

Rapport détaillé

[Bug report] – Login

Description

Dans le rapport de test "Login, si un administrateur remplit correctement les champs du Login, il devrait naviguer sur la page Dashboard", le test est passé au rouge.

To-do

Faire passer le test au vert en réparant la fonctionnalité.

On utilise chrome debugger pour mettre en œuvre des breakpoints, étape par étape et pas à pas dans le fichier JS Login.js du dossier et sous dossier /front/src/containers/ pour localiser le bug et ainsi modifier le code JS pour permettre à l'administrateur de se connecter.

```
//fix: login as admin
handleSubmitAdmin = (e) => {
  e.preventDefault();
  const user = {
    type: "Admin",
    email: e.target.querySelector(`input[data-testid="admin-email-input"]`).value,
    password: e.target.querySelector(`input[data-testid="admin-password-input"]`).value,
    status: "connected",
  };
};
```

Dans la fonction handleSubmitAdmin, les inputs utilisés pour le login admin sont ceux de l'employée (employee-email-input), je modifie donc le code JS pour utiliser les inputs admin avec « admin-email-input » et « admin-password-input »

On effectue les tests JEST coverage et unitaire sur Login.js

```
$ jest --coverage --noStackTrace --silent src/__tests__/Login.js
```

```
$ jest src/__tests__/Login.js
```

```
Test Suites: 1 passed, 1 total
Tests:       8 passed, 8 total
Snapshots:   0 total
Time:        3.016 s, estimated 4 s
Ran all test suites matching /src\/__tests__\/login.js/i.
```

[Bug report] – Bills

Description

Le test Bills / les notes de frais s'affichent par ordre décroissant est passé au rouge.

To-do

Faire passer le test au vert en réparant la fonctionnalité.

On utilise chrome debugger pour mettre en œuvre des breakpoints, étape par étape et pas à pas dans le fichier JS BillsUI.js du dossier et sous dossiers /front/src/views / pour localiser le bug et ainsi ajouter du code JS pour que les notes de frais s'affichent par ordre décroissant.

Je travaille sur `const rows= (data);` je trie les data par ordre décroissant avec `sort()`. Return `true (1)` si `new Date(a.date)` est inférieur `new Date(b.date)` sinon return `false(-1)`

```
20  });
21  //fix: trier les factures dans l'ordre décroissant
22  // fonction sort décroissante
23  // ex: 2012 - 2013 = -1
24  // 2012 - 2012 = 0
25  // 2013 - 2012 = 1
26  const rows = (data) => {
27    data.sort((a, b) => (new Date(a.date) < new Date(b.date) ? 1 : -1));
28    return data && data.length ? data.map((bill) => row(bill)).join("") : "";
29  };
30
```

Rapport de test

Rapport de test unitaire avant :

```
FAIL src/__tests__/Bills.js
  ● Given I am connected as an employee › When I am on Bills Page › Then bills should be ordered from earliest to latest

    expect(received).toEqual(expected) // deep equality

    - Expected  - 1
    + Received  + 1

      Array [
        "2004-04-04",
    +   "2001-01-01",
        "2003-03-03",
        "2002-02-02",
    -   "2001-01-01",
      ]

Test Suites: 1 failed, 1 total
Tests:       1 failed, 1 passed, 2 total
Snapshots:  0 total
Time:        3.903 s
```

Rapport de test unitaire après :

```
PASS src/__tests__/Bills.js
Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        2.304 s
Ran all test suites matching /src\\__tests__\\Bills.js/i.
```

[Bug Hunt] – Bills

Description

Je suis connecté en tant qu'employé, je saisis une note de frais avec un justificatif qui a une extension différente de jpg, jpeg ou png, j'envoie. J'arrive sur la page Bills, je clique sur l'icône "voir" pour consulter le justificatif : la modale s'ouvre, mais il n'y a pas d'image.

Si je me connecte à présent en tant qu'Admin, et que je clique sur le ticket correspondant, le nom du fichier affiché est null. De même, lorsque je clique sur l'icône "voir" pour consulter le justificatif : la modale s'ouvre, mais il n'y a pas d'image.

To-do

Comportements attendus :

- ☐ la modale doit afficher l'image.
- ☐ dans le dashboard, le formulaire correspondant au ticket doit afficher le nom du fichier.

