



# Jamf Protect API

## 101

**JNUC  
2022**  
SAN DIEGO

# Matteo Bolognini

Product Specialist, Jamf



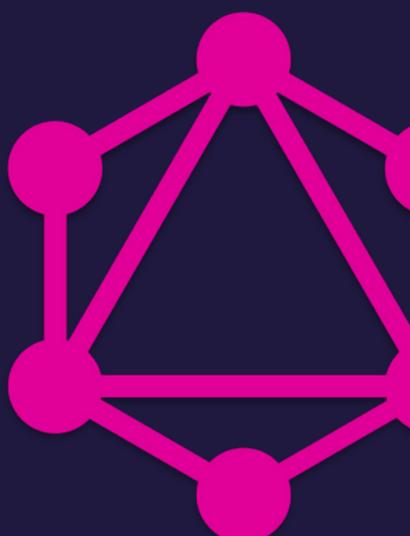
# Agenda

- Introduction to GraphQL
- GraphQL vs REST
- Get started with Protect API
- Workflows & ideas



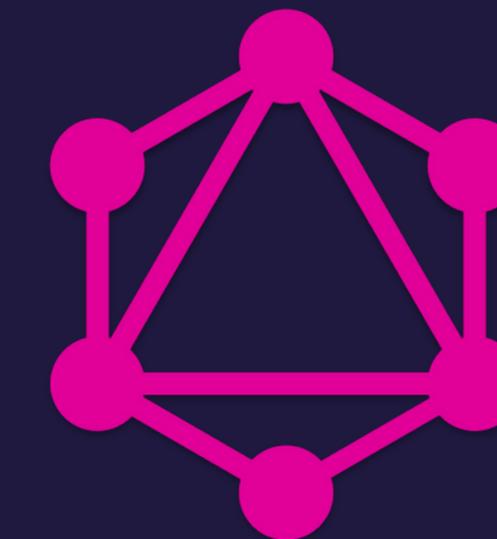
**JNUC  
2022**

# Introduction to GraphQL



**JNUC**  
**2022**





# GraphQL

*A Query Language for APIs*

- Developed & open-sourced by Facebook in 2012
- Declarative Data Fetching
- Server exposes a single endpoint to respond to queries

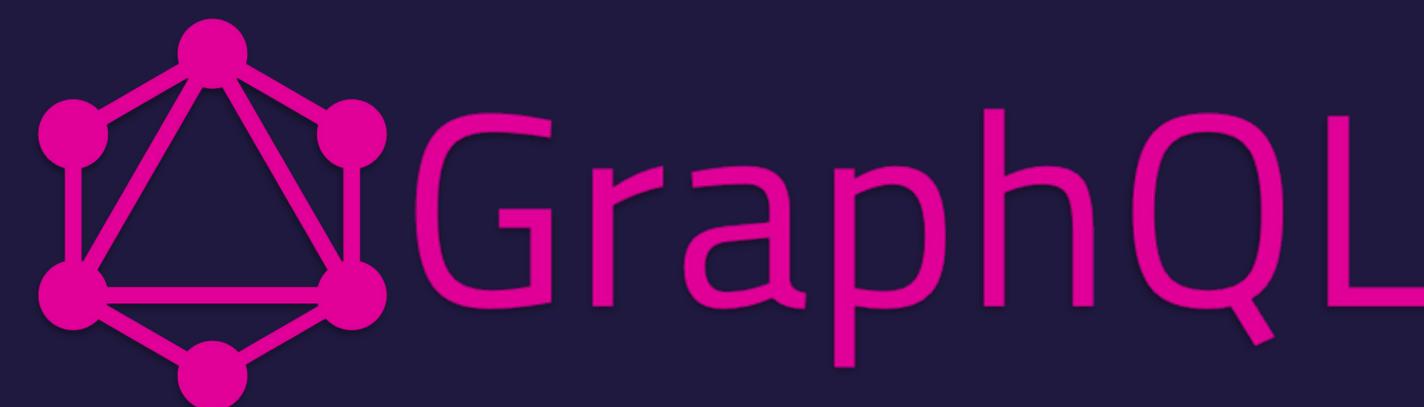


**JNUC**  
**2022**

# GraphQL vs REST

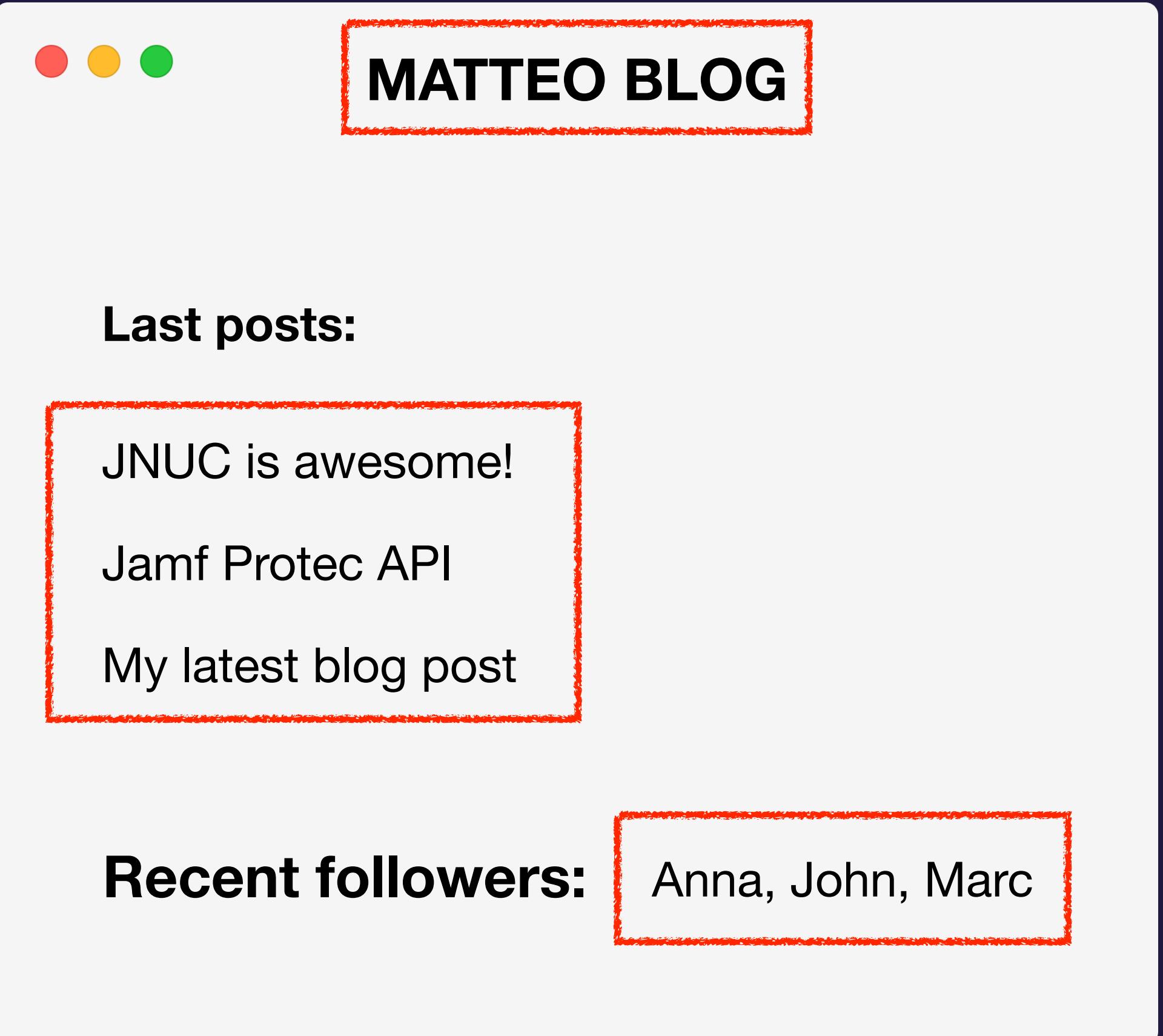


**JNUC  
2022**



HTTP POST

```
query {
  User(id:"abc123") {
    name
    posts {
      title
    }
    followers(last: 3) {
      name
    }
  }
}
```



A screenshot of a web browser window titled "MATTEO BLOG". The window shows a list of posts under the heading "Last posts:" and a list of recent followers under the heading "Recent followers:". The "Last posts:" section contains three items: "JNUC is awesome!", "Jamf Protec API", and "My latest blog post". The "Recent followers:" section contains one item: "Anna, John, Marc". All text elements are enclosed in red-bordered boxes.

