



# Wallet Management KIT

## Table of contents

- [1. Introduction](#)
- [2. General information](#)
  - [2.1 Format](#)
  - [2.2 HTTP codes](#)
- [3. Authentication](#)
- [4. List of wallet APIs](#)
  - [4.1 - Creating a wallet](#)
  - [4.2 - Consulting customer information](#)
  - [4.3 - Consultation of transaction history](#)
  - [4.4 - Balance consultation](#)
  - [4.5 - Cash IN](#)
  - [4.6 - Cash OUT](#)
  - [4.7 - Wallet to Wallet](#)
  - [4.8 - Transfer](#)
  - [4.9 - ATM withdrawal](#)
  - [4.10 - Wallet to Merchant transaction](#)
  - [4.11 - Creating a Merchant Wallet](#)
  - [4.12 - Merchant to Merchant transaction](#)
  - [4.13 - Dynamic QR code](#)
  - [4.14 - Merchant to Wallet transaction](#)
- [5 - Mock-er an API](#)
  - [5.1 - Mocking directly into the code](#)

## 1. Introduction

This documentation describes the endpoints made available as part of the Hackathon.

The APIs are not accessible in production: participants must mock-er the endpoints and answers.

## 2. General information

### 2.1 Format

- Exchange format: JSON
- Encoding: UTF-8
- Supported methods: GET, POST

### 2.2 HTTP codes

Wallet management KIT uses standard HTTP response codes to indicate the success or failure of an API request.

Code	Meaning
200	Success
201	Created
400	Invalid request
404	Not found
500	Fictitious server error

## 3. Authentication

No real authentication is used in the hackathon.

Participants must mock-enter calls without a token.

## 4. List of wallet APIs

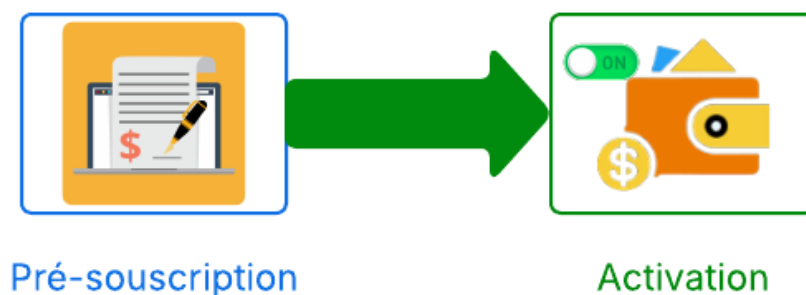
Wallet services provide a full range of functionality for managing electronic wallets. These services include :

- Wallet creation (customer / merchant)
- Consultation of customer information
- Consultation of transaction history
- Consultation of balance
- Cash IN / Cash OUT
- Wallet to Wallet transaction
- Merchant payment
- QR code generation to initiate payment
- Bank transfer
- ATM withdrawal

## 4.1 - Creating a wallet

Wallet creation is a two-step process, to ensure security and verify the customer's identity:

- Pre-subscription (enter customer details).
- Activation (identity verification by OTP).



Wallet creation

### 4.1.1 - Wallet pre-registration

#### Description

Pre-subscribe a customer wallet (enter customer information).

#### Method & URL

POST /wallet

### Parameters

state=precreate

### Body (JSON)

```
{
  "phoneNumber": " 212700446631",
  "phoneOperator": "IAM",
  "clientFirstName": "",
  "clientLastName": "",
  "email": "",
  "placeOfBirth": "",
  "dateOfBirth": "",
  "clientAddress": "",
  "gender": "",
  "legalType": "",
  "legalId": ""
}
```

### Sample response (JSON)

```
{
  "result": {
    "activityArea": null,
    "addressLine1": "",
    "addressLine2": null,
    "addressLine3": null,
    "addressLine4": null,
    "agencId": "211",
    "averageIncome": null,
    "birthDay": null,
    "channelId": "P",

