

# ENGAGE

## Extensions - Your Bullhorn Superpower

Amrutha Rajiv

Brian Kimball

Joshua Godi



**Brian Kimball**  
ENTERPRISE ARCHITECT



**Joshua Godi**  
PRINCIPAL UI  
MEOWNGINEER



**Amrutha Rajiv**  
SOFTWARE ARCHITECT

**ENGAGE**

# What we will cover today

- BOTs
- Field Interactions
- Page Interactions
- Novo-Elements Updates
- Bullhorn CLI

# Bots



# ENGAGE

# What can they do?



- Update associated records
  - When a job is lost update submission status to “Lost to Competition”
  - On end of placement (and no other active placements exist) update candidate status and date available
- Configure custom workflows
  - If placement pay rate or bill rate was updated, update corresponding record in ADP or Peoplesoft.

# How do they Work?

- Bots are executed on save
- Data updates done via bots can be seen on the Edit History of records. User who does the updates is the Bots User.
- Bots can be disabled or deleted

# How do you configure a bot?

- Add a Bot via the REST API
- Bots can have the following properties
  - Name
  - Conditions - When do you want the bot to execute?
  - Outcomes - What do you want the Bot to do?

```
1 const closeJob: Bot = {
2   name: 'Job Closed',
3   botType: 'POSTSAVE',
4   isEnabled: true,
5   onUpdate: true,
6   entity: 'JobOrder',
7   botConditionData: [
8     botConditions: [
9       {
10         entity: 'JobOrder',
11         field: 'status',
12         matchOperator: 'EQUALS',
13         value: 'Closed',
14       },
15     ],
16   ],
17   logicOperator: 'AND',
18 },
19 botOutcomes: [
20   {
21     outcomeType: 'UPDATEENTITY',
22     data: {
23       isOpen: false,
24     },
25     entity: 'JobOrder',
26   },
27 }
```

When a job status is updated to “Closed” update isOpen field to “false”

```
const placementApproved: Bot = {
  name: 'Placement Approved',
  botType: 'POSTSAVE',
  isEnabled: true,
  onUpdate: true,
  entity: 'Placement',
  botConditionData: {
    botConditions: [
      {
        entity: 'Placement',
        field: 'status',
        matchOperator: 'EQUALS',
        value: 'Approved',
      },
    ],
    logicOperator: 'AND',
  },
  botOutcomes: [
    {
      outcomeType: 'UPDATEENTITY',
      entity: 'Candidate',
      data: {
        status: 'Placed',
      },
    },
  ],
}
```

When a placement status is updated to “Approved” update Candidate status to “Placed”

```
1
2 const placementDateEnd: Bot = {
3   name: 'Placement dateEnd',
4   botType: 'POSTSAVE',
5   isEnabled: true,
6   onUpdate: true,
7   entity: 'Placement',
8   botConditionData: [
9     botConditions: [
10       {
11         entity: 'Placement',
12         field: 'dateEnd',
13         matchOperator: 'ISUPDATED',
14       },
15     ],
16     logicOperator: 'AND',
17   },
18   botOutcomes: [
19     {
20       outcomeType: 'UPDATEENTITY',
21       entity: 'Candidate',
22       data: {
23         dateAvailable: '$dateEnd+1',
24       },
25     },

```

When a placement dateEnd is updated, update candidate dateAvailable to dateEnd + 1.

```
4
5   botType: 'POSTSAVE',
6   isEnabled: true,
7   onUpdate: true,
8   entity: 'Placement',
9   botConditionData: {
10     botConditions: [
11       {
12         entity: 'Placement',
13         field: 'payRate',
14         matchOperator: 'ISUPDATED',
15       },
16       {
17         entity: 'Placement',
18         field: 'billRate',
19         matchOperator: 'ISUPDATED',
20       },
21     ],
22     logicOperator: 'OR',
23   },
24   botOutcomes: [
25     {
26       outcomeType: 'UPDATEENTITY',
27       entity: 'Candidate',
28       data: {
29         hourlyRate: '$payRate',
30       },
31     },
32     {
33       outcomeType: 'WEBHOOK',
34       url: 'https://us-central1-engage-shifts.cloudfunctions.net/updateADP',
35       httpMethod: 'POST',
36       queryParams: {
37         placementID: '$id',
38         candidateID: '$candidate.id',
39         payRate: '$payRate',
40         billRate: '$billRate',
41       },
42     },
43   ],
44 }
```

