

GETTING STARTED



PASS OPEN BANKING - UMWELTBANK XS2A

Getting Started

Eine Einführung in die Verwendung der xs2a-Schnittstelle

Version: 1.2

Status: Freigegeben

Vertraulichkeit: Öffentlich

Der Inhalt von elektronischen Kopien oder Ausdrucken kann vom Originaldokument abweichen. Bei Abweichungen ist das Originaldokument verbindlich, das beim Verantwortlichen (siehe Tabelle Dokumenteninformationen) angefordert werden kann.

DOKUMENTENINFORMATIONEN

Dokumentenname:	Getting Started
Beschreibung:	Eine Einführung in die Verwendung der xs2a-Schnittstelle
Aktuelle Versionsnummer:	1.2
Verantwortlich für den Inhalt:	PASS
Klassifizierung der Vertraulichkeit:	Öffentlich
Aktueller Status:	Freigegeben

Das vorliegende Dokument unterliegt dem Urheberrecht. Alle Rechte sind geschützt. Jegliche Vervielfältigung oder Verbreitung, ganz oder teilweise, ist ohne schriftliche Zustimmung der PASS Consulting Group unzulässig und strafbar.

Text, Gestaltung und Layout: © PASS Consulting Group

BEARBEITUNGSHISTORIE

Version	Autor	Gegenstand	Datum
1.0	PASS Consulting	Initiale Anlage	11.06.2019
1.1	PASS Consulting	Beispiel pain-Nachricht eingefügt	15.10.2020
1.2	PASS Consulting	Aktualisieren der Requests und Responses	02.02.2021

INHALT

1	Voraussetzungen.....	4
1.1	Zertifikat	4
1.2	Registrierung	4
2	Verwendung der Schnittstelle.....	5
2.1	Auslösung einer Zahlung	5
2.1.1	Request - POST /{payment-service}/{payment-product}.....	5
2.1.2	Response - POST /{payment-service}/{payment-product}	7
2.1.3	Request - POST /{payment-service}/{payment-product}/{paymentId}/authorisations	7
2.1.4	Response - POST /{payment-service}/{payment-product}/{paymentId}/authorisations	8
2.1.5	Request - PUT /{payment-service}/{payment-product}/{paymentId}/authorisations/{AuthorisationId}	8
2.1.6	Response - PUT /{payment-service}/{payment-product}/{paymentId}/authorisations/{AuthorisationId}	8
2.1.7	Request - PUT /{payment-service}/{payment-product}/{paymentId}/authorisations/{AuthorisationId}	9
2.1.8	Response - PUT /{payment-service}/{payment-product}/{paymentId}/authorisations/{AuthorisationId}	9
2.1.9	Request – GET /{payment-service}/{payment-product}/{paymentId}/status	10
2.1.10	Response – GET /{payment-service}/{payment-product}/{paymentId}/status.....	10
2.2	Abfrage einer Kontoübersicht	11
2.2.1	Request – POST /consents	11
2.2.2	Response – POST /consents	11
2.2.3	Request – POST /consents/{consentId}/authorisations	12
2.2.4	Response – POST /consents/{consentId}/authorisations.....	12
2.2.5	Request – PUT /consents/{consentId}/authorisations	12
2.2.6	Response – PUT /consents/{consentId}/authorisations	12
2.2.7	Request – PUT /consents/{consentId}/authorisations	13
2.2.8	Response – PUT /consents/{consentId}/authorisations	13
2.2.9	Request – GET /accounts.....	13
2.2.10	Response – GET /accounts.....	14

1 VORAUSSETZUNGEN

1.1 KONTAKT ZUR REGULIERENDEN STELLE

Damit Sie als TPP (Third Party Provider) die produktiven XS2A-Schnittstellen nutzen können, müssen Sie sich bei der regulierenden Stelle (in Deutschland ist dies die BaFin) registrieren oder eine Erlaubnis einholen.