**MATTEO BLOG**

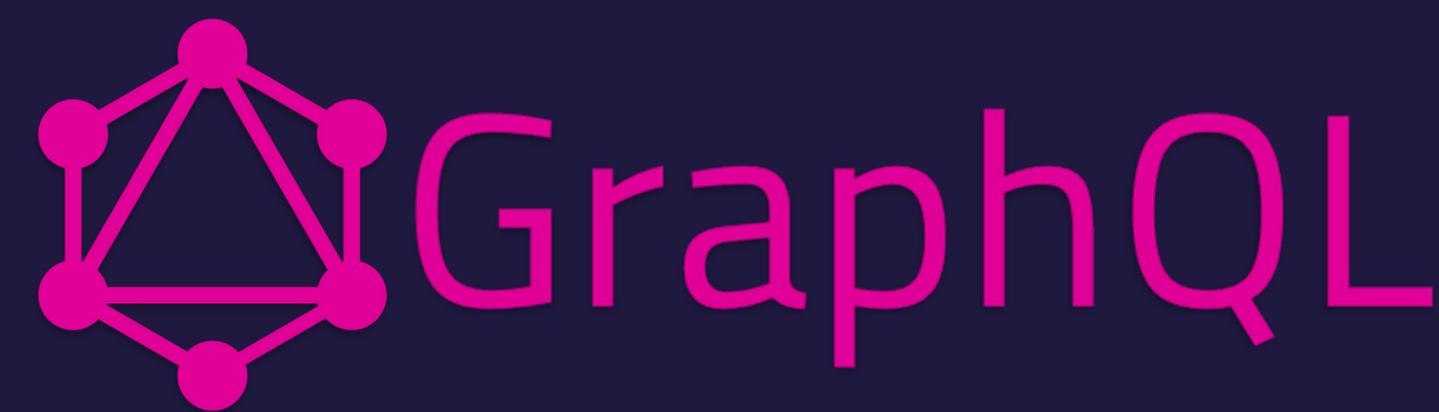
**Last posts:**

- JNUC is awesome!
- Jamf Protec API
- My latest blog post

**Recent followers:**

- Anna, John, Marc

**JNUC**  
**2022**



## HTTP Response

```
{  
  "data": {  
    "User": {  
      "name": "Matteo",  
      "posts": [  
        {title: "JNUC is awesome"},  
        {title: "Jamf Protecc API"},  
        {title: "My latest blogpost"}  
      ],  
      "followers": [  
        {name: "Anna"},  
        {name: "John"},  
        {name: "Marc"}  
      ]  
    }  
  }  
}
```

The screenshot shows a blog post interface with a header "MATTEO BLOG". Below it, there are two sections: "Last posts:" and "Recent followers:". The "Last posts:" section contains three items: "JNUC is awesome!", "Jamf Protecc API", and "My latest blog post". The "Recent followers:" section contains three names: "Anna, John, Marc". All text in these sections is enclosed in red-bordered boxes.

**MATTEO BLOG**

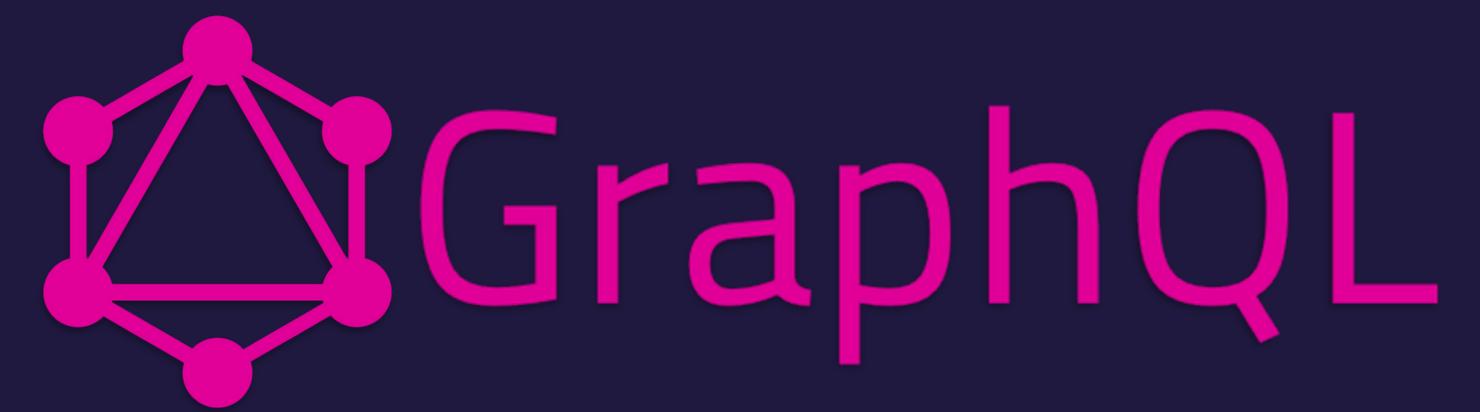
**Last posts:**

- JNUC is awesome!
- Jamf Protecc API
- My latest blog post

**Recent followers:**

- Anna, John, Marc

**JNUC**  
**2022**



Read Data -----> Queries

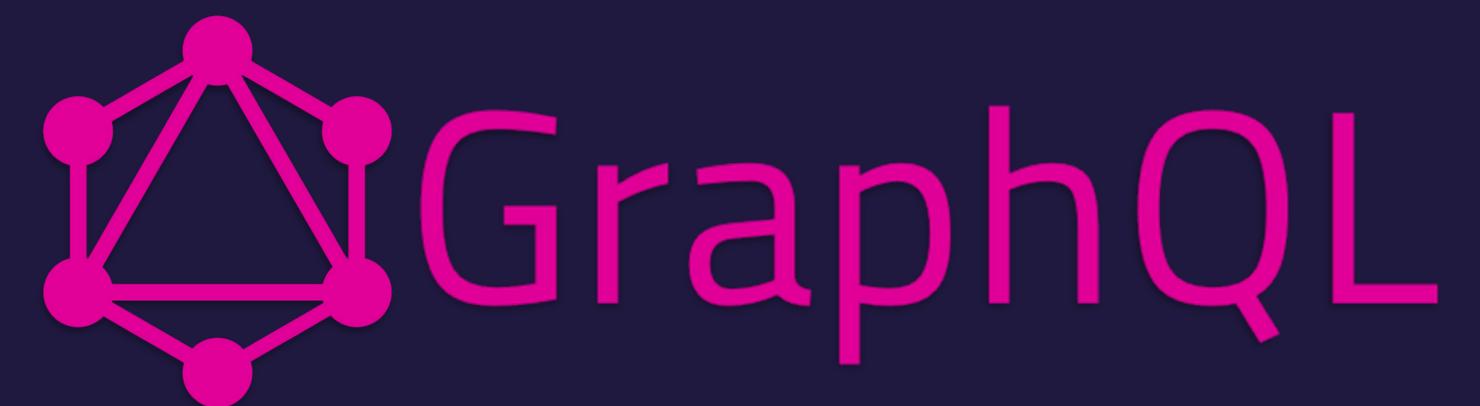
Create  
Update  
Delete



-----> Mutations



**JNUC**  
**2022**



Read Data ..... ➤ Queries

Create  
Update  
Delete } ..... ➤ Mutations

<https://example.protect.jamfcloud.com/graphql>



**JNUC**  
**2022**

# REST API

3 API endpoints

/users/<id>

/users/<id>/posts

/users/<id>/followers



source: [www.howtographql.com](http://www.howtographql.com)

MATTEO BLOG

Last posts:

- JNUC is awesome!
- Jamf Protec API
- My latest blog post

Recent followers:

- Anna, John, Marc

JNUC  
2022

© copyright 2002–2022 Jamf

# REST API

GET /users/<id>

Response

```
{  
  "users":{  
    "id": "abc123",  
    "name": "Matteo",  
    "address": {...} ,  
    "birthday": "January 01 1900"  
  }  
}
```



JNUC  
2022

# REST API

GET /users/<id>/posts

Response

```
{  
  "posts": [  
    {  
      "id": "abcd124"  
      "title": "JNUC is awesome!" ,  
      "content": "Hello everyone _ " ,  
      "comments": "[....]"  
    }, {  
      "id": "abcd1245"  
      "title": "Jamf Protecc API" ,  
      "content": "Hello everyone _ " ,  
      "comments": "[....]"  
    }, {  
      "id": "abcd12456"  
    }]  
}
```



MATTEO BLOG

Last posts:

JNUC is awesome!

