Rapport Projet BDD R5.A.08

Introduction **2**
Test **3**
Feature: Consulter les informations du site en tant que visiteur 3
Scénario : Afficher le feed Instagram sur la page d'accueil 3
Scénario : Faire défiler la page en tant que visiteur 4
Scénario : Redirection en cliquant sur les menus en tant que visiteur. 5
Scénario: Accéder à la section des courses en tant que visiteur 6
Scénario: Visiter une page qui n’existe pas 7

Leo Douville, Anthony Cabrillac

Introduction

Ce projet BDD, qui rentre dans la ressource R5_08 - Qualité de développement, intervient à la suite de notre SAE qui consistait au développement d'un site Wordpress pour un club de triathlon, recensant des informations sur leur événement principal : le Half du Lauragais.

Le langage de programmation utilisé par Wordpress est PHP. Nous avons donc cherché un moyen de réaliser des tests BDD avec PHP, c'est pour cela que nous avons utilisé behat. Behat est un outil BDD pour PHP, tandis que Cucumber est son équivalent pour d'autres langages comme Java. Les deux utilisent le langage Gherkin pour écrire des scénarios de test en langage naturel.

Pour créer les scénarios, nous avons utilisé les issues de notre projet.

	Title
1	[US] En tant que visiteur, je peux consulter le feed instragram sur le site #3
2	[US] En tant que admin, je peux consulter Google Analytics du site #4
3	[US] En tant que visiteur, je peux scroller le site afin de consulter les informations #5
4	[US] En tant que admin, je peux consulter un guide de sauvegarde du site et accès hébergement #6
5	[US] En tant que visiteur, je me fait rediriger sur les informations de la page lorsque je clique sur les menus #7
6	[US]: En tant que visiteur je peux consulter les courses #39
7	[US] En tant que admin, je suis capable d'assurer une migration #56

Pour réaliser les test nous avons installés avec composer :

- Le framework de test BDD qui utilise le langage Gherkin.
- Behat Mink est une bibliothèque de navigation web pour PHP qui permet d'automatiser les interactions avec des pages web lors de tests.
- Behat Mink Goutte Driver est l'un des drivers (pilotes) disponibles pour Behat Mink. Goutte est un client HTTP pour PHP, et dans le contexte de Behat Mink, le driver Goutte est utilisé pour simuler un navigateur en effectuant des requêtes HTTP directes sans utiliser un navigateur réel.

Leo Douville, Anthony Cabrillac

Test

Feature: Consulter les informations du site en tant que visiteur

Scénario : Afficher le feed Instagram sur la page d'accueil

```
Scenario: Afficher le feed Instagram sur la page d'accueil
  Given Je suis un visiteur
  When Je visite la page d'accueil du site
  Then Je devrais voir le feed Instagram incorporé
```

```php
/**
 * @Given Je suis un visiteur
 */
public function jeSuisUnVisiteur()
{
    $this->visit('http://localhost/migration/');

}

/**
 * @When Je visite la page d'accueil du site
 */
public function jeVisiteLaPageDaccueilDuSite()
{
    $this->visitPath('/');
}

/**
 * @Then Je devrais voir le feed Instagram incorporé
 */
public function jeDevraisVoirLeFeedInstagramIncorpore()
{

    $this->assertPageContainsText('Instagram');

}
```

Résultat des tests :

```
Scenario: Afficher le feed Instagram sur la page d'accueil # features\context.feature:3
  Given Je suis un visiteur                                # FeatureContext::jeSuisUnVisiteur()
  When Je visite la page d'accueil du site                 # FeatureContext::jeVisiteLaPageDaccueilDuSite()
  Then Je devrais voir le feed Instagram incorporé         # FeatureContext::jeDevraisVoirLeFeedInstagramIncorpore()
```