```

```
"city": null,  
"country": null,  
"dateOfBirth": "",  
"distributeurId": "000104",  
"documentExpiryDate1": null,  
"documentExpiryDate2": null,  
"documentScan1": "",  
"documentScan2": "",  
"documentType1": "",  
"documentType2": null,  
"email": "",  
"familyStatus": null,  
"firstName": "Prenom",  
"fonction": null,  
"gender": "",  
"institutionId": "0001",  
"landLineNumber": null,  
"lastName": "nom",  
"legalId1": "",  
"legalId2": null,  
"level": null,  
"mailaddress": null,  
"mobileNumber": "212700446631",  
"nationalite": null,  
"numberofchildren": null,  
"optField1": null,  
"optField2": null,  
"otp": "123456",  
"phoneNumber": null,  
"placeOfBirth": "",  
"postCode": null,  
"productId": "000",  
"productTypeId": "000",  
"profession": null,  
"provider": "INWI",  
"raisonSocial": null,  
"region": null,  
"registrationDate": null,
```

```
"title": null,  
"token": "TR2404781353895901"  
}  
}
```

#### 4.1.2 - Wallet activation

##### Description

Activate a customer wallet

##### Method & URL

POST /wallet

##### Parameters

state=activate

##### Body (JSON)

```
{  
  "otp": "123456", //otp reçu /  
  "token": " TR2404781353895901" //Token récupéré du ws de la précréatio  
  n//  
}
```

##### Sample response (JSON)

```
{  
  "result": {  
    "contractId": "LAN240478508299911",  
    "reference": "",  
    "level": "000",  
    "rib": "853780241716465970216211"  
  }  
}
```

## 4.2 - Consulting customer information

## Description

View a customer's profile.

Consultation is by phone number and ID.

## Method & URL

```
POST /wallet/clientinfo
```

## Parameters

## Body (JSON)

```
{
  "phoneNumber": "212700446211",
  "identificationType": "CIN",
  "identificationNumber": "BK12232"
}
```

## Sample response (JSON)

```
{
  "result": {
    "adressLine1": " ",
    "city": " ",
    "contractId": null,
    "country": "MAR",
    "description": null,
    "email": "",
    "numberOfChildren": null,
    "phoneNumber": "212700446211",
    "pidNUmber": null,
    "pidType": "",
    "products": [
      {
        "abbreviation": null,
        "contractId": "LAN233460579578271",
```

```
"description": null,
"email": "",
"level": "",
"name": "CDP BASIC",
"phoneNumber": "212700446211",
"productId": "000",
"productName": "PARTICULIER",
"provider": "ORANGE ",
"rib": "853455230818452878570832",
"solde": "0.00",
"statusId": "1",
"tierType": "03",
"uid": "000"
}
],
"radical": "",
"soldeCumule": "0.00",
"statusId": null,
"tierFirstName": "Prenom",
"tierId": "TR2334600322963601",
"tierLastName": "nom",
"userName": null,
"familyStatus": null
}
}
```

## 4.3 - Consultation of transaction history

### Description

Retrieve transaction history by contract number

### Method & URL

```
GET /wallet/operations
```

### Parameters



contractid= LAN193541347060000000001

## Body (JSON)

## Sample response (JSON)

```
{
  "result": [
    {
      "amount": "10.00",
      "Fees": "4 " ,
      "beneficiaryFirstName": "Prenom",
      "beneficiaryLastName": "nom",
      "beneficiaryRIB": null,
      "clientNote": "W2W",
      "contractId": null,
      "currency": "MAD",
      "date": "12/8/2023 5:59:39 PM",
      "dateToCompare": "0001-01-01T00:00:00Z",
      "frais": [],
      "numTel": null,
      "operation": null,
      "referenceId": "1181798513",
      "sign": null,
      "srcDestNumber": "212755123456",
      "status": "000",
      "totalAmount": "10.00",
      "totalFrai": "6.00",
      "type": "MMD",
      "isCanceled": false,
      "isTierCashIn": false,
      "totalPage": 163
    },
    {
      "amount": "10.00",
      "Fees": "4 " ,
      "beneficiaryFirstName": "DJGHJDGJ",
      "beneficiaryLastName": "DGJDGHJDG",

```

```
"beneficiaryRIB": null,
"clientNote": "W2W",
"contractId": null,
"currency": "MAD",
"date": "12/8/2023 5:54:44 PM",
"dateToCompare": "0001-01-01T00:00:00Z",
"frais": [],
"numTel": null,
"operation": null,
"referenceId": "1792782055",
"sign": null,
"srcDestNumber": "212666233333",
"status": "000",
"totalAmount": "10.00",
"totalFrai": "6.00",
"type": "MMD",
"isCanceled": false,
"isTierCashIn": false,
"totalPage": 163
},
{
"amount": "10.00",
"Fees": "4 " ,
"beneficiaryFirstName": "DJGHJDGJ",
"beneficiaryLastName": "DGJDGHJDG",
"beneficiaryRIB": null,
"clientNote": "W2W",
"contractId": null,
"currency": "MAD",
"date": "12/8/2023 5:54:20 PM",
"dateToCompare": "0001-01-01T00:00:00Z",
"frais": [],
"numTel": null,
"operation": null,
"referenceId": "0548510077",
"sign": null,
"srcDestNumber": "212666233333",
"status": "000",
```

```
"totalAmount": "10.00",  
"totalFrai": "6.00",  
"type": "MMD",  
"isCanceled": false,  
"isTierCashIn": false,  
"totalPage": 163  
}  
]  
}
```

