

Plateforme d'analytics

Cahier des charges du projet

CONTACT

kar@kmarques.dev

1 - Pré-requis

1 - Consignes

Nombre de personnes par groupe : **4**

Règles de constitution des groupes : **Libre**

2 - Objectifs

Sujet du projet : **Mise en place d'une plateforme de Web Analytics**

Type de rendu : **Code + Soutenance**

Date de rendu de projet :

- **Soutenance fin de semestre**

Ce qu'il va falloir développer :

- **Api de la plateforme (NodeJS)**
- **SDK frontend (VueJS)**
- **BackOffice de la plateforme (VueJS)**
- **Site de tests (VueJS)**
 - **Tracker plusieurs événements**
 - **Tracker changement de vues/pages**

2 - Cahier des charges

1 - Description du projet

Le but du projet est de développer une plateforme de Web analytics.

Le scope du projet s'arrête à la récupération de données côté frontend + backend côté api de la plateforme. En résumé, les interconnexions à créer sont celles entre le visiteur / le site de tests / l'API analytics / le backoffice de la plateforme.

2 - Processus d'inscription à la plateforme

Un webmaster s'enregistre sur la plateforme avec au minimum :

- Un nom de société
- Un KBIS
- Des informations de contact
- L'URL de base de son site

Une fois le webmaster inscrit, un email de confirmation lui est envoyé lui spécifiant que son compte est en attente de validation par un Admin.

Une fois le compte validé, le webmaster peut intégrer le SDK de la plateforme sur son site en y indiquant son APP_ID.

3 - SDK de la plateforme

Spécifications fonctionnelles :

- Détection d'un nouveau visiteur
- Détection des sessions de visites d'un visiteur (15 min d'inactivité depuis la dernière action = fin de visite)
- Suivi des pages vues : collecte des informations sur les pages visitées par les utilisateurs, y compris l'URL, le titre, etc.
- Suivi des événements : possibilité de suivre des événements spécifiques déclenchés par les utilisateurs, tels que des clics, des soumissions de formulaire, des mouvements de souris, etc.
- Personnalisation des données : permettre aux développeurs de définir des paramètres supplémentaires à suivre en fonction de leurs besoins spécifiques (TAG).
- Intégration facile : fournir une documentation et des exemples de code pour faciliter l'intégration du SDK dans différents sites web.

Spécifications techniques :

- Compatibilité : prise en charge des principaux navigateurs web et des versions récentes, y compris Chrome, Firefox, Safari, etc.
- Intégration avec le framework VueJS.
- Associer un événement à une visite à un visiteur

4 - API de la plateforme

L'API de la plateforme doit pouvoir être accessible pour le BackOffice ainsi que pour le SDK.

Le webmaster doit pouvoir :

- Créer des TAGs (id autogénéré + commentaire)
 - possibilité de modifier uniquement le commentaire
 - suppression impossible (voir soft delete autorisé)
- Créer des tunnels de conversions (tableau ordonné de TAGs + commentaire)
 - possibilité de modifier uniquement le commentaire
 - suppression impossible (voir soft delete autorisé)
- Créer des KPI/Grappe/heatmap (FULL CRUD)
 - spécifier les dimensions de la requête (filtres)
 - spécifier l'échelle de temps (début/fin/step, ex: 00H / 23H59 / 15min) si graphe souhaité
 - type de donnée à afficher (nb absolu ou taux)
 - type de visuel souhaité (KPI/Grappe/Heatmap)
- Créer des events (via SDK uniquement)

5 - Backoffice de la plateforme

Le webmaster doit pouvoir :

- Gérer des TAGs
- Gérer des tunnels de conversions
- Créer un/des dashboards entièrement personnalisable
 - Sélection du type de visuel et ses paramétrages
- Gérer son profil site web

L'admin doit pouvoir :

- Valider les comptes webmasters
- Lister les comptes webmasters
- Agir sur le site comme un webmaster (impersonate)

3 - Spécifications techniques

Plateforme :

- Le backend doit être codé en NodeJS avec le framework Express comme vu en cours
- Le frontend doit être codé en VueJS
- La BDD doit être Postgresql avec l'ORM Sequelize pour tout ce qui touche à l'aspect relationnel
- La BDD de recherche doit être MongoDB (events)
- L'API doit suivre la norme RESTFULL API
- L'authentification utilisateur doit être en JWT
- L'authentification SDK doit utiliser l'APP_ID ainsi que les cors
- L'affichage des stats doit être en temps réel

Site de test :

- Le site doit être en VueJS afin d'intégrer le SDK

SDK :

- Le SDK doit être en VueJS