

About Me

- Rising Junior majoring in Computer Engineering and History at NC State
- Started at SAS in 2019 as a Summer Intern in Performance and Integration Testing reporting to Susan Bartholow
- Interested in pursuing a Ph.D. in Computer Science with a focus in security to prepare for a career in research

Projects

- Test Rails Reporting Website
- Kubernetes deployment tests for SAS Viya
- Assisting with the automation of frequently run deployment tests
- Katacoda: Kubernetes, Docker, and Containers
- SAS Course: Programming 1
- SAS Course: Programming 2: Data Manipulation Techniques



Contact Information

- SAS: Elijah.Bouma-Sims@sas.com (919) 531-5457
- College: erboumas@ncsu.edu (205) 383-8054
- www.ElijahBoumaSims.com
- www.linkedin.com/in/elijah-bouma-sims

Test Rails Reporting Website

Challenge

Our team uses a web interface called Test Rails to track and report results from our test efforts. While the tool is extremely useful, it can be difficult to find the information you are looking for, especially if you are not directly familiar with the project. Additionally, many parts of the site do not link together as one would expect. This reporting website aims to make the data in the interface easier to consume, particularly for interested people outside of the team, like executives.

Implementation

The website is written in Python 3.7 using the Flask web framework. It is served on a WSGI server written in pure Python called CheryPy. I chose Flask as the base for this web application because 1) I have a working familiarity with it and 2) Python is highly portable and easy to learn. The graphs for the website are generated using Bokeh, an interactive visualization library targeted at web browsers.

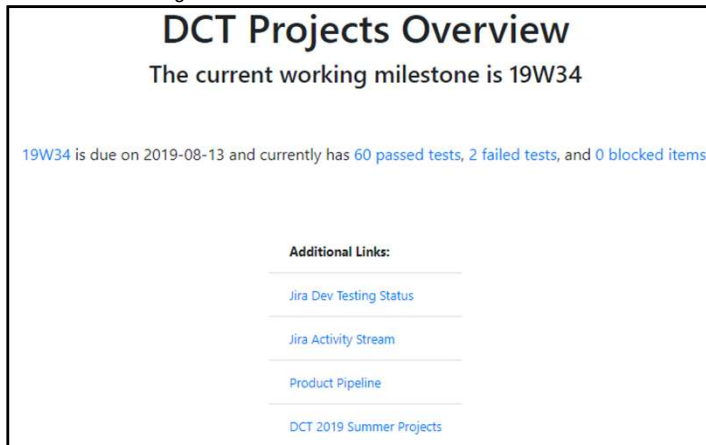
Challenges and Road Blocks

Much of the time dedicated to the project consisted of experimenting with the Test Rails API provided by Gurrock. While there is documentation, it has quirks which required testing to understand. Another significant challenge was the delay caused by pulling data from an external source (Test Rails). I had to implement background caching to ensure that users do not have to wait extended periods to view reports.

Implications / Applications

The webpage has been written in a way that, with configuration changes, it could be used for any Test Rails project. Many teams throughout SAS work with Test Rails, so this work has the potential to be used by those groups, in addition to the DCT team. Additionally, in the course of completing the project, I wrote several hundred lines of code to interface with the Test Rails API. This code has the potential to be used by others within the company who want to programmatically work with Test Rails.

Website Home Page:



Sample Graph:

