

使用FMC API资源管理器排除对象创建故障

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[限制](#)

[背景信息](#)

[配置](#)

[在FMC上启用API](#)

[遇到常见错误](#)

[创建主机对象](#)

[单个主机对象](#)

[批量主机对象](#)

[创建网络对象](#)

[单个网络对象](#)

[批量网络对象](#)

[创建网络范围对象](#)

[单个网络范围对象](#)

[批量网络范围对象](#)

[创建FQDN对象](#)

[单个FQDN对象](#)

[批量FQDN对象](#)

[创建服务对象](#)

[单个服务对象](#)

[批量服务对象](#)

[故障排除](#)

简介

本文档介绍API管理员如何使用API资源管理器将网络、端口和URL对象批量推送到FMC。

先决条件

要求

Cisco 建议您了解以下主题：

- 了解各种REST API（应用编程接口）调用。（[REST API基础](#)）
- 了解FMC（Firepower管理中心）中的API资源管理器（[关于API资源管理器](#)）
- FMC API快速[入门指南回顾](#)
- 了解FMC中不同类型的对象([FMC中的对象管理](#))

使用的组件

- 支持REST API (版本7.4或更高版本) 并启用REST API的Firepower管理中心
- API资源管理器集成到FMC中

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始 (默认) 配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

限制

- FMC不接受大于64个字符的对象名称。
- 对象名称不得在对象名称的开头有空格，在结尾不能有分号。
- 负载在一个批量推送中不能包含超过1,000个条目。
- 负载大小在单个批量推送中不能超过2MB。

背景信息

REST API因轻量级的可编程方法而日益流行，网络管理员可以使用该方法来配置和管理其网络。FMC支持使用任何REST客户端以及使用内置API资源管理器的配置和管理。

本文提供故障排除步骤，以解决通过Firepower管理中心(FMC)API资源管理器创建对象时遇到的问题。无论您使用的是网络对象、主机对象还是FTD配置，这些步骤都有助于确定常见问题并实施有效的解决方案。

在所有网络对象类型中保持通用的JSON属性包括：

1. Value — 此属性提供类似带有CIDR的Network值的信息，例如，主机对象可以将x.x.x.x作为IP，网络可以将x.x.x.x/<prefix>作为带有CIDR的子网。
2. Type — 此属性提供有关用于配置对象(主机、网络、服务、范围和FQDN)的对象类型的信息。
3. Name — 此属性提供有关对象名称的信息
4. 说明 — 此属性可用于提供有关对象的一些有用信息或注释
5. Overrideable — 对象覆盖允许您定义对象的替代值，系统将其用于您指定的设备。这是可用于创建对象的可选属性。

配置

在FMC上启用API

在Cisco FMC上启用REST API。有关“如何启用REST API”和“最佳实践”的详细信息，请参阅[在Cisco FMC上启用REST API](#)

导航到System > Configuration > REST API Preferences > Select "Enable REST API"

API Explorer位于管理中心，可通过管理中心访问：

https://<management_center_IP_or_name>:<https_port>/api/api-explorer



注意：如果使用标准HTTPS端口TCP 443，则从URL中安全删除“：<https_port”。

遇到常见错误

我们正在讨论使用API资源管理器和POST方法创建对象时遇到的最常见错误。之后将讨论每个网络对象的正确格式。

1.无效的访问令牌：

Code	Details
401	Error: 401 Response body <pre>{ "error": { "category": "FRAMEWORK", "messages": [{ "description": "Access token invalid." }], "severity": "ERROR" } }</pre>

无效访问令牌的JSON响应



注意：当会话过期时，会遇到此错误。重新加载FMC API资源管理器并重新进行身份验证。

2.使用的格式无效：

```
{
  "name": "API_1",
  "type": "Host",
  "value": "10.5.3.20"
  "description": "Test Description"
}
```

POST

/api/fmc_config/v1/domain/{domainUUID}/object/hosts

Retrieves, deletes, creates, or modifies the host object associated with the specified ID. If no ID is specified for a GET, retrieves list of all host objects. *Check the response section for applicable examples (if any).*

Parameters

Cancel

Reset

Name	Description
bulk boolean (query)	Enables bulk create for host objects. --
domainUUID * required string (path)	Domain UUID e276abec-e0f2-11e3-8169-6d9ed49b625f

Request body required

application/json

Input representation of host object.

Examples:

[Modified value]

```
{
  "name": "TestHost",
  "type": "Host",
  "value": "10.5.3.20"
  "description": "Test Description"
}
```

创建对象的JSON请求示例

Code

Details

422

Error: 422

Response body

```
{
  "error": {
    "category": "OTHER",
    "messages": [
      {
        "description": "Unprocessable Entity - Unexpected character ('\\' (code 34)): was expecting comma to separate Object entries at line: 5, column: 4"
      }
    ],
    "severity": "ERROR"
  }
}
```

由于缺少逗号，JSON响应失败



注意：JSON格式中缺少逗号时，最常遇到此错误。在本例中(“value”：“10.5.3.20”)没有导致此故障的逗号。只有JSON格式的最后一个Attribute不能有逗号，否则可能会导致故障。

