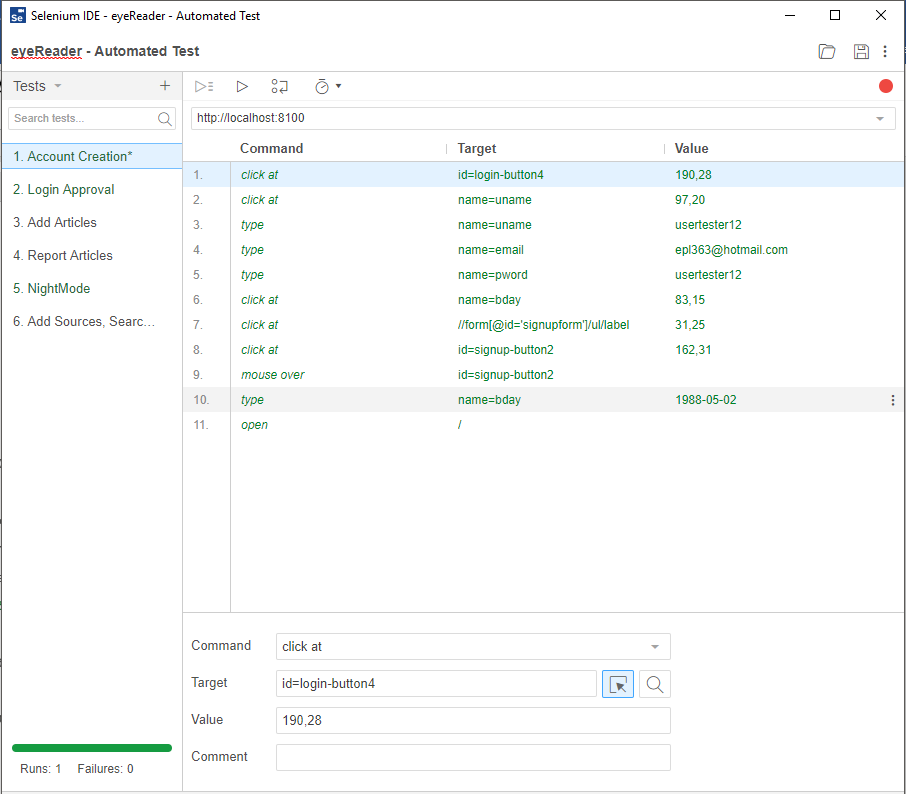


|  |  |
| --- | --- |
| 1. Unit Test Plan Scope (In Scope – Out of Scope) | |
| **In Scope** | **Out of Scope** |
| In Scope *List features/functions that are tested.*  - Sign up system  - Login system  - Journalist's note taking  - Requesting sources from the app's server  - Reporting articles  - Changing application settings  - Saving articles  - Article caching  - | Out of Scope *List features/functions that are not tested.* |

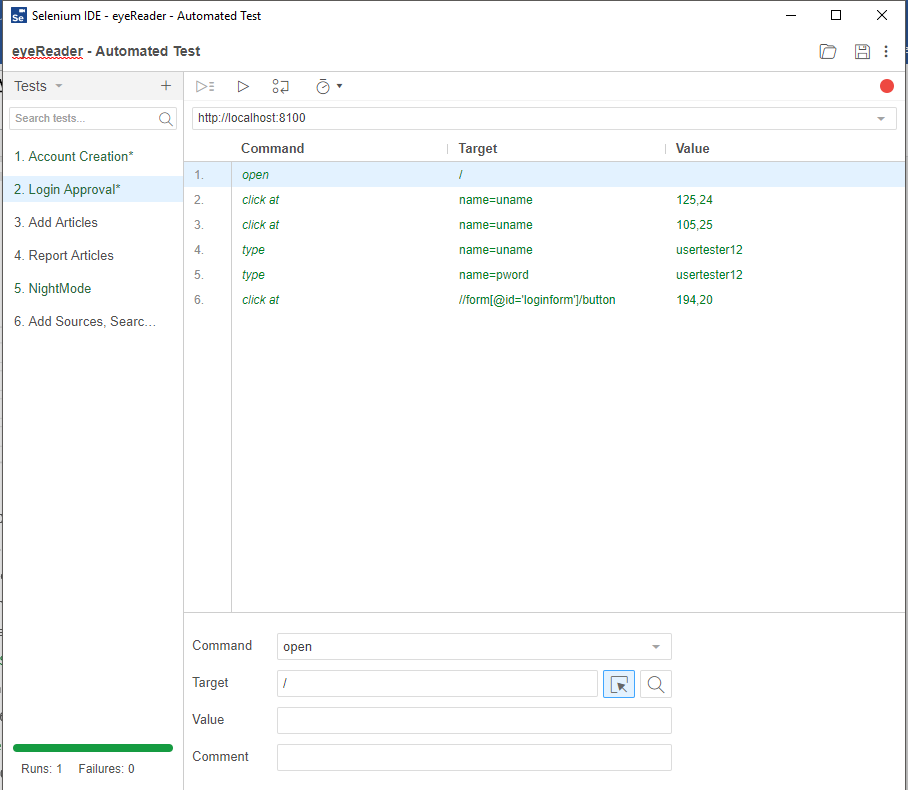
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2. Unit Test Cases | | | | |
| **ID** | **Test Cases** | **Input Value** | **Expected Output** |
| 2.1 | Signup System  **Test Procedure:**  Register with a journalist / casual user account.  **What the feature does:**  Receives the user's username, password, email, date of birth, gender and the indication whether the user is a journalist or not (By checking the Journalist box)  This information is then stored on the device's local storage in the form of JSON. | Username: epl363test  Password: epl363pass  Email:epl363@hotmail.com Date Of Birth: 01/02/1990  Gender: Female  Journalist: Checked | A json file with the values of the registered users will be generated |
| 2.2 | Login System  **Test Procedure:**  Login into a user’s account.  **What the feature does:**  Receives the user's username and password and validates the given input by checking the local storage for any credentials (username/password combination) that match the ones given in the username and password field of the app. | Username: epl363test  Password: epl363pass | Successful logging of the user to the application and redirecting them to the news feed page |
| 2.3 | Journalist’s Note Taking  **Test Procedure:**  Create a note on specific articles as a journalist.  **What the feature does:**  Receives the typed notes of the current user on the article that is currently being previewed and  stores it to the local storage in the form of a json file. The json file is matches the current user’s username with the ID of the articles that the user has typed notes on and the actual content of the notes. | Created Notes for various articles | A json file with the notes and the matching article that the user has typed the notes on |
| 2.4 | Requesting sources from the app's server  **Test Procedure:**  Execute an HTTP GET Request to the app’s Server  **What the feature does:**  Executes an HTTP GET request to the app’s server in order to receive the article sources that are available in the server | An HTTP GET Request to the App’s Server | A server response that is a json file that holds the sources with their IDs |
