CONFIDENTIEL

Documentation API
IMMS (java) - SOFTELELR

Version 1.0 du Jeudi 30 mai 2019

Historique

Version	Date	Origine de la mise à jour	Rédigée par	Validée par
1.0	30/05/2019	Création du document API version JAVA	Raoul Bille	

Sommaire

1. Introduction

L'API Mobile money lié à l'application SOFTELLER est un service de l'entreprise IWOMI TECHNOLOGIES qui offre un moyen sûr, efficace et sécurisé d'effectuer toutes les transactions MTN MoMo, et Orange Money vers tous les abonnés MTN et Orange.

2. Fonctionnement

2.1. Etapes d'exécution des transactions

Avant de pouvoir accéder à cette API sur notre plateforme, vous devez disposer d'un compte SOFTELLER. Celui-ci contient entre autre vos informations d'identification et d'authentification qui sont en effet un Login et un Mot de passe permettant de vous identifier et de vous donner accès à nos multiples services.

L'exécution des transactions MoMo via notre plateforme se décline principalement en 3 étapes :

1. Identification et authentification du client.
2. Envoie des données (numéros de téléphone[1], montant, type[2] et informations[3]) à transmettre.
3. Réception des résultats.

2.2. Appels des méthodes

[1] Chaque numéro de téléphone doit être soumis en respectant le format international décrit par la norme E.164
[2] Motif de la transaction à effectuer a Mobile Money
[3] Informations sur le pays, la ville et le nom du Client
MoMo :Mobile Money

Notre API MTN MoMo offre plusieurs méthodes (fonctions) de traitement de vos requêtes en fonction du service voulu et après le processus d'authentification.

Ces méthodes sont directement appelées via l'Url avec les paramètres de la requête.

Note :

- Format de données supporté pour la requête et les résultats : JSON
- Url de base pour tous les services de notre API : https://www.softeller.com/api_softeller/request_payment

Dans les exemples suivants, nous allons rédiger un client en java en utilisant sa méthode POST.

2.2.1 Préparation des données pour l'exécution d'une transaction Mobile Money

```
String userid = '1';          // votre user id fourni par softeler
String login = 'login';  // login du compte softeller fourni par IWOMI
String password = 'pass';  // fourni par iwomi
String country = 'Cameroun';   // pays ou le client émet la transaction
String town = 'Douala';     // ville our le client ...
String amount = '5';      // montant a débité au client (cest le montant exact qui sera débité au client (un entier)
String first_name = 'Softeller';     // prenom du client
String last_name = 'Soft';     // nom du client
String motif = 'Achat telephone';  // motif de la transaction
String phone = "237683559434";      // numéro du client qui sera débité
String currency = 'XAF';      // devise toujours XAF cest la seule devése traité pour le moment
```

```
String type = 'momo';                // type de transaction ('momo',

String email = 'info@softeller.com';    // juste pour les transactions
visa/mastercard
```

N	Paramètre	Type	Requis	Commentaires
01	Userid	Chaine de caractère (varchar)	oui	User id fourni par softeller
02	Login	Chaine de caractère (varchar)	oui	Login fourni par softeller
03	password	Chaine de caractère (varchar)	oui	Password fourni par softeller
04	Country	Chaine de caractère (varchar)	oui	Pays du client qui effectue la transaction. Pour l'instant toujours mettre Cameroun
05	Town	Chaine de caractère (varchar)	oui	Ville du client qui effectue la transaction.
07	Amount	Entier (int)	Oui	Montant à débiter chez le client doit etre de 100 FCFA minimum
08	First name	Chaine de caractère (varchar)	oui	Prénom du client
09	Last name	Chaine de caractère (varchar)	oui	Nom du client
10	motif	Chaine de caractère (varchar)	Oui	Motif de la transaction (exemple : paiement de salaire)
11	Phone	Chaine de caractère (varchar)	Oui	Numéro du client à débité au format international. (exemple : 23783559434)