Suggestion : empêcher la saisie d'un document qui a une extension différente de jpg, jpeg ou png au niveau du formulaire du fichier NewBill.js. Indice : cela se passe dans la méthode `handleChangeFile...`

On utilise chrome debugger pour mettre en œuvre des breakpoints, étape par étape et pas à pas dans le fichier JS `NewBill.js` du dossier et sous dossiers `/front/src/containers /` pour localiser le bug et puis on ajoute des nouvelles variables pour vérifier le format de l'image, on vérifie si le format est jpg jpeg ou png et si vrai on met `imgformatvalide` à vrai sinon faux.

Si le format n'est pas valide on supprime le fichier de l'input, on ajoute la class `is invalid` pour dire à l'utilisateur que c'est invalide et on supprime la classe `blue-border`, puis on affiche un message d'erreur

Si c'est valide on supprime la classe is invalid on ajoute la classe border-blue. Si le format de l'image est valide et le formulaire on débute l'upload

```
// Nouvelle variable pour vérifier le format de l'image
const fileInput = this.document.querySelector('input[data-testid="file"]');
const fileAcceptedFormats = ["jpg", "jpeg", "png"];
const fileNameParts = fileName.split(".");
const fileExtension = fileNameParts[fileNameParts.length - 1].toLowerCase();
this.ImgFormatValide = false;
// Si il y a au moins 2 parties dans le nom du fichier continuer la verification
if (fileNameParts.length > 1) {
    // verifie si le format de l'image est jpg jpeg ou png, si vrai met ImgFormatValide à vrai sinon faux
    fileAcceptedFormats.indexOf(fileExtension) !== -1
        ? (this.ImgFormatValide = true)
        : (this.ImgFormatValide = false);
}
if (!this.ImgFormatValide) {
    // Si le format d'image n'est pas valide
    fileInput.value = ""; // ... supprimer le fichier de l'input
    fileInput.classList.add("is-invalid"); // ... ajouter la class is invalid pour dire à l'utilisateur que c'est invalide
    fileInput.classList.remove("blue-border"); // ... supprime la classe blue-border
    alert(
        "Le format de votre fichier n'est pas pris en charge." +
        "\n" +
        "Seuls les .jpg, .jpeg, .png sont acceptés."
    ); // message d'erreur pour l'utilisateur
} else {
    // Si le format de l'image est valide
    fileInput.classList.remove("is-invalid"); // ... supprimer la classe is valide
    fileInput.classList.add("blue-border"); // ... ajouter la classe border-blue
    const formData = new FormData();
    const email = JSON.parse(localStorage.getItem("user")).email;
    formData.append("file", file);
    formData.append("email", email);
    this.formData = formData; // alors cela peut etre utiliser dans d'autres méthodes
    this.fileName = fileName;
}
};
```

```
<label for="file" class="bold-label">Justificatif</label>
<input required type="file" accept=".jpg, .jpeg, .png" class="form-control blue-border" data-testid="file" />
<small class="form-text text-muted">Les formats acceptés sont : .jpg, .jpeg, .png</small>
</div>
```

[Bug Hunt] – Dashboard

Description

Je suis connecté en tant qu'administrateur RH, je déplie une liste de tickets (par exemple : statut "validé"), je sélectionne un ticket, puis je déplie une seconde liste (par exemple : statut "refusé"), je ne peux plus sélectionner un ticket de la première liste.

To-do

Comportement attendu : pouvoir déplier plusieurs listes, et consulter les tickets de chacune des deux listes.

Pas besoin d'ajouter de tests

On utilise chrome debugger pour mettre en œuvre des breakpoints, étape par étape et pas à pas dans le fichier JS Dashboard.js du dossier et sous dossiers /front/src/containers / pour localiser le bug et ainsi ajouter du code JS manquant « \$(`#status-bills-container\${this.index}`) » pour qu'il soit possible de déplier la liste des tickets en même temps.

```
handleShowTickets(e, bills, index) {
  if (this.counter === undefined || this.index !== index) this.counter = 0;
  if (this.index === undefined || this.index !== index) this.index = index;
  if (this.counter % 2 === 0) {
    $(`#arrow-icon${this.index}`).css({ transform: "rotate(0deg)" });

    $(`#status-bills-container${this.index}`).html(
      cards(filteredBills(bills, getStatus(this.index)))
    );
    this.counter++;
  } else {
    $(`#arrow-icon${this.index}`).css({ transform: "rotate(90deg)" });
    $(`#status-bills-container${this.index}`).html("");
    this.counter++;
  }
}

bills.forEach((bill) => {
  // fix les erreurs d'expansion de liste dans l'admin
  $(`#status-bills-container${this.index} #open-bill${bill.id}`).click((e) =>
    this.handleEditTicket(e, bill, bills)
  );
});
```

[Bug Hunt] – Fix title display bug

Description

Quand le nom de la dépense est vide on ne peut quand même soumettre la new bill.