| 2.5 | Article Reporting  **Test Procedure:**  Clicking the report button on an article by sliding the article to the left and tapping the Report Exclamation Button  **What the feature does:**  Executes an HTTP GET request to the server in order to increase the counter of the reported articles of  a particular source | An HTTP GET request to the server in order to increase the counter of the reported articles of  a particular source | Server confirmation that the HTTP GET request was successful |
| 2.6 | Configuring Application Settings  **Test Procedure:**  Changing a multitude of settings within the application’s settings such as nightmode, article caching toggling and font resizing.  **What the feature does:**  Users can configure the application’s settings and then the settings are saved in the local storage in the form  of a json file. The json file is a match between the user's username and the configured settings. Then the application must be able to load the settings when the user logs in again. | Changed values of various settings in the application’s settings.  - NightMode  - Article Caching Toggle  - Font Resizing | A Json file containing the match between the user’s username and their respective settings |
| 2.7 | Article Saving  **Test Procedure:** Saving various articles across the News Feed page of the app by tapping on the Star Symbol  **What the feature does:**  The user has the option to save an article so that they can preview it while the application is offline. The article is stored on the device’s local storage in the form of a json file. The json file matches the user's username with the saved articles. | Saving various articles across the News Feed page of the app by tapping on the Star Symbol | A Json file with the user’s username and the saved articles |
| 2.8 | Article Caching  **Test Procedure:** Saving various articles across the News Feed page of the app by tapping on the Star Symbol  **What the feature does:**  When the application is online, it automatically saves (if the user has selected the caching option from the settings) the top 10 articles in their news feed in order to be able to load them in the news feed when the application is offline | Toggled Cache Setting in the Application’s Setting Menu | A json file with the user's username and the cached articles |
| 2.9 | Postman Testing: HTTP GET’/’  **Test Procedure:** Executed the HTTP GET’/’ in postman  **What the feature does:**  Sends a message that the server is operating. | None | Response message that the server is operating |
| 3.0 | Postman Testing: HTTP GET’/about’  **Test Procedure:** Executed the HTTP GET’’/about’ in postman  **What the feature does:**  Returns readMe, license and Credits files | None | Returns readMe, license and credits files |
| 3.1 | Postman Testing: HTTP GET’/sources’  **Test Procedure:** Executed the HTTP GET’’/sources’ in postman  **What the feature does:**  Returns all the sources of the database in a json file format | None | Returns all the sources of the database in a json file format |