Jamf Protec API

My latest blog post



JNUC  
2022

© copyright 2002–2022 Jamf

# REST API

GET /users/<id>/followers

## Response

```
{  
  "followers": [{  
    "id": "xyz987",  
    "name": "Anna",  
    "address": "...",  
    "birthday": "January 01 1901"  
  }, {  
    "id": "xyz988",  
    "name": "John",  
    "address": "...",  
    "birthday": "January 02 1902"  
  }, {  
    "id": "xyz989",  
    "name": "Marc",  
    "address": "...",  
    "birthday": "January 03 1903"}]
```



MATTEO BLOG

Last posts:

JNUC is awesome!

Jamf Protec API

My latest blog post

**Recent followers:**

Anna, John, Marc

**JNUC  
2022**

# Get started with Protect API



**JNUC  
2022**

**jamf | PROTECT**

**Overview**

**Detections** Apple Security Computers

**Computers**

0 Computers with alerts in the past 24 hours

2 Computers with noncompliant Insights

0 Computers not connected for 30 or more days

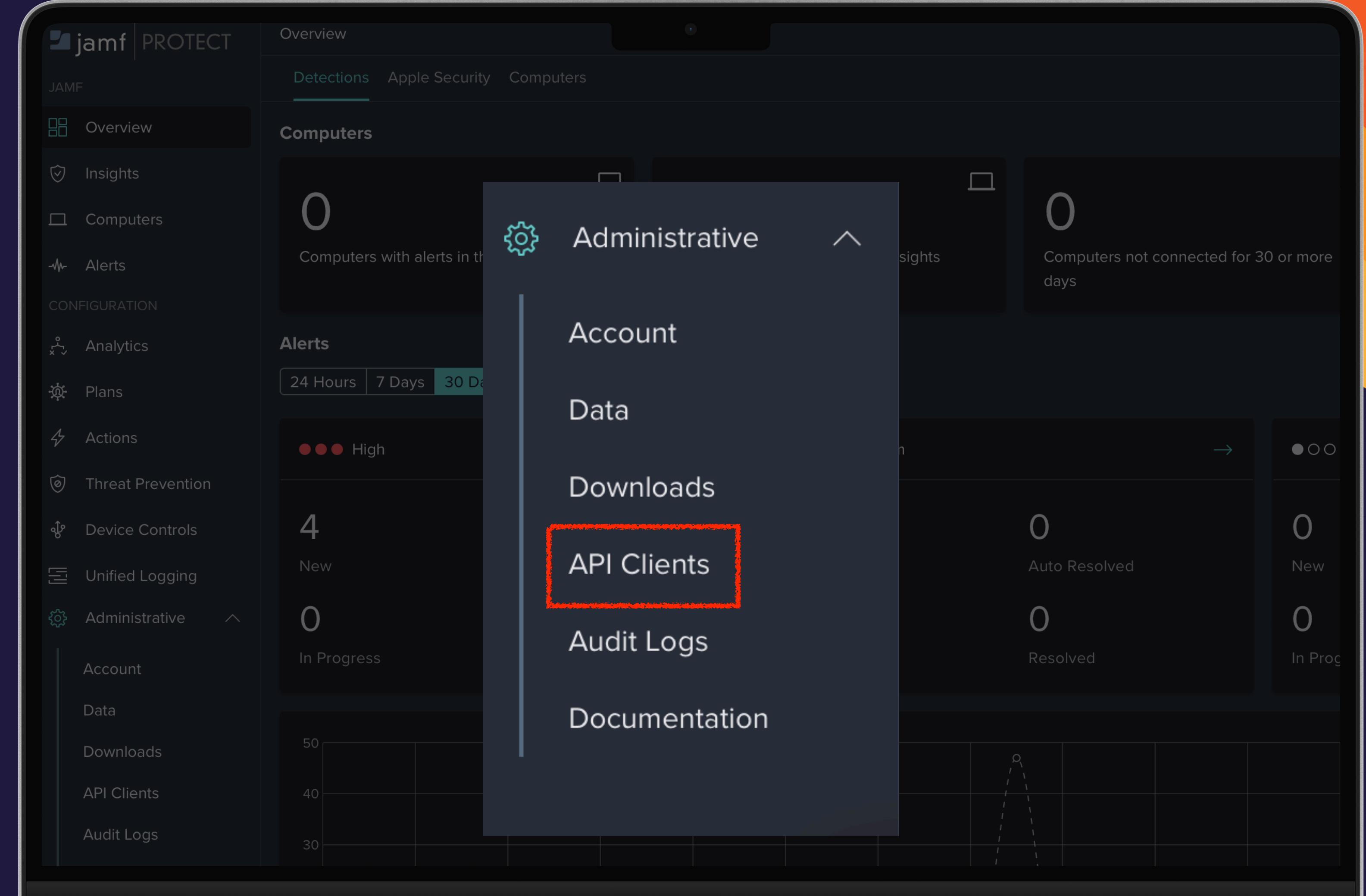
**Alerts**

24 Hours | 7 Days | 30 Days

Severity	New	Auto Resolved	In Progress	Resolved
High	4	16	0	0
Medium	38	0	0	0
Low	0	0	0	0

50  
40  
30

**JNUC**  
**2022**



JNUC  
2022

## ⚠ API Client Password

The password for API Client "JNUC2022" is:

```
AjwLDn4TMCFXHGUXD084Gy1XM1saFD17ZCNxLBduJAMKFygtVDQoJlcYYGzdA  
FE5AB49fissABttIl4vXxckXXoYHscITQ6UQ
```