## 4.4 - Balance consultation

### Description

Retrieve wallet balance

### Method & URL

GET /wallet/balance

### Parameters

contractid= **MLAN19303084416000000001** (obligatoire)

### Body (JSON)

### Sample response (JSON)

```
{  
  "result": {  
    "balance": [  
      {  
        "value": "12556,88"  
      }  
    ]  
  }  
}
```

## 4.5 - Cash IN

There are two steps to feeding a wallet:

- Cash IN Simulation: to enter all transaction information.
- Cash IN Confirmation: to validate the transaction.

### 4.5.1 - Cash IN Simulation

#### Description

CASH IN transaction simulation

#### Method & URL

```
POST /wallet/cash/in
```

#### Parameters

```
step=simulation  
Fees (Facultatif)
```

#### Body (JSON)

```
{  
  "contractId": "LAN230325007133701",  
  "level": "2",  
  "phoneNumber": "212666141490",  
  "amount": "10" ,  
  "fees": "0"  
}
```

#### Sample response (JSON)

```
{ "result": {  
  "Fees": "0.0",  
  "feeDetail": "[{Nature:\\"COM\\",InvariantFee:0.000,VariantFee:0.0000000}]",  
  "token": "9E120058B31A4B77BA3A1CEF96142681",  
  "amountToCollect": 10,  
  "isTier": true,  
  "cardId": "LAN230325007133701",
```

```
"transactionId": "200116022024182720001",  
"benFirstName": "cba",  
"benLastName": "Abc"  
}  
}
```

#### 4.5.2 - Cash IN Confirmation

##### Description

Confirmation of CASH IN operation

##### Method & URL

POST /wallet/cash/in

##### Parameters

step=confirmation  
Fees (Facultatif)

##### Body (JSON)

```
{  
  "token": "9E120058B31A4B77BA3A1CEF96142681",  
  "amount": "10",  
  "fees": "0"  
}
```

##### Sample response (JSON)

```
{"result": {  
  "Fees": "0.0",  
  "feeDetails": null,  
  "token": "9E120058B31A4B77BA3A1CEF96142681",  
  "amount": 10,  
  "transactionReference": "0288284881",  
  "optFieldOutput1": null,  
  "optFieldOutput2": null,  
}
```

```
"cardId": "LAN230325007133701"
}
}
```

## 4.6 - Cash OUT

Withdrawing funds from a wallet takes place in three stages:

- Cash OUT Simulation: enter transaction information.
- Cash OUT OTP: generation of the OTP for transaction validation.
- Cash OUT Confirmation: transaction validation.

### 4.6.1 - Cash OUT Simulation

#### Description

cash withdrawal simulation from a customer account

#### Method & URL

POST /wallet/cash/out

#### Parameters

step=simulation  
**Fees** (Facultatif)

#### Body (JSON)

```
{
  "phoneNumber": " 212672300232",
  "amount": "10",
  "fees":"0"
}
```

#### Sample response (JSON)

```
{ "result": {
  "Fees":"0.0",
  "token": "C03C2A65318440A093670380B276F2E2",
  "amountToCollect": 10,
```

```
"cashOut_Max": 1918,  
"optFieldOutput1": null,  
"optFieldOutput2": null,  
"cardId": "LAN230748934021281",  
"transactionId": "200816022024182925001",  
"feeDetail": "[{Nature:\"COM\",InvariantFee:0.000,VariantFee:0.00000000}]"  
}  
}
```

#### 4.6.2 - Cash OUT OTP

##### Description

Generates an OTP for the CASH OUT operation.

