



Handbook

pushTAN 2.0 (pushTAN decoupled)

Contents

1	 Änderungshistorie	2
2	 Management Summary	3
2.1	Zielgruppe des Dokuments	3
2.2	Terminplanung	3
3	 Veränderungen am Request Flow	3
3.1	Start Authorisation Request with PSU Authentication	5
3.2	Update Authorisation Request with Method Selection	7
3.3	Folgerequests	9
3.4	Verhalten bei einer veralteten pushTAN-App	12
4	 Support	12
4.1	Sandbox	12
4.2	Bei Rückfragen	12
4.3	Anpassung der Dokumentation	12

1 | Änderungshistorie

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

Das pushTAN-Verfahren wird angepasst, sodass Endkunden bei Freigabe eines Auftrags künftig in der pushTAN-App keine TAN angezeigt bekommen, die in der Banking-Anwendung (bzw. Drittdienste-Anwendung) manuell eingegeben werden muss. Stattdessen kann der Auftrag in der pushTAN-App durch eine Schaltfläche direkt freigegeben werden.

Die neue pushTAN-Variante wird als „**pushTAN 2.0**“ bezeichnet.

Der sogenannte DECOUPLED-Approach wird nur für die TAN-Eingabe angeboten.

2.1 | Zielgruppe des Dokuments

Zielgruppe sind alle Drittdienstleister, die den **Embedded SCA Approach** verwenden. Die hier beschriebenen Änderungen betreffen nur den authentication-Type PUSH_OTP (**pushTAN**).

Für den authenticationType CHIP_OTP und SMS_OTP ergeben sich keine Änderungen. Gleiches gilt für alle Drittdienstleister die den SCA Approach REDIRECT verwenden.

2.2 | Terminplanung

Seit dem 18. November 2020 steht die Sandbox für Tests zur Verfügung. Die Echtschnittstelle wird pushTAN 2.0 ab dem 09.03.2021 für alle Institute unterstützen.

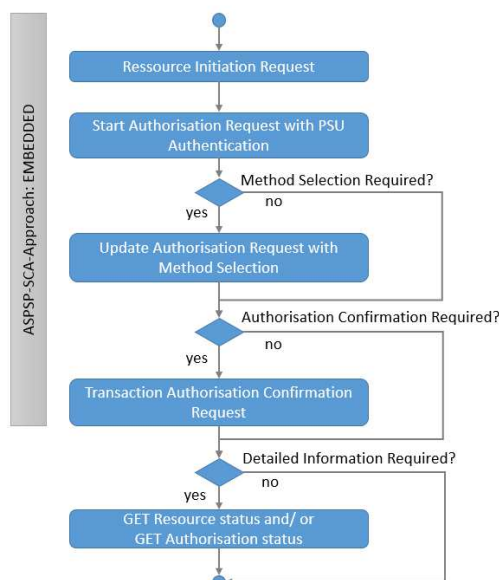
Die Aktivierung der pushTAN-Variante "pushTAN 2.0" (pushTAN decoupled) erfolgt unabhängig von der Softwareänderung in der XS2A-API durch die jeweiligen Institute. Das hat für die XS2A-Schnittstelle die Auswirkung, dass nur sehr vereinzelte Kunden mit der pushTAN-Variante 2.0 über die Schnittstelle zugreifen werden.

Der Flächen-Rollout ist für das 2. Quartal 2021 geplant. Ausführliche Informationen stellen wir Ihnen zeitgerecht im Vorfeld u. a. im Rahmen einer separaten Information zur Verfügung.

Wir bitten Sie daher im Sinne eines positiven Endbenutzer-Erlebnisses bis Ende März 2021 die Minimalanpassung vorzunehmen und empfehlen Ihnen bis Ende Juli 2021 den DECOUPLED-Approach zu unterstützen.

3 | Veränderungen am Request Flow

Bisher haben Sie eine Ressource (Payment oder Consent) initiiert, den Autorisierungs-Prozess gestartet, je nach Rückmeldung dann die Methodenauswahl vorgenommen und abschließend optional dann die TAN-Challenge beantwortet. Eine explizite Abfrage des Status einer Ressource oder einer Autorisierung erfolgte durch Sie wahrscheinlich nur im Fehlerfall.