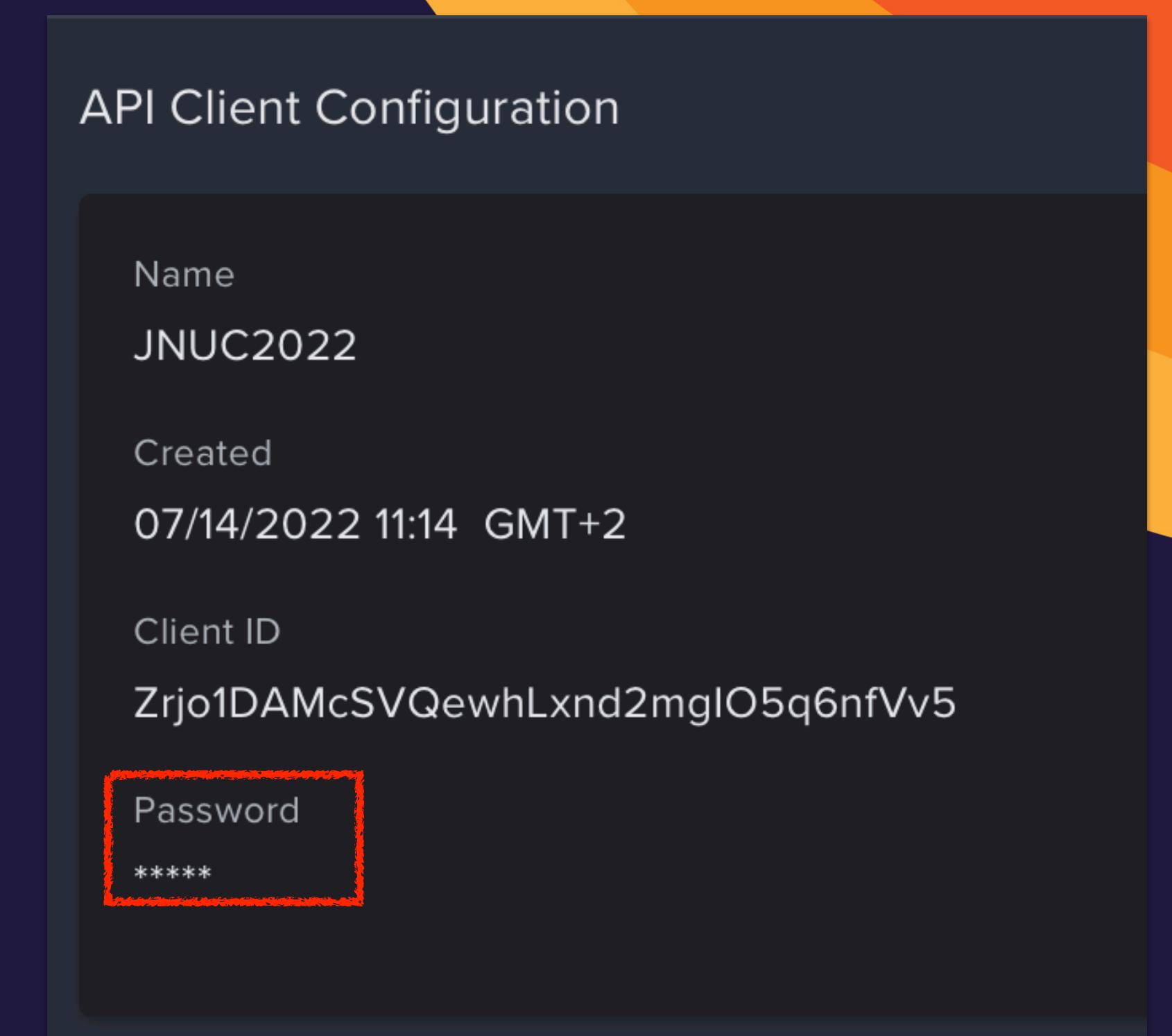
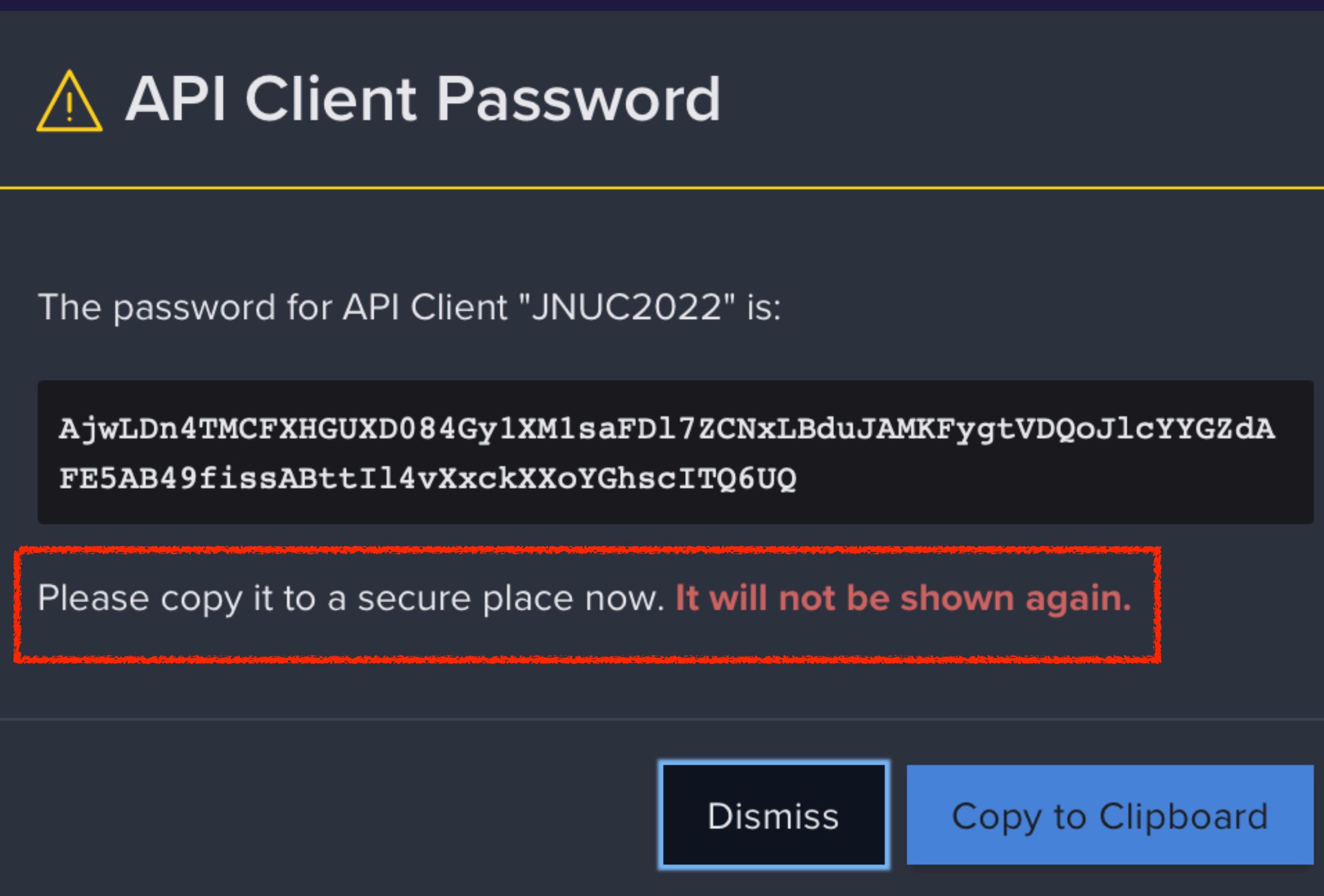
Please copy it to a secure place now. **It will not be shown again.**

[Dismiss](#)

[Copy to Clipboard](#)



**JNUC**  
**2022**



**JNUC**  
**2022**

© copyright 2002–2022 Jamf

# Token request URL

`https://example.protect.jamfcloud.com/token`

# API URL

`https://example.protect.jamfcloud.com/graphql`



# Token request - Bash

```
curl \  
  --silent \  
  --request POST \  
  --header 'content-type: application/json' \  
  --url "https://example.protect.jamfcloud.com/token" \  
  --data "{\"client_id\": \"$CLIENT_ID\",  
           \"password\": \"$PASSWORD\"}" \  
 | jq -r '.access_token' \  
|
```



# Token request - Python

```
token_url = f"https://example.protect.jamfcloud.com/token"
payload = {
    "client_id": client_id,
    "password": password,
}
resp = requests.post(token_url, json=payload)
resp.raise_for_status()
resp_data = resp.json()

return resp_data["access_token"]
```





## Altair GraphQL Client

- Easily interact with Protect API
- GraphQL debugging application
- GraphQL built-in schema



**JNUC  
2022**

## ENVIRONMENTS

Manage your environment variables and preferences

X

Global environment

JNUC 2022

 Delete

### Environments

⋮ JNUC 2022

Add Environment

```
1 v {  
2   "password": "AAAAAAAAAAAAAAA-  
3     GDYjKBkAibBDFEdMjkHEDpqIhUIaWk5SzYGJXIHSZ8ZQo2PSdELh1UBkkLIBdYFjIBXw",  
4  
5   "client_id": "BBBBBBBBBBBBBBBBBBBBBBBBBBBB"  
6 }
```

You can use any of your variables in the app using {{variable}} syntax.

