edX edge   **SUNY_Binghamton:** CS445 Software Engineering (U)

<div style="float:left">

‣ Getting
Started
Overview

‣ W0M:
Overview and
Introduction to
Software
Engineering
(Week 0,
Monday Aug.
31)

‣ W0W:
Beautifully
Engineered
Software, Plan
& Document
vs Agile (Week
0, Wednesday
Sept. 2)

‣ W1W:
Introduction to
Ruby (Week 1,
Wednesday
Sept. 9)

‣ W2W: More
Ruby and Intro
to BDD & TDD
(Week 2,
Wednesday
Sept. 16)

‣ W3M: SaaS
Architecture
and REST

</div>

PART 5: CONNECTING TO THE SERVICE (100/100 points)

And now the moment we've all been waiting for: *connecting to the actual Oracle Of Bacon service!!!*

Except we're not going to do that. Given that you're re-running tests every time you change the code, it would be inconsiderate (and for some sites, in violation of the terms of use of the API key) to hit the site every time you do so. Instead, we will use FakeWeb, which we'll learn more about later in the course, to fake a "canned" response by intercepting calls to Ruby's `Net::HTTP` library.

(Of course, this is only true when running tests. Your real code will really talk to OOB.)

Since we've already tested that the various types of XML responses are handled properly by the constructor of `OracleOfBacon::Response`, all we are testing here is the service connection itself, in `find_connections`. If successful, it should try to hand off the received blob of XML to the constructor for `Response`. If there's a network problem-- unfortunately, Ruby's HTTP library can raise many kinds of exceptions-- we convert any of them into a generic `OracleOfBacon::NetworkError` and re-raise that.

- Modify `find_connections` to handle both the successful response case and the network exception case. This should cause the specs under `describe 'service connection'` to pass.

Question for reflection:

- As mentioned earlier in the course, the .xml files under `spec` that contain the "canned" responses are sometimes called *fixtures*. How would you create these yourself when you're developing tests for your own RESTful service library? (Hint: the command-line utility `curl` may be useful.)

Again, please submit the lib/oracle_of_bacon.rb file for grading below:

[Choose Files] No file chosen

(Week 3,
Monday Sept.
21)

‣ W4M: SaaS
Architecture
and REST
(Week 4,
Monday Sept.
28)

‣ W4W: Rails
Intro (Week 4,
Wednesday
Sept. 30)

‣ W5M: Rails
cont. (Week 5,
Monday Oct.
5)

‣ W5W:
Enhancing
SaaS with
JavaScript
(Week 5,
Wednesday
Oct. 7)

‣ W6M: Agile
Methodology:
Working with
the Customer
(Week 6,
Monday Oct.
12)

‣ W6W: BDD
with
Cucumber and
Capybara

```
On Time
service connection
  should create XML if valid response [25 points]
  should raise OracleOfBacon::NetworkError if network problem [25
points]

Finished in 0.01577 seconds
2 examples, 0 failures
```

SUBMIT URL TO PAIRING VIDEO (10/10 points)

Please submit the URL to an unlisted youtube video recording of your
pairing session on this assignment below.

> just writing things

✔