

## GLOSSAIRE

**Version :** 0.1

**Date :** 10/12/2016

**Rédigé par :** L'équipe SmartLogger

**Relu par :** L'équipe SmartLogger

**Approuvé par :** ---

*Signature du superviseur :*

**Objectif :** Ce document est destiné à définir toutes les notions, termes techniques et acronymes utilisés par les différents documents, afin d'en centraliser leur définition et d'en simplifier leur compréhension.

## HISTORIQUE DE LA DOCUMENTATION

| Version | Date     | Modifications réalisées     |
|---------|----------|-----------------------------|
| 0.1     | 10/12/16 | Création du document        |
| 0.2     | 09/01/17 | Ajout de nouveaux mots-clés |

## 1. Définitions et Notions

**Événement** : Un événement représente un changement au sein d'un processus. Il peut s'agir d'un changement en mémoire (ajout d'une donnée), un changement dans l'interface (clic de souris), etc.

**Flux de données** : Un flux de données représente une quantité potentiellement illimitée de données qui est manipulée sur un laps de temps défini afin d'en extraire une certaine quantité.

**Framework** : Un framework est un ensemble de structure informatique cohérentes structurels facilitant la création et la mise en place d'une architecture logicielle. Ils permettent d'accélérer le développement d'application informatique.

**HTTP** : HTTP ou HyperText Transfer Protocol est un protocole réseau de communication Client-Serveur. Il est utilisé dans notre projet afin de recevoir des logs depuis les systèmes informatique externes.

**Itération** : Une itération correspond à une étape de développement du logiciel à fournir. A chaque itération, nous ajoutons un certain nombre de nouvelles fonctionnalités que nous testons vis à vis des précédentes itérations. A la fin de chaque itération, nous envoyons un livrable au client afin qu'il puisse voir les progrès de l'application, et au besoin, modifier le code afin de corriger les potentiels problèmes liées au développement.

**Log** : Un fichier log ou «log» est un fichier contenant l'historique des événements d'un processus en particulier. Ici, nous considérerons des logs issus d'applications WEB externes.

**Machine Learning (ML)** : Champ d'étude de l'intelligence artificielle ayant pour but de faire évoluer, au cours du temps, la façon dont un système effectue un même traitement. Par extension, un algorithme de Machine Learning est un algorithme permettant d'implanter une telle capacité d'apprentissage à un système.

**Traitement en temps réel** : Un traitement est réalisé en temps réel, si le système qui l'effectue peut adapter sa vitesse à l'évolution de ce dernier.

**User Interface (UI)** : Désigne une application pouvant être manipulée par un utilisateur dans le but d'utiliser un système.

## **2. Technologies employées**

**Angular2** : Framework web développé initialement en JavaScript puis depuis sa version 2.0 en TypeScript. Son utilisation dans le projet consistera à produire du projet du côté FrontEnd.

**Apache Kafka** : Projet open-source visant à fournir un système unifié en temps réel à latence faible pour la manipulation de flux de données.

**Apache Spark** : Apache Spark est un framework de calcul distribué. Il propose la mise en place de nombreux algorithmes de type Machine Learning. Dans le cadre de notre projet, Spark sera utilisé par le module analytique du projet.

**JSON** : JSON ou JavaScript Object Notation est un langage permettant de définir des objets dans le langage JavaScript.

**JUnit** : JUnit est un framework de test unitaire développé pour le langage de développement Java. Ce framework est conçu dans l'optique de réaliser des tests sur des classes données, avant de valider ces dites-classes.

**jQuery** : JQuery est une bibliothèque JavaScript facilitant le maniement du DOM ou encore des éléments CSS.

**Logstash** : Outil informatique permettant de gérer des événements et des logs.

**MongoDB** : SGBD orienté documents, il permet de manipuler des données sans avoir à concevoir la façon dont ces dernières seront gérées en interne.

**NoSQL** : Désigne une certaine famille de SGBDs pouvant manipuler de plus grands volumes de données en outrepassant d'anciennes règles pré-établies sur les autres types de SGBD.

**Scala** : Langage multi-paradigme (Programmation Orienté Objet et Programmation Fonctionnelle), étant une surcouche Java, utilisable notamment pour le développement du module analytique du projet.

**SGBD** : Système de gestion de base de données, permet de stocker, manipuler et organiser un (très) grand nombre de données diverses.

**Shinken** : Application permettant la surveillance de systèmes et de réseaux. Elle surveille les hôtes et services spécifiés, lançant une alerte lorsque les systèmes vont mal et quand ils vont mieux.

**Slack** : Logiciel de gestion de projets, principalement utilisé pour son système de communication entre membres d'équipes de projet.

**Spring Boot** : Spring est un framework conçu pour développer des applications en Java ; Dans notre projet, il sera présent dans toute la partie Backend du projet, afin d'uniformiser le code produit.

**TypeScript** : Langage de programmation libre ayant pour but d'améliorer et d'augmenter la sécurité de code JavaScript.

### 3. Définition des Rôles utilisateurs

- **Système Informatique** : Principaux bénéficiaires du projet. On parle des différents systèmes informatiques d'entrée tels que Shinken, Apache Kafka ou encore Logstash ou de sortie tels que Slack, ou encore d'e-mail.
- **Entraîneur** : L'entraîneur est un utilisateur de SmartLogger ayant pour rôle d'envoyer des données d'entraînement au système analytique<sup>1</sup> afin que celui-ci soit ensuite apte à travailler seul et pouvoir effectuer sa tâche.
- **Testeur** : Les testeurs sont des entraîneurs ayant la possibilité en plus de modifier le comportement du système analytique via une interface dédiée.
- **Opérateur** : Les opérateurs sont des utilisateurs polyvalents capable d'assurer le rôle de testeur et ayant en plus la capacité de donner des ordres directs au système afin que celui-ci puisse entre autre réutiliser des données exploitables.

---

1 ***Voir Machine Learning***