Handbook

pushTAN 2.0 (pushTAN decoupled)

Sparkassen-Finanzgruppe

finanz informatik

Contents

1 | Changelog

Version	Date	Revision
0.1	14.09.2020	Initial version
0.2	22.10.2020	Internal review
1.0	28.10.2020	Internal review and approval
1.1	23.11.2020	Update / clarification to read status
1.2	14.12.2020	Update/ clarification not supported APP versions
1.3	14.01.2021	Scheduling concretized

2 | Management Summary

The pushTAN procedure is being adapted so that when approving an order, end customers will no longer be shown a TAN in the pushTAN app that has to be entered manually in the banking application (or third-party application). Instead, the order can be approved directly in the pushTAN app with a button.

This new pushTAN variant is called "**pushTAN 2.0**".

This so-called DECOUPLED approach is only offered for TAN entries.

2.1 | Target Audience for this Document

The target audience for this document is all third-party service providers who make use of the **embedded SCA approach**. The changes described here only affect the authenticationType PUSH_OTP (**pushTAN**).

There are no changes to the authenticationType CHIP_OTP and SMS_OTP. The same applies to all third-party service providers who use the REDIRECT SCA approach.

2.2 | Schedule

The sandbox has been available for tests since November 18, 2020. The real interface will support pushTAN 2.0 for all institutes from March 9th, 2021.

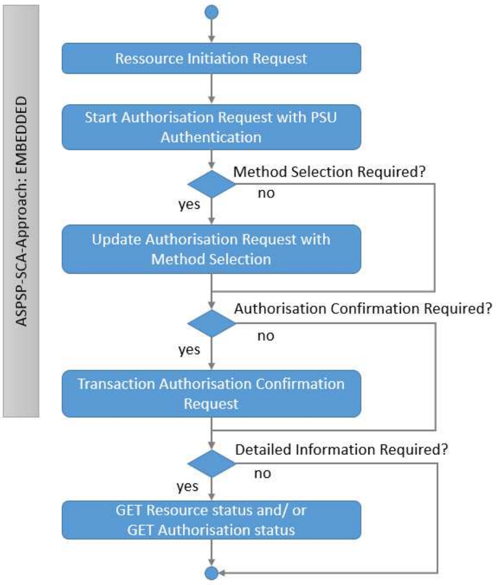
The activation of the pushTAN variant "pushTAN 2.0" (pushTAN decoupled) takes place independently of the software change in the XS2A-API by each individual institute. For the XS2A interface, this has the effect that only a small amount of customers will access the interface with the pushTAN variant 2.0.

The area rollout is planned for the second quarter of 2021.

We therefore ask you to make the minimum adjustment by the end of March 2021 in the interests of a positive end-user experience. We recommend that you support the DECOUPLED approach by the end of July 2021.

3 | Changes to the Request Flow

Up to now you have initiated a resource (payment or consent), begun the authorization process, then selected the method based on the feedback and, finally, optionally answered the TAN challenge. It is likely that you have only requested the status of a resource or an authorization explicitly in the event of an error.


ASPSP-SCA-Approach: EMBEDDED
Ressource Initiation Request
Start Authorisation Request with PSU Authentication
Method Selection Required?
no
yes
Update Authorisation Request with Method Selection
Authorisation Confirmation Required?
no
yes
Transaction Authorisation Confirmation Request
Detailed Information Required?
no
yes
GET Resource status and/ or GET Authorisation status

With the introduction of the pushTAN variant known as “pushTAN 2.0” (pushTAN decoupled) among the Sparkassen-Finanzgruppe, this process flow will change. A DECOUPLED procedure will be introduced that enables the TAN to be entered outside of your application.

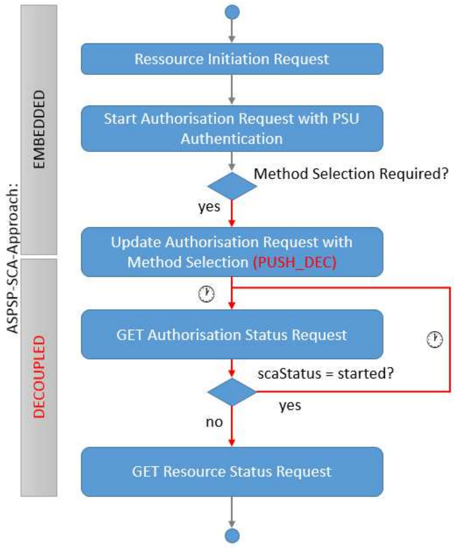
If a pushTAN customer is already able to make use of pushTAN decoupled (because their pushTAN app supports the process and the process can be used in theory), you will always be asked to select a method. The already familiar authenticationType PUSH_OTP as well as the new authenticationType PUSH_DEC are then offered.

