INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES

Raphaël Esteveny
Quentin Bigot
Pierre Testart
Clément Fournier
Jordan Le Bongoat
Tom Richardon
Arnaud Cornillon
Jason Barrier

Spécialité INFO

Rapport de pré-étude

Projet E-Yaka

16 octobre 2017

Table des matières

Introduction

INTRO

Partie 1

Étude de l'existant Raph, Jason

Étude de l'application, fonctionnalités maintenant

1.1 Principe de l'application

1.2 Fonctionnalités de l'application

Partie 2

Description du projet

Intro : Eyaka se diversifie vers des Moocs, essaie d'appliquer des stratégies de learning analytics, puis explications

Plan provisoire

2.1 Contexte

2.1.1 MOOC Raph

2.1.2 Learning analytics Tom, Quentin

2.2 Cahier des charges Arnaud pour rédaction

Problématique

2.2.1 Collecte de traces

2.2.2 Analyse des traces

Partie 3

Technologies

3.1 Experience API Pierre

3.1.1 Présentation et objectif

Experience API, abrégée *xAPI* et anciennement connue sous le nom de TinCan, est une norme pour la déclaration d'informations relatives à un processus d'apprentissage. Son objectif est de mettre en place une spécification pour la communication entre les plateformes d'apprentissage et le contenu des formations.

Experience API est une norme assez récente, sa première version datant de 2013. Elle vise à remplacer SCORM, un modèle plus ancien également utilisé pour normaliser les échanges de données dans le milieu du e-learning. SCORM permet notamment de tracer la complétion, le succès et le temps passé sur une activité.

xAPI offre cependant une plus grande souplesse dans les déclarations d'activités : elle permet d'*enregistrer des actions faites par une équipe* et non pas un seul apprenant, ou bien des activités annexes qui sortent du cadre de la formation en ligne (par exemple, lire un livre en rapport avec la formation).

De plus, SCORM définit un format d'échange spécifiquement pour le Web, ce qui l'empêche de fonctionner avec des technologies plus récentes, comme les applications mobiles et l'Internet des objets. xAPI n'a pas ces limitations car *son fonctionnement est indépendant du système.*

3.1.2 Fonctionnement

Une déclaration, ou *statement*, représente la trace d'une activité d'apprentissage. Ils sont écrits au format *JSON* (*Javascript Object Notation*), qui permet de représenter simplement des objets sous forme de texte. Chaque déclaration doit renseigner au minimum les trois propriétés suivantes :

- *« actor »* : la source de l'action, qui peut être un agent (un individu ou un système), ou bien un groupe d'agents ;
- *« verb »* : l'action effectuée par « actor » ;
- *« object »* : l'activité ou agent sur lequel l'action est faite.

Il existe d'autres propriétés optionnelles, comme « context » qui donne plus d'informations sur le contexte dans lequel s'est déroulée l'activité, ou « result » qui détaille le résultat de l'activité.

Voici un exemple de déclaration valide, qui indique que Paul Durand a assisté à une conférence sur le e-learning :

```
{
  "actor": "Paul Durand",
  "verb": {
    "id": "http://activitystrea.ms/schema/1.0/attend",
    "display": {
      "en-US": "attended"
    }
  },
  "object": "E-learning conference"
}
```

L'exemple ci-dessus montre qu'il est possible de renseigner uniquement du texte pour une propriété, ou de donner davantage de détails, comme c'est le cas ici pour « verb ». L'ID du verbe est une URI (l'identifiant d'une ressource) qui fait référence au verbe « attend » défini dans l'Experience API Registry (une base de ressources en ligne qui permet d'éviter aux utilisateur de xAPI d'avoir à définir leurs propres ressources). L'attribut « display » indique comment afficher le verbe, et il est possible d'y définir un affichage qui dépend de la langue utilisée.

Les déclarations créées par xAPI sont enregistrés dans une base de données appelée *LRS* (*Learning Record Store*). Lorsque la plateforme d'apprentissage (aussi appelée *LMS*, pour *Learning Management System*) a besoin d'informations sur le déroulement de la formation d'un apprenant, elle effectue une requête vers le LRS.

Un LRS peut être intégré à un LMS, ou bien exister de façon séparée. L'important est de stocker l'information de façon normalisée, afin qu'elle soit indépendante du LMS, et puisse être interprétée par des agents extérieurs. Cela ouvre de nombreuses possibilités qui n'étaient pas envisageables avec un système de stockage d'informations spécifique à la plateforme d'apprentissage, par exemple le partage d'informations entre différentes plateformes, ou des études regroupant des données de sources diverses.

3.1.3 Intérêt dans le cadre d'E-Yaka

Experience API est une norme qui s'impose dans le monde de l'e-learning aujourd'hui. Comme nous l'avons décrit précédemment, elle permet une grande flexibilité d'utilisation.

Des bibliothèques existent dans différents langages de programmation pour faciliter l'utilisation de xAPI. E-Yaka est programmé en langage PHP, pour lequel existe la bibliothèque TinCanPHP.

Il semble donc judicieux d'utiliser Experience API dans le cadre de notre projet.

3.2 Greylog Clément

3.3 Autres Jordan

Partie 4

Planification du projet Raph

Conclusion

Organisation du projet / Gestion de projet