Save



**JNUC**  
**2022**

```
{  
  "password": "${CLIENT_ID}",  
  "client_id": "${PASSWORD}"  
}
```

## ENVIRONMENTS

Manage your environment variables and preferences

Global environment

### Environments

⋮ JNUC 2022

Add Environment

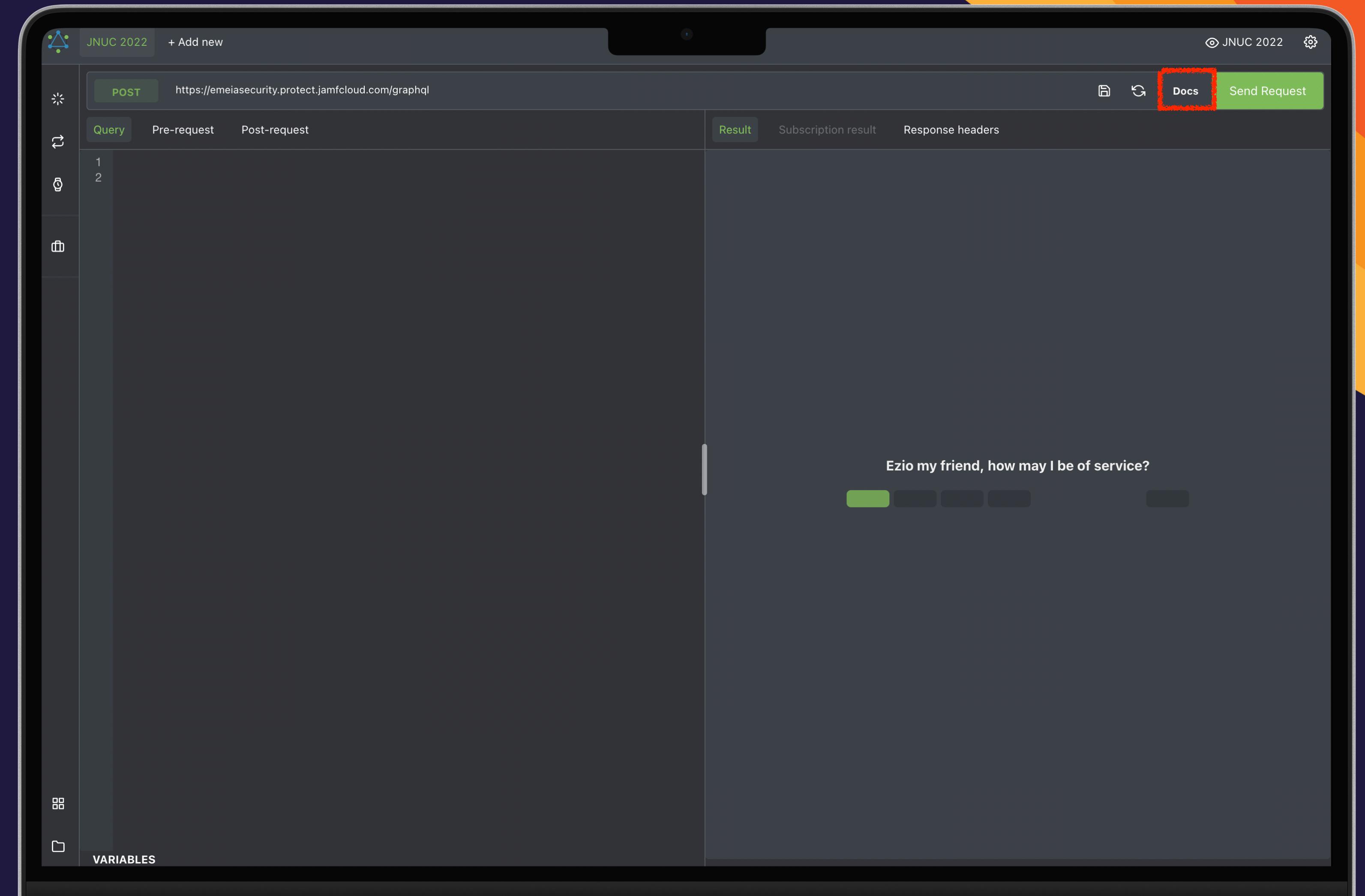
JNUC 2022

```
1 {  
2   "password": "AAAAAAAAAAAAAAA-  
3   GDYjKBIkAiBBDFEdMjkHEDpqIhUIaWk5SzYGJXIHL  
4  
5   "client_id": "BBBBBBBBBBBBBBBBBBBBBBBBB  
6 }
```

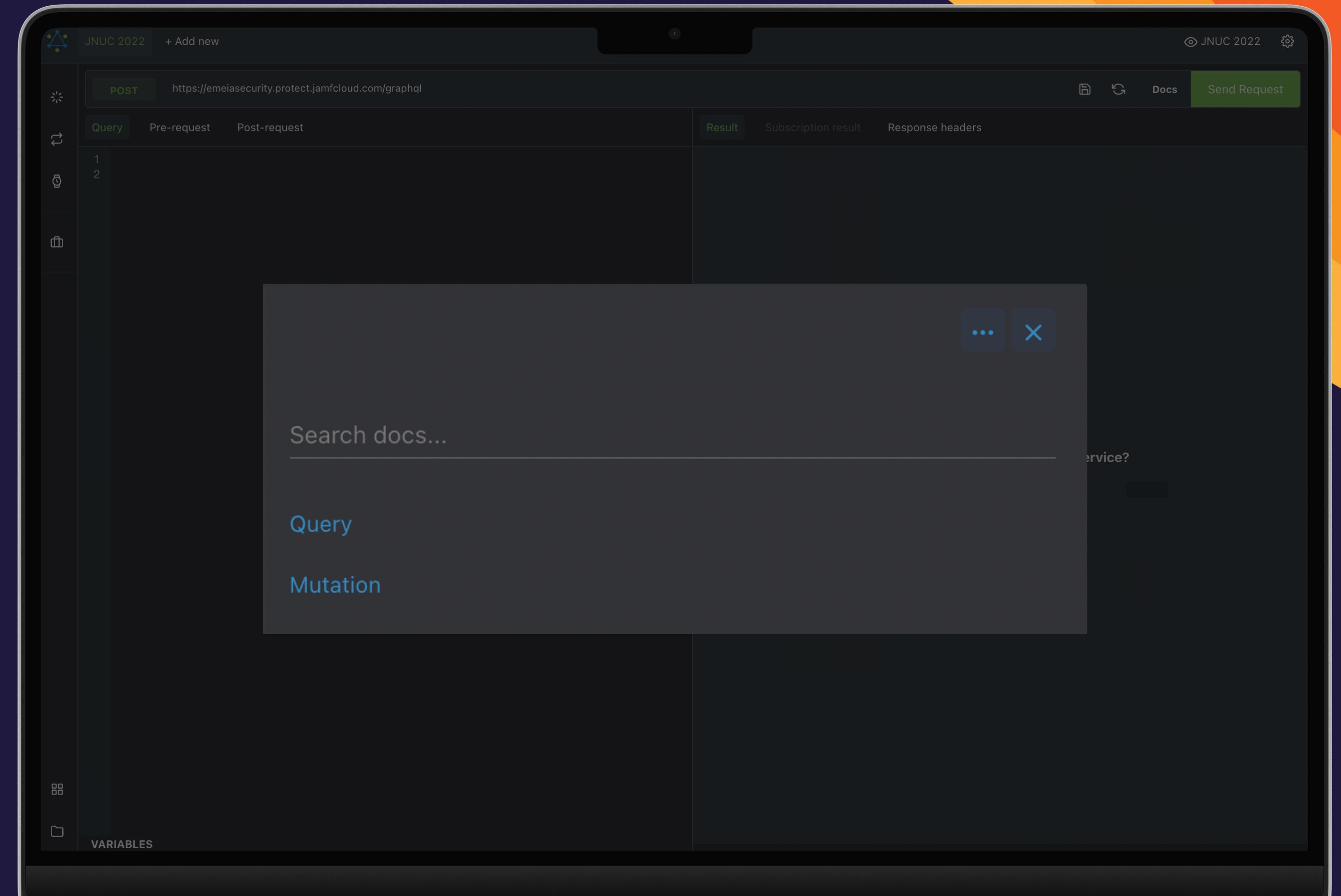
You can use any of your variables in the app using {{variable}} syntax



