Billed

.. projet 7 ..

Déboguer et tester un SaaS RH

Jonhathan
HO TZU YU HO FUH

OPENCLASSROOMS

Billed

#1

#2

#3

#4

Context

Billed est une entreprise qui produit des solutions Saas destinées aux équipes de ressources humaines.
Dans deux semaines, l'équipe doit montrer la solution qui fonctionne à l'ensemble de l'entreprise

Objectif

L'objectif est la correction des derniers bugs et la mise en place de tests sur la fonctionnalité “note de frais”, avant de la lancer officiellement auprès de nos clients d'ici 2 semaines.

Fix bug #1:

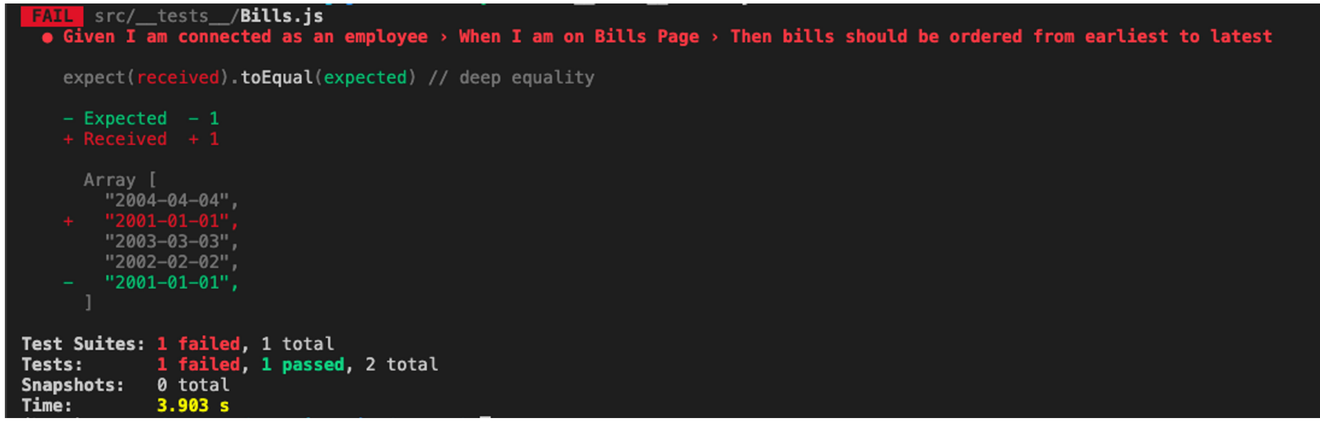
[Bug report] - Bills

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.



Fix bug #1:

[Bug report] - Bills

Localisation du problème

Le test nous indique que l'égalité n'est pas bonne.

Ici on cherche à tester la fonction **const rows** dans le fichier s**rc\views\BillsUI.js** qui retourne un **array des bill**

```
31  test("Then bills should be ordered from earliest to latest", () => {
32    document.body.innerHTML = BillsUI({ data: bills })
33    const dates = screen.getAllByText(/^(19|20)\d\d[- /.](0[1-9]|1[012])[- /.](0[1-9]|[12][0-9]|3[01])$/i).map(a => a.innerHTML)
34    const antiChrono = (a, b) => ((a < b) ? 1 : -1)
35    const datesSorted = [...dates].sort(antiChrono)
36    expect(dates).toEqual(datesSorted)
37  })
```

```
22  const rows = (data) => {
23    return (data && data.length) ? data.map(bill => row(bill)).join("") : ""
24  }
```

=> Corriger le tableau retour de la fonction

Billed

#1

#2

#3

#4

Fix bug #1:

[Bug report] - Bills

Type	Nom	Date
Transports	Vol Paris Marseille	22 Nov. 21
Hôtel et logement	Hôtel du centre ville	22 Nov. 21
Restaurants et bars	Repas d'affaire	22 Nov. 21
Transports	hjjhj	1 Déc. 22
Restaurants et bars	kk	8 Sep. 22
Transports	44m	8 Oct. 22
Restaurants et bars	jii	6 Mai. 22
Transports	ty	8 Jui. 22

Fix bug #1:

