

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	Jason	1
PROVISOIRE, n’hésitez pas à changer		1
1 Étude de l’existant	Raph, Jason	2
1.1	Principe de l’application	2
1.2	Fonctionnalités de l’application	2
2 Description du projet		3
2.1	Contexte	3
2.1.1	MOOC Raph	3
2.1.2	Learning analytics Tom, Quentin	3
2.2	Cahier des charges Arnaud pour rédaction	3
2.2.1	Collecte de traces	3
2.2.2	Analyse des traces	3
3 Technologies		4
3.1	Experience API Pierre	4
3.1.1	Présentation et objectif	4
3.1.2	Fonctionnement	4
3.1.3	Intérêt dans le cadre d’E-Yaka	5
3.2	Greylog Clément	5
3.3	Autres technologies	5
3.3.1	Les cookies, une techno efficace pour le logging?	5
3.3.2	Apache Log4php 2	6
3.3.3	BEAMPULSE : Vers l’analyse d’interface	7
4 Planification du projet	Raph	8
Conclusion	Pierre	9

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 utilisateurs 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 enregistrées 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 technologies

3.3.1 Les cookies, une techno efficace pour le logging ?

Un cookie est un fichier de petite taille enregistré sur un ordinateur ou d'autres appareils afin d'autoriser un site web à vous reconnaître quand vous retournez sur ce

site web en utilisant le même ordinateur et moteur de recherche, aussi bien le temps d'une visite (utilisant un « cookie de session ») ou lors de multiples visites (un « cookie persistant »). D'autres fichiers similaires fonctionnent de la même manière et nous utilisons le mot « cookie » pour faire référence à tous les fichiers qui collectent des informations de cette façon.

Les cookies ont beaucoup d'usages différents, par exemple enregistrer vos préférences ou encore vous aider à mieux naviguer entre les différentes pages d'un site. Mais les cookies aident aussi les développeurs de site web à identifier quels contenus du site sont populaire et par la suite les aider à définitivement améliorer leur site et les services offerts à l'utilisateur.

Il existe plusieurs variétés de types de cookies et de technologies de tracking, et une de ces variétés est les Tracking Cookies, qui permettent d'enregistrer les requêtes HTTP faites par une adresse IP (un utilisateur donc).

Exemple de Tracking Cookie simplifié :

```
212.959.120.150 - - [31/Jul/1999:16:58:41 -0400] « GET /last.gif HTTP/1.1 » "http://www.une.org/us  
"Mozilla/4.0" "cookie;d = 999999"
```

On peut voir ici que l'utilisateur correspondant à l'IP 212.959.120.150 à demander à accéder au fichier last.gif sur la page d'accueil du site www.une.org

Cependant, nous ne pourrions pas utiliser cette technologie pour plusieurs raisons, notamment pour une question d'éthique puisque les utilisateurs doivent donner leur consentement préalablement à l'insertion de ces cookies, ce qui pourrait nous empêcher de connaître leurs activités sur l'application. De plus, cela pose des complications technologiques puisque le principe même du cookie est d'être stocké sur l'ordinateur de l'utilisateur en local, empêchant donc la consultation de celui ci si l'utilisateur est hors ligne.

3.3.2 Apache Log4php 2

Presque toutes les grosses applications utilisent leurs propres API de log et de traces. Les frameworks de logging tel que Log4php sont conçus pour limiter la consommation en ressources nécessaires à la mise en œuvre d'une API de logging.

Log4php est un projet open source proposant une API qui permet aux développeurs d'utiliser et paramétrer un système de gestion de journaux (logs). Log4j gère plusieurs niveaux de gravités et les messages peuvent être envoyés dans plusieurs flux : un fichier sur disque, le journal des événements de Windows, une connexion TCP/IP, une base de données, etc ...

Log4php utilise trois composants principaux pour assurer l'envoi de messages selon un certain niveau de gravité et contrôler à l'exécution le format et la ou les cibles de destination des messages :

- Category/Logger : ces classes permettent de gérer les messages associés à un niveau de gravité
- Appenders : ils représentent les flux qui vont recevoir les messages de log
- Layouts : ils permettent de formater le contenu des messages de log

Ces trois types de composants sont utilisés ensemble pour émettre des messages vers différentes cibles de stockage.

Ceci permet au framework de déterminer les messages qui doivent être loggués, la façon de les formater et vers quelle cible les messages seront envoyés.

La popularité de Log4J est largement liée à ses nombreuses fonctionnalités extensibles et sa fiabilité. Cependant, il est le plus souvent utilisé en tant qu'outil de débog et de sécurité et nous paraît plus difficile à implémenter.

3.3.3 BEAMPULSE : Vers l'analyse d'interface

Il existe d'autres technologies intéressantes afin d'améliorer l'utilisation de l'application E-Yaka. Une amélioration possible est l'augmentation de l'efficacité de l'interface web à l'aide d'outils tel que BEAMPULSE. BEAMPULSE est une solution afin d'analyser, améliorer et personnaliser votre site web à l'aide de fonctionnalités telles que :

- Des cartes de chaleurs (Heatmaps) : Heatmaps de clic et de scroll pour visualiser les incompréhensions et blocage sur chaque page ;
- Session Recording (Playback) : Enregistrer et observer le tracé des mouvements de la souris, les scrolls, les clics et les temps d'attente sur le site afin de les rejouer devant le développeur ;
- Personnalisation temps-réel : Modifier le contenu de vos pages ou déclencher automatiquement des interactions ciblées en temps-réel.

Cet objectif d'amélioration de l'interface ne fait pas partis de notre cahier des charges, mais il pourra cependant être envisagé si notre projet se termine en avance, ou encore comme suite du projet.

Partie 4

Planification du projet **Raph**

Conclusion

Organisation du projet / Gestion de projet

