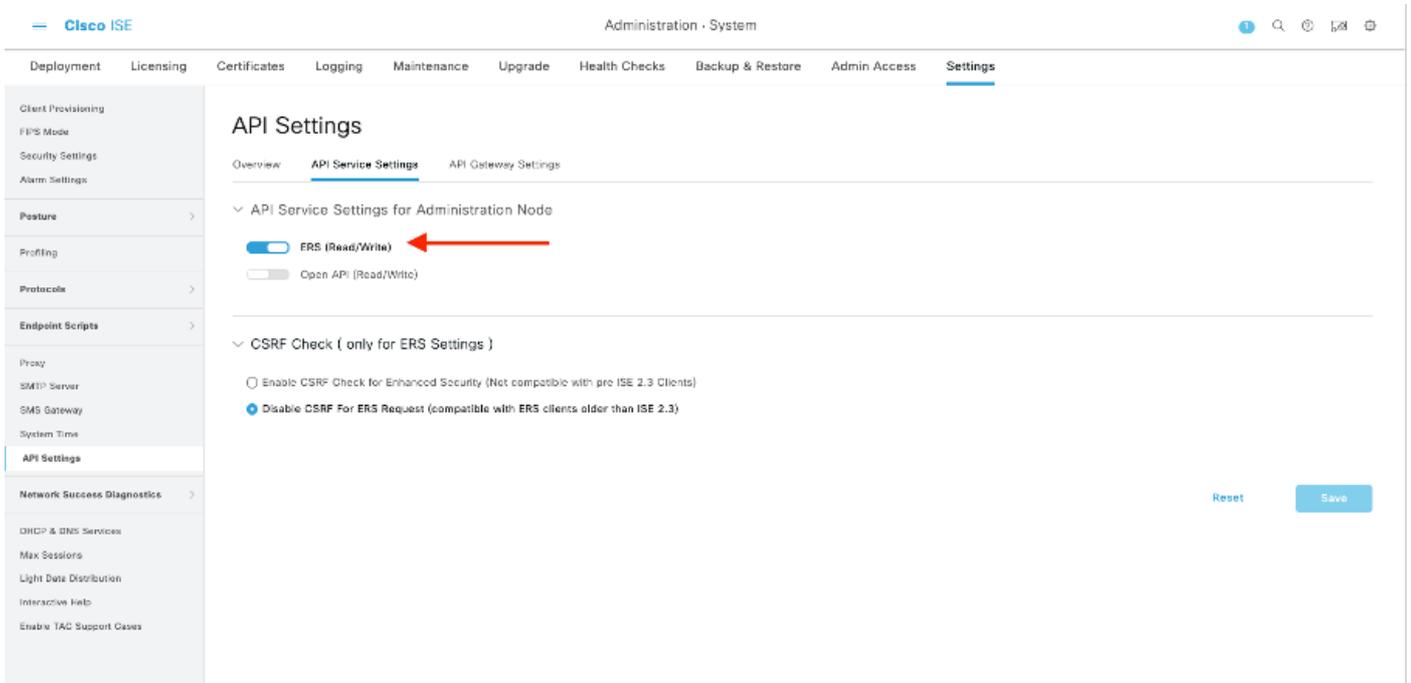
## Configurar

### Ativar ERS (Porta 9060)

As APIs ERS são APIs REST somente HTTPS que operam nas portas 443 e 9060. A porta 9060 é fechada por padrão, por isso precisa ser aberta primeiro. Um tempo limite do servidor será apresentado se os clientes que tentarem acessar essa porta não ativarem o ERS primeiro.

Portanto, o primeiro requisito é ativar o ERS na interface do usuário do administrador do Cisco ISE.

Navegue até Administration > Settings > API Settings e ative o botão de alternância ERS (Read/Write).





Observação: as APIs ERS suportam TLS 1.1 e TLS 1.2. As APIs ERS não suportam TLS 1.0, independentemente da ativação do TLS 1.0 na janela Configurações de segurança da GUI do Cisco ISE (Administração > Sistema > Configurações > Configurações de segurança). A habilitação do TLS 1.0 na janela Configurações de segurança está relacionada apenas ao protocolo EAP e não afeta as APIs ERS.

---

## Criar ERS Admin

Crie um administrador do Cisco ISE, atribua uma senha e adicione o usuário ao grupo admin como ERS Admin. Você pode deixar o restante da configuração vazio.

Admin User

\* Name **ERS-USER** ←

Status **Enabled** ▾

Email   Include system alerts in emails

Expires

Hard Expiry

Inactive account never expires

---

Password

\* Password  ⓘ ←

\* Re-Enter Password  ⓘ

[Generate Password](#)