If you or the user select the PUSH_OTP type of authenticationMethodId, subsequent processing continues to follow the familiar process. This allows you to flexibly set the period of time for adapting to the new procedure. After the transition phase ends, the artificially introduced method selection is no longer required if the user only has one scaMethod.

If you or the user select the PUSH_DEC type of authenticationMethodId, then we indicate in the response that you have switched to the DECOUPLED ASPSP-SCA approach (response header: AS-PSP-SCA-Approach = DECOUPLED). Subsequent authorization is decoupled via the S-pushTAN app. You will no longer trigger the transaction authorization confirmation request.

As part of this new procedure, to be able to identify whether the authorization or the resource could be successfully created/executed and you can display the proper banking result, the "GET status" request must in certain cases be invoked multiple times for the authorization. If the authorization has a positive final status, then "GET status" can be used for the resource to determine its status.

The user has a maximum of 12 minutes to carry out the authorization. The status can change during this period.


ASPSP-SCA-Approach:
EMBEDDED
DECOUPLED
Ressource Initiation Request
Start Authorisation Request with PSU Authentication
Method Selection Required?
yes
Update Authorisation Request with Method Selection (PUSH_DEC)
GET Authorisation Status Request
scaStatus = started?
yes
no
GET Resource Status Request

In the following subsections, the requests are described again in technical detail.

3.1 | Start Authorisation Request with PSU Authentication

The response (**scaMethods**) is being expanded to include the authenticationType PUSH_DEC. For pushTAN users, PUSH_OTP and PUSH_DEC will always be delivered during the transition phase. If you have not yet adapted your process at the start, then please filter out the PUSH_DEC entry and select only the PUSH_OTP type of authenticationMethodIds. Or vice versa: If you support the new process, please filter out authenticationMethodId PUSH_OTP.

	Authentication Type	Authentication Version	Authentication MethodId
	SMS_OTP	-	Description of the end user device
OLD	PUSH_OTP	-	Description of the end user device
NEW	PUSH_DEC	-	Description of the end user device
	CHIP_OTP	HHD1.3.2	MANUAL
	CHIP_OTP	HHD1.3.2OPT	OPTICAL
	CHIP_OTP	HHD1.3.2QR	QR

You will carry out the method selection in the next step.

Example request:

```
curl -X POST \
  https://...xs2a-
api/{bankcode}/v1/{resource}/{resourceID}/authorisations/{authorisationID} \
  -H 'Accept: */*' \
  -H 'Content-Type: application/json' \
  -H 'PSU-ID: Test123' \
  -H 'X-Request-ID: 97e8bd97-41ab-4761-a498-e847dec28f39' \
  -d '{
          "psuData": {
          "password": "Geheim"
  }
}
'
```

Example response:

```
HTTP/1.1 201
status: 201
Content-Type: application/json;charset=utf-8
Location: https://...xs2a-
api/>/{bankcode}/v1/{resource>/{resourceID}/authorisations/{authorisationID
}
ASPSP-SCA-Approach: EMBEDDED
X-Request-ID: 97e8bd97-41ab-4761-a498-e847dec28f39
{
   "scaStatus": "psuAuthenticated",
   "authorisationId": "{authorisationID}",
   "scaMethods": [
      {
         "authenticationType": "PUSH_OTP",
         "authenticationVersion": "",
         "authenticationMethodId": "Classic - Privat",
         "name": "pushTAN | Privat (******9387)"
      },
      {
         "authenticationType": "PUSH_OTP",
         "authenticationVersion": "",
         "authenticationMethodId": "Classic - Firma",
         "name": "pushTAN | BW (******7890)"
      },
      {
         "authenticationType": "PUSH_DEC",
         "authenticationVersion": "",
         "authenticationMethodId": "Privat",
         "name": "pushTAN | Privat (******9387)"
      },
      {
         "authenticationType": "PUSH_DEC",
         "authenticationVersion": "",
         "authenticationMethodId": "Firma",
         "name": "pushTAN | BW (******7890)"
      }
   ],
   "_links": {
      "scaStatus": {
         "href": "https://.../ xs2a-
api/{bankcode}/v1/{resource}/{resourceID}/authorisations/{authorisationID}"
      },
      "selectAuthenticationMethod": {
         "href": "https://.../xs2a-
api/{bankcode}/v1/{resource}/{resourceID}/authorisations/{authorisationID}"
```

```
      }
    },
    "psuMessage": "Bedienungshinweis an den Endanwender."
}
```

3.2 | Update Authorisation Request with Method Selection

The request for selecting the method remains unchanged, but the response in the API changes depending on the selection that has been made:

- The ASPSP-SCA-Approach header attribute contains either the value EMBEDDED or DECOUPLED.
- In the decoupled case, challengeData is not delivered and no link to authoriseTransaction is returned, since the approval takes place entirely within the S-pushTAN app.