[Bug report] - Bills

Utilisation de la méthode sort

Pour organiser le tableau on utilise la méthode sort

```
22  /**
23   * fix #1 : [Bug report] - Bills
24   * @param {array} data
25   * @returns tableau rangé
26   */
27  const rows = (data) => {
28    return data && data.length ?
29      data // ajout de .sort() pour trié les dates par ordre croissant [Bug report] - Bills
30      .sort((a, b) => new Date(b.date) - new Date(a.date))
31      .map((bill) => row(bill))
32      .join("") :
33      "";
34  };
```

Pour raison technique, choix de retirer la fonction de formatage de date **formatDate** dans le fichier **src\containers\Bills.js**

```
58          date: doc.date, //retrait de la fonction formatDate (solution a vérifier) __ fix #1 : [Bug report] - Bills
59          status: formatStatus(doc.status)
```

Billed

#1

#2

#3

#4

Fix bug #1:

[Bug report] - Bills

Type	Nom	Date
Transports	hjjhj	2022-12-01
Transports	44m	2022-10-08
Restaurants et bars	kk	2022-09-08
Transports	ty	2022-07-08
Restaurants et bars	jii	2022-05-06
Transports	Vol Paris Marseille	2021-11-22
Hôtel et logement	Hôtel du centre ville	2021-11-22
Restaurants et bars	Repas d'affaire	2021-11-22

Fix bug #2:

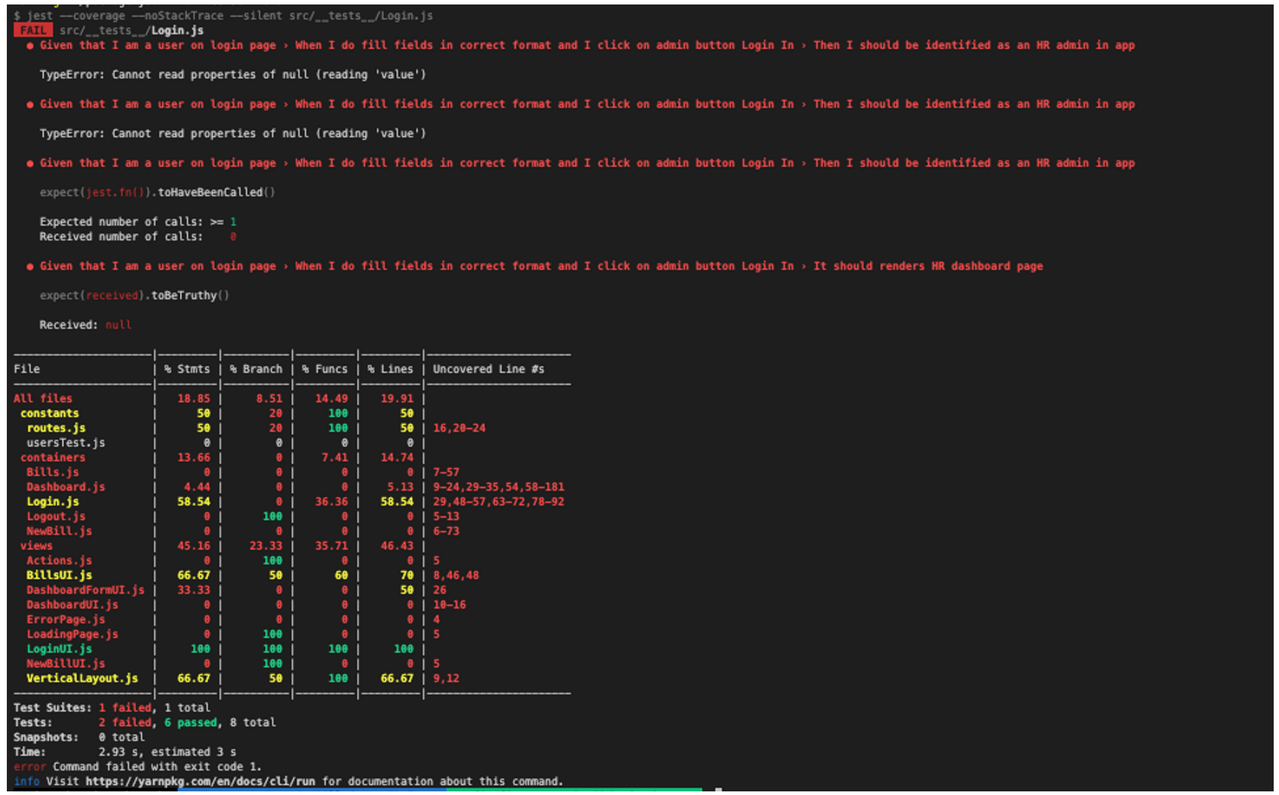
[Bug report] - Login

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 (cf. copie d'écran).

