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 API that allows a Twitter user to *review* his followers, and their

“scores” based on a rubric (see Example 2)

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, in case you are not able to create a Twitter developer account - https://gist.githubusercontent.com/pandemonia/21703a6a303e0487a73b2610c8db41ab/raw/9667fc19a0f89193e894da7aaadf6a4b7758b45e/twubric.json

Make a **network request** from your Twubric app to fetch the above json data.

1.4 Modelling a Rubric

If you were to model *Example 1* using pseudo code, one possible solution could look something like:

```
// A rubric is a set of criterion with an absolute maximum score.
// Σ criteria.weightage <= points
Rubric {
  Criteria[] criteria,
  double points
}

// Each criterion has a label (Eg. Handwriting), a weightage (eg. 1) and is
// mapped to a “Scale”.
Criteria {
  string label,
  double weightage,
  Scale scale
}

// A scale has a set of attributes and a maximum. (could be 100% if it is a
// percentage scale or an absolute double number otherwise)
Scale {
```

```
  ScaleAttribute[] attribute,
  double max
}

// Each attribute has a label (eg. Extraordinary) optionally has a lower and
// upper range both of which are less than Scale.max (eg. 90% - 100%)
ScaleAttribute {
  numeric lowerRange,
  numeric upperRange,
  string label
}
```

Two things to note here are:

1. There are *other* ways to model a rubric.
2. I have ignored the model(s) for keeping tracking of user “scores”.

2. Requirements

1. A web page (`/app`) from which I can login to the app using my Twitter account.

2. On logging in successfully, I’m shown a list of my Twitter followers (`/app/followers`). I click on one follower to go to 3.

Note: Fetching the data using the Twitter API should be done only once (on login), and all data that is required for this view and the next should be stored on the web site.

3. An endpoint (`/app/follower/(id)/twubric.json`) where I can see specific profile information about the follower (like the ones highlighted from the Twitter API response below).

```
{
 ...
 "followers_count": 110,
 "friends_count": 202,
 "listed_count": 10,
 ...
 "favourites_count": 55,
 "statuses_count": 1102
 ...
 "twubric": {
   "total": 6,            // user’s twubric score out of 10
   "friends": 1.5,        // user’s friend score out of 2
   "influence": 3,        // user’s influence score out of 4
   "chirpy": 2,           // user’s chirpiness score out of 4
  }
```

```
  ...
}
```

3. Deliverables

1. A Drupal 7 module (and other contributed module dependencies if any) that together meet the requirements as described in the Requirements section above. We would like to see the following in your implementation:

- Incorporating Drupal's features - content type, entity etc. to model and store the data.
- *Bonus*: Using a Drupal view for showing the list of Twitter followers (requirement 2).
- *Bonus*: Using Drupal services for the endpoint (requirement 3).

If you're not using Drupal, then please submit all the required files for testing out your solution in a framework of your choice. Note that we are looking for a structured solution using features/components provided in the selected framework. *Ex*. a set of PHP files that interacts with a custom database table can meet the above requirements but will not be considered a full solution.

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 id is teamiecareers.

> **Note:**
> Part of being an efficient Drupal developer includes finding good contributed projects and extending them smartly to help solve problems at hand. So, I'd strongly urge you look for existing Drupal modules that you think will help you in this task and use them freely as long as the requirements are accurately met. Likewise, if you're using any other PHP framework, consider using existing contributed projects.

This is a useful article to understand the Drupal design paradigm.
http://www.garfieldtech.com/blog/mvc-vs-pac

4. Drupal and other References

Rubrics
Rubrics wiki, http://en.wikipedia.org/wiki/Rubric_(academic)
Rubric tools I, http://www.edtechteacher.org/index.php/teaching-technology/assessment-rubrics
Rubric tools II, http://www.screenr.com/8x3

Twitter
Twitter API, https://dev.twitter.com/docs/api/1.1
Twitter Developers, https://dev.twitter.com

Drupal
Requirements, http://drupal.org/requirements
Installing from the command line, http://www.coderintherye.com/install-drupal-7-using-drush
Drush project http://drupal.org/project/drush
Drupal module developers guide, http://drupal.org/developing/modules
Database API, http://drupal.org/developing/api/database
Drupal on Stackexchange, http://drupal.stackexchange.com

5. Evaluation

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

Delivering a working solution	20%
Integrating with Twitter APIs / Mock JSON (login + retrieving info)	30%
Logic for computing Influence, Chirpiness	30%
Overall Code Quality	20%

6. Support

If you are not clear on something, feel free to send us an email at careers@theteamie.com