|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SandboxSimulator Card Operations Service** | | | |  |
|  | | | | |
|  |  |  |  | |
| **Version** | BiB-1.27 |  |  | |

Content

[1. Revision History 2](#_Toc58860119)

[2. Card Authorization 3](#_Toc58860120)

[3. Operation Festopeme 5](#_Toc58860121)

[4. OpeRation Festopeme Partial Clearing 6](#_Toc58860122)

[5. TOTAL REVERSAL 7](#_Toc58860123)

[6. Partial reversal 7](#_Toc58860124)

[7. Apple Pay Enrollement 8](#_Toc58860125)

[7.1 Specifics Ressources 8](#_Toc58860126)

[7.2 Green Flow Enrolment 9](#_Toc58860127)

[7.3 Yellow Flow Enrolment 10](#_Toc58860128)

[7.4 Orange Flow Enrolment 11](#_Toc58860129)

[7.5 Red Flow Enrolment 11](#_Toc58860130)

[8. Custom Card Authorization 11](#_Toc58860131)

[9. Custom clearing 13](#_Toc58860132)

[10. Card Opposition 14](#_Toc58860133)

[11. DAC / Fuel DIspenser 14](#_Toc58860134)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version / revision** | **Date** | **Author** | **Description** |
| BiB-v1.0 | 18-06-2019 | Yassine Bouafif | 1st version of the doc for BiB SandboxSimulator service |
| BIB-V1.1 | 08-07-2019 | Hadrien Lecoq | Update with Authorization type’s Description |
| BIB-V1.2 | 09-07-2019 | Giampiero Miccoli | §2.4 : update output Authorization  Add §3 : SCT IN API description |
| BIB-V1.3 | 10-07-2019 | Yassine Bouafif | Add operation festopeme |
| BIB-V1.4 | 24-07-2019 | Giampiero Miccoli | Add SCT IN RECALL API description |
| BIB-V1.5 | 16-08-2019 | Nabil Hamza | Add SDD Out API description |
| BIB-V1.6 | 02-10-2019 | Yassine Bouafif | Add partial clearing |
| BIB-V1.7 | 14-10-2019 | Alexandru Magereanu | Removed domainId parameter from SCT IN doc |
| BIB-V1.8 | 13-11-2019 | Giampiero Miccoli | Removed ThirdPartyBankAccountId parameter from SCT IN doc |
| BIB-V1.9 | 18-11-2019 | Alexandru Magereanu | Added SandboxSimulator URL |
| BIB-V1.10 | 18-12-2019 | Fahd Aloui | Added DAC Payment and DAC redressment |
| BIB-V1.11 | 19-12-2019 | Giampiero Miccoli | Add “Delete a user” section §9 + |
| BIB-V1.12 | 02-01-2020 | Simon ROUX, Fahd Aloui | Add some info on how to use DAC API  + correct some API URLs + Error messages for DAC API |
| BIB-V1.13 | 08-01-2020 | Yassine Bouafif | Add total and partiel reversal (Update Doc) |
| BIB-V1.14 | 26/02/2020 | Fahd Aloui | Added Apple Pay Enrollment |
| BIB-V1.15 | 14/05/2020 | Mohamed Marouani | Add Custom Card Authorization |
| BIB-V1.17 | 18/05/2020 | Mohamed Marouani | Add New Params For Custom Card Authorization |
| BIB-V1.18 | 20/05/2020 | Yassine Bouafif | Add New Params For Custom Clearing |
| BIB-V1.19 | 03/07/2020 | Yassine Bouafif | Add Card Opposition |
| BIB-V1.20 | 20/07/2020 | Hadrien Lecoq | Custom Clearing Corrections |
| BIB-V1.21 | 37/07/2020 | Hadrien Lecoq | Document separation (Card Ope / Payment Ope) |
| BIB-V1.22 | 24/11/2020 | Hadrien Lecoq | DAC case suppression |
| BIB-V1.23 | 24/11/2020 | Hadrien Lecoq | Xpay Urls |
| BIB-V1.24 | 11/12/2020 | Hadrien Lecoq | Add DAC context |
| BIB-V1.25 | 14/12/2020 | Hadrien Lecoq | Add OrangeFlow and RedFlow |
| BIB-V1.26 | 17/03/2021 | Hadrien Lecoq | added explanations for APIs |
| BIB-V1-27 | 27/09/2021 | Hadrien Lecoq | modification of the wording of the refusal code 116 |

# Card Authorization

The API simulator/cardauthorization allows to generate an authorization request with pre-determined contexts.