To-do

J'ai mis en place un test If sur l'input data-testid= « expense-name » pour n'autoriser le submit de la new bil seulement si le nom de la dépense est supérieur à 5. On affiche un message d'erreur si ce n'est pas le cas.

```
// fix bug affichage titre vide
// Submit quand la valeur du nom de la dépense est supérieur à 5
if (e.target.querySelector(`input[data-testid='expense-name']`).value.length > 4) {
  document.getElementById("errorExpenseName").classList.add("hideErrorMessage");
  const email = JSON.parse(localStorage.getItem("user")).email;
  const bill = {
    email,
    type: e.target.querySelector(`select[data-testid="expense-type"]`).value,
    name: e.target.querySelector(`input[data-testid="expense-name"]`).value,
    amount: parseInt(e.target.querySelector(`input[data-testid="amount"]`).value),
    date: e.target.querySelector(`input[data-testid="datepicker"]`).value,
    vat: e.target.querySelector(`input[data-testid="vat"]`).value,
    pct: parseInt(e.target.querySelector(`input[data-testid="pct"]`).value) || 20,
    commentary: e.target.querySelector(`textarea[data-testid="commentary"]`).value,
    fileUrl: this.fileUrl,
    fileName: this.fileName,
    status: "pending",
  };
  this.updateBill(bill);
  this.onNavigate(ROUTES_PATH["Bills"]);
} else {
  document.getElementById("errorExpenseName").classList.remove("hideErrorMessage");
  this.document.querySelector(`input[data-testid='expense-name']`).value = null;
}
};
```

[Ajout de tests unitaires et d'intégration]

Le rapport de couverture de branche de Jest indique que le fichiers suivants ne sont pas couverts (cf. copie d'écran) :

Respecter la structure des tests unitaires en place : Given / When / Then avec le résultat attendu. Un exemple est donné dans le squelette du test **tests/Bills.js**

- [] composant container/Bills :
 - [] couvrir un maximum de "statements" c'est simple, il faut qu'après avoir ajouté tes tests unitaires et d'intégration [le rapport de couverture du fichier container/Bills](#) soit vert. Cela devrait permettre d'obtenir un taux de couverture aux alentours de 80% dans la colonne "statements".

All files containers

85.05% Statements 182/214 71.64% Branches 48/67 64.81% Functions 35/54 87.06% Lines 175/201

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

File ▲	Statements ▾	Branches ▾
Bills.js	<div><div></div></div> 83.33%	20/24

- [] composant views/Bills : Le taux de couverture est à 100% néanmoins si tu regardes le premier test il manque la mention "expect". Ajoute cette mention pour que le test vérifie bien ce que l'on attend de lui.

```
describe("Given I am connected as an Employee", () => {
  // Test pour BillsUI.js
  // Quand je suis sur la page bills il y a une icone bill sur le cote vertical, doit etre surligné
  describe("When I am on Bills page, there are a bill icon in vertical layout", () => {
    test("Then, the bill icon in vertical layout should be highlighted", () => {
      Object.defineProperty(window, "localStorage", { value: localStorageMock });
      window.localStorage.setItem(
        "user",
        JSON.stringify({
          type: "Employee",
        })
      );
    });
    const html = BillsUI({ data: [] });
    document.body.innerHTML = html;
    expect(document.querySelector("#layout-icon1").classList.contains("active-icon"));
  });
});
```


All files views

100% Statements 33/33 100% Branches 32/32 100% Functions 15/15 100% Lines 29/29

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

File ▲	Statements ▾
Actions.js	100%
BillsUI.js	100%
DashboardFormUI.js	100%
DashboardUI.js	100%
ErrorMessage.js	100%
LoadingPage.js	100%
LoginUI.js	100%
NewBillUI.js	100%
VerticalLayout.js	100%

