
eyJhbDciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJjbGllbnRdaWQiOiI5NWQzYzY2MiOyM2

Ý ®ùÝÍ×®¯ ·®¯ Ý þøþÝ Ý ööº ¯®×ÍñÍ®Ý


```
{
  "items": [
    {
      "id": "string",
      "disps
```

```
                "Value": "C\u0141"
```

```
{
  "id": "string",
  "displayName": "string",
  "conditions": [
    {
      "key": "string",
      "operator": 0,
      "value": {}
    }
  ],
  "supportedMeth   o
```

```
{
```

```
{
  "id": "string",
  "disp
```

TyT


```

```


```

```

```

```

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

```
{
  "jsObject": {
    "content": {
      "deviceContent": {
        "P
```


Sample RespoSo

Returns

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

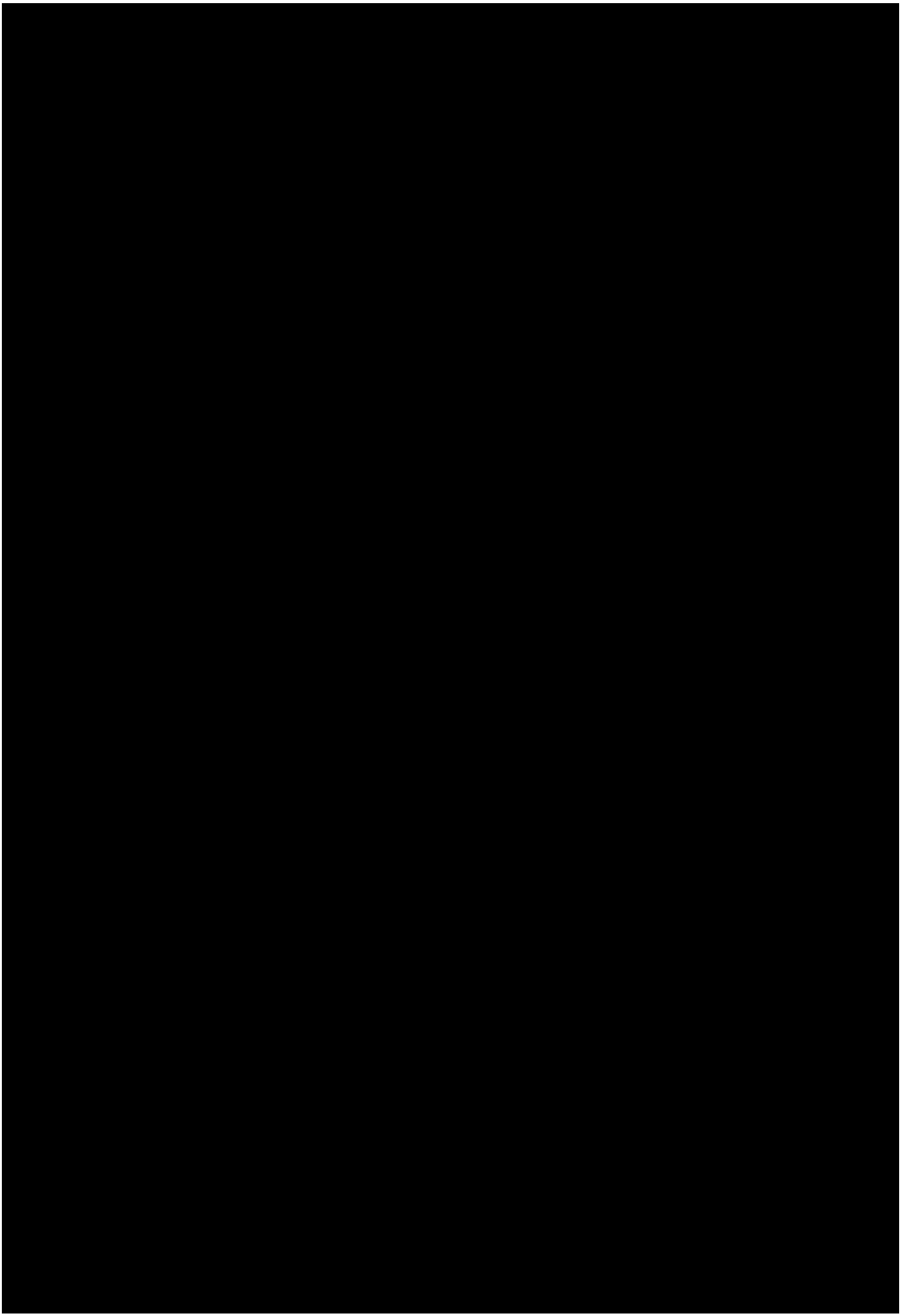
ConfigType	Type which specifies the type of configuration	Yes, If PackageType is DeviceConfiguration
DeviceIds	List of DeviceIds to be targetted by Configuration	No, Can be provided if configuration needs to targeton=needr eef eef eef t