Mit der Einführung der pushTAN-Variante „pushTAN 2.0“ (pushTAN decoupled) in der Sparkassen-Finanzgruppe wird sich dieser Prozessablauf ändern. Es wird ein DECOUPLED-Verfahren eingeführt, dass die TAN-Eingabe außerhalb Ihrer Applikation ermöglicht.

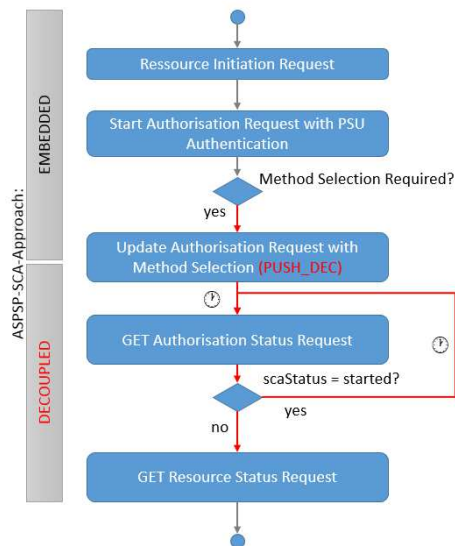
Sofern es sich um einen pushTAN-Kunden handelt, der pushTAN decoupled bereits nutzen kann (weil seine pushTAN-App in der Lage ist das Verfahren zu unterstützen sowie das Verfahren grundsätzlich nutzbar ist), werden Sie immer zur Methodenauswahl aufgefordert. Angeboten werden dann der bisher bekannte authenticationType PUSH_OTP und zusätzlich der neue authenticationType PUSH_DEC.

Wenn Sie oder der Benutzer eine authenticationMethodId vom Typ PUSH_OTP wählen, dann folgt die weitere Verarbeitung nach dem bisher bekannten Prozess. Damit können Sie flexibel den Zeitraum für die Anpassung an das neue Verfahren definieren. Nach Ende der Übergangsphase entfällt die künstlich eingeführte Methodenauswahl, wenn der Benutzer nur eine scaMethod hat.

Wenn Sie oder der Benutzer eine authenticationMethodId vom Typ PUSH_DEC wählen, dann signalisieren wir in der Response das Sie in den ASPSP-SCA-Approach DECOUPLED gewechselt sind (Response-Header: ASPSP-SCA-Approach = DECOUPLED). Die weitere Autorisierung erfolgt entkoppelt über die S-pushTAN-App. Der Transaction Authorisation Confirmation Request wird von Ihnen nicht mehr ausgelöst.

Um im neuen Verfahren identifizieren zu können, ob die Autorisierung bzw. die Ressource erfolgreich angelegt/ ausgeführt werden konnte und Sie das bankfachliche Ergebnis anzeigen können, muss der Request „GET status“ für die Autorisierung ggf. mehrfach aufgerufen werden. Wenn die Autorisierung einen positiven finalen Status hat, dann kann „GET status“ für die Ressource genutzt werden, um deren Status zu ermitteln.

Der Benutzer hat maximal 12 Minuten Zeit für die Durchführung der Autorisierung. In diesem Zeitraum kann sich der Status ändern.



In den folgenden Unterkapiteln werden die Requests noch einmal technisch detailliert beschrieben.

3.1 | Start Authorisation Request with PSU Authentication

Die Rückantwort (**scaMethods**) wird erweitert um den authenticationType PUSH_DEC. Für pushTAN Benutzer werden in der Übergangsphase immer PUSH_OTP und PUSH_DEC geliefert. Wenn Sie Ihren Prozess zu Beginn noch nicht angepasst haben, dann filtern Sie bitte den Eintrag PUSH_DEC aus und wählen Sie nur authenticationMethodIds vom Typ PUSH_OTP. Oder auch umgekehrt: Wenn Sie den neuen Prozess unterstützen, dann filtern Sie bitte authenticationMethodId PUSH_OTP aus.

