

# ERS API를 사용하여 ISE 네트워크 디바이스 생성

## 목차

---

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[구성](#)

[ERS 활성화\(포트 9060\)](#)

[ERS 관리자 생성](#)

[Postman 설정](#)

[ISE SDK 및 기본 Postman 권한 부여](#)

[XML을 사용하여 NAD 생성](#)

[JSON을 사용하여 NAD 생성](#)

[다음을 확인합니다.](#)

[문제 해결](#)

---

## 소개

이 문서에서는 PostMan을 REST 클라이언트로 사용하여 ERS API를 통해 ISE에서 NAD(Network Access Device)를 만드는 프로세스에 대해 설명합니다.

## 사전 요구 사항

### 요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- ISE(Identity Services Engine)
- ERS(외부 RESTful 서비스)
- REST 클라이언트 Postman, REST, Insomnia 등

### 사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 버전을 기반으로 합니다.

- Cisco ISE (Identity Services Engine) 3.1 패치 6
- Postman REST 클라이언트 v10.17.4

---

참고: 절차는 다른 ISE 버전 및 REST 클라이언트에서 비슷하거나 동일합니다. 달리 명시되지 않는 한 모든 2.x 및 3.x ISE 소프트웨어 릴리스에서 이 단계를 사용할 수 있습니다.

---

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

## 구성

### ERS 활성화(포트 9060)

ERS API는 포트 443 및 포트 9060을 통해 작동하는 HTTPS 전용 REST API입니다. 포트 9060은 기본적으로 닫혀 있으므로 먼저 열어야 합니다. 이 포트에 액세스하려는 클라이언트가 ERS를 먼저 활성화하지 않을 경우 서버의 시간 초과가 표시됩니다. 따라서 첫 번째 요구 사항은 Cisco ISE 관리 UI에서 ERS를 활성화 하는 것 입니다.

Administration(관리) > Settings(설정) > API Settings(API 설정)로 이동하고 ERS(Read/Write) 토글

버튼을 활성화합니다.

— Cisco ISE Administration - System

Deployment Licensing Certificates Logging Maintenance Upgrade Health Checks Backup & Restore Admin Access **Settings**

Client Provisioning  
FIPS Mode  
Security Settings  
Alarm Settings

Feature >  
Profiling  
Protocols >  
Endpoint Scripts >  
Proxy  
SMTP Server  
SMS Gateway  
System Time  
**API Settings**  
Network Success Diagnostics >  
DHCP & DNS Services  
Max Sessions  
Light Data Distribution  
Interactive Help  
Enable TAC Support Cases

## API Settings

Overview **API Service Settings** API Gateway Settings

API Service Settings for Administration Node

ERS (Read/Write) ←

Open API (Read/Write)

CSRF Check ( only for ERS Settings )

Enable CSRF Check for Enhanced Security (Not compatible with pre ISE 2.3 Clients)

Disable CSRF For ERS Request (compatible with ERS clients older than ISE 2.3)