To-do

Faire passer le test au vert en réparant la fonctionnalité.



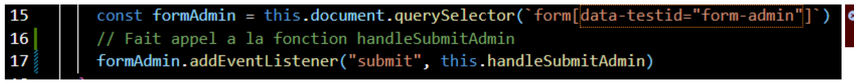
```
$ jest --coverage --noStackTrace --silent src/__tests__/Login.js
 FAIL  src/__tests__/Login.js
  ● Given that I am a user on login page › When I do fill fields in correct format and I click on admin button Login In › Then I should be identified as an HR admin in app

    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 |     8.51 |   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.
```

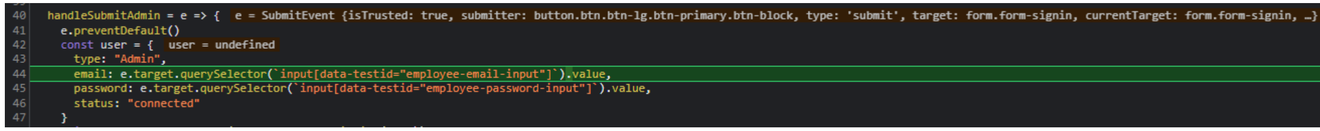
Fix bug #2:

[Bug report] - Login

Test sur debugger chrome

Tentative de connexion sur le dashboard avec les ID admin.
Dans le debugger on place des points d'arrets sur la fonction faisant appel au submit du fomulaire de connexion admin.
Un message d'erreur nous montre que la valeur n'est pas accessible.
On va donc voir sur la fonction appelé **handleSubmitAdmin**
et on remarque que **input[data-testid** s'attendent à trouver employee et non admin





=> Corriger les ligne 44 et 45 du fichier **Billed-app-FR-Front\src\containers\Login.js.** Changer **employee** en **admin**

Fix bug #2:

[Bug report] - Login

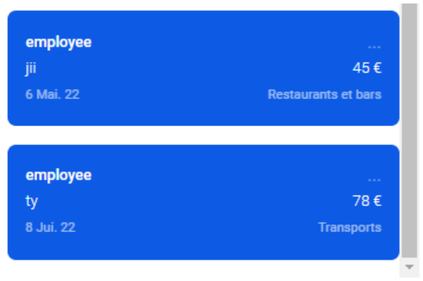
```
40  handleSubmitAdmin = e => {
41    e.preventDefault()
42    const user = {
43      type: "Admin",
44      // fix #2 : [Bug report] - Login
45      email: e.target.querySelector(`input[data-testid="admin-email-input"]`).value, //correction employee-email-input => admin-email-input
46      password: e.target.querySelector(`input[data-testid="admin-password-input"]`).value, //correction employee-password-input => admin-password-input
47      status: "connected"
48    }
```

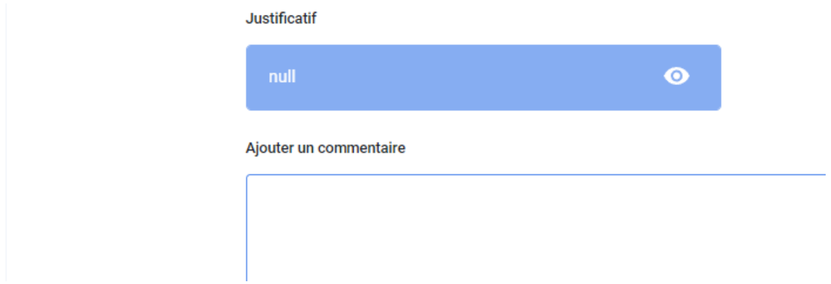
Fix bug #3:

[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.



