Documentation technique

Demonetik WebService

Table des matières

Prérequis 1

Fonction d’accès 1

Fonction initialisation 1

Fonction montant 1

Fonction infos porteur 2

Fonction demande d’autorisation 2

Fonction fin de transaction 2

Fonction get transaction 2

Fonction get etat 2

Fonction get montant 2

Fonction reset transaction 3

Machine à état 3

Index Etat 3

Description états 3

WebSocket 4

# Prérequis

Serveur : Tomcat 8

# Fonction d’accès

## Fonction initialisation

Initialise la transaction, passage à l’état : EtatInit

Type Requête : **GET**

URL : *localhost:8080/DemonetikWebService/demonetik/transaction/inittransaction*

Paramètre : Aucun

Retour :

## Fonction montant

Définis le montant de la transaction, passage à l’état : EtatMontant

Type Requête : **POST**

URL : *localhost:8080/DemonetikWebService/demonetik/transaction/montant*

Paramètre : **int** montant

Retour :

## Fonction infos porteur

Définis le porteur de la transaction, passage à l’état : EtatPorteurIdent

Type Requête : **POST**

URL : *localhost:8080/DemonetikWebService/demonetik/transaction/infoporteur*

Paramètre : **String** nom, **String** prenom, **String** numCarte

Retour :

## Fonction demande d’autorisation

Effectue une demande d’autorisation, passage à l’état : EtatDemandeAuto

Type Requête : **POST**

URL : *localhost:8080/DemonetikWebService/demonetik/transaction/demandeauto*

Paramètre : **int** pin

Retour :

## Fonction demande d’autorisation resultat

Renvoi le résultat de la demande d’autorisation, passage à l’état : EtatDemandeAutoRes

Type Requête : **GET**

URL : *localhost:8080/DemonetikWebService/demonetik/transaction/demandeautoresultat*

Paramètre :

Retour : int le resultat de la demande d’autorisation (1 = Ok, 0 = Ko)

## Fonction fin de transaction

Termine la transaction en cours, passage à l’état : EtatFinTransaction

Type Requête : **GET**

URL : *localhost:8080/DemonetikWebService/demonetik/transaction/endtransaction*

Paramètre : **aucun**

Retour :

## Fonction get transaction

Fonction de récupération de la transaction en cours.

Type Requête : **GET**

URL : *localhost:8080/DemonetikWebService/demonetik/transaction/gettransaction*

Paramètre : **aucun**

Retour : La transaction au format JSON

## Fonction get etat

Fonction de récupération de l’état de la transaction en cours.

Type Requête : **GET**

URL : *localhost:8080/DemonetikWebService/demonetik/transaction/getetat*

Paramètre : **aucun**

Retour : Le label de l’état en cours

## Fonction get montant

Fonction de récupération de du montant de la transaction en cours

Type Requête : **GET**

URL : *localhost:8080/DemonetikWebService/demonetik/transaction/getmontant*

Paramètre : **Aucun**

Retour : Le montant de la transaction

## Fonction reset transaction

Réinitialise la transaction, passage à l’état : EtatStart

Type Requête : **GET**

URL : *localhost:8080/DemonetikWebService/demonetik/transaction/resettransaction*

Paramètre : **Aucun**

Retour : « Transaction réinitialisée »

# Machine à état

Un pattern State a été utilisé pour représenter les différents états d’une transaction (simplifié).

## Index Etat

|  |  |  |
| --- | --- | --- |
| Etat | Id | Type |
| EtatStart | 0 | Etat de départ |
| EtatInit | 1 | TPE |
| EtatMontant | 2 | TPE |
| EtatPorteurIdent | 3 | Carte |
| EtatDemandeAutoAsk | 4 | TPE |
| EtatDemandeAutoProcess | 5 | Banque |
| EtatDemandeAutoRes | 6 | Banque |
| EtatFinTransaction | 7 | TPE |

## Description états

|  |  |  |
| --- | --- | --- |
| Etat | Description | Actions |
| EtatStart | Etat de départ d’une transaction | NULL |
| EtatInit | La transaction est initialisé par le TPE |  |
| EtatMontant | Le montant a été transmis au web service |  |
| EtatPorteurIdent | Les données porteur ont été transmises au webservice |  |
| EtatDemandeAutoAsk | Une demande d’autorisation a été reçu |  |
| EtatDemandeAutoProcess | Une demande d’autorisation est en cours de traitement |  |
| EtatDemandeAutoRes | Une demande d’autorisation a été traitée |  |
| EtatFinTransaction | La transaction est terminée |  |

Pour paramétrer le label des états (ce qui sera affiché par le client Web), il faut modifier le fichier stringEtat.properties du WebService.

# WebSocket

Un WebSocket est implémenté avec le WebService, celui-ci sert pour la communication avec le client Web. Ainsi on a une communication full duplex en temps réel. Ceci dans le but d’éviter les clients à faire des requêtes en boucle pour avoir l’information.

Le WebService envoi aux clients connecté au WebSocket une mise à jour à chaque changement d’état de la transaction.