**Input**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Exemple** | **Description** |
| authorisation\_type | int | 1 | Authorization type = determine the case that will be simulated. |
| appcardid | String | ‘CardID’ | Card reference to create the transaction |
| amount | int | 10 | Transaction amount |

|  |  |
| --- | --- |
| **Authorization type** | **Label** |
| 1 | POS € Contact |
| 2 | POS € Contacless Card |
| 3 | POS € Contacless xPay |
| 4 | POS foreign currency (CHF) |
| 5 | eCommerce 3DS (ERT 24) |
| 6 | eCommerce unsecured |
| 7 | ATM € |
| 8 | ATM foreign currency (MAD) |
| 9 | DAC payment |
| 10 | xPay => authorization to check the card |
| 11 | POS € Custom Contact |

**Output**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Exemple** | **Description** |
| Id | int | 1254 | Id of the transaction |
| ActionCode | int | 0,160 | The result of the response to the request |
| Message | string | ‘Approuved’ | Output message |

**Request**

POST

{{URL}}/simulator/cardauthorization

**Body**

{

    "authorisation\_type" : 1,

    "appcardid" :"SP27STAG2\_CV",

    "Amount" : 1.23

}

**Response**

{

    "id": 11492,

    "actionCode": 0

}

**Action code and description**

|  |  |
| --- | --- |
| **Value** | **Description** |
| **0** | **Approved** |
| **100** | **Denied, do not honor** |
| **116** | **Denied, insufficient funds or limits reached** |
| **119** | **Denied, transaction not permitted to cardholder** |

# Operation Festopeme

The API simulator/cardauthorization/processfestopeme allows to generate a FestOpeme following the generation of an authorization. You can complete and then cancel an operation.

By default, the API takes the original authorization amount.

**Input**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Example** | **Description** |
| idTransaction | Long | 1234 | Id Transaction from espace partenaire |
| Operation | Int | 0,1 | 1: Validate transaction  0: Refund transaction |

NB : Refund transaction is only possible on a Validate transaction

**Request**

POST

{{URL}}/simulator/cardauthorization/processfestopeme

**Body example**

{

    "IdTransaction" : 11492,

    "Operation" : 1

}

**Response**

*Operation completed*

{

    "id": 11492,

    "statut": "Completed"

}

*Operation refunded*

{

    "id": 11492,

    "statut": "Refunded"

}

# OpeRation Festopeme Partial Clearing

The API simulator/cardauthorization/processfestopeme allows to generate a FestOpeme following the generation of an authorization. You can complete and then cancel an operation.  
  
In this case, the API allows to give to the FestOpeme operation a different amount (partial or higher than the initial amount of the authorization)

**Input**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Example** | **Description** |
| idTransaction | Long | 1234 | Id Transaction from espace partenaire |
| Operation | Int | 0,1 | 1: Validate transaction  0: Refund transaction |
| ClearedAmount | Decimal | 1 | New amount |

**Request**

POST

{{URL}}/simulator/cardauthorization/processfestopeme

**Body example**

{

    "IdTransaction" : 11494,

    "Operation" : 1,

    "ClearedAmount" : 1.5

}

**Response**

{

    "id": 11494,

    "statut": "Completed"

}

# TOTAL REVERSAL

The cardauthorization/cardredressment API is used to redress an authorization. The adjustment is total (the amount of the adjustment is equal to the amount of the initial authorisation)

**Input**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Example** | **Description** |
| idTransaction | Long | 1234 | Id Transaction from espace partenaire |

**Request**

GET

{{URL}}/cardauthorization/cardredressement

**URL example**

{{URL}}/simulator/cardauthorization/cardredressement?idTransaction=4122

**Response**

{

    "id": 7558,

    "actionCode": 400

}

**Action code list**

|  |  |
| --- | --- |
| **Value** | **Description** |
| 400 | Reversal accepted |

# Partial reversal

The cardauthorization/cardredressement/partial API allows for the partial adjustment of an authorization (the amount of the adjustment is less than the amount of the initial authorization)

**Input**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Example** | **Description** |
| idTransaction | Long | 1234 | Id Transaction from espace partenaire |
| Amount | Decimal | 12.5 | New amount (must be less than initial amount) |

**Request**

GET :

{{URL}}/simulator/cardauthorization/cardredressement/partial

**URL example:**

{{URL}}/simulator/cardauthorization/cardredressement/partial?idTransaction=4270&amount=5

**Response**

{

    "id": 7568,

    "actionCode": 400

}

**Action code list**

|  |  |
| --- | --- |
| **Value** | **Description** |
| 400 | Reversal accepted |

# Apple Pay Enrollement

Functionality to simulate the enrollment of a card in apple wallet.