If placement pay rate or  
bill rate was updated,  
update corresponding  
record in ADP or  
PeopleSoft.

```
1
2 const placementApproved: Bot = {
3   name: 'Placement Approved',
4   botType: 'POSTSAVE',
5   isEnabled: true,
6   onUpdate: true,
7   entity: 'Placement',
8   botConditionData: {
9     botConditions: [
10       {
11         entity: 'Placement',
12         field: 'status',      Brian Kimball, 13 days ago • feat(Bots): Added s
13         matchOperator: 'EQUALS',
14         value: 'Approved',
15       },
16     ],
17     logicOperator: 'AND',
18   },
19   botOutcomes: [
20     { ...
21     },
22     {
23       outcomeType: 'WEBHOOK',
24       url: 'https://us-central1-engage-shifts.cloudfunctions.net/sendToSlack',
25       httpMethod: 'POST',
26       queryParams: {
27         entity: 'Placement',
28         entityId: '$id',
29         status: '$status',
30         candidate: '$candidate.name',
31         job: '$jobOrder.title',
32         startDate: '$dateBegin',
33       },
34     },
35   ],
36 },
37 },
38 },
39 },
40 }
```

If a placement is  
approved, send me a  
slack notification!

```
1 const { SLACK_WEBHOOK_URL } = require('./config.json');
2 const { IncomingWebhook } = require('@slack/client');
3
4
5 const webhook = new IncomingWebhook(SLACK_WEBHOOK_URL)
6
7 // subscribe is the main function called by Cloud Functions.
8 exports.sendToSlack = (req, res) => {
9   return Promise.resolve()
10  .then(() => {
11    if (req.method !== 'POST') {
12      const error = new Error('Only POST requests are accepted');
13      error.code = 405;
14      throw error;
15    }
16
17    // Send message to Slack.
18    const message = createSlackMessage(req.query);
19    return webhook.send(message, (result) => {
20      console.warn('Sent', result);
21      return res.status(200).send('SUCCESS!');
22    });
23  }).catch((err) => {
24    console.error(err);
25    return res.status(err.code || 500).send(err);
26  });
27}
```

## Building a webhook

# Tasks, Emails and more..

- Add tasks and send emails using bots
  - When a job is lost update submission status to “Lost to Competition”. **Also, send an email to the job owner that the job was lost.**
  - When a placement dateEnd is updated, update candidate dateAvailable to dateEnd + 1. **Also, add a task for the recruiter with a due date closer to the candidate’s dateAvailable to place the candidate in a new job.**
- A single App to rule them all
  - Build a single webhook application to handle requests for all your clients. Session relevant to the client can be created using auth code passed into the webhook by the bot

# Resources



[bullhorn/extension-configuration-starter](#)



[bullhorn/bot-samples](#)

What would your  
bot do?



# Field Interactions

ENGAGE



Nothing new... wah wah

# Recap

- Exist on all Add/Edit Forms
- Allow you to modify fields based on user input
- Hide/Show/Add fields into any form!
- Documentation inside NovoElements
  - <http://bullhorn.github.io/novo-elements/#/field-interactions>

# Page Interactions

ENGAGE

# What can they do?

modifying your Bullhorn experience



- Modify Record tabs
- Modify Record actions
- Modify Overview fields
- Pre/post saving on add/edit forms
- Customize workflow icons
- More to come!
  - Fast Find / Fast Add / etc...

What can they modify?

# Record Tabs label/style/visibility/disable

The screenshot shows the Bullhorn software interface for managing company records. At the top, there's a navigation bar with the Bullhorn logo, search, and add functions. Below it, a blue header bar displays the company name "C W D C Metal Fabricators". The main content area is divided into two columns. The left column contains detailed company information, and the right column contains a notes section and other details. A red box highlights the top navigation bar where tabs are located.

