

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bmRvdCI6IjUwZyZyZWV1b2Y2



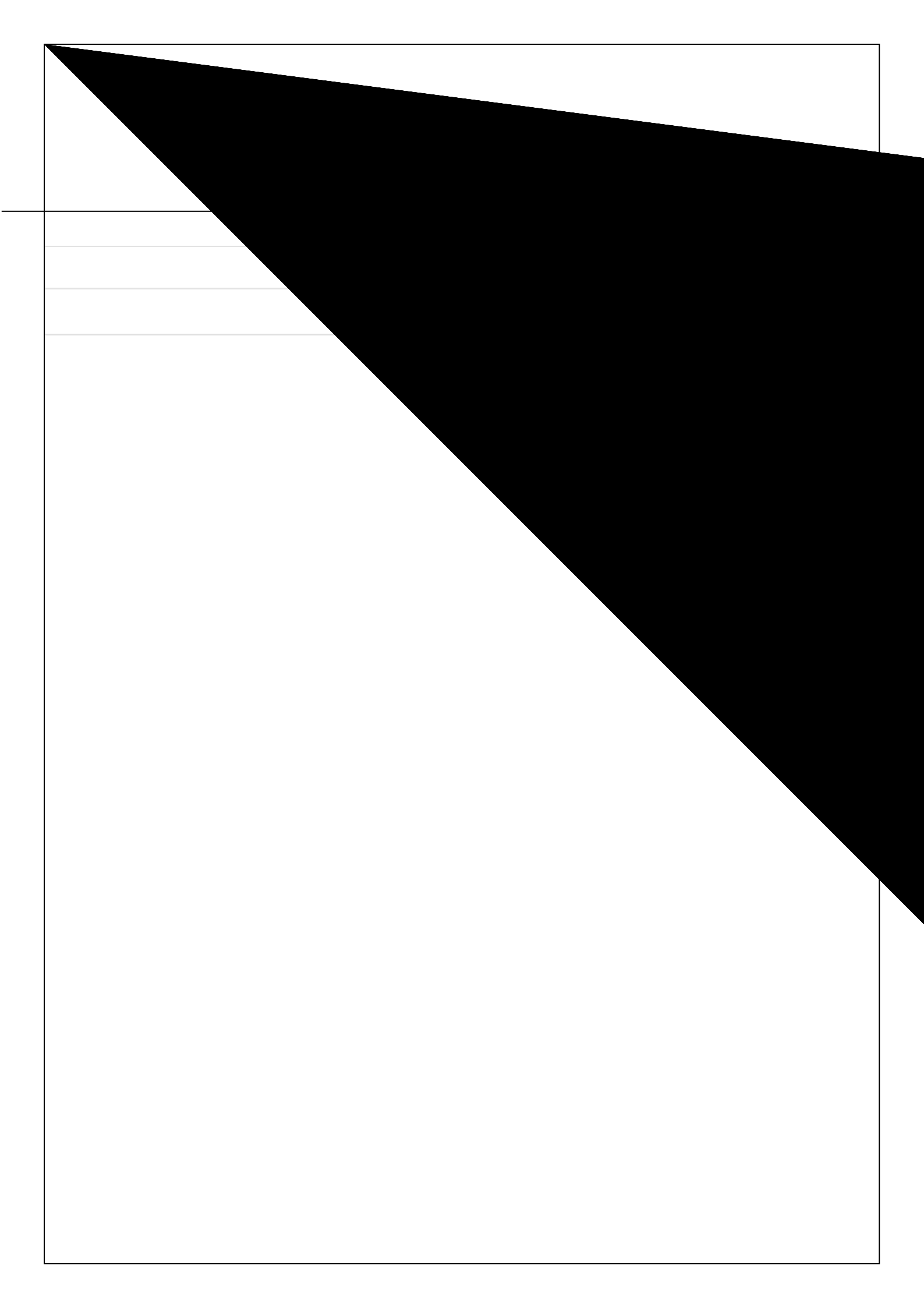
Authorize - Logout

Service: Identity Gateway

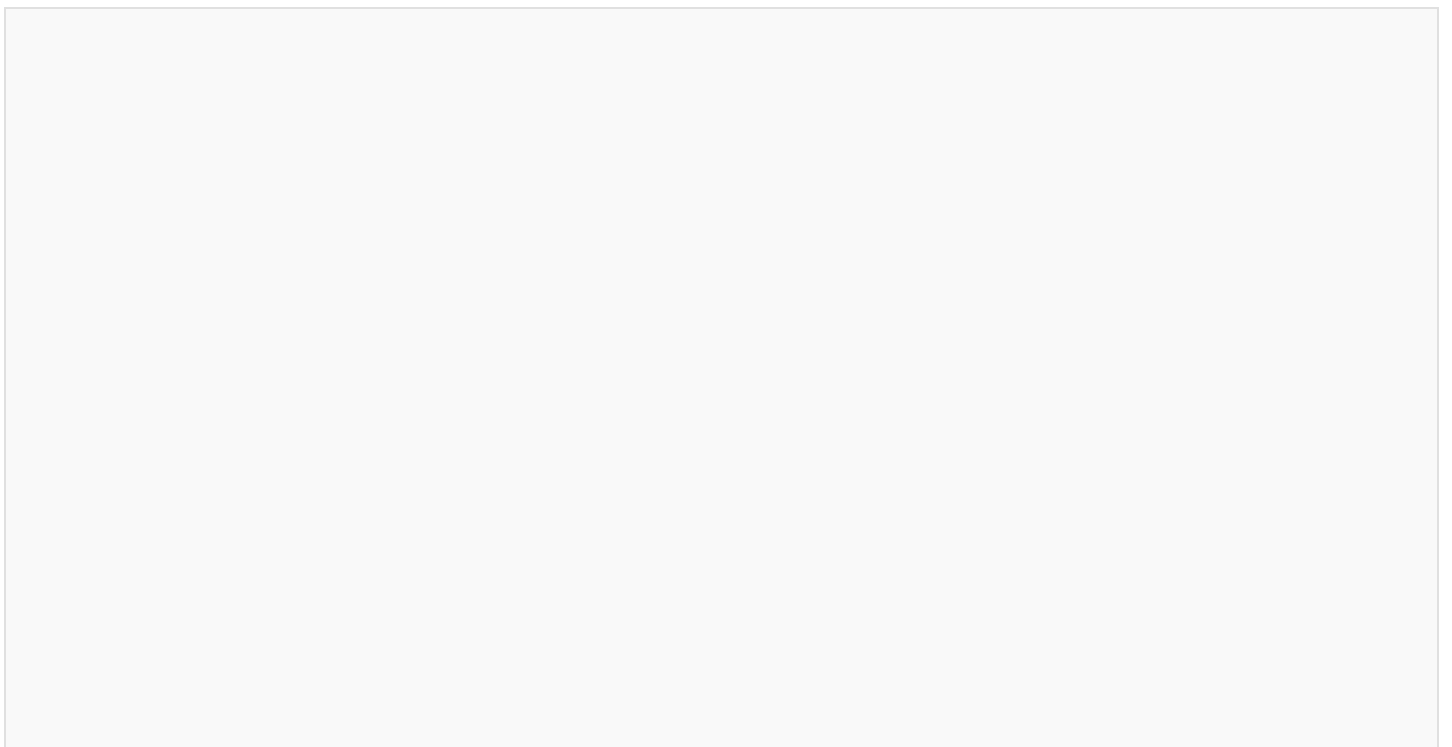
Returns to the redirect url passed on successful logout

Http Request

GET https://t-t-m- -°




```
{
  "items": [
    {
      "id": "string",
      "di sps
```

