**Protect Test Writeup Skeleton**

Introduction

When adding features to existing software, tests will often break, necessitating their repair as part of the development process.

Repairing these broken tests is a task that takes up a considerable portion of development time.<Insert fact/citation>

The typical work pattern <flow?> goes something like this: Implement feature or bugfix, run test suite, examine test failures, determine if failure is due to a newly outdated test or actual app functionality has been broken, and repair either test/functionality or both. <this sentence needs a lot of work>

Several strategies and techniques have been developed and implemented to automate the resolution of these problems, typically ex post facto. <Elaborate, cite, examples, etc.>

With this project, we ask, and attempt to answer, whether we can take the information present in a test, and notify a developer of a potential breakage, either prior to or at the time that it occurs.

This project uses Selenium tests, which are used to test web applications from the browser’s perspective. The project is implemented as a plugin to the popular text editor Sublime Text. When a user edits HTML that will result in a Selenium test being broken, the user is notified of this fact, with an option to undo the changes leading to this breakage.

Background

From Wikipedia:

**Selenium** is a portable [software-testing](https://en.wikipedia.org/wiki/Software_testing) [framework](https://en.wikipedia.org/wiki/Software_framework) for [web applications](https://en.wikipedia.org/wiki/Web_application). Selenium provides a record/playback tool for authoring tests without the need to learn a test [scripting language](https://en.wikipedia.org/wiki/Scripting_language) (Selenium IDE). It also provides a test [domain-specific language](https://en.wikipedia.org/wiki/Domain-specific_language) (Selenese) to write tests in a number of popular programming languages, including [C#](https://en.wikipedia.org/wiki/C_Sharp_%28programming_language%29), [Groovy](https://en.wikipedia.org/wiki/Groovy_%28programming_language%29), [Java](https://en.wikipedia.org/wiki/Java_%28software_platform%29), [Perl](https://en.wikipedia.org/wiki/Perl), [PHP](https://en.wikipedia.org/wiki/PHP), [Python](https://en.wikipedia.org/wiki/Python_%28programming_language%29), [Ruby](https://en.wikipedia.org/wiki/Ruby_%28programming_language%29) and [Scala](https://en.wikipedia.org/wiki/Scala_%28programming_language%29). The tests can then run against most modern [web browsers](https://en.wikipedia.org/wiki/Web_browser). Selenium deploys on [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://en.wikipedia.org/wiki/Linux), and [OS X](https://en.wikipedia.org/wiki/OS_X) platforms. It is [open-source software](https://en.wikipedia.org/wiki/Open-source_software), released under the [Apache 2.0 license](https://en.wikipedia.org/wiki/Apache_License): web developers can download and use it without charge.

**Sublime Text** is a text-editor available for Windows, Mac OSX, and Linux. It supports syntax-highlighting and auto-completion for virtually all existing languages, as well as other optional features chosen ala-carte style, implemented as plugins. Sublime Text plugins are written in Python, and make use of the Sublime Text API.

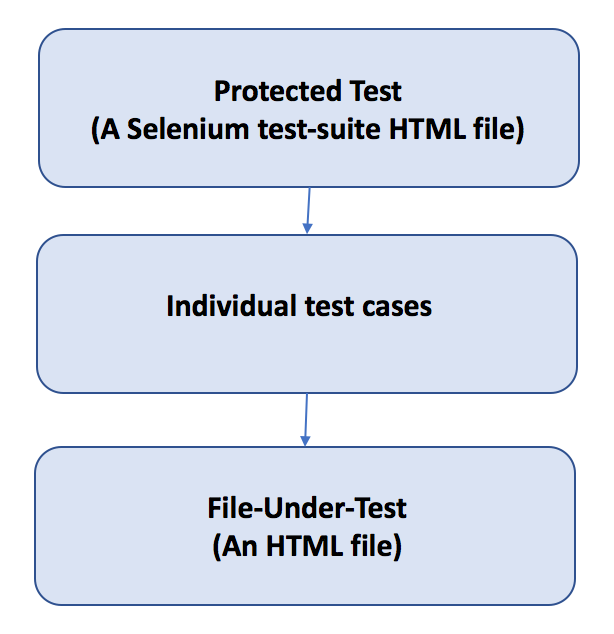
**Lxml** is a Python library for parsing XML and HTML. It allows for parsing and forming a tree of elements in accordance with the Document Object Model (DOM). A tree can then be used to access, modify, or add elements via Xpath or css selectors.

**XPath** (**XML Path Language**) is a [query language](https://en.wikipedia.org/wiki/Query_language) for selecting [nodes](https://en.wikipedia.org/wiki/Node_%28computer_science%29) from an [XML](https://en.wikipedia.org/wiki/XML) document. (From Wikipedia)

<Insert some info on test breakages/ taxonomy here>

Implementation

**Overview:** Protect Test is implemented as a plugin for Sublime Text. The Protect Test structure assumes the following model: a Selenium test suite is a single html file, composed of one or more individual tests, which verify properties of one or more html files.



**Usage:** To use Protect Test, a user specifies the tests he/she would like to protect in the settings for Protect Test. This is just JSON file with a list of tests to “protect”.

**Example – A Protect Test settings file:**

**{**

**"protected\_tests":**

**[**

**"/path/to/tests/test\_suite.html"**

**]**

**}**

Once this has been done, if the user opens a file that is protected, they will find that areas that, if modified, will break a test, those areas will be “folded” so that they are not visible. See Figures 1 and 2.



Figure - Prior to enabling Protect Test

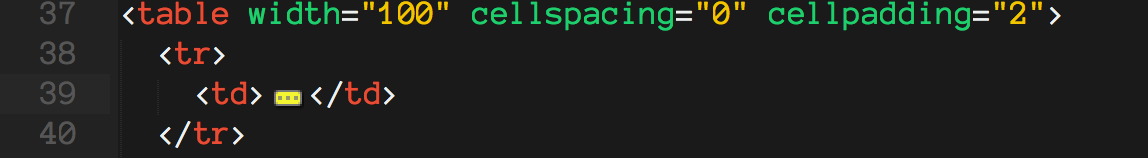


Figure - With the test-suite added to the list of protected tests, the area that, if modified would break the test, is "folded"

If a user “unfolds” this region and performs any action, the region will be immediately folded again. The purpose for this is to indicate to the user that this region is “protected”, i.e., modifying this text will break a test.

If a user would like to allow this code to be unfolded, they may hover over the region and select that option. (Figure below)



Figure - The Protect Test hover window

If a user deletes the folded area, or unfolds this area and makes an edit which would break the test, they are met with a warning. This warning will persist until either the changes are undone or the warning is disabled.