---

User Information

First Name

Last Name

---

Account Options

Description

Change password on next login

---

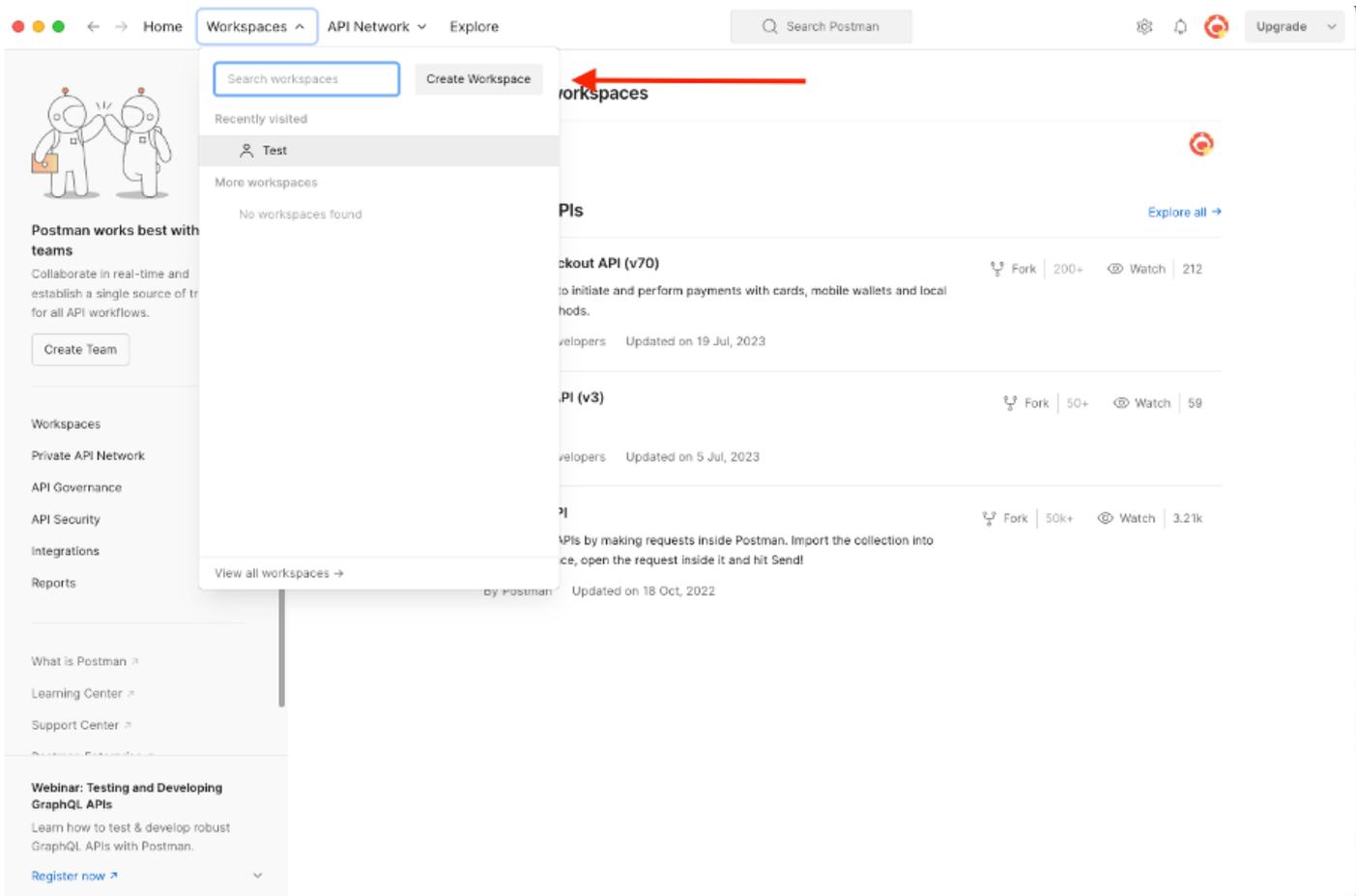
Admin Groups

ERS Admin ▾ + ←

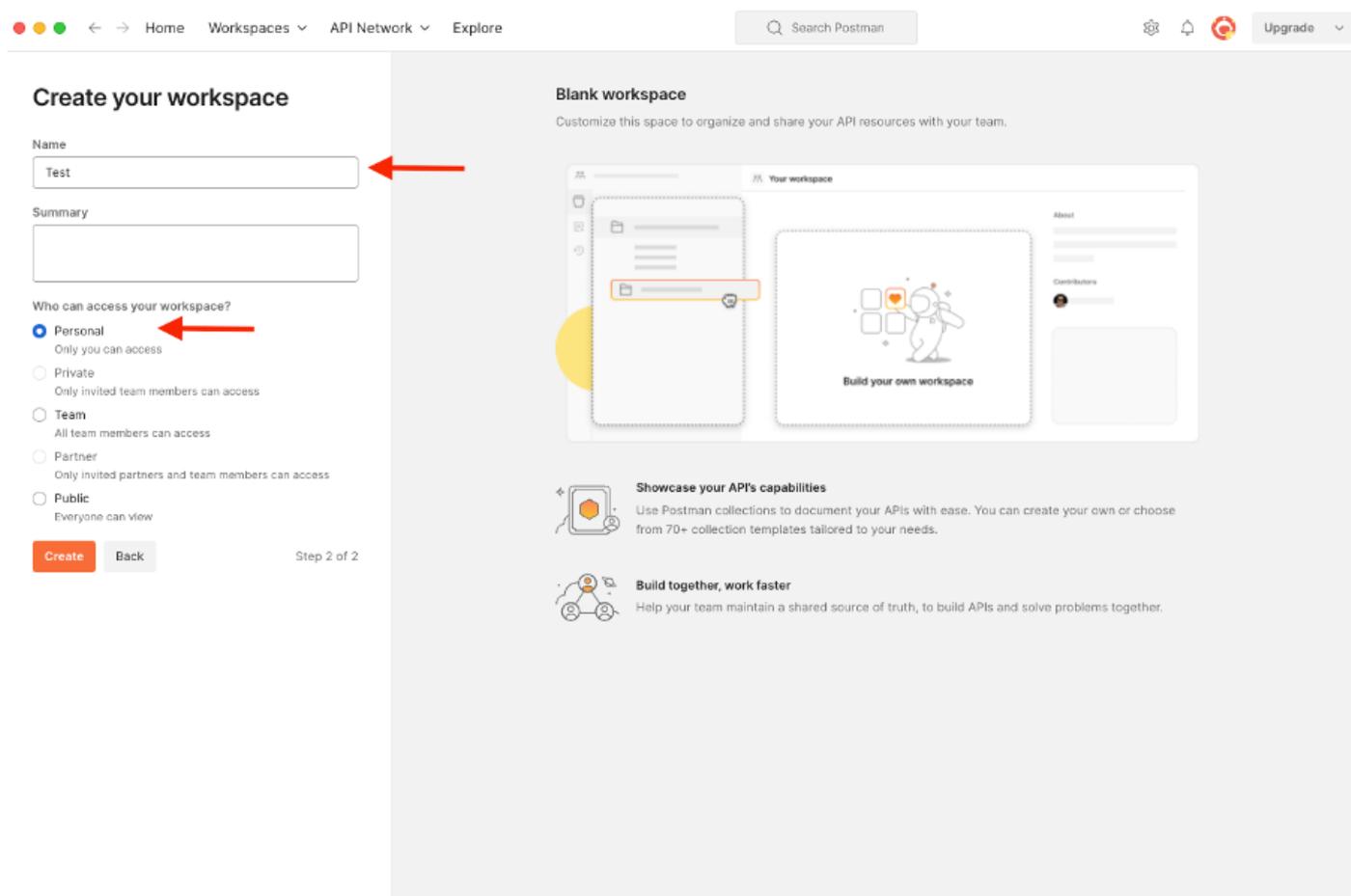
## Configurar Postman

Faça o download ou use a versão online do Postman.