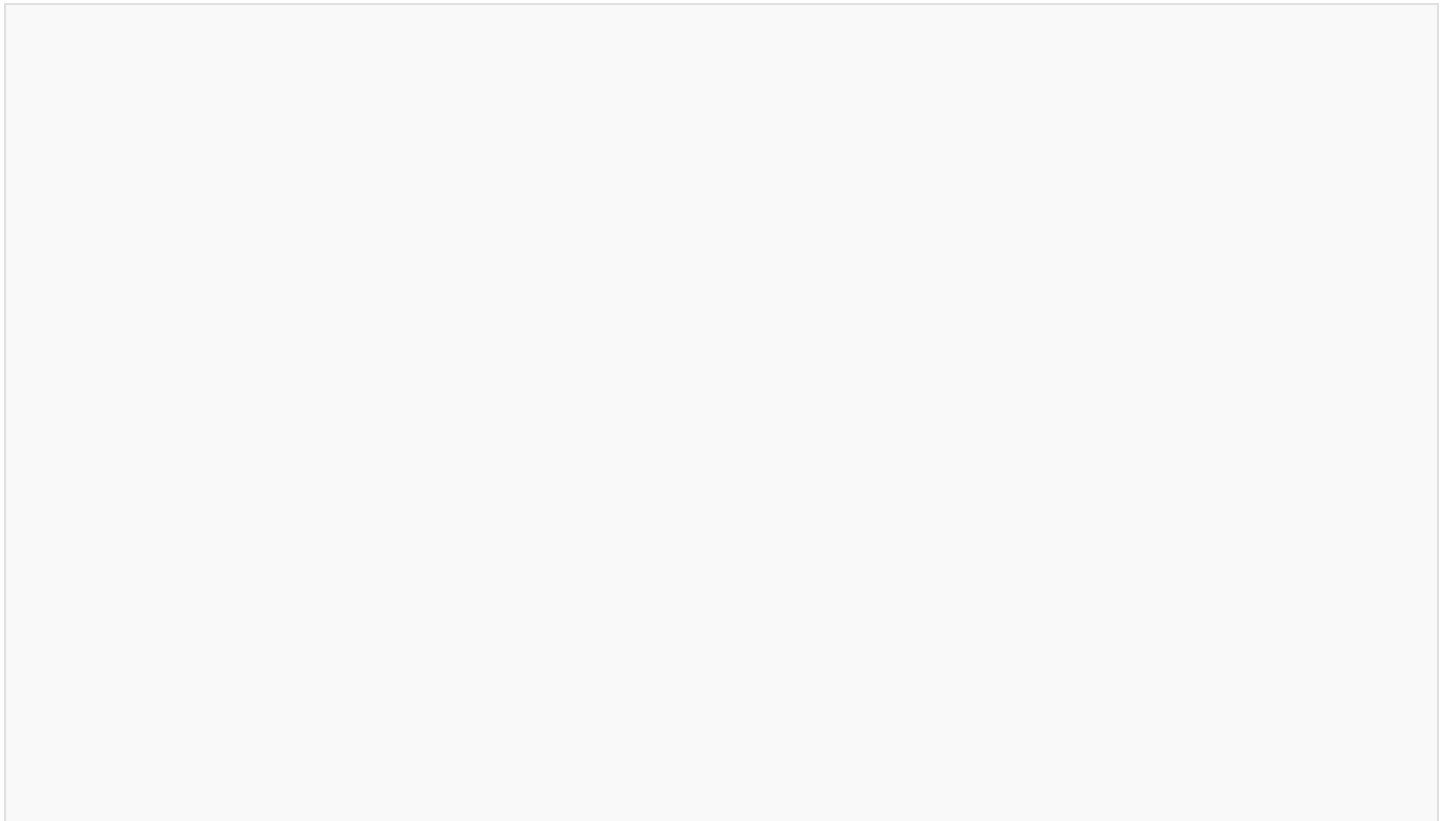


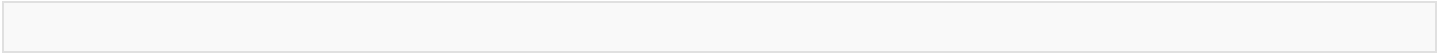
"Value": "C "


```
{
  "id": "string",
  "displayName": "string",
  "conditions": [
    {
      "key": "string",
      "operator": Q,
      "value": {}
    }
  ],
  "supportedMethod": o
```



```
{  
  "id": "string",  
  "di sp
```

