

PROJET – MISE EN ŒUVRE BDD

SAE5.A.01

TABLE DES MATIÈRES

Introduction.....	2
Mise en place de cucumber sur un projet angularjs.....	3
tests d'acceptation suivant la us.....	7
US : Crée un nouveau vacataire.....	7
US : Modifier un vacataire.....	8
US : Supprimer un vacataire.....	10
US : Crée un nouveau cours.....	12
US : Modifier un cours.....	13
US : Supprimer un cours.....	15

INTRODUCTION

L'objectif de ce projet est de mettre en place, en Gherkin des tests d'acceptation en lien avec les US du projet de la SAE5.A.01. Pour cela, nous devrions implémenter avec Cucumber les features et les implémentations correspondantes.

MISE EN PLACE DE CUCUMBER SUR UN PROJET ANGULARJS

Dans un premier temps, nous devons intégrer Cucumber à notre projet. Celui-ci étant un projet AngularJS avec la version 6 voila comment il faut procéder :

Dans un premier temps, installer la Cypress au projet :

```
ng add @cypress/schematic
```

Ensuite il faut intégrer Cucumber à Cypress avec cette commande :

```
npm i @cypress/webpack-preprocessor @badeball/cypress-cucumber-preprocessor  
cypress
```

Il faut ensuite modifié le fichier : *cypress.config.ts* pour qu'il ressemble ainsi :

```

1 import { defineConfig } from "cypress";
2 import * as webpack from "@cypress/webpack-preprocessor";
3 import { addCucumberPreprocessorPlugin } from "@badeball/cypress-cucumber-preprocessor";
4
5 async function setupNodeEvents(
6   on: Cypress.PluginEvents,
7   config: Cypress.PluginConfigOptions
8 ): Promise<Cypress.PluginConfigOptions> {
9   await addCucumberPreprocessorPlugin(on, config);
10
11   on(
12     "file:preprocessor",
13     webpack({
14       webpackOptions: {
15         resolve: {
16           extensions: [".ts", ".js"],
17         },
18         module: {
19           rules: [
20             {
21               test: /\.ts$/,
22               exclude: [/node_modules/],
23               use: [
24                 {
25                   loader: "ts-loader",
26                 },
27               ],
28             },
29             {
30               test: /\.feature$/,
31               use: [
32                 {
33                   loader: "@badeball/cypress-cucumber-preprocessor/webpack",
34                   options: config,
35                 },
36               ],
37             },
38           ],
39         },
40       },
41     ))
42   );
43
44   // Make sure to return the config object as it might have been modified by the plugin.
45   return config;
46 }
47
48 export default defineConfig({
49   e2e: {
50     specPattern: "**/*.feature",
51     supportFile: false,
52     setupNodeEvents,
53   },
54 });
55

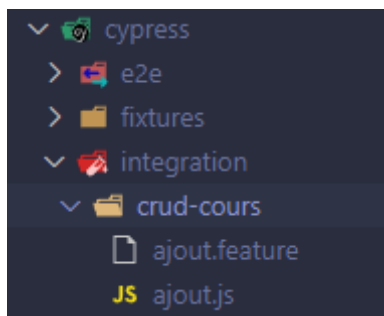
```

Veillez à bien ajouter également Cypress au *package.json* :



```
1 "cypress-cucumber-preprocessor": {  
2   "nonGlobalStepDefinitions": false,  
3   "stepDefinitions": "cypress/integration/**/*.{js,ts}"  
4 }
```

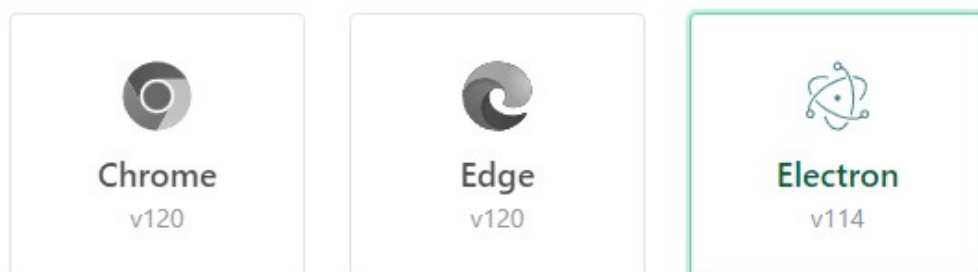
Enfin dans le dossier cypress, il faut ajouter un répertoire pour y stocker toutes nos features et nos StepDefinitions :



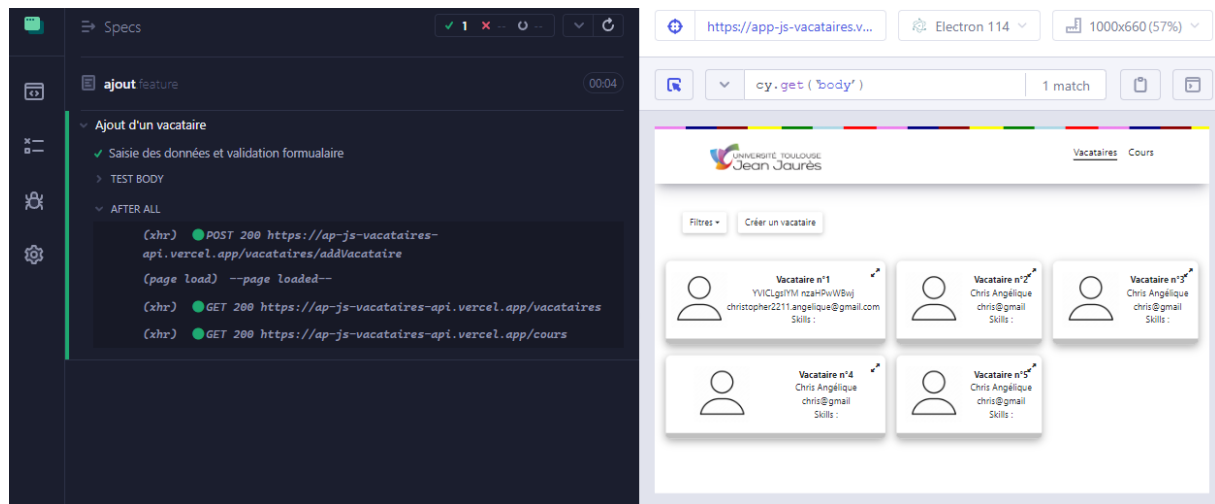
Pour tester l'application, soit lancer les tests, il faut entrer la commande : **npx cypress open** et choisir un Navigateur (Electron pour ma part) :

Choose a browser

Choose your preferred browser for E2E testing.



