Application Exercises

Release 0.1

Florian Genilloud

January 14, 2015

CONTENTS

Contenu:

CHAPTER

ONE

L'APPLICATION EXERCICE, C'EST QUOI ?

1.1 Introduction

Voici la documentation de l'application Exercice présente sur le site suivant . Celle-ci vous premettra de pouvoir utiliser la partie création ainsi que la partie résolution des exercices de manière complète et détaillée. Cette application servira par la suite à un professeur de pouvoir créer un exercice (principalement de factorisation ou bien de calcul) et de pouvoir le mettre en ligne. Il suffira de donner le lien de l'exercice à l'élève pour qu'il puisse le résoudre.

Cette application consiste en premier lieu à avoir un support internet sur lequel un élève du Csud pourrait s'entraîner en prévision de ses examens ou alors tout simplement pour perfectionner ses capacités en mathématique dans le domaine de la factorisation et dans celui du calcul. Elle est essentielle au projet pour que les professeurs puissent créer des exercices selon le besoin de leur élève et pour pouvoir analyser les erreurs que les élèves font par rapport à ceux-ci. Cela permet aussi à un élève de savoir où sont ses difficultés et de savoir quelles sont les thèmes qu'il doit travailler. Django permet de pouvoir stocker les données créées par les professeurs dans une base de données et de pouvoir récupérer celle-ci pour en faire des pages. Cela est exactement ce dont nous avons besoin pour cette application car le professeur en question créé un exercice et la partie backend très développée de Django s'occupe de créer la page web contenant les données entrées précédemment.

1.2 La collaboration dans le projet

Pour ce qui est de la collaboration avec les autres applications du projet, il faudrait au minimum que les fonctionnalités suivantes soit disponible:

- La collaboration avec le dashboard élève:
  - L'élève doit pouvoir ajouter les liens des exercices qu'il a trouvé compliqués.
  - Il doit pouvoir avoir un Feedback des exercices qu'il a déjà fait.
  - Il doit pouvoir mettre les liens des exercices à faire pour les devoirs ou autres dans un dossier.
- La collaboration avec le dashboard professeur :
  - Le professeur doit pouvoir faire des dossiers avec les exercices qu'il a créés.
  - Il doit également pouvoir prendre des exercices d'autres professeurs pour les intégrer dans un dossier.
- La collaboration avec les cours:
  - Un cours doit pouvoir contenir les liens des exercices qui sont en rapport avec ceux-ci.
  - Il faudrait pouvoir faire un dossier dans le cours avec les corrigés des exercices faisant partis de celui-ci.
- La collaboration avec les quiz:

- Il faudrait pouvoir mettre en relation un exercice ou bien un groupe d'exercices avec un ou plusieurs quiz ayant le même but pédagogique.

1.3 Intégration de l'application

L'intégration de cette application au reste du projet ne devrait normalement pas poser trop de problèmes. La manière la plus simple de faire correspondre les exercices à des cours ou autres est d'utiliser les liens des exercices pour pouvoir y accéder.

1.4 Le template de base du site

Pour ce qui est du Frontend, suite à une mise en commun avec Benoît Léo, le créateur de l'application quiz, nous nous sommes mis d'accord pour utiliser le même template de base bootstrap ce trouvant ici .

Pour ce qui est de la barre latéral se trouvant à gauche des pages du site, il a fallu mettre des liens vers les différents template. Ceci se fait non pas en recopiant le lien de la page web directement mais en utilisant une commande django simple qui permet, si il y a un changement d'url par la suite dans le fichier urls.py de faire automatiquement le changement pour éviter les erreurs de redirection.

le code est le suivant :

```
<div class="list-group">
    <a href="{% url 'exercises:index' %}" class="list-group-item {% block active-home %}
    active{% endblock %}">Accueil</a>
    <a href="{% url 'exercises:find' %}" class="list-group-item {% block active-reso %}
    {% endblock %}">Résoudre un exercice</a>
    <a href="{% url 'exercises:create' %}" class="list-group-item {% block active-create %}
    {% endblock %}">Création d'exercice</a>
</div>
```

Comme vous pouvez le constater, un block {% block active %} a été ajouté à chaque lien. Ceci permet d'activer la classe "list-group-item" dans la page actuel.

1.5 Les modèles

Les modèles de l'application Exercice ne sont pas très nombreux. Ils servent surtout à la création et à la résolution des exercices. (A compléter)