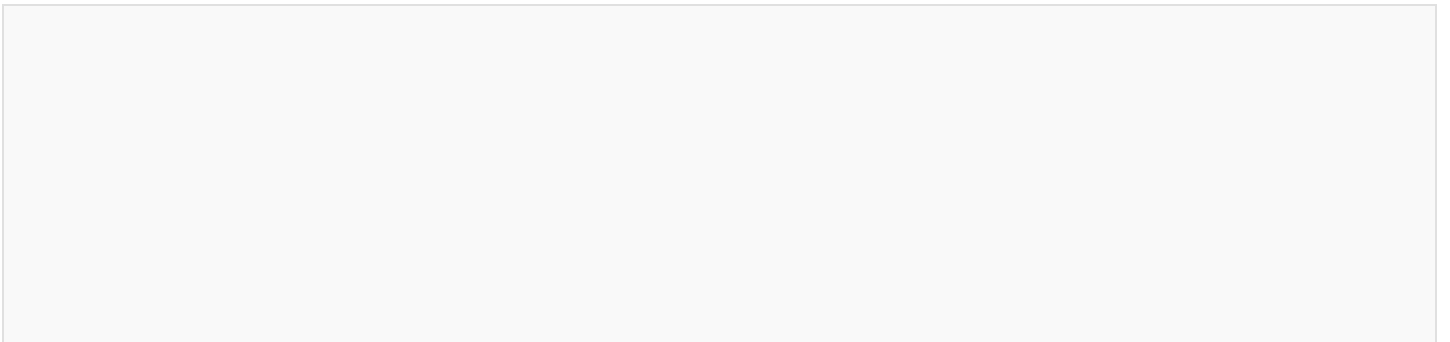


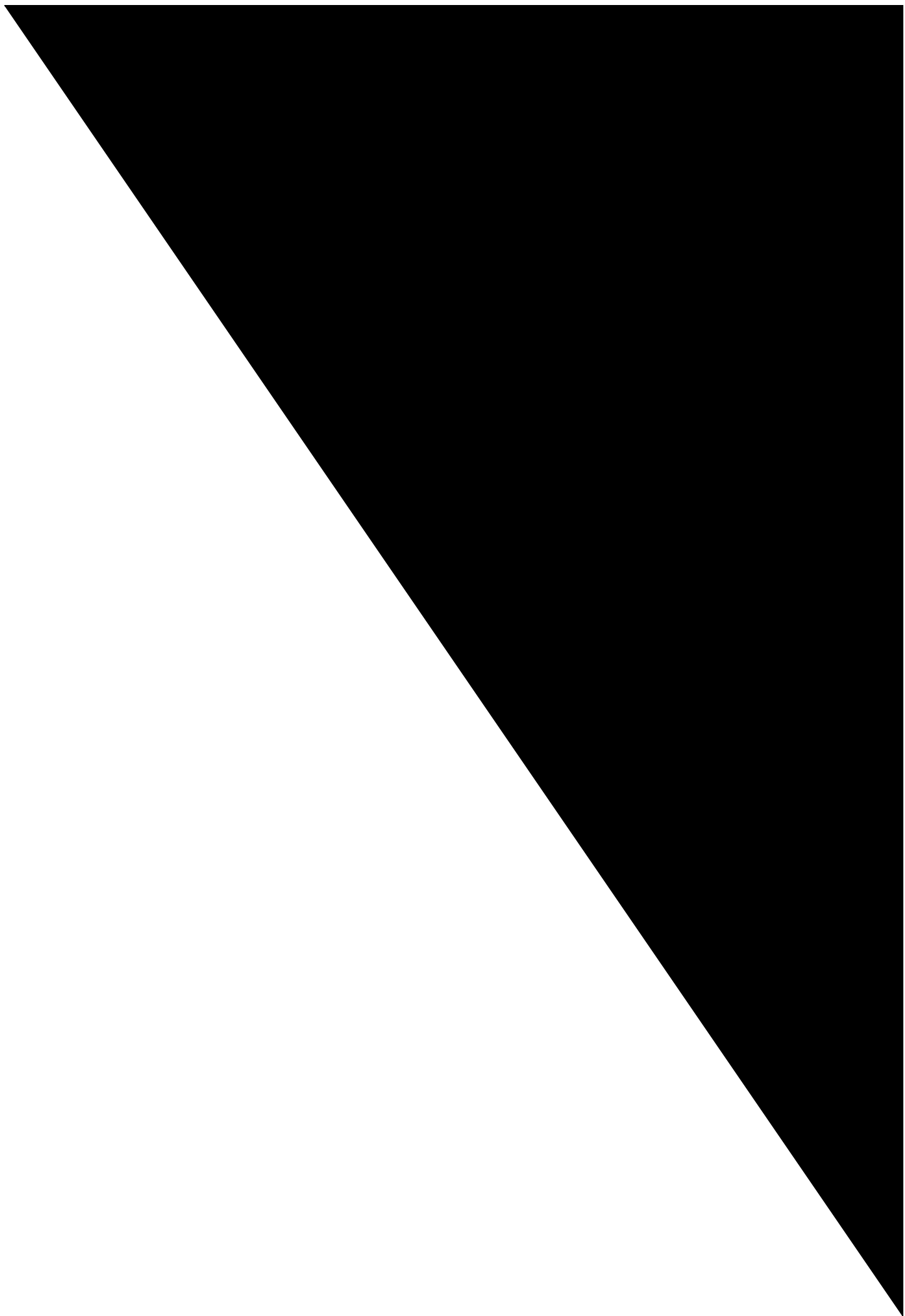



```
{  
  "jsObject": {  
    "content": {
```



```
{
  "jsObject": {
    "content": {
      "deviceContent": {
        "p
```






```
public Task<AlarmListByRuleApiModel> PostAsync(string id, QueryApiModel body)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	id	
QueryApiModel	body	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<AlarmListByRuleApiModel>	

--

--

TYPE	NAME	DESCRIPTION
System.String	id	
RuleApiModel	rule	

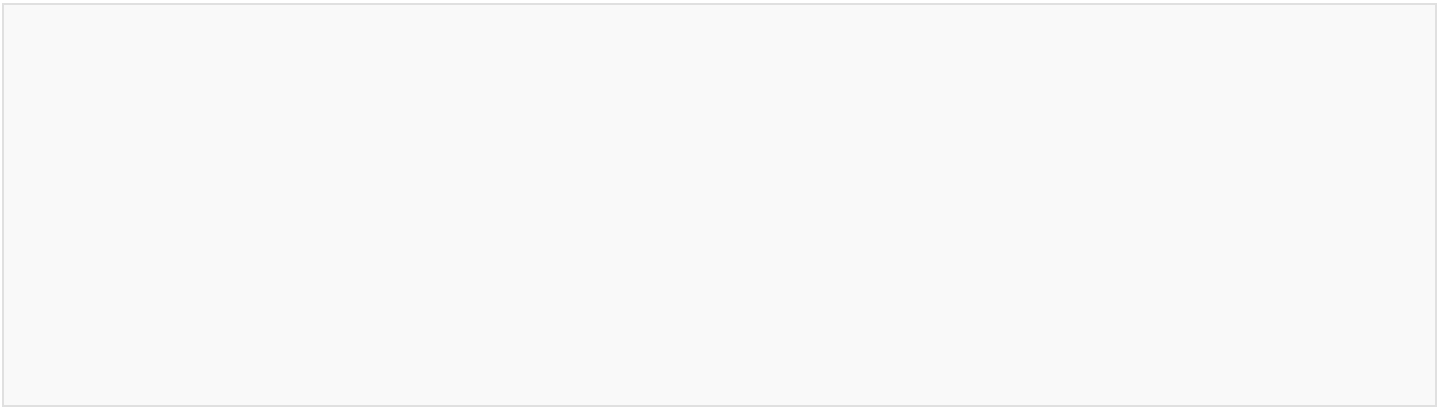
[illegible]

```
{
  "deviceOpenId": "string",
  "name": "string",
  "createTimeUtc": "2020-10-29T15:57:07.672Z",
  "deviceGroup": "string",
  "deviceGroupName": "string",
  "deviceGroup": "2020-10-29T15:57:07.672Z"
}
```



```
{
  "DeviceGroupID": "2f411a20-7632-45f0-bf43-5ae449b8727a",
  "DeviceGroupName": "TestDeviceGroup",
  "DeviceGroupQuery": "[{\"key\": \"Tags. TestDevice\", \"operator\": \"EQ\", \"value\": \"Yes\"}]",
  "Name": "TestDeviceGroupDeployment",
  "PackageID": "c5e6e9ab-1567-4f99-aa5b-  °
```

```
{
  r  "DeviceGroupID": "34b35ac5-837d-4fa4-9aE 7s     ê3
```



[The following text is a dense, continuous block of illegible characters and symbols, likely representing a corrupted scan of a document page. It contains no discernible words or structure.]