**Top Navigation Bar:**

- Bullhorn logo
- Find
- Add
- Privacy
- Help
- Novo SL9

**Header Bar:**

- Menu
- C W D C Metal Fabricators
- ACTIONS

**Record Tabs:** (highlighted by a red box)

- OVERVIEW
- EDIT
- ACTIVITY
- EMAILS
- NOTES (0)
- BHTEST1
- FILES (0)
- SUBMISSIONS (0)
- BLUEPRINT
- PULSE
- CONTACTS (2)
- INVOICE
- CERTIFICATIONS
- TEST

**Left Column (Company Overview):**

- COMPANY OVERVIEW
- COMPANY DESCRIPTION
- OWNERS: rest.sl9 rest.sl9  
S Release SL9
- CHILD CLIENT CORPORATIONS
- # OF EMPLOYEES: 0
- # OF OFFICES: 1
- OWNERSHIP: Select ▾
- YEAR FOUNDED
- TAX %: 0.00%
- INVOICE FORMAT
- CULTURE / PERKS
- EXTERNAL ID
- TICKER SYMBOL
- FUNDING STATUS
- ANNUAL REVENUE (MILLIONS): US\$0.00
- COMPETITORS
- CLIENT CONTACT NOTES
- ADDRESS: United States

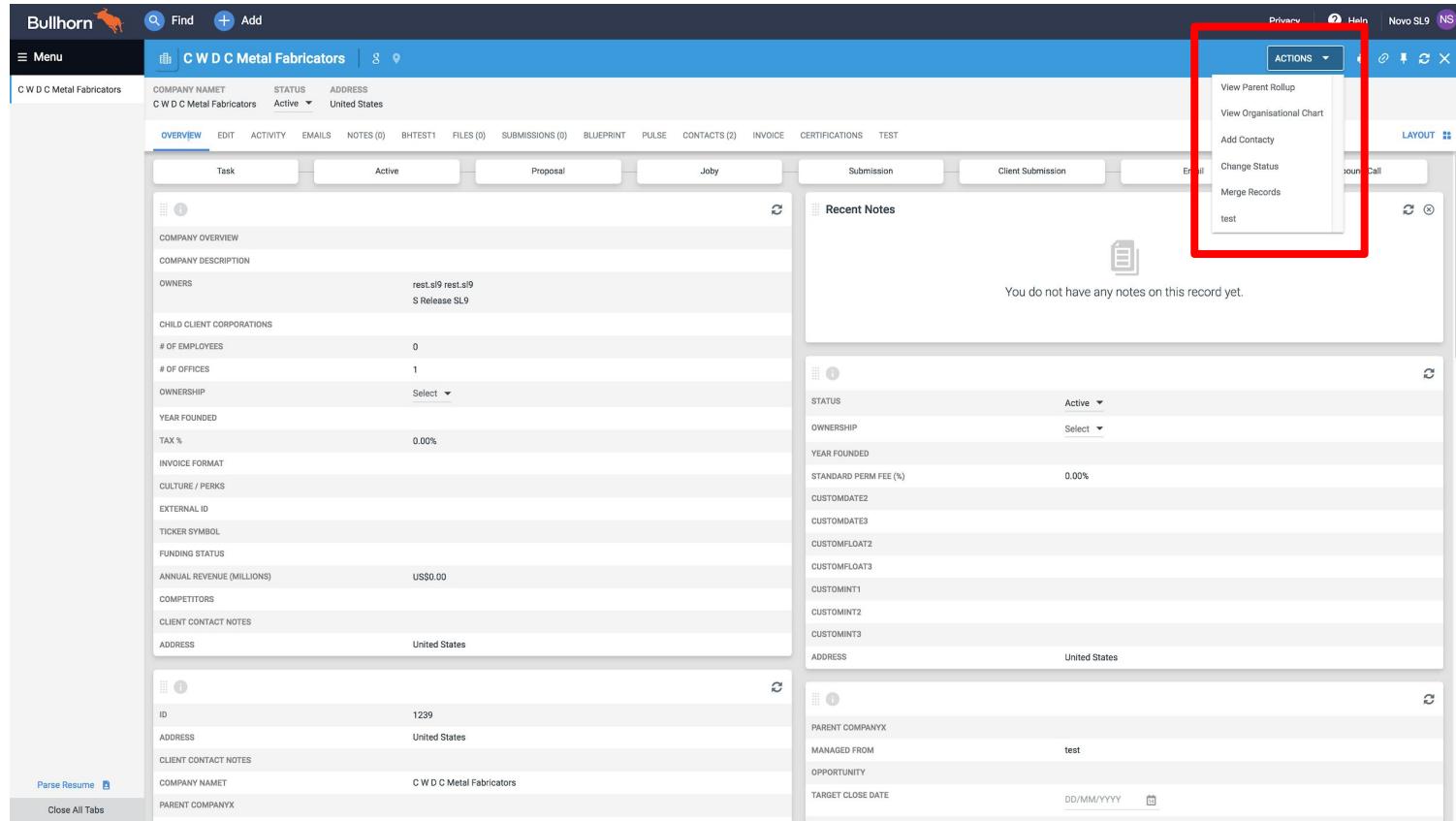
**Right Column (Notes and Details):**

- Recent Notes: You do not have any notes on this record yet.
- STATUS: Active ▾
- OWNERSHIP: Select ▾
- YEAR FOUNDED
- STANDARD PERM FEE (%): 0.00%
- CUSTOMDATE2
- CUSTOMDATE3
- CUSTOMFLOAT2
- CUSTOMFLOAT3
- CUSTOMINT1
- CUSTOMINT2
- CUSTOMINT3
- ADDRESS: United States
- PARENT COMPANYX
- MANAGED FROM: test
- OPPORTUNITY
- TARGET CLOSE DATE: DD/MM/YYYY

**Bottom Left Buttons:**

- Parse Resume
- Close All Tabs

# Record Actions label/style/visibility/disable



The screenshot shows a Bullhorn software interface for managing company records. The main view displays details for 'C W D C Metal Fabricators', including its name, status (Active), address (United States), and various operational metrics like employees and offices. On the right side, there's a 'Recent Notes' section which is currently empty. A red box highlights the 'ACTIONS' dropdown menu, which contains several options: View Parent Rollup, View Organisational Chart, Add Contact, Change Status, Merge Records, and test.

**ACTIONS**

- View Parent Rollup
- View Organisational Chart
- Add Contact
- Change Status
- Merge Records
- test

**Recent Notes**

You do not have any notes on this record yet.

