Beapp

Don't Worry, Be Appli *

Exercice d'évaluation Web / PHP

Sujet de l'exercice

Sujet

La société TapAndGo mets à disposition des parcs de vélos pour les usagers dans plusieurs villes de France. Elle souhaite faire développer une application mobile permettant de visualiser **l'emplacement des stations** et le **nombre de vélos** disponibles pour chacune.

Pour cela, vous devrez développer un backoffice fournissant les webservices Rest JSON nécessaires (voir annexe 1) ainsi qu'une page d'administration.

L'administrateur doit pouvoir gérer :

- les villes (nom, position GPS, status activé/désactivé)
- les stations d'une ville (nom, adresse, position GPS, capacité, nombre de vélos disponibles, status activé/désactivé)

L'application doit pouvoir accéder aux webservices détaillés en fin de document.

La société insiste sur le fait que le code et les commentaires devront être rédigés en anglais.

Beapp

Don't Worry, Be Appli *

Exercice d'évaluation Web / PHP

Sujet de l'exercice

Éléments techniques

- Utiliser PHP 7+
- Utiliser **Composer** pour gérer les dépendances
- Utiliser un framework type **Symfony/CakePHP/Slim/...** pour la mise en place de l'application web
- Utiliser un moteur de template type **Twig/...** pour générer les pages webs
- Utiliser **SASS** ou **LESS** pour les feuilles de style
- Utilisation d'une base de données de votre choix (SGBDR classique ou NoSQL)
- Utiliser un ORM pour décrire la base de données et y accéder
- Sécuriser l'accès aux webservices
- Externaliser la configuration
- Utiliser Docker pour fournir en environnement prêt à l'emploi
- Mise en place de tests unitaires (quelques uns pour montrer le principe)
- (Optionnel) Ajouter un système d'authentification des webservices au choix (http basic/digest, OAuth, ...)
- (Optionnel) Sécuriser l'appel aux méthodes des services
- (Optionnel) Utiliser Bower pour les librairies JS
- (Optionnel) Utiliser Gulp pour les builds JS et SASS/LESS

Conseils

Ce projet servira de base de discussion pour l'entretien technique.

Pour votre information, l'évaluation portera principalement sur ces critères :

- Qualité et lisibilité du code
- Respect des bonnes pratiques (isolation de responsabilité, injection de dépendances, normes de nommage, ...)
- Capacité à être force de suggestion
- Pertinence d'utilisation des librairies

Exercice d'évaluation Web / PHP

Sujet de l'exercice

Webservices

Liste des villes

URI	GET /api/cities?page=0&limit=10
Paramètres	**page** : indice de pagination (à utiliser avec limit) **limit** : nombre d'éléments à afficher (à utiliser avec page)

Réponse	Liste d'objets JSON représentant les villes : [
	{
	id: Long,
	name: String,
	latitude: Double,
	longitude: Double
	}
	]

Liste des stations

URI	GET /api/stations/<cityId>?page0&limit=10
Paramètres	**cityId** : id de la ville **page** : indice de pagination (à utiliser avec limit) **limit** : nombre d'éléments à afficher (à utiliser avec page)

Paramètres	Dans le corps de la requête :
	{
	id: Long,
	creationDate: String (format ISO8601),
	lastUpdate: String (format ISO8601),
	name: String,
	address: String,
	description: String,
	latitude: Double,
	longitude: Double,
	bikesCapacity: Int,
	bikesAvailable: Int
	}

Exercice d'évaluation Web / PHP

Sujet de l'exercice

Liste des stations proches

URI	GET /api/stations/near?lat=47.2322086&lng=-1.6134232&radius=10
Paramètres	**lat** : latitude du point de recherche **lng**: longitude du point de recherche **radius**: rayon de recherche (en km)
Réponse	Voir résultat pour la liste des stations

(Optionnel) Déclarer le retrait d'un vélo d'une station

URI	POST /api/stations/<stationId>/take

Réponse	{
	result: Boolean
	}

(Optionnel) Déclarer le dépôt d'un vélo dans une station

URI	POST /api/stations/<stationId>/drop

Réponse	{
	result: Boolean
	}