

CS 169 – Discussion 2

HTTP & RESTful APIs

Working in pairs, read through and try to answer the following questions.

1. GitHub

- Without looking at any documentation, discuss what the following HTTP requests might do.
 - GET <https://github.com/orgs/cs169/repos>
Retrieve repos for organization 'cs169'
 - GET <https://github.com/repos/cs169/homework2>
Retrieve repo 'homework2' for organization 'cs169'
 - POST <https://github.com/orgs/cs169/repos>
Publish/create repo for organization 'cs169'
- Read through GitHub's API documentation, which can be found at <https://developer.github.com/v3>. Find answers to the following questions.
 - How can I get publicly available information about a user?
<https://developer.github.com/v3/users/#get-a-single-user>
Example: `curl https://api.github.com/users/cs169`
 - How can I list all pull requests of a repository?
<https://developer.github.com/v3/pulls/#list-pull-requests>
Example: `curl https://api.github.com/repos/rails/rails/pulls`
 - How can I get all closed pull requests of a repository?
Example: `curl https://api.github.com/repos/rails/rails/pulls?state=closed`
 - How can I create a new pull request?
<https://developer.github.com/v3/pulls/#create-a-pull-request>
This would be a POST request, and requires authentication/additional data in order to work correctly. `POST /repos/:owner/:repo/pulls`
 - Which input values are required to create a new pull request?
<https://developer.github.com/v3/pulls/#create-a-pull-request>
From the above link, required parameters: title, head, and base.

2. Finish the following ruby implementation for a simple todo API. Assume you have access to a params hash with any necessary query parameters, along with a `format_as_json(object)` function. Return results as JSON when applicable.

Reference: <http://sinatrarb.com/intro.html>

Solution:

```
require 'sinatra'
require 'json'

# Note: by default, sinatra hosts the server on http://localhost:4567
# Go to this page in your web browser to see your app running. However, the app
# does not define a route handler for get '/', so you will see Sinatra's error
# page. You can also test specific routes with curl from the command line.

#
# Data models
#
class User
  attr_reader :id
  attr_accessor :todo

  def initialize
    @id = app.get_new_user_id
    @todo = Todo.new
  end
end

class Todo
  def initialize
    @items = Array.new
  end

  def add item
    if not @items.include? item
      @items << item
    end
  end

  def delete item
    @items.delete item
  end

  def view_item item_id
    # assume each item instance has an accessible id field
    # and all items/ids are unique
    item = @items.find {|i| i.id = item_id }
    format_as_json item unless item.nil?
  end
end

### Continued below...
```

```
#
# Route handling (Sinatra)
#
get '/user/:id/todo' do
  user = get_user_by_id params[:id]
  user.todo.to_json
end

get '/user/:id/todo/:item_id' do
  user = get_user_by_id params[:id]
  todo = user.todo
  todo.view_item params[:item_id]
end

post '/user/:id/todo' do
  user = get_user_by_id params[:id]
  request.body.rewind
  raw_item = JSON.parse request.body.read
  user.todo.add raw_item
  201 # Resource created.
end

delete '/user/:id/todo/:item' do
  user = get_user_by_id params[:id]
  user.todo.delete params[:item]
  200 # Resource delete ok.
end
```