- [] ajouter un test d'intégration GET Bills. Tu peux t'inspirer celui qui est fait (signalé en commentaires) pour Dashboard.

```
// Test d'intégration GET 404 et 500
describe("Given I am a user connected as Employee", () => {
  describe("When I navigate to Bill", () => {
    test("Then, fetches bills from mock API GET", async () => {
      const getSpy = jest.spyOn(store, "get");
      // obtenir les bills et le nouveau bill
      const bills = await store.get();
      // getspy doit etre appelé au une fois
      expect(getSpy).toHaveBeenCalledTimes(1);
      // le nombre de bills doit etre de 4
      expect(bills.data.length).toBe(4);
    });
    test("Then, fetches bills from an API and fails with 404 message error", async () => {
      store.get.mockImplementationOnce(() => Promise.reject(new Error("Erreur 404")));
      // création interface user avec erreur de code
      const html = BillsUI({ error: "Erreur 404" });
      document.body.innerHTML = html;
      const message = await screen.getByText(/Erreur 404/);
      // attendre le message erreur 404
      expect(message).toBeTruthy();
    });
    test("Then, fetches messages from an API and fails with 500 message error", async () => {
      store.get.mockImplementationOnce(() => Promise.reject(new Error("Erreur 500")));
      // création interface user avec erreur de code
      const html = BillsUI({ error: "Erreur 500" });
      document.body.innerHTML = html;
      // attendre le message erreur 500
      const message = await screen.getByText(/Erreur 500/);
      expect(message).toBeTruthy();
    });
  });
});
```


J'ai mis en place des tests d'intégration GET sur Bills (erreurs 404/500) pour tester ces erreurs.

[] composant container/NewBill :

- [] ajouter un test d'intégration POST new bill. Tu peux t'inspirer celui qui est fait (signalé en commentaires) pour Dashboard.

```
//test intégration POST
describe("When I navigate to Dashboard employee", () => {
  test("Add a bill from mock API POST", async () => {
    const postSpy = jest.spyOn(store, "post");
    const bill = {
      id: "47qAXb6fIm2zOKkLzMro",
      vat: "80",
      imageUrl:
        "https://test.storage.tld/v0/b/billable-677b6.a..f-1.jpg?alt=media&token=c1640e12-a24b-4b11-ae52-529112e9602a",
      status: "accepted",
      type: "Hôtel et logement",
      commentAdmin: "ok",
      commentary: "séminaire billed",
      name: "encore",
      fileName: "preview-facture-free-201801-pdf-1.jpg",
      date: "2004-04-04",
      amount: 400,
      email: "a@a",
      pct: 20,
    };
    const bills = await store.post(bill);
    expect(postSpy).toHaveBeenCalledTimes(1);
    expect(bills.data.length).toBe(5);
  });
  test("Add bills from an API and fails with 404 message error", async () => {
    store.post.mockImplementationOnce(() => Promise.reject(new Error("Erreur 404")));
    const html = BillsUI({ error: "Erreur 404" });
    document.body.innerHTML = html;
    const message = await screen.getByText(/Erreur 404/);
    expect(message).toBeTruthy();
  });
  test("Add bill from an API and fails with 500 message error", async () => {
    store.post.mockImplementationOnce(() => Promise.reject(new Error("Erreur 500")));
    const html = BillsUI({ error: "Erreur 500" });
    document.body.innerHTML = html;
    const message = await screen.getByText(/Erreur 500/);
    expect(message).toBeTruthy();
  });
});
```

- [] composant container/NewBill :
 - [] couvrir un maximum de "statements" : c'est simple, il faut que le rapport de couverture du fichier container/NewBill soit vert (accessible à [cette adresse](#) quand tu auras lancé le serveur). Cela devrait permettre d'obtenir un taux de couverture aux alentours de 80% dans la colonne "statements".