Reset Save

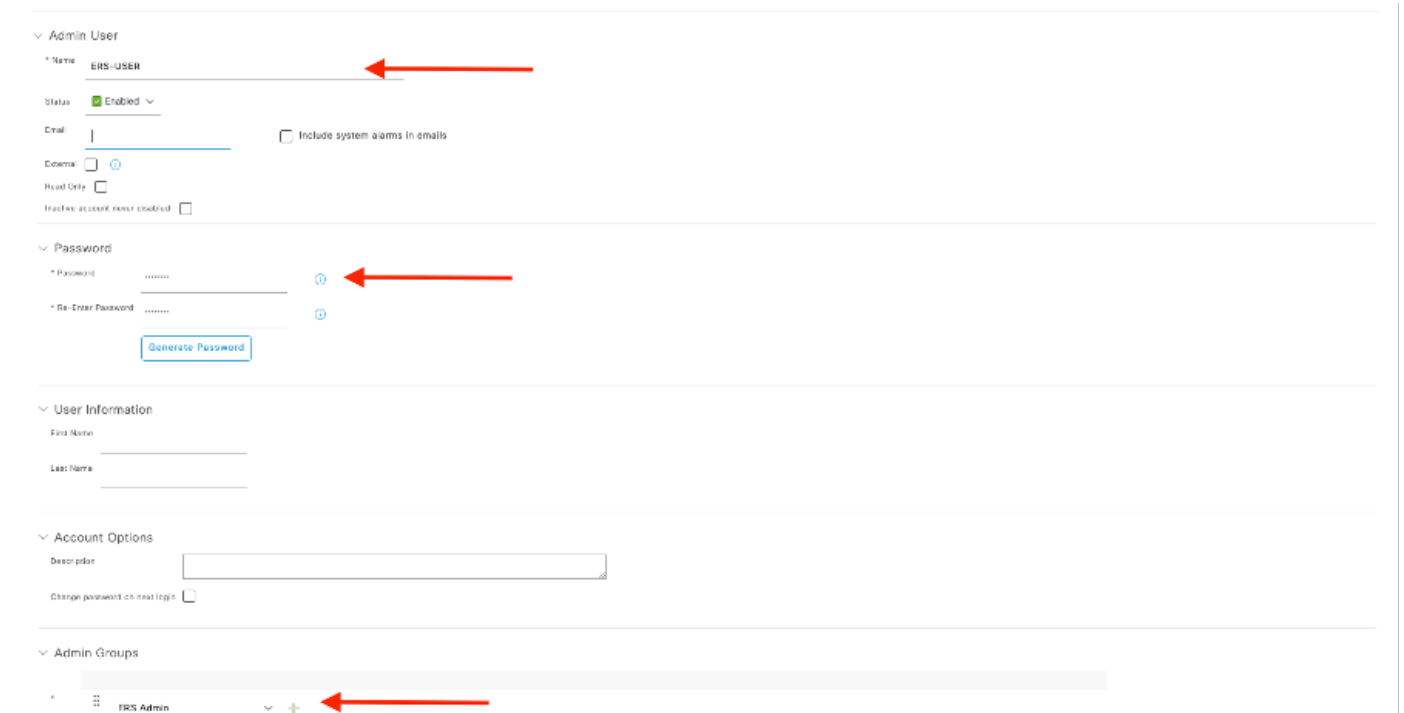


참고: ERS API는 TLS 1.1 및 TLS 1.2를 지원합니다. ERS API는 Cisco ISE GUI의 Security Settings(보안 설정) 창에서 TLS 1.0을 활성화해도 TLS 1.0을 지원하지 않습니다 (Administration(관리) > System(시스템) > Settings(설정) > Security