Variant 1: An authenticationMethodId with "authenticationType": "PUSH_OTP" is selected.

```
curl -X PUT \
  https://.../xs2a-
api/{bankcode}/v1/{resource}/{resourceID}/authorisations/{authorisationID} \
  -H 'Accept: */*' \
  -H 'Content-Type: application/json' \
  -H 'X-Request-ID: 85dd4796-103d-4aa1-89fd-c5a7a32fdce9' \
  -d '{
            "authenticationMethodId": "Classic - Firma"
}
'
```

Response:

```
HTTP/1.1 200
status: 200
Content-Type: application/json; charset=utf-8
ASPSP-SCA-Approach: EMBEDDED
X-Request-ID: 85dd4796-103d-4aa1-89fd-c5a7a32fdce9
Location: https://.../xs2a-api/{bankcode}/v1/{resourceID}

{
    "scaStatus": "scaMethodSelected",
    "chosenScaMethod": {
      "authenticationType": "PUSH_OTP",
      "authenticationMethodId": "Classic - Firma",
      "name": "pushTAN | Classic - pushTAN_Med1"
    },
    "challengeData": {
      "otpMaxLength": 6,
      "otpFormat": "integer",
      "additionalInformation": "Bitte tragen Sie die TAN aus der S-pushTAN-App
ein."
    },
    "_links": {
      "authoriseTransaction": {
        "href":                                              "https://.../xs2a-
api/{bankcode}/v1/{resource}/{resourceID}/authorisations/{authorisationID}"
      },
      "scaStatus": {
        "href":                                              "https://.../xs2a-
api/{bankcode}/v1/{resource}/{resourceID}/authorisations/{authorisationID}"
      }
    },
    "psuMessage": " Bitte tragen Sie die TAN aus der S-pushTAN-App ein."
}
```

Variant 2: An authenticationMethodId with "authenticationType": "PUSH_DEC" is selected.

```
curl -X PUT \
  https://.../xs2a-
api/{bankcode}/v1/{resource}/{resourceID}/authorisations/{authorisationID} \
  -H 'Accept: */*' \
  -H 'Content-Type: application/json' \
  -H 'X-Request-ID: 6dcf9f3f-f4d6-47f6-b7fe-1a08e57ede17' \
  -d '{
          "authenticationMethodId": "Firma"
}
'
```

Response:

```
HTTP/1.1 200
status: 200
Content-Type: application/json; charset=utf-8
ASPSP-SCA-Approach: DECOUPLED
X-Request-ID: 6dcf9f3f-f4d6-47f6-b7fe-1a08e57ede17
Location: https://.../xs2a-api/{bankcode}/v1/{resourceID}

{
    "scaStatus": "started",
    "chosenScaMethod": {
      "authenticationType": "PUSH_DEC",
      "authenticationMethodId": "Firma",
      "name": "pushDecTAN | Firma"
    },
    "_links": {
      "scaStatus": {
        "href":                                              "https://.../xs2a-
api/{bankcode}/v1/{resource}/{resourceID}/authorisations/{authorisationID}"
      }
    },
    "psuMessage": "Bitte bestätigen Sie die Transaktion mit ihrer PushTAN-APP."
}
```

The following table compares the response variants once again:

	Variant 1: "PUSH_OTP"	Variant 2: "PUSH_DEC"
Header: ASPSP-SCA-Approach	EMBEDDED	DECOUPLED
Body: scaStatus	scaMethodSelected	started

Body: chosenScaMethod	{
	"authenticationType": "PUSH_OTP",
	"authenticationMethodId": "BW_Old",
	"name": "pushTAN | BW_old"
	}
		{
		"authenticationType": "PUSH_DEC",
		"authenticationMethodId": "BW",
		"name": "pushTAN decoupled | BW"
		}

Body: challengeData	{
	"otpMaxLength": 6,
	"otpFormat": "integer",
	"additionalInformation": "Bitte tragen Sie die TAN aus der S-pushTAN-App ein."
	}
		n/a