| 3.2 | Postman Testing: HTTP GET’/articles/’  **Test Procedure:** Executed the HTTP GET’’/sources’ in postman  **What the feature does:**  Returns all the articles of the database in a json file format | None | Returns all the articles of the database in a json file format |
| 3.3 | Postman Testing: HTTP GET '/articles/:id/click/'  **Test Procedure:** Executes the HTTP GET’'/articles/:id/click/' in postman  **What the feature does:**  Increases by 1 the click counter for the specific (:id) article | The ID of the article clicked | Click counter of the specific (:id) article increased by one |
| 3.4 | Postman Testing: Increased click counter and executed HTTP GET '/articles/'  **Test Procedure:** Executed the HTTP GET’'/articles/’ and checked whether the click counter for the specific article has been increased | None | Returns all the articles of the database in a json file format and the specific (:id) article that the counter was increased for, has its click counter increased |
| 3.5 | Postman Testing: HTTP GET '/articles/:id/report/'  **Test Procedure:** Executes the HTTP GET’'/articles/:id/report/' in postman  **What the feature does:**  Increases by 1 the report counter for the specific (:id) article | The ID of the article reported | Report counter of the specific (:id) article increased by one |
| 3.6 | Postman Testing: Increased report counter and executed HTTP GET '/articles/'  **Test Procedure:** Executed the HTTP GET’'/articles/’ and checked whether the report counter for the specific article has been increased | None | Returns all the articles of the database in a json file format and the specific (:id) article that the counter was increased for, has its report counter increased |
| 3.8 | Statistics (Graphs)  **Test Procedure:** In the test conducted we selected the “All Sources” selection to generate the graphs with the statistics regarding all the sources by clicking the “Fetch” button. Afterwards, we selected a specific source to retrieve data and graphs for by clicking the “Fetch” button again.  **What the feature does:**  When the user clicks the "Fetch" button, the bar charts are generated with the selected statistic. The data for the charts are retrieved from the HTTP requests to the server. | The user's selection for source and type of statistic to generate | When the user clicks the "Fetch" button, the bar charts should appear with the selected statistic. The data for the charts are retrieved from the HTTP requests to the server. |