1. Crie um usuário e um espaço de trabalho clicando em Criar espaço de trabalho na guia Espaços de trabalho.



2. Selecione Espaço de Trabalho em Branco e atribua um nome ao espaço de trabalho. Você pode adicionar uma descrição e torná-la pública. Para este exemplo, Personalis foi selecionado.



Depois de criar o espaço de trabalho, você pode configurar as chamadas à API.

## SDK do ISE e autorização de carteiro básico

Para configurar qualquer chamada, acesse primeiro o ISE ERS SDK (Software Developer Kit). Esta ferramenta compila toda a lista de chamadas de API que o ISE pode executar:

1. Navegue até <https://{ise-ip}/ers/sdk>.
2. Faça login usando suas credenciais de administrador do ISE.
3. Expanda a Documentação da API.
4. Role para baixo até encontrar Network Device e clique nele.
5. Nessa opção, você pode encontrar todas as operações disponíveis que pode executar para dispositivos de rede no ISE. Selecione Criar.

External RESTful Services (ERS) Online SDK

Quick Reference

API Documentation

- Filter Policy
- Guest Location
- Guest Smp Notification Configur
- Guest Ssid
- Guest Type
- Guest User
- Hotspot Portal
- IP To SCT Mapping
- IP To SCT Mapping Group
- ISE Service Information
- Identity Group
- Identity Sequence
- Internal User
- My Device Portal
- Native Supplicant Profile
- Network Device
- Network Device Group
- Node Details
- PSN Node Details with Radius Set
- Portal
- Portal Theme
- Profiler Profile
- Pull Deployment Info
- Pxgrid Node
- Pxgrid Settings
- Radius Server Sequence
- RestID Store
- SMS Server
- SXP Connections
- SXP Local Bindings
- SXP Vpns
- Security Groups
- Security Groups ACLs
- Security Groups to Virtual Netwo
- Self Registered Portal
- Sponsor Group
- Sponsor Group Member
- Sponsor Portal
- Sponsored Guest Portal
- Support Bundle Download

Network Device

- Overview
- Resource definition
- Revision History
- Update-By-Name
- Delete-By-Name
- Get-By-Name
- Get-By-Id
- Update
- Get-All
- Delete
- Create
- Get Version
- Bulk Request
- Monitor Bulk Status

Overview

Network Device API allows the client to add, delete, update, and search Network Devices. In this documentation, for each available API you will find the request syntax including the required headers and a response example of a successful flow. Please note that each API description shows weather the API is supported in bulk operation. The Bulk section is showing only 'create' bulk operation however, all other operation which are bulk supported can be used in same way.

Please note that these examples are not meant to be used as is because they have references to DB data. You should treat it as a basic template and edit it before sending to server.

Back to top

Resource definition

Attribute	Type	Required	Default value	Description
name	String	Yes		Resource name
id	String	No		Resource UUID, mandatory for update

Developer Resources

6. Agora você pode ver a configuração necessária para executar a chamada de API usando XML ou JSON em qualquer Cliente Rest, bem como um exemplo de resposta esperada.

Quick Reference

API Documentation

Network Device

Create

Request:

Method: POST

URI: https://10.201.230.99/ers/config/networkdevice

HTTP 'Content-Type' Header: application/xml | application/json

HTTP 'Accept' Header: application/xml | application/json

HTTP 'ERS-Media-Type' Header (Not Mandatory): network.networkdevice.1.1

HTTP 'X-CSRF-TOKEN' Header (Required Only if Enabled from GUI): The Token value from the GET X-CSRF-TOKEN fetch request

