****

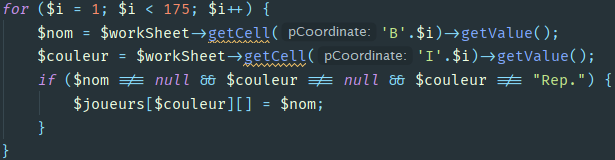
**** [**https://gitlab.com/awuzi/pragolf**](https://gitlab.com/Awuzi/pragolf) ****

**Mission PRAGolf 2**

Dans la continuité de la phase 1, nous avons commencé par créer les entités métiers qui serviront par la suite à alimenter la base de donnée en passant par l’ORM. Nous avons dans un premier temps chercher à comprendre comment extraire les données d’un fichier excel. Pour ce faire, nous avons utilisé une librairie (PhpSpreadsheet) qui permet de lire et d’extraire des informations de différents types de fichier (Word, Excel, etc) et de les transformer en un tableau php.

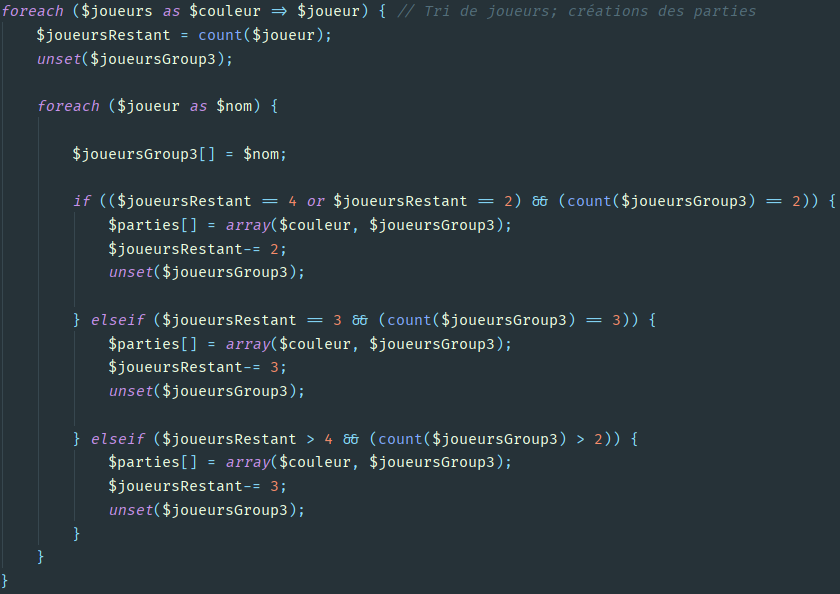
**1) Extraction**

Nous commençons d’abord par instancier cette classe de la manière suivante : $reader = *new* Xlsx();. Ensuite nous chargeons le fichier Excel qui a été au préalable passé en argument de la méthode $spreadsheet = $reader->load($excelFile); . Nous récupérerons ensuite les valeurs des cellules avec les deux méthodes suivante : ->getCell('B12')->getValue(); de ce fait, il nous faut juste parcourir toutes les lignes d’une colonne et vérifier à quelle couleurs elles appartiennent pour les stocker dans un double tableau $joueurs[$couleur][] = $nom; qui stock les noms des joueurs selon leurs couleurs, nous aurons à la fin un “grand” tableau contenant deux autres “petits” tableau qui contiendront tous les joueurs appartenant à la même couleurs.



Une fois que notre tableau de joueurs est rempli, nous passons à la partie qui regroupe les noms par équipes de 3 ou 2 selon leurs couleurs. Pour ce faire, nous parcourons ligne par ligne chacun des deux “sous-tableau” et nous ajoutons les noms des joueurs regroupé dans un autre tableau intermédiaire qui contiendras la couleurs de la partie et le nom des joueurs faisant parti de cette partie.

Il faut cependant prendre en compte le fait que les joueurs ne peuvent pas être seul dans une partie donc nous vérifions grâce à une autre condition notre tableau pour savoir combien il reste de joueur, s’il en reste 4 nous faisons donc deux parties de 2 joueurs.



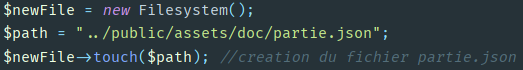
Nous gardons ensuite les informations de la compétition dans un tableau que nous fusionnerons par la suite avec le tableau principal comme ci-dessous.