| 3.9 | Save / Delete / Report buttons in article view  **Test Procedure:** In the test conducted, we checked whether the functions save, delete and report work through the implemented buttons when viewing an article in article’s view. This was done by clicking each method and checking whether the respective action was done.  **What the feature does:**  The three buttons that have been added have the following functions:  - Save  - Delete  - Report  Similarly, like the ones available when sliding to the left on an article in the News Feed window, they offer the same functionalities as described above. This was implemented to enhance the experience of the user. | None | When the user clicks one of the buttons Save / Delete / Report the respective action should happen based on their selection. |
| 4.0 | Infinite Scrolling  **Test Procedure:** Scroll to the bottom of the article news feed and check whether the list will be refreshed if any remaining articles from the current sources are available  **What the feature does:**  Enables the user to infinitely scroll through their article news feed if the are any available articles based on their selected sources. Therefore, to load the new articles the user can instantly fetch them by scrolling down their phone screen. | None | When the user scrolls to the bottom of the page, if there are any more articles left to display starting from the previous article, fetch those articles from the server. Afterwards, some of them will be added to the bottom of the feed for the user to view. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3. Unit Test Results | | |  |  |
| **ID** | **Test Cases** | **Pass/Fail** | **Tested By** | **Date Tested** |
| 3.1 | Signup System  **Test Procedure:**  Register with a journalist / casual user account.  **What the feature does:**  Receives the user's username, password, email, date of birth, gender and the indication whether the user is a journalist or not (By checking the Journalist box)  This information is then stored on the device's local storage in the form of JSON. | PASS | Giorgos Hadjidemetriou | 23/02/18 |
| 3.2 | Login System  **Test Procedure:**  Login into a user’s account.  **What the feature does:**  Receives the user's username and password and validates the given input by checking the local storage for any credentials (username/password combination) that match the ones given in the username and password field of the app. | PASS | Giorgos Hadjidemetriou  Antrea Chrysanthou | 23/02/18 |
| 3.3 | Journalist’s Note Taking  **Test Procedure:**  Create a note on specific articles as a journalist.  **What the feature does:**  Receives the typed notes of the current user on the article that is currently being previewed and  stores it to the local storage in the form of a json file. The json file is matches the current user’s username with the ID of the articles that the user has typed notes on and the actual content of the notes. | PASS | Giorgos Hadjidemetriou  Antrea Chrysanthou | 25/03/18 |
