|  |  |
| --- | --- |
| **SandboxSimulator Service – SCT/SDD** |  |
|  |
|  |  |  |  |
| **Version** | BiB-1.0 |  |  |

Content

[1. Revisions 3](#_Toc47453722)

[2. SANDBOX Simulator URL 3](#_Toc47453723)

[3. SCT IN 3](#_Toc47453724)

[3.1 Input 3](#_Toc47453725)

[3.2 Output 4](#_Toc47453726)

[3.3 Request 4](#_Toc47453727)

[3.4 Response 4](#_Toc47453728)

[4. SCT IN Recall 5](#_Toc47453729)

[4.1 Input 6](#_Toc47453730)

[4.2 SEPA reason code allowed 6](#_Toc47453731)

[4.3 Output 6](#_Toc47453732)

[4.4 Request 6](#_Toc47453733)

[4.5 Response 6](#_Toc47453734)

[4.6 Input 6](#_Toc47453735)

[4.7 Request 6](#_Toc47453736)

[4.8 Response 6](#_Toc47453737)

[4.9 Input 6](#_Toc47453738)

[4.10 Request 6](#_Toc47453739)

[4.11 Response 6](#_Toc47453740)

[5. SDD OUT 6](#_Toc47453741)

[5.1 Input 6](#_Toc47453742)

[5.2 Output 6](#_Toc47453743)

[5.3 Request 6](#_Toc47453744)

[5.4 Response 6](#_Toc47453745)

[5.5 Input 6](#_Toc47453755)

[5.6 OutPut 6](#_Toc47453756)

[5.7 Request 6](#_Toc47453757)

[5.8 Request 6](#_Toc47453758)

[5.9 Response 6](#_Toc47453759)

[5.10 Green Flow Enrollment 6](#_Toc47453760)

[5.10.1 Input 6](#_Toc47453761)

[5.10.2 Ouput 6](#_Toc47453762)

[5.10.3 API 6](#_Toc47453763)

[5.11 Yellow Flow Enrollment 6](#_Toc47453764)

[5.11.1 Input 6](#_Toc47453765)

[5.11.2 Ouput 6](#_Toc47453766)

[5.11.3 API 6](#_Toc47453767)

[5.12 Input 6](#_Toc47453768)

[5.13 Ouput 6](#_Toc47453769)

[5.14 Request 6](#_Toc47453770)

[5.15 ssResponse 6](#_Toc47453771)

[5.16 Input 6](#_Toc47453772)

[5.17 Ouput 6](#_Toc47453773)

[5.18 Request 6](#_Toc47453774)

[5.19 Response 6](#_Toc47453775)

[5.20 Input 6](#_Toc47453776)

[5.21 Ouput 6](#_Toc47453777)

[5.22 Request 6](#_Toc47453778)

[5.23 Response 6](#_Toc47453779)

# Revisions

|  |  |  |  |
| --- | --- | --- | --- |
| **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 AuthorizationAdd §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 |

# SANDBOX Simulator URL

The URL in Sandbox environment for SandboxSimulator API service is:

<https://rest-pp.s-money.fr/api/bib/sandbox/sandboxsimulator/api>

# SCT IN

## Input

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Example** | **Description** |
| UserId | Long | 123 | Internal User identification |
| OrderId | String | Moneyout-1 | Payment identifier in the third-party application |
| AccountId | SubAccountRef |  | Item containing user subaccountIf not indicated, the account used is the main sub-account of the connected user |
| ThirdPartyBIC | String |  | BIC reference |
| ThirdPartyIban | String |  | IBAN reference |
| ThirdPartyFullName | String |  | Name and surname of user bank account |
| Amount | Long | 2350 | Payment amount in euros (VAT included) |
| FeeAmountTTC | Long | 150 | Fee amount in euros (VAT not included) |
| FeeVAT | Decimal | 19.6 | VAT |
| FeeRecipientId | Long |  |  |
| ExecutionDate | Date | 2019-09-30 | Desired Execution date |
| Message | String | transfer S-money of 30/09/2019 | Payment message |
| Reference | String | REF123456  | Thirdparty-reference |
| Motif | String | Transfer S-money | Transfer Label. Will show up of the counterpart account statement |
| UniqueIdentification | String |  |  |
| Type | Int | 506 | Operation type. Possible values:506 SCT\_In |

## Output

Same data structure defined into input section.