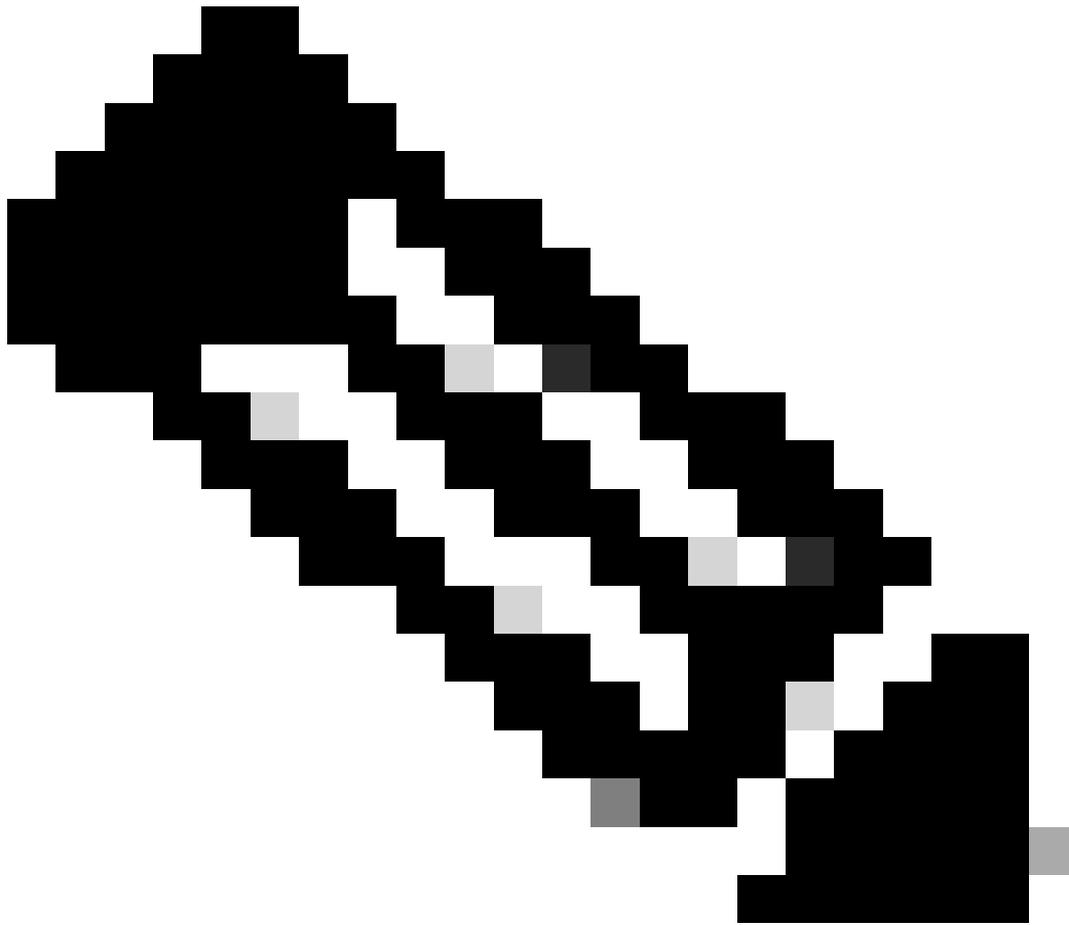
Request Content:

```

XML
<?xml version="1.0" encoding="UTF-8">
<ns0:networkdevice xmlns:ns0="network.ers.ise.cisco.com" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:ns1="ers.ise.cisco.com" xmlns:ers="ers.ise.cisco.com" description="example nd" ns="">
  <authenticationSettings>
    <dtlsRequired>true</dtlsRequired>
    <enableKeyWrap>true</enableKeyWrap>
    <keyEncryptionKey>1234567890123456</keyEncryptionKey>
    <keyInputFormat>ASCII</keyInputFormat>
    <messageAuthenticatorCodeKey>12345678901234567890</messageAuthenticatorCodeKey>
    <radiusSharedSecret>aaaa</radiusSharedSecret>
  </authenticationSettings>
  <coaPort>1700</coaPort>
  <dtlsDnsName>ISE111.il.com</dtlsDnsName>
  <NetworkDeviceIPList>
    <NetworkDeviceIP>
      <ipaddress>1.1.1.1</ipaddress>
      <mask>32</mask>
    </NetworkDeviceIP>
  </NetworkDeviceIPList>
  <NetworkDeviceGroupList>
    <NetworkDeviceGroupLocation#All Locations</NetworkDeviceGroup>
    <NetworkDeviceGroupDevice Type#All Device Types</NetworkDeviceGroup>
  </NetworkDeviceGroupList>
  <profileName>Cisco</profileName>
  <smptSettings>
    <linkTrapQuery>true</linkTrapQuery>
    <macTrapQuery>true</macTrapQuery>
    <originatingPolicyServicesNode>Auto</originatingPolicyServicesNode>
    <pollingInterval>300</pollingInterval>
    <roCommunity>v0aaa</roCommunity>
  </smptSettings>
</ns0:networkdevice>

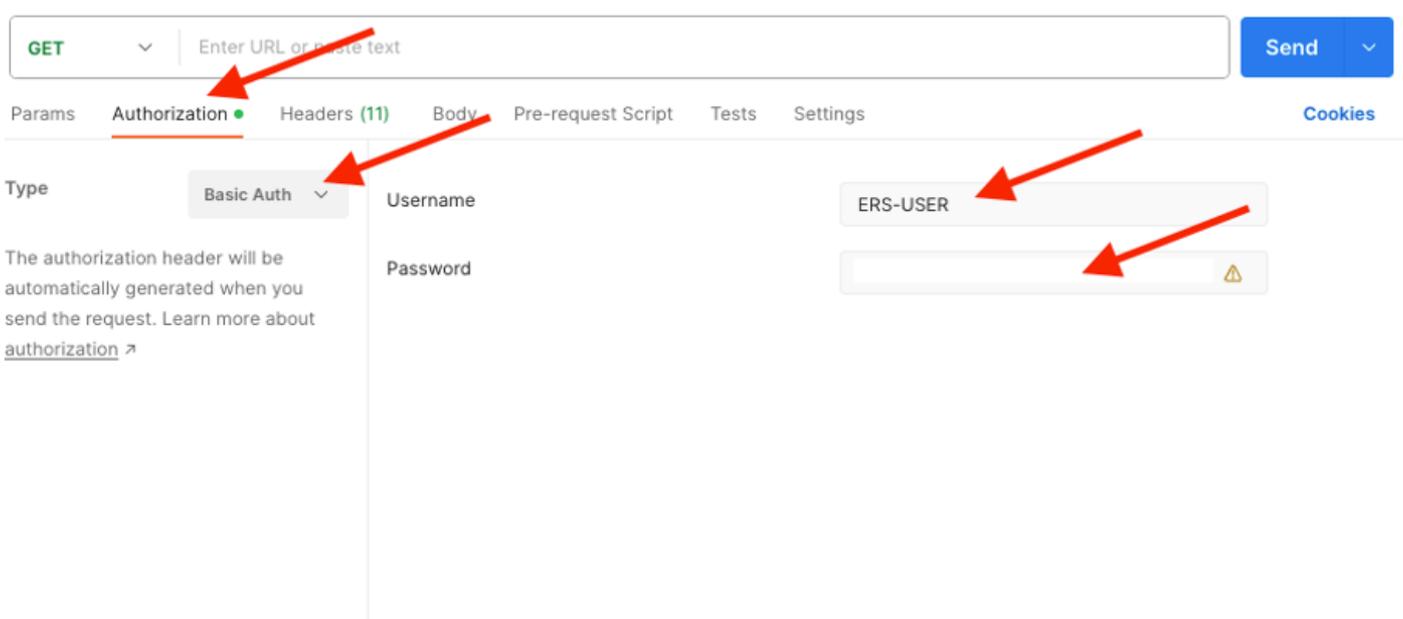
```

7. Voltar para Postman configurar a autenticação básica para ISE. Na guia Authorization, selecione Basic Auth como o tipo de autenticação e adicione as credenciais do usuário do ISE ERS criadas anteriormente no ISE.