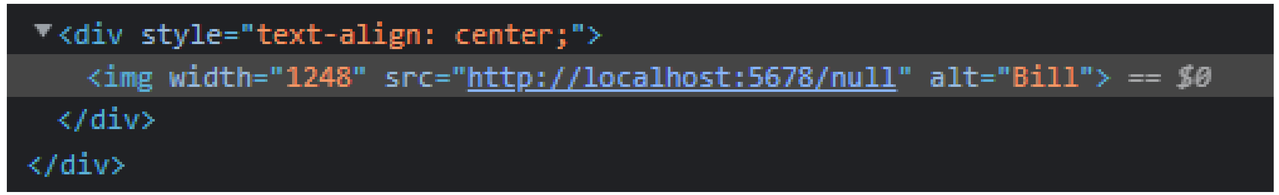


Si je me connecte à présent en tant qu'Admin, et que je clique sur le ticket correspondant, le nom du fichier affiché est null. De même, lorsque je clique sur l'icône «voir» pour consulter le justificatif : la modale s'ouvre, mais il n'y a pas d'image.

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...

Fix bug #3:

[Bug Hunt] - Bills

```
18  handleChangeFile = e => {
19    e.preventDefault()
20    const file = this.document.querySelector(`input[data-testid="file"]`).files[0]
21    const filePath = e.target.value.split(/\\/g)
22    const fileName = filePath[filePath.length-1]
23    const formData = new FormData()
24    const email = JSON.parse(localStorage.getItem("user")).email
25    formData.append('file', file)
26    formData.append('email', email)
27
28    this.store
29      .bills()
30      .create({
31        data: formData,
32        headers: {
33          noContentType: true
34        }
35      })
36      .then(({fileUrl, key}) => {
37        console.log(fileUrl)
38        this.billId = key
39        this.fileUrl = fileUrl
40        this.fileName = fileName
41      }).catch(error => console.error(error))
42  }
```

Fix bug #3:

[Bug Hunt] - Bills

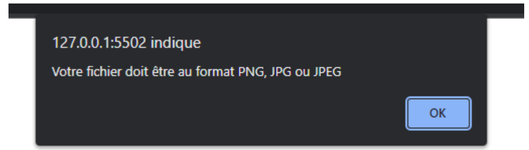
Modification de la fonction

Dans le fichier **src\containers\NewBill.js** mise en place d'un test **RegExp** pour vérifier si le fichier importé à la bonne extension.

Si oui on ajouter le fichier dans le bill

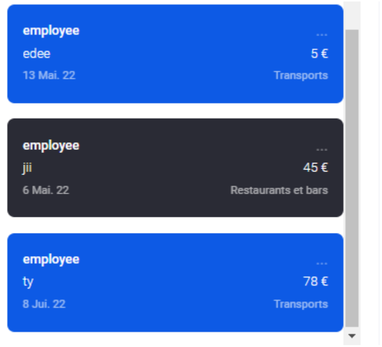
Si non un module **alert** apparait avec un message indiquant le bon format de fichier accepté et aucun changement dans l'**input**

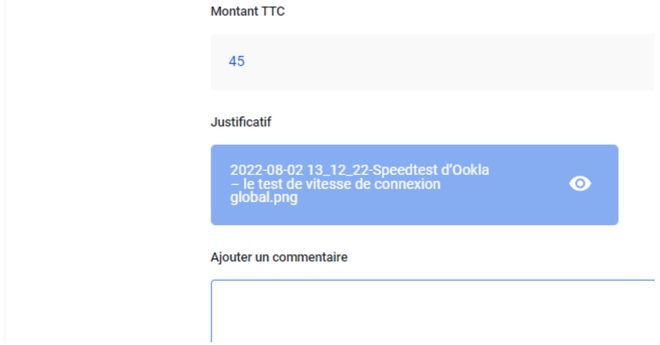
```
handleChangeFile = (e) => {
  e.preventDefault();
  const inputFile = this.document.querySelector(`input[data-testid="file"]`);
  const file = this.document.querySelector(`input[data-testid="file"]`)
    .files[0];
  // fix #3 : [Bug Hunt] - Bills
  // vérifier si l'extension du fichier est bien dans les format jpg|png|jpeg
  // Récupère le fichier
  const fileType = file.type;
  //initialisation du RegExp
  const regexFile = /(jpg|png|jpeg)$/i;
  //test le fichier avec le RegExp
  const testFile = regexFile.test(fileType);
  const filePath = e.target.value.split(/\\/g);
  const fileName = filePath[filePath.length - 1];
  const formData = new FormData();
  const email = JSON.parse(localStorage.getItem("user")).email;
  formData.append("file", file);
  formData.append("email", email);
  //si le test est ok alors on continue
  if (testFile) {
    this.store
      .bills()
      .create({
        data: formData,
        headers: {
          noContentType: true,
        },
      })
      .then(({
        fileUrl,
        key
      }) => {
        this.billId = key;
        this.fileUrl = fileUrl;
        this.fileName = fileName;
      })
      .catch((error) => console.error(error));
    //si le test n'est pas bon on affiche une alert
  } else {
    alert("Votre fichier doit être au format PNG, JPG ou JPEG");
    //le nom de l'image avec mauvaise extension ne doit pas apparaitre
    inputFile.value = "";
  }
};
```

