


SUNY_Binghamton: CS445 Software Engineering (U)

▶ 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

PART 2: VALID INSTANCE OBJECT (100/100 points)

Before even attempting to find connections, an `OracleOfBacon` object must have at least one of `from` or `to` specified (if one is missing, it's assumed to be Kevin Bacon), and a nonblank API key (although we won't be able to tell if it's valid until we try hitting the service).

We use the `ActiveModel::Validations` module that is part of Rails to streamline our validity checks. It is a module that mixes in validity-checking methods for object attributes and gives our object an instance method `valid?` that checks all the constraints and returns true or false. We've included validity checks for the presence of non-blank `From`, `To`, and `APIKey` attributes in an `OracleOfBacon` instance.

1. In the root directory of this homework (the one containing `lib` and `spec` subdirectories), start `autotest`. It will look for tests in `spec` and for your code in `lib/oracle_of_bacon.rb`. If `autotest` appears to do nothing: Check that you are running it in the code's root directory (the one that has `lib` and `spec` as subdirectories) and that the `.rspec` file exists in this root directory.
2. Delete `,:pending => true` from the `describe 'instance'` block, which should immediately cause that set of tests to fail red. Visually inspect the failing specs to get a feel for how they are written.
3. Define the constructor for `OracleOfBacon` so that both `From` and `To` initially default to Kevin Bacon, rather than being blank. This should cause a subset of the validity specs to pass green.
4. Since it makes no sense to connect an actor to herself, we should also validate that the `From` and `To` fields aren't the same. `validate` takes a symbol naming a method that can perform a desired validation. Fill in the method `from_does_not_equal_to` to perform this validation. Note: you'll have to check the documentation for `ActiveModel::Validation` and `ActiveModel::Errors` at api.rubyonrails.org to learn how to record your own error message "From cannot be the same as To", which the spec (test) expects to be generated in this case.

(Week 3,
Monday Sept.
21)

When you've done the above steps correctly, all the specs in `describe 'instance'` should pass green. Please submit the `lib/oracle_of_bacon.rb` file for grading:

Choose Files No file chosen

```
On Time
instance
  when new [20 points]
    should not be valid
  when only From is specified [20 points]
    should be valid
  from
    should == "Carrie Fisher"
  to
    should == "Kevin Bacon"
  when only To is specified [20 points]
    should be valid
  from
    should == "Kevin Bacon"
  to
    should == "Ian McKellen"
  when From and To are both specified
    and distinct [20 points]
    should be valid
  from
    should == "Carrie Fisher"
  to
    should == "Ian McKellen"
  and the same [20 points]
    should not be valid

Finished in 0.06776 seconds
11 examples, 0 failures
```

▶ 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

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 wrting things

