**Hobsons Release Notes**

**In-Scope Items:**

1. [Hobsons.com](http://hobsons.com/) home page renders as expected.
2. On the home screen there is a Hero graphic with the text “**We help students** across the journey of a lifetime.” Click the down arrow. Assert the page scrolls the next sections “How can we help **your students?**” into the viewport window. Assert that it is correctly aligned with the top of the visible screen
3. When user clicks the “hamburger” menu at the top of the screen then the items the list drops down as expected with the sub-menu items. The menu I’m referring to is Solutions, Services, Resources, About & Blog. Assert that the “*Resources*” menus contain a list of child links including “E*vents*”.
4. Navigate to Resources > Events. Assert that all events on the page occurs in the future. Note you do NOT need to apply the filters, just assert on the default list that is displayed

**Tool:**

Cypress (Mocha Framework)

**Installation:**

Cypress node modules will be required.

* **Install Cypress via npm:**

cd /your/project/path

npm install cypress --save-dev

<https://docs.cypress.io/guides/getting-started/installing-cypress.html#Installing>

**Commands:**

* To open GUI:

-npx cypress open

* To run in CLI (headless):

-npx cypress run

* To run in chrome:

-npx cypress run --browser chrome

**Approach:**

* All test files (spec.js) are in integration folder and cases are independent.
* Fixture contains the config.js in which url is handled.
* Command.js contains the reusability code for opening website and gtg exception code is handled.
* BDD assertions are used i.e. should().
* Video folder contains the last execution.

**Limitation:**

For the 4th scenario i.e. “Assert that all events on the page occurs in the future”, partial logic was created to assert that whether event occurs in future. Limitation is that the dates in front-end do not follow a specific format and for some events dates are missing. From QA point of view better approach would be to follow a standard date template and an expected mock data should be provided.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*