Body: _links	authoriseTransaction scaStatus	scaStatus

3.3 | Follow-up Requests

3.3.1 When PUSH_DEC Is Selected

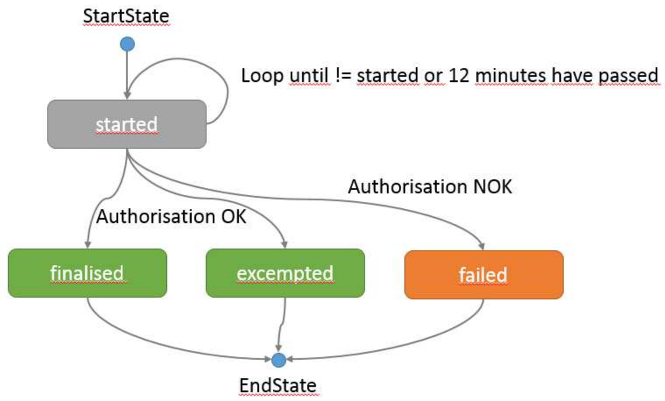
Going forward, you cyclically call the "GET authorization" endpoint to receive information about whether the order has been approved by the customer in the S-pushTAN app. Further requests on the actual resource only make sense if you receive a positive result.

```
curl -X GET \
  https://.../xs2a-
api/{bankcode}/v1/{resource}/{resourceID}/authorisations/{authorisationID} \
  -H 'Accept: application/json' \
  -H 'X-Request-ID: 97e8bd97-41ab-4761-a498-e847dec28f39' \
```

Beginning with this process step, the scaStatus attribute can have the following characteristics:

- started (New): The authorization has begun. The user has to perform the approval via the S-pushTAN app. Approval has not yet taken place.
- finalized: The authorization was successfully completed with a second factor.
- failed: The authorization was not completed successfully. In the decoupled approach, this means that either the time has expired or the user has not confirmed.

- exempted: The authorization was successfully completed without a second factor.



If the status of the authorization is OK (scaStatus "finalised" or "exempted"), the resource to be authorized is also updated and its status changes.

If the status of the authorization is failed, the resource to be authorized remains in the “received” or “RVCD” status. In this case, the authorization can be started again. You get further informations about the error by calling “GET Authorization”.

```
curl -X GET \
  https://.../xs2a-
api/{bankcode}/v1/{resource}/{resoureceID}/authorisations/{authorisationID} \
```

To query the status of the resource, please use:

```
curl -X GET \
  https://.../xs2a-api/{bankcode}/v1/{resource}/{resourceID}/status \
  -H 'X-Request-ID: 97e8bd97-41ab-4761-a498-e847dec28f39' \
```

The endpoint responds with an http status code 200 and signals the status of the resource via the response body:

consentStatus	transactionStatus	Comment
received	RCVD (Received)	This combination can only exist if the authorization has failed, but the status of the resource is still being read. The authorization can be started again beginning with Start Authorization with PSU Authentication
rejected	RJCT (Rejected)	Executing/creating the resource was rejected for reasons that have nothing to do with the

consentStatus	transactionStatus	Comment
		authorization (e.g., insufficient funds).
valid	ACCC (AcceptedSettlementCompleted) ACCP (AcceptedCustomerProfile) ACTC (AcceptedTechnicalValidation) PART (PartiallyAccepted)	The resource was created/executed.
-	PATC (PartiallyAcceptedTechnical Correct)	The resource was created/executed. It involves an instant payment that has not yet been fully executed.

The endpoint also provides a psuMessage with detailed information for the end user: "psuMessage": "{No.} - {Text}." Reasons for rejection can be found here.

3.3.2 When PUSH_OTP Is Selected

As usual, you conclude the authorization flow with authoriseTransaction:

```
curl -X PUT \
  https://.../xs2a-
api/{bankcode}/v1/{resource}/{resourceID}/authorisations/{authorisationID} \
  -H 'Content-Type: application/json' \
  -H 'X-Request-ID: 97e8bd97-41ab-4761-a498-e847dec28f39' \
  -d '{
          "scaAuthenticationData": "123456"
 }'
```

3.4 | Behavior by an outdated pushTAN-App

If a costumer can use pushTAN decoupled in principle, but is still using an outdated app that does not support pushTAN decpoupled, the followeing bevavior is to be expected:

If you or the customer select an authenticationMethodId of the type PUSH_DEC, the authorization can't be completed successfully. If you call "GET authorization" you get the following response:

```
{
   "scaStatus": "failed",
   "psuMessage": "3015- Abrufversuch durch inkompatiblen Client"
}
```

In this case, you have to restart the authorization with an authenticationMethodId of the type PUSH_OTP. Further processing is carried out according to the known procedure.

4 | Support

4.1 | Sandbox

The pushTAN decoupled process is also simulated in the sandbox we have provided. The behavior is linked to a PSU ID. The PSU ID is called "pushDecTAN".

Attribute	**Value**
PSU-ID	pushDecTAN
password	okok1
scaAuthenticationData	111111

4.2 | Questions?

Should you have any further questions, please contact our support team:
https://xs2a.sparkassen-hub.com/home

4.3 Adaptation of the documentation

The provided YAML in the carepackage on the XS2A Welcome-Page has been revised in the course of providing pushTAN decoupled.