##### Method & URL

```
POST /wallet/cash/out/otp
```

##### Parameters

##### Body (JSON)

```
{  
  "phoneNumber": " 212672300232"  
}
```

##### Sample response (JSON)

```
{  
  "result": [  
    {  
      "codeOtp": "771401"  
    }  
  ]  
}
```

#### 4.6.3 - Cash OUT Confirmation

## Description

Confirmation of CASHOUT operation

## Method & URL

POST /wallet/cash/out

## Parameters

step=confirmation

Fees (Facultatif)

## Body (JSON)

```
{
  "token": " C03C2A65318440A093670380B276F2E2",
  "phoneNumber": "212655305530",
  "otp": "123456",
  "amount": "10",
  "fees": "0"
}
```

## Sample response (JSON)

```
{
  "result": {
    "Fees": "0.0",
    "feeDetails": null,
    "token": "C03C2A65318440A093670380B276F2E2",
    "amount": 10,
    "transactionReference": "1354275502",
    "optFieldOutput1": null,
    "optFieldOutput2": null,
    "cardId": "LAN230748934021281"
  }
}
```



## 4.7 - Wallet to Wallet

The transfer between two wallets takes place in three stages:

- Wallet to Wallet Simulation: enter transaction information.
- Wallet to Wallet OTP: generate the OTP for transaction validation.
- Wallet to Wallet Confirmation: validate the transaction.

### 4.7.1 - Wallet to Wallet Simulation

#### Description

issuing a wallet-to-wallet transfer

#### Method & URL

```
POST /wallet/transfer/wallet
```

#### Parameters

```
step=simulation  
Client Note(Facultatif)  
Fees (Facultatif)
```

#### Body (JSON)

```
{  
  "clentNote": "W2W",  
  "contractId": "LAN193541347060000000001",  
  "amout": "2",  
  "fees": "0",  
  "destinationPhone": "212755123456",  
  "mobileNumber": "212666233333"  
}
```

#### Sample response (JSON)

```
{  
  "result": {  
    "amount": "2",  
    "Fees": "0.0",
```

```
"beneficiaryFirstName": "Prenom",
"beneficiaryLastName": "nom",
"beneficiaryRIB": null,
"contractId": null,
"currency": null,
"date": null,
"dateToCompare": "0001-01-01T00:00:00Z",
"frais": [
{
"currency": "MAD",
"fullName": "",
"name": "COM",
"referenceId": "2067400459",
"value": 5
},
{
"currency": "MAD",
"fullName": "",
"name": "TVA",
"referenceId": "2067400459",
"value": 1
}
],
"numTel": null,
"operation": null,
"referenceId": "2067400459",
"sign": null,
"srcDestNumber": null,
"status": null,
"totalAmount": "8.00",
"totalFrai": "6.00",
"type": "TT",
"isCanceled": false,
"isTierCashIn": false
}
}
```

#### 4.7.2 - Wallet to Wallet OTP

## Description

Generate OTP for wallet to wallet confirmation service

## Method & URL

```
POST /wallet/transfer/wallet/otp
```

## Parameters

## Body (JSON)

```
{
  "phoneNumber": " 212666233333"
}
```

## Sample response (JSON)

```
{
  "result": [
    {
      "codeOtp": "556263"
    }
  ]
}
```

### 4.7.3 - Wallet to Wallet confirmation

## Description

Confirmation that a wallet-to-wallet transfer has been sent

## Method & URL

```
POST /wallet/transfer/wallet
```

## Parameters

```
step=confirmation
```

## Body (JSON)

```
{
  "mobileNumber": "212666233333",
  "contractId": "LAN1935413470600000000001",
  "otp": "123456",
  "referenceId": "2067400459",
  "destinationPhone": "212650134393",
  "fees": "0"
}
```

### Sample response (JSON)

```
{
  "result": {
    "item1": {
      "creditAmounts": null,
      "debitAmounts": null,
      "depot": null,
      "retrait": null,
      "value": "-245384.090"
    },
    "item2": "000",
    "item3": "Successful"
  }
}
```

## 4.8 - Transfer

The wire transfer transaction takes place in three stages:

- Transfer Simulation: enter transaction information.
- Transfer OTP: generate the transaction validation OTP.
- Transfer Confirmation: validate the transaction.

### 4.8.1 - Transfer Simulation

#### Description

The simulation of the send transfer operation.