← Switch testing type



TESTS D'ACCEPTATION SUIVANT LA US

US : CRÉE UN NOUVEAU VACATAIRE

Ce test concerne la US : Crée un nouveau Vacataire.

La feature mise en place est ainsi :

```
1 Feature: Ajout d'un vacataire
2
3 Scenario: Saisie des données et validation formulaire
4     Given je suis sur le modal formulaire de création d'un vacataire
5     When je saisis le prenom : "prenom", la nom : "nom", le numéro : "numero", l'email : "email", le profil : "profil" et je valide le formulaire
6     Then le champ email devrait contenir : "email"
```

“ Si on valide les saisis du formulaire de création d'un vacataire alors il est créé.

Voici son implémentation :

```
1 import { Given, When, Then } from '@badeball/cypress-cucumber-preprocessor';
2
3 Given('je suis sur le modal formulaire de création d'un vacataire', () => {
4     cy.visit('https://app-js-vacataires.vercel.app/vacataires');
5     cy.get('div.btn').click()
6 });
7
8 When('je saisis le prenom : {string}, la nom : {string}, le numéro : {string}, l'email : "email", le profil : {string} et je valide le formulaire', () => {
9     cy.get('#createVacataireModal > .modal-dialog > .modal-content > .modal-body > .row > :nth-child(1) > #firstName').clear().type("Chris")
10    cy.get('#createVacataireModal > .modal-dialog > .modal-content > .modal-body > .row > :nth-child(2) > #lastName').clear().type("Angélique")
11    cy.get('#createVacataireModal > .modal-dialog > .modal-content > .modal-body > .row > :nth-child(3) > #phone').clear().type("0649837162")
12    cy.get('#createVacataireModal > .modal-dialog > .modal-content > .modal-body > .row > :nth-child(4) > #email').clear().type("chris@gmail")
13    cy.get('#createVacataireModal > .modal-dialog > .modal-content > .modal-body > .row > :nth-child(5) > #github').clear().type("github//Christopher")
14
15    cy.get('#createVacataireModal > .modal-dialog > .modal-content > .modal-body > .row > .modal-footer > .btn-primary').click()
16 });
17
18 Then('le champ email devrait contenir : {string}', () => {
19     let email = 'chris@gmail'
20
21     cy.get(':nth-child(2) > .content > .email')
22         .last()
23         .should('have.text', email);
24 });
25
26
```

“ Le GIVEN emmène sur le modal du formulaire de création d'un vacataire.

“ Le WHEN saisit les informations du nouveau vacataire et les soumet en cliquant sur le bouton de validation.

“ Le THEN Vérifie si le vacataire a été ajouté, en comparant les données du dernier avec celle saisit précédemment

```
▼ Ajout d'un vacataire
✓ Saisie des données et validation formulaire
▼ TEST BODY
1 > Given je suis sur le modal formulaire de création d'un vacataire 3
5 > When je saisis le prenom : "prenom", la nom : "nom", le numéro : "numero", l'email 19
   : "email, le profil : "profil" et je valide le formulaire
23 ▼ Then le champ email devrait contenir : "email"
24   get :nth-child(2) > .content > .email
25   last
26   -assert expected <div.email> to have text chris@gmail
```

US : MODIFIER UN VACATAIRE

Ce test concerne la US : Modification d'un Vacataire.

La feature mise en place est ainsi :

```
1 Feature: Modification d'un vacataire
2
3 Scenario: Saisie des nouvelle données et validation formulaire
4   Given je suis sur le modal formulaire de modification d'un vacataire
5   When je change le prénom par : "newPrenom", le nom par "newNom" et je valide le formulaire
6   Then le prenom du vacataire doit avoir été modifier par : "newPrenom" et le nom par : "newNom"
```

“ Si on valide les saisit du formulaire de modification d'un vacataire alors il est modifié.

Voici son implémentation :


```

1 import { Given, When, Then } from '@badeball/cypress-cucumber-preprocessor';
2
3 // Fonction pour générer une chaîne aléatoire
4 function generateRandomString(length) {
5   const characters = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
6   let result = '';
7   for (let i = 0; i < length; i++) {
8     result += characters.charAt(Math.floor(Math.random() * characters.length));
9   }
10  return result;
11 }
12
13 let idVacataire;
14 let newPrenom = generateRandomString(10);
15 let newNom = generateRandomString(10);
16
17 Given('je suis sur le modal formulaire de modification d\'un vacataire', () => {
18   cy.visit('https://app-js-vacataires.vercel.app/vacataires');
19
20   // sauvegarde l'id du vacataires
21   cy.get('.content > :nth-child(5)').last().invoke('text').then((text) => {
22     idVacataire = text.trim();
23   }).then(() => {
24     cy.log('La valeur de idCours est : ${idVacataire}');
25
26     cy.get('.expand > .fa-solid').last().click();
27     cy.get('#vacataireModal-' + idVacataire + ' > .modal-dialog > .modal-content > :nth-child(3) > .btn-secondary').click()
28   });
29
30 });
31
32 When('je change le prénom par : {string}, le nom par {string} et je valide le formulaire', () => {
33   cy.get('#editVacataireModal-' + idVacataire + ' > .modal-dialog > .modal-content > .modal-body > .row > :nth-child(1) > #firstName').clear().type(newPrenom)
34   cy.get('#editVacataireModal-' + idVacataire + ' > .modal-dialog > .modal-content > .modal-body > .row > :nth-child(2) > #lastName').clear().type(newNom)
35   cy.get('#editVacataireModal-' + idVacataire + ' > .modal-dialog > .modal-content > .modal-body > .row > .modal-footer > .btn-primary').click()
36 });
37
38 Then('le prenom du vacataire doit avoir été modifier par : {string} et le nom par : {string}', () => {
39   const fullName = `Vacataire : ${newPrenom} ${newNom}`;
40
41   cy.get('.expand > .fa-solid').last().click()
42
43   cy.get('#vacataireModal-' + idVacataire + ' > .modal-dialog > .modal-content > .modal-body > :nth-child(5)')
44     .should('include.text', idVacataire)
45     .then(() => {
46       cy.get('#vacataireModal-' + idVacataire + ' > .modal-dialog > .modal-content > .modal-header > #vacataireModallabel').should('have.text', fullName);
47     });
48 });
49

