Cas d'usage : Scripter les opérations d'arrivée d'un cours à partir d'une livraison externe

Scénario

Il s'agit ici d'un scénario d'intégration dans lequel un composant (block_publishflow) transportant des cours entre plusieurs plates-formes Moodle a besoin d'effectuer un certain nombre d'opérations “post-transport” pour recaler certains paramètres et dispositifs du cours à son nouvel environnement. Le scénario ci-dessous décirt le besoin d'opération suite à la mutualisation d'un cours à partir d'une plate-forme A dans un moodle collecteur de mutualisation B.

Parmi les opérations à réaliser :

- Nommer l'utilisateur actif comme “déployeur” (rôle personnalisé) de ce cours
- Supprimer un bloc (course_recycle)
- Déplacer le cours dans une catégorie propre de l'utilisateur qui est à l'origine de l'émission du cours. Crer cette catégorie si elle n'existe pas.
- ajouter un méthode d'inscription par "profil utilisateur" au cours pour accepter les collègues enseignants dans le cours.
- enfin, réaliser une sauvegarde du cours dans le contexte de la publication de cous pour que les collègues puissent déployer.

Note : Ce scénario est en application sur les plates-formes moodle académiques du Rectorat de Rennes et de la plate-forme Atrium Paca.

Réalisation à l'ancienne

En l'absence de MoodleScript, l'ensemble de ce scénario devrait être codé en PHP avec un appel à des API complexes de gestion de fichiers Moodle, les API de role, l'API de catégorie de cours. les API d'inscriptions plus du code spécifique pour le traitement des blocs.

L'estimation de ce développement est de plus de 800 lignes de code et il ne sera pas adaptatif.

Implementation en Moodlescript

La séquence Moodlescript écrite pour effectuer le postprocessing du cours après restauration est :

```
ENROL current IN current AS editingteacher USING manual

ASSIGN ROLE deployer TO current IN current

REMOVE BLOCK course_recycle FROM current

ADD ENROL METHOD profilefield TO current HAVING
profilefield: profile_field_enseignant
profilevalue: 1
role: deployer
```

```
 ADD CATEGORY PATH ":userhostname/:userfullname" IN idnumber:ARRIVALS IF NOT
EXISTS HAVING
 idnumber: ":userwwwroot_:username"

 MOVE COURSE current TO runtime:idnumber:":userwwwroot_:username"

 BACKUP COURSE current FOR publishflow
```

Compte tenu des implicites de contexte courant suivants :

- L'utilisateur courant est celui qui est en train d'effectuer le transfert reconnu par son $USER courant dans la plate-forme d'arrivée du cours.
- Le cours courant est le cours qui vient d'être déployé.

Le script est invoqué dans une instance de moteur de script par le composant block_publishflow qui en a pris le contrôle. Dans ce contexte le bloc publishflow injecte dans la machine de script les contextes ad hoc suivants :

- :userwwwroot : l'identité de la plateforme d'origine de l'utilisateur émetteur du cours (URL).
- :username : l'identifiant de connexion (login) de cet utilisateur
- :userhostname : Le nom “en clair” de la plate-forme d'origine de l'utilisateur
- :userfullname : Le nom “en clair” de l'utilisateur qui mutualise le cours.

Revenir à l'index du composant

From:
https://docs.activeprolearn.com/ - **Documentation Moodle ActiveProLearn**

Permanent link:
https://docs.activeprolearn.com/doku.php?id=local:moodlescript:developerapplicationexample&rev=1553613876

Last update: **2024/04/04 15:52**