3.创建对象时使用的输入无效：

```
{  
  "name": "net2",  
  "value": "1.1.0.0/24",  
  "overridable": false,  
  "description": "Network obj 2",  
  "type": "Network"  
}
```

Curl

```
curl -X 'POST' \
  'https://[redacted]/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/hosts' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -H 'X-auth-access-token: e5ebd0a0-0960-4f03-a9f9-71e91cf5478d' \
  -d '{
    "name": "net2",
    "value": "1.1.0.0/24",
    "overridable": false,
    "description": "Network obj 2",
    "type": "Network"
  }'
```

Request URL

https://[redacted]/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/hosts

Server response

Code

400

Details

Error: 400

Response body

```
{
  "error": {
    "category": "FRAMEWORK",
    "messages": [
      {
        "description": "Input type mismatch. You have entered an invalid value 1.1.0.0/24 for selected type Host."
      }
    ],
    "severity": "ERROR"
  }
}
```

无效类型的JSON响应失败



注意：在本示例中，尝试在网络对象的具有JSON负载的Hosts类别中创建对象，结果为错误“Input type mismatch”。使用为每个网络对象指定的正确格式（主机、网络、FQDN、服务和范围）。

4.使用GET方法响应负载创建对象

```
{
  "links": {
    "self": "https://10.197.207.3/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/obj",
    "parent": "https://10.197.207.3/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/o",
  },
  "type": "Host",
  "name": "API_2",
  "id": "00000000-0000-0ed3-0000-034359741703"
}
```

Responses

Curl

```
curl -X 'GET' \
  'https://[redacted]/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/hosts?offset=2&limit=1' \
  -H 'accept: application/json' \
  -H 'X-auth-access-token: e5ebd0a0-0960-4f03-a9f9-71e91cf5478d'
```

Request URL

```
https://[redacted]/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/hosts?offset=2&limit=1
```

Server response

Code	Details
------	---------

200	
-----	--

Response body

```
{
  "links": {
    "self": "https://[redacted]/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/hosts?offset=2&limit=1"
  },
  "items": [
    {
      "links": {
        "self": "https://[redacted]/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/hosts/00000000-0000-0ed3-0000-034359741703",
        "parent": "https://[redacted]/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/networkaddresses"
      },
      "type": "Host",
      "name": "API_2",
      "id": "00000000-0000-0ed3-0000-034359741703"
    }
  ]
}
```

获取主机对象的GET请求

Code

Details

400

Error: 400

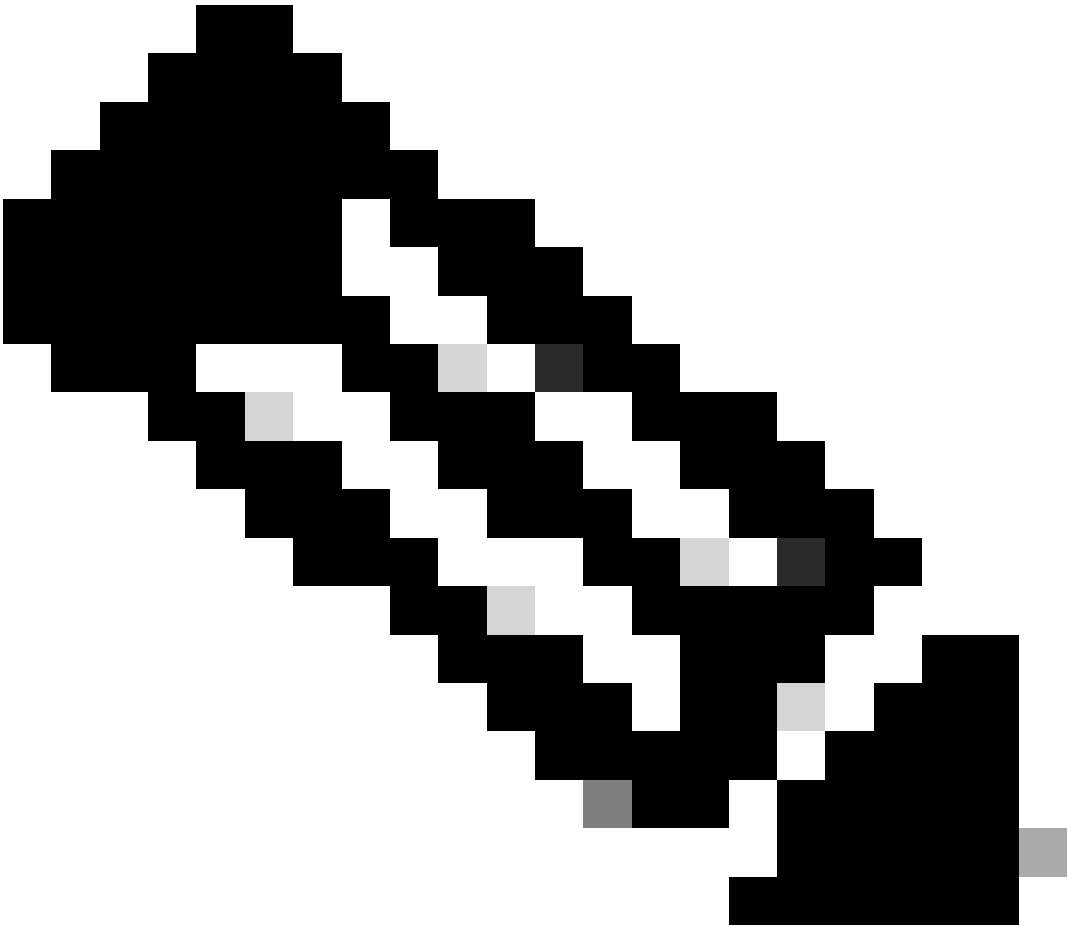
Response body

```
{
  "error": {
    "category": "FRAMEWORK",
    "messages": [
      {
        "description": "The value is mandatory as override is not enabled ."
      }
    ],
    "severity": "ERROR"
  }
}
```