				Note : pas de +, ni de 00
12	Currency	Chaine de caractère (varchar)	Oui	Devise pour l'instant toujours mettre le « XAF »
13	Type	Chaine de caractère (varchar)	Oui	Type de transaction. Pour l'instant « momo » pour MTN Mobile Money
14	Email	Chaine de caractère (varchar)	Oui/non	Le champ email est requis pour les paiements par carte de type Visa ou MasterCard uniquement

2.2.2 Envoi de la requête

Une fois les données de la requête préparées, vous pouvez donc utiliser la méthode POST pour effectuer les requêtes

```
String salt = md5(userid);

String pwd = "softeller2";

String cc_string  = salt+pwd;

String pass = Encrypt256(cc_string);

String                                  data                                    =
"{\"Login\":\""+login+"\",\"Password\":\""+pass+"\",\"Country\":\""+countr
y+"\",\"Town\":\""+town+"\""

                    +
",\"Phone\":\""+phone+"\",\"First_name\":\""+first_name+"\",\"Last_name\":
\""+last_name+"\",\"Motif\":\""+motif+"\""

                    +
",\"Amount\":\""+amount+"\",\"Type\":\""+type+"\",\"Currency\":\""+currenc
y+"\",\"Email\":\""+email+"\"}";

URL url = new URL(url1);

Map<String,Object> params = new LinkedHashMap<>();

params.put("key", data); // All parameters, also easy

StringBuilder postData = new StringBuilder();

        // POST as urlencoded is basically key-value pairs, as with GET
```

```
        // This creates key=value&key=value&... pairs
for (Map.Entry<String,Object> param : params.entrySet()) {
            if (postData.length() != 0) postData.append('&');
            postData.append(URLEncoder.encode(param.getKey(), "UTF-8"));
            postData.append('=');

postData.append(URLEncoder.encode(String.valueOf(param.getValue()),   "UTF-
8"));
        }


        // Convert string to byte array, as it should be sent
byte[] postDataBytes = postData.toString().getBytes("UTF-8");


        // Connect, easy
        HttpURLConnection conn = (HttpURLConnection)url.openConnection();
        // Tell server that this is POST and in which format is the data
        conn.setRequestMethod("POST");
        conn.setRequestProperty("Content-Type",    "application/x-www-form-
urlencoded");
        conn.setRequestProperty("Content-Length",
String.valueOf(postDataBytes.length));
        conn.setDoOutput(true);
        conn.getOutputStream().write(postDataBytes);


        // This gets the output from your server
        Reader          in          =          new          BufferedReader(new
InputStreamReader(conn.getInputStream(), "UTF-8"));


        for (int c; (c = in.read()) >= 0;)
            System.out.print((char)c);
```

2.2.3 Reponses du Serveur

En fonction des paramètres envoyés (valides ou non) au serveur et du statut de traitement ou d'envoi de vos requêtes, plusieurs types de réponses peuvent être retournés par le serveur.

Les réponses du serveurs sont au format JSON avec les parametres suivant :

{

'status' : '1000',

'message' : 'Pending',

'TransactionCode' : 'SOFT_3OA4GCU6I14809288867863',

'redirectUrl':'https:\/\/ucollect.ubagroup.com\/cipg-payportal\/paytran?id=CTNT348589'

}

N	Parametre	Type	commentaire
01	Status	Chaine de caractère (varchar)	C'est le code du statut de la trasaction
02	Message	Chaine de caractere (varchar)	Message de retour
03	TransactionCode	Chaine de caractère (varchar)	Code de la transaction
04	redirectUrl	Chaine de caractère (varchar)	Page de redirection pour les paiement par carte

Le tableau suivant récapitule les cas possibles :

01	Success
100	Fail
1000	Pending
320	No credit
102	Auth faild
103	Syntax error

400	Imporssible de trouver la transaction

2.3. APPELS DE LA METHODES DE CHECK STATUS

Cette méthode est utilisée pour avoir le statut d'une transaction a partir du code de la transaction.

