

Front-end test

After exploring the Vue Real World Example App, the Conduit project, the following questions were answered:

- **P2 What are the 3 most critical flows of this application? Explain why for each one.**

In my studies, I consider this flows as the most critical ones:

- **Sign up:** Without an account, the user cannot log in to the app, and all features to logged users are unavailable.
 - **Sign in:** If the user is not logged in, the user cannot interact with the main features on the website, such as creating a new article, posting a comment to other user's articles, following users, etc.
 - **Access Articles:** The main feature of the application is to be able to read articles. Guarantee that the articles access is working and all the features on it, such as Favorite the article, Follow the user, have the user details, etc.
-
- **Based on the 3 flows above, write how many test cases you consider important to ensure that these flows continue working.**

There are the success cases and the edge cases for the previous flows:

- **Sign up:**
 - Main case:
 - Create a new user;
 - Edge cases:
 - Error when create a new user without an email;
 - Error when create a new user without an username;
 - Error when create a new user without a password
 - Error when username is already being used;
 - Error when email is already being used;
- **Sign in:**
 - Main case:
 - Login to the app;
 - Edge cases:
 - Error when log in without an email
 - Error when log in without a password
 - Error when email is invalid
 - Error when password is invalid
- **Article access:**
 - Main cases:
 - Create a new article;
 - Access an article with a not-logged user;
 - Access an article with a logged user;

All these cases are available on the Test Case - Conduit document found on the Documents folder, on GitHub.

- **Automate, using the framework or language you feel most comfortable with, the test case that you consider the most important from each critical.**

The automated tests are performed in Cypress (<https://www.cypress.io/>), which is a testing framework which I, Dhiego Bastos, am familiar with, and is one of the most used testing tools in the market nowadays.

The test runs are considering the test cases created for the "Sign Up", "Sign In" and "Articles" flows, considering the main cases for each one:

- **Sign up:**
 - Create a new user;
- **Sign in:**
 - Login to the app;
- **Article access:**
 - Create a new article;

All information for installation will be found on the README file on GitHub

- **Report any bugs found, define the priority of each bug, and explain your prioritization.**

To report this bug, I would report it by separating the ticket in 7 categories:

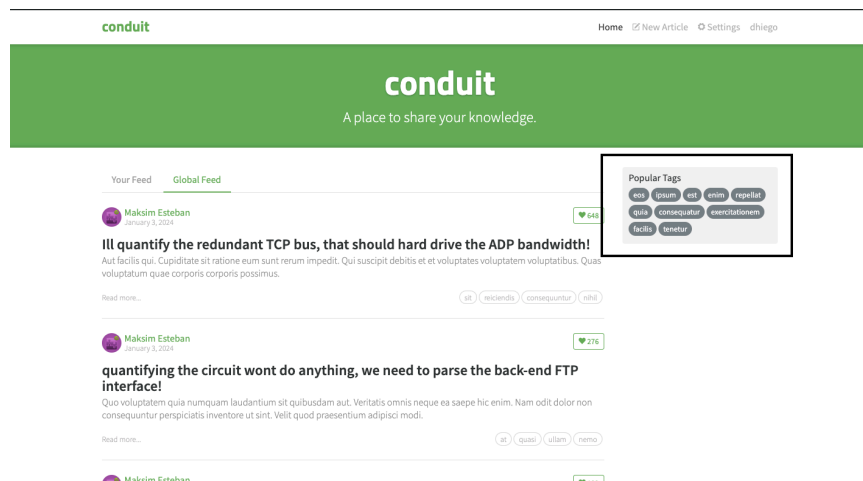
- Situation: Would explain what the bug was, with videos and/or screenshots;
- How to reproduce it: Step by step on what the user needs to do to reproduce the bug;
- Expected Behavior: What the app behavior is expected in that scenario;
- Actual Behavior: How the app is behaving at the current version;
- Environment: Which environment the bug was found (SIT, UAT, Production);
- Priority: From Highest to Lowest, I would rank the issue's impact and criticality on the app;
- Browsers: Which browsers the bug was detected.