| 3.4 | Requesting sources from the app's server  **Test Procedure:**  Execute an HTTP GET Request to the app’s Server  **What the feature does:**  Executes an HTTP GET request to the app’s server in order to receive the article sources that are available in the server | PASS | Giorgos Hadjidemetriou | 23/02/18 |
| 3.5 | Article Reporting  **Test Procedure:**  Clicking the report button on an article by sliding the article to the left and tapping the Report Exclamation Button  **What the feature does:**  Executes an HTTP GET request to the server in order to increase the counter of the reported articles of  a particular source | PASS | Giorgos Hadjidemetriou  Constantinos Stylianou | 26/02/18 |
| 3.6 | Configuring Application Settings  **Test Procedure:**  Changing a multitude of settings within the application’s settings such as nightmode, article caching toggling and font resizing.  **What the feature does:**  Users can configure the application’s settings and then the settings are saved in the local storage in the form  of a json file. The json file is a match between the user's username and the configured settings. Then the application must be able to load the settings when the user logs in again. | PASS | Giorgos Hadjidemetriou  Constantinos Stylianou | 20/03/18 |
| 3.7 | Article Saving  **Test Procedure:** Saving various articles across the News Feed page of the app by tapping on the Star Symbol  **What the feature does:**  The user has the option to save an article so that they can preview it while the application is offline. The article is stored on the device’s local storage in the form of a json file. The json file matches the user's username with the saved articles. | PASS | Giorgos Hadjidemetriou | 10/03/18 |
| 3.8 | Article Caching  **Test Procedure:** Saving various articles across the News Feed page of the app by tapping on the Star Symbol  **What the feature does:**  When the application is online, it automatically saves (if the user has selected the caching option from the settings) the top 10 articles in their news feed in order to be able to load them in the news feed when the application is offline | PASS | Giorgos Hadjidemetriou | 26/03/18 |
| 3.9 | Postman Testing: HTTP GET ’/’  **Test Procedure:** Executed the HTTP GET’’/’ in postman  **What the feature does:**  Sends a message that the server is operating. | PASS | Marios Kelepeshis | 26/03/18 |
| 4.0 | Postman Testing: HTTP GET ’/about’  **Test Procedure:** Executed the HTTP GET’’/about’ in postman  **What the feature does:**  Returns readMe, license and Credits files | PASS | Marios Kelepeshis | 23/03/18 |
| 4.1 | Postman Testing: HTTP GET ’/sources’  **Test Procedure:** Executed the HTTP GET’’/sources’ in postman  **What the feature does:**  Returns all the sources of the database in a json file format | PASS | Marios Kelepeshis | 22/02/18 |