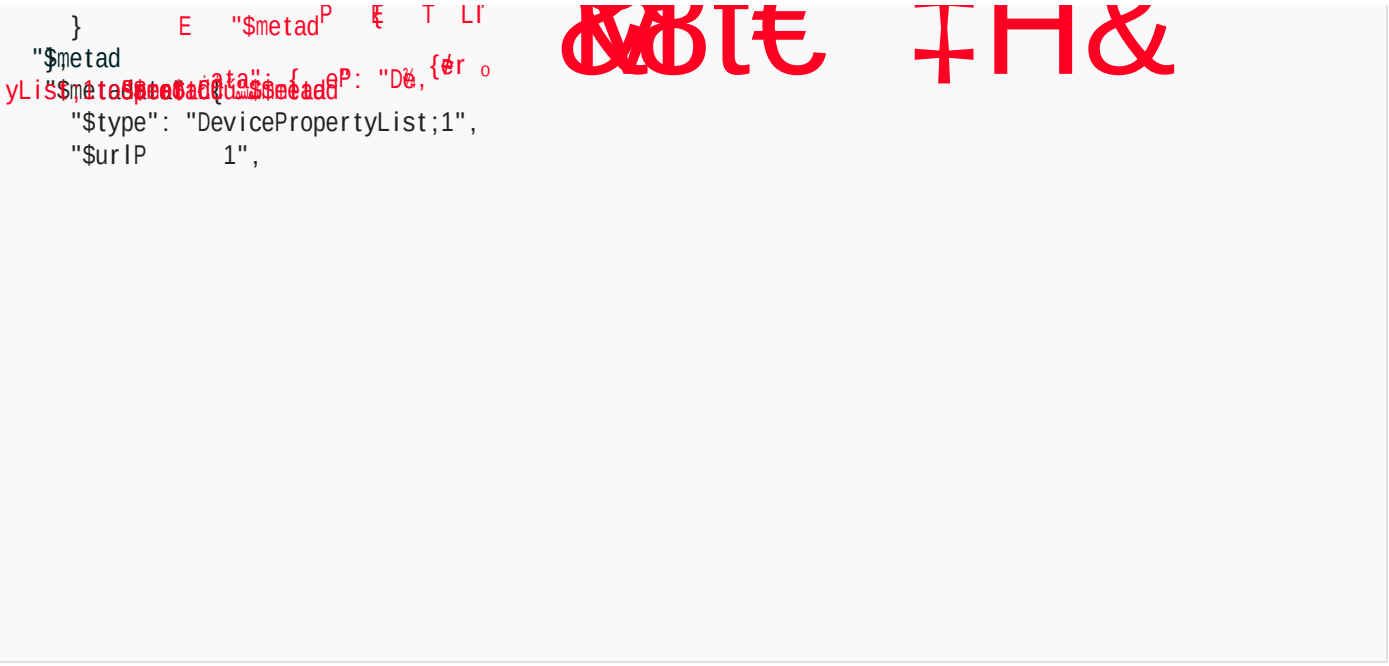
```
{
   "DeviceGroupId":"2f411a20-7632-45f0-bf43-5ae449b8727a",
   "DeviceGroupName":"TestDeviceGroup",
   "[illegible]
```

```
{
  "deviceGroupId": "string",
  "name": "string",
  "createdDateTimeUtc": "2020-10-29T15:57:07.672Z",
  "deviceGroupId": "string",
  "deviceGroupName": "string",
  "deviceGroupId": "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-aC\S6Đ Đ Ô
```

```
{
 r  "DeviceGroupId":"334b35ac5-837d-4fa4-9aE 7           Ď3
```





```
"addition ₱ion ₱ion ⱡ⅂on ⱡ⅂
```

```
{
  "$met "$met
```

```