#### Method & URL

POST /wallet/transfer/virement

### Parameters

step=simulation

### Body (JSON)

```
{
  "clientNote": "W2W",
  "ContractId": "LAN252387936812761",
  "Amount": "12",
  "destinationPhone": "212665873350",
  "mobileNumber": "212669268097",
  "RIB": "230780530712622100950179"
}
```

### Sample response (JSON)

```
{
  "result": [
    {
      "frais": "0",
      "fraisSms": null,
      "totalAmountWithFee": "12",
      "deviseEmissionCode": null,
      "fraisInclus": false,
      "montantDroitTimbre": 0,
      "montantFrais": 0,
      "montantFraisSMS": 0,
      "montantFraisTotal": 0,
      "montantTVA": 0,
      "montantTVASMS": 0,
      "tauxChange": 0
    }
  ]
}
```

## 4.8.2 - Transfer OTP

### Description

Sends an OTP code to the specified phone number.

### Method & URL

```
POST /wallet/transfer/virement/otp
```

### Parameters

### Body (JSON)

```
{  
  "PhoneNumber": "212669268097"  
}
```

### Sample response (JSON)

```
{  
  "result": "123456"  
}
```

## 4.8.3 - Transfer Confirmation

### Description

Confirmation of the transfer operation.

### Method & URL

```
POST /wallet/transfer/virement
```

### Parameters

```
step=confirmation
```

### Body (JSON)

```
{
  "mobileNumber": "212669268097",
  "ContractId": "LAN252387936812761",
  "Otp": "123456",
  "referenceId": "0152475499",
  "destinationPhone": "212665873350",
  "fees": "0",
  "Amount": "12",
  "RIB": "230780530712622100950179",
  "NumBeneficiaire": "212665873350",
  "DestinationFirstName": "test firstname",
  "DestinationLastName": "test lastname"
}
```

### Sample response (JSON)

```
{
  "result": {
    "contractId": "LAN252387936812761",
    "reference": "709846211156"
  }
}
```

## 4.9 - ATM withdrawal

There are three steps to ATM withdrawal:

- ATM withdrawal Simulation: enter transaction information.
- ATM Withdrawal OTP: generate transaction confirmation OTP.
- ATM Withdrawal Confirmation: transaction validation.

### 4.9.1 - ATM Withdrawal Simulation

#### Description

API used to simulate an ATM withdrawal from a wallet. It validates the input parameters (amount, issuing wallet) and returns a transaction identifier with the data required for the rest of the process.

#### Method & URL

```
POST /wallet/cash/gab/out
```

### Parameters

```
step=simulation
```

### Body (JSON)

```
{
  "ContractId": " LAN251276004694521",
  "Amount": "200"
}
```

### Sample response (JSON)

```
{
  "result": {
    "totalFrai": "3.00",
    "feeDetails":
    "[{Nature:\"COM\",InvariantFee:3.000,VariantFee:0.0000000},{Nature:\"TV
    A\",Invariant
    Fee:0.000,VariantFee:0.27}]",
    "token": "2383FD7C082545C696E8597FDAE8EB21",
    "totalAmount": 203,
    "referenceId": "0273802387"
  }
}
```

## 4.9.2 - ATM withdrawal OTP

### Description

API triggering the sending of an OTP to the number linked to the issuing wallet. It is used to secure the operation before confirmation, with code expiration management.

### Method & URL

```
POST /wallet/cash/gab/otp
```



## Parameters

## Body (JSON)

```
{
  "phoneNumber": "0671219423"
}
```

## Sample response (JSON)

```
{
  "result": [
    {
      "codeOtp": "123456"
    }
  ]
}
```

### 4.9.3 - ATM withdrawal Confirmation

#### Description

ATM withdrawal confirmation API. It consumes the OTP entered by the user and finalizes the

the transaction by debiting the wallet and generating a proof of withdrawal.

#### Method & URL

POST /wallet/cash/gab/out

## Parameters

step=confirmation

## Body (JSON)

```
{
  "ContractId": " LAN251276004694521",
  "PhoneNumberBeneficiary": "0671219423",
}
```

```
"Token": "FC10F77242AA41D9A7632623EAD0FF96",
"ReferenceId": "1644819990",
"Otp": "123456"
}
```

### Sample response (JSON)

```
{
  "result": {
    "fee": "0.00",
    "feeDetails": null,
    "token": "CED3B51B4FD245C39B961C5CABA816EF",
    "amount": 200,
    "transactionReference": "",
    "cardId": "LAN251276004694521",
    "transactionId": 1321768214,
    "transfertCihExpressReference": "00230110002126023062025",
    "redCode": null,
    "greenCode": null
  }
}
```

## 4.10 - Wallet to Merchant transaction

This transaction takes place in three stages:

- Wallet to Merchant Simulation.
- Wallet to Merchant OTP
- Wallet to Merchant Confirmation.