```
} E "$netadP t I L  
"$netad  
yLi "$netadP: {er  
"$netadP: {er  
"$type": "Devi cePropertyLi st; 1",  
"$urIP 1",
```

NOTE THE

"addi ti on P on P on ti on ti

```
{  
  "$net" "$net"
```

Response

```
{
  "$metadata": {
    "$type": "DeviceList;1",
    "$uri": "/v1/devices"
  },
  "ContinuationToken": null,
  "Items": [
    {
      "EOL": true
    }
  ]
}
```

```
{  
  "metadaaa"
```



```
{  
  "metadata": {  
    "addi ti on
```

```
{
  "$metadata": {
    "$type": "DeviceList;1",
    "$uri": "/v1/devices"
  },
  "ContinuationToken": null,
  "Items": [
    {
      "Reported": {},
      "Id": "r " " }
    ]
  },
  "Reported": {
    "r " " " }
  },
  "Id": "r " " " }
}
```

```
{
  "$metadata": {
    "$type": "DeviceList;1",
    "$uri": "/v1/devices"
  },
  "ContinuationToken": "$metadata": {
```

```
{  
  "$metadata": {  
    "$type": "Devicelist;1"H
```



Get it let e etP ê É Í úÊe ú ê Ê

--


```
{
  "Items": [
    "Tags. TestDevice",
    "Tags. Org",
    "Properties. Reported. Protocol",
    "Properties. Reported. SupportedMethods",
    "Properties. Reported. DeviceMethodStatus",
    "Properties. Reported. FirmwareUpdateStatus",
    "Properties. Reported. firmware. currentFwVer VurrentFwVer VaReporte
```



```
{
  "metadata": {
    "additional Prop1": "string"l1": "str stitio l Prop1": "string"l1-i tio l Prp1": " =g"      e+  op1": "str
```



```
{  
  "Metadata":
```



```
{
  "metadata": {
    "additional Prop1": "string",
    "additional Prop2": "string",
    "additional Prop3": "string"
  },
  "conversionTo": "a string"
```



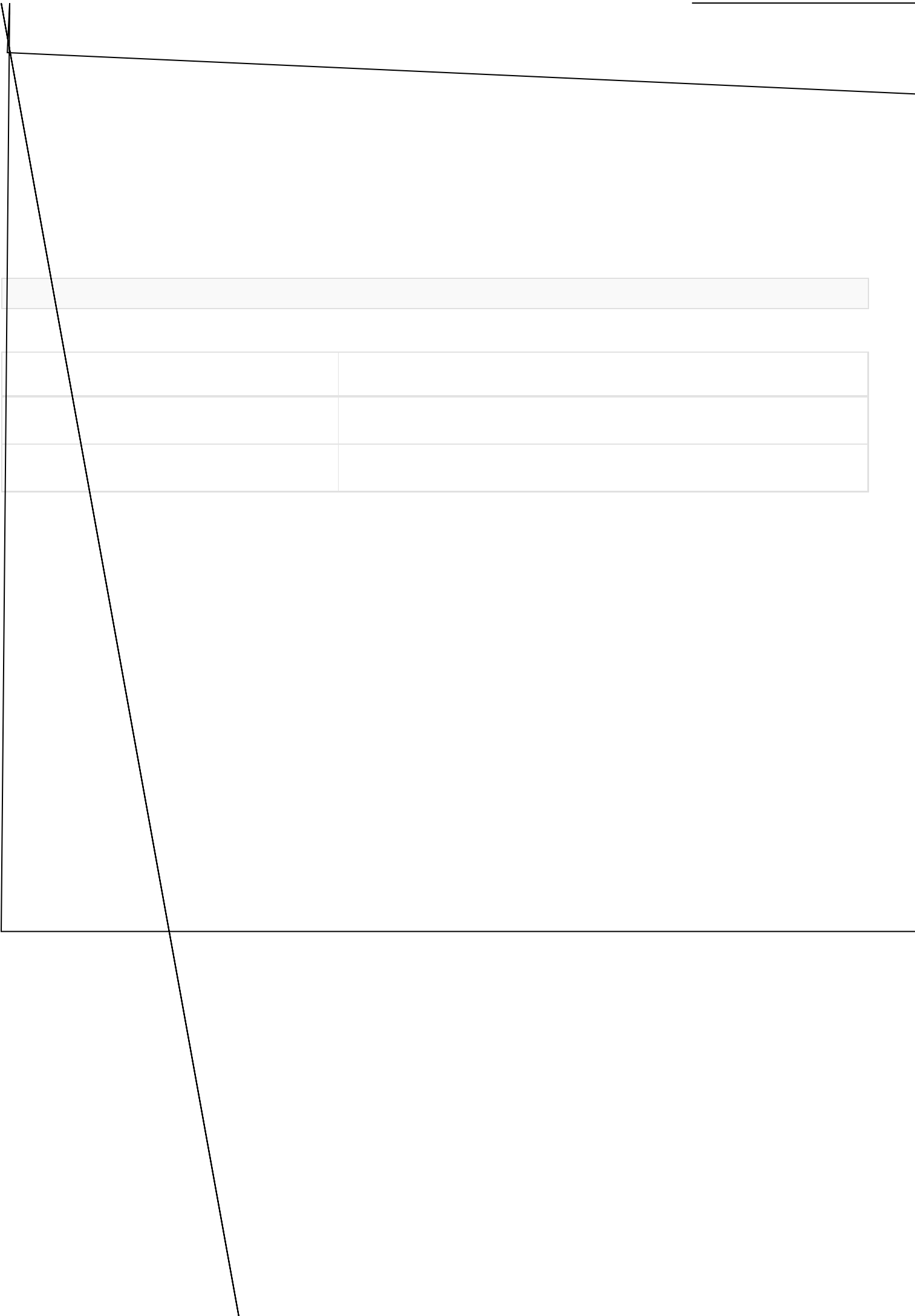
```
{
  "$metadata": {
    "$type": "DeviceList;1",
    "$uri": "/v1/d{
```