**Company Overview**

**Company Description**

**Owners**: rest.sl9 rest.sl9  
S Release SL9

**Child Client Corporations**

**# of Employees**: 0

**# of Offices**: 1

**Ownership**: Select ▾

**Year Founded**

**Tax %**: 0.00%

**Invoice Format**

**Culture / Perks**

**External ID**

**Ticker Symbol**

**Funding Status**

**Annual Revenue (Millions)**: US\$0.00

**Competitors**

**Client Contact Notes**

**Address**: United States

**ID**: 1239

**Address**: United States

**Client Contact Notes**

**Company Name**: C W D C Metal Fabricators

**Parent Company**: X

**Status**: Active ▾

**Ownership**: Select ▾

**Year Founded**

**Standard Perm Fee (%)**: 0.00%

**Custom Date 2**

**Custom Date 3**

**Custom Float 2**

**Custom Float 3**

**Custom Int 1**

**Custom Int 2**

**Custom Int 3**

**Address**: United States

**Parent Company**: X

**Managed From**: test

**Opportunity**

**Target Close Date**: DD/MM/YYYY

# Activity Sections visibility/style

The screenshot displays the Bullhorn software interface for managing company records. A red box highlights the 'ACTIVITY' section at the bottom of the page.

**Top Navigation:** Bullhorn logo, Find, Add, Privacy, Help, Novo SL9, NS.

**Header:** C W D C Metal Fabricators, COMPANY NAME, STATUS, ADDRESS, Active, United States.

**Toolbar:** OVERVIEW, EDIT, ACTIVITY, EMAILS, NOTES (0), BHTEST1, FILES (0), SUBMISSIONS (0), BLUEPRINT, PULSE, CONTACTS (2), INVOICE, CERTIFICATIONS, TEST, ACTIONS, LAYOUT.

**RELATED RECORDS:**

- Contacts (2):** Last Activity 27/10/2017. Items per page: 10. Columns: Date Added, Name, Occupation, Owner, Status, Direct Phone. Rows:
  - 27/10/2017 13:49 Roma\_Fname ClientContact Alvin2017-10-27 13:48:09.477 rest.s19 rest.s19 New Lead
  - 18/04/2016 12:02 Leslie Threats S Release SL9 Imported 914-861-9748
- Jobs (0)**
- Placements (0)**
- Opportunities (0)**
- Leads (0)**

**ACTIVITY:**

- Joby Activity (2):** Email (0), Notes (0), Appointments (0), Tasks (0).

