

#BullhornEngage

ENGAGE

API Developer Training: Insights for Integrations

Ted
Soper

ted.soper@bullhorn.com

Barry
Shevlin

barry.shevlin@bullhorn.com

Ethan
Schroeder

ethan.schroeder@bullhorn.com

Bullhorn API Overview


Bullhorn
Open Source

Guides

Docs


Libraries

Design



Documentation


All the knowledge you need to build with the Bullhorn platform



REST

RESTful API service that lets you perform actions against Bullhorn data.


[View REST Docs](#)



OAUTH

OAuth is a simple and secure method for authenticating users and allowing third party access to your server.

[View OAuth Docs](#)



SOAP

Bullhorn API service based on SOAP (Simple Object Access Protocol) for exchanging structured information using XML.

[View SOAP Docs](#)

API Features & Code Examples

ENGAGE

BULLHORN API AUTHENTICATION

Bullhorn Help

Looking for answers? Check out all of our [Support articles](#) for common questions or contact Bullhorn [Support](#) directly.



Using The Application



Configuring The Application



Integrating The Application

Start with Canvas

Learn About Novo

Contact Support

View All Topics

Interested in joining the Marketplace?

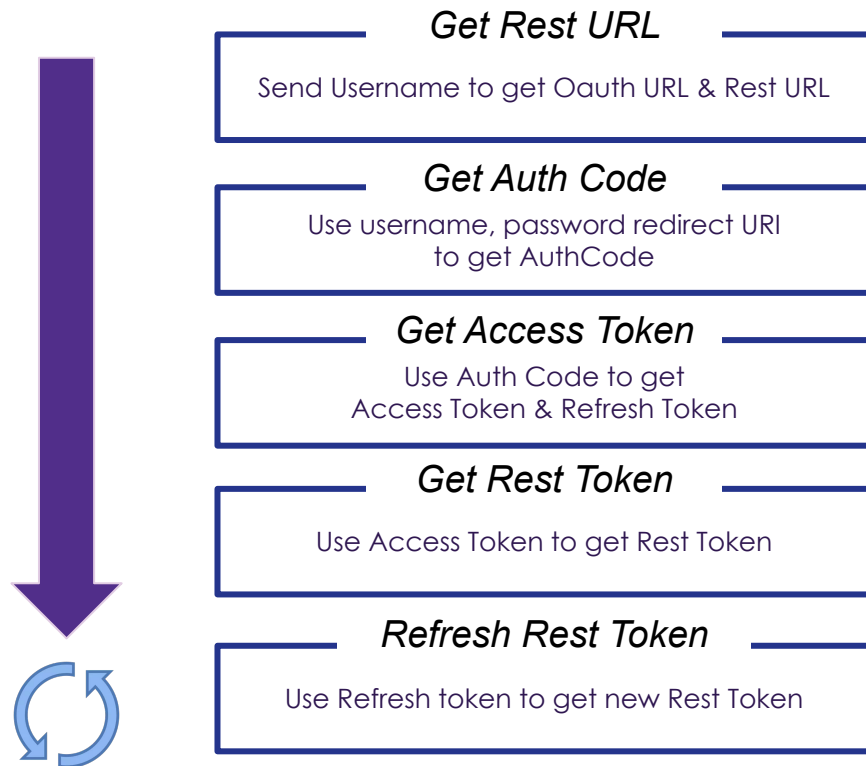
Bullhorn is always adding new partners to the ecosystem.

Contact us to learn more about our partnership options.

