La société Tamtamers

(Société bretonne de bord de mer)

# Description de la société

Percussionniste depuis plusieurs années déjà, voulant ouvrir leurs activités sur de nouveau horizons, les fondateurs de la société Tamtamers d’ouvrir leur propre boutique de tamtam. Aillant trouvé un entrpot bon marché au bord des côtes bretonne, le nom de leur entreprise est apparut comme une évidence.

La force de leur produit se repose sur un choix de bois et de peaux de qualité et un assemblage à la mains, professionnel. Chaque tamtam est testé par une équipe d'expert pour s'assurer de la qualité du produit. Chaque instrument est également sélectionné en fonction du prix, afin d'offrir toujours au client les tamtam les moins chère du marché.

Même si ce travail et suffisant pour en faire un vendeur de choix dans le monde du tamtam, la société Tamtamers ne pouvait s'arrêter là. Afin de toujours offrir plus à ses clients, la société donne également la possibilité de personnaliser le bois de son tamtam. Cette personnalisation se concrétise par la séléction d’un design dans un catalagogue varié et complet afin de satisfaire tout les goût.

Les valeur de Tamtamers sont :

* qualité
* service
* beurre salé

# Description des services

## Choix d’implémentation

Dans le but d'ouvrir son marché au-delà des côtes bretonnes, Tamtamers à décider de suivre les tendances du marché et de s’adapter aux nouvelles technologies. Ainsi la société à décider de mettre en place plusieurs service afin d’ouvrir ses ventes au travers d’internet et des différent marché en ligne.

L’implémentation et les connecteurs de nos services sont décrits dans la [JavaDoc](javadoc/index.html).

Nous allons expliquer et défendre nos services et leur implémentation.

Tout dabord, il à falu s’intéroger sur quel implémentation choisir. En cherchant les différence entr les deux service, nous avons vu que REST été dans la majeur partie des cas une meilleure solution que SOAP. Notament du fais de ses meilleures performance et une meilleure scalabilité. L’utilisation du protocole http ajoute au côté simple de REST. SOAP semble pertinant dans des cas particulier de service sécurisé qui necessite une conformité ACID.

Ainsi du fait de sa simplicité et de son adaptabilité il à été préféré REST à ACID pour l’implémentation des services dans ce projet.

Plusieurs service on en suite été implémenté. Plusieur (objet/acteur/element) on été identifié necessitant chacun une base de donnée, un objet le représentant et un service pour le manipulé.  
Ces « truc » ont été séparé ainsi :

## Tamtam

Ce service gère la définition des différents tamtams. La base de donnée enregistre un tamtam avec un id, un boi, une peau, une marque, une description, une image, un price, une liste de livreur disponible, et une liste de décorations dipsonible. Le service offre la possibilité de lister tout les tamtam enregistré, de chercher un tamtam par id, de récupérer toutes les caractéristique d’un tamtam, de chercher un tamtam sur les critères le définissant (marque, peau, bois).

Le but de ce service est de d’afficher et de gérer le catalogue des tamtam. Grâc aux fonction de rechercher et de récupération de caractéristique, le ti et la recherche de tamtam est facilement gérable.

## Stock

Ce service gère les stock effectif d’un entrepot. La base de donné enregistre les stock en item. Ces item ont un id, un nombre d’element restant en stock, le réaprovisionnement. Le service offre la possibilité de gérer le stock en ajoutant ou retirant des élément à un item du stock, en ajoutant ou retirant des item du stock, en listant les items du stock, rechercher un item dans le stock.

## Commande

Ce service gère la réception des commande. Une commande est composé d’un id, une liste de tamtam, une liste de décoration correspondante, une livraison associé, un total, un client et un status. Le status correspond à l’étape dans le processus d’envoi de la commande. Le service offre comme fonctionnalité l’obtention de la liste des commandes, de chercher une commande par id ou par status.

## Client

Ce service gère les client de la société. Un client est identifé par son id, un nom, une adresse, et un numéro de téléphone. Le service offre la possibilité de lister tout les client de la base de donné, d’ajouter un client, de chercher un client par id, de supprimer un client, de créer une commande et de récupérer les commandes de cette utilisateur.

## Paiement

Ce service gère la validation du paiement d’un client. Un paiement est lié à une commande par son id, le prix à payé, le type de transation, le status de la transaction, le type de transaction. Le service permet de faire le règlement par carte ou par Paypal.

## Livraison

Ce service manipule les différent type de livraison que peuvent être proposé aux clients. Un livraison et identifé par un id, un nom, un délai, un prix et une disponibilité. Le service permet de récupérer le meilleur fournisseur en liraison expresse et celui en livraison standar.

**Information relative aux tamtams**

**GET**  
*/tamtams*:   
 Retourne la liste de tous les tamtams dans la base de donnée.

**GET**  
*/tamtams/:id(/)* :

Paramètre :

* (int) id : id correspondant à un tamtam.

Retourne le détail d'un tamtam.

**GET**  
*/tamtams/types(/)* :   
 Retourne les éléments caractérisant les tamtams. (Marque, peau, bois).

**GET**  
*/tamtams/search(/brand/:brand)(/wood/:wood)(/skin:/skin)(/)* :

Paramètre :

* (int) brand : (optional) id d’une marque
* (int) wood : (optional) id d’un bois
* (int) skin : (optional) id d’une peau

Retourne la liste filtrée des tamtams selon les particularités souhaitées.

**Informations sur les commandes**

**GET**  
*/orders* :  
 Retourne la liste de toutes les commandes de la base de donnée.

**GET**  
*/orders/:id(/)* :

Paramètre :

* (int) id :

Retourne une commande.

**POST**  
*/orders/tamtam/:tamtam/*s*hipment/:* s*hipment(/decoration/:decoration)/user/ :user(/)* :

Paramètre :

* (int) tamtam :
* (int) shipment :
* (int) decoration : (optional)
* (int) user :

Créé une nouvelle commande.

**Informations sur les utilisateurs**

**GET**  
*/users* :  
 Retourne la liste de tous les utilisateurs de la base de donnée.

**POST**  
*/orders/fullname/:fullname/phone/:phone/*adress1/:adress1/adress2/:adress2/zip-code/:zip-code/state/:state*(/) :*

Paramètre :

* (string) fullname : Nom de l’utilisateur
* (int) phone : numéro de téléphone
* (string) adress1 : adresse de l’utilisateur
* (string) adress2 : adresse secondaire
* (int) zip-code : code postal
* (string) country : pays de l’utilisateur

Crée un nouvel utilisateur dans la base de données.

**GET**  
*/users/ :id(/)* :

Paramètre :

* (string) fullname : Nom de l’utilisateur

Retourne un utilisateur.

**DELETE**  
*/users/ :id(/)* :

Paramètre :

* (string) fullname : Nom de l’utilisateur

Supprime un utilisateur de la base de donnée.

**PUT**  
*/users/ :id(/)* :

Paramètre :

* (string) fullname : Nom de l’utilisateur
* (int) phone : numéro de téléphone
* (string) adress1 : adresse de l’utilisateur
* (string) adress2 : adresse secondaire
* (int) zip-code : code postal
* (string) country : pays de l’utilisateur

Mise à jour d’un utilisateur dans la base de donnée.