Elle est appellée de la meme manière que la méthodede paiement les seuls changement sont :

- L'URL : https://www.softeller.com/api_softeller/check_status
    - Les parametres :
    - Userid
    - Login
    - Password
- transactionCode

```
    String userid = '3';

    String login = 'softeller';

    String password = 'softeller2';  // fourni par iwomi

    String TransactionCode = 'SOFT_3HJJTBVVV14805899547197';


String url = 'http://www.softeller.com/api_softeller_momo/check_status';
    String salt = md5(userid);
     String pwd = "softeller2";
     String cc_string  = salt+pwd;
     String pass = Encrypt256(cc_string);
```

```
    String                                  data                                    =
"{\"Login\":\""+login+"\",\"Password\":\""+pass+"\",\"'TransactionCode'
\":\""+ TransactionCode +"\"}";

URL url = new URL(url1);

Map<String,Object> params = new LinkedHashMap<>();

params.put("key", data); // All parameters, also easy

StringBuilder postData = new StringBuilder();

for (Map.Entry<String,Object> param : params.entrySet()) {

    if (postData.length() != 0) postData.append('&');

    postData.append(URLEncoder.encode(param.getKey(), "UTF-8"));

    postData.append('=');

    postData.append(URLEncoder.encode(String.valueOf(param.getValue()),
"UTF-8"));

}// Convert string to byte array, as it should be sent

byte[] postDataBytes = postData.toString().getBytes("UTF-8");

// Connect, easy

HttpURLConnection conn = (HttpURLConnection)url.openConnection();

// Tell server that this is POST and in which format is the data

conn.setRequestMethod("POST");

conn.setRequestProperty("Content-Type",           "application/x-www-form-
urlencoded");

conn.setRequestProperty("Content-Length",
String.valueOf(postDataBytes.length));

conn.setDoOutput(true);

conn.getOutputStream().write(postDataBytes);

  // This gets the output from your server
```

```
Reader in = new BufferedReader(new InputStreamReader(conn.getInputStream(),
"UTF-8"));

for (int c; (c = in.read()) >= 0;)

 System.out.print((char)c);
```

2.4. APPELS DE LA METHODES DE MAKE DEPOSIT

Cette méthode est utilisée pour faire un dépôt dans un compte mobile money, ce dépôt débite votre compte partenaire chez softeller et crédite le compte du client final.

- L'URL : https://www.softeller.com/api_softeller/make_deposit

2.4.1. Préparation des données pour l'exécution d'une transaction

```
String userid = '1';          // votre user id fourni par softeler

String login = 'login';  // login du compte softeller fourni par IWOMI

String password = 'pass';  // fourni par iwomi

String country = 'Cameroun';   // pays ou le client émet la transaction

String town = 'Douala';     // ville our le client ...

String amount = '5';      // montant a débité au client (cest le montant
exact qui sera débité au client (un entier)

String first_name = 'Softeller';     // prenom du client

String last_name = 'Soft';     // nom du client
```

```
String motif = 'Achat telephone';  // motif de la transaction

Striphone = "237683559434";      // numéro du client qui sera débité

$currency = 'XAF';      // devise toujours XAF cest la seule devése traité
pour le moment

$type = 'momo';              // type de transaction ('momo',
```

N	Paramètre	Type	Requis	Commentaires
01	Userid	Entier (Int)	oui	User id fourni par softeller
02	Login	Chaine de caractère (varchar)	oui	Login fourni par softeller
03	password	Chaine de caractère (varchar)	oui	Password fourni par softeller
04	Country	Chaine de caractère (varchar)	oui	Pays du client qui effectue la transaction. Pour l'instant toujours mettre Cameroun
05	Town	Chaine de caractère (varchar)	oui	Ville du client qui effectue la transaction.
07	Amount	Entier (int)	Oui	Montant à débiter chez le client doit etre de 100 FCFA minimum
08	First name	Chaine de caractère (varchar)	oui	Prénom du client
09	Last name	Chaine de caractère (varchar)	oui	Nom du client
10	motif	Chaine de caractère (varchar)	Oui	Motif de la transaction (exemple : paiement de salaire)