NewBill.js	<div><div></div></div>	96.15%	50/52
------------	------------------------	--------	-------

Rapport de tests avant :

```
$ jest --coverage --noStackTrace --silent src/___tests___/Login.js
FAIL src/___tests___/Login.js
  ● Given that I am a user on login page › When I do fill fields in correct format and I click on admin button Login In › Then I should be identified as an HR admin in app

    TypeError: Cannot read properties of null (reading 'value')

  ● Given that I am a user on login page › When I do fill fields in correct format and I click on admin button Login In › Then I should be identified as an HR admin in app

    TypeError: Cannot read properties of null (reading 'value')

  ● Given that I am a user on login page › When I do fill fields in correct format and I click on admin button Login In › Then I should be identified as an HR admin in app

    expect(jest.fn()).toHaveBeenCalled()

    Expected number of calls: >= 1
    Received number of calls: 0

  ● Given that I am a user on login page › When I do fill fields in correct format and I click on admin button Login In › It should renders HR dashboard page

    expect(received).toBeTruthy()

    Received: null

File | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
--- | --- | --- | --- | --- | ---
All files | 18.85 | 0.51 | 14.49 | 19.91 | 
constants | 50 | 20 | 100 | 50 | 
routes.js | 50 | 20 | 100 | 50 | 16,20-24
usersTest.js | 0 | 0 | 0 | 0 | 
containers | 13.66 | 0 | 7.41 | 14.74 | 
Bills.js | 0 | 0 | 0 | 0 | 7-57
Dashboard.js | 4.44 | 0 | 0 | 5.13 | 9-24,29-35,54,58-181
Login.js | 58.54 | 0 | 36.36 | 58.54 | 29,48-57,63-72,78-92
Logout.js | 0 | 100 | 0 | 0 | 5-13
NewBill.js | 0 | 0 | 0 | 0 | 6-73
views | 45.16 | 23.33 | 35.71 | 46.43 | 
Actions.js | 0 | 100 | 0 | 0 | 5
BillsUI.js | 66.67 | 50 | 60 | 70 | 8,46,48
DashboardFormUI.js | 33.33 | 0 | 0 | 50 | 26
DashboardUI.js | 0 | 0 | 0 | 0 | 10-16
ErrorPage.js | 0 | 0 | 0 | 0 | 4
LoadingPage.js | 0 | 100 | 0 | 0 | 5
LoginUI.js | 100 | 100 | 100 | 100 | 
NewBillUI.js | 0 | 100 | 0 | 0 | 5
VerticalLayout.js | 66.67 | 50 | 100 | 66.67 | 9,12

Test Suites: 1 failed, 1 total
Tests: 2 failed, 6 passed, 8 total
Snapshots: 0 total
Time: 2.93 s, estimated 3 s
error Command failed with exit code 1.
info Visit https://yarnpkg.com/en/docs/cli/run for documentation about this command.
```

Rapport de tests après :

```
$ npm run test

> test
> jest --coverage --noStackTrace --silent

PASS src/__tests__/ErrorPage.js
PASS src/__tests__/LoadingPage.js
PASS src/__tests__/VerticalLayout.js
PASS src/__tests__/routes.js
PASS src/__tests__/Actions.js
PASS src/__tests__/DashboardFormUI.js
PASS src/__tests__/Login.js
PASS src/__tests__/NewBill.js
PASS src/__tests__/Logout.js
PASS src/__tests__/Dashboard.js
PASS src/__tests__/Bills.js
```

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	87.5	81.73	72.86	89.12	
constants	100	100	100	100	
routes.js	100	100	100	100	
usersTest.js	0	0	0	0	
containers	85.05	71.64	64.81	87.06	
Bills.js	83.33	33.33	71.43	81.82	38-63
Dashboard.js	83.33	82.22	64	88.61	18-19,89,156,166-176,185-188
Login.js	73.17	0	45.45	73.17	27,47,58-70,76-92
Logout.js	100	100	100	100	
NewBill.js	96.15	75	77.78	96.15	108-110
views	100	100	100	100	
Actions.js	100	100	100	100	
BillsUI.js	100	100	100	100	
DashboardFormUI.js	100	100	100	100	
DashboardUI.js	100	100	100	100	
ErrorPage.js	100	100	100	100	
LoadingPage.js	100	100	100	100	
LoginUI.js	100	100	100	100	
NewBillUI.js	100	100	100	100	
VerticalLayout.js	100	100	100	100	

```
Test Suites: 11 passed, 11 total
Tests:       58 passed, 58 total
Snapshots:   0 total
Time:        9.534 s
```

Conclusion :

J'ai présenté ici les différentes solutions apportées aux bug, ainsi que les rapports de test et de couverture JEST.