## Request

**POST :**

**/api/sct/in/registration**

**Body example:**

{

 "Amount": 1000,

 "ExecutionDate": "2019-09-30T22:00:00.000Z",

 "UserId": 123,

 "ThirdPartyBIC": "xxxxx",

 "ThirdPartyIban": "xxxxxxxxxxxxxxxxxxxx",

 "ThirdPartyFullName": "xxxxxxxxxxx xxxxxxxxxxxx",

 "Type": 506,

 "OrderId":"moneyout-1",

 "Message": "transfer S-money of 09/07/2019",

 "Reference": "ref123456",

 "UniqueIdentification": "string",

 "Motif": " transfer S-money of xxxxx",

 "FeeAmountTTC": 180,

 "FeeVAT": 20,

 "FeeRecipientId": 0

}

## Response

{

 "Id": 147,

 "OperationId": 142,

 "CommissionId": 35,

 "Status": 1,

 "ValueDate": "2019-07-09T14:40:27.587",

 "Amount": 1000,

 "ExecutionDate": "2017-09-30T22:00:00",

 "UserId": 123,

 "DomainId": 12,

 "ThirdPartyBankAccountId": 50,

 "ThirdPartyBIC": null,

 "ThirdPartyIban": null,

 "ThirdPartyFullName": null,

 "Type": 506,

 "OrderId": "moneyout-1",

 "Message": "transfer S-money of 09/07/2019",

 "Reference": null,

 "UniqueIdentification": "string",

 "Motif": " transfer S-money of xxxxx",

 "FeeAmountTTC": 180,

 "FeeVAT": 20,

 "FeeRecipientId": 0

}

# SCT IN Recall

## Input

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Property** | **Type** | **Example** | **Description** | **Mandatory** |
| Id | Long | 123 | Transaction identification | Yes |
| DomainId | Int | 12 | Domain identification | Yes |
| Comment | String | Refund request for Invalid file format | Refund comment operation | No |
| RejectionCode | String | AC03 | SEPA Reason Code | Yes |
| RefundUniqueRef | String | refund-2 | OrderId of the refunded operation | No |

## SEPA reason code allowed

|  |  |
| --- | --- |
| **Value** | **Description** |
| AC03 | Wrong IBAN |
| AM09 | Wrong Amount |
| CUST | Customer decision |
| DUPL | Duplicate sending |
| FRAD | Fraudulent originated credit transfer |
| TECH | Technical problem |

## Output

Same data structure defined into SCT section.

## Request

**POST :**

**/api/sct/in/refund**

**Body example:**

{

 "Id": 147,

 "DomainId": 12,

 "Comment": "Refund for wrong IBAN",

 "RejectionCode": "AC03",

 "RefundUniqueRef": "Refund #147"

}

## Response

*Operation completed – SCT IN data*

{

 "Id": 147,

 "OperationId": 142,

 "CommissionId": 35,

 **"Status": 2,**

 "ValueDate": "2019-07-09T14:40:27.587",

 "Amount": 1000,

 "ExecutionDate": "2017-09-30T22:00:00",

 "UserId": 123,

 "DomainId": 12,

 "ThirdPartyBankAccountId": 50,

 "ThirdPartyBIC": null,

 "ThirdPartyIban": null,

 "ThirdPartyFullName": null,

 "Type": 506,

 "OrderId": "moneyout-1",

 "Message": "transfer S-money of 09/07/2019",

 "Reference": null,

 "UniqueIdentification": "string",

 "Motif": " transfer S-money of xxxxx",

 "FeeAmountTTC": 180,

 "FeeVAT": 20,

 "FeeRecipientId": 0

}

*Operation canceled – error message*

{

“Operation is already refund.“,

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

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

}

## Input

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Example** | **Description** |
| idTransaction | Long | 1234 | Id Transaction from espace partenaire |
| Operation | Int | 0,1 | 1: Validate transaction0: Refund transaction  |

## Request

**POST :**

**/api/CardAuthorization/ProcessFestopeme**

**Body example:**

{
    "IdTransaction" : 5138,
    "Operation" : 1,
}

## Response

*Operation completed*

{

 "Id": 4122,

 "Statut": " Completed "

}

*Operation refunded*

{

 "Id": 4122,

 "Statut": " Refunded "

}

## Input

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Example** | **Description** |
| idTransaction | Long | 1234 | Id Transaction from espace partenaire |
| Operation | Int | 0,1 | 1: Validate transaction0: Refund transaction  |
| ClearedAmount | Decimal | 1 | New amount |

## Request

**POST :**

**/api/CardAuthorization/ProcessFestopeme**

*(nb : same URL as for Process FestOpeme, but with input param Cleared Amount)*

**Body example:**

{
    "IdTransaction" : 5138,
    "Operation" : 1,
    "ClearedAmount" : 0.5
}

## Response

*Operation completed*

{

 "Id": 5138,

 "Statut": " Completed "

}

# SDD OUT

## Input

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Property** | **Type** | **Example** | **Description** | **Mandatory** |
| Amount | decimal | 123 | Operation Amount | Yes |
| DomainId | Int | 12 | Domain identification | Yes |
| UserId | long | 79 | The database Id of the user | Yes |
| ThirdPartyBankAccountId | long | 7 | The bankAccount Id that was already created | Yes |

## Output

Same data structure defined into SCT section.

