

Mini rapport de stage:

Contexte:

L'entreprise EPIXELIC utilise l'application TeamViewer afin d'intervenir sur des machines à distance et ainsi régler les problèmes techniques de leurs clients. Par ailleurs, ils utilisent GLPI afin de suivre les demandes en cours, les problèmes réglé/à régler. Même si il est possible de déposer des commentaires sur TeamViewer, les deux applications sont indépendantes l'une de l'autre

Mission:

Faire en sorte que lorsqu'un technicien dépose un commentaire dans TeamViewer, ce commentaire soit directement mit dans GLPI de manière automatique.

Cahier des charges:

1. Enregistrer la date du dernier lancement du script afin de ne récupérer que les nouveaux commentaires
2. Utiliser les API des applications afin de les mettre en relation.
3. Vérifier si le commentaire commence par @@last, @@no, @@[integer] ou rien pour déterminer dans quel ticket mettre le commentaire
4. Mettre les tâches enregistré en "fait"
5. Lorsqu'un ticket est créer, le créateur est mis en technicien plutôt que demandeur
6. Faire en sorte que les "@@" ne s'affiche pas dans les commentaires
7. Lorsqu'un ticket est spécifié avec @@[integer], vérifier que celui ci est dans le même groupe que l'ordinateur où l'intervention a été faite

Langages de programmations et technologies utilisés:

(Voir Scriptapifinal.php pour étudier le script)

Le script utilise le langage de programmation PHP (version 8.2.6) et un peu de JSON. Afin de déplacer un commentaire d'une application vers une autre, le script fait appel aux API de TeamViewer et GLPI.

Environnement de travail:

J'ai eu le droit d'utiliser un ordinateur fixe avec un écran et le matériel adéquat prêtés pour l'occasion par l'entreprise EPIXELIC. Celui-ci est connecté à Internet et ne possède aucun Localhost (Apache, Xampp, etc). J'ai donc utilisé SublimeText pour écrire mon script PHP et utiliser Powershell pour lancer et tester mon script. (*Voir Notes.txt pour plus de détails*)

Commandes, technologies et environnement découvert:

(*Voir Scriptapifinal.php pour illustrer cette partie*)

Pour la première fois, j'ai travaillé sans aucun localhost. J'ai été obligé d'utiliser Powershell pour lancer mon script. J'ai ainsi pu découvrir comment configurer PHP sans localhost ainsi que ses extensions. J'ai également appris à utiliser Powershell de manière plus approfondie qu'en cours.

Dans les commandes découvertes, la majeure est cURL qui est une extension de PHP (*Voir notes.txt pour savoir comment l'activer*). C'est grâce à ceci que j'ai pu interroger depuis mon script PHP les différentes API des applications.

J'ai également découvert comment enregistrer une date dans un fichier texte puis récupérer celle-ci. J'ai également découvert comment mettre une date sous un bon format.

J'ai ensuite dû utiliser les tableaux JSON car les API n'accepte que ce format de tableau. J'ai ainsi pu découvrir comment transformer un tableau PHP en JSON et inversement. J'ai appris à récupérer des informations rangées dans un tableau PHP (il n'est pas possible de récupérer des informations qui sont stockées dans un tableau JSON, il faut d'abord le transformer en tableau PHP associatif) et ainsi les utiliser aux endroits nécessaires.

Pour continuer, j'ai utilisé la commande "explode" afin de récupérer le premier mot des commentaires récupérés dans TeamViewer et ainsi vérifier grâce aux conditions "if" dans quel ticket ces derniers doivent aller.

J'ai ensuite utilisé les fonctions isset et is_null afin de vérifier si telle ou telle variable est nulle ou non. J'ai également utilisé la fonction preg_match afin de vérifier si un integer se trouve derrière le premier @@ comme demandé dans le cahier des charges et aussi récupérer cette integer si il y en a un.

Pour finir, j'ai utilisé les fonctions implode et array_slice afin de retirer les "@@" des commentaires

(*Voir notes.txt*)

Pour terminer j'ai découvert comment utiliser des API. Chacunes des API utilisent un identifiant qui permet de certifier que je suis autorisé à utiliser ces API.

Pour TeamViewer, il faut commencer par créer un script dans la Management console (En haut à droite, clique sur ton profil => Éditer le profil => Applications => Créer un script. Attention a bien coché les bonnes cases sinon il y aura un problème d'autorisation (dans mon cas j'ai tout coché (à éviter au maximum))). Une fois fait, nous obtenons un Jeton. Il n'y a plus qu'à mettre dans l'en-tête (ou header): "Authorization: Bearer [Lejeton]" (Attention à l'orthographe de "Authorization" et ne pas oublier de mettre Bearer, cela dépend des versions de TeamViewer mais dans celle que j'utilise il le faut obligatoirement). Ensuite il faut rentrer le bon lien ainsi que la bonne méthode (GET, POST ou PUT). Les liens Teamviewer commencent toujours par: "<https://webapi.teamviewer.com/api/v1/>" (dépend des versions TeamViewer).

Pour GLPI, les paramètres nécessaires sont: un App-Token et un Session-Token. L'App-Token s'obtient dans Configuration => Générale => API. Normalement il fonctionne tous mais prudence. Pour obtenir le Session-Token, il faut utiliser le lien: "<https://dev.glpix.net/apirest.php/initSession>". Ce lien a lui-même besoin de l'App-Token et de l'user_token mis dans l'autorization (*voir ScriptapiGLPI.php*). Le user_token est trouvable dans les paramètres profiles, en bas de la première page. Il faut régénérer le "Jeton D'API" une première fois puis sauvegarder. Ne pas oublier de réinitialiser régulièrement les sessions sinon des problèmes de synchronisation entre GLPI et l'API peuvent arriver. Pour détruire une session (ou tuer une session), il faut utiliser le lien suivant: "<https://dev.glpix.net/apirest.php/killSession>". Celui-ci a besoin de l'App-Token et du Session-Token (*voir ScriptapiGLPIkillsession*).