|  |
| --- |
| **API-3DS 2.0 – Top-up with Hosted-Fields** |
|  |  |  |  |
| **Version** | 1.8 |  |  |
| **Reference** | S-money-API- 3DS 2.0 (hosted fields) |
| **Objet** | Spécifications |  |

Contents

[1. Revisions 2](#_Toc55202202)

[2. Workflow 3](#_Toc55202203)

[3. Ressources 4](#_Toc55202204)

[3.1 Main fields 4](#_Toc55202205)

[3.2 Objects 4](#_Toc55202206)

[4. HostedFields integration 7](#_Toc55202207)

[5. Registration of a bank card 10](#_Toc55202208)

[5.1 Add a bank card 10](#_Toc55202209)

[5.2 Retrieve a bank card 11](#_Toc55202210)

[5.3 Retrieve all bank cards 12](#_Toc55202211)

[5.4 Delete a bank card 13](#_Toc55202212)

[6. Top-up 14](#_Toc55202213)

[6.1 Top-up with 3DS 2.0 (no card registration) 14](#_Toc55202214)

[6.2 Top-up with 3DS 2.0 with card registration 15](#_Toc55202215)

[6.3 Top-up without 3DS 2.0 (no card registration) 17](#_Toc55202216)

[6.4 Top-up OnECLICK with a 3DS 19](#_Toc55202217)

[6.5 Top-up ONECLICK without a 3DS 20](#_Toc55202218)

[7. Retrieve a top-up 22](#_Toc55202219)

[8. Retrieve all top-up 24](#_Toc55202220)

[9. Refund a top-up 26](#_Toc55202221)

[10. Callbacks 27](#_Toc55202222)

[11. Error List 28](#_Toc55202223)

# Revisions

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description** |
| 1.0 |  |  | 1st version of the doc for Bib. |
| 1.1 | 28/08/2020 | Alexis Coloneaux | Update Documentation |
| 1.2 | 10/09/2020 | Alexis Coloneaux | Add Top-up with a registered cardAdd GET Top-upAdd CallbacksAdd Errors list |
| 1.3 | 14/09/2020 | Alexis Coloneaux | Add GET All Top-ups |
| 1.4 | 21/10/2020 | Bilel Msekni | Add Delete card |
| 1.5 | 21/10/2020 | Alexis Coloneaux | Add Top-up Refund |
| 1.6 | 28/10/2020 | Alexis Coloneaux | Add GET Card list |
| 1.7 | 2/11/2020 | Bilel Msekni | Add 1.0 Api Gateway routes |
| 1.8 | 10/11/2020 | Njaka Raz | Add new new fields on POST Top-Up“TermsAndConditionsValidationDate” |
| 1.9 | 05/05/2023 | Michel Ruckebusch | Update sequence diagramAdded prod key values and JS UrlsUpdate URL versions v1.0 -> v1.1Update link in HTML form actionAdded field SubscriptionTopUpRemoved obsolete field *fee* |

# Workflow

Simple Payment – Hosted fields 3DSV2 challenge



# Ressources

## Main fields

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Input Mandatory** | **Description** |
| AppAccountId | String | YES | User’s ThirdParty app identifier |
| urlReturn | String | YES | URL of redirection. At the end of a 3DS validation, the user will be redirected on this URL. |
| Status | Int | OUTPUT ONLY | Status for card registration:0 = Waiting1 = Completed2 = FailedStatus for Top-up transaction:-1 = unknow0 = waiting1 = completed2 = refunded3 = refused4 = Waiting for validation5 = Cancelled6 = Waiting to be sent |
| Amount | Long | YES | Payment amount (in cents) |
| AuthenticationFormHtml | String | OUTPUT ONLY | Returned Authentication form (HTML form encoded in base64) |
| OrderId | String | YES | Top-up identifier in the third-party application |
| Id | Long | OUTPUT ONLY | Top-up ID |
| Message | String | YES | Top-up message |
| ErrorCode | Int | OUTPUT ONLY | 0 = 0XXX operation succeeded or still processing10 = 1XXX operation rejected because of bad request configuration11 = 2XXX operation rejected because of bad reference usage12 = 3XXX operation rejected because of bad account configuration13 = 4XXX operation rejected because of the bank or the supplier14 = 5XXX operation rejected because of a system error15 = 6XXX operation rejected because of anti-fraud engine |
| TermsAndConditionsValidationDate | DateTime | Depends on partner parameter | Terms and conditions validation dateThis field is optional if the partner parameter is disablee.g: 2020-11-10T11:16:29.054+01:00 |
| SubscriptionTopUp | Boolean | NO | Only applicable to Top-up Oneclick. * If false (default value): indicates that the transaction is CIT (client initiated)
* If true: indicates that the transaction is MIT (merchant initiated)
 |