**Bottom Left:** Parse Resume, Close All Tabs.

# Overview Fields label/value/visibility/style

The screenshot shows the Bullhorn software interface for managing company profiles. The main title bar includes the Bullhorn logo, a search icon, and an 'Add' button. The top navigation bar has tabs for 'Menu', 'C W D C Metal Fabricators', and other options like 'Find', 'Help', and 'Novo SL9'. Below the navigation is a toolbar with icons for 'Actions' (Print, Copy, etc.).

The main content area displays the 'C W D C Metal Fabricators' record. On the left, there's a sidebar with sections like 'OVERVIEW', 'EDIT', 'ACTIVITY', 'CONTACTS', 'NOTES (0)', 'BRIEFS (1)', 'FILES (0)', 'SUBMISSIONS (0)', 'BLOCK PRINT', 'PURCHASE', 'CONTACTS (2)', 'INVOICE', 'CERTIFICATIONS (1)', and 'TICKET'. The 'OVERVIEW' tab is selected.

The central part of the screen shows various fields for the company profile. A red box highlights the top row of buttons: 'Task', 'Active', 'Proposal', 'Joby', 'Submission', 'Client Submission', 'Email', and 'Outbound Call'. Below this, the 'COMPANY OVERVIEW' section contains fields for 'OWNERS', 'CHILD CLIENT CORPORATIONS', 'YEAR FOUNDED', 'TAX %', 'INVOICE FORMAT', 'CULTURE / PERKS', 'EXTERNAL ID', 'TICKER SYMBOL', 'FUNDING STATUS', 'ANNUAL REVENUE (MILLIONS)', 'COMPETITORS', 'CLIENT CONTACT NOTES', 'ADDRESS', 'ID', 'ADDRESS', 'CLIENT CONTACT NOTES', 'COMPANY NAME', and 'PARENT COMPANYX'. The right side of the screen shows a 'Recent Notes' section with a message: 'You do not have any notes on this record yet.' and a status section with fields for 'STATUS', 'OWNERSHIP', 'YEAR FOUNDED', 'STANDARD PERM FEE (%)', 'CUSTOMDATE2', 'CUSTOMDATE3', 'CUSTOMFLOAT2', 'CUSTOMFLOAT3', 'CUSTOMINT1', 'CUSTOMINT2', 'CUSTOMINT3', and 'ADDRESS'.

At the bottom, there are buttons for 'Parse Resume' and 'Close All Tabs'.

# Add/Edit Form Pre/Post-Save validation/toast/redirect

Bullhorn Find + Add Privacy Help Novo SL9 NS

C W D C Metal Fabricators | g

OVERVIEW EDIT ACTIVITY EMAILS NOTES (0) BI TEST1 FILES (0) SUBMISSIONS (0) BLUEPRINT PULSE CONTACTS (2) INVOICE CERTIFICATIONS TEST LAYOUT

COMPANY NAME: C W D C Metal Fabricators 25/100

PARENT COMPANY:

MANAGED FROM: test

STATUS: Active

TYPE:

OPPORTUNITY: USD

TARGET CLOSE DATE:

COMPANY URL:

FACEBOOK PROFILE NAME:

TWITTER HANDLE:

BILLING FREQUENCY:

WORK WEEK BEGIN:

BUSINESS SECTORS:

Parse Resume Close All Tabs

SAVE ✓

+ Add

- Company
- Contacty
- Lead
- Candidate
- Opportunity
- Joby
- Placement
- Note
- Task
- Distribution List