Observação: a senha é mostrada como texto claro, a menos que as variáveis sejam configuradas no Postman.

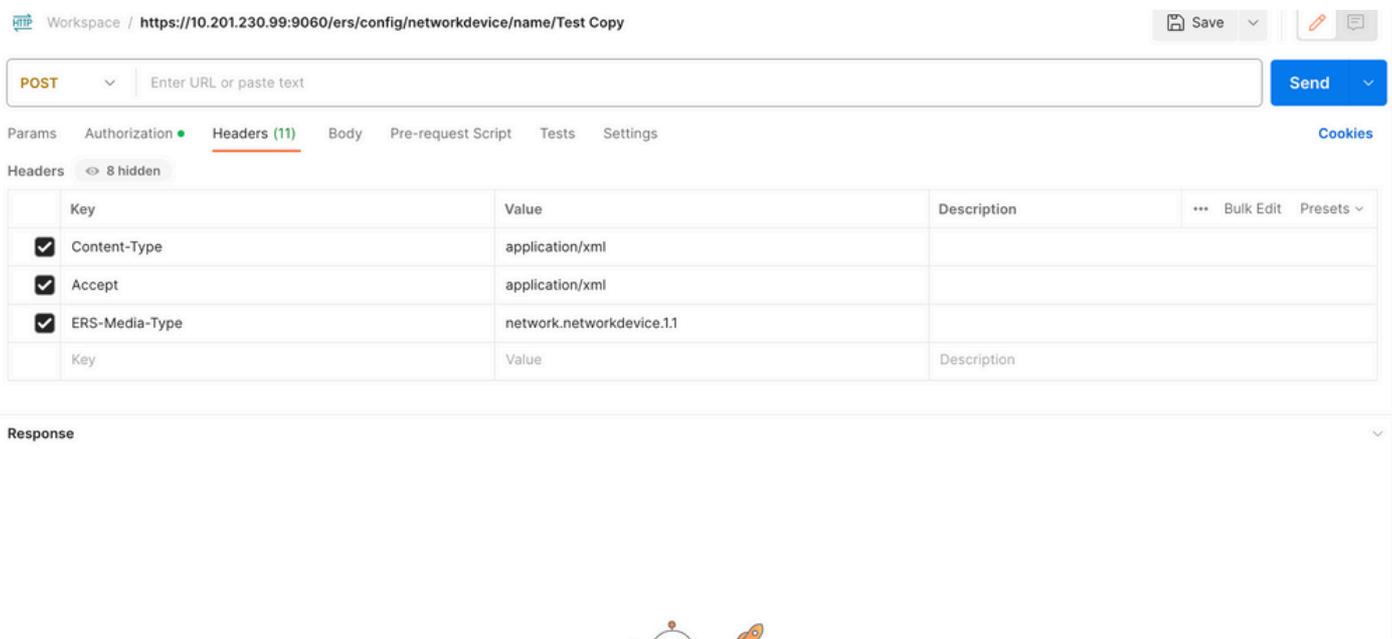
---



## Criar NAD usando XML

Crie TESTNAD1 com as configurações RADIUS TACACS, SNMP e TrustSec usando XML.

1. No SDK, em Criar, estão os cabeçalhos e modelos necessários para executar a chamada, bem como a resposta esperada.
2. Vá até a guia Cabeçalhos e configure os cabeçalhos necessários para a chamada de API conforme visto no SDK. A configuração do cabeçalho deve ser semelhante a esta:



3. Vá até o cabeçalho Body e selecione raw. Isso permite colar o modelo XML necessário para criar o NAD.

Workspace / <https://10.201.230.99:9060/ers/config/networkdevice/name/Test Copy> Save

POST Enter URL or paste text Send

Params Authorization Headers (11) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL XML Beautify