## Objects

#### AccountRef

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Input Mandatory** | **Description** |
| Id | Long | OUTPUT ONLY | Account identifier. |
| AppAccountId | String | Yes | Account identifier provided by the third-party application. |
| DisplayName | Long | OUTPUT ONLY | Account display name |
| Email | String | OUTPUT ONLY | Account email |
| Phonenumber | String | OUTPUT ONLY | Account phone number |
| Href | Uri | OUTPUT ONLY | Uri for the account details |

#### Card

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Input Mandatory** | **Description** |
| Id | Long | OUTPUT ONLY | Card ID |
| AppAccountId | String | Yes | AppAccountId of the card user. |
| AppCardId | String | NO | Card identifier in the third-party application. |
| Network | Long | OUTPUT ONLY | UNKNOWN = -1,CB = 1,MASTERCARD = 2,VISA = 3, |
| Hint | String | OUTPUT ONLY | Hide the card number. 6XXXX4 format |
| Country | String | OUTPUT ONLY | ISO country code of the card |
| ExpiryDate | Date | OUTPUT ONLY | Card expiration date |
| Status | Int | OUTPUT ONLY | Status value available :  Pending = 0, Succeeded = 1, Failed = 2 |

#### Holder/PayerInfo

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Input Mandatory** | **Description** |
| Name | String | YES | The holder’s full name (as described on the payment method). |
| Mail | String | YES | The user’s email. |
| UserAgent | String | YES | The user’s browser informations |
| UrlReferrer | String | YES | Url  |
| IpAddress | String | YES | The User’s IP Address |

#### Extra Result

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Input Mandatory** | **Description** |
| threedsecure | Boolean | OUTPUT ONLY | True / False |
| threedsecureauthstatus | Int | OUTPUT ONLY | 0 = Yes1 = No2 = unavaible3 = Attempted |
| threedsecuresignaturestatus | Int | OUTPUT ONLY | 0 = Yes1 = No |
| threedsecureglobalstatus | Int | OUTPUT ONLY | 0 = Yes1 = Not\_enrolled2 = unavaible3 = Not\_required |
| ThreeDsCardEnrolled | Int | OUTPUT ONLY | 0 = Yes1 = No2 = unavaible |
| ExecCode | String (4) | OUTPUT ONLY | The operation result code |
| ExecMessage | String | OUTPUT ONLY | The operation result description linked to EXECCODE |
| warrantyresult | String | OUTPUT ONLY | Yes / no |
| 3dsecureversion | Int | OUTPUT ONLY | The kind of 3-D Secure protocol version.Ex: 1 |
| bankauthresult | Int | OUTPUT ONLY | Return code of the authorisation request returned by the issuing bank |

# HostedFields integration

**Principle**:

1. You display a payment page including a registration form, except that the bankcard input fields are replaced by hosted-fields containers. These containers could be any HTML tag: div, p, span… the hosted-fields JavaScript library injects Dalenys-hosted iframes in these containers, each containing the card data input fields
2. At the submit process, you should call the createToken method of the hosted-fields library which will trigger the tokenization of the cardholder data (card number, expiry date and cryptogram)
3. If the tokenization is successful, you must add the received token to your form submission request (e.g. by adding a hidden input)

**Creation of the form**:

You must own a TLS certificate to host a valid HTTPS payment page, otherwise the user’s browser will display security alerts and is likely to block it.

The hosted-fields library must always be called online. Using a downloaded version hosted on your own server can cause serious malfunctions, especially in the case of an update of the API.

1. **First, you must include the Dalenys hosted fields dedicated library**, by adding the following code between the <head> and </head> tags in your HTML:
* <script type="text/javascript" src="https://js.**sandbox**.dalenys.com/hosted-fields/v2.0.0/hosted-fields.min.js" />
* <script type="text/javascript" src="https://js.**sandbox**.dalenys.com/hosted-fields/v2.0.0/brand-selector-widget.min.js" />

Please note that these URLs are different in production environment:

* <script type="text/javascript" src="https://js.dalenys.com/hosted-fields/v2.0.0/hosted-fields.min.js" />
* <script type="text/javascript" src="https://js.dalenys.com/hosted-fields/v2.0.0/brand-selector-widget.min.js" />
1. **Declare the UTF-8 encoding**

<meta charset="UTF-8">

1. **Create a <form> </ form> form with 4 containers identified by an id attribute.** The choice of the identifier used is free but must be unique and will be used in the javascript code presented in the following points.

In the following example, we are generating a payment form using 4 span tags as containers:

<form method="post" action="URL">

<span id="card-container"></span>

<span id="expiry-container"></span>

<span id="cvv-container"></span>

<span id="brand-container"></span>

<input type="submit" value="Pay">

</form>

The URL to enter in the action attribute will the URL specify on the documation below (See request of Card registration and Top-up).