# How do they work?

let's use a Record Tab PI as our example

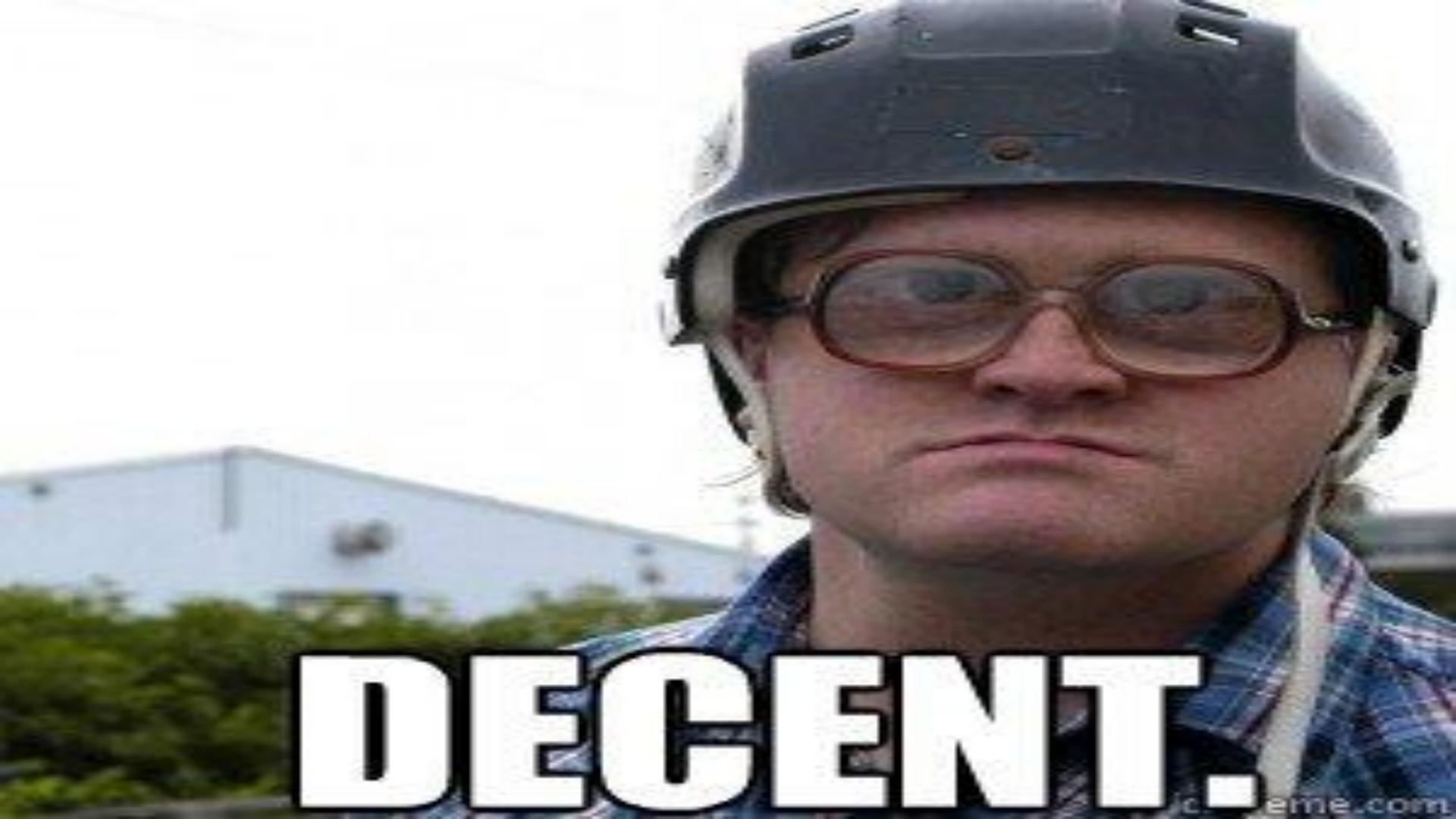
- Fetch all PI on page load
- Block rendering of Record tabs until PI are ran
- Pass an array of objects that represent the tabs to render to PI
- Execute PI in order, asynchronously
- PI modify the objects inside the array
- Record tabs are displayed

How are they managed?

[Find](#)[Add](#)[Privacy](#)[Help](#)

Johnathan Novo III

[☰ Menu](#)[Parse Resume](#)[Close All Tabs](#)



**DECENT.**

ceme.com

# Novo-Elements

ENGAGE

# What is new?

- Data Tables
- Accordion
- Enhanced Pickers / Forms
- A lot more!!



