

[Courseware \(/courses/BerkeleyX/CS169.1x/2013_Spring/courseware/\)](/courses/BerkeleyX/CS169.1x/2013_Spring/courseware/)[Course Info \(/courses/BerkeleyX/CS169.1x/2013_Spring/info/\)](/courses/BerkeleyX/CS169.1x/2013_Spring/info/)[Syllabus \(/courses/BerkeleyX/CS169.1x/2013_Spring/syllabus/\)](/courses/BerkeleyX/CS169.1x/2013_Spring/syllabus/)[Textbook & VM \(/courses/BerkeleyX/CS169.1x/2013_Spring/textbook_vm/\)](/courses/BerkeleyX/CS169.1x/2013_Spring/textbook_vm/)[Tutorials & Resources \(/courses/BerkeleyX/CS169.1x/2013_Spring/tutorials_resources/\)](/courses/BerkeleyX/CS169.1x/2013_Spring/tutorials_resources/)[Discussion \(/courses/BerkeleyX/CS169.1x/2013_Spring/discussion/forum/\)](/courses/BerkeleyX/CS169.1x/2013_Spring/discussion/forum/)[Wiki \(/courses/BerkeleyX/CS169.1x/2013_Spring/course_wiki/\)](/courses/BerkeleyX/CS169.1x/2013_Spring/course_wiki/)[Progress \(/courses/BerkeleyX/CS169.1x/2013_Spring/progress/\)](/courses/BerkeleyX/CS169.1x/2013_Spring/progress/)

HOMEWORK 3: BEHAVIOR-DRIVEN DESIGN

In this homework you will create user stories to describe a feature of a SaaS app, use the Cucumber tool to turn those stories into executable acceptance tests, and run the tests against your SaaS app.

Specifically, you will write Cucumber scenarios that test the happy paths of parts 1-3 of HW 2. The GitHub repo

```
saasbook/hw3_rottenpotatoes
```

contains a "canonical" solution to HW 2 against which to write your scenarios. It also contains the necessary scaffolding for the first couple of scenarios. We suggest you first fork that repo on GitHub, then clone from your own fork:

```
git clone git@github.com:YourGitHubAccount/hw3_rottenpotatoes.git
```

We recommend that you do a

```
git commit
```

as you get each part working. As an optional additional help, git allows you to associate tags -- symbolic names -- with particular commits. For example, immediately after doing a commit, you could say

```
git tag hw3-part1b
```

and thereafter you could use

```
git diff hw3-part1b
```

to see differences since that commit, rather than remembering its commit ID. Note that after creating a tag in your local repo, you need to say

```
git push origin --tags
```

to push the tags to a remote. (Tags are ignored by deployment remotes such as Heroku, so there's no point in pushing tags there.)

[Show Discussion](#)[New Post](#)



[Find Courses \(/courses\)](/courses) [About \(/about\)](/about) [Blog \(http://blog.edx.org/\)](http://blog.edx.org/) [Jobs \(/jobs\)](/jobs) [Contact \(/contact\)](/contact)



<http://youtube.com/user/edxonline>



<https://plus.google.com/108235383044095082735>



<http://www.facebook.com/EdxOnline>



<https://twitter.com/edXOnline>

© 2013 edX, some rights reserved.

[terms of service \(/tos\)](/tos)

[privacy policy \(/privacy\)](/privacy)

[honor code \(/honor\)](/honor)

[help \(/help\)](/help)