1. **Configure the hosted fields library**

The next step is to designate the containers in which to create the secured hosted fields, using JavaScript.

To do so, initialize the hosted-fields library with your Public API Key and your desired configuration: S-money must provide the necessary keys to the hosted fields.

<script type="text/javascript">

 // Initialize the hosted-fields library

 var hfields = be2bill.hostedFields({

 // Use the Public API Key

 // In sandbox environment:

 key: {

 id: "5f9e0aff-b16e-4e46-9c8c-4d9e6380a6ce",

 value: "@,S#)8Nm?gs39r#I"

 },

 // In production environment:

 //key: {

 // id: "9f77c485-1594-49eb-a288-ebd25a2d9f93",

 // value: "NS,zWQ+=@p+Txknq"

 // },

 // Link and configure each hosted input field by providing the corresponding container ID

 fields: {

 'card': {

 id: 'card-container',

 //enable the card field format auto spacing

 enableAutospacing: true

 },

 'expiry': {

 id: 'expiry-container'

 },

 'cryptogram': {

 id: 'cvv-container'

 },

 'brand': {

 id: 'brand-container'

 }

 },

 // Choose the language for error messages

 location: "fr"

 });

</script>

The identifiers used in the script to designate the fields are those used for the identifiers of the containers created in the form. This allow the script to inject the fields hosted by Dalenys in these containers.

1. **Load the hosted fields library**

<script type="text/javascript">

 hfields.load();

</script>

At this point, your web browser may display the bank card input fields into their containers.

1. **Token generation**

Once the user submits the form, you should call the createToken method to trigger the tokenization process.

This method expects a callback in parameter which will be triggered once the tokenization request is finished.

Your callback will receive a result object containing these properties:

**execCode**: The technical result code of the tokenization call. In case of success, 0000 will be sent

**message**: The description linked to the execCode

**cardType**: The card type

**cardCode**: The first 6 and last 4 digits of the cardholder’s card number

**cardValidityDate**: Card expiry date

**selectedBrand**: Cardholder selected brand in case of co-branded card.

**hfToken**: Of course the generated token

In case of success you have to add the received token to the form submit request (by adding an hidden input for example) :

<form method="post" name="formName" action="https://partherBackEnd/initiatePayment" **onsubmit="return tokenizeHandler()"**>

 <!-- ... -->

 <input type="hidden" name="hftoken" id="hftoken">

 <!-- ... -->

</form>

<script type="text/javascript">

 function tokenizeHandler() {

 hfields.createToken(function (result) {

 //console.log(result); // Debug

 if (result.execCode == '0000') {

 // Set the token in an hidden input field to transmit it to the merchant

 document.getElementById('hftoken').value = result.hfToken;

 // Send the form request

 document.formName.submit();

 }

 });

 // Prevents the submit of the form in case of failed tokenization request

 return false;

 }

</script>

#### Registration with 3d secure Authentication

Depending on the card used during the payment, the submission of the registration form, a 3DSecure authentication may be necessary.

In this case, the encoded base-64 HTML page of the 3DSecure form is sent in response content with a http status code 200 (OK).

The submission of the 3DSecure form will end with a 302 redirection to the url entered in the UrlReturn field of the initial request.

# Registration of a bank card

## Add a bank card

#### With A 3DS Required Card

This part is about the card registration with a card who required a 3DS form validation (asked by the end- user bank). Once the request sends to S-money, you must decode the base-64 html form to redirect the end- user. If the 3DS form is correctly validated, the end-user will be redirect to the “UrlReturn”.

NOTE: If the end-user fills the 3DS form and is redirected to the UrlReturn it does not mean that the Card is available in S-money system. The end-user must wait for the notification send by Dalenys. This notification will trigger the validation of the registered card.

Headers :

POST /api/v1.1/cards/registrations

**Accept** : application/vnd.s-money.v4+json

**Content-Type** : application/vnd.s-money.v4+json

**Authorization** : Bearer ACCESS\_TOKEN

Body :

**Request with a 3DS required Card:**

{

    "UrlReturn": "http://myReturnUrl.com/",

    "Holder": {

        "Name": "test-stub",

        "Email": "test-stub@gmail.com",

        "UserAgent": "Mozilla/5.0 CK={} (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko",

        "UrlReferrer": "https://www.s-money.fr?cart=473",

        "IpAddress": "127.0.0.1"

    },

    "Card": {

        "HFToken": "fdc57a40-156b-4c0f-847d-52d7229b669f",

        "AppCardId": "gbthg",

    },

    "AppAccountId": "alice"

}

**Réponse :**

Headers :

HTTP/1.1 200 OK

Body :