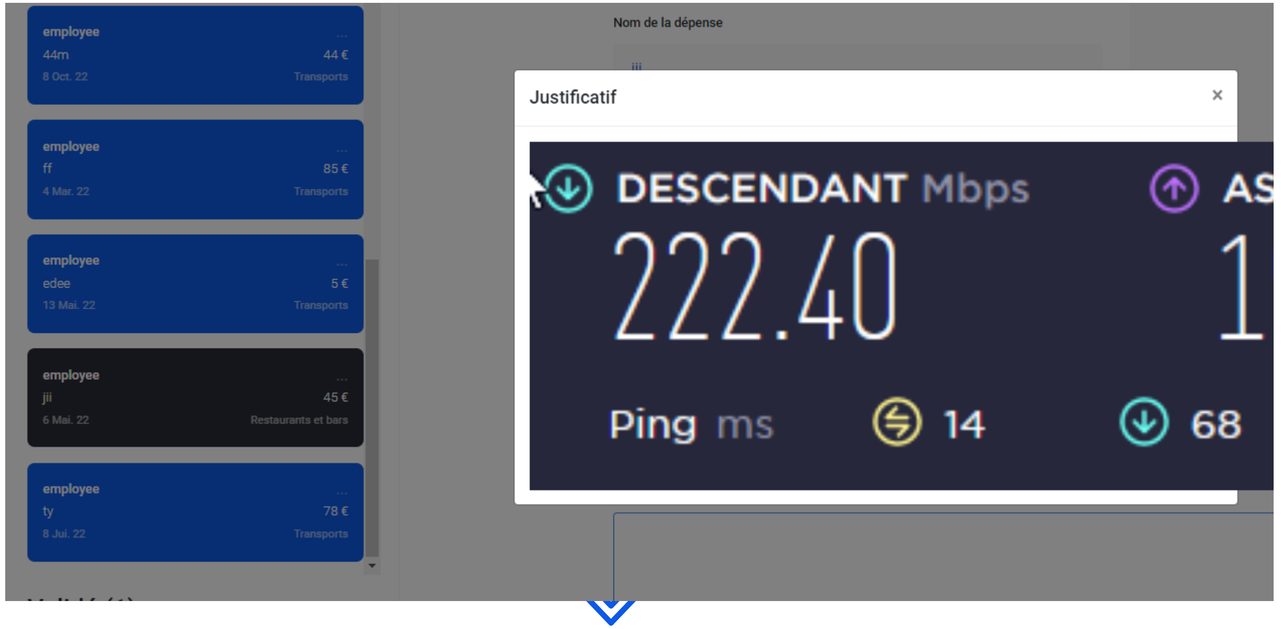


Fix bug #3:

[Bug Hunt] - Bills







Fix bug #4:

[Bug Hunt] - Dashboard

Description

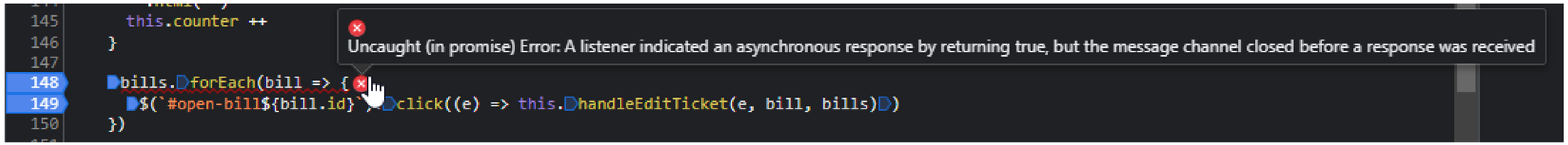
Je suis connecté en tant qu'administrateur RH, je déplie une liste de tickets (par exemple : statut «validé»), je sélectionne un ticket, puis je déplie une seconde liste (par exemple : statut «refusé»), je ne peux plus sélectionner un ticket de la première liste.

