La société Tamtamers

(Société bretonne de bord de mer)

# Description de la société

Percussionniste depuis plusieurs années déjà, voulant ouvrir leurs activités sur de nouveaux horizons, les fondateurs de la société Tamtamers ouvrent leur propre boutique de tamtam. Ayant trouvé un entrepôt bon marché au bord des côtes bretonne, le nom de leur entreprise est apparu 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 main. 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 tamtams les moins chers 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 sélection d’un design dans un catalogue varié et complet afin de satisfaire tous les goûts.

Les valeurs de Tamtamers sont :

* La qualité
* Le service
* Le beurre salé

# 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 d’abord, il a fallu s’interroger sur quelle implémentation choisir. En cherchant les différences entre les deux services, nous avons vu que REST était dans la majeure partie des cas une meilleure solution que SOAP. Notamment du fait de ses meilleures performances et une meilleure évolutivité. L’utilisation du protocole ajoute au côté simple de REST. SOAP semble pertinent dans des cas particulier de service sécurisé qui nécessite une conformité ACID.

Ainsi du fait de sa simplicité et de son adaptabilité nous avons préféré REST à SOAP pour l’implémentation des services dans ce projet.

Nous avons implémenté plusieurs services. Nous avons fait le choix de les isoler car ils étaient faiblement couplés entre eux.

# Description des services

## Tamtamers

Ce service permet de gérer la partie « utile » à l’utilisateur : il permet de consulter la liste des tamtams avec leurs différentes décorations, mais également de gérer son profil et créer une commande.

Pour plus de clarté dans le catalogue, nous avons ajouté un système de filtre dans la liste des tamtams.

==== On garde ? ===

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 cette manière, il est possible de retirer un catalogue complet et détaillé de tout les tamtams. Afin de facilité la recherche de tamtam spécifique, il est possible de récupérer dans un premier temps toutes les caractéristique possible d’un tamtam. Il est rendu le nom des famille de caractéristique avec leur éléments (ex : bois : acajou, pommier, …). Deux mode de recherche sont disponible, une par id pour obtenir un tamtam précis, et une par caractéristique pour affiner le catalogue par critère.

## Stock

Ce service est connecté directement à l’entrepôt et permet d’informer sur l’état du stock pour chaque article.

Il est également possible de connaitre, pour un article souhaité, sont délai de réapprovisionnement.

## Banque

Ce service est directement connecté à la banque de la société, et permet de payer une commande enregistrée.

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. Ce service est orrité vers une automatisation de la gestion des stock. Ainsi il est possible de manipuler les différent produit du stock en ajoutant de nouveau produit ou en supprimant d’autres. Il est possible sur ces produit de modifier leur nombre en stock en fonction des réaprovisionnement et des commandes. Il est aussi possible d’obtenir la liste de tout les élément du stock et de leur nombre ou bien de chercher un seul élément dans le stock en fonction de son id.

## 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. Ce service est orienté pour ceux qui gère les stock et l’envoit des colis. Il permet d’obtenir la liste complète des commande ou bien de les filtré par status et ainsi savoir quel command il reste à envoyer. Le service permet en suite de faire avancer le status d’une commande.

## 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. Ce service permet de gérer un historique des client et d’associer une commande à un client. Nous pouvons lister tout les clients, en ajouter dans la base de donnée, chercher un utilisateur précis ou en supprimer un. L’association d’une commande à un client ce fait au travers de ce service. Les coordonné d’un client se mette à jours avec celles de la dernière commande, dans le but de proposer une adresse par défaut correspondant à la dernière utilisé. Il est possible également de changer les information d’un client, indépendament des commande, au travers de ce service.

## 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. Ce service permet de gérer les mode de livraison disponible et de proposer le meilleur lors de la commande. Le service donne le meilleur mode de livraison express (rapide mais couteux) et le meilleur mode de livraison normal (plus lent mais peut être gratuit).

# Documentation du Web Service

## Service Tamtamers

Adresse mère : <http://localhost:8181/cxf/tamtamers/>

### Tamtams

**Liste des tamtams (filtrables par marque, bois et peau)**

**GET**  
 /tamtams

* + ***QUERY***(int) brand : Identifiant de la marque souhaitée.
  + ***QUERY*** (int) wood : Identifiant du bois souhaité.
  + ***QUERY*** (int) skin : Identifiant de la peau souhaitée.

**Détail d’un tamtam**

**GET**

/tamtams/{id}

* + ***ROUTE*** (int) id : Identifiant d’un tamtam.