**JNUC**  
**2022**



JNUC  
2022



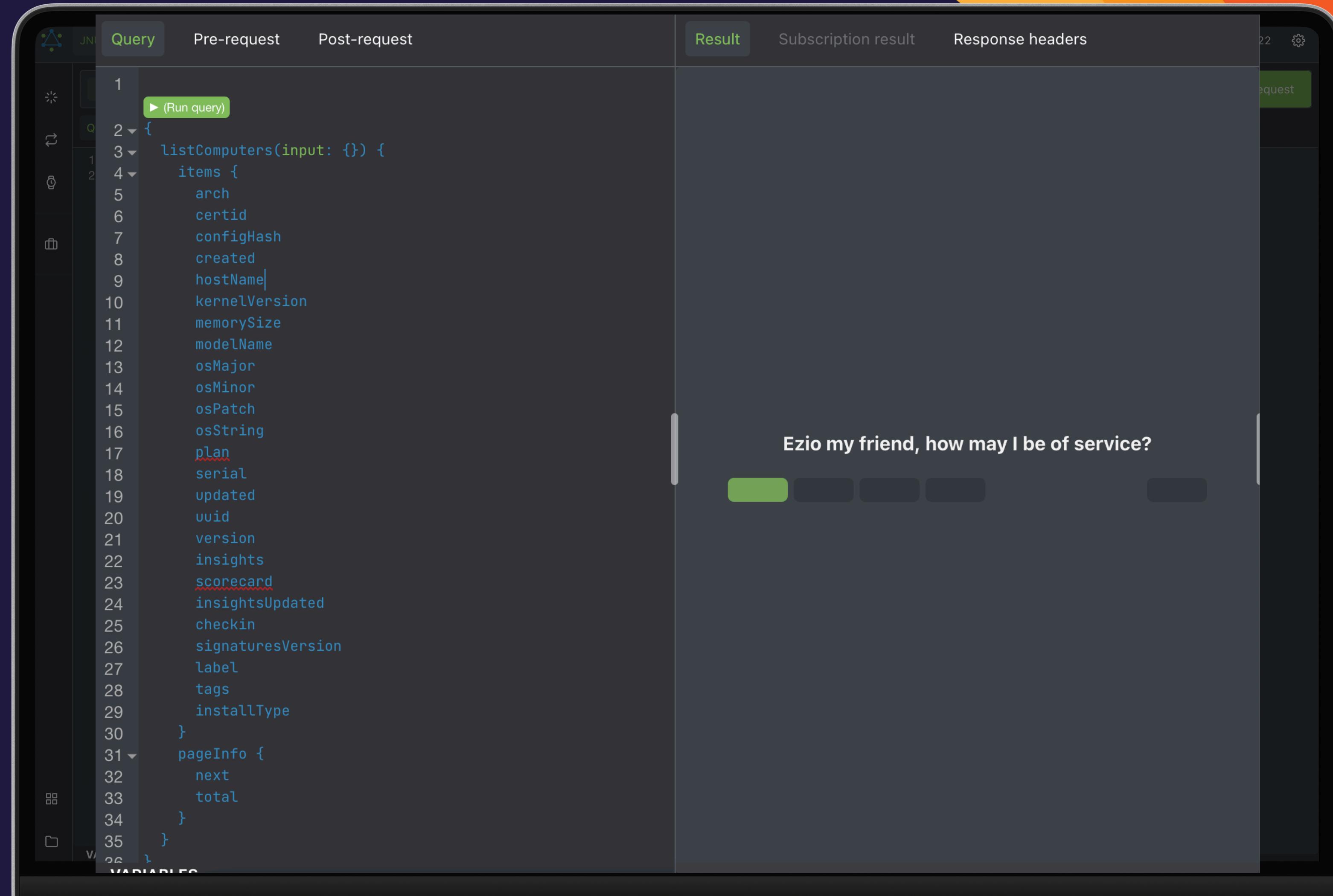
JNUC  
2022

© copyright 2002–2022 Jamf

The screenshot shows a GraphQL playground interface with the following details:

- Title:** JNUC 2022
- Endpoint:** https://emeiassecurity.protect.jamfcloud.com/graphql
- Method:** POST
- Section:** Query
- Available Queries:**
  - getComputer ( uuid ID! ) Computer  
Get information about a specific Computer
  - getPlan ( id ID! ) Plan  
Get a specific Plan
  - getPreventList ( id ID! ) PreventList  
Get a specific Custom Prevent List
  - getUnifiedLoggingFilter ( uuid ID! ) UnifiedLoggingFilter  
Get a specific Unified Logging filter
  - listAlerts ( input AlertQueryInput! ) AlertConnection  
Get one or more Alerts with optional filtering criteria
  - listAnalytics AnalyticConnection  
Get one or more Analytics
  - listComputers ( input ComputerQueryInput ) ComputerConnection  
Get information about one or more Computers with optional filtering criteria
  - listPlans ( input PlanQueryInput ) PlanConnection  
Get one or more Plans with optional filtering criteria
- Buttons:** ADD QUERY (highlighted with a red box), Result, Subscription result, Response headers, Docs, Send Request

JNUC  
2022



A smartphone screen showing the GraphQL playground interface. The top navigation bar includes tabs for "Query", "Pre-request", "Post-request", "Result", "Subscription result", and "Response headers". The "Result" tab is active, showing a dark gray background with a green button at the bottom left. The main area displays a GraphQL query:

```
1  ► (Run query)
2 { 
3   listComputers(input: {}) {
4     items {
5       arch
6       certid
7       configHash
8       created
9       hostName
10      kernelVersion
11      memorySize
12      modelName
13      osMajor
14      osMinor
15      osPatch
16      osString
17      plan
18      serial
19      updated
20      uuid
21      version
22      insights
23      scorecard
24      insightsUpdated
25      checkin
26      signaturesVersion
27      label
28      tags
29      installType
30    }
31   pageInfo {
32     next
33     total
34   }
35 }
```

The number 22 is visible in the top right corner of the phone's status bar.

JNUC  
2022

© copyright 2002–2022 Jamf

The image shows a mobile device displaying the GraphQL playground interface. The screen is divided into two main sections: 'Query' on the left and 'Result' on the right.

**Query Section:**

```
1 (Run query)
2 {
3   listComputers(input: {}) {
4     items {
5       arch
6       certid
7       configHash
8       created
9       hostName
10      kernelVersion
11      memorySize
12      modelName
13      osMajor
14      osMinor
15      osPatch
16      osString
17      plan
18      serial
19      updated
20      uuid
21      version
22      insights
23      scorecard
24      insightsUpdated
25      checkin
26      signaturesVersion
27      label
28      tags
29      installType
30    }
31    pageInfo {
32      next
33      total
34    }
35 }
```

**Result Section:**

200 OK 232ms

```
1 v {
2   "data": null,
3   "errors": [
4     {
5       "path": null,
6       "locations": [
7         {
8           "line": 17,
9           "column": 7,
10          "sourceName": null
11        }
12      ],
13      "message": "Validation error of type
SubSelectionRequired: Sub selection required for type Plan
of field plan @ 'listComputers/items/plan'"
14    },
15    {
16      "path": null,
17      "locations": [
18        {
19          "line": 23,
20          "column": 7,
21          "sourceName": null
22        }
23      ],
24      "message": "Validation error of type
SubSelectionRequired: Sub selection required for type null
of field scorecard @ 'listComputers/items/scorecard'"
25    }
26  ]
27 }
```

CLEAR DOWNLOAD

JNUC Query + Add Pre-request Post-request

Result Subscription result Response headers © JNUC 2022

1 POST https://emeiasecurity.protect.jamfcloud.com/graphql (Run query)

Query 2 Pre-request Post-request

```
1 3 listComputers(input: {}) {  
2 4   items {  
3 5     arch  
3 6     certid  
3 7     configHash  
3 8     created  
3 9     hostName  
3 10    kernelVersion  
3 11    memorySize  
3 12    modelName  
3 13  }  
3 14}
```