```

- “ Le GIVEN emmène sur le modal du formulaire de modification du vacataire et sauvegarde son id dans une variable (que l'on va utiliser par la suite).
- “ Le WHEN saisit les nouvelles informations du vacataire et les soumet en cliquant sur le bouton de validation.
- “ Le THEN compare les données du dernier vacataire pour s'assurer qu'elles correspondent à celle saisit précédemment.

```

  Modification d'un vacataire
  ✓ Saisie des nouvelles données et validation formulaire
  TEST BODY
  1 > Given je suis sur le modal formulaire de modification d'un vacataire 12
  12 > When je change le prénom par : "newPrenom", le nom par "newNom" et je valide le formulaire 9
  21 > Then le prénom du vacataire doit avoir été modifié par : "newPrenom" et le nom par : "newNom"
  22 get .expand > .fa-solid 2
  23 last
  24 -click
      (xhr) ● PUT 200 https://ap-js-vacataires-api.vercel.app/vacataires/editVacataire/65848cafd3e00cfe7ba8b64a
      (page load) --page loaded--
      (xhr) ● GET 200 https://ap-js-vacataires-api.vercel.app/vacataires
      (xhr) ● GET 200 https://ap-js-vacataires-api.vercel.app/cours
  25 get #vacataireModal-65848cafd3e00cfe7ba8b64a > .modal-dialog > .modal-content > .modal-body > :nth-child(5)
  26 -assert expected <p> to contain text 65848cafd3e00cfe7ba8b64a
  27 get #vacataireModal-65848cafd3e00cfe7ba8b64a > .modal-dialog > .modal-content > .modal-header > #vacataireModalLabel
  28 -assert expected <h5#vacataireModalLabel.modal-title> to have text Vacataire : JiXiHGhcfC xunJGVbzEU

```

US : SUPPRIMER UN VACATAIRE

Ce test concerne la US : Suppression d'un Vacataire.

La feature mise en place est ainsi :

```

  Feature: Suppression d'un vacataire
  2
  3 Scenario: Supprimer un vacataire en cliquant sur le bouton
  4   Given je suis sur le modal des informations d'un vacataire
  5   When je clique sur le bouton de suppression et valide l'alerte

```

“ Si on clique sur le bouton de suppression d'un vacataire, alors il est supprimé.

Voici son implémentation :

```
1 import { Given, When, Then } from '@badeball/cypress-cucumber-preprocessor';
2 import { idText } from 'typescript';
3
4 let idVacataire;
5
6 Given('je suis sur le modal des informations d'un vacataire', () => {
7   cy.visit('https://app-js-vacataires.vercel.app/vacataires');
8
9   cy.get('.content > :nth-child(5)').last().invoke('text').then((text) => {
10     idVacataire = text.trim();
11   });
12
13   cy.get('.expand > .fa-solid')
14     .last()
15     .click();
16 });
17
18 When('je clique sur le bouton de suppression et valide l\'alerte', () => {
19   cy.get('#vacataireModal-' + idVacataire + ' > .modal-dialog > .modal-content > :nth-child(3) > .btn-danger').click()
20 });
21
22 Then('le vacataire : {string} doit avoir été Supprimer', () => {
23   cy.get('.content > :nth-child(5)').last().should(($element) => {
24     const text = $element.text().trim(); // Obtenez Le texte de l'élément et supprimez les espaces blancs
25     expect(text).to.not.equal(idText);
26   });
27 });
28
```

- “ Le GIVEN emmène sur le modal du dernier vacataire et ajoute son id à une variable (que l'on utilisera par la suite)
- “ Le WHEN clique sur le bouton de suppression
- “ Le THEN vérifie que le vacataire a bien été supprimé en comparant l'id sauvegardé avec la dernière id des vacataires

Suppression d'un vacataire

✓ Supprimer un vacataire en cliquant sur le bouton

TEST BODY

1 > Given je suis sur le modal des informations d'un vacataire 9

9 > When je clique sur le bouton de suppression et valide l'alerte 3

12 > Then le vacataire : "idVacataire" doit avoir été Supprimer

13 get .content > :nth-child(5) 2

14 last

15 invoke .text()

16 get .content > :nth-child(5) 2

17 last

18 - assert expected 658485bdd3e00cfe7ba8b5e7 to not equal 658485c400fc07c8f43c625b

US : CRÉE UN NOUVEAU COURS

Ce test concerne la US : Créer un nouveau cours.

La feature mise en place est ainsi :

```
1  Feature: Ajout d'un cours
2
3  Scenario: Saisie des données et validation formulaire
4    Given je suis sur le modal formulaire de création d'un cours
5    When je saisis le nom : "nom", le groupe : "groupe" et je valide le formulaire
6    Then le cours : "nom" a été ajouté et le nom du dernier cours est différent d'avant
```

“ Si on valide les saisit du formulaire de création d'un cours alors il est créé.