{

    "UrlReturn": "http://myReturnUrl.com/",

    "AuthenticationFormHtml": "PGh0bWw+CjxoZWFkPgogICAgPG1ldGEgY2hhcnNldD0idXRmLTgiIC8+CiAgICA8dGl0bGU+IlByb2Nlc3NpbmcgcGF5bWVudCAuLi4iPC90aXRsZT4KICAgIDxsaW5rIHJlbD0ic3R5bGVzaGVldCIgaHJlZj0iaHR0cHM6Ly9zZWN1cmUtdGVzdC5iZTJiaWxsLmNvbS9mc[……]+CjwvaHRtbD4=",

    "Holder": {

        "Name": "test-stub",

        "Email": "test-stub@gmail.com”

    },

    "Card": {

        "AppCardId": "myCard",

        "Status": "0",

        "Id": "6153"

    }

}

## Retrieve a bank card

**Request :**

GET /api/v1.1/users/{appuserid}/cards/registered/{appcardid}

**Accept** : application/vnd.s-money.v4+json

**Content-Type** : application/vnd.s-money.v4+json

**Authorization** : Bearer ACCESS\_TOKEN

**Réponse :**

Header :

HTTP/1.1 200 OK

Body :

{

        "Id": 8833,

        "AppCardId": "TEST-0",

        "Network": "CB",

        "Hint": "423460XXXXXX0000",

        "Country": "FR",

        "ExpiryDate": "01-2023",

        "Status": "1",

        "ExtraResult": {

            "TransactionId": "A14786733",

            "ExecCode": "0000",

            "Message": "Operation+succeeded.",

            "Currency": "EUR",

            "ThreeDSecure": "YES",

            "ThreeDSAuthenticationStatus": "U",

            "AuthorizationNumber": "504063",

            "CardNetwork": "CB",

            "CardValidityDate": "01-23",

            "CardCountry": "FR",

            "CardFullname": "Alice Dylan",

            "CardThreeDSecureEnrolled": "N",

            "ThreeDSecureVersion": 1,

            "ThreeDSGlobalStatus": "NOT\_ENROLLED",

            "Version": null,

            "CardType": "VISA"

        }

    }

## Retrieve all bank cards

**Request :**

GET /api/v1.1/users/{userid}/cards/registered

**Accept** : application/vnd.s-money.v4+json

**Content-Type** : application/vnd.s-money.v4+json

**Authorization** : Bearer ACCESS\_TOKEN

**Réponse :**

Header :

HTTP/1.1 200 OK

Body :

[

 {

        "Id": 8833,

        "AppCardId": "azfsfdrh",

        "Network": "CB",

        "Hint": "423460XXXXXX0000",

        "Country": "FR",

        "ExpiryDate": "01-2023",

        "Status": "1",

        "ExtraResult": {

            "TransactionId": "A14786733",

            "ExecCode": "0000",

            "Message": "Operation+succeeded.",

            "Currency": "EUR",

            "ThreeDSecure": "YES",

            "ThreeDSAuthenticationStatus": "U",

            "AuthorizationNumber": "504063",

            "CardNetwork": "CB",

            "CardValidityDate": "01-23",

            "CardCountry": "FR",

            "CardFullname": "test-stub",

            "CardThreeDSecureEnrolled": "N",

            "ThreeDSecureVersion": 1,

            "ThreeDSGlobalStatus": "NOT\_ENROLLED",

            "Version": null,

            "CardType": "VISA"

        }

    },

 ………

]

## Delete a bank card

**Request :**

DELETE /api/v1.1/users/{userid}/cards/{appcardid}

**Accept** : application/vnd.s-money.v2+json

**Content-Type** : application/vnd.s-money.v2+json²

**Authorization** : Bearer ACCESS\_TOKEN

**Réponse :**

Header :

HTTP/1.1 204 NoContent

# Top-up

## Top-up with 3DS 2.0 (no card registration)

POST /api/v1.1/payins/cardpayments

Headers :

POST /api/v1.1/payins/cardpayments

**Accept** : application/vnd.s-money.v4+json

**Content-Type** : application/vnd.s-money.v4+json

**Authorization** : Bearer ACCESS\_TOKEN

Body:

{

    "OrderId": "v4jnw7",

    "UrlReturn": "http://myUrlReturn.com",

    "Payer": {

        "Name": "Alice",

        "Email": "Alice@gmail.com",

        "UserAgent": "Mozilla/5.0 CK={} (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko",

        "UrlReferrer": "https://www.myUrlReferrer.fr?cart=473",

        "IpAddress": "127.0.0.1"

    },

    "Card": {

        "HFToken": "5cc47973-5da9-45f8-bd08-20a502b23eaa"

    },

    "Payments": [

        {

            "OrderId": "v4jnw7-1",

            "Amount": 10,

            "Beneficiary": {

                "AppAccountId": "alexis-part"