11	Phone	Chaine de caractère (varchar)	Oui	Numéro du client à débité au format international. (exemple : 23783559434) Note : pas de +, ni de 00
12	Currency	Chaine de caractère (varchar)	Oui	Devise pour l'instant toujours mettre le « XAF »
13	Type	Chaine de caractère (varchar)	Oui	Type de transaction. Pour l'instant « momo » pour MTN Mobile Money

2.4.2 Envoi de la requête

Une fois les données de la requête préparées, vous pouvez donc utiliser la méthode CURL pour effectuer les requêtes

```
String salt = md5(userid);

String pwd = "softeller2";

String cc_string  = salt+pwd;

String pass = Encrypt256(cc_string);

String                                  data                                    =
"{\"Login\":\""+login+"\",\"Password\":\""+pass+"\",\"Country\":\""+countr
y+"\",\"Town\":\""+town+"\""

                    +
",\"Phone\":\""+phone+"\",\"First_name\":\""+first_name+"\",\"Last_name\":
\""+last_name+"\",\"Motif\":\""+motif+"\""

                    +
",\"Amount\":\""+amount+"\",\"Type\":\""+type+"\",\"Currency\":\""+currenc
y+"\"}";

URL url = new URL(url1);

Map<String,Object> params = new LinkedHashMap<>();

params.put("key", data); // All parameters, also easy

StringBuilder postData = new StringBuilder();

        // POST as urlencoded is basically key-value pairs, as with GET
```

```
        // This creates key=value&key=value&... pairs
for (Map.Entry<String,Object> param : params.entrySet()) {
            if (postData.length() != 0) postData.append('&');
            postData.append(URLEncoder.encode(param.getKey(), "UTF-8"));
            postData.append('=');

postData.append(URLEncoder.encode(String.valueOf(param.getValue()),   "UTF-
8"));
        }


        // Convert string to byte array, as it should be sent
byte[] postDataBytes = postData.toString().getBytes("UTF-8");


        // Connect, easy
        HttpURLConnection conn = (HttpURLConnection)url.openConnection();
        // Tell server that this is POST and in which format is the data
        conn.setRequestMethod("POST");
        conn.setRequestProperty("Content-Type",    "application/x-www-form-
urlencoded");
        conn.setRequestProperty("Content-Length",
String.valueOf(postDataBytes.length));
        conn.setDoOutput(true);
        conn.getOutputStream().write(postDataBytes);


        // This gets the output from your server
        Reader          in          =          new          BufferedReader(new
InputStreamReader(conn.getInputStream(), "UTF-8"));


        for (int c; (c = in.read()) >= 0;)
            System.out.print((char)c);
```

2.4.3 Reponses du Serveur

En fonction des paramètres envoyés (valides ou non) au serveur et du statut de traitement ou d'envoi de vos requêtes, plusieurs types de réponses peuvent être retournés par le serveur.

Les réponses du serveurs sont au format JSON avec les parametres suivant :

{

'status' : '1000',

'message' : 'Pending',

'TransactionCode' : 'SOFT_3OA4GCU6I14809288867863'

}

N	Parametre	Type	commentaire
01	Status	Chaine de caractère (varchar)	C'est le code du statut de la trasaction
02	Message	Chaine de caractere (varchar)	Message de retour
03	TransactionCode	Chaine de caractère (varchar)	Code de la transaction

3. Code exemple complet

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.Reader;
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLEncoder;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.LinkedHashMap;
import java.util.Map;

public class Test {

    public static void main(String[] args) throws UnsupportedEncodingException, MalformedURLException, IOException {