200 OK 232ms

1 Result Subscription result Response headers

```
1 "data": null,  
2 "errors": [  
3   {  
4     "path": null,  
5     "locations": [  
6       {  
7         "line": 17,  
8         "column": 7,  
9         "sourceName": null  
10      }  
11    }  
12  }  
13 ]  
14   "message": "Validation error of type  
15   SubSelectionRequired: Sub selection required for type null  
16   of field scorecard @ 'listComputers/items/scorecard'"  
17 }  
18 ]  
19 }  
20 ]  
21 }  
22 ]  
23 ]  
24 ]  
25 ]  
26 ]  
27 ]
```

"message": "Validation error of type SubSelectionRequired: Sub selection required for type Plan of field plan @ 'listComputers/items/plan'"

version  
insights  
scorecard  
insightsUpdated  
checkin  
signaturesVersion  
label  
tags  
installType  
}  
pageInfo {  
 next  
 total  
}  
}

18 v i  
19 "line": 23,  
20 "column": 7,  
21 "sourceName": null  
22 }  
23 ]  
24 "message": "Validation error of type  
25 SubSelectionRequired: Sub selection required for type null  
26 of field scorecard @ 'listComputers/items/scorecard'"  
27 }

DOWNLOAD



The image shows a mobile device displaying the GraphQL playground interface. The screen is divided into two main sections: 'Query' on the left and 'Result' on the right.

**Query Section:**

```
1 1 (Run query)
2 {
3   listComputers(input: {}) {
4     items {
5       arch
6       certid
7       configHash
8       created
9       hostName
10      kernelVersion
11      memorySize
12      modelName
13      osMajor
14      osMinor
15      osPatch
16      osString
17      plan
18      serial
19      updated
20      uuid
21      version
22      insights
23      scorecard
24      insightsUpdated
25      checkin
26      signaturesVersion
27      label
28      tags
29      installType
30    }
31    pageInfo {
32      next
33      total
34    }
35 }
```

The 'scorecard' field is highlighted with a red box.

**Result Section:**

200 OK 232ms

```
1 v {
2   "data": null,
3   "errors": [
4     {
5       "path": null,
6       "locations": [
7         {
8           "line": 17,
9           "column": 7,
10          "sourceName": null
11        }
12      ],
13      "message": "Validation error of type
SubSelectionRequired: Sub selection required for type Plan
of field plan @ 'listComputers/items/plan'"
14    },
15    {
16      "path": null,
17      "locations": [
18        {
19          "line": 23,
20          "column": 7,
21          "sourceName": null
22        }
23      ],
24      "message": "Validation error of type
SubSelectionRequired: Sub selection required for type null
of field scorecard @ 'listComputers/items/scorecard'"
25    }
26  ]
27 }
```

A 'CLEAR' button is located at the top right of the Result section.

JNUC Query + Add Pre-request Post-request

Result Subscription result Response headers © JNUC 2022

1 POST https://emeiasecurity.protect.jamfcloud.com/graphql (Run query)

Query 2 Pre-request Post-request

```
1 3 listComputers(input: {}) {
2   items {
3     arch
4     certid
5     configHash
6     created
7     hostName
8     kernelVersion
9     memorySize
10    modelName
11    osMajor
12    osMinor
13    osPatch
14    osString
15
16 }
```

200 OK 232ms

1 Result Subscription result Response headers

```
1 "data": null,
2 "errors": [
3   {
4     "path": null,
5     "locations": [
6       {
7         "line": 17,
8         "column": 7,
9         "sourceName": null
10       }
11     ],
12   },
13   "message": "Validation error of type
SubSelectionRequired: Sub selection required for type Plan
of field plan @ 'listComputers/items/plan'"
```

✖ Field "plan" of type "Plan" must have a selection of subfields. Did you mean "plan { ... }"?

```
23   scorecard
24   insightsUpdated
25   checkin
26   signaturesVersion
27   label
28   tags
29   installType
30 }
31 pageInfo {
32   next
33   total
34 }
35 }
```

20 "column": 7,
21 "sourceName": null
22 }
23 ],
24 "message": "Validation error of type
SubSelectionRequired: Sub selection required for type null
of field scorecard @ 'listComputers/items/scorecard'"
25 }
26 ]
27 }

DOWNLOAD

- `listComputers` ( input [ComputerQueryInput](#) )  
[ComputerConnection](#)

Get information about one or more Computers with optional filtering criteria

#### FIELDS

- `items` [\[Computer\]](#)  
Fields to return in a query about Computers

- `pageInfo`  [PageInfo](#)

- `plan`  [Plan](#)

Plan assigned to the Computer



## Plan

A comprehensive security configuration that includes Analytics, an Action Configuration, and check-in settings

### FIELDS

- **analytics** [\[Analytic\]](#)

Analytics added to the Plan

- **created** [AWSDateTime](#)

Date when the Plan was created

- **description** [String](#)

Description of the Plan

- **hash** [String](#)

The hash of the Plan

- **id** [ID!](#)

A unique identifier assigned by Jamf Protect

- **uuid** [ID](#)

A globally unique identifier assigned by Jamf Protect

- **name** [String](#)

The name of the Plan

- **updated** [AWSDateTime](#)

Date when the Plan was last updated

- **logLevel** [LOG\\_LEVEL\\_ENUM](#)

The log level used by Computers with this Plan

### Plan

A comprehensive security configuration that includes Analytics, an Action Configuration, and check-in settings

#### FIELDS

- **analytics** [\[Analytic\]](#)  
Analytics added to the Plan
- **created** [AWSDateTime](#)  
Date when the Plan was created
- **description** [String](#)  
Description of the Plan
- **hash** [String](#)  
The hash of the Plan
- **id** [ID!](#)  
A unique identifier assigned by Jamf Protect
- **uuid** [ID](#)  
A globally unique identifier assigned by Jamf Protect
- **name** [String](#)  
The name of the Plan
- **updated** [AWSDateTime](#)  
Date when the Plan was last updated
- **logLevel** [LOG\\_LEVEL\\_ENUM](#)  
The log level used by Computers with this Plan

```
plan{  
    id  
    name  
    updated  
}
```



# Protect GraphQL API tips

Pagination

Default - 100

Max - 200



# Protect GraphQL API tips

```
query listComputers_pagination {  
  listComputers(input: { pageSize: 200 })
```

```
query listComputers($page_size: Int, $next: String) {  
  listComputers(input: { pageSize: $page_size, next: $next })
```



# Protect GraphQL API tips

No rate limiting on API calls



**JNUC  
2022**

# Protect GraphQL API tips

No rate limiting on API calls

Token lifetime 86400s (24 hours)

Re-use tokens



# Workflows & ideas



**JNUC  
2022**

# Workflows & ideas

Example 1

“Is the Threat Prevention database up to date  
on the Macs?”



JNUC  
2022

# Check TP version per device

```
query listComputers($page_size: Int, $next: String) {  
    listComputers(input: { pageSize: $page_size, next: $next }) {  
        items {  
            serial  
            checkin  
            signaturesVersion  
            version  
        }  
        pageInfo {  
            next  
        }  
    }  
}
```



# Response

```
{  
  "data": {  
    "listComputers": {  
      "items": [  
        {  
          "serial": "A1B2C3D4E5F",  
          "checkin": "2022-09-27T02:39:36.160007Z",  
          "signaturesVersion": 4972,  
          "version": "3.3.0.607"  
        },  
        {  
          "serial": "Z9Y8X7W6V5",  
          "checkin": "2022-09-27T12:52:09.897357Z",  
          "signaturesVersion": 4972,  
          "version": "3.2.0.557"  
        }  
      ],  
      "pageInfo": {  
        "next": null  
      }  
    }  
  }  
}
```



# Check Jamf Pro TP database

```
query check_Threat_Prevention_db{  
    listThreatPreventionVersions{  
        id  
        created  
    }  
}
```