1.2 TRUST CENTER ZERTIFIKAT

Um die XS2A-API der Umweltbank nutzen zu können, benötigen Sie ein Echtzertifikat, welches von einem zugelassenen Trust Center ausgestellt ist. In Deutschland werden diese Zertifikate z.B. durch die Bundesdruckerei ausgestellt. Weitere Informationen sind beispielsweise unter <https://www.bundesdruckerei.de/de/PSD2> zu finden. Eine Liste der internationalen Trust Center finden Sie unter <https://webgate.ec.europa.eu/tl-browser/#/>, diese Trust-Center werden im produktiven Kontext unterstützt.

Dieses Zertifikat können Sie erst nach der Zulassung durch die regulierende Stelle beantragen.

1.3 REGISTRIERUNG

Neben dem QWAC-Zertifikat ist eine Registrierung notwendig. Hierzu genügt eine formlose E-Mail an xs2a_umweltbank@pass-consulting.com mit folgenden Informationen:

- E-Mail-Adresse zu Kontaktzwecken
- Name ihrer Organisation
- Webseite ihrer Organisation
- Rollen ihres Zertifikates (z.B. „PSP_PI“)
- PSP-ID ihres Zertifikates (z.B. PSDDE-BAFIN-123456 – entspricht OID 2.5.4.97)
- Aussteller (inkl. Land) ihres Zertifikates

Bei konkreten Fragen zu durchgeführten API-Aufrufen, senden Sie uns bitte in der Nachricht folgende zusätzliche Informationen:

X-Request-ID, aufgerufener API-Endpunkt, paymentId oder ConsentId und Datum/Uhrzeit des Requests.

2 VERWENDUNG DER SCHNITTSTELLE

Bei der Schnittstelle handelt es sich bei den meisten Endpunkten um einen HTTP-JSON-Schnittstelle. Unter <https://open-banking.pass-consulting.com/> sind weiterführende Informationen und eine Sandbox-Umgebung zu finden.

Anhand von zwei Beispielen – der Auslösung einer Zahlung und der Abfrage einer Kontoübersicht – soll im Folgenden erläutert werden, wie die Schnittstelle verwendet werden kann.

2.1 AUSLÖSUNG EINER ZAHLUNG

Ein Endkunde möchte eine einmalige Zahlung auslösen.

2.1.1 REQUEST - POST /{PAYMENT-SERVICE}/{PAYMENT-PRODUCT}

Zunächst wird ein POST-Aufruf an /{payment-service}/{payment-product} getätigt. Dabei sind die Variablen im Pfad zu setzen. Da es sich um eine einmalige Zahlung handelt und die Zahlung im pain-xml-Format übergeben wird, lautet der Pfad /payments/pain.001-sepa-credit-transfers.

Im Header sollten – soweit vorhanden – die Daten über die Anfrage des Endkunden in den Feldern PSU-* erfasst werden. Die Identifikationsnummer PSU-ID muss auf jeden Fall übergeben werden. Die Nachricht kann auch noch über die Header-Felder Digest, Signature und TPP-Signature-Certificate signiert werden.