| 4.2 | Postman Testing: HTTP GET ’/articles/’  **Test Procedure:** Executed the HTTP GET’’/sources’ in postman  **What the feature does:**  Returns all the articles of the database in a json file format | PASS | Marios Kelepeshis | 22/02/18 |
| 4.3 | Postman Testing: HTTP GET '/articles/:id/click/'  **Test Procedure:** Executes the HTTP GET’/articles/:id/click/' in postman  **What the feature does:**  Increases by 1 the click counter for the specific (:id) article | PASS | Marios Kelepeshis | 13/03/18 |
| 4.4 | Postman Testing: Increased click counter and executed HTTP GET '/articles/'  **Test Procedure:** Executed the HTTP GET’/articles/’ and checked whether the click counter for the specific article has been increased | PASS | Marios Kelepeshis | 13/03/18 |
| 4.5 | Postman Testing: HTTP GET '/articles/:id/report/'  **Test Procedure:** Executes the HTTP GET’/articles/:id/report/' in postman  **What the feature does:**  Increases by 1 the report counter for the specific (:id) article | PASS | Marios Kelepeshis | 13/03/18 |
| 4.6 | Postman Testing: Increased report counter and executed HTTP GET '/articles/'  **Test Procedure:** Executed the HTTP GET’/articles/’ and checked whether the report counter for the specific article has been increased | PASS | Marios Kelepeshis | 13/03/18 |
| 4.7 | Statistics (Graphs)  **Test Procedure:** In the test conducted we selected the “All Sources” selection to generate the graphs with the statistics regarding all the sources by clicking the “Fetch” button. Afterwards, we selected a specific source to retrieve data and graphs for by clicking the “Fetch” button again.  **What the feature does:**  When the user clicks the "Fetch" button, the bar charts are generated with the selected statistic. The data for the charts are retrieved from the HTTP requests to the server. | PASS | Antrea Chrysanthou  Constantinos Stylianou | 17/04/18 |
| 4.8 | Save / Delete / Report buttons in article view  **Test Procedure:** In the test conducted, we checked whether the functions save, delete and report work through the implemented buttons when viewing an article in article’s view. This was done by clicking each method and checking whether the respective action was done.  **What the feature does:**  The three buttons that have been added have the following functions:  - Save  - Delete  - Report  Similarly, like the ones available when sliding to the left on an article in the News Feed window, they offer the same functionalities as described above. This was implemented to enhance the experience of the user. | PASS | Antrea Chrysanthou | 20/04/18 |
| 4.9 | Infinite Scrolling  **Test Procedure:** Scroll to the bottom of the article news feed and check whether the list will be refreshed if any remaining articles from the current sources are available  **What the feature does:**  Enables the user to infinitely scroll through their article news feed if the are any available articles based on their selected sources. Therefore, to load the new articles the user can instantly fetch them by scrolling down their phone screen. | PASS | Antrea Chrysanthou | 24/04/18 |

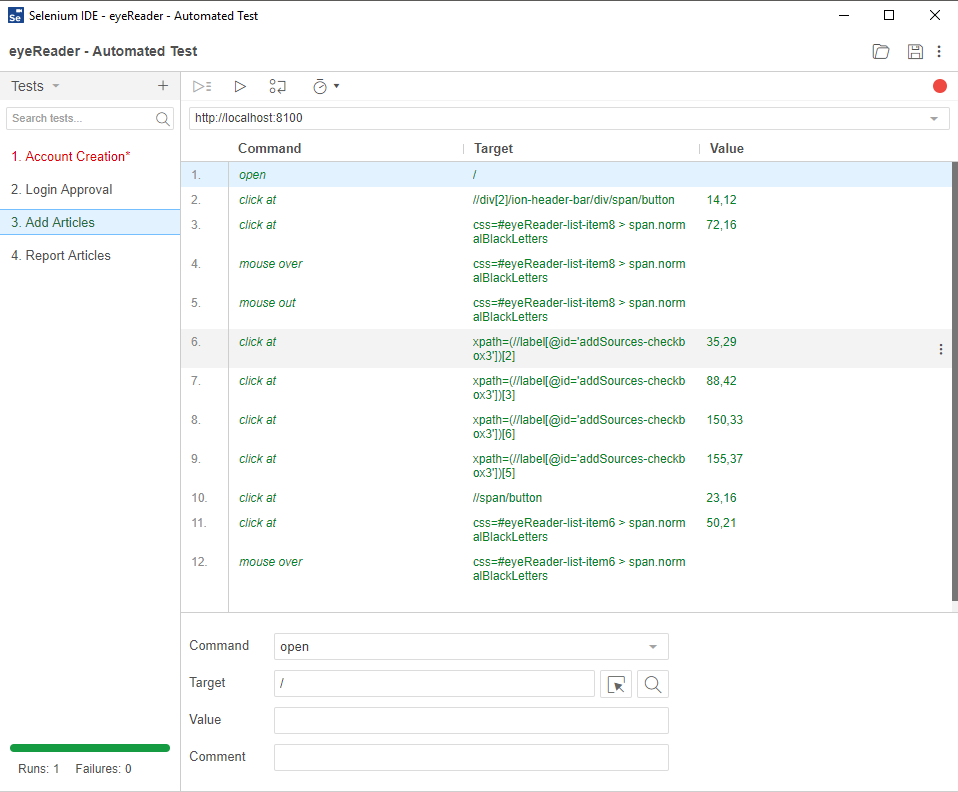
**Automated Tests using Selenium:**



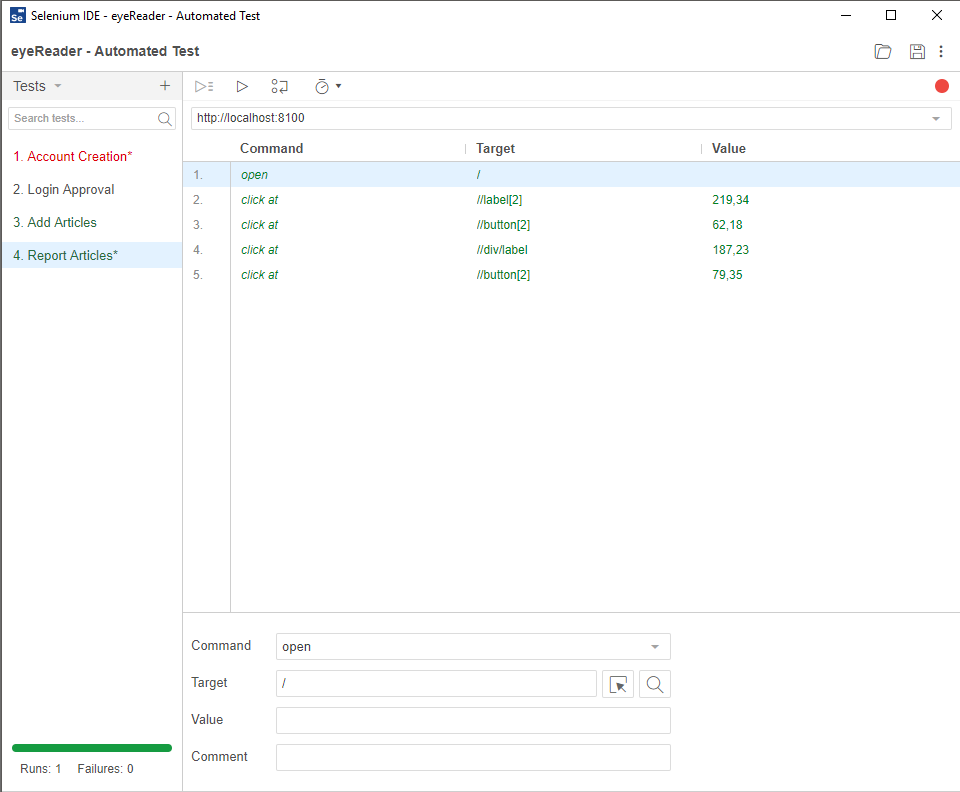
* **Running '1. Account Creation'**
* 1.Trying to execute clickAt on id=login-button4 with value 190,28... Success
* 2.Trying to execute clickAt on name=uname with value 97,20... Success
* 3.Trying to execute type on name=uname with value usertester12... Success
* 4.Trying to execute type on name=email with value epl363@hotmail.com... Success
* 5.Trying to execute type on name=pword with value usertester12... Success
* 6.Trying to execute clickAt on name=bday with value 83,15... Success
* 7.Trying to execute clickAt on //form[@id='signupform']/ul/label with value 31,25... Success
* 8.Trying to execute clickAt on id=signup-button2 with value 162,31... Success
* 9.Trying to execute mouseOver on id=signup-button2... Success
* 10.Trying to execute type on name=bday with value 1988-05-02... Success
* 11.Trying to execute open on /... Success
* **'1. Account Creation' completed successfully**



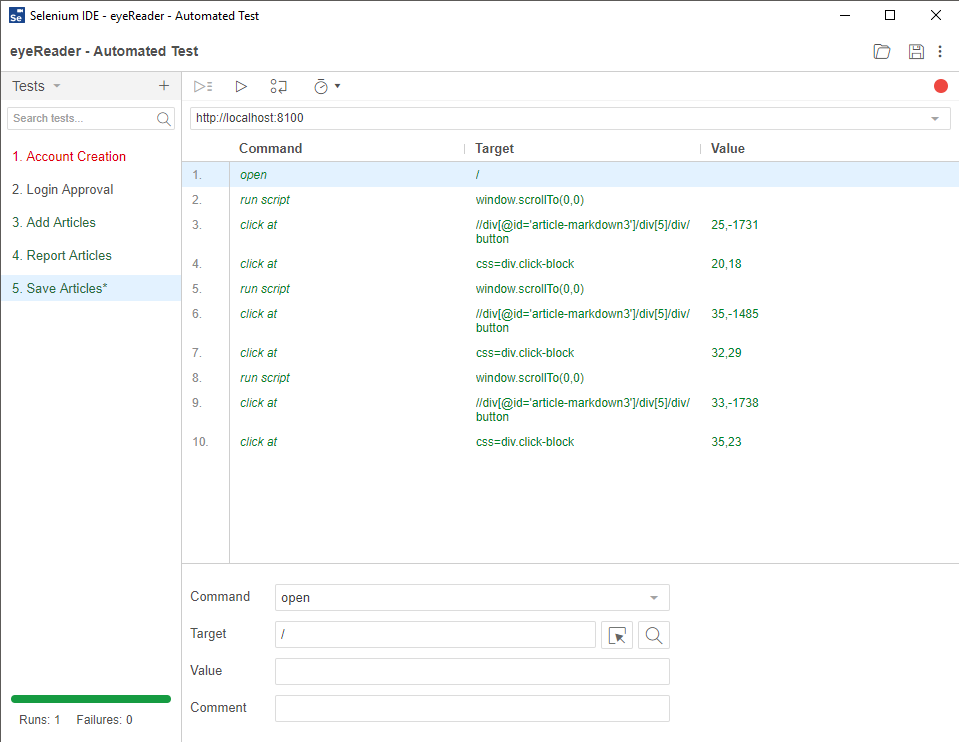
* **Running '2. Login Approval'**
* 1.Trying to execute open on /... Success
* 2.Trying to execute clickAt on name=uname with value 125,24... Success
* 3.Trying to execute clickAt on name=uname with value 105,25... Success
* 4.Trying to execute type on name=uname with value usertester12... Success
* 5.Trying to execute type on name=pword with value usertester12... Success
* 6.Trying to execute clickAt on //form[@id='loginform']/button with value 194,20... Success
* **'2. Login Approval' completed successfully**



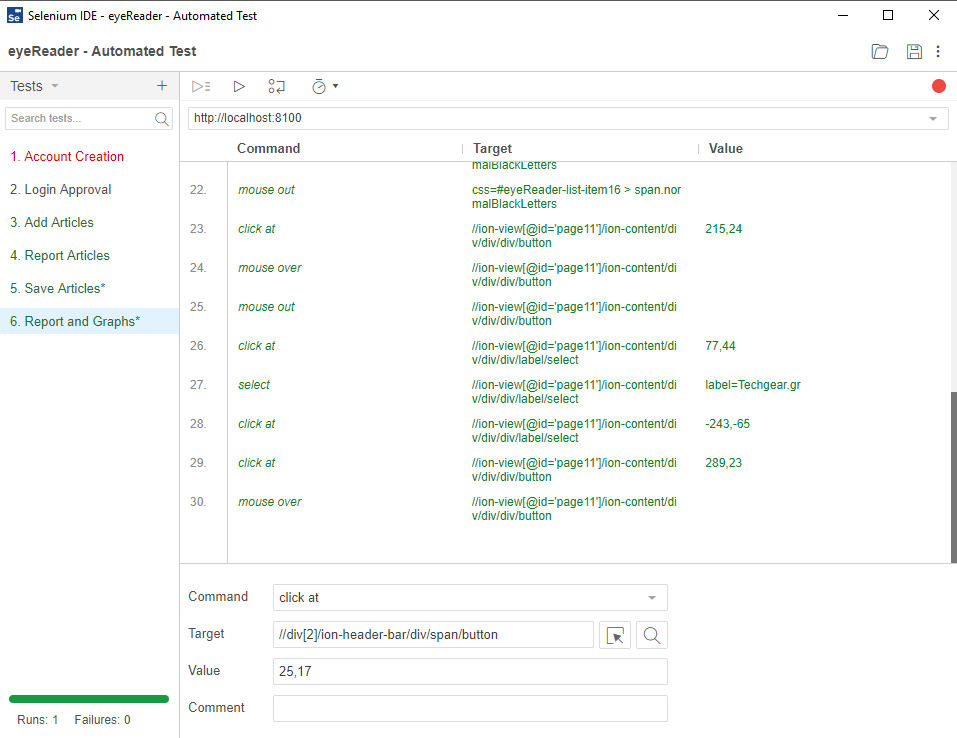
* **Running '3. Add Articles'**
* 1.Trying to execute open on /... Success
* 2.Trying to find //div[2]/ion-header-bar/div/span/button... Success
* 3.Trying to execute clickAt on css=#eyeReader-list-item8 > span.normalBlackLetters with value 72,16... Success
* 4.Trying to execute mouseOver on css=#eyeReader-list-item8 > span.normalBlackLetters... Success
* 5.Trying to execute mouseOut on css=#eyeReader-list-item8 > span.normalBlackLetters... Success
* 6.Trying to execute clickAt on xpath=(//label[@id='addSources-checkbox3'])[2] with value 35,29... Success
* 7.Trying to execute clickAt on xpath=(//label[@id='addSources-checkbox3'])[3] with value 88,42... Success
* 8.Trying to execute clickAt on xpath=(//label[@id='addSources-checkbox3'])[6] with value 150,33... Success
* 9.Trying to execute clickAt on xpath=(//label[@id='addSources-checkbox3'])[5] with value 155,37... Success
* 10.Trying to execute clickAt on //span/button with value 23,16... Success