```
{  
  "ETag": "\"NTU4M1c1Njcv\\\"|AAAAAAAAAc=",  
  "Id": "TestDevice",  
  "C2 " 4M1c1Njlc
```



```
{  
  "metadata": {  
    "addi ti on
```

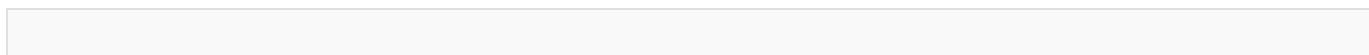



```
{
  "eTag": "string",
  "id": "string",
  "lastMessageCount": 0,
  "lastActivity": "2020-10-30T05:41:54.391Z",
  "lastStage2.type": "string",
  "enabled": true,
  "lastStage": "2020-10-30T05:41:54.391Z",
}
```

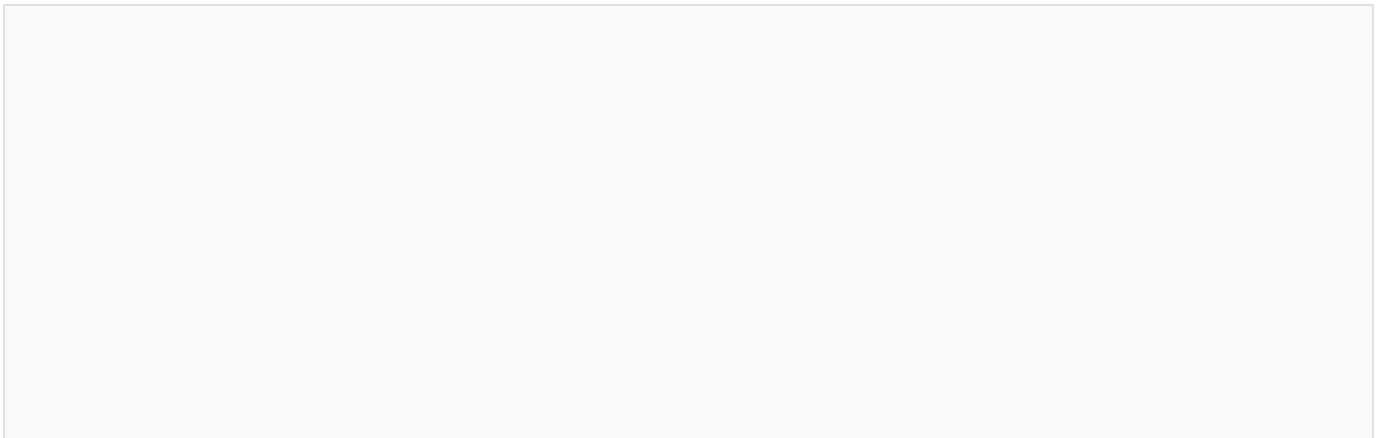
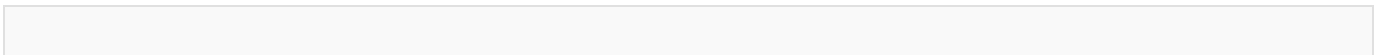
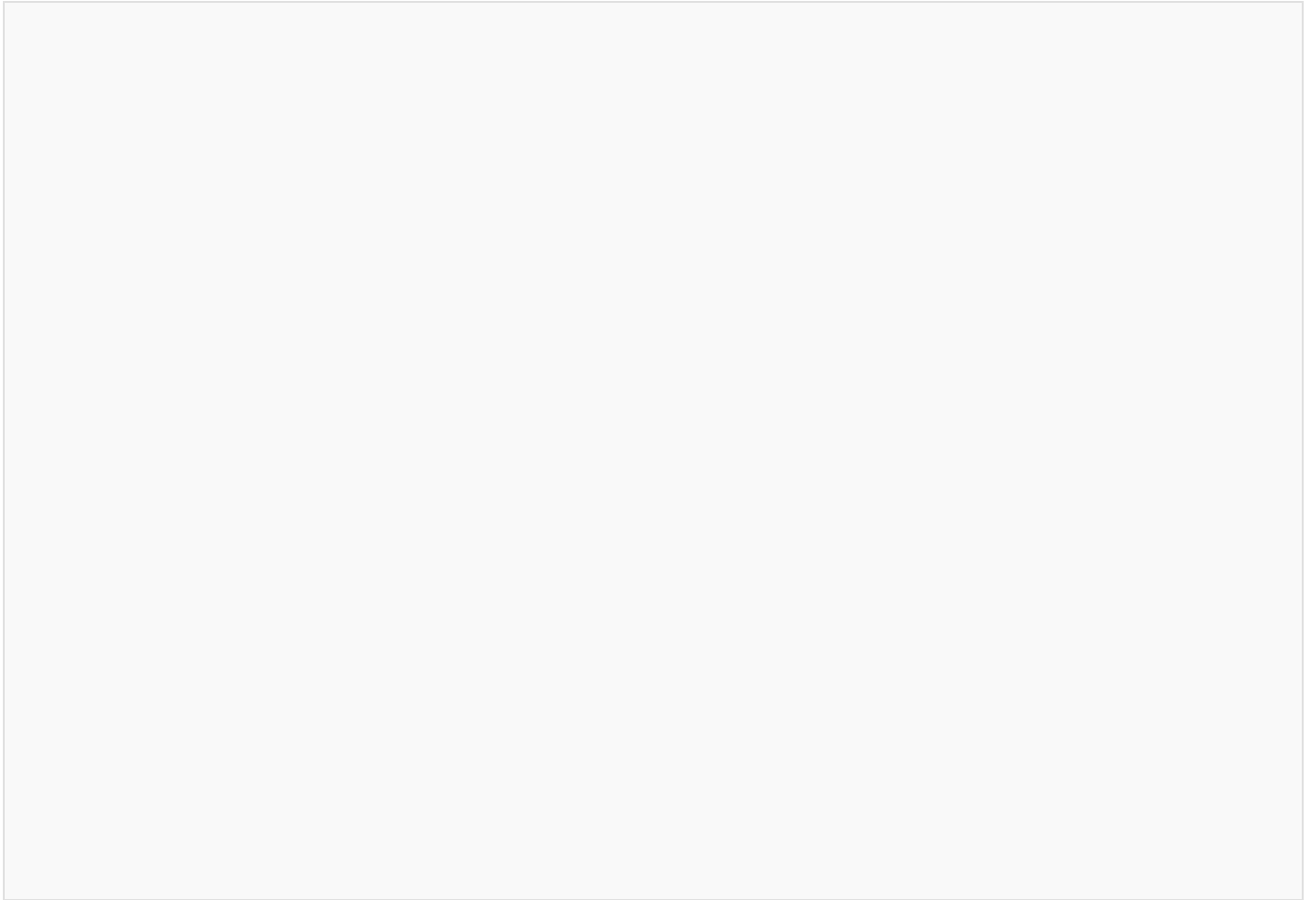
Response Schema