# Response



```
{  
  "data": {  
    "listThreatPreventionVersions": [  
      {  
        "id": "4972",  
        "created": "2022-07-14T00:30:38.315389Z"  
  
      },  
      {  
        "id": "4971",  
        "created": "2022-07-12T16:15:14.302903Z"  
      },  
      {  
        "id": "4970",  
        "created": "2022-07-11T17:40:45.694394Z"  
      },  
      {  
        "id": "4969",  
        "created": "2022-07-08T13:51:29.234721Z"  
      },  
      {  
        "id": "4968",  
        "created": "2022-07-07T18:06:26.606884Z"  
      },  
      {  
        "id": "4967",  
        "created": "2022-07-06T15:36:34.770172Z"  
      },  
      {  
        "id": "4966",  
        "created": "2022-07-04T15:16:08.423850Z"  
      },  
      {  
        "id": "4965",  
        "created": "2022-07-01T22:20:02.087873Z"  
      },  
      {  
        "id": "4964",  
        "created": "2022-06-30T16:01:06.574628Z"  
      },  
      {  
        "id": "4963",  
        "created": "2022-06-29T21:37:33.745058Z"  
      }  
}
```

**JNUC  
2022**

# Response

```
{  
  "data": {  
    "listThreatPreventionVersions": [  
      {  
        "id": "4972",  
        "created": "2022-07-14T00:30:38.315389Z"  
      },  
      {  
        "id": "4969",  
        "created": "2022-07-08T13:51:29.234721Z"  
      },  
      {  
        "id": "4968",  
        "created": "2022-07-07T18:06:26.606884Z"  
      },  
      {  
        "id": "4967",  
        "created": "2022-07-06T15:36:34.770172Z"  
      },  
      {  
        "id": "4966",  
        "created": "2022-07-04T15:16:08.423850Z"  
      }  
    ]  
  }  
}
```



# Workflows & ideas

Example 2 - Notify new Analytics

“I want to be notified when a new Analytic is added to the Protect tenant”



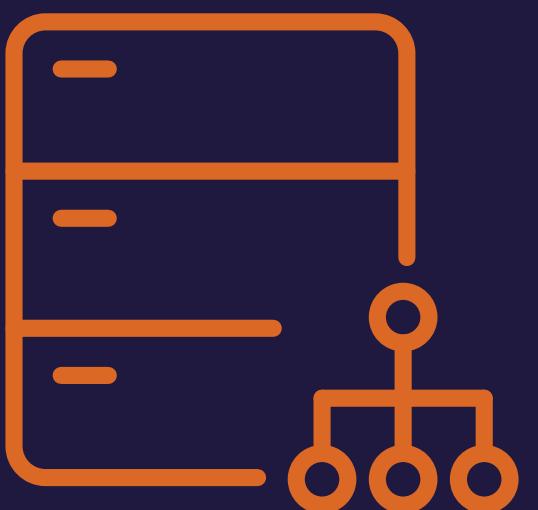
# Example 2 - Notify new Analytics

```
query listAnalytics{  
    listAnalytics{  
        items {  
            hash  
            name  
            created  
            updated  
            description  
            jamf  
            severity  
            uuid  
        }  
    }  
}
```



# Example 2 - Notify new Analytics: First run

```
query listAnalytics{  
    listAnalytics{  
        items {  
            hash  
            name  
            created  
            updated  
            description  
            jamf  
            severity  
            uuid  
        }  
    }  
}
```



{JSON}



JNUC  
2022

# Example 2 - Notify new Analytics: Next run

```
query listAnalytics{  
    listAnalytics{  
        items {  
            hash  
            name  
            created  
            updated  
            description  
            jamf  
            severity  
            uuid  
        }  
    }  
}
```



{JSON}



{JSON}



JNUC  
2022

# Example 2 - Notify new Analytics

**Name:**

SuspiciousOfficeFileCreation

**Owner:**

Jamf

**Description:**

Suspicious file created under productivity  
software

**Severity:**

Medium

**Created:**

2021-12-15 22:39

**Modified:**

2022-05-25 20:21

[Click Here to View Analytic](#)



**JNUC**  
**2022**

© copyright 2002–2022 Jamf

# Jamf Protect GitHub repo



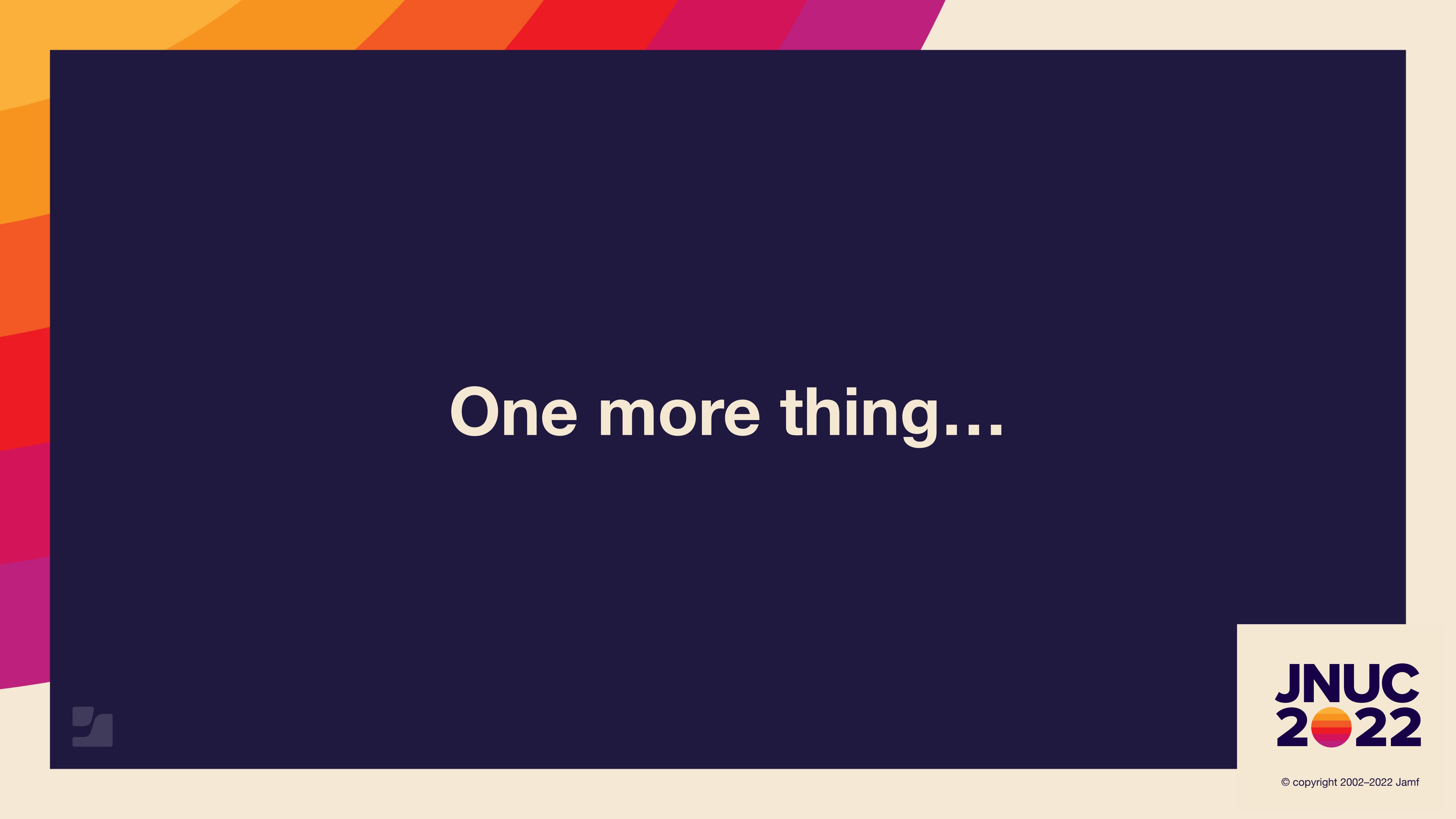
**JNUC  
2022**

# Recap

- Introduction to GraphQL
- GraphQL vs REST
- Get started with Protect API
- Workflows & ideas



**JNUC  
2022**



# One more thing...



**JNUC**  
**2022**



**MAC ADMINS  
FOUNDATION**



**JNUC  
2022**

© copyright 2002–2022 Jamf



Thank you for  
listening!



**JNUC**  
**2022**