Voici son implémentation :

```
1  import { Given, When, Then } from '@badeball/cypress-cucumber-preprocessor';
2
3  let nbeCours = 0;
4
5  let nom ;
6  let groupe;
7
8  Given('je suis sur le modal formulaire de création d'un cours', () => {
9    cy.visit('https://app-js-vacataires.vercel.app/cours');
10
11    // sauvegarde du nbe de cours
12    cy.get('.content > .cours-indice').last().invoke('text').then((text) => {
13      nbeCours = text.trim();
14    }).then(() => {
15      cy.log('La valeur de nbe de cours est : ${nbeCours}');
16    });
17
18    cy.get('div.btn').click()
19  });
20
21  When('je saisis le nom : {string}, le groupe : {string} et je valide le formulaire', () => {
22    nbeCours++;
23    nom = 'Nouveau Cours ' + nbeCours;
24    groupe = 'BUT' + nbeCours;
25
26    cy.get('#createCoursModal > .modal-dialog > .modal-content > .modal-body > .row > :nth-child(1) > #name').clear().type(nom)
27    cy.get('#createCoursModal > .modal-dialog > .modal-content > .modal-body > .row > :nth-child(3) > #group').clear().type(groupe)
28
29    cy.get('#createCoursModal > .modal-dialog > .modal-content > .modal-body > .row > .modal-footer > .btn-primary').click()
30  });
31
32  Then('le cours : {string} a été ajouté et le nom du dernier cours est différent d'avant', () => {
33    cy.get('.content > .cours-name > h1').last()
34      .should('have.text', nom);
35  });
36
37
38
```

“ Le GIVEN sauvegarde le nbe de cours et emmène sur le modal de la création d'un cours.

- “ Le WHEN saisit les données du cours et les soumet grâce au formulaire.
- “ Le THEN vérifie que le dernier cours est bien celui créer précédemment en comparant son nom

```

  ✓ Ajout d'un cours
  ✓ Saisie des données et validation formulaire
  ✓ TEST BODY
    1 > Given je suis sur le modal formulaire de création d'un cours 13
    13 > When je saisis le nom : "nom", le groupe : "groupe" et je valide le formulaire 8
    22 > Then le cours : "nom" a été ajouté et le nom du dernier cours est différent d'avant
    23   get .content > .cours-name > h1 15
    24   last
    25   -assert expected <h1> to have text Nouveau Cours 16
      (xhr) ● POST 200 https://ap-js-vacataires-api.vercel.app/cours/addCours
      (page load) --page loaded--
      (xhr) ● GET 200 https://ap-js-vacataires-api.vercel.app/cours
      (xhr) ● GET 200 https://ap-js-vacataires-api.vercel.app/vacataires

```

US : MODIFIER UN COURS

Ce test concerne la US : Modifier un Cours.

La feature mise en place est ainsi :

```

  1 Feature: Modification d'un cours
  2
  3 Scenario: Saisie des nouvelle données et validation formulaire
  4   Given je suis sur le modal formulaire de modification d'un cours
  5   When je change le nom : "nom", par le nouveau nom : "newNom" et je valide le formulaire
  6   Then le nom du cours avec l'id : "idCours" doit avoir été modifier par : "newNom"
  7

```

- “ Si on clique sur le bouton de suppression d'un vacataire, alors il est supprimé.

Voici son implémentation :

```

1  import { Given, When, Then } from '@badeball/cypress-cucumber-preprocessor';
2
3  // Fonction pour générer une chaîne aléatoire
4  function generateRandomString(length) {
5      const characters = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
6      let result = '';
7      for (let i = 0; i < length; i++) {
8          result += characters.charAt(Math.floor(Math.random() * characters.length));
9      }
10     return result;
11 }
12
13 let newNom = generateRandomString(10);
14 let idCours;
15
16 Given('je suis sur le modal formulaire de modification d'un cours', () => {
17     cy.visit('https://app-js-vacataires.vercel.app/cours');
18
19     // sauvegarde l'id du cours
20     cy.get(':nth-child(1) > .content > .cours-id').invoke('text').then((text) => {
21         idCours = text.trim();
22     }).then(() => {
23         cy.log(`La valeur de idCours est : ${idCours}`);
24     });
25
26     cy.get('.expand > .fa-solid').first().click()
27
28     cy.get('#coursModal-658462b0a0ad6fb15886d59d > .modal-dialog > .modal-content > .modal-footer > .btn-secondary').click()
29 });
30
31 When('je change le nom : {string}, par le nouveau nom : {string} et je valide le formulaire', () => {
32     cy.get('#editCoursModal-' + idCours + ' > .modal-dialog > .modal-content > .modal-body > .row > :nth-child(1) > #name').clear().type(newNom)
33     cy.get('#editCoursModal-' + idCours + ' > .modal-dialog > .modal-content > .modal-body > .row > .modal-footer > .btn-primary').click()
34 });
35
36 Then('le nom du cours avec l'id : {string} doit avoir été modifier par : {string}', () => {
37     cy.get(':nth-child(1) > .content > .cours-name > h1')
38         .should('have.text', newNom)
39
40     cy.get(':nth-child(1) > .content > .cours-id')
41         .should('include.text', idCours);
42 });
43

```

- “ Le GIVEN emmène sur le modal du premier cours et ajoute son id à une variable (que l’on utilisera par la suite).
- “ Le WHEN clique sur le bouton de modification et insère les nouvelles données du cours.
- “ Le THEN vérifie que le cours a bien été supprimé en comparant l’id sauvegardé avec la dernière id des vacataires ainsi que si le nom correspond au nouveau nom après la modification.

```
✓ Modification d'un cours
✓ Saisie des nouvelle données et validation formulaire
✓ TEST BODY

1 > Given je suis sur le modal formulaire de modification d'un cours 11
11 > When je change le nom : "nom", par le nouveau nom : "newNom" et je valide le formulaire 6
17 > Then le nom du cours avec l'id : "idCours" doit avoir été modifier par : "newNom"
18   get :nth-child(1) > .content > .cours-name > h1
19   -assert expected <h1> to have text asZmdJaQsc
    (xhr) ● PUT 200 https://ap-js-vacataires-api.vercel.app/cours/editCours/658462b0ad6fb15886d59d
    (page load) --page loaded--
    (xhr) ● GET 200 https://ap-js-vacataires-api.vercel.app/cours
    (xhr) ● GET 200 https://ap-js-vacataires-api.vercel.app/vacataires
20   get :nth-child(1) > .content > .cours-id
21   -assert expected '<div.cours-id>' to contain text '658462b0ad6fb15886d59d'
```

US : SUPPRIMER UN COURS

Ce test concerne la US : Supprimer un cours.