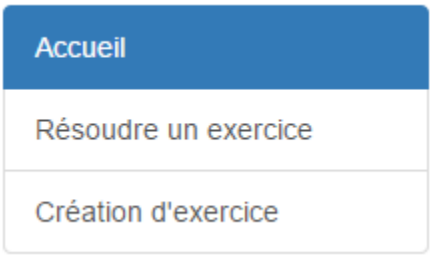
CHAPTER

TWO

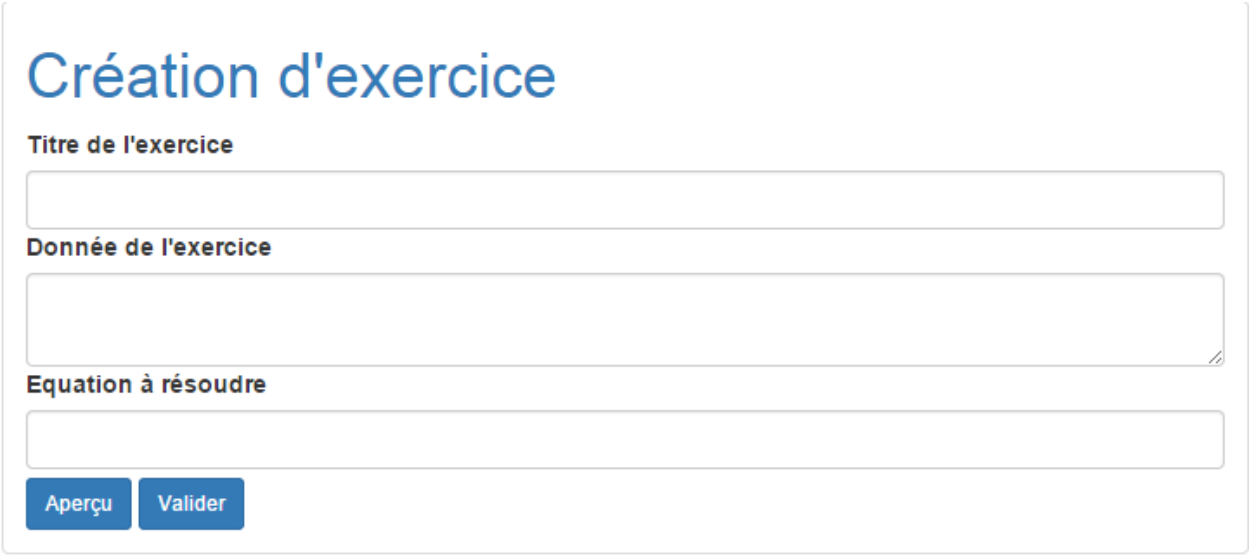
COMMENT, OÙ CRÉER UN EXERCICE

2.1 La page de création d'exercice

L'application exercice possède un onglet création d'exercice , présent sur le menu latéral permettant de renvoyer à la page de création d'exercice.



Cliquez sur l'onglet Création d'exercice pour aller sur la page de création. Une fois sur la page, un encadré ressemblant à ça devrait apparaître:



Tout est dit, il suffit juste de spécifier le titre, la donnée ainsi que l'équation à résoudre pour pouvoir créer un exercice. Si vous observez bien, vous remarquez un bouton aperçu présent à côté de celui nommé valider. Celui-ci , sert une fois avoir insérer les données voulu d'avoir un aperçu de ce qui sera présent dans l'exercice.

2.2 Le code permettant la création

2.2.1 Le template

Dans le template exercises/templates/create.html, rien de spécial y apparait mise à part les 2 <input> et le <textarea> qui seront utilisés dans la vue "create" plus tard.

```
<input type="text" id="title" name="title" class="form-control">
<textarea id="donnee" class="form-control" name="donnee"></textarea>
<input type="text" id="equation" name="equation" class="form-control">
```

Il est à ajouter que pour le bouton aperçu, un code javascript a été nécéssaire.

Le voici:

```
$(document).ready(function() {
    $("#voir").click(function() {
        var formule = $("#equation").val();
        var donnee= $("#donnee").val();
        $("#formule").text(formule);
        $("#donnee-apercu").text(donnee)
        $("#apercu").css({display : "block"})
    })
});
```

Jquery permet de faire une modification du DOM beaucoup plus rapide que du javascript simple.

2.2.2 La vue

Pour ce qui est du code fonctionnant derrière cette partie de mon application, la difficulté se trouve surtout dans la sauvegarde des données.

En effet, il a fallu que pour chaques balises <textarea> ou <input> permettant d'entrer les valeurs du titre, de la donnée et de l'équation puissent être enregistrer dans une variable et les enregistrer dans la base de donnée dans la table "Exercices". Le code qui m'a permis de faire cela se trouve dans le fichier views.py dans la vue "create".

```
def create(request):
if request.method == 'POST': # sauvegarde des données dans la db
    title = request.POST['title']
    donnee = request.POST['donnee']
    equation = request.POST['equation']

    Exercise(title=title, donnee=donnee, equation=equation).save()

    return HttpResponseRedirect(reverse("exercises:index"))
else:
    return render(request, 'exercises/create.html')
```

La partie se trouvant dans le "if" permet de mettre dans des variables les valeurs récupérées. Il est a noté qu'il y a une partie else présente dans cette vue. Cela indique seulement que si il n'y a pas de données à enregistrer, le template se charge normalement. Dès le moment où des données sont enregistrées, l'utilisateur est renvoyé à la page d'accueil. Ceci permet juste que l'utilisateur ne crée pas de doublons en cliquant plusieurs fois sur "Valider" et qu'il comprenne que son exercice a bien été enregistré.