Leo Douville, Anthony Cabrillac

Scénario : Faire défiler la page en tant que visiteur

```
Scenario: Faire défiler la page en tant que visiteur
  Given Je suis un visiteur
  When Je visite le site
  And Je fais défiler la page vers le bas
  Then Je devrais voir toutes les informations disponibles
```

```php
/**
 * @Given Je suis un visiteur
 */
public function jeSuisUnVisiteur()
{
    $this->visit('http://localhost/migration/');

}
```

```php
/**
 * @When Je visite le site
 */
public function jeVisiteLeSite()
{
    $this->visitPath('/');
}

/**
 * @When Je fais défiler la page vers le bas
 */
public function jeFaisDefilerLaPageVersLeBas()
{
    $this->visitPath('/#colophon');
}

/**
 * @Then Je devrais voir toutes les informations disponibles
 */
public function jeDevraisVoirToutesLesInformationsDisponibles()
{

    $infos = array( 'erreur', 'épreuves', 'inscriptions', 'contact',
    'médias', 'triathlon', 'swimrun', 'aquathlon', 'pratique', 'accès', 'hébergement',
    'village', 'éco-responsabilité', 'merci à eux');
    foreach($infos as $info){
        $this->assertPageContainsText($info);
    }
```

Résultats :

```
Scenario: Faire défiler la page en tant que visiteur       # features\context.feature:8
  Given Je suis un visiteur                                # FeatureContext::jeSuisUnVisiteur()
  When Je visite le site                                   # FeatureContext::jeVisiteLeSite()
  And Je fais défiler la page vers le bas                  # FeatureContext::jeFaisDefilerLaPageVersLeBas()
  Then Je devrais voir toutes les informations disponibles # FeatureContext::jeDevraisVoirToutesLesInformationsDisponibles()
```

Erreures :

Leo Douville, Anthony Cabrillac

```
Scenario: Faire défiler la page en tant que visiteur      # features\context.feature:8
  Given Je suis un visiteur                               # FeatureContext::jeSuisUnVisiteur()
  When Je visite le site                                  # FeatureContext::jeVisiteLeSite()
  And Je fais défiler la page vers le bas                 # FeatureContext::jeFaisDefilerLaPageVersLeBas()
  Then Je devrais voir toutes les informations disponibles # FeatureContext::jeDevraisVoirToutesLesInformationsDisponibles()
    The text "erreur" was not found anywhere in the text of the current page. (Behat\Mink\Exception\ResponseTextException)
```

Scénario : Redirection en cliquant sur les menus en tant que visiteur.

```
Scenario: Redirection en cliquant sur les menus en tant que visiteur
  Given Je suis un visiteur
  When Je clique sur le menu de navigation "Informations"
  Then Je devrais être redirigé vers la section des informations de la page
```

```php
/**
 * @When Je clique sur le menu de navigation :arg1
 */
public function jeCliqueSurLeMenuDeNavigation()
{

$path = '//a[contains(@href, "?page_id=596")]';
$moreInfos = $this->getSession()->getPage()->find('xpath', $path);
$moreInfos->click();
}


/**
 * @Then Je devrais être redirigé vers la section des informations de la page
 */
public function jeDevraisEtreRedirigeVersLaSectionDesInformationsDeLaPage()
{
    $expectedUrl = 'http://localhost/migration/?page_id=596';

$currentUrl = $this->getSession()->getCurrentUrl();

if ($currentUrl !== $expectedUrl) {
    throw new \RuntimeException(sprintf(
        'Expected to be redirected to "%s", but current URL is "%s"',
        $expectedUrl,
        $currentUrl
    ));
}
}
```

Résultats :

```
Scenario: Redirection en cliquant sur les menus en tant que visiteur      # features\context.feature:14
  Given Je suis un visiteur                                               # FeatureContext::jeSuisUnVisiteur()
  When Je clique sur le menu de navigation "Informations"                 # FeatureContext::jeCliqueSurLeMenuDeNavigation()
  Then Je devrais être redirigé vers la section des informations de la page # FeatureContext::jeDevraisEtreRedirigeVersLaSectionDesInformationsDeLaPage()
```