To-do

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.

Fix bug #4:

[Bug Hunt] - Dashboard



Fix bug #4:

[Bug Hunt] - Dashboard

```
175       bills.forEach(bill => {
176         // fix #4 : [Bug Hunt] - Dashboard
177         // Pour chaque ticket faire en sorte qu'il ne soit cliqué qu'une fois en fonction de son id
178         // Ajout de .off("click") qui est un évènement jQuery La méthode .off() supprime les gestionnaires d'événements qui étaient attachés avec .on()
179         $(`#open-bill${bill.id}`).off('click').on('click', (e) => this.handleEditTicket(e, bill, bills))
180         return bills
181       })
182     }
```

Fix #5:

[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) :

- [ ] 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.
- [x] ~~composant views/NewBill~~
- [ ] 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".
  - [ ] ajouter un test d'intégration GET Bills. Tu peux t'inspirer de celui qui est fait (signalé en commentaires) pour Dashboard.
- [ ] 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".
  - [ ] ajouter un test d'intégration POST new bill.
- [x] ~~composant views/VerticalLayout~~

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

Billed

#1

#2

#3

#4

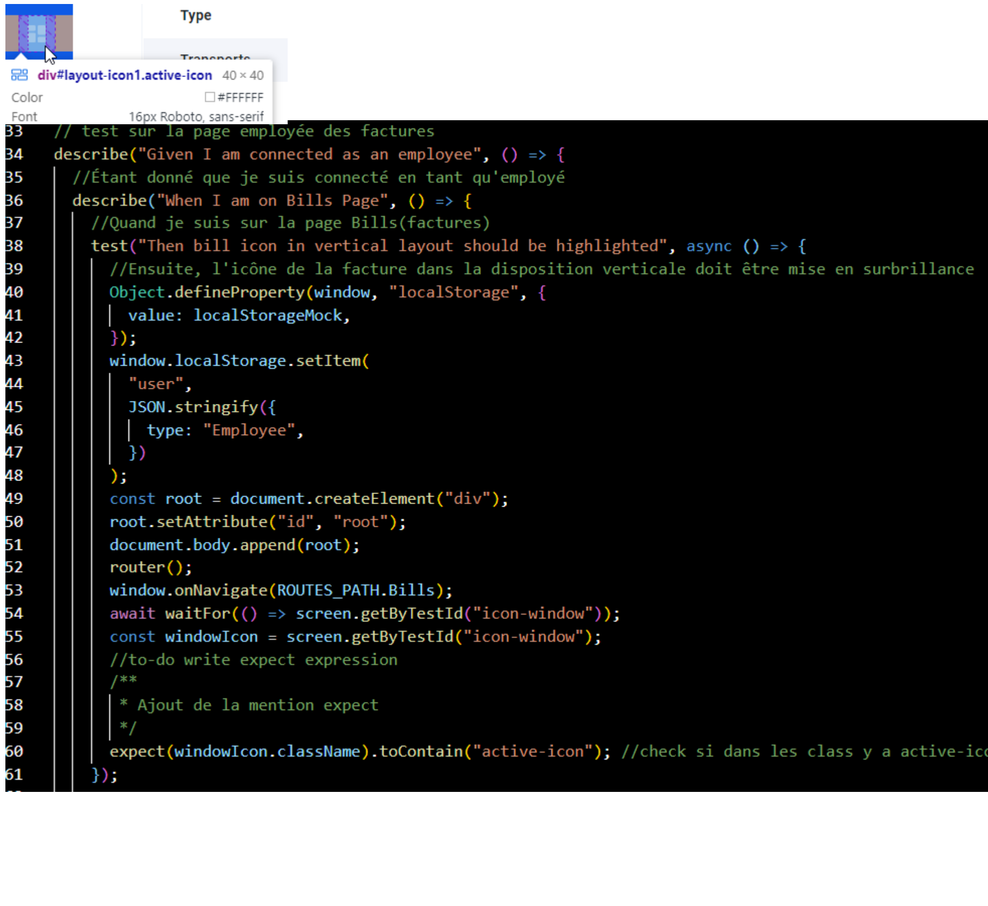
Fix #5:

[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) :

- [x] 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.
- [x] ~~composant views/NewBill~~
- [ ] 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".
  - [ ] ajouter un test d'intégration GET Bills. Tu peux t'inspirer de celui qui est fait (signalé en commentaires) pour Dashboard.
- [ ] 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".
  - [ ] ajouter un test d'intégration POST new bill.
- [x] ~~composant views/VerticalLayout~~

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



Ajout de la mention expect qui retourne un objet « attendu » qui est la class active-icon et on veut qu’il soit “true” pour que le test passe au vert.