Settings(보안 설정)). Security Settings(보안 설정) 창에서 TLS 1.0을 활성화하는 것은 EAP 프로토콜에만 관련되며 ERS API에는 영향을 주지 않습니다.

---

## ERS 관리자 생성

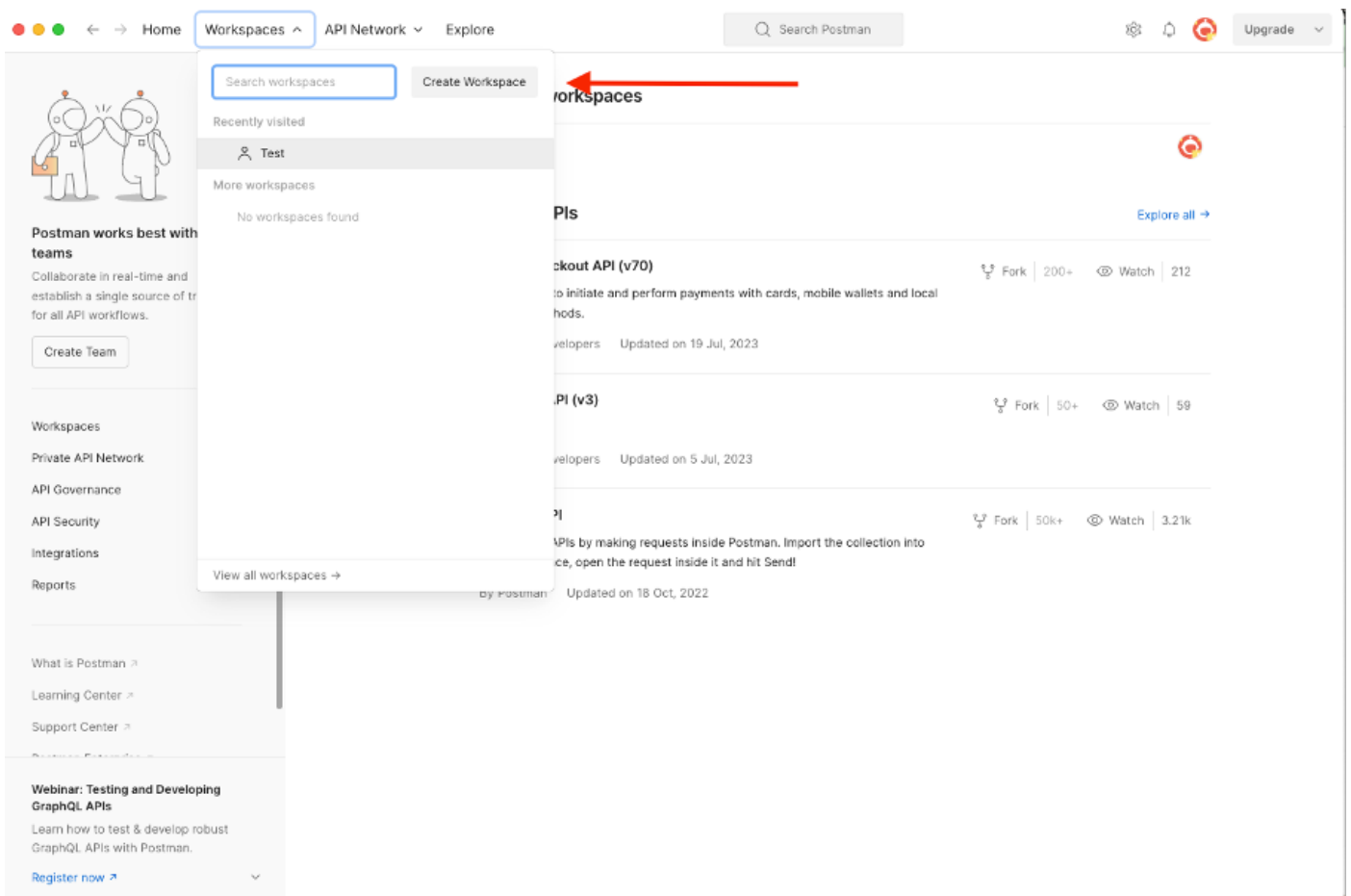
Cisco ISE 관리자를 만들고 암호를 할당한 다음 사용자를 관리자 그룹에 ERS 관리자로 추가합니다 . 나머지 컨피그레이션은 비워둘 수 있습니다.