La feature mise en place est ainsi :

```
1 Feature: Suppression d'un cours
2
3 Scenario: Supprimer un cours en cliquant sur le bouton
4   Given je suis sur le modal des informations d'un cours
5   When je clique sur le bouton de suppression d'un cours et valide l'alerte
6   Then le cours : "idCours" doit avoir été Supprimer
```

“ Si on clique sur le bouton de suppression d'un cours, alors il est supprimé.

Voici son implémentation :


```

1 import { Given, When, Then } from '@badeball/cypress-cucumber-preprocessor';
2 import { idText } from 'typescript';
3
4 let idCours;
5
6 Given('je suis sur le modal des informations d'un cours', () => {
7   cy.visit('https://app-js-vacataires.vercel.app/cours');
8
9   // sauvegarde l'id du cours
10  cy.get('.content > .cours-id').last().invoke('text').then((text) => {
11    idCours = text.trim();
12  }).then(() => {
13    cy.log(`La valeur de idCours est : ${idCours}`);
14    cy.get('.expand > .fa-solid').last().click();
15  });
16
17 });
18
19 When('je clique sur le bouton de suppression d'un cours et valide l'alerte', () => {
20   cy.get('#coursModal-' + idCours + ' > .modal-dialog > .modal-content > .modal-footer > .btn-danger')
21     .click()
22 });
23
24 Then('le cours : {string} doit avoir été Supprimer', () => {
25   let tmpCours;
26
27   cy.get('.content > .cours-id').last().invoke('text').then((text) => {
28     tmpCours = text.trim();
29   });
30
31   cy.get('.content > .cours-id').last().should(($element) => {
32     const text = $element.text().trim(); // Obtenez le texte de l'élément et supprimez les espaces blancs
33     expect(text).to.not.equal(tmpCours);
34   });
35 });
36

```

- “ Le GIVEN emmène sur le modal du dernier cours et ajoute son id à une variable (que l'on utilisera par la suite).
- “ Le WHEN clique supprime le cours en cliquant sur le bouton de suppression.
- “ Le THEN vérifie que le cours a bien été supprimé en comparant l'id sauvegardé avec la dernière id des cours.


```

  ✓ Suppression d'un cours
  ✓ Supprimer un cours en cliquant sur le bouton
  ✓ TEST BODY
    1 > Given je suis sur le modal des informations d'un cours 10
    10 > When je clique sur le bouton de suppression d'un cours et valide l'alerte 3
    13 > Then le cours : "idCours" doit avoir été Supprimer
    14   get .content > .cours-id 4
    15   last
    16   invoke .text()
    17   get .content > .cours-id 4
    18   last
    19   -assert expected 6584936c051ef29d6b90fae8 to not equal 6584936fdbf39d34f91a3a25
      (xhr) ●DELETE 200 https://ap-js-vacataires-
      api.vercel.app/cours/deleteCours/6584936fdbf39d34f91a3a25
      (page load) --page loaded--
      (xhr) ●GET 200 https://ap-js-vacataires-api.vercel.app/cours
      (xhr) ●GET 200 https://ap-js-vacataires-api.vercel.app/vacataires

```

US : FILTRE VACATAIRE (AFFECTER)

La feature mise en place est ainsi :

```

Feature: Filtrage des vacataires
  En tant qu'administrateur
  Je souhaite filtrer la liste des vacataires
  Afin de visualiser les vacataires en fonction de leur affectation

Scenario: Filtrer les vacataires qui son affecté
  Given Je me connecte en tant qu'admin
  When Click du bouton "Filtres"
  And Je choisis l'affectation "Affecté"
  Then La liste des vacataires devrait être filtrée pour afficher uniquement les affecté à un module

```

Voici son implémentation :

```

Given('Je me connecte en tant qu\'admin', () => {
  cy.visit('http://localhost:4200/vacataires');
  cy.get('#pseudo').type('admin');
  cy.get('#password').type('testps');
  cy.get('#pseudo').should('have.value', 'admin');
  cy.get('#password').should('have.value', 'testps');
  cy.contains('Connexion').click();
});

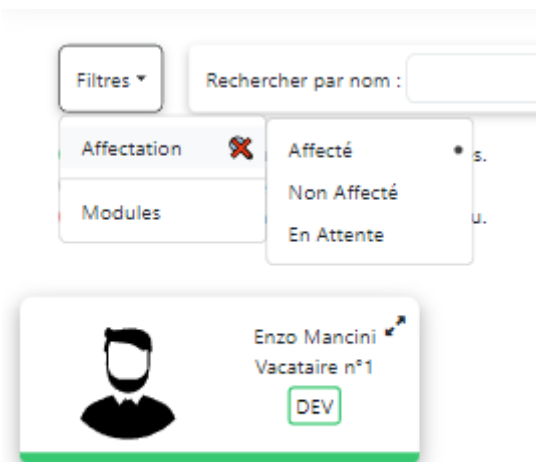
When('Click du bouton "Filtres"', () => {
  cy.wait(1000);
  cy.contains('Filtres').click();
  cy.get('.dropdown-menu').should('be.visible');
});

When('Je choisis l\'affectation "Affecté"', () => {
  cy.get('#affectation').trigger('mouseover');
  cy.wait(1000);
  cy.contains('Affecté').click({ force: true });
});

Then('La liste des vacataires devrait être filtrée pour afficher uniquement les affecté à un module', () => {
  cy.get('.name').should('not.contain', 'Bastien Balmes');
  cy.get('.name').should('contain', 'Enzo Mancini');
  cy.get('.name').should('not.contain', 'Victor Thompson');
});