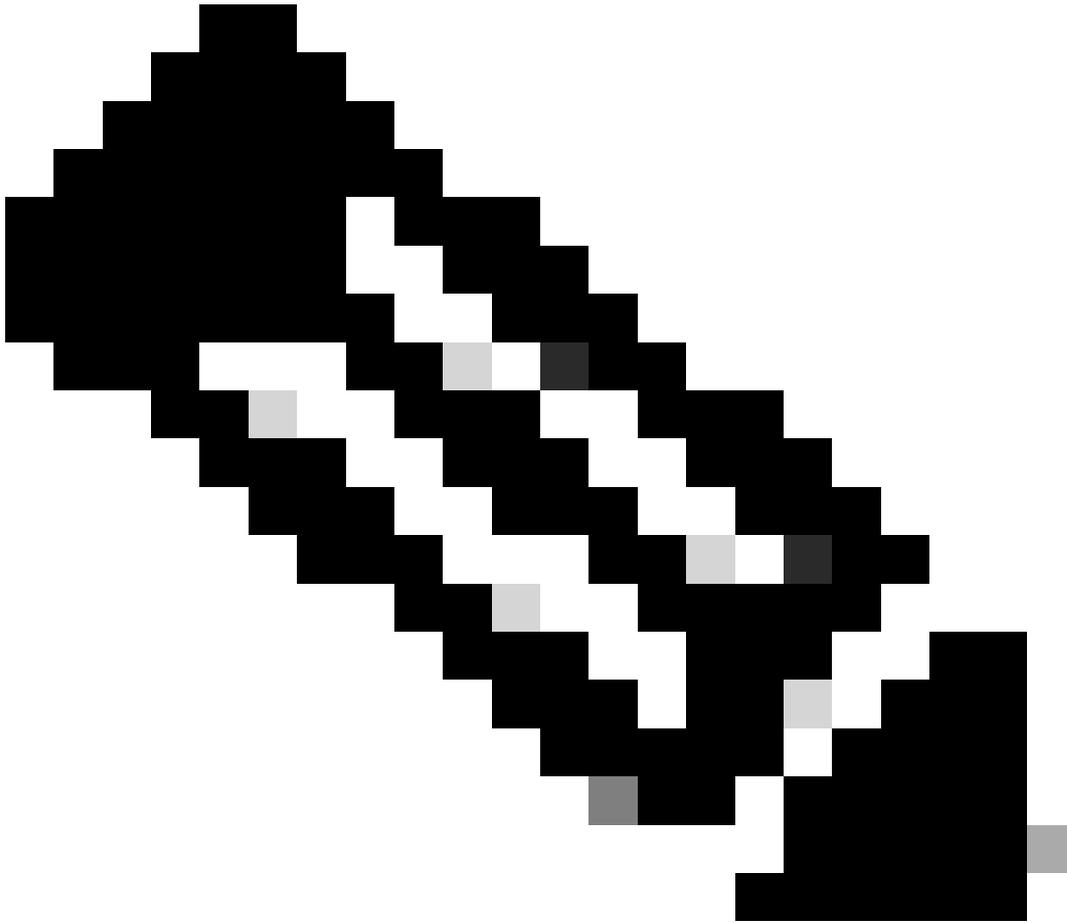
1

Response



#### 4. O modelo XML é semelhante a este (altere os valores conforme necessário):

```
<?xml version="1.0" encoding="UTF-8"?> <ns0:networkdevice xmlns:ns0="network.ers.ise.cisco.com" xmlns:xs="Schema XML File"
xmlns:ns1="ers.ise.cisco.com" xmlns:ers="ers.ise.cisco.com" description="This NAD was added via ERS API" name="TESTNAD1">
<authenticationSettings> <dtlsRequired>true</dtlsRequired> <enableKeyWrap>true</enableKeyWrap>
<keyEncryptionKey>1234567890123456</keyEncryptionKey> <keyInputFormat>ASCII</keyInputFormat>
<messageAuthenticatorCodeKey>12345678901234567890</messageAuthenticatorCodeKey>
<radiusSharedSecret>cisco123</radiusSharedSecret> </authenticationSettings> <coaPort>1700</coaPort>
<dtlsDnsName>Domain</dtlsDnsName> <NetworkDeviceIPList> <NetworkDeviceIP> <ipaddress>NAD IP Address</ipaddress>
<mask>32</mask> </NetworkDeviceIP> </NetworkDeviceIPList> <NetworkDeviceGroupList> <NetworkDeviceGroup>Location#All
Locations#LAB</NetworkDeviceGroup> <NetworkDeviceGroup>Device Type#All Device Types#Access-Layer</NetworkDeviceGroup>
</NetworkDeviceGroupList> <profileName>Cisco</profileName> <snmpsettings> <linkTrapQuery>true</linkTrapQuery>
<macTrapQuery>true</macTrapQuery> <originatingPolicyServicesNode>Auto</originatingPolicyServicesNode>
<pollingInterval>3600</pollingInterval> <roCommunity>aaa</roCommunity> <version>ONE</version> </snmpsettings> <tacacsSettings>
<connectModeOptions>ON_LEGACY</connectModeOptions> <sharedSecret>cisco123</sharedSecret> </tacacsSettings> <trustsecsettings>
<deviceAuthenticationSettings> <sgaDeviceId>TESTNAD1</sgaDeviceId> <sgaDevicePassword>cisco123</sgaDevicePassword>
</deviceAuthenticationSettings> <deviceConfigurationDeployment> <enableModePassword>cisco123</enableModePassword>
<execModePassword>cisco123</execModePassword> <execModeUsername>Admin</execModeUsername>
<includeWhenDeployingSGTUpdates>true</includeWhenDeployingSGTUpdates> </deviceConfigurationDeployment>
<pushIdSupport>false</pushIdSupport> <sgaNotificationAndUpdates> <coaSourceHost>ise3-1test</coaSourceHost>
<downloadEnvironmentDataEveryXSeconds>86400</downloadEnvironmentDataEveryXSeconds>
<downloadPeerAuthorizationPolicyEveryXSeconds>86400</downloadPeerAuthorizationPolicyEveryXSeconds>
<downloadSGACLListsEveryXSeconds>86400</downloadSGACLListsEveryXSeconds>
<otherSGADevicesToTrustThisDevice>false</otherSGADevicesToTrustThisDevice>
<reAuthenticationEveryXSeconds>86400</reAuthenticationEveryXSeconds>
<sendConfigurationToDevice>false</sendConfigurationToDevice>
<sendConfigurationToDeviceUsing>ENABLE_USING_COA</sendConfigurationToDeviceUsing> </sgaNotificationAndUpdates>
</trustsecsettings> </ns0:networkdevice>
```



**Observação:** é importante observar que as próximas linhas serão necessárias somente se `<enableKeyWrap>{false|true}</enableKeyWrap>` for definido como **true**. Caso contrário, o mesmo pode ser excluído do modelo XML:

---

```
<keyEncryptionKey>1234567890123456</keyEncryptionKey> <keyInputFormat>ASCII</keyInputFormat>  
<messageAuthenticatorCodeKey>12345678901234567890</messageAuthenticatorCodeKey>
```