            },

            "Message": “My topup”

        }

    ],

  "TermsAndConditionsValidationDate": "2020-11-10T11:16:29.054+01:00"

}

Body :

HTTP/1.1 201 CREATED

Header :

{

    "OrderId": "v4jnw7",

    "Status": "0",

    "ExecMessage": "3D secure authentication required.",

    "ExecCode": "0001",

    "AuthenticationFormHtml": "PGh0bWw+CjxoZWFkPgogICAgPG1ldGEgY2hhcnNldD0idXRmLTgiIC8+CiAgICA8dGl0bGU+IlByb2Nlc3NpbmcgcGF5bWVudCAuLi4iPC90aXRsZT4KICAgIDxsaW5rIHJlbD0ic3R5bGVzaGEM5MGNtRnVjMkZqZEdsdmJuTmNMMjFsZEdodlpGOXViM1JwWm1sallYUnBiMjRpTENKMGFISmxaVVJUVTJWeWRtVnlWSEpoYm5OSlJDSTZJakZtTmpVek5EVTBMVGd4T0RFdE5HSmlOaTFpWlRRM0xXUmhNVEF4T1RObFlUYzNaQ0o5IiAvPgogICAgICAgIDxpbnB1dCB0eXBlPSJoaWRkZW4iIG5hbWU9ImxvZ19pZCIgdmFsdWU9IjVmNGUxMjJjNzU0MmExLjE0ODcwODI2IiAvPgogICAgPC9mb3JtPgo8L2JvZHk+CjwvaHRtbD4=",

    "Payer": {

        "Name": "alice",

        "Email": " alice@gmail.com”

    },

    "Payments": [

        {

            "OrderId": "v4jnw7-1",

            "Amount": 10,

            "Beneficiary": {

                "AppAccountId": "Alice"

            },

            "Message": ”My topup”,

            "Status": "0"

        }

    ],

  "TermsAndConditionsValidationDate": "2020-11-10T11:16:29.054+01:00"

}

## Top-up with 3DS 2.0 with card registration

**Request**

Headers :

POST /api/v1.1/payins/cardpayments

**Accept** : application/vnd.s-money.v4+json

**Content-Type** : application/vnd.s-money.v4+json

**Authorization** : Bearer ACCESS\_TOKEN

Body:

{

    "OrderId": "ki053k",

    "UrlReturn": "http://myUrlReturn.com",

    "Payer": {

        "Name": "Alice",

        "Email": "Alice@gmail.com",

        "UserAgent": "Mozilla/5.0 CK={} (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko",

        "UrlReferrer": "https://www.MyUrlReferrer.fr?cart=473",

        "IpAddress": "127.0.0.1"

    },

    "Card": {

        "HFToken": "a94b7895-af67-4900-b1ff-eeb839c3efb5",

        "AppCardId": "vzsrgvsfg"

    },

    "Payments": [

        {

            "OrderId": "ki053k-1",

            "Amount": 10,

            "Beneficiary": {

                "AppAccountId": "alexis-part"

            },

            "Message": “My topup with card registration”

        }

    ],

  "TermsAndConditionsValidationDate": "2020-11-10T11:16:29.054+01:00"

}

**Response**

Body :

HTTP/1.1 201 CREATED

Header :

{

    "OrderId": "ki053k",

    "Status": "0",

    "ExecMessage": "3D secure authentication required.",

    "ExecCode": "0001",

    "AuthenticationFormHtml": "PGh0bWw+CjxoZWFkPgogICAgPG1ldGEgY2hhcnNldD0idXRmLTgiIC8+CiAgICA8dGl0bGU+IlByb2Nlc3NpbmcgcGF5bWVudCAuLi4iPC90aXRsZT4KICAgIDxsaW5rIHJlbD0ic3R5bGVzaGVldCIgaHJlZj0iaHR0cHM6Ly9zZWN1cmUtdGVzdC5iZTJiaWxsLmNvbS9mcm9udC8vcmVkaXJlY3QvY3NzL2xvYWRpbmctcGFnZS5jc3MiLz4KICAgIDxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2NyaXB0Ij4KICAgIHZhcidlpGOXViM1JwWm1sallYUnBiMjRpTENKMGFISmxaVVJUVTJWeWRtVnlWSEpoYm5OSlJDSTZJall5TVdSbU4yUmpMVEptWlRndE5HUTROaTA1TnpnekxXSXdNbUUzTVRFeU1UTXdOQ0o5IiAvPgogICAgICAgIDxpbnB1dCB0eXBlPSJoaWRkZW4iIG5hbWU9ImxvZ19pZCIgdmFsdWU9IjVmNGUxNzRhODhjODgxLjIwNDU2Mjk1IiAvPgogICAgPC9mb3JtPgo8L2JvZHk+CjwvaHRtbD4=",