        String url1 = "https://www.softeller.com/api_softeller/request_payment/";
        String userid = "3";
        String login = "softeller";
        String country = "CAMEROUN";
        String town = "Douala";
        String amount = "5";
        String first_name = "Softeller";
        String last_name = "Soft";
        String motif = "Test";
        String phone = "237683559434";
        String currency = "xaf";
        String type = "card";
        String email = "info@softeller.com";
        String salt = md5(userid);
        String pwd = "softeller2";
        String cc_string  = salt+pwd;
        String pass = Encrypt256(cc_string);
        String data =  "{\"Login\":\""+login+"\",\"Password\":\""+pass+"\",\"Country\":\""+country+"\",\"Town\":\""+town+"\""
                +
"\"Phone\":\""+phone+"\",\"First_name\":\""+first_name+"\",\"Last_name\":\""+last_name+"\",\"Motif\":\""+motif+"\""
                + ",\"Amount\":\""+amount+"\",\"Type\":\""+type+"\",\"Currency\":\""+currency+"\",\"Email\":\""+email+"\"}";
```

```
URL url = new URL(url1);

    Map<String,Object> params = new LinkedHashMap<>();
    params.put("key", data); // All parameters, also easy



    StringBuilder postData = new StringBuilder();
    // POST as urlencoded is basically key-value pairs, as with GET
    // This creates key=value&key=value&... pairs
    for (Map.Entry<String,Object> param : params.entrySet()) {
       if (postData.length() != 0) postData.append('&');
       postData.append(URLEncoder.encode(param.getKey(), "UTF-8"));
       postData.append('=');
       postData.append(URLEncoder.encode(String.valueOf(param.getValue()), "UTF-8"));
    }

    // Convert string to byte array, as it should be sent
    byte[] postDataBytes = postData.toString().getBytes("UTF-8");

    // Connect, easy
    HttpURLConnection conn = (HttpURLConnection)url.openConnection();
    // Tell server that this is POST and in which format is the data
    conn.setRequestMethod("POST");
    conn.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
    conn.setRequestProperty("Content-Length", String.valueOf(postDataBytes.length));
    conn.setDoOutput(true);
    conn.getOutputStream().write(postDataBytes);

    // This gets the output from your server
    Reader in = new BufferedReader(new InputStreamReader(conn.getInputStream(), "UTF-8"));

    for (int c; (c = in.read()) >= 0;)
       System.out.print((char)c);

  }

  public static String md5(String s) {
    final String MD5 = "MD5";
    try {
       // Create MD5 Hash
       MessageDigest digest = java.security.MessageDigest
             .getInstance(MD5);
       digest.update(s.getBytes());
```

```
byte messageDigest[] = digest.digest();
            // Create Hex String
            StringBuilder hexString = new StringBuilder();
            for (byte aMessageDigest : messageDigest) {
                String h = Integer.toHexString(0xFF & aMessageDigest);
                while (h.length() < 2)
                    h = "0" + h;
                hexString.append(h);
            }
            return hexString.toString();

        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        return "";
    }
    public static String Encrypt256(String strSrc) {
        MessageDigest md = null;
        String strDes = null;
        byte[] bt = strSrc.getBytes();
        try {
            md = MessageDigest.getInstance("sha-256");
            md.update(bt);
            strDes = bytes2Hex(md.digest()); // to HexString
        } catch (NoSuchAlgorithmException e) {
            return null;
        }
        return strDes;
    }

    public static String bytes2Hex(byte[] bts) {
        String des = "";
        String tmp = null;
        for (int i = 0; i < bts.length; i++) {
            tmp = (Integer.toHexString(bts[i] & 0xFF));
            if (tmp.length() == 1) {
                des += "0";
            }
            des += tmp;
        }
        return des;
    }
}
```

4. Contact

IWOMI Technologies Ltd
BP 7218 BONANJO, Douala - Cameroun
Tel : +237 677 85 91 47 / 6 76 35 18 13
Email : info@softeller.com
Website: www.softeller.com
Facebook: www.facebook.com/softeller
Twitter: @SoftellerCM