Você pode remover a configuração que não precisa do modelo e apenas deixar os dados que realmente precisam ser adicionados durante a criação do NAD. Como exemplo, aqui está o mesmo modelo, mas somente com a configuração TACACS. Independentemente da configuração necessária, certifique-se de que o modelo termine com `</ns0:networkdevice>`.

```
<?xml version="1.0" encoding="UTF-8"?> <ns0:networkdevice xmlns:ns0="network.ers.ise.cisco.com" xmlns:xs="Schema XML File"
```

```
xmlns:ns1="ers.ise.cisco.com" xmlns:ers="ers.ise.cisco.com" description="This NAD was added via ERS API" name="TESTNAD1">
<NetworkDeviceIPList> <NetworkDeviceIP> <ipaddress>NAD IP Address</ipaddress> <mask>32</mask> </NetworkDeviceIP>
</NetworkDeviceIPList> <NetworkDeviceGroupList> <NetworkDeviceGroup>Location#All Locations#LAB</NetworkDeviceGroup>
<NetworkDeviceGroup>Device Type#All Device Types#Access-Layer</NetworkDeviceGroup> </NetworkDeviceGroupList>
<profileName>Cisco</profileName> <tacacsSettings> <connectModeOptions>ON_LEGACY</connectModeOptions>
<sharedSecret>cisco123</sharedSecret> </tacacsSettings> </ns0:networkdevice>
```

5. Cole o modelo XML para **raw** no cabeçalho **Body**.

6. Selecione **POST** como o método, cole `https://{ISE-ip}/ers/config/networkdevice` e clique em **Send**. **Se tudo tiver sido configurado corretamente, você deverá ver uma mensagem 201 Created** e o resultado vazio.

The screenshot shows a REST client interface with the following details:

- Workspace: `https://10.201.230.99:9060/ers/config/networkdevice/name/Test Copy`
- Method: **POST**
- URL: `https://10.201.230.99/ers/config/networkdevice`
- Body tab selected, showing XML payload (lines 50-59).
- Response status: **201 Created** (Time: 791 ms, Size: 1.22 KB)
- Buttons: Save, Beautify, Cookies, Headers (19), Test Results, XML, Raw, Preview, Visualize.

7. Confirme se o NAD foi criado executando uma **chamada GET** para o NAD ou verificando a lista ISE NAD.

Workspace / <https://10.201.230.99:9060/ers/config/networkdevice/name/Test Copy>

GET <https://10.201.230.99/ers/config/networkdevice> Send

Params Authorization Headers (13) Body Pre-request Script Tests Settings

Headers 10 hidden

Key	Value	Description
Content-Type	application/json	
Accept	application/json	
ERS-Media-Type	network.networkdevice.1.1	

Body Cookies (2) Headers (15) Test Results Status: 200 OK Time: 237 ms Size: 3.13 KB Save as Example

Pretty Raw Preview Visualize JSON

```

52   "type": "application/json"
53   }
54 }
55 {
56   "id": "afe572d0-5bcc-11ee-9ab7-9a446445bd4f",
57   "name": "TESTNAD1",
58   "description": "This NAD was added via ERS API",
59   "link": {
60     "rel": "self",
61     "href": "https://10.201.230.99/ers/config/networkdevice/afe572d0-5bcc-11ee-9ab7-9a446445bd4f",
62     "type": "application/json"
63   }
64 },
65 {
66   "id": "63efbc20-4f5a-11ed-b560-6e7768fe732e",
67   "name": "Wireless-9800",
68   "description": "Wireless Controller C9800",
69   "link": {
70     "rel": "self"

```

Cisco ISE Administration - Network Resources

Network Devices Network Device Groups Network Device Profiles External RADIUS Servers RADIUS Server Sequences NAC Managers External MDM Location Services

### Network Devices

Selected 0 Total 6

Edit + Add Duplicate Import Export Generate PAC Delete

Name	IP/Mask	Profile Name	Location	Type	Description
TESTNAD1	1.1.1.1/32	Cisco	LAB	Access-Layer	This NAD was added via ERS API

## Criar NAD usando JSON

Crie TESTNAD2 com as configurações RADIUS TACACS, SNMP e TrustSec usando JSON.

1. No SDK, em **Criar**, estão os cabeçalhos e modelos necessários para executar a chamada, bem como a resposta esperada.
2. Vá até a guia **Cabeçalhos** e configure os cabeçalhos necessários para a chamada de API conforme visto no SDK. A configuração do cabeçalho deve ser semelhante a esta:

Workspace / <https://10.201.230.99:9060/ers/config/networkdevice/name/Test> Save Send

POST  Send

Params Authorization Headers (12) Body Pre-request Script Tests Settings Cookies

Headers 9 hidden

Key	Value	Description	Bulk Edit	Presets
<input checked="" type="checkbox"/> Content-Type	application/json			
<input checked="" type="checkbox"/> Accept	application/json			
<input checked="" type="checkbox"/> ERS-Media-Type	network.networkdevice.1.1			
Key	Value	Description		

3. Vá até o cabeçalho **Body** e selecione **raw**. Isso permite colar o modelo JSON necessário para criar o NAD.

Workspace / <https://10.201.230.99:9060/ers/config/networkdevice/name/Test Copy> Save Send

POST  Send

Params Authorization Headers (11) **Body** Pre-request Script Tests Settings Cookies