由于缺少值属性，JSON响应失败

Code	Details
422	Error: 422 Response body <pre>{ "error": { "category": "OTHER", "messages": [{ "description": "Unprocessable Entity - Unrecognized Field 'items' in request payload" }], "severity": "ERROR" }}</pre>

以错误格式创建对象的JSON响应失败



注意：在本示例中，使用GET方法获取的JSON负载中缺少Value属性，展开的参数设置为

默认值。在JSON负载中可以有用对象创建的强制属性，这些属性是大多数网络对象(主机、网络、范围、服务和FQDN)的类型、名称和值。使用GET方法的JSON负载创建对象时，请仅使用对象相关信息(在items属性内)并删除id属性，这样就不会出现UUID冲突。

创建主机对象

单个主机对象

导航到对象(Object)> /api/fmc_config/v1/domain/{domainUUID}/object/hosts > POST，然后点击Try it Out。

用于推送单个主机对象的JSON格式：

```
{
  "name": "Obj_API_100.1.1.1",
  "type": "Host",
  "value": "100.1.1.1",
  "description": "Host Object Pushed via API"
}
```

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

- Curl:** A cURL command is shown for a POST request to `/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/hosts` with a JSON body.
- Request URL:** The URL `https://.../api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/hosts` is entered and highlighted with a red box.
- Server response:**
 - Code:** The status code `201` is shown and highlighted with a red box.
 - Response body:** The JSON response is displayed, with a red box highlighting the object details: `{ "type": "Host", "value": "100.1.1.1", "overridable": false, "description": "Host Object Pushed via API", "name": "Obj_API_100.1.1.1", "id": "00000000-0000-0ed3-0000-034359741413", ... }`

创建主机对象的POST操作



注意：对单个主机对象使用POST操作时，请使用默认的“批量参数”。

批量主机对象

导航到对象(Object)> /api/fmc_config/v1/domain/{domainUUID}/object/hosts > POST，然后点击Try it Out。

用于推送批量主机对象的JSON格式：

```
[
  {
    "name": "Obj_API_172.16.0.1",
    "type": "Host",
    "value": "172.16.0.1",
    "description": "Bulk API Operation"
  },
  {
    "name": "Obj_API_172.16.0.2",
```

```

"type": "Host",
"value": "172.16.0.2",
"description": "Bulk API Operation"
}
]

```

Name	Description
bulk boolean (query)	Enables bulk create for host objects. true
domainUUID * required string (path)	Domain UUID e276abec-e0f2-11e3-8169-6d9ed49b625f

Request body required application/json

Input representation of host object.

Examples:
[Modified value]

```

[
  {
    "name": "Obj_API_172.16.0.1",
    "type": "Host",
    "value": "172.16.0.1",
    "description": "Bulk API Operation"
  },
  {
    "name": "Obj_API_172.16.0.2",
    "type": "Host",
    "value": "172.16.0.2",
    "description": "Bulk API Operation"
  }
]

```

Execute Clear

创建批量对象的JSON请求示例

```

curl -X 'POST' \
  'https://[redacted]/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/hosts?bulk=true' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -H 'X-auth-access-token: 56a398c4-8791-44f0-aaba-aab150f9e781' \
  -d '[
    {
      "name": "Obj_API_172.16.0.1",
      "type": "Host",
      "value": "172.16.0.1",
      "description": "Bulk API Operation"
    },
    {
      "name": "Obj_API_172.16.0.2",
      "type": "Host",
      "value": "172.16.0.2",
      "description": "Bulk API Operation"
    }
  ]'

```

Request URL
[https://\[redacted\]/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/hosts?bulk=true](https://[redacted]/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/hosts?bulk=true)