```

- “ Le GIVEN connecte l'utilisateur et l'amène sur la page d'accueil
- “ Le WHEN clique sur le bouton « Filtres »
- “ Le WHEN passe ensuite le curseur de souris sur « Affectation » puis clique sur « Affecté »
- “ Le THEN affiche tout les vacataires affecté à un cours



```

Filtrage des vacataires
✓ Filtrer les vacataires qui son affecté
▼ TEST BODY
1 > Given Je me connecte en tant qu'admin
13 > When Click du bouton "Filtres"
19 > And Je choisis l'affectation "Affecté"
25 ▼ Then La liste des vacataires devrait être filtrée pour afficher uniquement les affecté à un module
26   get .name
27   -assert expected <div.name> not to contain Bastien Balmes
28   get .name
29   -assert expected <div.name> to contain Enzo Mancini
30   get .name
31   -assert expected <div.name> not to contain Victor Thompson

```

US : FILTRE VACATAIRE (NON AFFECTER)

La feature mise en place est ainsi :

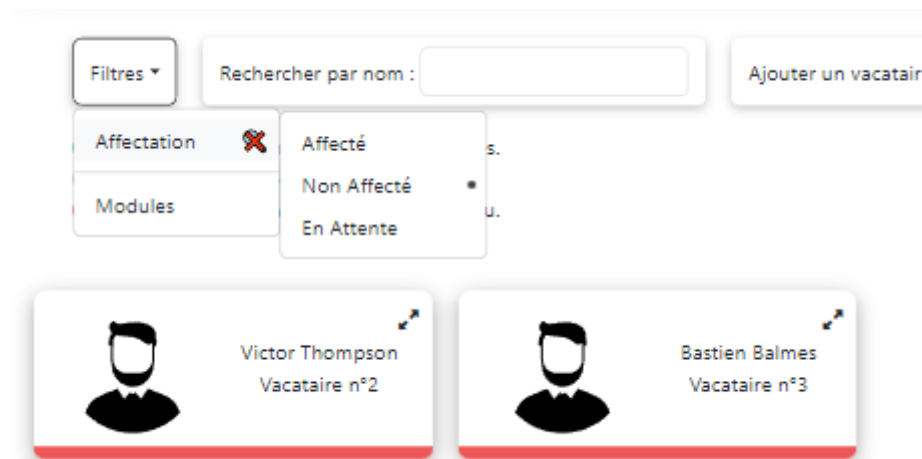
```
Feature: Filtrage des vacataires
  En tant qu'administrateur
  Je souhaite filtrer la liste des vacataires
  Afin de visualiser les vacataires en fonction de leur affectation

  Scenario: Filtrer les vacataires qui son non affecté
    Given Je me connecte en tant qu'admini
    When Clique du bouton "Filtres"
    And Je choisis l'affectation " Non Affecté"
    Then La liste des vacataires devrait être filtrée pour afficher uniquement ceux qui ne son pas affecté
```

Voici son implémentation :

```
Given('Je me connecte en tant qu\'admini', () => {
  cy.visit('http://localhost:4200/vacataires');
  cy.get('#pseudo').type('admin');
  cy.get('#password').type('testps');
  cy.get('#pseudo').should('have.value', 'admin');
  cy.get('#password').should('have.value', 'testps');
  cy.contains('Connexion').click();
});
When('Clique du bouton "Filtres"', () => {
  cy.wait(1000);
  cy.contains('Filtres').click();
  cy.get('.dropdown-menu').should('be.visible');
});
When('Je choisis l\'affectation " Non Affecté"', () => {
  cy.get('#affectation').trigger('mouseover');
  cy.wait(1000);
  cy.contains('Non Affecté').click({ force: true });
});
Then('La liste des vacataires devrait être filtrée pour afficher uniquement ceux qui ne son pas affecté', () => {
  cy.get('.name').should('contain', 'Bastien Balmes');
  cy.get('.name').should('not.contain', 'Enzo Mancini');
  cy.get('.name').should('contain', 'Victor Thompson');
});
```

- “ Le GIVEN connecte l'utilisateur et l'amène sur la page d'accueil
- “ Le WHEN clique sur le bouton « Filtres »
- “ Le WHEN passe ensuite le curseur de souris sur « Affectation » puis clique sur « Non Affecté »
- “ Le THEN affiche tout les vacataires qui ne sont pas affecté à un cours



```

Filtrage des vacataires
✓ Filtrer les vacataires qui son non affecté
✓ TEST BODY
1 > Given Je me connecte en tant qu'admin 15
13 > When Clique du bouton "Filtres" 8
19 > And Je choisis l'affectation " Non Affecté" 6
25 > Then La liste des vacataires devrait être filtrée pour afficher uniquement ceux qui ne son pas affecté
26   get .name 2
27   - assert expected [ <div.name>, 1 more... ] to contain Bastien Balmes 2
28   get .name 2
29   - assert expected [ <div.name>, 1 more... ] not to contain Enzo Mancini 2
30   get .name 2
31   - assert expected [ <div.name>, 1 more... ] to contain Victor Thompson 2

```

US : FILTRE VACATAIRE (EN ATTENTE)

La feature mise en place est ainsi :

```

Feature: Filtrage des vacataires
  En tant qu'administrateur
  Je souhaite filtrer la liste des vacataires
  Afin de visualiser les vacataires en fonction de leur affectation

  Scenario: Filtrer les vacataires qui son en attente
    Given Je me connecte en tant qu'Admin
    When Clique Du bouton "Filtres"
    And Je choisis l'affectation " En Attente"
    Then La liste des vacataires devrait être filtrée pour afficher uniquement ceux qui ne son en attente