Im Body der Anfrage wird die Zahlung im pain-xml-Format übergeben.

```
POST https://.../api/v1/payments/pain.001-sepa-credit-transfers
accept: application/json
content-type: application/xml
psu-id: <Login-ID des Endkunden>
x-request-id: <UUID für diesen Request>

<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03">
  <CstmrCdtTrfInitn>
    <GrpHdr>
      <MsgId>ABC/090928/CCT001</MsgId>
      <CreDtTm>yyyy-mm-ddThh:mm:ss</CreDtTm>
      <NbOfTx> ... </NbOfTx>
      <CtrlSum>11500000</CtrlSum>
      <InitgPty>
        <Nm>ABC Corporation</Nm>
        <PstAdr>
          <StrtNm> ... </StrtNm>
          <BldgNb> ... </BldgNb>
          <PstCd> ... </PstCd>
          <TwnNm> ... </TwnNm>
          <Ctry> ... </Ctry>
        </PstAdr>
      </InitgPty>
    </GrpHdr>
    <PmtInf>
      <PmtInfId>ABC/086</PmtInfId>
      <PmtMtd>TRF</PmtMtd>
```

```

<BtchBookg>>false</BtchBookg>
<ReqdExctnDt>2020-03-03</ReqdExctnDt>
<Dbtr>
  <Nm>ABC Corporation</Nm>
  <PstlAdr>
    <StrtNm> ... </StrtNm>
    <BldgNb> ... </BldgNb>
    <PstCd> ... </PstCd>
    <TwnNm> ... </TwnNm>
    <Ctry> ... </Ctry>
  </PstlAdr>
</Dbtr>
<DbtrAcct>
  <Id>
    <IBAN> ... </IBAN>
  </Id>
</DbtrAcct>
<DbtrAgt>
  <FinInstnId>
    <BIC> ... </BIC>
  </FinInstnId>
</DbtrAgt>
<CdtTrfTxInf>
  <PmtId>
    <InstrId>ABC/090928/CCT001/01</InstrId>
    <EndToEndId>ABC/4562/yyyy-mm-dd</EndToEndId>
  </PmtId>
  <Amt>
    <InstdAmt Ccy="EUR">0.01</InstdAmt>
  </Amt>
  <ChrgBr>SHAR</ChrgBr>
  <CdtrAgt>
    <FinInstnId>
      <BIC> ... </BIC>
    </FinInstnId>
  </CdtrAgt>
  <Cdtr>
    <Nm>DEF Electronics</Nm>
    <PstlAdr>
      <AdrLine> -street- </AdrLine>
      <AdrLine> -postalcode + city- </AdrLine>
      <AdrLine> -country- </AdrLine>
    </PstlAdr>
  </Cdtr>
  <CdtrAcct>
    <Id>
      <IBAN> ... </IBAN>
    </Id>

```

```

</CdtrAcct>
<Purp>
  <Cd>CINV</Cd>
</Purp>
<RmtInf>
  <Strd>
    <RfrdDocInf>
      <Nb> -number- </Nb>
      <RltdDt> yyyy-mm-dd </RltdDt>
    </RfrdDocInf>
  </Strd>
</RmtInf>
</CdtTrfTxInf>
</PmtInf>
</CstmrCdtTrfInitn>
</Document>

```

2.1.2 RESPONSE - POST /{PAYMENT-SERVICE}/{PAYMENT-PRODUCT}

201

ASPSP-SCA-Approach: EMBEDDED

Content-Type: application/json

Location: /api/v1/payments/pain.001-sepa-credit-transfers/<Payment-ID>

x-request-id: <UUID für diesen Request>

```

{
  "transactionStatus" : "RCVD",
  "paymentId" : "<Payment-ID>",
  "_links" : [ {
    "self" : {
      "href" : "/api/v1/payments/pain.001-sepa-credit-transfers/<Payment-ID>"
    }
  }, {
    "status" : {
      "href" : "/api/v1/payments/pain.001-sepa-credit-transfers/<Payment-ID>/status"
    }
  }, {
    "startAuthorisationWithPsuAuthentication" : {
      "href" : "/api/v1/payments/pain.001-sepa-credit-transfers/<Payment-ID>/authorisations"
    }
  }
]
}

```

In der Antwort sind zwei wichtige Informationen enthalten. Einerseits eine eindeutige ID der Zahlung (im Response-Body unter „paymentId“ und im Header unter „Location“) und andererseits das Verfahren für die SCA (im Header unter „ASPSP-SCA-Approach“ sowie im Body). Im Folgenden wird davon ausgegangen, dass das EMBEDDED-Verfahren für den SCA verwendet wird.

Die Zahlung befindet sich im Zustand „received“. Die Zahlung wurde also vom Zahlungsdienstleister empfangen. Ausgeführt wird die Zahlung allerdings erst nach der Autorisierung.

2.1.3 REQUEST - POST /{PAYMENT-SERVICE}/{PAYMENT-

PRODUCT/{PAYMENTID}/AUTHORISATIONS

Mit diesem Request wird das Autorisierungsverfahren gestartet

```
POST https://.../api/v1/payments/pain.001-sepa-credit-transfers/<Payment-ID>/authorisations
accept: application/json
content-type: application/json
x-request-id: <UUID für diesen Request>
```

```
{ }
```

2.1.4 RESPONSE - POST /{PAYMENT-SERVICE}/{PAYMENT-PRODUCT}/{PAYMENTID}/AUTHORISATIONS

201

```
ASPSP-SCA-Approach: EMBEDDED
Content-Type: application/json
x-request-id: <UUID für diesen Request>
```

```
{
  "scaStatus" : "psuIdentified",
  "authorisationId" : "auth-wS41LbMLE5IZ38p",
  "_links" : [ {
    "updatePsuAuthentication" : {
      "href" : "/api/v1/payments/pain.001-sepa-credit-transfers/<Payment-ID>/authorisations/<Authorisation-ID>"
    }
  } ]
}
```

Die Autorisierung ist nun angelegt.