### 4.10.1 - Wallet to Merchant Simulation

#### Description

Initiate Wallet to Merchant transaction.

#### Method & URL

POST /wallet/Transfer/WalletToMerchant

#### Parameters

step=simulation

### Body (JSON)

```
{
  "clientNote": "test",
  "clientContractId": "LAN250383003224941",
  "Amout": "10",
  "clientPhoneNumber": "212665873350",
  "merchantPhoneNumber": "212657575733"
}
```

### Sample response (JSON)

```
{
  "result": {
    "amount": "10",
    "beneficiaryFirstName": "cddsds",
    "beneficiaryLastName": "EDFdd",
    "beneficiaryRIB": null,
    "clientNote": "test",
    "contractId": null,
    "currency": null,
    "date": null,
    "dateToCompare": "0001-01-01T00:00:00Z",
    "frais": [],
    "numTel": null,
    "operation": null,
    "referenceId": "0116178499",
    "sign": null,
    "srcDestNumber": null,
    "status": null,
    "totalAmount": "10",
    "totalFrai": "0",
    "type": "TM",
    "isCanceled": false,
    "isTierCashIn": false,
    "feeDetails": null,
  }
}
```

```
"token": null,  
"optFieldOutput1": null,  
"optFieldOutput2": null,  
"cardId": null,  
"isSwitch": false  
}  
}
```

#### 4.10.2 - Wallet to Merchant OTP

##### Description

Send OTP for Wallet to Merchant transaction.

##### Method & URL

```
POST /wallet/walletToMerchant/cash/out/otp
```

##### Parameters

##### Body (JSON)

```
{  
  "phoneNumber": "212665873350"  
}
```

##### Sample response (JSON)

```
{  
  "result": [  
    {  
      "codeOtp": "326746"  
    }  
  ]  
}
```

#### 4.10.3 - Wallet to Merchant Confirmation

##### Description

Confirm a fund transfer from a Wallet to a merchant after running the simulation, including the OTP to verify the transaction.

### Method & URL

```
POST /wallet/Transfer/WalletToMerchant
```

### Parameters

```
step=confirmation
```

### Body (JSON)

```
{
  "ClientPhoneNumber": "0612345678",
  "ClientContractId": "CONTRACT123",
  "OTP": "123456",
  "ReferenceId": "REF123456789",
  "DestinationPhone": "0698765432",
  "QrCode": "QR123456789",
  "MCC": "5411",
  "AmountInputMode": "ENTERED",
  "fees": "0"
}
```

### Sample response (JSON)

```
{
  "result": {
    "item1": {
      "creditAmounts": null,
      "debitAmounts": null,
      "depot": null,
      "retrait": null,
      "value": "17.460",
      "transactionId": null,
      "cardId": null,
      "optFieldOutput2": null,
      "optFieldOutput1": null,
    }
  }
}
```

```
"transactionReference": null,  
"amount": null,  
"token": null,  
"fee": null,  
"feeDetails": null  
},  
"item2": "000",  
"item3": "Successful"  
}  
}
```

## 4.11 - Creating a Merchant Wallet

Merchant wallet creation takes place in two stages:

- Merchant Wallet Pre-creation.
- Merchant Wallet activation.

### 4.11.1 - Merchant Wallet Pre-creation

#### Description

Merchant wallet pre-creation.

#### Method & URL

POST /merchants

#### Parameters

#### Body (JSON)

```
{  
  "FirstName": "El",  
  "LastName": "Ta",  
  "MobileNumber": "06000000000",  
  "Provider": "IAM",  
  "Email": "el.ta@example.com",  
  "NumberOfChildren": "4",  
  "Profession": "Artisan",  
}
```

```
"AverageIncome": "50000",
"ActivitySector": "Banking",
"ActivityType": "Software",
"Gender": "M",
"IdentificationDocumentType": "CIN",
"IdentificationNumber": "BK4444",
"IdentificationExpiryDate": "31122030",
"PlaceOfBirth": "Casablanca",
"DateOfBirth": "20051985",
"LandlineNumber": "0522222222",
"AddressLine1": "Rue 1",
"AddressLine2": "Appt 5",
"City": "Rabat",
"JobPosition": "Gérant",
"Nationality": "Marocaine",
"PostalCode": "10000",
"Country": "MAROC",
"CompanyName": "SARL ShoPech",
"BusinessAddress": "Avenue des entreprises",
"CommercialRegistrationNumber": "RC123456",
"TaxIdentificationNumber": "IF789456",
"CompanyRegistrationId": "ICE789123",
"TradeLicenseNumber": "Lic123456",
"MerchantCategory": "Retail",
"ProfessionalTaxNumber": "TP789456",
"RegistrationCenter": "Centre Rabat",
"LegalStructure": "SARL",
"RegistryRegistrationNumber": "RR123456"
}
```