```

Voici son implémentation :

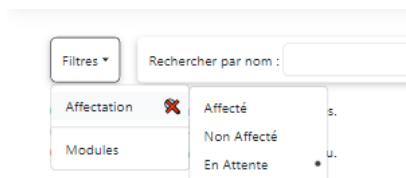
```
Given('Je me connecte en tant qu\'Admin', () => {
  cy.visit('http://localhost:4200/vacataires');
  cy.get('#pseudo').type('admin');
  cy.get('#password').type('testps');
  cy.get('#pseudo').should('have.value', 'admin');
  cy.get('#password').should('have.value', 'testps');
  cy.contains('Connexion').click();
});

When('Clique Du bouton "Filtres"', () => {
  cy.wait(1000);
  cy.contains('Filtres').click();
  cy.get('.dropdown-menu').should('be.visible');
});

When('Je choisis l\'affectation " En Attente"', () => {
  cy.get('#affectation').trigger('mouseover');
  cy.wait(1000);
  cy.contains('En Attente').click({ force: true });
});

Then('La liste des vacataires devrait être filtrée pour afficher uniquement ceux qui ne son en attente', () => {
  cy.contains('Bastien Balmes').should('not.exist');
  cy.contains('Enzo Mancini').should('not.exist');
  cy.contains('Victor Thompson').should('not.exist');
});
```

- “ Le GIVEN connecte l'utilisateur et l'amène sur la page d'accueil
- “ Le WHEN clique sur le bouton « Filtres »
- “ Le WHEN passe ensuite le curseur de souris sur « Affectation » puis clique sur « En Attente »
- “ Le THEN affiche tout les vacataires qui sont en attente



Filtrage des vacataires		
✓ Filtrer les vacataires qui son en attente		
TEST BODY		
1	> Given	Je me connecte en tant qu'Admin 15
13	> When	Clique Du bouton "Filtres" 8
19	> And	Je choisis l'affectation " En Attente" 6
25	▼ Then	La liste des vacataires devrait être filtrée pour afficher uniquement ceux qui ne son en attente
26	-contains	Bastien Balmes 0
27	-assert	expected undefined not to exist in the DOM
28	-contains	Enzo Mancini 0
29	-assert	expected undefined not to exist in the DOM
30	-contains	Victor Thompson 0
31	-assert	expected undefined not to exist in the DOM

US : FILTRE VACATAIRE (MODULES)

La feature mise en place est ainsi :

```
Feature: Filtrage des vacataires
  En tant qu'administrateur
  Je souhaite filtrer la liste des vacataires
  Afin de visualiser les vacataires en fonction de leur module affecté

Scenario: Filtrer les vacataires qui son affecté à un certain module
  Given Je me connect en tant qu'admin
  When Click du Bouton "Filtres"
  And Je choisis le module "DEV"
  Then La liste des vacataires devrait être filtrée pour afficher uniquement les vacataires affecter au module "DEV"
```

Voici son implémentation :

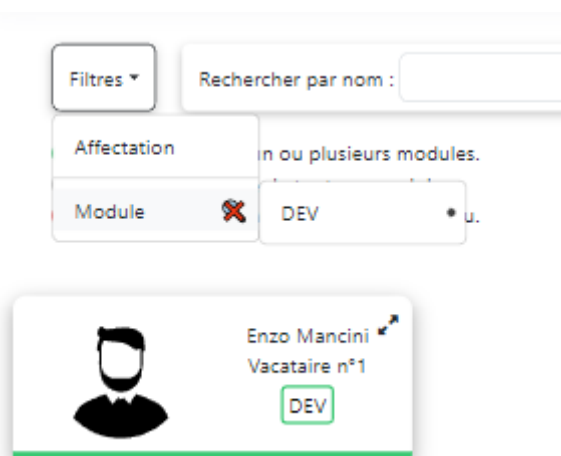
```
Given('Je me connect en tant qu\'admin', () => {
  cy.visit('http://localhost:4200/vacataires');
  cy.get('#pseudo').type('admin');
  cy.get('#password').type('testps');
  cy.get('#pseudo').should('have.value', 'admin');
  cy.get('#password').should('have.value', 'testps');
  cy.contains('Connexion').click();
});

When('Click du Bouton "Filtres"', () => {
  cy.wait(1000);
  cy.contains('Filtres').click();
  cy.get('.dropdown-menu').should('be.visible');
});

When('Je choisis le module "DEV"', () => {
  cy.get('#affectation').trigger('mouseover');
  cy.wait(1000);
  cy.contains('DEV').click({ force: true });
});

Then('La liste des vacataires devrait être filtrée pour afficher uniquement les vacataires affecter au module "DEV"', () => {
  cy.contains('Bastien Balmes').should('not.exist');
  cy.contains('Enzo Mancini');
  cy.contains('Victor Thompson').should('not.exist');
});
```

- “ Le GIVEN connecte l'utilisateur et l'amène sur la page d'accueil
- “ Le WHEN clique sur le bouton « Filtres »
- “ Le WHEN passe ensuite le curseur de souris sur «Modules» puis clique sur «DEV»
- “ Le THEN affiche tout les vaca taires qui sont affecté au module DEV



US : FILTRE COURS (MATIERE)

La feature mise en place est ainsi :

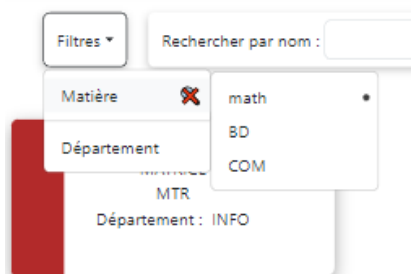
```
Feature: Filtrage des modules
  En tant qu'administrateur
  Je souhaite filtrer la liste des modules
  Afin de visualiser les modules en fonction de différents critères

Scenario: Filtrer les modules par matière
  Given Je me connecte en tant qu'administrateur
  When Click sur le bouton "Filtre"
  And Je choisis la matière "math"
  Then La liste des modules devrait être filtrée pour afficher uniquement les modules de la catégorie "math"
```