	Authentication Type	Authentication Version	Authentication MethodId
	SMS_OTP	-	Bezeichnung des Benutzer-Endgerätes
ALT	PUSH_OTP	-	Bezeichnung des Benutzer-Endgerätes
NEU	PUSH_DEC	-	Bezeichnung des Benutzer-Endgerätes
	CHIP_OTP	HHD1.3.2	MANUAL
	CHIP_OTP	HHD1.3.2OPT	OPTICAL
	CHIP_OTP	HHD1.3.2QR	QR

Im nächsten Schritt führen Sie die Methodenauswahl durch.

Beispiel-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"
    }
  }
'
```

Beispiel-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

Der Request für die Methodenauswahl ist unverändert, aber abhängig von der getroffenen Auswahl verändert sich die Rückmeldung in der API:

- Das Header-Attribut ASPSP-SCA-Approach enthält entweder den Wert EMBEDDED oder DECOUPLED.
- Im Decoupled-Fall wird challengeData nicht ausgeliefert und es wird kein Link auf authoriseTransaction zurückgegeben, da die Freigabe komplett in der S-pushTAN-App erfolgt.

Variante 1: Es wird eine authenticationMethodId mit authenticationType: "PUSH_OTP" gewählt.

```

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
ASPP-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}/au-
thorisations/{authorisationID}"
  },
  "scaStatus": {
    "href": "https://.../xs2a-api/{bankcode}/v1/{resource}/{resourceID}/au-
thorisations/{authorisationID}"
  }
},
"psuMessage": " Bitte tragen Sie die TAN aus der S-pushTAN-App ein."
}

```

Variante 2: Es wird eine authenticationMethodId mit authenticationType: "PUSH_DEC" gewählt.

```

curl -X PUT \
  https://.../xs2a-api/{bankcode}/v1/{resource}/{resourceID}/authorisa-
tions/{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."
  }

```

Die folgende Tabelle stellt die Antwortvarianten nochmals gegenüber:

	Variante 1: "PUSH_OTP"	Variante 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." }	entfällt
Body: _links	authoriseTransaction scaStatus	scaStatus

3.3 | Folgerequests

3.3.1 Bei Auswahl von PUSH_DEC

Sie rufen ab jetzt zyklisch den Endpunkt „GET authorisation“ auf, um die Information zu erhalten, ob der Auftrag durch den Kunden in der S-pushTAN-App freigegeben wurde. Nur wenn sie ein positives Ergebnis erhalten machen weitere Abfragen auf die eigentliche Ressource Sinn.