## Specifics Ressources

**Message Reason Code**

|  |  |
| --- | --- |
| **Value** | **Description** |
| 1400 | Création token |
| 1401 | Suppression token |
| 1402 | Suspension token |
| 1403 | Réactivation token |
| 1411 | Résultat du devise provisioning |
| 1412 | Activation par OTP |
| 1413 | Activation par call center |
| 1414 | Activation par in App Provisioning |
| 1415 | Confirmation de réapprovisionnement des clés à usage lilmité |
| 1416 | Expiration du token |
| 1420 | Actualisation de la date d’expiration carte |
| 1421 | Actualisation du numéro de carte |
| 1427 | Enrôlement refusé par TSP |
| 1264 | Declined due to active account management (AAM) |
| 1300 | Device provisioning update results |
| 1454 | Declined due to CAM Failure |

**Token Status**

|  |  |
| --- | --- |
| A | Active for payment |
| I | Inactive for payement (not yet active) |
| S | Temporaly Suspended for payments |
| D | Permanently deactivated for payements |

## Green Flow Enrolment

The API can simulate a GreenFlow enrolment that does not require user authentication.

The system will generate :

|  |  |  |
| --- | --- | --- |
| 1 | an authorization that will verify the eligibility of the card | = callback type 20 |
| 2 | an authorization notice that informs of the creation of a token | = callback type 25  (with message reason code : 1400 and token status : I) |
| 3 | an authorization notice that informs of token activation | = callback type 25  (with message reason code : 1411 and token status : A) |

**Input**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Example** | **Description** |
| appCardId | String | 6Dt4io\_CP | The external card Id |

**OuTput**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Example** | **Description** |
| Id | Long | 1234 | The transaction id of the last authorization notice |
| ActionCode | Int | 0 | Will always be 0, since |

**API**

Verb: Post

Link: {{URL}}/simulator/cardauthorization/applepaygreenflow

Example: {{URL}}/simulator/cardauthorization/applepaygreenflow?appCardId=TEST10\_CP

Response:

{

    "id": 11488,

    "actionCode": 0

}

## Yellow Flow Enrolment

The API allows you to simulate a YellowFlow enrolment with user authentication by OTP.

The system will generate :

|  |  |  |
| --- | --- | --- |
| 1 | an authorization that will verify the eligibility of the card | = callback type 20 |
| 2 | an authorization notice that informs of the creation of a token | = callback type 25  (with message reason code : 1400 and token status : I) |
| 3 | an authorization notice indicating that enrolment is in progress | = callback type 25  (with message reason code : 1411 and token status : I) |
| 4 | a randomly generated OTP | = callback type 26 with OTP |
| 5’ | an authorization notice indicating that the token has been activated (when the OTP is validated) | = callback type 25  (with message reason code : 1412 and token status : A) |
| 5’’ | an authorization notice that indicates that the token is still inactive (when the OTP has not been validated) | = callback type 25  (with message reason code : 1412 and token status : I) |

**Input**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Example** | **Description** |
| appCardId | String | CxfZjn\_CP | The external card Id |
| otpOk | Boolean | True or False |  |

**Output**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Example** | **Description** |
| Id | Long | 1234 | The transaction id of the last authorization notice |
| ActionCode | Int | 0 | Will always be 0, since |

**API**

Verb: Post

Link: {{URL}}/simulator/cardauthorization/applepayyellowflow

Example :

Otp ok :

{{URL}}/simulator/cardauthorization/applepayyellowflow?appCardId=CxfZjn\_CP&otpOk=True

Otp ko :

{{URL}}/simulator/cardauthorization/applepayyellowflow?appCardId=CxfZjn\_CP&otpOk=false

**Response:**

{

    "id": 11490,

    "actionCode": 0

}

## Orange Flow Enrolment

The API allows you to simulate an OrangeFlow enrolment. In this case, enrolment is refused. In reality, the user must be asked to contact his bank.

The system will generate :

|  |  |  |
| --- | --- | --- |
| 1 | an authorization that will verify the eligibility of the card | = callback type 20 |
| 2 | an authorization notice that informs of the creation of a token | = callback type 25  (with message reason code : 1400 and token status : I) |
| 3 | an authorization notice indicating that enrolment is in progress | = callback type 25  (with message reason code : 1411 and token status : I) |
| 4 | a notice of authorization indicating that enrolment is refused | = callback type 25  (with message reason code : 1454 and token status : I) |

**API**

Verb: Post

Link: {{URL}}/simulator/cardauthorization/applepayorangeflow

Example: {{URL}}/simulator/cardauthorization/applepaygreenflow?appCardId=TEST10\_CP

Response:

{

    "id": 12235,

    "actionCode": 0

}

## Red Flow Enrolment

The API is used to simulate a refusal of enrolment. The refusal is made directly on the authorization request that allows the card to be checked.