```

```

```

Response


```
{
  "$metadata": {
    "$type": "DeviceList;1",
    "$uri": "/v1/devices"
  },
  "ContinuationToken": null,
  "Items": [
    {
   P{ "E OĊĎ< »ű    ű     Ò
```

```
{
  "$metadata
```

```
{
  "metadata": {
    "addition
```

```
{
  "$metadata": {
    "$type": "DeviceList;1",
    "$uri": "/v1/devices"
  },
  "ContinuationToken": null,
  "Items": [
    {
      "Reported": {},
  gDtd": {r ű "}ns aÖÖT: ű/"ű Đ  ¨ÄDtiné p: {rĐű {   »ű    ű     Đ
```

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

```
{
  "$metadata": {
    "$type": "DeviceList;1"H     ű       l  7
```

Get [illegible]


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

```
{
  "metadata": {
    "additionalProp1": "string"Ì: "str stitio lProp1": "string"Ì¾litio †Prp1": "  ‡g"      Ð+  op1": "str
```

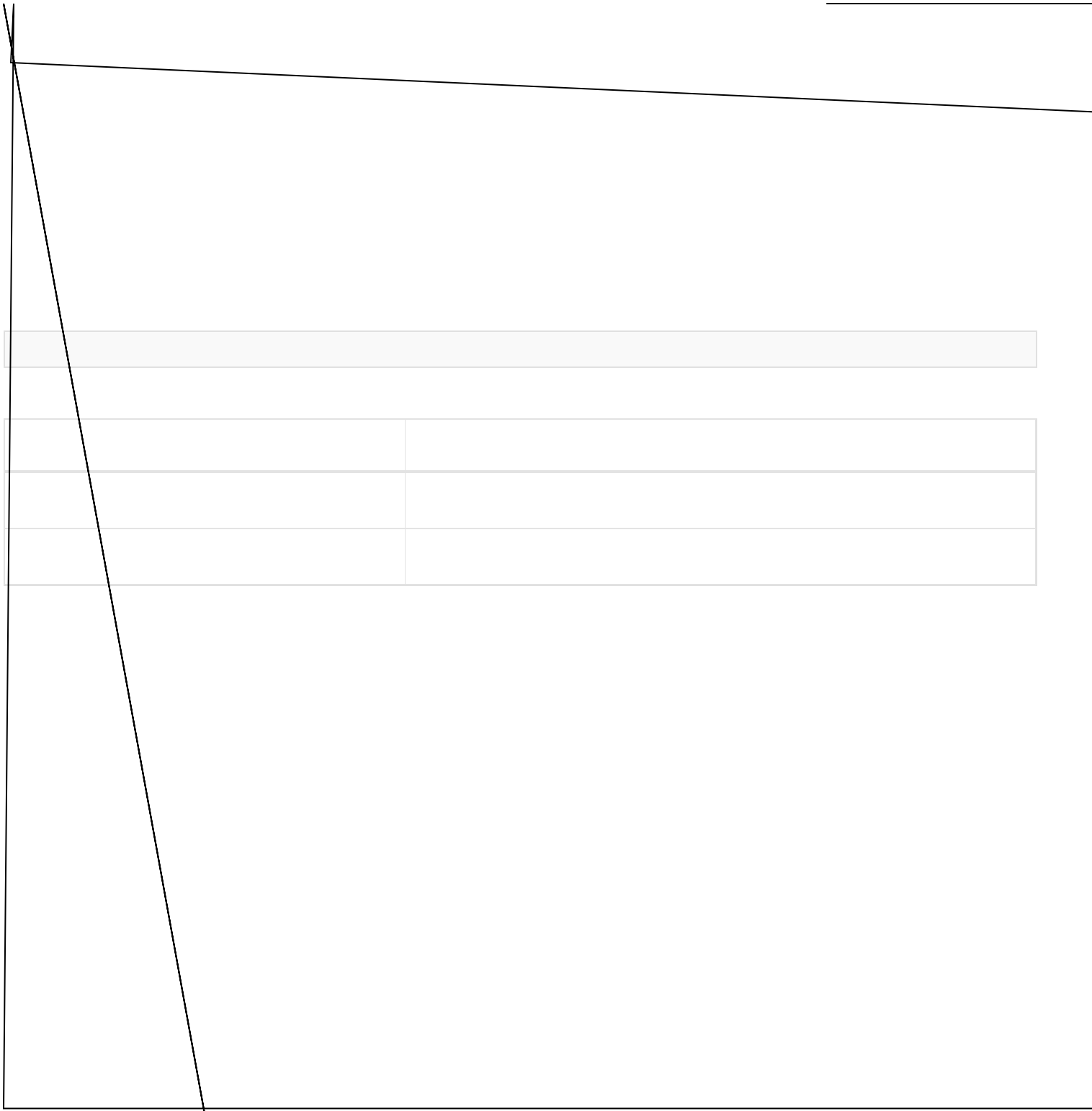
```
{
"$metadata
```

```
{
  "metadata": {
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
  },
  "continuationTo": "a string"
}
```