$infoCompetition = [$nomCompet, $nbJoueurs, $dateCompetition]; array\_push($parties, $infoCompetition);

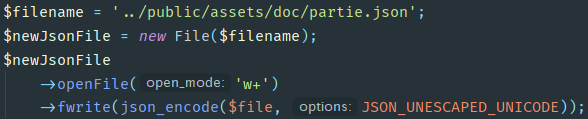
**2) Conversion**

Pour la partie conversion en json nous utilisons le composant FileSystem de symfony qui nous permet de faire des manipulation sur des fichiers et répertoires. Nous spécifions un nouveau chemin et nous créons le nouveau fichier avec la méthode ->touch($path);





De ce point-ci nous récupérons le fichier qui à été ouvert et nous écrivons le json à l'intérieur

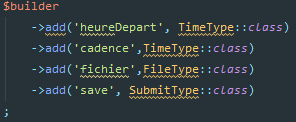


Le paramètre ‘w+’ de la méthode openFile(); nous ouvre un fichier qui est passé en paramètre de la méthode phpToJson($file), et nous supprime ce qu’il y a à l'intérieur, de plus si le fichier n’existe pas il est créé automatiquement. $file, est récupéré grâce au formulaire Symfony expliqué plus bas.

**3) Upload**

Le formulaire d’upload à été réalisé grâce au composant Forms de Symfony.

La première étape est de créer un FormType qui contiendra le builder de notre formulaire (localisé dans src/Form), via la commande > php bin/console make:form. Ce builder permet de créer tous les champs de notre formulaire et de leur donner un type :



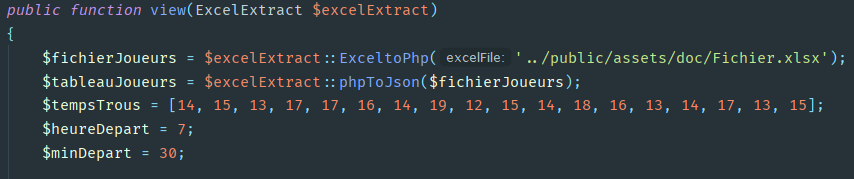
Dans le cas présent, le champ heureDepart permet à l’arbitre de préciser l’heure de départ qu’il souhaite, le champ cadence donnera l’écart de temps entre les parties *(ajouté manuellement au début puis récupéré automatiquement depuis la base de donnée)*, puis enfin le champ fichier lui permet d’envoyer la liste des joueurs qui lui est fourni par la FFGolf.



Le code en image montre la séquence d’envoi du fichier Excel soumis par l’arbitre. Le getter permet de récupérer les données soumises dans le formulaire (notre fichier) qui est ensuite renommé en la valeur suivante $filename = "Fichier" . "." . "xlsx";.

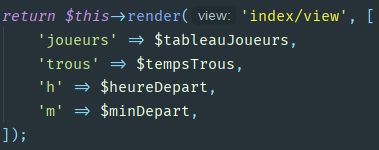
Une fois renommé il est déplacé dans le chemin sauvegardé dans le paramètre 'upload\_directory' qui déplace le fichier dans notre dossier /public/assets/doc du projet pour pouvoir être récupéré pour son extraction. La méthode retourne enfin vers la vue “view” qui affiche le tableau.

**4) Affichage du tableau**

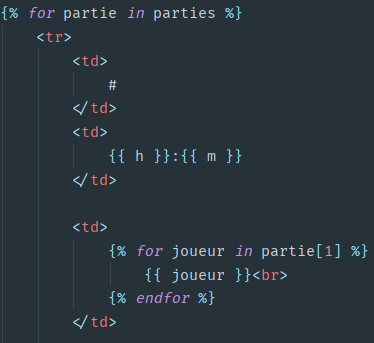
****

D’un point de vu Front-End, nous devons d’abord récupérer le fichiers que nous convertissons, ensuite nous définissons manuellement le temps de marche et le l’heure de départ de la compétition.

Nous envoyons ensuite ces informations au moteur de templates Twig à travers duquel nous pourrons afficher nos informations.



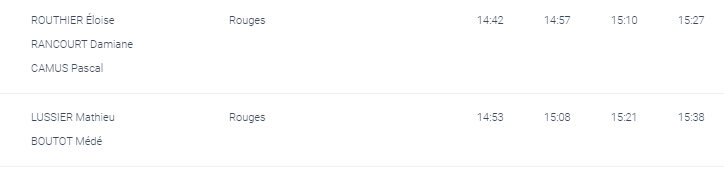
Du côté Twig nous parcourons le tableau 'joueurs', de la manière suivante et nous affichons grâce à une boucle les temps de départ est les différents temps de jeu des parties



Le résultat de la vue twig est le suivant : 4 équipes sont affichées sur 4 trous pour des raisons de lisibilité.



Fin de la liste avec découpage en groupes “égaux”.



**Menu de sélection Administrateur (Back-end / Maintenance) et Arbitre (Client)**

