

Objective: Consume our API and store responses in a relational database.

What you'll need:

1. Docs: <http://docs.tapresearch.com/>
2. Base URL: <https://staging.tapresearch.com> -- don't use www.tapresearch.com.
3. Email: codetest@tap.com
4. API Token: 1c7dd6fd2a94b2e6431b367189aead01

Instructions:

1. Get all campaigns and store in DB
 - a. <http://docs.tapresearch.com/#list-your-campaigns>
 - b. Important attributes: length_of_interview, cpi, name, id
2. Get all quotas and qualifications and store in DB
 - a. <http://docs.tapresearch.com/#get-a-specific-campaign>
 - b. Important attributes: question_id, pre_codes
3. Define a custom route named ordered_campaigns inside the campaigns controller.
 - a. localhost:3000/campaigns/ordered_campaigns
 - b. Return an array of campaigns, including nested quotas and qualifications, ordered by campaign qualification count in JSON.

Notes:

1. This coding challenge must be completed using Ruby on Rails.
2. You won't need to create table columns for all API response attributes. The attributes listed above will suffice.
3. We expect 3 database tables
 - a. A campaign has many campaign_quotas
 - b. A campaign_quota has many campaign_qualifications
4. Store pre_codes as a comma-separated string inside the campaign_qualifications table.
 - a. This string may be longer than 255 characters.
5. Due to the simplicity of this test, you'll also be evaluated based on thoroughness and thoughtfulness of your solution. Make sure you take scalability and maintainability of the code into consideration.