Leo Douville, Anthony Cabrillac

Scénario: Accéder à la section des courses en tant que visiteur

```
Scenario: Accéder à la section des courses en tant que visiteur
  Given Je suis un visiteur
  When Je clique sur le menu de navigation "Courses"
  Then Je devrais voir la liste des cours disponibles
  And Chaque cours devrait afficher le titre, la description, et des liens pertinents
```

```
/**
 * @When Je clique sur le menu de navigation :arg1
 */
public function jeCliqueSurLeMenuDeNavigation()
{

$path = '//a[contains(@href, "?page_id=596")]';
$moreInfos = $this->getSession()->getPage()->find('xpath', $path);
$moreInfos->click();
}
```

```
    /**
     * @Then Je devrais voir la liste des cours disponibles
     */
    public function jeDevraisVoirLaListeDesCoursDisponibles()
    {
        $infos = array( 'triathlon', 'swimrun', 'aquathlon');
        foreach($infos as $info){
            $this->assertPageContainsText($info);
        }
    }
```

```
    /**
 * @Then Chaque cours devrait afficher le titre, la description, et des liens pertinents
 */
public function chaqueCoursDevraitAfficherLeTitreLaDescriptionEtDesLiensPertinents()
{
    $expectedUrl = $this->locatePath('/?page_id=596');
    $actualUrl = $this->getSession()->getCurrentUrl();
    if ($expectedUrl !== $actualUrl) {
        throw new \RuntimeException("Expected to be redirected to $expectedUrl, but got $actualUrl.");
    }
}
```

Résultats :

```
Scenario: Accéder à la section des courses en tant que visiteur                  # features\context.feature:19
  Given Je suis un visiteur                                                      # FeatureContext::jeSuisUnVisiteur()
  When Je clique sur le menu de navigation "Courses"                             # FeatureContext::jeCliqueSurLeMenuDeNavigation()
  Then Je devrais voir la liste des cours disponibles                            # FeatureContext::jeDevraisVoirLaListeDesCoursDisponibles()
  And Chaque cours devrait afficher le titre, la description, et des liens pertinents # FeatureContext::chaqueCoursDevraitAfficherLeTitreLaDescriptionEtDesLiensPertinents()
```

Scénario: Visiter une page qui n’existe pas

```
Scenario: Visiter une page qui n'existe pas
  Given Je suis un visiteur
  When Je visite une page qui n'existe pas
  Then Je devrais recevoir une réponse avec un code d'erreur 404
```

```php
    /**
     * @When Je visite une page qui n'existe pas
     */
    public function jeVisiteUnePageQuiNexistePas()
    {
        $this->visitPath('/une-page-qui-nexiste-pas');
    }

   /**
 * @Then Je devrais recevoir une réponse avec un code d'erreur :arg1
 */
public function jeDevraisRecevoirUneReponseAvecUnCodeDerreur($arg1)
{
    $expectedStatusCode = (int) $arg1;
    $actualStatusCode = $this->getSession()->getStatusCode();

    if ($actualStatusCode !== $expectedStatusCode) {
        throw new \RuntimeException(
            sprintf('Expected status code %d but received %d', $expectedStatusCode, $actualStatusCode)
        );
    }
}
}
```

Résultats :

```
Scenario: Visiter une page qui n'existe pas                  # features\context.feature:25
  Given Je suis un visiteur                                  # FeatureContext::jeSuisUnVisiteur()
  When Je visite une page qui n'existe pas                   # FeatureContext::jeVisiteUnePageQuiNexistePas()
  Then Je devrais recevoir une réponse avec un code d'erreur 404 # FeatureContext::jeDevraisRecevoirUneReponseAvecUnCodeDerreur()
```

Leo Douville, Anthony Cabrillac