[Become a Partner](#)

- OAUTH 2.0
- Registration
- Shared key - Best practice
- Individual Key - Per case basis only

BULLHORN OAUTH



BULLHORN OAUTH

1. Get Rest URL

2. Get Auth Code

3. Get Access Token

4. Get Rest Token

> Refresh Rest Token

Get Rest URL

```
rest.bullhornstaffing.com/rest-services/loginInfo  
?username=username
```

- Returns the Rest URL & Auth URL for the Account
- Check for Datacenter API URLs
<http://bullhorn.github.io/Data-Center-URLs/>

BULLHORN OAUTH

1. Get Rest URL

2. Get Auth Code

3. Get Access Token

4. Get Rest Token

> Refresh Rest Token

GET Auth Code

```
/authorize?client_id=clientId&response_type=code  
&action=Login&username=username&password=password  
&redirect_uri=redirectUrl
```

POST for Access Token

```
/token?grant_type=authorization_code&code= authCode  
e&client_id=clientId&client_secret=clientSecret&r  
edirect_uri=redirectUrl
```

GET Rest Token

```
/login?version=*&access_token= accessToken
```

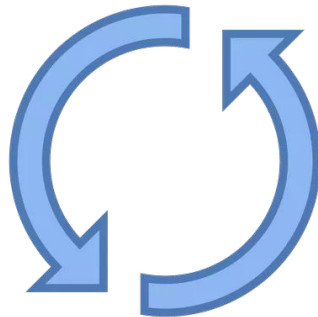
BULLHORN OAUTH

1. Get Rest URL
2. Get Auth Code
3. Get Access Token
4. Get Rest Token
- > **Refresh Rest Token**

Refresh Rest Token

```
/token?grant_type=refresh_token  
&refresh_token=refreshToken&client_id=clientId  
&client_secret=clientSecret
```

- Default REST token lasts for 10 minutes
- Get a new REST token with a Refresh request



Entities & Metadata Best Practices

Entity Lookup

- **/search**

- Lucene-based search
- Available for most common entities
- Faster than Query

```
/search/ClientContact  
?query=(clientCorporation.id:123)  
&fields=name,id,status
```

- **/query**

- JPQL (SQL-like) querying
- Use when search isn't available

```
/query/CorporateUser  
?where=(name='Ted Soper')  
&fields=id,email
```

- **/find (FastFind)**

- Searches several entities in one request
- Useful with user input

```
/find  
?query=monster  
&countPerEntity=3
```

Entity Pagination

- **Count**

- How many objects to return
- Default: 30
- Max: 500

- **Start**

- How many objects to skip
- Default: 0

1

Get first page

```
/search/...&count=500&start=0
```

2

Get total from response

```
{  
  data: [500],  
  total: 1700  
}
```

3

Get remaining pages in parallel

```
/search/...&count=500&start=500
```

```
/search/...&count=500&start=1000
```

```
/search/...&count=500&start=1500
```

Efficient Lookups

- Only get the fields you need
 - Avoid “**fields=***”
 - Exception: one-time metadata calls
- Don't duplicate requests
- Use */search* if possible
- If you only need the count, leave off fields/layout

Get notes for one entity

```
/entity/JobOrder/1234/notes  
?fields=comments,action
```

Search for list of IDs & count

```
/search/ClientContact  
?query=firstName:Barry
```

Metadata Lookup

- Full structural data about an entity
- Field constraints
- Reference types
- Custom field definitions
- Job track information

Lookup before creation

```
/meta/Candidate  
?fields=*  
&meta=basic
```

Add to another request

```
/entity/JobOrder/12345  
?layout=RecordEdit  
&meta=track
```

Metadata Example Responses

Reference Field

```
[
  {
    "name": "assignedUsers",
    "type": "TO_MANY",
    "label": "Assigned to",
    "optionsType": "CorporateUser",
    "optionsUrl": "/options/CorporateUser",
    "associatedEntity": {
      "entity": "CorporateUser",
      "entityMetaUrl": "/meta/CorporateUser",
      "label": "Corporate User",
      "fields": [...]
    }
  },
  {...},
  {...}
]
```

