

Étude

Choix de formats de dates utilisés par la timbreuse

Francis LE ROY

16 juin 2017

Date de réalisation : 2 Mai, 2017

Projet : P1699

Protocole parent : Aucun

1 Introduction

Les formats de dates en informatique sont aussi décisifs que variés. Trois possibilités étaient envisageables : utiliser le système de date EPOCH, utiliser le format normalisé de date ISO8601 [2] sous l'heure UTC, ou bien finalement le format ISO8601 sous l'heure locale avec le décalage horaire inclus dans la date. Les trois ont leurs avantages et inconvénients.

En revanche, nativement, le Javascript utilise le format ISO8601 sous l'heure UTC lors de la création d'un objet `Date`. J'ai décidé d'utiliser ce format.

Pour formater l'heure correctement et faire diverses opérations sur la date et l'heure, j'utilise la bibliothèque Javascript MomentJs [1]

2 Résultats attendus

Si le format de dates adopté est bon, alors les heures affichées sur les différents panels et affichages graphiques devraient être corrects, c'est-à-dire correspondants à l'heure locale actuelle.

3 Procédure de test

Pour tester cette fonctionnalité, il faut par exemple timbrer à l'aide d'une carte correspondant à un élève, puis aller observer, depuis le panel des enseignants, si l'évènement a été enregistré dans les logs de l'élève.

4 Résultats obtenus

Lors de l'utilisation du format ISO8601 sous l'heure UTC, toutes les heures étaient décalées de deux heures (La différence entre l'heure locale et l'heure UTC). Pour la suite, toutes les heures ont été remplacées par des dates formatées sous le format ISO8601 avec le décalage horaire inclus.

5 Conclusion

La conversion entre la bibliothèque `MomentJs` et l'objet `Date` natif au Javascript ne reconvertissait pas en heure locale. Le format ISO8601 avec le décalage horaire est donc la meilleure solution.

Références

[1] JS FOUNDATION. *MomentJs*. Version 2.17.1. 2017. URL : <http://momentjs.com/>.

[2] WIKIPEDIA. *ISO8601*. 1998. URL : https://fr.wikipedia.org/wiki/ISO_8601.