Server response

Code
201

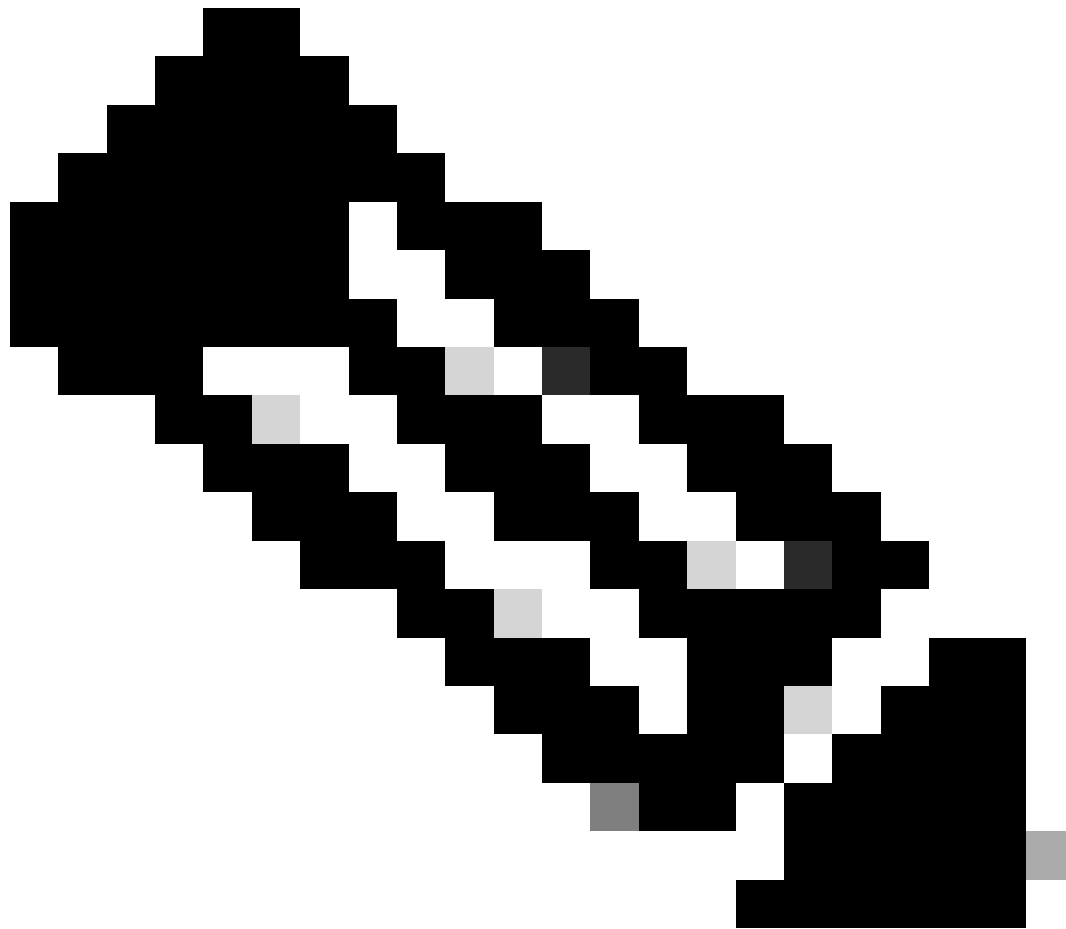
Details

Response body

```

{
  "links": {
    "self": "https://[redacted]/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/hosts?bulk=true"
  },
  "items": [
    {
      "links": {
        "self": "https://[redacted]/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/hosts/00000000-0000-0ed3-0000-034359741432",
        "parent": "https://[redacted]/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/networkaddresses"
      },
      "type": "Host",
      "value": "172.16.0.1",
      "overridable": false,
      "description": "Bulk API Operation",
      "name": "Obj_API_172.16.0.1",
      "id": "00000000-0000-0ed3-0000-034359741432",
      "metadata": {
        "timestamp": 0,
        "lastUser": {
          "name": "api"
        },
        "domain": {
          "name": "Global",
          "id": "e276abec-e0f2-11e3-8169-6d9ed49b625f",
          "type": "Domain"
        }
      }
    }
  ]
}

```



单个网络对象

用于推送单个网络对象的JSON格式：

```
{
  "name": "Obj_API_network",
```

```

"value": "7.0.0.0/24",
"overridable": false,
"description": "Single Network object created",
"type": "Network"
}

```

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

- Name:** bulk
- Description:** Enables bulk create for network objects.
- domainUUID:** e276abec-e0f2-11e3-8169-6d9ed49b625f
- Request body:** application/json
- Request body content:**

```

{
  "name": "Obj_API_network",
  "value": "7.0.0.0/24",
  "overridable": false,
  "description": "Single Network object created",
  "type": "Network"
}

```

创建单个网络对象的JSON请求示例

批量网络对象

导航到对象(Object)> /api/fmc_config/v1/domain/{domainUUID}/object/networks > POST，然后点击Try it Out。

用于推送批量网络对象的JSON格式：

```

[
  {
    "name": "Obj_API_net1",
    "value": "1.0.0.0/24",
    "overridable": false,
    "description": "Bulk Operation",
    "type": "Network"
  },
  {
    "name": "Obj_API_net2",
    "value": "1.1.0.0/24",
    "overridable": false,
    "description": "Bulk Operation",
    "type": "Network"
  }
]

```

]

Name	Description
bulk boolean (query)	Enables bulk create for network objects. true
domainUUID * required string (path)	Domain UUID e276abec-e0f2-11e3-8169-6d9ed49b625f