```
{
  "eTag": "string",
  "id": "string",
  "c2dmMessageCount": 0,
  "lastActivity": "2020-Ds"
```

```
{
  "$metadata": {
    "$type": "DeviceList;1",
    "$uri": "/v1/devices"
  },
  "ContinuationToken": null,
  "Items": [
    {
      "ETag": "AAAAAAAAAYc=|AAAAAAAAAYc=",
      "a:
```



```
{  
  "Id": "B      I      K c      K
```




```
{  
  "ETag": "\"NDE1Nj1Q3OTM\\\" | AAAAAAAAAAE=",  
  "Id": "TestDevice",  
  "C2DMessageCount": 0,  
  "LastActivity": "0001-01-01T00:00:00",  
  "Connecteut"
```



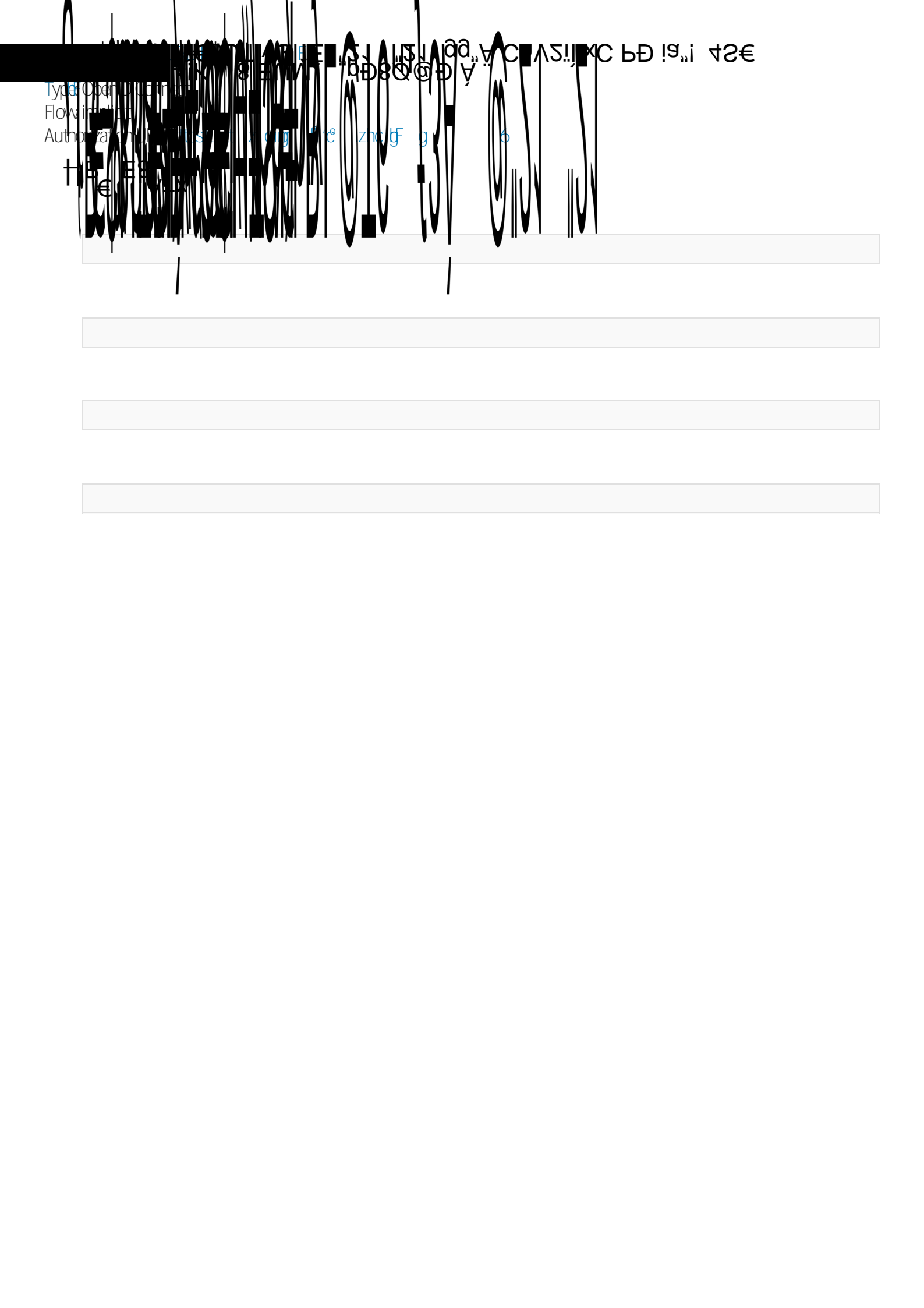
```
{
  "eTag": "string",
  "id": "string",
  "c2dmMessageCount": 0,
  "lastActivity": "2020-05-01T00:00:00.000Z"
}
```

Response Schema

