

# Let's build a Twubric!

## 1. Introduction

### 1.1 What is a Rubric?

In a learning setting, a scoring rubric, typically, has a predefined set of weighted attributes mapped to a custom scale that together aid instructors to assess the quality of student submissions in an effective way.

Sometimes, these rubrics are defined by organizations (such as schools and universities) or domains under organizations (such as departments Eg. Department of IT) to keep evaluation of students (and employees) consistent and transparent.

Eg. A simple rubric for grading an essay worth **10 points** could be:

*Example 1*

Criteria	Weightage	Scale
Handwriting	1	Bad, Average, Good
Understanding	5	Limited, Some, In-depth, Extraordinary
Concision	4	Too short, Too long, Apt

As you can imagine, a more complex rubric can probably have criteria groups, a hierarchical scale and so on. But we'll just be using a simple rubric like the one in the example above for our problem.

### 1.2 Problem Statement

If you have an account on Twitter, you'd probably know that apart from normal people, there are:

1. bots
2. accounts that have zero activity. These were probably created by people unknowingly only for them to never come back and use it again.

While it is flattering to have these *tweeps* follow you, you'd want to weed them out at some point, I think (I would). So:

We want to build a simple web app that allows a Twitter user to *review* his followers, view their twubric "scores" (see Example 2), filter & sort them and then decide to optionally remove certain followers.

*Example 2 (Twubric)*

Criteria	Weightage	Scale
Friends	2	High, Average, Low
Influence	4	High, Average, Low
Chirpy	4	High, Average, Low

## 1.3 API Response

A mock API response in JSON format is given here -

<https://gist.github.com/pandemonia/21703a6a303e0487a73b2610c8db41ab/raw/82e3ef99cde5b6e313922a5ccce7f38e17f790ac/twubric.json>

You can save this data in a JSON file and load it using React.

## 1.4 UI mockup

Below is a skeletal UI for the Twubric page.

*<SampleUI>*

**Sort By**

Twubric Score	Friends	Influence	Chirpiness
---------------	---------	-----------	------------

**Joined Twitter between**

Start Date	End Date
↓	↓

Sample User One			3.5
1 Friends	2 Influence	0.5 Chirpiness	
Jan 23, 2013	Remove		

Sample User Two			5
2 Friends	2 Influence	1 Chirpiness	
Feb 13, 2012	Remove		

Sample User Three			7
2 Friends	2 Influence	3 Chirpiness	
Mar 11, 2010	Remove		

*</SampleUI>*

Some pointers about the UI.

1. The UI displays a list of the user's Twitter followers in a grid. Each item in the grid can be displayed as shown in the mockup above.

2. Clicking on the date filter field should display a date popup from which the user may select a From date and a To Date. (Any date picker plugin of your choice). The aim is to allow the user to filter the list based on the dates when their followers joined Twitter.
3. The buttons under "Sort By" could be a toolbar or button group (See [this](#) for reference). Clicking once sorts the grid in ascending order by that criteria. Clicking again sorts the grid in descending order by that criteria. Clicking on one criteria, deactivates the others. The idea is to allow the user to view his list of followers in whichever order he chooses. You're welcome to design an alternative UI as long as it achieves this aim.
4. Clicking on "Remove" must remove the user from the listing.
5. Feel free to use any CSS framework for basic styling ([Foundation](#), [Bootstrap](#), anything works!)

## 2. Requirements

1. Create a web app for the above problem using React.
2. If using React, create the above web app using the architecture.
3. Do not use jQuery to handle inputs or the sorting - we are looking for a solution that uses React features and functionality for building the above UI. You can use any available public library in your solution
4. Your solution code should be compatible with [ES6](#).
5. The above mockup is a barebones UI for displaying the overall structure of the view. Any alternate display or view that conveys the same data and has the same functionality will also be accepted

### 2.1 Bonus Points

You get bonus points if you can do any (or all if you're feeling adventurous!) of the following...

- Create an alternate view (like a cool visualization) instead of the boring grid layout to convey the info
- Adding keyboard shortcuts for sorting, selecting a user and removing

## 3. Deliverables

1. A web app that meets the requirements in the Requirements section above.
2. An optional README.txt file that describes your solution.
3. Create a Git/Mercurial repository. You can share the repository with us via BitBucket or GitHub once you submit the app. Our BitBucket/Github id is teamiecareers.

## 4. Evaluation

Here's how we'll be evaluating your solution.

Fetching and listing users	15%
Structuring components and data interactions efficiently	25%
Implementing sort, filter & remove actions	30%
Implementation of the UI, overall design and UX	15%
Code Quality + compliance to coding standards	15%

## 5. Support

If you are not clear on something, feel free to send us an email at [info@exhibytesolution.com](mailto:info@exhibytesolution.com)