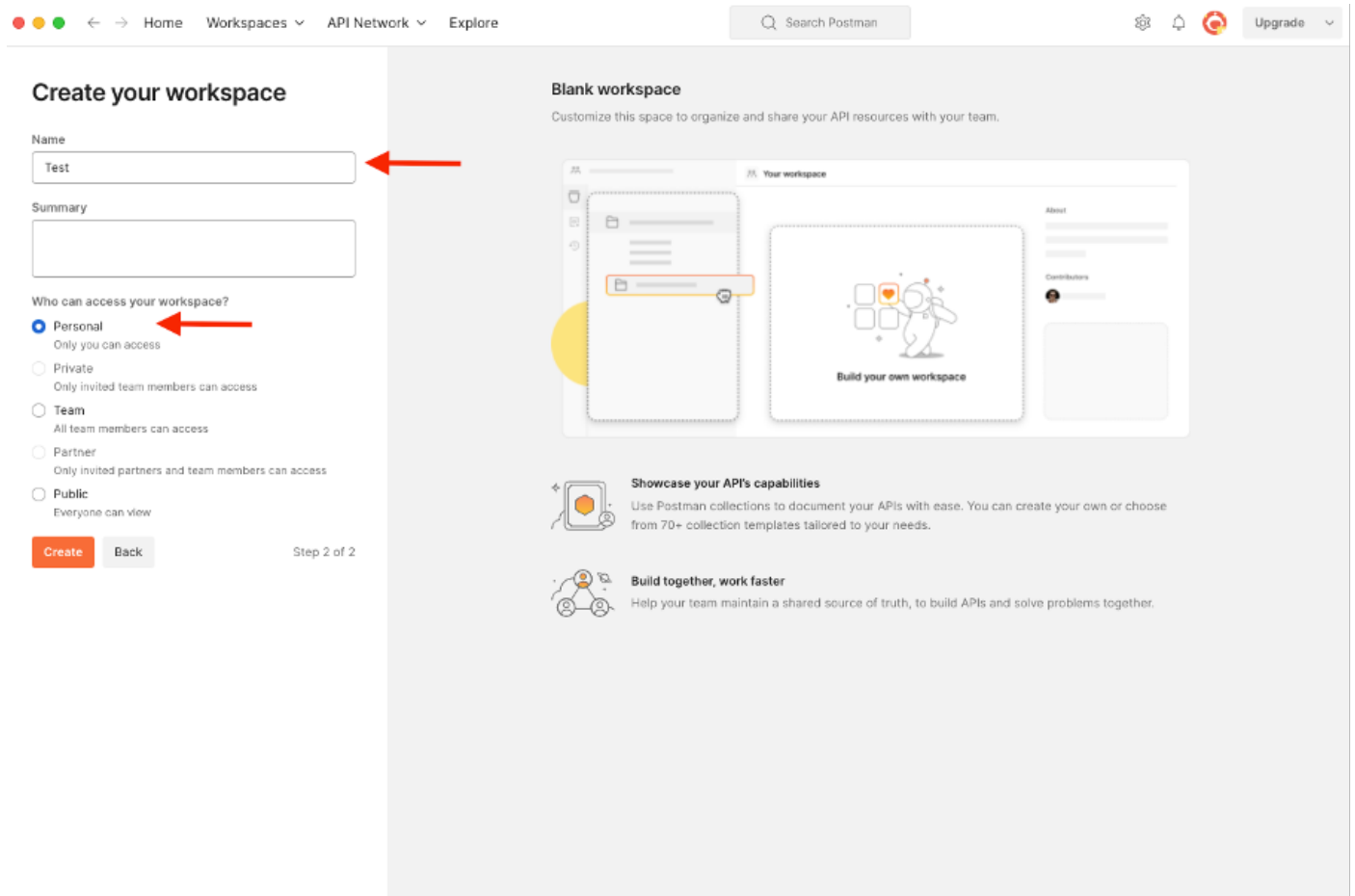
## Postman 설정

Postman의 온라인 버전을 다운로드하거나 사용합니다.

1. 작업공간(Workspaces) 탭에서 작업공간 생성(Create Workspace)을 클릭하여 사용자를 생성하고 작업공간을 생성합니다.



2. 빈 Workspace를 선택하고 Workspace에 이름을 지정합니다. 설명을 추가하고 공개할 수 있습니다. 이 예에서는 Personalis가 선택됩니다.



작업 영역을 생성했으면 이제 API 호출을 구성할 수 있습니다.

## ISE SDK 및 기본 Postman 권한 부여

통화를 구성하려면 먼저 ISE ERS SDK(소프트웨어 개발자 키트)에 액세스합니다. 이 도구는 ISE에서 수행할 수 있는 API 호출의 전체 목록을 컴파일합니다.

1. <https://{{ise-ip}}/ers/sdk>로 이동합니다.
2. ISE 관리자 자격 증명을 사용하여 로그인합니다.
3. API 문서를 확장합니다.
4. Network Device(네트워크 디바이스)를 찾을 때까지 아래로 스크롤하고 클릭합니다.
5. 이제 이 옵션에서 ISE의 네트워크 디바이스에 대해 수행할 수 있는 모든 작업을 찾을 수 있습니다. 생성을 선택합니다.

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. 이제 모든 Rest 클라이언트에서 XML 또는 JSON을 사용하여 API 호출을 수행하는 데 필요한 컨피그레이션과 예상 응답 예를 볼 수 있습니다.

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

Create

Request:

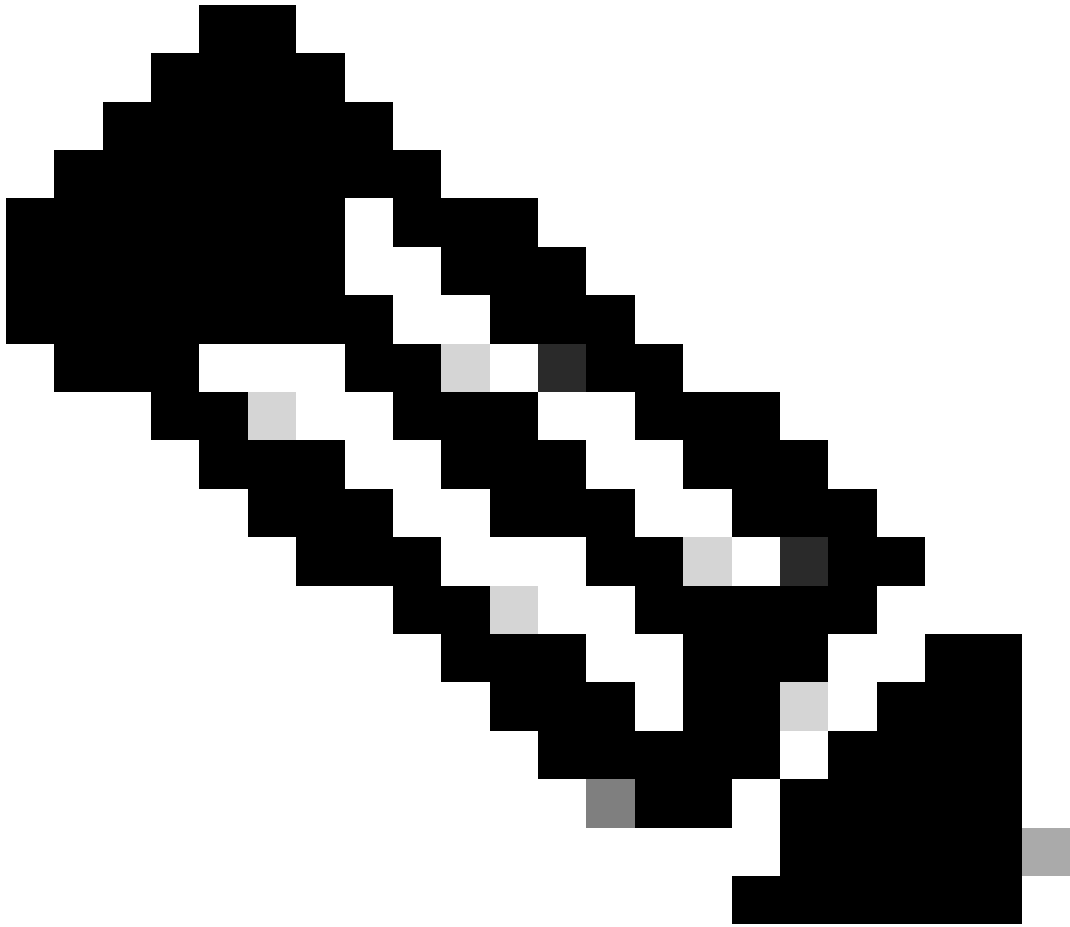
```

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

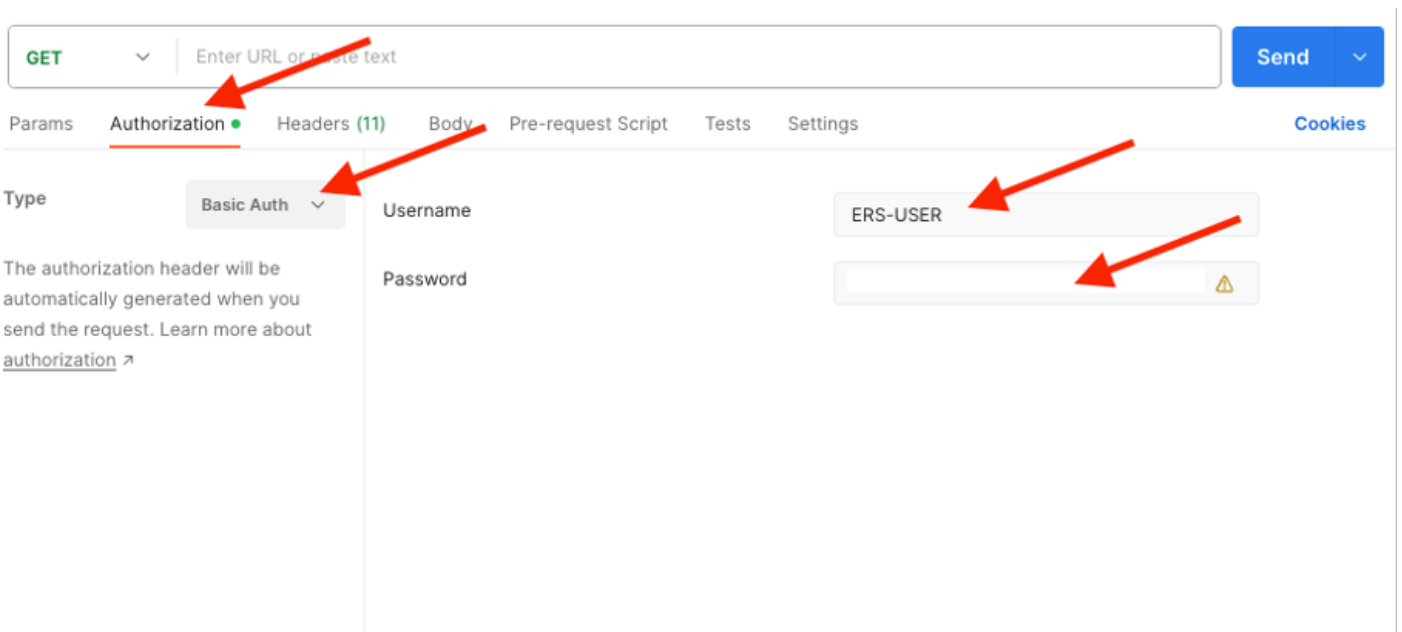
Request Content:
XML
<?xml version="1.0" encoding="UTF-8">
<ns0:networkdevice xmlns:ns0="network.ers.ise.cisco.com" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:ns1="ers.ise.cisco.com" xmlns:ers="ers.ise.cisco.com" description="example nd" ns="">
  <authenticationSettings>
    <dtlsRequired>true</dtlsRequired>
    <enableKeyWrap>true</enableKeyWrap>
    <keyEncryptionKey>1234567890123456</keyEncryptionKey>
    <keyInputFormat>ASCII</keyInputFormat>
    <messageAuthenticatorCodeKey>12345678901234567890</messageAuthenticatorCodeKey>
    <radiusSharedSecret>aaaa</radiusSharedSecret>
  </authenticationSettings>
  <coaPort>1700</coaPort>
  <dtlsDnsName>ISE211.il.com</dtlsDnsName>
  <NetworkDeviceIPList>
    <NetworkDeviceIP>
      <ipaddress>1.1.1</ipaddress>
      <mask>32</mask>
    </NetworkDeviceIP>
  </NetworkDeviceIPList>
  <NetworkDeviceGroupList>
    <NetworkDeviceGroupLocationAll Locations</NetworkDeviceGroup>
    <NetworkDeviceGroupDevice TypeAll Device Types</NetworkDeviceGroup>
  </NetworkDeviceGroupList>
  <profileName>Cisco</profileName>
  <smppSettings>
    <linkTrapQuery>true</linkTrapQuery>
    <macTrapQuery>true</macTrapQuery>
    <originatingPolicyServicesNode>Auto</originatingPolicyServicesNode>
    <pollingInterval>300</pollingInterval>
    <roCommunity>v0aaa</roCommunity>
  </smppSettings>
</ns0:networkdevice>