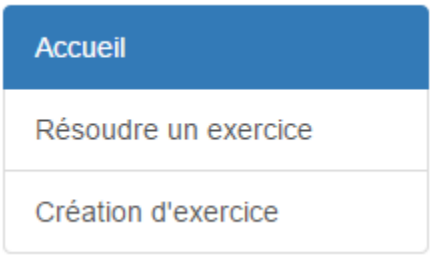
CHAPTER

THREE

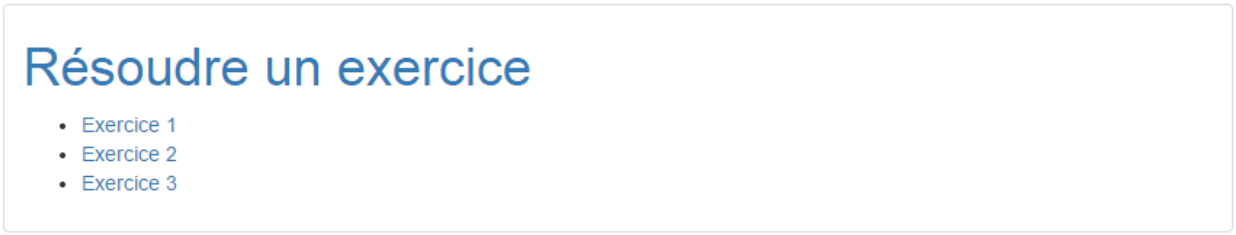
COMMENT, OÙ RÉSOUDRE UN EXERCICE?

3.1 La page de résolution d'exercice

L'application exercice possède un onglet Résoudre un exercice , présent sur le menu latéral permettant de renvoyer à la page de résolution d'exercice.



Cliquez sur l'onglet Résoudre un exercice pour aller sur la page de résolution. Une fois sur la page, une liste ressemblant à ça devrait apparaître:



Comme vous pouvez le constater, tous les exercices créés sont présents sur cette page. Il suffit juste de cliquer sur l'exercice dont vous avez besoin pour être redirigé vers celui-ci.

3.2 Le code permettant la résolution :

3.2.1 Les templates

Le template find.html

Le template de cette page se trouve sous le fichier exercises/templates/find.html . Dans celui-ci, il y a une chose notable qui mérite d'être expliqué c'est la récupération d'urls des exercices.

```
<ul>
    {% for exercise in exercises_list %}
        <li><a href="{% url 'exercises:resolve' exercise.id %}">{{ exercise.title }}</a></li>
    {% endfor %}
</ul>
```

Nous verrons par la suite d'où vient la liste " exercises_list". Mais cette liste permet de créer une liste html avec tous les exercices présents dans la base de données et de faire un lien permettant d'accéder à leur page

Le template resolve.html

Le template resolve.html est le template où apparaissent tous les exercices qui ont été créés par les professeurs. Chaque exercice possède une page et pour chaque nouveau exercice, une nouvelle page est créée. Ceci permet aux exercices de ne pas dépendre les uns des autres et d'être totalement indépendants.

Nous pouvons faire resortir l'appel des éléments de chaque objet appartenant à "Exercises" en les utilisant dans de simples balises html.

```
<h1 id="title">{{ exercise.title }}</h1>
<div class="thumbnail">
    <p id ="donnee">{{ exercise.donnee }}</p>
    <p>{{ exercise.equation }}</p>
    <h6>crée le :{{ exercise.created_on  }}</h6>
    <input type="text" id="equation" name="equation" class="form-control">
</div>
```

3.2.2 Les vues

La vue find

La vue find se trouvant dans le fichier views.py est la vue qui nous permet de récupérer tous les exercices présent dans la base de données. Cette vue est importante car sans elle, le seul moyen d'accéder aux exercices serait de connaître l'url de ceux-ci et de les écrire à la main. Tandis que là, il suffit de se rendre sur la page Résoudre un exercice pour voir tous les exercices qui ont été créés. Cela est très pratique notamment si un élève veut pouvoir s'entraîner sur plusieurs matières ou si tout simplement il a envie de faire quelques exercices de manière aléatoire.

Le code de cette vue est le suivant :

```
def find(request):
latest_exercise_list = Exercise.objects.all()
return render(request, 'exercises/find.html', {"exercises_list" : latest_exercise_list})
```

La fonction objects.all() permet de récuperer tous les exercices présents dans la table Exercises. Il suffit de faire une boucle "for" dans le template pour les afficher.

La vue resolve

La vue resolve se trouvant dans le fichier views.py est la vue qui nous permet d'afficher un exercice dans son template resolve.html et si il n'y a pas d'exercice suite à l'url entré par l'utilisateur, elle renvoit une erreur 404. Grâce à celle-ci, chaque exercice à sa propre page.

Le code de cette vue est assez rudimentaire mais l'import ainsi que l'utilisation de "get_object_or_404" est à noter.

```
from django.shortcuts import get_object_or_404

def resolve(request, n_exercise):
exercise = get_object_or_404(Exercise, id=n_exercise)
return render(request, 'exercises/resolve.html', {"exercise" : exercise})
```

CHAPTER

FOUR

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*