The system will generate :

|  |  |  |
| --- | --- | --- |
| 1 | an authorization that will verify the eligibility of the card | = callback type 20 (refused) |

**API**

Verb: Post

Link: {{URL}}/simulator/cardauthorization/applepayredflow

Example: {{URL}}/simulator/cardauthorization/applepaygreenflow?appCardId=TEST10\_CP

Response:

{

    "id": 12236,

    "actionCode": 100

}

# Custom Card Authorization

The API simulator/cardauthorization/customauthorization allows to generate a custom authorization request.

**Input**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Exemple** | **Description** |
| AuthorisationType | int | 11 | Authorization type: Must be 11 |
| AppCardId | String | “CardID” | Card reference to create the transaction |
| Amount | int | 10 | Transaction amount |
| CodePays | int | 250 |  |
| TransactionDescription | String | “00” | Must be in (“00“, “01”) |
| MerchantType | int | 4000 | MCC must be numeric 4 |
| CardDataInputMode | int | 1 | Must be in (1,5,9) |
| MerchantName | String | "xx" | Mondatory (Max 38) |
| MerchantCity | String | “yy” | Optionnal (Max 38) |
| MerchantStreet | String | “zz” | Optionnal (Max 38) |
| ERT | int | 10 | Mast be in (10, 20, 21, 22, 24, 64, 25, 27, 28, 65, 70, 80, 84, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57) |

**OuTput**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Exemple** | **Description** |
| Id | int | 1254 | Id of the transaction |
| ActionCode | int | 0,160 | The result of the response to the request |

**Request**

POST:

{{URL}}/simulator/cardauthorization/customauthorization

**Body:**

{

 "AuthorisationType" : 11,

 "AppCardId" : "171018-130520-TEST-2532713",

 "Amount" : 1,

 "CodePays" : 250,

 "TransactionDescription":"00",

 "MerchantType":2500,

 "CardDataInputMode":1,

 "MerchantName":"xx",

 "MerchantCity":"yy",

 "MerchantStreet":"zz",

 "ERT":10

}

**Response**

{

    "id": 11490,

    "actionCode": 0

}

# Custom clearing

The API simulator/cardauthorization/processfestopmcustom is used to simulate a custom compensation.

**Input**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Exemple** | **Description** |
| AppCardId | String | “CardID” | Card reference to create the transaction |
| Amount | decimal | 10.2 | Transaction amount |
| operationDate | dateTime | “ 2020-05-15T00:00:00” | Operation date |
| direction | int | 1,2 | 1 : Crédit  2 : Débit |
| merchantName | String | “XXXX” | Merchant name |
| merchantCategoryCode | N4 | "2348" | Merchant category code |

**Ouput**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Exemple** | **Description** |
| Id | int | 1254 | Id of the transaction |
| Statut | string | Completed |  |

**Request**

POST:

{{URL}}/simulator/cardauthorization/processfestopmcustom

**Body:**

{

 "AppCardId": "TEST10\_CP",

 "Amount": 1.06,

 "OperationDate": "2020-11-20T17:27:00",

 "Direction": 2,

 "MerchantName": "MARCHANDPF",

 "MerchantCategoryCode": "7011"

}

**Response**

{

    "id": 11489,

    "statut": "Completed"

}

# Card Opposition

The API simulator/cardauthorization/CardOpposition allows to oppose a card

**Input**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Exemple** | **Description** |
| AppCardId | String | “CardID” | Card reference to create the transaction |

**Ouput**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Exemple** | **Description** |
| actionCode | int | 300 | Card Opposed |

**Request**

POST:

{{URL}}/simulator/cardauthorization/CardOpposition?appcardId=CARDTEST

**Response**

{"actionCode":300}

# DAC / Fuel DIspenser

The first step is to pre-authorize with the CardAuthorization API (chapter 2) using context 9.  
  
Then the cardDacRedressement API allows to simulate the final amount of the transaction once the user has finished taking gasoline and the amount is known.

Example =

* The user inserts his card and the distributor makes a pre-authorization of 120 euros. The balance of the user is impacted
* The user uses and takes 54 euros of fuel
* When the user has finished and the final amount is known, the distributor sends an adjustment to S-money which recalculates the user’s new balance.

**Request**

GET :

{{URL}}/simulator/cardauthorization/carddacredressement

**URL example** (NB : no body, input parameters (idtransaction, new amount) are provided in the URL) :

{{URL}}/simulator/cardauthorization/carddacredressement?idTransaction=5058&amount=2

**Response :**

* If no error :

{

"Id": 5058,

"ActionCode": 0

}

* The transaction is still pending, but the initial amount has been updated
* If error :

{

"Transaction amount should be greater than zero and less or equal than the initial transaction.",

 "UniqueId": "37c6452b-cd34-4fc8-adfc-84a9de9d02f8",

  "DateUtc": "2020-05-25T13:55:32.9357352Z"

}