    "Payer": {

        "Name": "test-stub",

        "Email": "test-stub@gmail.com"

    },

    "Card": {

        "AppCardId": "vzsrgvsfg",

        "Status": 0

    },

    "Payments": [

        {

            "OrderId": "ki053k-1",

            "Amount": 10,

            "Beneficiary": {

                "AppAccountId": "Alice"

            },

            "Message": “My topup with card registration”,

            "Status": "0"

        }

    ],

  "TermsAndConditionsValidationDate": "2020-11-10T11:16:29.054+01:00"

}

## Top-up without 3DS 2.0 (no card registration)

**Request**

Header:

POST /api/v1.1/payins/cardpayments

**Accept** : application/vnd.s-money.v4+json

**Content-Type** : application/vnd.s-money.v4+json

**Authorization** : Bearer ACCESS\_TOKEN

Body:

{

    "OrderId": "xumvtw",

    "UrlReturn": "http://myUrlReturn.com/ ",

    "Payer": {

        "Name": "Alice",

        "Email": "alice@gmail.com",

        "UserAgent": "Mozilla/5.0 CK={} (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko",

        "UrlReferrer": "https://www.s-money.fr?cart=473",

        "IpAddress": "127.0.0.1"

    },

    "Card": {

        "HFToken": "92cac5c1-c793-4a41-a862-e018ce8dc07b"

    },

    "Payments": [

        {

            "OrderId": "xumvtw-1",

            "Amount": 10,

            "Beneficiary": {

                "AppAccountId": "alice"

            },

            "Message": ”My topup with card no 3DS”

        }

    ],

  "TermsAndConditionsValidationDate": "2020-11-10T11:16:29.054+01:00"

}

**Response**

Body :

HTTP/1.1 201 CREATED

Header :

{

    "OrderId": "uy5wiy",

    "Status": "1",

    "ExecMessage": "Operation succeeded.",

    "ExecCode": "0000",

    "Payer": {

        "Name": "alice",

        "Email": "alice@gmail.com",

        "UserAgent": null,

        "UrlReferrer": null,

        "IpAddress": null

    },

    "Payments": [

        {

            "OrderId": "uy5wiy-1",

            "Amount": 0.1,

            "Beneficiary": {

                "AppAccountId": "alice"

            },

            "Message": ”My topup with card no 3DS”,

            "Status": "1"

        }

    ],

  "TermsAndConditionsValidationDate": "2020-11-10T11:16:29.054+01:00"

}

## Top-up ONECLICK with a 3DS

**Request**:

Headers :

POST /api/v1.1/payins/cardpayments

**Accept** : application/vnd.s-money.v4+json

**Content-Type** : application/vnd.s-money.v4+json

**Authorization** : Bearer ACCESS\_TOKEN

Body :

{

    "OrderId": "bocbuh",

    "UrlReturn": "http://myUrlReturn.com",

    "Payer": {

        "Name": "alice",

        "Email": "alice@gmail.com",

        "UserAgent": "Mozilla/5.0 CK={} (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko",

        "UrlReferrer": "https://www.s-money.fr?cart=473",

        "IpAddress": "127.0.0.1"

    },

    "Card": {

        "AppCardId": "jtyuj",

    },

    "Payments": [

        {

            "OrderId": "bocbuh-1",

            "Amount": 10,

            "Beneficiary": {

                "AppAccountId": "alice”

            },

            "Message": null

        }

    ],

  "TermsAndConditionsValidationDate": "2020-11-10T11:16:29.054+01:00"

}

**Response :**

Header:

HTTP/1.1 201 CREATED

Body :

{

    "OrderId": "bocbuh",

    "Status": "0",

    "ExecMessage": "3D secure authentication required.",

    "ExecCode": "0001",

    "AuthenticationFormHtml": "PGh0bWw+CjxoZWFkPgogICAgPG1ldGEgY2hhcnNldD0idXRmLTgiIC8+CiAgICA8dGl0bGU+IlByb2Nlc3NpbmcgcGF5bWVudCAuLi4iPC90aXRsZT4KICAgIDxsaW5rIHJlbD0ic3R5bGVzaGVldCIgaHJlZj0iaHR0cHM6Ly9zZWN1cmUtdGVzdC5iZTJiaWxsLmNvbS9mcm9udC8vcmVkaXJlY3QvY3NzL2xvYWRpbmctcGFnZS5jc3MiLz4KICAgIDxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2NyaXB0Ij4KICAgIHZhciB0YXJnZXQgPSAibWFpbiI7CiAgdSelpXTjFjbVV0Wm5KdmJuUmNMM1l5WEM5MGNtRnVjMkZqZEdsdmJuTmNMMjFsZEdodlpGOXViM1JwWm1sallYUnBiMjRpTENKMGFISmxaVVJUVTJWeWRtVnlWSEpoYm5OSlJDSTZJalkyTkRVeU16VTJMVGN5TTJZdE5HTmtZUzA0TW1JeExUUTNNelU0WlRNMU1HTXlaaUo5IiAvPgogICAgICAgIDxpbnB1dCB0eXBlPSJoaWRkZW4iIG5hbWU9ImxvZ19pZCIgdmFsdWU9IjVmNWE0N2RjYTE3MDQzLjk0ODc0NzI2IiAvPgogICAgPC9mb3JtPgo8L2JvZHk+CjwvaHRtbD4=",