**Types de tamtams**

**GET**

/tamtams/types

**Ajout d’un tamtam**

**POST**

/tamtams

* + ***QUERY*** (string) name : Nom du produit.
  + ***QUERY*** (string) description : Description du produit.
  + ***QUERY*** (string) image : URL vers l’image du produit.
  + ***QUERY*** (string) brand : Marque du produit.
  + ***QUERY*** (string) wood : Bois du produit.
  + ***QUERY*** (string) skin : Peau du produit.
  + ***QUERY*** (double) price : Prix du produit.
  + ***HEADER*** (JSON) decorations : Décorations disponibles pour ce produit.
  + ***HEADER*** (JSON) shipments : Modes de livraison disponible pour ce produit.

Syntaxe des décorations et des modes de livraison.

[1, 2, 3]

**Modification d’un tamtam**

**PUT**

/tamtams/{id}

* + ***ROUTE*** (int) id : Identifiant du tamtam
  + ***QUERY*** (string) name : Nom du produit.
  + ***QUERY*** (string) description : Description du produit.
  + ***QUERY*** (string) image : URL vers l’image du produit.
  + ***QUERY*** (string) brand : Marque du produit.
  + ***QUERY*** (string) wood : Bois du produit.
  + ***QUERY*** (string) skin : Peau du produit.
  + ***QUERY*** (double) price : Prix du produit.
  + ***HEADER*** (JSON) decorations : Décorations disponibles pour ce produit.
  + ***HEADER*** (JSON) shipments : Modes de livraison disponible pour ce produit.

Syntaxe des décorations et des modes de livraison.

[1, 2, 3]

### Commandes

**Liste des commandes (filtrable par statut)**

**GET**  
 /orders

* + ***QUERY***(string) status: Nom du statut recherché.

**Détail d’une commande**

**GET**

/orders/{id}

* + ***ROUTE*** (int) id : Identifiant d’une commande.

// TODO : Validations

### Clients

**Liste des clients (filtrables par noms)**

**GET**  
 /customers

* + ***QUERY***(string) name : Nom de la personne recherchée.

**Détail d’un client**

**GET**

/customers/{id}

* + ***ROUTE*** (int) id : Identifiant d’un client.

**Modification d’un client**

**PUT**

/customers/{id}

* + ***ROUTE*** (int) id : Identifiant d’un client.
  + ***QUERY*** (string) fullname : Nom et prénom du client.
  + ***QUERY*** (string) phone : Numéro de téléphone du client.
  + ***QUERY*** (string) address1 : Adresse du client.
  + ***QUERY*** (string) address2 : Complément d’adresse.
  + ***QUERY*** (string) zip-code : Code postal de son domicile.
  + ***QUERY*** (string) state : Pays du domicile.

**Création d’un client**

**POST**

/customers/

* + ***QUERY*** (string) fullname : Nom et prénom du client.
  + ***QUERY*** (string) phone : Numéro de téléphone du client.
  + ***QUERY*** (string) address1 : Adresse du client.
  + ***QUERY*** (string) address2 : Complément d’adresse.
  + ***QUERY*** (string) zip-code : Code postal de son domicile.
  + ***QUERY*** (string) state : Pays du domicile.

**Commandes d’un client**

**GET**

/customers/{id}/orders

* + ***ROUTE*** (int) id : Identifiant d’un client.

**Création d’une commande**

**POST**

/customers/{id}/orders

* + ***ROUTE*** (int) id : Identifiant d’un client.
  + ***HEADER***(JSON) order : Commande formatée en JSON.

{  
 "items":  
 [  
 {  
 "tamtam": 1,  
 "decoration": 2  
 },  
 {  
 "tamtam": 2  
 }  
 ],  
 "shipment": 1  
}

## Service Banque

Adresse mère : <http://localhost:8181/cxf/bank/>

### Paiement

**Paiement d’une commande en CB**

**PUT**

/payment/

* + ***QUERY*** (string) cardnumber : Numéro de la carte bancaire.
  + ***QUERY*** (string) owner : Propriétaire de la carte bancaire.
  + ***QUERY*** (string) verification-code : Code de vérification de la carte bancaire.
  + ***QUERY*** (int) order : Identifiant de la commande.
  + ***QUERY*** (double) amount : Montant total de la commande.

**Paiement d’une commande avec PayPal**

**PUT**

/payment/

* + ***QUERY*** (string) account : Numéro de compte de paypal.
  + ***QUERY*** (int) order : Identifiant de la commande.

## **Service Entrepot**

Adresse mère : <http://localhost:8181/cxf/entrepot/>

### Stock

**Liste des articles en stock**

**GET**  
 /stock

**Détail d’un article en stock**

**GET**

/ stock/{id}

* + ***ROUTE*** (int) id : Identifiant d’un article.

### **Livraison**

**Liste des modes de livraison (filtrable par type)**

**GET**  
 /shipments

* + ***QUERY*** (string) type : Type de livraison (express / normal)

**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.