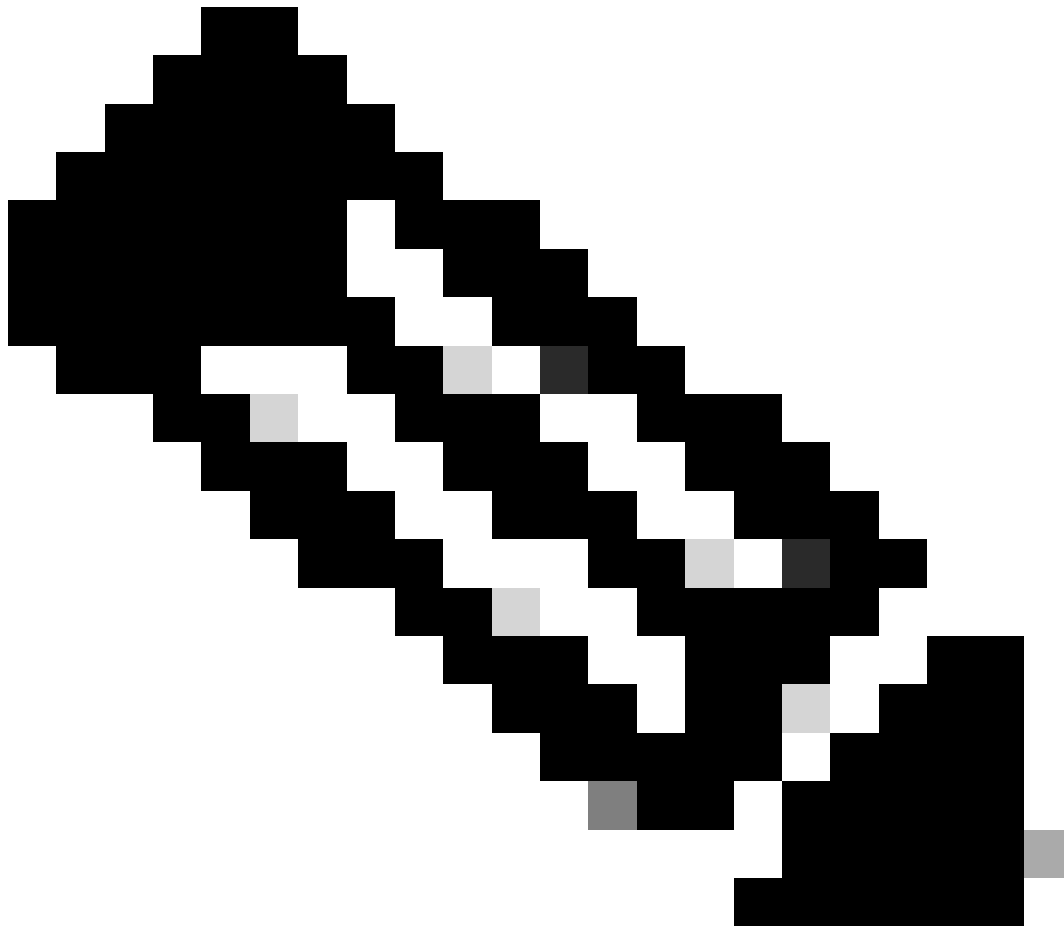
Request body required

The input network object model.

Examples:

```
[
  {
    "name": "Obj_API_net1",
    "value": "1.0.0/24",
    "overridable": false,
    "description": "Bulk Operation",
    "type": "Network"
  },
  {
    "name": "Obj_API_net2",
    "value": "1.1.0.0/24",
    "overridable": false,
    "description": "Bulk Operation",
    "type": "Network"
  }
]
```

创建批量网络对象的JSON请求示例



注意：创建批量网络对象时，选择批量参数作为真。在单个Bulk Push中，最多可使用POST方法创建1000个条目。

创建网络范围对象

单个网络范围对象

导航到对象(Object)> /api/fmc_config/v1/domain/{domainUUID}/object/ranges > POST，然后点击 Try it Out。

用于推送单个范围对象的JSON格式：

```
{
  "name": "Obj_API_TestRange",
  "value": "10.2.30.40-10.2.30.50",
  "type": "Range",
  "description": "Create Single Range Object"
```