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

```
{
    "ETag": "\"NTU4MTc1Mjcw\"|AAAAAAAAAAc=",
    "Id": "TestDevice",
    "C2 "d4MTc1Mjc
```

```
{
  "metadata": {
    "addition
```



```
{
  "eTag": "string",
  "id": "string",
  "lastMessageCountDT": 0,
  "lastActivity": "2020-10-30T05:41:54.391Z",
  "connectedOn": "2020-10-30T05:41:54.391Z",
  "enabled": true,
  "lastSeen": "2020-10-30T05:41:54.391Z",
```

Response Schema


```
{
  "eTag": "string",
  "id": "string",
  "c2DMessageCount": 0,
  "lastActivity": "2020-Ds
```

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

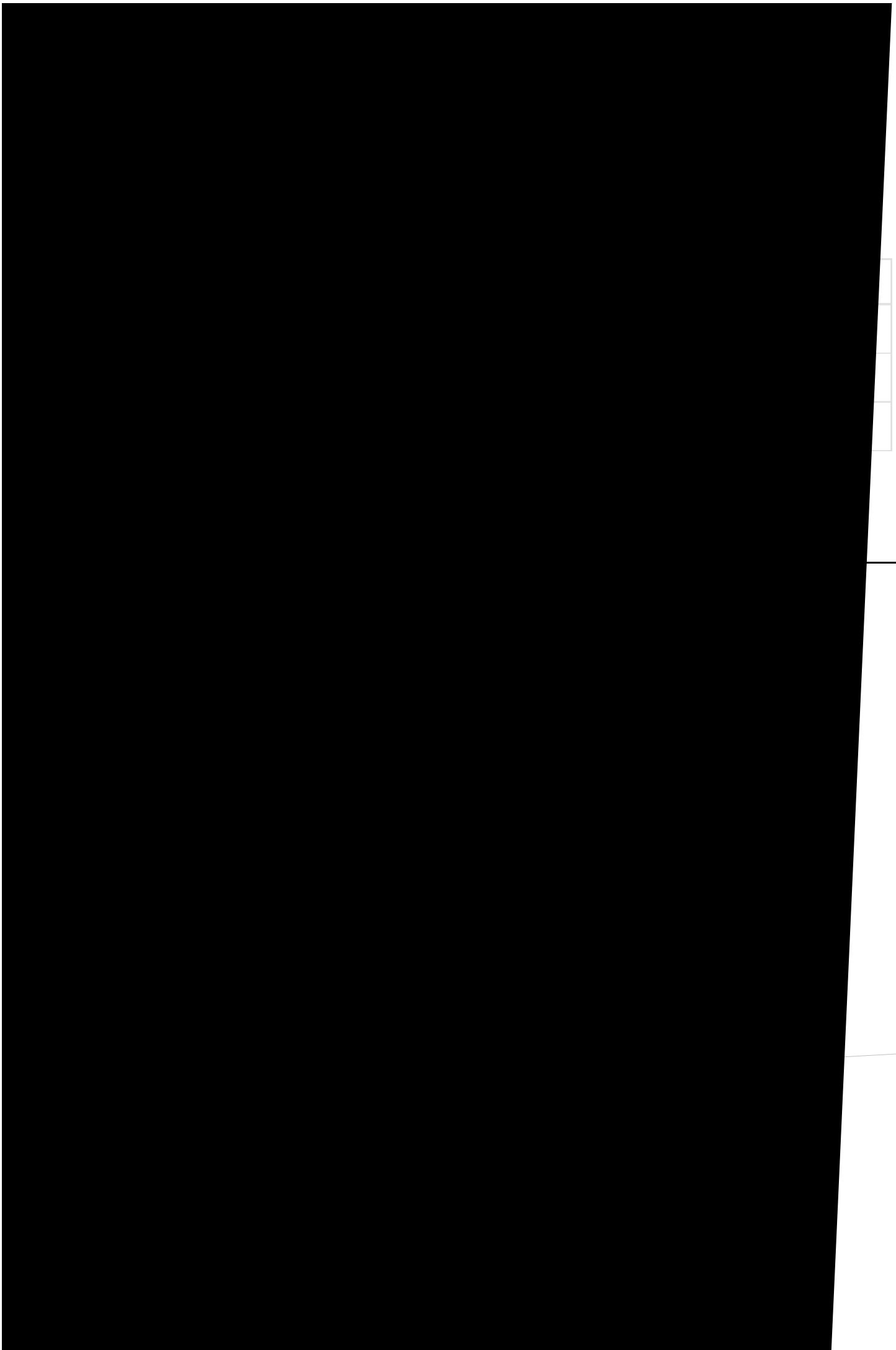
```
{
    "Id": "B ^ ﷺ lű ♂K c g ˉ * ŧ ┘ ﷺ ű K
```

```
{
    "ETag": "\"NDE1MjQ3OTMx\"|AAAAAAAAAAE=",
    "Id": "TestDevice",
    "C2DMessageCount": 0,
    "LastActivity": "0001-01-01T00:00:00",
    "Connecteutž
```

```
{
  "eTag": "string",
  "id": "string",
  "c2DMessageCount": 0,
  "lastActivity": "2020-Ds0--
```

Response Schema


```
{
  "eTag": "string",
  "id": "string",
  "c2DMessageCount": 0,
  "lastActivity": "2020-Ds
