Obtain and display data from a REST API using AngularJS

RESTful APIs are everywhere, providing access to data and services of all shapes and sizes. Some examples include

Twitter - https://dev.twitter.com/rest/public
Facebook - https://developers.facebook.com/docs/graph-api
Google Calendar - https://developers.google.com/google-apps/calendar/
Open Weather Map - http://openweathermap.org/api
SoundCloud - https://developers.soundcloud.com/docs/api/guide

Your mission, if you choose to accept, is to build a simple web application that uses AngularJS to obtain and render data from an API of your choice (you are not limited to the above). The following steps provide some guidelines and requirements to completing the challenge:

Requirements:

You must complete the following requirements to the best of your ability. Use your own judgement as much as possible to determine the adequacy of your own effort prior to submission.

IMPORTANT: First, register for an account with GitHub, if you don't already have one and create a public repository for your project. Email us with the link to that repository immediately.

In this project, you'll build an application to be served from NodeJS using AngularJS to do the following.

1. Make an AJAX call from within an Angular Controller. Render the data in a list/table like fashion on a web page.
   a. How you implement this is very flexible. You may add a button to make the request and render as a simple table.
   b. However you choose to render this data, use some visual elements from the following framework:
      i. https://material.angularjs.org/latest/#/
2. Refactor the above to execute within a Service, so that the AJAX request itself is happening in the Service, and not directly in the Controller, though the call to the service will happen from within the Controller.
3. Transform the data to AggieFeed Activities.
   a. Transform the data into a format that can be posted to AggieFeed. (IMPORTANT: pay close attention to the schema required to build an activity,

particularly what fields are required or optional). For more information about AggieFeed, see the description at the end of this exercise.

   b. AggieFeed API Reference Documentation: https://ucdavis.jira.com/wiki/display/IETP/Aggiefeed+REST+API
   c. ActivityStream Specification
      i. http://activitystrea.ms/specs/json/1.0/

4. Render the activities in a vertical stream-like fashion, and modify the AJAX request to insert new ‘activities’ at the top of the ‘stream’. Acquiring new data can be facilitated by a page refresh, a button on the page, or a function that is run on a timer of some sort, or just about any other way you can imagine.
5. Create custom directive to modify the contents of the activity rendered in your list.
   a. Modify contents within directive somehow
   b. Example: display age of post in top right hand corner (see https://aggiefeed.ucdavis.edu)

Project Setup

(Other than NodeJS and AngularJS, the exact tools or frameworks used for building your application are left to your discretion)

1. **NodeJS** - https://nodejs.org/download/ (NOTE: This will also install NPM)
2. **Grunt** - npm install -g grunt-cli
   a. http://gruntjs.com/getting-started
3. **Bower** - npm install -g bower
4. **GIT** - https://git-scm.com/download/
5. **Yeoman** (optional**)** - http://yeoman.io/
   a. For this exercise, you may use yeoman to setup your project.
      i. npm install -g yo
   b. For an intro to Yeoman, you can complete the following tutorial:
      i. http://yeoman.io/codelab/index.html
      ii. NOTE: To install the generator-angular, you may just want to run the following without the version specified.
         1. npm install --global generator-angular
6. **GitHub** -
   a. Set up this project as a public repository within GitHub.

Summary of Tasks

1. Install all of the required software listed in ‘Project Setup’
2. Complete the intro to yeoman (Optional): http://yeoman.io/codelab/index.html

3. Optional, but recommended:
    a. https://code.angularjs.org/1.5.11/docs/guide (See list of tutorials)
    b. https://code.angularjs.org/1.5.11/docs/tutorial (Official Tutorial, also included in list referenced in above link)
4. Complete all of the ‘Requirements’ listed above for building out a lightweight stream of activities

Background on AggieFeed

What is Aggie Feed?

Aggie Feed is a campus activity stream that displays information and events submitted by approved campus departments and organizations. This service of Information and Educational Technology (IET) provides date-driven content in real time, such as events and activities, WarnMe alerts, and SmartSite notifications as well as Quick Polls and online course evaluations. Aggie Feed provides content pertinent to students, faculty, and staff, and messages can be targeted directly to specific groups and individuals or made public for anyone to read.

Benefits of Aggie Feed

Aggie Feed can be used to post events, activities, messages and other content. It can help you simplify your processes for approving and distributing notices, plus reduce your reliance on mass emails. Aggie Feed can be used for internal communication and can also be posted on your own website.

Using Aggie Feed

Much like Twitter or Facebook, consumers of Aggie Feed can expect to see a stream of content with the most current posts at the top of the Feed. However, unlike similar services, Aggie Feed aggregates information specifically related to UC Davis and targets it directly to specific groups and individuals. When logged into Aggie Feed, an individual will see all the posts in their Feed by default, but can see different views by selecting a specific category such as Campus Life or Academics. Individuals can also personalize their feed by selecting which groups' posts will appear.