Billed

#1

#2

#3

#4

Fix #5:

[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) :

- [x] 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.
- [x] ~~composant views/NewBill~~
- [x] composant container/Bills :
  - [x] 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".
  - [x] ajouter un test d'intégration GET Bills. Tu peux t'inspirer de celui qui est fait (signalé en commentaires) pour Dashboard.
- [ ] 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".
  - [ ] ajouter un test d'intégration POST new bill.
- [x] ~~composant views/VerticalLayout~~

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

```
182     /**
183      * test d'integration get bill
184      */
185     describe("When I get bills", () => {
186       test("Then it should render bills", async () => {
187         const bills = new Bills({
188           document,
189           onNavigate,
190           store: mockStore,
191           localStorage: window.localStorage,
192         });
193         const getBills = jest.fn(() => bills.getBills());
194         const value = await getBills();
195         expect(getBills).toHaveBeenCalled();
196         expect(value.length).toBe(4);
197       });
198     });
```

Avec ce test je vérifie si le nombre de factures récupérées correspond au nombre de factures dans le store

Fix #5:

[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) :

- [x] 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.
- [x] ~~composant views/NewBill~~
- [x] composant container/Bills :
  - [x] 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".
  - [x] ajouter un test d'intégration GET Bills. Tu peux t'inspirer de celui qui est fait (signalé en commentaires) pour Dashboard.
- [x] composant container/NewBill :
  - [x] 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".
  - [x] ajouter un test d'intégration POST new bill.
- [x] ~~composant views/VerticalLayout~~

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

Billed

#1

#2

#3

#4

Rapport de test

```
 PASS  src/__tests__/ErrorPage.js
 PASS  src/__tests__/DashboardFormUI.js
 PASS  src/__tests__/Actions.js
 PASS  src/__tests__/VerticalLayout.js
 PASS  src/__tests__/routes.js
 PASS  src/__tests__/Logout.js
 PASS  src/__tests__/LoadingPage.js
 PASS  src/__tests__/Bills.js
 PASS  src/__tests__/Login.js
 PASS  src/__tests__/NewBill.js
 PASS  src/__tests__/Dashboard.js
----------------------|---------|----------|---------|---------|-------------------
File                  | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
----------------------|---------|----------|---------|---------|-------------------
All files             |   90.12 |     83.7 |   81.16 |   92.04 |
 constants            |     100 |      100 |     100 |     100 |
  routes.js           |     100 |      100 |     100 |     100 |
  usersTest.js        |       0 |        0 |       0 |       0 |
 containers           |   88.18 |    73.68 |   75.47 |   90.43 |
  Bills.js            |   92.86 |    66.67 |     100 |      92 | 64-65
  Dashboard.js        |   89.29 |    80.49 |   79.17 |   95.83 | 167-172
  Login.js            |   73.17 |        0 |   45.45 |   73.17 | 30,53,65-74,80-94
  Logout.js           |     100 |      100 |     100 |     100 |
  NewBill.js          |   95.35 |    83.33 |      75 |   95.35 | 70,114
 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:       53 passed, 53 total
Snapshots:   0 total
Time:        20.053 s
PS D:\openclassroom\Projet_9\projet\Billed-app-FR-Front>
```

Billed

#1

#2

#3

#4

Rapport de test

All files containers

88.18% Statements `179/203` **73.68%** Branches `42/57` **75.47%** Functions `40/53` **90.43%** Lines `170/188`

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

File ▲	Statements		Branches	
Bills.js	92.86%	26/28	66.67%	4/6
Dashboard.js	89.29%	75/84	80.49%	33/41
Login.js	73.17%	30/41	0%	0/4
Logout.js	100%	7/7	100%	0/0
NewBill.js	95.35%	41/43	83.33%	5/6