```

Get Jobs

Lists the jobs from IoTHub based on timeframe, jobtype, jobstatus and pagesize.


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

[

Typography
Flow
Authorization


```
    "additionalProp3": [
    ]
```

```
"type": 0,
"status": 0,
"metu   <
```

```
    "createdDateTimeUtc": "2020-10-30T10:21:05.676Z",
    "lasa"":1"0"2"0"-"1"0"-"3"0"T"1"0":"2"1":"0"5".6"7"6"Z",
```

Get Modules

Lists the Modules based on zes based on Modules Modui ya Mav Mod 1 dul s haha ha Modul s Modules es da a Ğ


```
{
  "metadata": {
    "addition
```

```
{
    "$metadata": {
        "$type": "DeviceList;1",
        "$uri": "/v1/devices"
    },
    "ContinuationToken": null,
    "Items": [
        {
            "Reported": {},
            "schemaVersion": "1.0",
```

CONDITIONS	URL ENCODED DATA
`deviceId IN ['TestDevice'] AND moduleId = '$edgeAgent'`	`deviceId+IN+%5B%27TestDevice%27%5D+AND+moduleId+%3D+%27%24edgeAgent%27`

Respop vhIEn ₀ G fG ˆ ³ ‘O ? O

Query Modules

Lists the Modules based on the query conditions from request body.