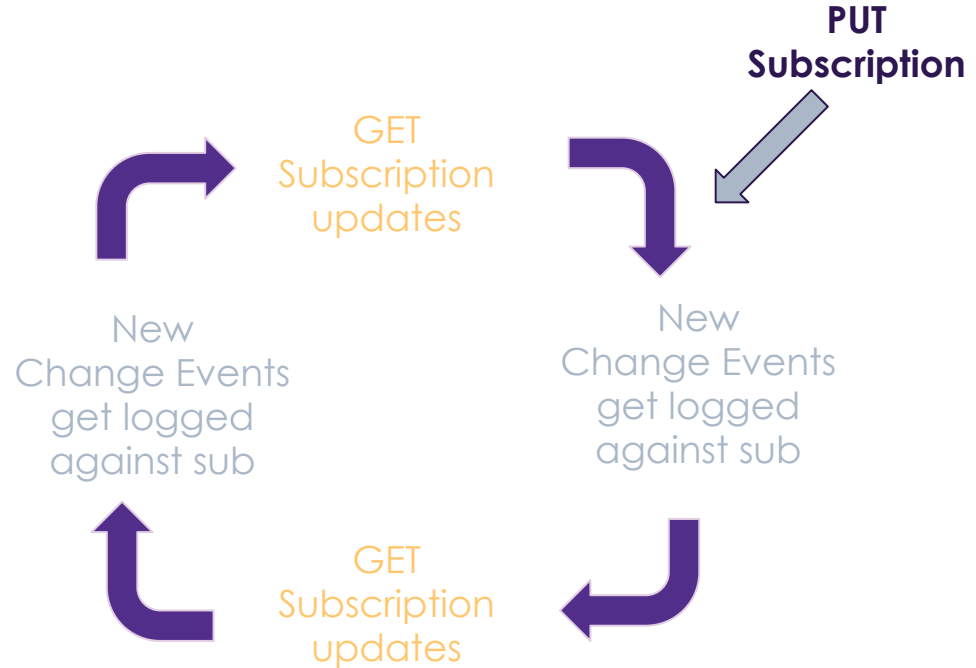
Custom Field w/ Options

```
[
  {
    "name": "customText3",
    "type": "SCALAR",
    "dataType": "String",
    "maxLength": 100,
    "label": "Primary Department",
    "options": [
      {
        "value": "Boston MA Dept",
        "label": "Boston MA Dept"
      }, {
        "value": "Orlando FL Dept",
        "label": "Orlando FL Dept"
      }
    ]
  },
  {...}
]
```

Subscriptions Overview

Subscriptions for data sync

- Create a Subscription based on data change events
 - Entity create/update
 - Field map change
 - Job match search



Subscriptions for data sync

- Parameters
 - type
 - (entity)names
 - eventTypes

PUT /event/subscription

Lets you subscribe to event types.

HTTP Request

{corpToken}/event/subscription/{subscriptionId}

Parameter	Required	Description
type	yes	entity
names	yes	Required with "entity" type. A comma-separated list of entity names.
eventTypes	yes	Required with "entity" type. A comma-separated list of entity event types: INSERTED, UPDATED, DELETED.
BhRestToken	no	Token that represents a session established by the login process. Must be sent with all subsequent requests to the API. The session key can be provided in the BhRestToken query string, a cookie, or an HTTP header.

Example

...event/subscription/**MySubName**?type=**entity**&names=**Candidate**
&eventTypes=**INSERTED , UPDATED , DELETED**

Custom Tabs

Custom Tabs

- **Custom Tab**
 - Pulls in an iframe application
 - Appbridge user send request from extension to Bullhorn
 - Through appbridge make requests to the bullhorn api

BULLHORN OAUTH & Custom Tabs

1. Get Access Token
 2. Get Rest Token
- > Refresh Rest Token

Use Auth Code from App Bridge

Grab Auth Code from the **CustomComponent/redirect**

POST for Access Token

```
/token?grant_type=authorization_code&code= authCode  
&client_id=clientId&client_secret=clientSecret&  
redirect_uri=redirectUrl
```

GET Rest Token

```
/login?version=*&access_token= accessToken
```

API Rate Limiting

Rate Limiting

429 - “Too Many Requests”

- You have a bucket of API requests per minute
- **Don't** throw your request away, just wait 1 second and retry!
- Multiple apps should have **multiple** keys.

***This is NEW!** - *if you have an existing app talk to Bullhorn support if you have questions.*

Questions?

ENGAGE

Links

- Guides: bullhorn.github.io
- API Documentation: bullhorn.github.io/rest-api-docs