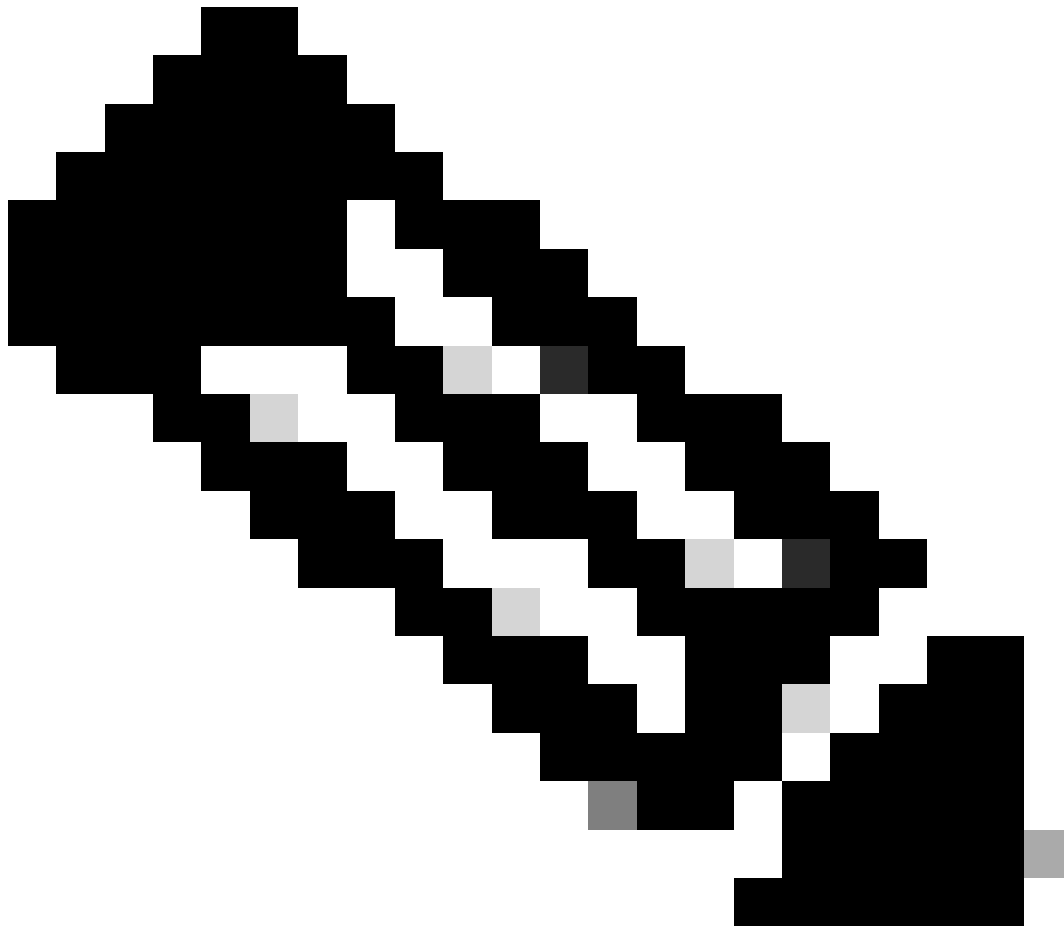
```
}
```

批量网络范围对象

导航到对象(Object)> /api/fmc_config/v1/domain/{domainUUID}/object/ranges > POST , 然后点击 Try it Out。

用于推送批量范围对象的JSON格式：

```
[
  {
    "name": "Obj_API_TestRange1",
    "value": "10.4.30.40-10.4.30.50",
    "type": "Range",
    "description": "Bulk Operation"
  },
  {
    "name": "Obj_API_TestRange2",
    "value": "10.5.30.40-10.5.30.50",
    "type": "Range",
    "description": "Bulk Operation"
  }
]
```



注意：创建批量范围对象时，选择批量参数作为真。在单个Bulk Push中，最多可使用POST方法创建1000个条目。

创建FQDN对象

单个FQDN对象

导航到Object > /api/fmc_config/v1/domain/{domainUUID}/object/fqdns > POST，然后点击Try it Out。

用于推送单个网络对象的JSON格式：

```
{
  "name": "TestFQDN",
  "type": "FQDN",
  "value": "cloud.cisco.com",
  "dnsResolution": "IPV4_ONLY",
```

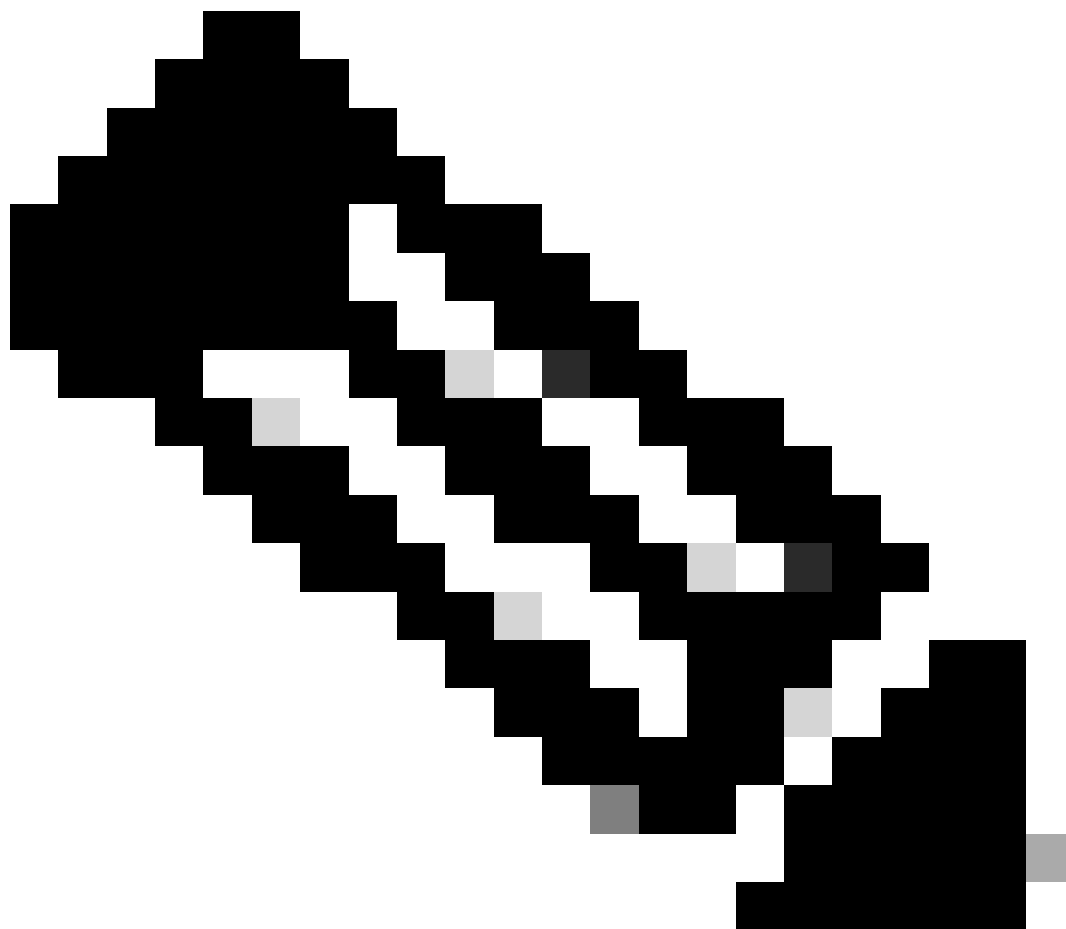
```
"description": "Create Single FQDN Object"
}
```