PerPerm p T


```
{
  "metadata": {
    "addition
```

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

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

```
{
  "reported": {
    "additionalProp1": [
```

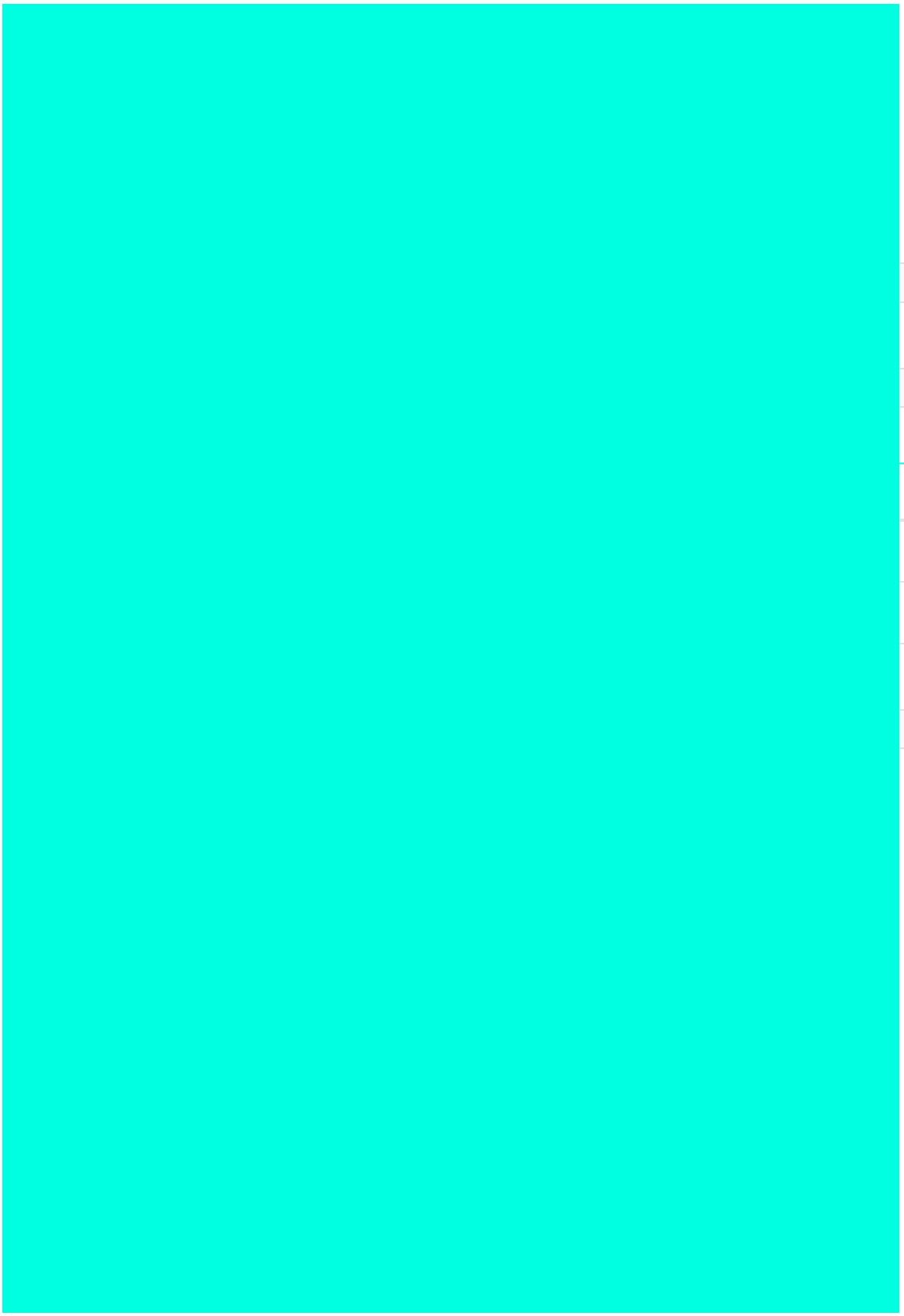
Security

Type: OpenID Connect
Flow: implicit
Authorization URL: https://login.microsoftonline.com/crsltestab2c.onmicrosoft.com

Examples

1. Get Module Data based on DeviceId and ModuleId

```
GET /io Iwu
```



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

T