    "Payer": {

        "Name": "test-stub",

        "Email": "test-stub@gmail.com",

        "UserAgent": null,

        "UrlReferrer": null,

        "IpAddress": null

    },

    "Payments": [

        {

            "OrderId": "bocbuh-1",

            "Amount": 0.1,

            "Beneficiary": {

                "AppAccountId": "alexis-part"

            },

            "Message": null,

            "Status": "0"

        }

    ],

  "TermsAndConditionsValidationDate": "2020-11-10T11:16:29.054+01:00"

}

## Top-up ONECLICK without a 3DS

**Request :**

Headers :

POST /api/v1.1/ payins/cardpayments

**Accept** : application/vnd.s-money.v4+json

**Content-Type** : application/vnd.s-money.v4+json

**Authorization** : Bearer ACCESS\_TOKEN

Body:

{

    "OrderId": "aomhle",

    "UrlReturn": "http://myUrlReturn.com",

    "Payer": {

        "Name": "alice",

        "Email": "alice@gmail.com",

        "UserAgent": "Mozilla/5.0 CK={} (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko",

        "UrlReferrer": "https://www.s-money.fr?cart=473",

        "IpAddress": "127.0.0.1"

    },

    "Card": {

        "AppCardId": "jtyuj",

    },

    "Payments": [

        {

            "OrderId": "aomhle-1",

            "Amount": 10,

            "Beneficiary": {

                "AppAccountId": "alexis-part"

            },

            "Message": null

        }

    ],

  "TermsAndConditionsValidationDate": "2020-11-10T11:16:29.054+01:00",

 **"SubscriptionTopUp": true**

}

Response :

Header:

HTTP/1.1 201 CREATED

Body :

{

    "OrderId": "aomhle",

    "Status": "1",

    "ExecMessage": "Operation succeeded.",

    "ExecCode": "0000",

    "Payer": {

        "Name": "alice",

        "Email": "alice@gmail.com"

    },

    "Payments": [

        {

            "OrderId": "aomhle-1",

            "Amount": 0.1,

            "Beneficiary": {

                "AppAccountId": "alice"

            },

            "Message": null,

            "Status": "1"

        }

    ],

  "TermsAndConditionsValidationDate": "2020-11-10T11:16:29.054+01:00"

}

# Retrieve a top-up

**Request**

Headers :

GET /api/v1.1/users/{userid}/payins/cardpayments/{orderid}

**Accept** : application/vnd.s-money.v4+json

**Content-Type** : application/vnd.s-money.v4+json

**Authorization** : Bearer ACCESS\_TOKEN

**Response :**

HTTP/1.1 200 OK

{

    "Message": null,

    "OrderId": "307iid",

    "Status": "1",

    "Payer": {

        "Name": "alice",

        "Email": "alice@gmail.com"

    },

    "Card": {

        "AppCardId": "etyhetyh",

        "Status": "1",

        "Hint": "446421XXXXXX0001",

        "ExpiryDate": "12-2022",

        "IsActive": true

    },

    "Payments": [

        {

            "OrderId": "307iid-1",

            "Amount": 0.10,

            "Beneficiary": {

                "AppAccountId": "alice"

            },

            "Message": null,

            "Status": "1"

        }

    ],

  "TermsAndConditionsValidationDate": "2020-11-10T11:16:29.054+01:00",

    "ExtraResult": {

        "TransactionId": "A14453350",

        "ExecCode": "0000",

        "Message": "Operation succeeded.",

        "Currency": "EUR",

        "ThreeDSecure": "YES",

        "ThreeDSAuthenticationStatus": null,

        "AuthorizationNumber": "446284",

        "CardNetwork": "VISA",

        "CardValidityDate": "12-22",

        "CardCountry": "FR",

        "CardFullname": "alice",

        "CardThreeDSecureEnrolled": "Y",

        "ThreeDSecureVersion": 2,

        "ThreeDSGlobalStatus": "OK",

        "Version": "3.0",

        "CardType": "VISA"

    },

 "Refunds": [

        {

            "Id": 10789,

            "Amount": 0.10,

            "OrderId": "refund-1",

            "Status": "1"