```

7. Postman으로 돌아가서 ISE에 대한 기본 인증을 구성합니다. Authorization(권한 부여) 탭에서 Basic Auth(기본 인증)를 인증 유형으로 선택하고 ISE에서 이전에 생성한 ISE ERS 사용자 자격 증명을 추가합니다.



참고: Postman에 변수가 구성되지 않은 경우 비밀번호는 일반 텍스트로 표시됩니다.





## XML을 사용하여 NAD 생성

XML을 사용하여 RADIUS TACACS, SNMP 및 TrustSec 설정으로 TESTNAD1을 생성합니다.

1. SDK의 Create(생성)에는 호출을 수행하는 데 필요한 헤더와 템플릿과 예상 응답이 있습니다.
2. Headers(헤더) 탭으로 이동하고 SDK에 표시된 대로 API 호출에 필요한 헤더를 구성합니다. 헤더 구성은 다음과 같아야 합니다.

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

Headers 8 hidden

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

Response

3. 본문 헤더로 이동하고 raw를 선택합니다. 이렇게 하면 NAD를 만드는 데 필요한 XML 템플릿을 붙여넣을 수 있습니다.

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

1

Response

4. XML 템플리트는 다음과 같습니다(필요에 따라 값 변경).

```
<?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>
```

---

**참고:** `<enableKeyWrap>{false|true}</enableKeyWrap>`이 `true`로 설정된 경우에만 다음 줄이 필요합니다. 그렇지 않으면 XML 템플릿에서 동일한 항목을 삭제할 수 있습니다.