## Change Pagination Style

Standard	Basic
----------	-------

## Toggle Global Search

Show	Hide
------	------

## Configure Columns

**CONFIGURE COLUMNS**

## Configure Columns

**SHOW ROW DETAILS (FIRST TABLE)**

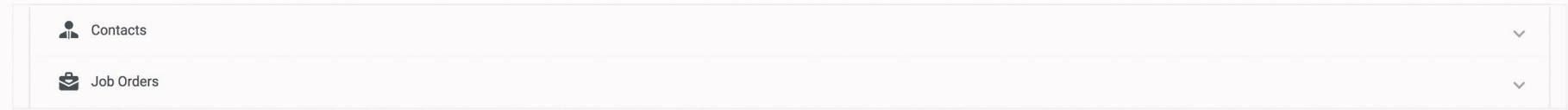
**HIDE ROW DETAILS (FIRST TABLE)**

## PASSING AN ARRAY OF ROWS

Actions		ID	Status	Enabled	Action
<input type="checkbox"/>	>	0	(1) Status 0	true	
<input type="checkbox"/>	>	1	(1) Status 1	false	
<input type="checkbox"/>	>	2	(1) Status 2	true	
<input type="checkbox"/>	>	3	(1) Status 3	false	
<input type="checkbox"/>	>	4	(1) Status 4	true	
<input type="checkbox"/>	>	5	(1) Status 5	false	
<input type="checkbox"/>	>	6	(1) Status 6	true	
<input type="checkbox"/>	>	7	(1) Status 7	false	
<input type="checkbox"/>	>	8	(1) Status 8	true	
<input type="checkbox"/>	>	9	(1) Status 9	false	

## THEMED EXPANSION PANEL

This is an example of a themed list.



```
<novo-expansion-panel theme="contact" padding="false">
  <novo-expansion-panel-header>
    <novo-panel-title>
      <novo-icon>person</novo-icon>
      <label>Contacts</label>
    </novo-panel-title>
  </novo-expansion-panel-header>

  <novo-table class="table-condensed">
    [rows] = "details.rows"
    [columns] = "details.columns"
    [config] = "details.config"
    rowIdentifier = "name"
  </novo-table>
```

# Lots more to come!

- Angular 6!
- Revamped demos / examples!
- Much more!!!



<http://bullhorn.github.io/novo-elements/>

# Bullhorn CLI

ENGAGE

- Accelerate Development
- Standardize Workflow
- Build System Support  
(Jenkins)
- **Extension Management**

# Why did we build it?

bullhorn [namespace] <action>

# What can it do?

`bullhorn [namespace] <action>`

- Accelerate Development
- Standardize Workflow
- Build System Support (Jenkins)
- **Extension Management**

HOW DOES IT WORK?

# Installation

→ `npm install -g @bullhorn/bullhorn-cli`

→ `bullhorn --version`

→ `bullhorn --help`

# Authentication

```
→ bullhorn auth login
```

```
? Username: bkimball
```

```
? Password: [hidden]
```

```
Authorizing...
```

```
Authorization Complete: bf0fce1b-cdbb-4503-be49-700921b58961
```

```
→ bullhorn auth login -u bkimball -p [hidden]
```

# Configuration

- `bullhorn config get environment`  
`environment` was `set` to `https://universal.bullhornstaffing.com`
- `bullhorn config set environment https://universal.bullhorns...`  
`environment` was `set` to `https://universal.bullhornstaffing.com`

# Extensions

→ **bullhorn extensions --help**

Usage: extensions [options] [**command**] <action>

*commands to manage extensions*

**Options:**

-h, --**help** *output usage information*

**Commands:**

extract      *Extract an extension from the extension config JSON file*

list          *List installed extensions*

upload        *Upload an extension after extracting*

# Extensions List

→ `bullhorn extensions list`

`finding extensions...`

<code>id</code>	<code>name</code>	<code>description</code>	<code>numExts</code>
--	--	--	--
50	Shift Scheduling	null	3
55	Test Extension	null	1
56	Integration Test	null	2
67	PS-Test 1	null	5
74	Custom Cards	null	2

# typings for TypeScript Support

→ `bullhorn` typings generate

Or

→ `npm install --save @bullhorn/bullhorn-types`

*“That's all Folks!”*



- [bullhorn extensions](#) \*
- [bullhorn functions](#) \*
- [bullhorn events](#) \*
- [bullhorn data](#) \*

<https://github.com/bullhorn/bullhorn-cli>

ENGAGE

# Questions?



#BullhornEngage

ENGAGE