### Sample response (JSON)

```
{
  "result": {
    "token": "ME2519962089316801"
  }
}
```

### 4.11.2 - Merchant Wallet activation

#### Description

Merchant wallet activation

#### Method & URL

```
POST /merchant/activate
```

#### Parameters

#### Body (JSON)

```
{
  "Token": "ME2519962089316801",
  "Otp": "123456"
}
```

#### Sample response (JSON)

```
{
  "result": {
    "contractId": "LAN251996372325421"
  }
}
```

## 4.12 - Merchant to Merchant transaction

The Merchant to Merchant Transaction takes place in three stages:

- Merchant to Merchant Simulation.
- Merchant to Merchant OTP.
- Merchant to Merchant Confirmation.

### 4.12.1 - Merchant to Merchant Simulation

#### Description

Transfer funds between two merchant accounts.

#### Method & URL



POST /merchant/transaction/simulation

### Parameters

### Body (JSON)

```
{
  "ClientNote": "M2M",
  "ContractId": "LAN251114678086481",
  "Amount": "100",
  "DestinationPhone": "212645478824",
  "MobileNumber": "212758692536"
}
```

### Sample response (JSON)

```
{
  "result": [
    {
      "amount": "100",
      "beneficiaryFirstName": "Itechia",
      "beneficiaryLastName": "Itechia",
      "beneficiaryRIB": null,
      "clientNote": "M2M",
      "contractId": null,
      "currency": null,
      "date": null,
      "dateToCompare": "0001-01-01T00:00:00Z",
      "frais": [
        {
          "currency": "MAD",
          "fullName": "",
          "name": "COM",
          "referenceId": "0009251901",
          "value": 3.33
        },
        {

```

```
"currency": "MAD",
"fullName": "",
"name": "TVA",
"referenceId": "0009251901",
"value": 0.67
}
],
"numTel": null,
"operation": null,
"referenceId": "0009251901",
"sign": null,
"srcDestNumber": null,
"status": null,
"totalAmount": "100",
"totalFrai": "4.00",
"type": "CC",
"isCanceled": false,
"isTierCashIn": false,
"walletType": ""
}
]
```

#### 4.12.2 - Merchant to Merchant OTP

##### Description

OTP generation.

##### Method & URL

POST /merchant/transaction/otp

##### Parameters

##### Body (JSON)

```
{
  "phoneNumber": "212758692536"
```

```
}
```

### Sample response (JSON)

```
{
  "result": [
    {
      "codeOtp": "320237"
    }
  ]
}
```

## 4.12.3 - Merchant to Merchant Confirmation

### Description

Confirmation and execution of a funds transfer between two merchant accounts.

### Method & URL

POST /merchant/transaction/confirmation

### Parameters

### Body (JSON)

```
{
  "MobileNumber": "212758692536",
  "ContractId": "LAN251114678086481",
  "Otp": "123456",
  "ReferenceId": "0009251901"
}
```

### Sample response (JSON)

```
{
  "result": {
    "creditAmounts": null,

```

```

    "debitAmounts": null,
    "depot": null,
    "retrait": null,
    "value": "800.000"
  }
}

```

## 4.13 - Dynamic QR code

**Description** Generate a QR code by the merchant (containing his information as well as the amount to be received) which will be scanned by the customer to make the payment.