---

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

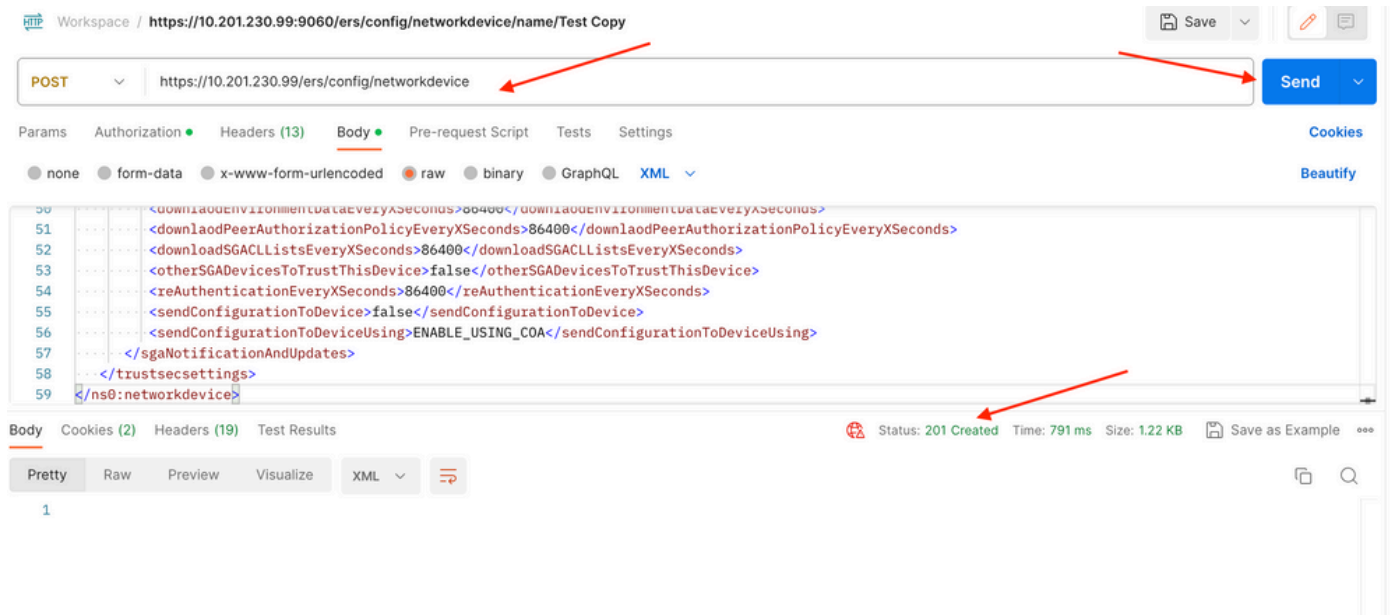
필요 없는 컨피그레이션을 템플릿에서 제거하고 NAD를 생성하는 동안 실제로 추가해야 하는 데이터를 그대로 둘 수 있습니다. 예를 들어, 동일한 템플릿이지만 TACACS 컨피그레이션에서만 사용할 수 있습니다. 필수 컨피그레이션과 상관없이 템플릿이 `</ns0:networkdevice>`로 끝나야 합니다.

```
<?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. 본문 헤더 아래에 원시의 XML 템플릿을 붙여넣습니다.

6. 방법으로 POST를 선택하고 <https://{ISE-ip}/ers/config/networkdevice>를 붙여 넣고 Send(보내기)를 클릭합니다. 모든 것이 올바르게 구성된 경우 201 Created(201 생성) 메시지와 결과가 비어 있어야 합니다.



7. NAD에 대한 GET 호출을 수행하거나 ISE NAD 목록을 확인하여 NAD가 생성되었는지 확인합니다.

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 Cookies

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

Default Device Device Security Settings

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

## JSON을 사용하여 NAD 생성

JSON을 사용하여 RADIUS TACACS, SNMP 및 TrustSec 설정을 사용하는 TESTNAD2를 생성합니다.

1. SDK의 Create(생성)에는 호출을 수행하는 데 필요한 헤더와 템플릿과 예상 응답이 있습니다.
2. Headers(헤더) 탭으로 이동하고 SDK에 표시된 대로 API 호출에 필요한 헤더를 구성합니다. 헤더 구성은 다음과 같아야 합니다.

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

POST Enter URL or paste text

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. 본문 헤더로 이동하고 raw를 선택합니다. 이렇게 하면 NAD를 생성하는 데 필요한 JSON 템플릿을 붙여넣을 수 있습니다.

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


POST Enter URL or paste text

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. JSON 템플릿은 다음과 같아야 합니다(필요에 따라 값 변경).