        }

    ]

}

# Retrieve all top-up

**Request**

Headers :

GET /api/v1.1/users/{userid}/payins/cardpayments

**Accept** : application/vnd.s-money.v4+json

**Content-Type** : application/vnd.s-money.v4+json

**Authorization** : Bearer ACCESS\_TOKEN

**Response**

HTTP/1.1 200 OK

[

    {

        "Message": null,

        "OrderId": "307iid",

        "Status": "1",

        "Payer": {

            "Name": "alice",

            "Email": "alice@gmail.com"

        },

        "Card": {

            "AppCardId": "etyhetyh",

            "Status": "1",

            "Hint": "446421XXXXXX0001",

            "ExpiryDate": "12-2022",

            "IsActive": true

        },

        "Payments": [

            {

                "OrderId": "307iid-1",

                "Amount": 0.10,

                "Beneficiary": {

                    "AppAccountId": "alice"

                },

                "Message": null,

                "Status": "1"

            }

        ],

  "TermsAndConditionsValidationDate": "2020-11-10T11:16:29.054+01:00",

        "ExtraResult": {

            "TransactionId": "A14453350",

            "ExecCode": "0000",

            "Message": "Operation succeeded.",

            "Currency": "EUR",

            "ThreeDSecure": "YES",

            "ThreeDSAuthenticationStatus": null,

            "AuthorizationNumber": "446284",

            "CardNetwork": "VISA",

            "CardValidityDate": "12-22",

            "CardCountry": "FR",

            "CardFullname": "alice",

            "CardThreeDSecureEnrolled": "Y",

            "ThreeDSecureVersion": 2,

            "ThreeDSGlobalStatus": "OK",

            "Version": "3.0",

            "CardType": "VISA"

        },

 "Refunds": [

        {

            "Id": 10789,

            "Amount": 0.10,

            "OrderId": "refund-1",

            "Status": "1"

        }

     ]

    }

    ...

    ...

]

# Refund a top-up

**Request**

Headers :

POST /api/v1.1/users/{userid}/payins/cardpayments/{id}/payments/{paymentId}/refunds

**Accept** : application/vnd.s-money.v1+json

**Content-Type** : application/vnd.s-money.v1+json

**Authorization** : Bearer ACCESS\_TOKEN

Body :

{

        "orderid" : "refunds-order",

        "amount" : 10

}

**Response**

HTTP/1.1 201 CREATED

{

    "Id": 1343,

    "OrderId": "refunds-orderid",

    "Amount": 0.10,

    "Status": 1,

    "PaymentDate": "2020-10-20T15:52:06.1938879+02:00",

    "OriginalPayment": {

        "Id": 1332,

        "OrderId": "weytqd-1",

        "Href": "http://my-partner-space//payins/weytqd-1"

    },

    "Type": 1

}

# Callbacks

Lists of operations triggering a callback :

* Bank card validation (Type 5)
* Top-up validation (Type 1)
* Top-up refund (Type 37)

Examples :

POST [https://youCallbackUrl.com/?type=5&id=[AppCardId]&userid=[Beneficiary](https://youCallbackUrl.com/?type=5&id=%5bAppCardId%5d&userid=%5bBeneficiary)]

POST [https://youCallbackUrl.com/?type=1&id=[OrderId]&error=0&status=1](https://youCallbackUrl.com/?type=1&id=%5bOrderId%5d&error=0&status=1)

# Error List

|  |  |  |
| --- | --- | --- |
| **HTTP Code** | **Error Code** | **Details** |
| 401 | 362 | Opération non autorisée |
| 401 | 569 | Token oauth expiré |
| 401 | 570 | Token oauth invalide |
| 400 | 704 | Paramètre d’appel manquant |
| 400 | 177 | Paramètre d’appel invalide |
| 404 | 147 | Compte introuvable |
| 400 | 500 | Le compte de l’envoyeur est en opposition |
| 400 | 111 | Statut de l’envoyeur invalide |
| 400 | 127 | L’envoyeur et le bénéficiaire ne peuvent pas être le même compte |
| 400 | 345 | Compte du bénéficiaire incorrect ou incomplet |
| 400 | 110 | Solde insuffisant pour effectuer l’opération |
| 400 | 701 | Plafond de transaction atteint (client non bancarisé) |
| 400 | 702 | Plafond de transaction atteint (client KYC) |
| 400 | 703 | Plafond de transaction atteint (client non KYC) |
| 400 | 177 | Valeur de paramètre invalide (orderId déjà existant) |
| 400 | 177 | Valeur de paramètre invalide (Tous les paramètres orderId doivent être unique) |
| 400 | 195 | Carte introuvable |
| 400 | 356 | Opération introuvable |
| 400 | 354 | Psp Invalide |