### Method & URL

POST /wallet/pro/qrcode/dynamic

### Parameters

### Body (JSON)

```

{
  "phoneNumber": "21266587335",
  "contractId": "LAN250383003221",
  "amount": "100"
}

```

### Sample response (JSON)

```

{
  "result": {
    "phoneNumber": "212665873350",
    "reference": "",
    "token": "050620251247094208472",
    "base64Content":
    "iVBORw0KGgoAAAANSUHEUgAAB0QAAAdEAQAAAAD89IUSAAQTEIEQ
    VR4nO3YSbLjyg0A
    QN6A978lbyCHHa1XmKjuBfWHcmKhEFmFlaGdjtf/SVzH3z3BXxWk+wXpfk
  }
}

```

```
G6X5DuF6T7Bel+
+NvH7/ +Q7hek+
Q7hek+wXpFkG6X5DuF6T7Bel+QbpFkO4XpPsF6X5Bul+Q7hek+wXpFkG6X
5DuF6T7Bel+QbpF
kO4XpPsF6X5Bul+Q7hek+8X/kfQ/V0JH4IBFAtAAAAAASUVORK5CYII=",
"binaryContent":
"000201010212269300325bb66a92d69c0ea742dd4f754590fa0a02011050
100624ibCx5kz
meNMW6EcTvEaaBw==0715+212665#####50520454115303504540310
05802MA5907Epi
cier6010Casablanca6241020003000421050620251247094208472050006
0064330002en01
06GROCER0213CASABLANCA0008088003237b3a355b830b3bf0974d236
08a6f1620101102
160001-01-01T00:00031833.529704,-7.639210401063045987"
}
}
```

## 4.14 - Merchant to Wallet transaction

The Merchant to Wallet transaction takes place in three stages:

- Merchant to Wallet Simulation.
- Merchant to Wallet OTP.
- Merchant to Wallet Confirmation.

### 4.14.1 - Merchant to Wallet Simulation

#### Description

Initiate a Merchant to Wallet transfer.

#### Method & URL

```
POST /merchant/merchantToWallet/simulation
```

#### Parameters

#### Body (JSON)

```
{
  "ContractId": "LAN251996372325421",
  "Amount": 15,
  "BeneficiaryPhoneNumber": "212600000554"
}
```

#### Sample response (JSON)

```
{
  "result": {
    "amount": 15.0,
    "feeAmount": 0.0
  }
}
```

### 4.14.2 - Merchant to Wallet OTP

#### Description

Sends an OTP code to the specified phone number.

#### Method & URL

POST /merchant/otp/send

#### Parameters

#### Body (JSON)

```
{
  "phoneNumber": "212627248340"
}
```

#### Sample response (JSON)

```
{
  "result": ""
}
```

---

### 4.14.3 - Merchant to Wallet Confirmation

#### Description

Merchant to wallet transfer confirmation.

#### Method & URL

```
POST /merchant/merchantToWallet/confirmation
```

#### Parameters

#### Body (JSON)

```
{
  "ContractId":"LAN251996372325421",
  "BeneficiaryPhoneNumber":"212600000554",
  "Amount": 15,
  "Otp":"123456"
}
```

#### Sample response (JSON)

```
{
  "result": {
    "contractId": null,
    "reference": "170436959354",
    "transferAmount": 0
  }
}
```

## 5 - Mock-er an API

During the hackathon, if the API is not available, you can simulate it to continue developing.

There are two ways of doing this: mock-ing in the code or using an external tool.

---

## 5.1 - Mocking directly into the code

You create functions that return "fake" data instead of calling the real API.

Example:

```
const mockLogin = () => ({  
  token: "demo-token",  
  user: { id: 1, name: "demo" }  
});
```

✓ Ultra fast

✓ No external tool

🔗 *Quick tutorial (MDN fetch):*

[https://developer.mozilla.org/fr/docs/Web/API/fetch\\_api/Using\\_Fetch](https://developer.mozilla.org/fr/docs/Web/API/fetch_api/Using_Fetch)

---

## 5.2 - Mocking an API with tools

### Postman Mock Server

Online mock, accessible by the whole team.

Steps: Collection → Examples → Create Mock Server

🔗 Documentation:

<https://learning.postman.com/docs/designing-and-developing-your-api/mocking-data/setting-up-mock/>

---

### Mockoon

Desktop application to create a local mock very easily.

🔗 Download :

<https://mockoon.com/download/>

🔗 Beginner's guide :

<https://mockoon.com/tutorials/getting-started/>


---

### JSON Server

Complete REST API in 30 seconds.



```
npm install -g json-server  
json-server --watch db.json --port 3000
```

 Documentation :

<https://github.com/typicode/json-server>

### 5.3 - Essential rule

Respect EXACTLY the endpoints and signatures provided in the Wallet Management Kit.

- Same URL
- Same method (GET/POST/etc.)
- Same JSON structure
- Same parameters

Otherwise, your integration won't work once plugged into the real API.

**PS:** You can extend the Wallet Management Kit as long as the official endpoints remain unchanged.