批量FQDN对象

导航到Object > /api/fmc_config/v1/domain/{domainUUID}/object/fqdns > POST，然后点击Try it Out。

用于推送批量网络对象的JSON格式：

```
[
  {
    "name": "Obj_API_FQDN_1",
    "type": "FQDN",
    "value": "downloads.cisco.com",
    "dnsResolution": "IPV4_ONLY",
    "description": "Bulk Operation"
  },
  {
    "name": "Obj_API_FQDN_2",
    "type": "FQDN",
    "value": "support.cisco.com",
    "dnsResolution": "IPV4_ONLY",
    "description": "Bulk Operation"
  }
]
```



注意:创建批量FQDN对象时，选择批量参数作为真。在单个Bulk Push中，最多可使用POST方法创建1000个条目。

创建服务对象

单个服务对象

导航到对象(Object)> /api/fmc_config/v1/domain/{domainUUID}/object/protocolportobjects > POST，然后点击Try it Out。

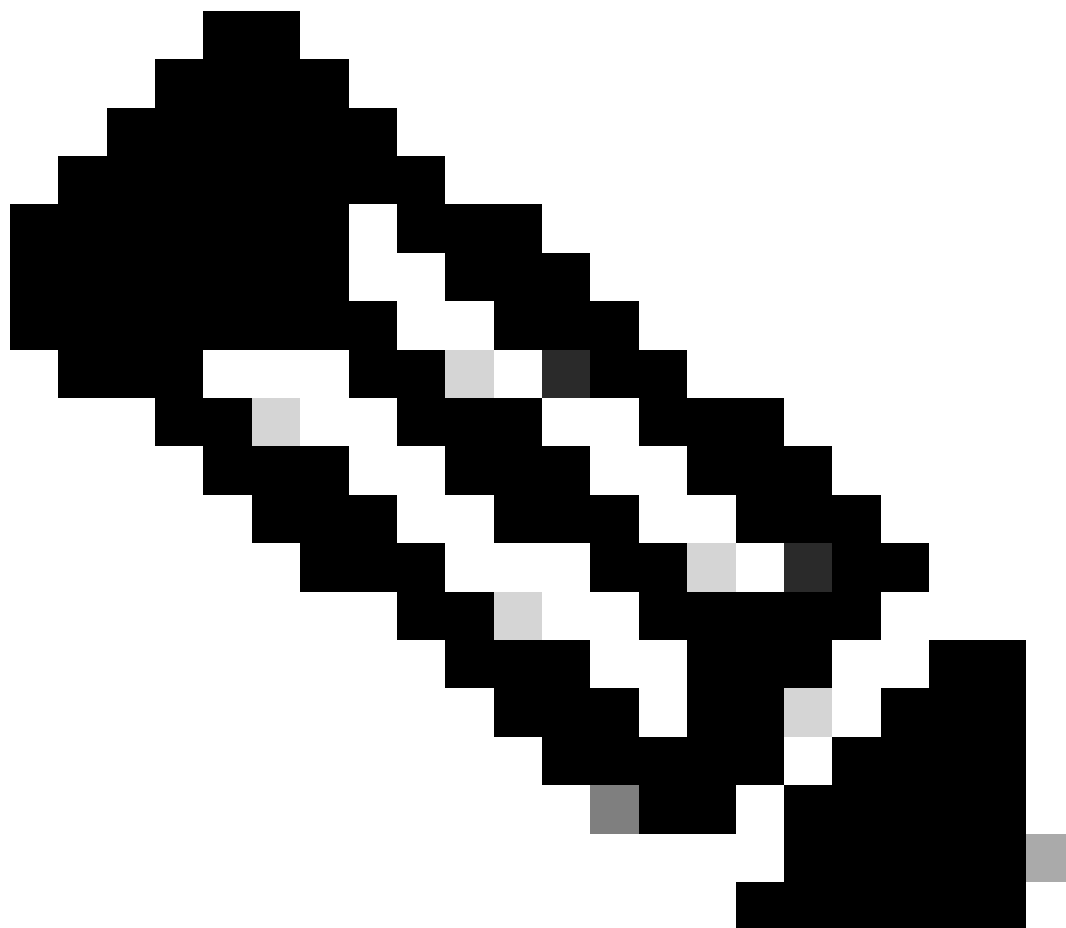
用于推送单个服务对象的JSON格式：

```
{
  "name": "Obj_API_Telnet_Port",
  "protocol": "TCP",
  "port": 23,
  "type": "ProtocolPortObject"
```

```
}
```