```
{ "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" ] }
```

---

**참고:** `enableKeyWrap`이(가) `true`로 설정된 경우에만 다음 줄이 필요합니다. 그렇지 않으면 JSON 템플릿에서 동일한 항목을 삭제할 수 있습니다.

---

`"keyEncryptionKey": "1234567890123456", "messageAuthenticatorCodeKey": "12345678901234567890", "keyInputFormat": "ASCII"` 템플릿에서 필요하지 않은 컨피그레이션을 제거하고 NAD를 생성하는 동안 실제로 추가해야 하는 데이터만 남겨 둘 수도 있습니다.

5. **Body 헤더** 아래에 `raw`에 대한 JSON 템플릿을 붙여넣습니다.

6. 방법으로 **POST**를 선택하고 <https://{ISE-ip}/ers/config/network device>를 붙여 넣고 **Send(보내기)**를 클릭합니다. 모든 것이 올바르게 구성된 경우 **201 Created(201 생성) 메시지**와 결과가 비어 있어야 합니다.

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

7. NAD에 대한 GET 호출을 수행하거나 ISE NAD 목록을 확인하여 NAD가 생성되었는지 확인합니다.

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

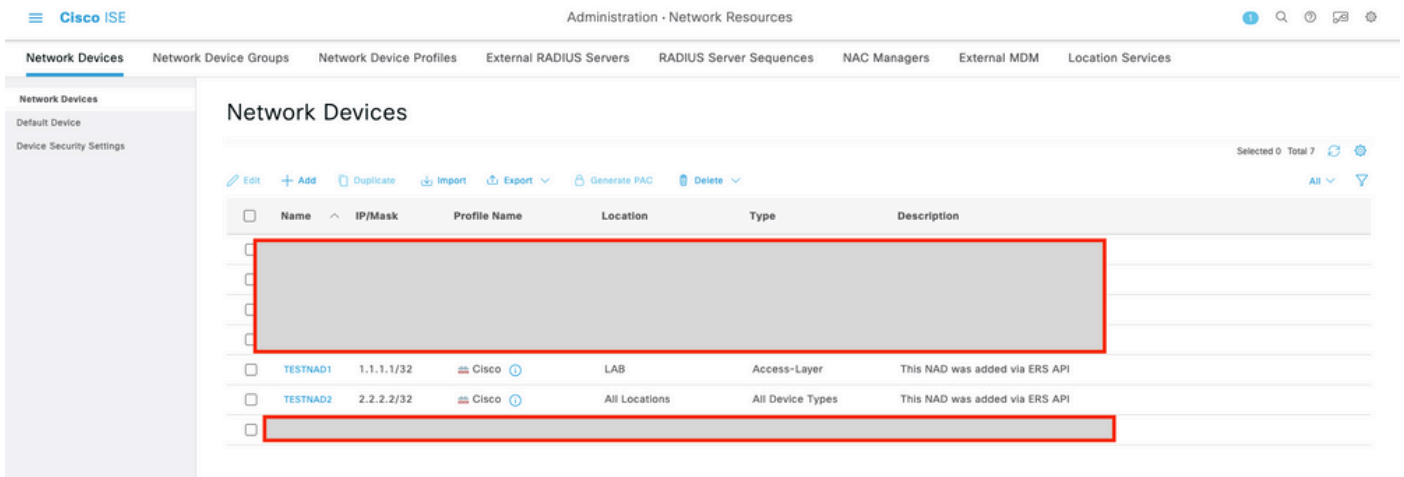
```
1 {
2   "NetworkDevice": {
3     "name": "TESTNAD2",
4     "description": "This NAD was added via ERS API",
5     "authenticationSettings": {
6       "radiusSharedSecret": "cisco123",
7       "enableKeyWrap": true,
8       "dtlsRequired": true,
9       "keyEncryptionKey": "1234567890123456",
10      "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 {
58   "name": "TESTNAD1",
59   "description": "This NAD was added via ERS API",
60   "link": {
61     "rel": "self",
62     "href": "https://10.201.230.99/ers/config/networkdevice/afe572d0-5bcc-11ee-9ab7-9a446445bd4f",
63     "type": "application/json"
64   }
65 },
66 {
67   "id": "9dd45a60-5bd7-11ee-9ab7-9a446445bd4f",
68   "name": "TESTNAD2",
69   "description": "This NAD was added via ERS API",
70   "link": {
71     "rel": "self",
72     "href": "https://10.201.230.99/ers/config/networkdevice/9dd45a60-5bd7-11ee-9ab7-9a446445bd4f",
73     "type": "application/json"
74   }
75 }
```





다음을 확인합니다.

API 서비스 GUI 페이지에 액세스할 수 있는 경우(예: <https://{iseip}:{port}/api/swagger-ui/index.html> 또는 <https://{iseip}:9060/ers/sdk>)는 API 서비스가 예상대로 작동함을 의미합니다.

### 문제 해결

- 모든 REST 작업은 감사되며 로그는 시스템 로그에 기록됩니다.
- Open API와 관련된 문제를 해결하려면 Debug Log Configuration(로그 컨피그레이션 디버그) 창에서 **apiservice** 구성 요소의 **Log Level(로그 레벨)**을 **DEBUG(디버그)**로 설정합니다.
- ERS API와 관련된 문제를 해결하려면 Debug Log Configuration(로그 컨피그레이션 디버그) 창에서 **ers** 구성 요소의 **Log Level(로그 레벨)**을 **DEBUG(디버그)**로 설정합니다. 이 창을 보려면 Cisco ISE GUI로 이동하여 메뉴 아이콘을 클릭하고 **Operations(운영) > Troubleshoot(문제 해결) > Debug Wizard(디버그 마법사) > Debug Log Configuration(디버그 로그 컨피그레이션)**을 선택합니다.
- Download Logs(로그 다운로드) 창에서 **로그**를 다운로드할 수 있습니다. 이 창을 보려면 Cisco ISE GUI로 이동하여 **메뉴** 아이콘을 클릭하고 **Operations(운영) > Troubleshoot(문제 해결) > Download Logs(로그 다운로드)**를 선택합니다.
- Support Bundle(지원 번들) 탭의 **Download(다운로드)** 버튼을 클릭하여 지원 번들을 다운로드하거나, Debug Logs(디버그 로그) 탭에서 **api-service** 디버그 로그에 대한 **Log File(로그 파일)**을 클릭하여 **api-service 디버그 로그**를 다운로드하도록 선택할 수 있습니다.

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.