LIGHTBOT

An Adaptive Traffic Light Control Solution

TESTING POLICY DOCUMENT

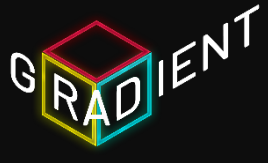

Table of Contents

INTRODUCTION

LightBot is primarily made up of two major subsystems.

A web application to view and control the system configuration, user and state data and a NodeJs Express server which serves as the API for the web application as well as the interface for running machine learning sub processes.

We have decided to use AWS solutions sufor our deployment environments. The server is deployed to an Elastic Compute machine combined with a load balancer whilst the web application is deployed on aws amplify console which has a built in continuous integration suite.

Due to the nature of our systems make-up and deployment, we have restricted upon the following testing technologies and policies for our components:

SERVER

- Technologies used - Postman, loadtest, Jest, Supertest
- Policies - services and functions are tested internally using jest endpoints are also tested internally by importing and mocking the server and database connections the models are also somewhat tested furthermore all endpoints are then tested by the external automated api suite Postman, which includes response time logging before being deployed to the EC2 server. Once deployed it is then tested with the load balancer using loadtest. It gains further integration testing during the deployment of the web application.

WEB APPLICATION

- Technologies used - Cypress.io and Cypress CI
- Policies - front end , unit and action tests as well as scores of smaller tests were carried out by the cypress.io testing server. Further more deployment and integration testing. Is carried out by cypress ci on the aws amplify deployment console.

REINFORCEMENT LEARNING CONTROLLER

- Technologies - Selenium , UnitTest
- Policies each time the machine learning controller was edited, adapted or changed to support more parameters, data outputs and visualizations it was test using the visualization testing suite selenium as well as unit tests were run on internal functions.

CONTINUOUS INTEGRATION

- Technology used - Cypress
- Policies - due to our extensive use of AWS solutions we have favoured Cypress CI for our continuous integration tests. It is also an extremely powerful and lightweight tool with a vast array of features. Our deployment pipeline contains cypress tests which initiate on each commit to master. This means that an entirety of end to end, front end, functionality, API and component integration tests are carried out before admitting the commit to live. Only once it is completely free of errors will it complete it's deployment to our domain https://lightbot.co.za