That being said, I could find three different bugs and they are reported as below:

Bug 1:

Ticket: Not possible to filter articles by tags on Global feed page

Situation: When the user is trying to filter articles by clicking on the Popular Tags field, on the Home page, nothing happens.



- How to reproduce it:**
1. Logged into the app, access Home page;
 2. Click on any tags on the "Popular Tags";

Expected Behavior: The selected tag should filter the articles and these should be displayed to the user.

Actual Behavior: Nothing happens.

Environment: Production.

Priority: Medium.

Browsers: Chrome, Safari, Firefox, Opera.

The Medium priority was chosen as it does not stop the main flow, and the user can access the articles by other ways.

Bug 2:

Ticket: Not possible to add articles as favorite.

Situation: When the user is trying to add articles as favorite, the API response is Unauthorized, and the article is not favorited.

Request URL:	https://api.realworld.io/api/articles/ill-quantify-the-redundant-TCP-bus-that-should-hard-drive-the-ADP-bandwidth!-553/favorite
Request Method:	POST
Status Code:	401 Unauthorized
Remote Address:	172.67.186.217:443
Referrer Policy:	strict-origin-when-cross-origin



How to reproduce it: 1. Logged into the app, access Home page;

2. Select any Article;

3. Click on the "Favorite Article" button;

Expected Behavior: The article should be favorited and available to the user.

Actual Behavior: Nothing happens and the API has a Unauthorized error as a request response.

Environment: Production.

Priority: Medium.

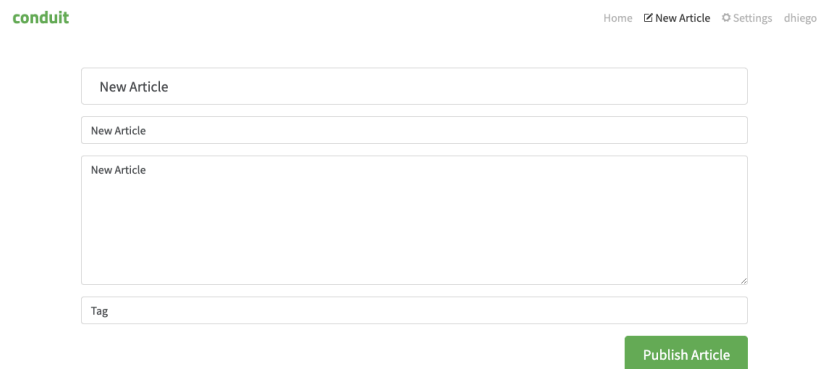
Browsers: Chrome, Safari, Firefox, Opera.

The Medium priority was chosen as it does not stop the main flow, and the user can access the articles by other ways.

Bug 3:

Ticket: Not possible to add tags on articles.

Situation: When the user creates a new article and adds the tags, the article is created but without a tag. The API request payload does not carry the Tag information with it.



The screenshot shows the 'New Article' form in the conduit app. The form has four input fields: 'New Article' (title), 'New Article' (author), 'New Article' (description), and 'Tag' (tags). A green 'Publish Article' button is at the bottom right.



```

X Headers Payload Preview Response Initiator Timing
▼Request Payload view source
  ▼ {, ...}
    ▼ article: {author: {}, title: "New Article", description: "New Article",
      author: {}
      body: "New Article"
      description: "New Article"
      tagList: []
      title: "New Article"

```

How to reproduce it: 1. Logged into the app, access Home page;
2. Select "New Article" on the header;
3. Add the article information and create it;

Expected Behavior: The article should be created with the tags chosen and it should be displayed within the article.

Actual Behavior: The article is created but without tags.

Environment: Production.

Priority: Medium.

Browsers: Chrome, Safari, Firefox, Opera.

The Medium priority was chosen as it does not stop the main flow, and the user can access the articles by other ways.