Voici son implémentation :

```
Given('Je me connecte en tant qu\'administrateur', () => {
  cy.visit('http://localhost:4200/vacataires');
  cy.get('#pseudo').type('admin');
  cy.get('#password').type('testps');
  cy.get('#pseudo').should('have.value', 'admin');
  cy.get('#password').should('have.value', 'testps');
  cy.contains('Connexion').click();
});
When('Click sur le bouton "Filtre"', () => {
  cy.wait(1000);
  cy.contains('Cours').click();
  cy.contains('Filtres').click();
  cy.get('.dropdown-menu').should('be.visible');
});
When('Je choisis la matière "math"', () => {
  cy.get('#matieres').trigger('mouseover');
  cy.contains('math').click({ force: true });
});
Then('La liste des modules devrait être filtrée pour afficher uniquement les modules de la catégorie "math"', () => {
  cy.contains('GRAPHES');
  cy.contains('MATRICE');
  cy.contains('COM').should('not.contain');
  cy.contains('BD').should('not.contain');
});
```

- “ Le GIVEN connecte l'utilisateur et l'amène sur la page d'accueil
- “ Le WHEN on se dirige vers la page « cours » puis clique sur le bouton « Filtres »
- “ Le WHEN passe ensuite le curseur de souris sur « Matière » puis clique sur « math »
- “ Le THEN affiche tout les modules de la matière « math »



```
Filtrage des modules
✓ Filtrer les modules par matière
✓ TEST BODY
1 > Given Je me connecte en tant qu'administrateur
13 > When Click sur le bouton "Filtre"
21 > And Je choisis la matière "math"
26 ✓ Then La liste des modules devrait être filtrée pour afficher uniquement les modules de la catégorie "math"
27   -contains GRAPHES
28   -contains MATRICE
29   -contains COM
30   -assert expected <a.dropdown-item> not to contain undefined
31   -contains BD
32   -assert expected <a.dropdown-item> not to contain undefined
```

US : FILTRE COURS (DEPARTEMENT)

La feature mise en place est ainsi :

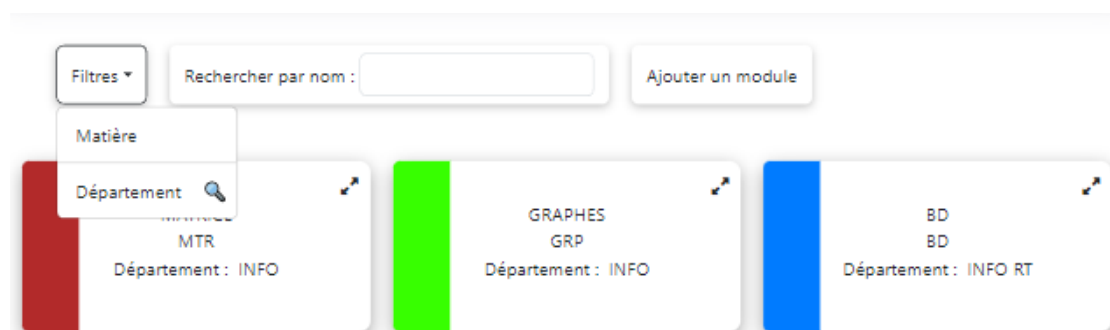
```
Feature: Filtrage par département
  En tant qu'administrateur
  Je souhaite filtrer la liste des modules
  Afin de visualiser les modules en fonction de leur département

  Scenario: Filtrer les modules par département
    Given Je suis connecté en tant qu'administrateur
    When Je clique sur le bouton "Filtre"
    And Je choisis le département "INFO"
    Then La liste des modules devrait être filtrée pour afficher uniquement les modules d'INFO
```

Voici son implémentation :

```
Given('Je suis connecté en tant qu\'administrateur', () => {
  cy.visit('http://localhost:4200/vacataires');
  cy.get('#pseudo').type('admin');
  cy.get('#password').type('testps');
  cy.get('#pseudo').should('have.value', 'admin');
  cy.get('#password').should('have.value', 'testps');
  cy.contains('Connexion').click();
});
When('Je clique sur le bouton "Filtre"', () => {
  cy.wait(1000);
  cy.contains('Cours').click();
  cy.contains('Filtres').click();
  cy.get('.dropdown-menu').should('be.visible');
});
When('Je choisis le département "INFO"', () => {
  cy.get('#matieres').trigger('mouseover');
  cy.contains('INFO').click({ force: true });
});
Then('La liste des modules devrait être filtrée pour afficher uniquement les modules d\'INFO', () => {
  cy.contains('GRAPHES');
  cy.contains('MATRICE');
  cy.contains('BD');
  cy.contains('COM').should('not.contain'); //à modifier
});
```

- “ Le GIVEN connecte l'utilisateur et l'amène sur la page d'accueil
- “ Le WHEN on se dirige vers la page « cours » puis clique sur le bouton « Filtres »
- “ Le WHEN passe ensuite le curseur de souris sur «Département» puis clique sur «INFO»
- “ Le THEN affiche tout les modules du département «INFO»



```
Filtrage par département
✓ Filtrer les modules par département
✓ TEST BODY
1 > Given Je suis connecté en tant qu'administrateur 15
13 > When Je clique sur le bouton "Filtre" 12
21 > And Je choisis le département "INFO" 5
26 ✓ Then La liste des modules devrait être filtrée pour afficher uniquement les modules d'INFO
27   -contains GRAPHES
28   -contains MATRICE
29   -contains BD
30   -contains COM
31   -assert expected <a.dropdown-item> not to contain undefined
```