## Request

**POST :**

**/api/sdd/out/registration**

**Body example:**

{

 "Amount": 20,

 "UserId": 79,

 "DomainId": 12,

 "ThirdPartyBankAccountId": 13

}

## Response

*Operation completed – SDD Out data*

{

 "CommissionId": 0,

 "Id": 30,

 "OperationId": 41,

 "Status": 1,

 "ValueDate": "2019-08-16T16:30:28.707",

 "Amount": 20,

 "ExecutionDate": "2019-08-16T14:30:23.783",

 "UserId": 79,

 "MandateId": 37,

 "Type": 509,

 "DomainId": 12,

 "ThirdPartyBankAccountId": 13,

 "ThirdPartyAccountIBAN": null,

 "ThirdPartyBankBIC": null,

 "ThirdPartyFullName": null,

 "OrderId": "ORDER-20190816-043023-782013",

 "Message": null,

 "Reference": null,

 "UniqueIdentification": null,

 "Motif": null,

 "FeeAmountTTC": 0,

 "FeeVAT": 0,

 "FeeRecipientId": 0

}

**Context :**

The aim of the DAC redressment is to simulate the automated distributor in gaz station, the holder will introduce his card to have an authorization of a fixed amount (150 example) then after refueling his car an authorization notice with the real amount will be sent.

* To simulate the authorization request, send an authorization request using the **authorization type 9 (cf. 3)**.
* To redress this estimated amount, send an authorization notice using the API call shown below.
1.
2.
3.
4.
5.
6.
7.
8.
9.

## Input

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Example** | **Description** |
| idTransaction | Long | 1234 | Transaction ID |
| Amount | Int | 10 | The **real** amount of the transaction (must be lower than initial amount) |

## OutPut

If no error :

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Example** | **Description** |
| Id | Long | 1234 |  |
| ActionCode | Int | 0 | Will always be 0, since  |

* Or Error message according to the table below:

|  |  |
| --- | --- |
| **Condition** | **Error Message** |
| Real amount > Estimated amount OrReal amount <= 0 | Transaction amount should be greater than zero and less or equal than the initial transaction. |
| Status != 0The status of the initial transaction is different from pending | The existing transaction is not in a pending status. |
| Initial transaction context <> 9(Initial transaction type must be 9, to allow DAC reversal) | The existing transaction is incompatible with an authorization notice modifying the amount. |
| The transaction id is incorrect | Transaction doesn't exist. |

## Request

**GET : /api/CardAuthorization/CardDacRedressement**

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

 {URL} api/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"

}

Functionality to delete a user from the system (valid into Sandbox environment only).

## Request

**DELETE :**

**/api/user/{AppUserId}**

## Response

|  |  |
| --- | --- |
| **HTTP code**  | **Description**  |
| 200 | User deleted |
| 500 | User not Found |

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

## Green Flow Enrollment

The green flow enrollment doesn’t require a verification.

The system will generate three authorizations:

* Authorization 1100
* Authorization notice 1120 to create the token (message code: 1400 [Token creation], token status: 5 [Inactivated])
* Authorization notice 1120 for device provisioning (message code: 1411 [Provisioning], Token status: 1 [Activated])

## Input

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

## Ouput

|  |  |  |  |
| --- | --- | --- | --- |
| **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: /api/CardAuthorization/ApplePayGreenFlow

Example: /api/CardAuthorization/ApplePayGreenFlow?appCardId=6Dt4io\_CP

Response:

{

    "Id": 6126,

    "ActionCode": 0

}

## Yellow Flow Enrollment

The yellow flow enrollment require a verification via OTP.

The system will generate three auhorizatoions:

* Authorization 1100
* Authorization notice 1120 to create the token (message code: 1400 [Token creation], token status: 5 [Inactivated])
* Authorization notice 1120 for device provisioning (message code: 1411 [Provisioning], Token status: 5 [Inactivated])
* Send pass code callback
* If OTP True: Authorization notice 1120 to activate the token (message code: 1412 [OTP Activation], token status: 1 [Activated]
* If OTP False: Authorization notice 1120 to keep token inactivated (message code: 1412 [OTP Activation], token status: 5 [Inactivated]

## Input

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

## Ouput

|  |  |  |  |
| --- | --- | --- | --- |
| **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: /api/CardAuthorization/ApplePayYellowFlow

Example :

Otp ok : /api/CardAuthorization/ApplePayYellowFlow?appCardId=CxfZjn\_CP&otpOk=True

Otp ko : /api/CardAuthorization/ApplePayYellowFlow?appCardId=CxfZjn\_CP&otpOk=false

Response:

{

    "Id": 6129,

    "ActionCode": 0

}

## 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) |

## Ouput

|  |  |  |  |
| --- | --- | --- | --- |
| **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:**

**api/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

}

## ssResponse

{

 " Id ": 7196

 "ActionCode": 0,

}

## 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 : Crédit2 : 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:**

**api/CardAuthorization/ProcessFestOpmCustom**

 **Body:**

{

 "AppCardId": "card001",

 "Amount": 200.00,

 "OperationDate": "2020-07-20T17:27:00",

 "Direction": 2,

 "MerchantName": "MARCHANDPF",

 "MerchantCategoryCode": "7011"

}

## Response

{

 " Id ": 7196

 "Statut":Completed,

}

## 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:**

**api/CardAuthorization/CardOpposition?appcardId=CARDTEST**

## Response

actionCode = 300