```

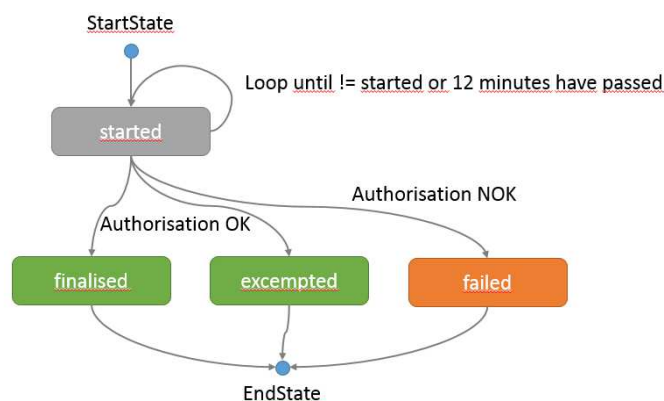
curl -X GET \
  https://.../xs2a-api/{bankcode}/v1/{resource}/{resourceID}/authorisations/{authorisationID} \
  -H 'Accept: application/json' \

```

-H 'X-Request-ID: 97e8bd97-41ab-4761-a498-e847dec28f39' \

Das Attribut scaStatus kann ab diesem Prozessschritt folgende Ausprägungen haben:

- **started (Neu):** Die Autorisierung wurde gestartet. Der Benutzer muss die Freigabe über die S-pushTAN-App vornehmen. Die Freigabe ist noch nicht erfolgt.
- **finalized:** Die Autorisierung wurde erfolgreich mit einem zweiten Faktor abgeschlossen.
- **failed:** Die Autorisierung wurde nicht erfolgreich abgeschlossen. Im Decoupled-Approach heißt das konkret: Entweder die Zeit ist abgelaufen oder der Benutzer hat nicht bestätigt.
- **exempted:** Die Autorisierung wurde erfolgreich ohne einen zweiten Faktor abgeschlossen.



Wenn der Status der Autorisierung OK ist (scaStatus „finalised“ oder „exempted“) wird auch die zu autorisierende Ressource aktualisiert und wechselt den Status.

Wenn der Status der Autorisierung „failed“ ist, bleibt die zu autorisierende Ressource im Status „received“ bzw. „RCVD“. In diesem Fall kann die Autorisierung nochmals gestartet werden. Weiterhin erhalten Sie durch den Aufruf von „GET Authorisation“ Informationen zum vorhandenen Fehler.

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

Um den Status der Ressource abzufragen nutzen Sie bitte

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

Der Endpunkt antwortet mit einem http Status Code 200 und signalisiert den Zustand der Ressource über den Response Body:

consentStatus	transactionStatus	Erläuterung
received	RCVD (Received)	Diese Kombination kann es nur geben, wenn die Autorisierung fehlgeschlagen ist, aber trotzdem der Status der Ressource gelesen wird. Die Autorisierung kann nochmals gestartet werden beginnend mit Start Authorisation with PSU Authentication
rejected	RJCT (Rejected)	Die Ausführung/ Anlage der Ressource wurde abgelehnt aus Gründen die nichts mit der Autorisierung zu tun haben. Bsp.: Keine Deckung.
valid	ACCC (AcceptedSettlementCompleted) ACCP (AcceptedCustomerProfile) ACTC (AcceptedTechnicalValidation) PART (PartiallyAccepted)	Die Ressource wurde angelegt/ ausgeführt.
-	PATC (PartiallyAcceptedTechnical Correct)	Die Ressource wurde angelegt/ ausgeführt. Es handelt sich um ein Instant Payment das noch nicht komplett ausgeführt ist.

Der Endpunkt liefert zusätzlich eine psuMessage mit Detailinformationen aus denen der Grund (insbesondere der Ablehnungen) für den Endbenutzer ersichtlich ist: "psuMessage": "{Nr.}- {Text}."

3.3.2 Bei Auswahl von PUSH_OTP

Sie schließen wie gewohnt den Autorisierungs-Flow mit authorise Transaction ab:

```
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 | Verhalten bei einer veralteten pushTAN-App

Sofern ein Kunde das pushTAN decoupled Verfahren prinzipiell nutzen kann, jedoch noch eine veraltete App nutzt, welche decoupled-Variante nicht unterstützt, ist mit dem folgenden Verhalten zu rechnen:

Wenn Sie oder der Benutzer eine authenticationMethodId vom Typ PUSH_DEC wählen, kann die Autorisierung nicht erfolgreich beendet werden. Wenn Sie nun den Endpunkt „GET authorisation“ aufrufen, erhalten Sie die folgende Rückmeldung:

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

In diesem Fall müssen Sie die Autorisierung der Ressource erneut starten und die authenticationMethodId vom Typ PUSH_OTP wählen, dann folgt die weitere Verarbeitung nach dem bisher bekannten Prozess.

4 | Support

4.1 | Sandbox

In der von uns bereitgestellten Sandbox wurde das pushTAN decoupled Verfahren ebenfalls simuliert. Das Verhalten ist an eine PSU-ID gekoppelt. Die PSU-ID lautet: „pushDecTAN“.

Attribut	Wert
PSU-ID	pushDecTAN
password	okok1
scaAuthenticationData	111111

4.2 | Bei Rückfragen

Wenn Sie weitere Fragen haben wenden Sie sich bitte an unseren Support: <https://xs2a.sparkassen-hub.com/home>

4.3 | Anpassung der Dokumentation

Die bereitgestellte YAML im carepackage auf der XS2A Welcome-Page ist im Zuge der Anpassungen für das pushTAN decoupled Verfahren überarbeitet worden.