* 11.Trying to execute clickAt on css=#eyeReader-list-item6 > span.normalBlackLetters with value 50,21... Success
* 12.Trying to execute mouseOver on css=#eyeReader-list-item6 > span.normalBlackLetters... Success
* **'3. Add Articles' completed successfully**

 **Running '4. Report Articles'**

* 1.Trying to execute open on /... Success
* 2.Trying to find //label[2]... Success
* 3.Trying to execute clickAt on //button[2] with value 62,18... Success
* 4.Trying to execute clickAt on //div/label with value 187,23... Success
* 5.Trying to execute clickAt on //button[2] with value 79,35... Success
* **'4. Report Articles' completed successfully**



* **Running '5. Save Articles'**
* 1.Trying to execute open on /... Success
* 2.Trying to execute runScript on window.scrollTo(0,0)... Success
* 3.Trying to find //div[@id='article-markdown3']/div[5]/div/button... Success
* 4.Trying to execute clickAt on css=div.click-block with value 20,18... Success
* 5.Trying to execute runScript on window.scrollTo(0,0)... Success
* 6.Trying to execute clickAt on //div[@id='article-markdown3']/div[5]/div/button with value 35,-1485... Success
* 7.Trying to execute clickAt on css=div.click-block with value 32,29... Success
* 8.Trying to execute runScript on window.scrollTo(0,0)... Success
* 9.Trying to execute clickAt on //div[@id='article-markdown3']/div[5]/div/button with value 33,-1738... Success
* 10.Trying to execute clickAt on css=div.click-block with value 35,23... Success
* **'5. Save Articles' completed successfully**



* **Running '6. Report and Graphs'**
* 1.Trying to execute open on /... Success
* 2.Trying to execute runScript on window.scrollTo(0,0)... Success
* 3.Trying to execute clickAt on css=html with value 161,135... Success
* 4.Trying to find //div/label... Success
* 5.Trying to execute clickAt on //label[2] with value 215,30... Success
* 6.Trying to execute clickAt on //div[3]/button[2] with value 93,15... Success
* 7.Trying to execute clickAt on css=div.click-block with value 38,29... Success
* 8.Trying to execute runScript on window.scrollTo(0,0)... Success
* 9.Trying to execute clickAt on css=html with value 168,113... Success
* 10.Trying to find //label[2]... Success
* 11.Trying to execute clickAt on //div/label with value 181,26... Success
* 12.Trying to execute clickAt on //div[3]/button[2] with value 61,24... Success
* 13.Trying to execute clickAt on css=div.click-block with value 37,31... Success
* 14.Trying to execute runScript on window.scrollTo(0,0)... Success
* 15.Trying to execute clickAt on css=html with value 160,98... Success
* 16.Trying to execute clickAt on //div/label with value 118,12... Success
* 17.Trying to execute clickAt on //div[3]/button[2] with value 63,19... Success
* 18.Trying to execute clickAt on css=div.click-block with value 34,24... Success
* **'6. Report and Graphs' completed successfully**