Automation Home Task

1. Go to the [Internet Heroku App](https://the-internet.herokuapp.com/) web site and take an initial quick peek.
2. You need to write an automation testing framework in the Java programming language.    
   The framework must use a popular java testing framework, such as TestNG, or JUnit. You can use an IDE of your choice (such as IntelliJ IDEA or Eclipse).
3. Create a ReadMe meta data file for the project. It should include an overview on your project, detailed setup & prerequisites instructions. Keep in mind that it needs to be able to run on a different host and environment than your own.    
   You can also use this file to share your decision making process, why you chose to do X over Y, and so on.
4. Once you start coding, you have up to **4 hours** for this exercise. Try to free-up this time frame in advance, so you won't be interrupted as much as possible, and if you do happen to breach the  time limit specify why you thought it was justified in your ReadMe file.
5. **Any test that will be run**, will first open a web browser and then navigate to the Internet Heroku App web site - which will be your home page. You can choose any browser you like to run the tests on, but your framework should address the fact that in the future it will need to support multiple browsers. In order to save considerable time, it is **highly recommended** to use the open source Augmented Driver project which extends Selenium 2.0 Web Driver to achieve the above.
6. **Any test that will be run**, should report meaningful informative messages during the run, on different levels, to a logging system of your choice.  There must be persistence for the test reports, so only writing to the console is not enough.  Just to be crystal clear, do not implement your own logging framework! Instead, use a 3rd party dependency to accomplish the above.
7. Create runnable tests that you think are necessary for testing the following:
   1. Go to the [Checkboxes page](https://the-internet.herokuapp.com/checkboxes), modify the checkboxes states (checked/unchecked) for each of the checkboxes, to the opposite state than the one in which the page loaded, and then validate  that the state changed as expected.
   2. Go to the Frames page, and then to the iframe page. Write your name inside the TinyMCE Editor text paragraph, and ensure that it was written.
   3. Go to the Dynamic Loading page, and then click the *Example 2* link. Click the *Start* button, and ensure that "*Hello World!*" appears on the page.
   4. Go to the Public [APIs page](https://github.com/public-apis/public-apis) go to the Weather section (at the bottom),  select one of the weather apis (for example <https://www.metaweather.com/api/> ) and create some api tests to check the API functionality.

8. Additional Guidelines:

○ Use Object Oriented Programming principles as best you can.

○ Aim to use a meaningful & consistent code convention.

○ Add as much java doc as you see fit.

○ When you are done write down the total time that the task took.

○ Upload the project to a sharable location on a cloud of your choice, such as Google Drive, OneDrive, or Dropbox.

○ When sharing it with me use my email address [odedm@morphisec.com](mailto:odedm@morphisec.com)

I will be available for questions on my Gmail email address.

Good Luck,