```
{
  "eTag": "string",
  "id": "string",
  "c2dmMessageCount": 0,
  "lastActivity": "2020-Ds"
```



```
"tags": {  
  "addi
```

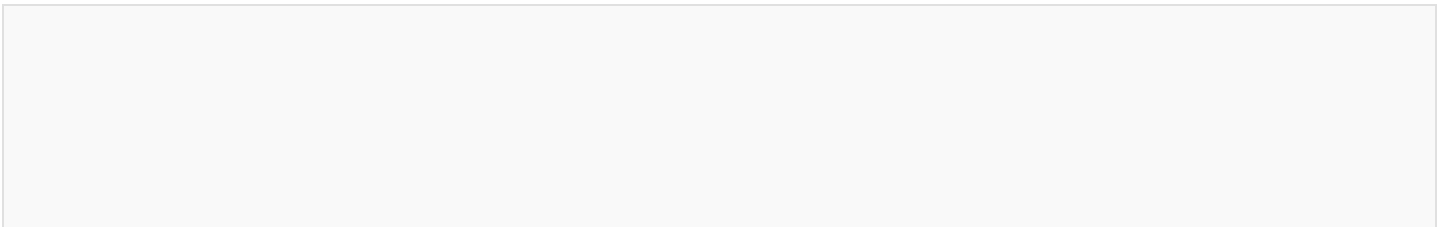



type Operating Instructions
Flowchart
Authorization

the E

[illegible]

additional Props : [



"type": 0,
"status": 0,
"netu" <

```
"createdDateUtc": "2020-10-30T10:21:05.676Z",  
"last": ">120c13d85d330395a75P2##x35"2# -x372##x342##x35"2# -x3P2##x35"2# -x3P2##x35"
```



```
{  
  "metadata": {  
    "addi ti on
```

```
{
  "$metadata": {
    "$type": "DeviceList;1",
    "$uri": "/v1/devices"
  },
  "ContinuationToken": null,
  "Reported": {
    "Reported": {},
    "DeviceList": {
      "$uri": "/v1/devices",
      "schemaVersion": "1.0",
      "id": "1.0",
      "name": "1.0",
      "description": "1.0",
      "version": "1.0",
      "author": "1.0",
      "license": "1.0",
      "url": "1.0",
      "email": "1.0",
      "phone": "1.0",
      "address": "1.0",
      "website": "1.0",
      "socialMedia": "1.0",
      "other": "1.0"
    }
  }
}
```



```
{  
  "metadata": {  
    "addi ti on
```

```
{
  "$metadata": {
    i  "$type": "DeviceList; 1",
    "$uri": "/v1/devices"
```

```
{
  "$metadata": {
    "$type": "DeviceList;1",
    "$uri": "/v1/devices"
  },
  "Cos
```



```
{
  "reported": {
    "additional Prop1": [
      "additional Prop2"
    ]
  }
}
```

Security

Type: OpenID Connect

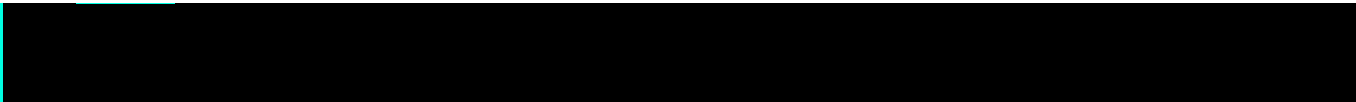
Flow: implicit

Authorization URL: <https://login.microsoftonline.com/ors-testab2con.microsoft.com>

Examples

1. Get Module Data based on DeviceId and ModuleId

```
GET /i o l w u
```



--

--

--


```
{  
  "Name": "Tenant Manager",  
  "Status": {  
    "IsHealthy": true,  
    "Message": "Alive and well!"  
  }  
}
```