none
  form-data
  x-www-form-urlencoded
  **raw**
 binary
  GraphQL
  XML

1

Response



4. O modelo JSON deve ter esta aparência (altere os valores conforme necessário):

```
{ "NetworkDevice": { "name": "TESTNAD2", "description": "This NAD was added via ERS API", "authenticationSettings": {
"radiusSharedSecret": "cisco123", "enableKeyWrap": true, "dtlsRequired": true, "keyEncryptionKey": "1234567890123456",
"messageAuthenticatorCodeKey": "12345678901234567890", "keyInputFormat": "ASCII" }, "snmpsettings": { "version": "ONE",
"roCommunity": "aaa", "pollingInterval": 3600, "linkTrapQuery": true, "macTrapQuery": true, "originatingPolicyServicesNode": "Auto" },
"trustsecsettings": { "deviceAuthenticationSettings": { "sgaDeviceId": "TESTNAD2", "sgaDevicePassword": "cisco123" },
"sgaNotificationAndUpdates": { "downloadEnvironmentDataEveryXSeconds": 86400, "downloadPeerAuthorizationPolicyEveryXSeconds":
86400, "reAuthenticationEveryXSeconds": 86400, "downloadSGACLListsEveryXSeconds": 86400, "otherSGADevicesToTrustThisDevice":
false, "sendConfigurationToDevice": false, "sendConfigurationToDeviceUsing": "ENABLE_USING_COA", "coaSourceHost": "ise3-1test" },
"deviceConfigurationDeployment": { "includeWhenDeployingSGTUpdates": true, "enableModePassword": "cisco123", "execModePassword":
"cisco123", "execModeUsername": "Admin" }, "pushIdSupport": "false" }, "tacacsSettings": { "sharedSecret": "cisco123",
"connectModeOptions": "ON_LEGACY" }, "profileName": "Cisco", "coaPort": 1700, "dtlsDnsName": "Domain", "NetworkDeviceIPList": [ {
"ipaddress": "NAD IP Adress", "mask": 32 } ], "NetworkDeviceGroupList": [ "Location#All Locations", "Device Type#All Device Types" ] }
```



**Observação:** é importante observar que as próximas linhas serão necessárias somente se `enableKeyWrap`:{false|true}, for definido como **true**. Caso contrário, o mesmo pode ser excluído do modelo JSON:

---

`"keyEncryptionKey": "1234567890123456", "messageAuthenticatorCodeKey": "12345678901234567890", "keyInputFormat": "ASCII"` Você também pode remover a configuração que não precisa do modelo e apenas deixar os dados que realmente precisam ser adicionados durante a criação do NAD.

5. Cole o modelo JSON para **raw** no cabeçalho **Body**.

6. Selecione **POST** como o método, cole `https://{ISE-ip}/ers/config/networkdevice` e clique em Send. **Se tudo tiver sido configurado corretamente, você deverá ver uma mensagem 201 Created** e o resultado vazio.

Workspace / <https://10.201.230.99:9060/ers/config/networkdevice/name/Test Copy> Save

POST <https://10.201.230.99/ers/config/networkdevice> Send

Params Authorization Headers (13) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "NetworkDevice": {
3     "name": "TESTNAD2",
4     "description": "This NAD was added via ERS API",
5     "authenticationSettings": {
6       "radiusSharedSecret": "cisco123",
7       "enableKeyWrap": true,
8       "dtlsRequired": true,
9       "keyEncryptionKey": "1234567890123456",
10      "messageAuthenticatorCodeKey": "12345678901234567890",
11      "keyFormat": "ASCII"
12    }
13  }
14 }
```

Body Cookies (2) Headers (17) Test Results Status: 201 Created Time: 678 ms Size: 1.03 KB Save as Example

Pretty Raw Preview Visualize JSON

1

7. Confirme se o NAD foi criado executando uma chamada GET para o NAD ou verificando a lista ISE NAD.

Workspace / <https://10.201.230.99:9060/ers/config/networkdevice/name/Test Copy> Save

GET <https://10.201.230.99/ers/config/networkdevice> Send

Params Authorization Headers (13) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "NetworkDevice": {
3     "name": "TESTNAD2",
4     "description": "This NAD was added via ERS API",
5     "authenticationSettings": {
6       "radiusSharedSecret": "cisco123",
7       "enableKeyWrap": true,
8       "dtlsRequired": true,
9       "keyEncryptionKey": "1234567890123456",
10      "messageAuthenticatorCodeKey": "12345678901234567890",
11      "keyFormat": "ASCII"
12    }
13  }
14 }
```

Body Cookies (2) Headers (18) Test Results Status: 200 OK Time: 659 ms Size: 3.74 KB Save as Example

Pretty Raw Preview Visualize JSON

```
57   "name": "TESTNAD1",
58   "description": "This NAD was added via ERS API",
59   "link": {
60     "rel": "self",
61     "href": "https://10.201.230.99/ers/config/networkdevice/afe572d0-5bcc-11ee-9ab7-9a446445bd4f",
62     "type": "application/json"
63   }
64 },
65 {
66   "id": "9dd45a60-5bd7-11ee-9ab7-9a446445bd4f",
67   "name": "TESTNAD2",
68   "description": "This NAD was added via ERS API",
69   "link": {
70     "rel": "self",
71     "href": "https://10.201.230.99/ers/config/networkdevice/9dd45a60-5bd7-11ee-9ab7-9a446445bd4f",
72     "type": "application/json"
73   }
74 },
75 }
```

## Verificar

Se você puder acessar a página da GUI do serviço de API, por exemplo, <https://{iseip}:{port}/api/swagger-ui/index.html> ou <https://{iseip}:9060/ers/sdk>, isso significa que o serviço de API está funcionando conforme esperado.

## Troubleshooting

- Todas as operações REST são auditadas e os registros são registrados nos registros do sistema.
- Para solucionar problemas relacionados às APIs abertas, defina o **Nível de Log** do componente **apiservice** como **DEBUG** na janela **Configuração do Log de Depuração**.
- Para solucionar problemas relacionados às APIs ERS, defina o **Nível de Log** do componente **ers** como **DEBUG** na janela **Debug Log Configuration**. Para visualizar essa janela, navegue até a GUI do Cisco ISE, clique no ícone Menu e escolha **Operations > Troubleshoot > Debug Wizard > Debug Log Configuration**.
- Você pode fazer o download dos logs na janela **Download Logs**. Para visualizar essa janela, navegue até a GUI do Cisco ISE, clique no ícone **Menu** e escolha **Operações > Solução de problemas > Logs de download**.
- Você pode optar por fazer download de um pacote de suporte na guia Pacote de suporte clicando no botão **Download** na guia, ou fazer download dos logs de depuração do api-service na guia Logs de depuração clicando no valor do Arquivo de log para o log de depuração do api-service.

## Sobre esta tradução

A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês ([link fornecido](#)) seja sempre consultado.