用于推送单个服务对象的JSON格式：

```
{  
  "name": "Obj_API_obj1",  
  "protocol": "UDP",  
  "port": "1025-65535",  
  "type": "ProtocolPortObject"  
}
```



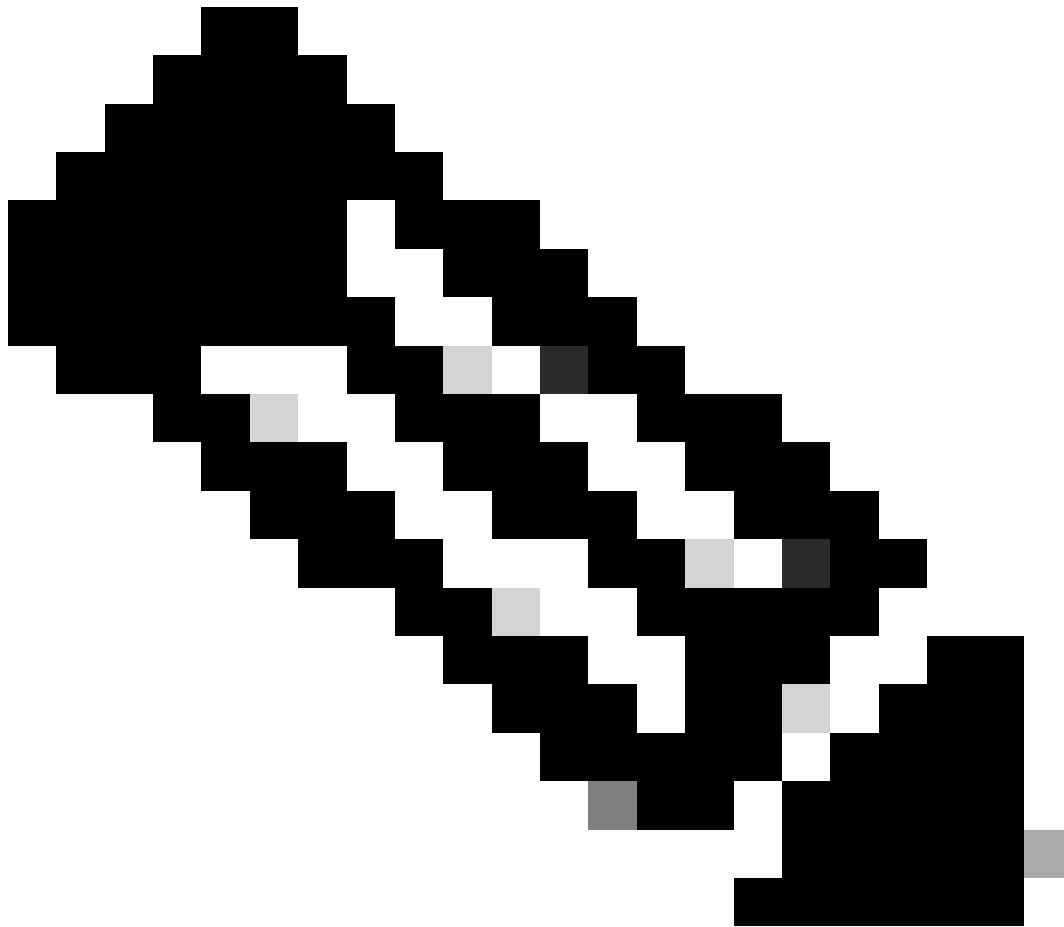
注意：当您以JSON格式定义端口范围时，请使用双引号，例如“port”：“1111-1115”，否则可能会遇到错误：“说明”:“无法处理的实体 — 意外字符('' (代码45)):需要逗号来分隔行中的对象条目：4，列：17”

批量服务对象

导航到对象(Object)> /api/fmc_config/v1/domain/{domainUUID}/object/protocolportobjects > POST，然后点击Try it Out。

用于推送批量服务对象的JSON格式：

```
[
  {
    "name": "Obj_API_UDP_Port",
    "protocol": "UDP",
    "port": 123,
    "type": "ProtocolPortObject"
  },
  {
    "name": "Obj_API_TCP_Range",
    "protocol": "TCP",
    "port": "1025-65535",
    "type": "ProtocolPortObject"
  }
]
```



注意：创建批量服务对象时，选择批量参数作为True。在单个Bulk Push中，最多可使用POST方法创建1000个条目。

故障排除

- 有关HTTP状态代码([REST API基础知识](#))，[请参阅文档](#)
- 用户凭证不能同时用于REST API和GUI界面，如果同时用于这两种界面，用户将在无警告的情况下注销。
- 所有REST请求都会记录到FMC上的这两个日志文件中。搜索您的URL(例如../object/hosts)和正确的操作(如果您正在查找GET操作错误，请确保日志启动类似于GET的操作.....对象/主机)

```
tail -f /var/opt/CSC0px/MDC/tomcat/logs/stdout.logs  
tail -f /var/opt/CSC0px/MDC/log/operation/usmsharedsvcs.log
```

关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。