2.1.5 REQUEST - PUT /{PAYMENT-SERVICE}/{PAYMENT-PRODUCT}/{PAYMENTID}/AUTHORISATIONS/{AUTHORISATIONID}

Nachdem das Passwort des Endkunden abgefragt wurde, kann es an die Schnittstelle übertragen werden.

```
PUT https://.../api/v1/payments/pain.001-sepa-credit-transfers/<Payment-ID>/authorisations/<Authorisation-ID>
accept: application/json
content-type: application/json
psu-id: <Login-ID des Endkunden> (Optional)
x-request-id: <UUID für diesen Request>
```

```
{
  "psuData" : {
    "password" : "<Password des Endkunden>"
  }
}
```

2.1.6 RESPONSE - PUT /{PAYMENT-SERVICE}/{PAYMENT-PRODUCT}/{PAYMENTID}/AUTHORISATIONS/{AUTHORISATIONID}

200

```
ASPSP-SCA-Approach: EMBEDDED
Content-Type: application/json
```


x-request-id: <UUID für diesen Request>

```
{
  "scaStatus" : "psuAuthenticated",
  "chosenScaMethod" : {
    "authenticationType" : "SMS_OTP",
    "authenticationVersion" : "1",
    "authenticationMethodId" : "Mtan",
    "name" : "Mtan an die registrierte Handynummer",
    "explanation" : "Generiert eine Mtan und verschickt diese an die registrierte Handynummer"
  },
  "challengeData" : {
    "otpMaxLength" : 6,
    "otpFormat" : "integer"
  },
  "_links" : [ ]
}
```

Das Passwort wurde akzeptiert und das hinterlegte SCA-Verfahren (hier ein TAN-Verfahren per SMS) wird gestartet.

2.1.7 REQUEST - PUT /{PAYMENT-SERVICE}/{PAYMENT-PRODUCT}/{PAYMENTID}/AUTHORISATIONS/{AUTHORISATIONID}

Der Benutzer bekommt seine TAN per SMS und diese TAN kann nun zur Autorisierung der Zahlung verwendet werden.

```
PUT https://.../api/v1/payments/pain.001-sepa-credit-transfers/<Payment-ID>/authorisations/<Authorisation-ID>
accept: application/json
connection: keep-alive
content-type: application/json
psu-id: <Login-ID des Endkunden> (Optional)
x-request-id: <UUID für diesen Request>

{
  "scaAuthenticationData" : "<TAN des Endkunden>"
}
```

2.1.8 RESPONSE - PUT /{PAYMENT-SERVICE}/{PAYMENT-PRODUCT}/{PAYMENTID}/AUTHORISATIONS/{AUTHORISATIONID}

```
200
Content-Type: application/json
x-request-id: <UUID für diesen Request>

{
  "scaStatus" : "finalised",
  "_links" : [ ]
}
```

Bei korrekter TAN bestätigt der Server das Ergebnis. Die Autorisierung ist abgeschlossen.

2.1.9 REQUEST – GET /{PAYMENT-SERVICE}/{PAYMENT-PRODUCT}/{PAYMENTID}/STATUS

Der Status der Zahlung kann jederzeit abgefragt werden (auch bevor die Zahlung autorisiert ist).

```
GET https://.../api/v1/payments/pain.001-sepa-credit-transfers/<Payment-ID>/status
```

```
accept: application/json
```

```
x-request-id: <UUID für diesen Request>
```

2.1.10 RESPONSE – GET /{PAYMENT-SERVICE}/{PAYMENT-PRODUCT}/{PAYMENTID}/STATUS

```
200
```

```
Content-Type: application/json
```

```
x-request-id: <UUID für diesen Request>
```

```
{  
  "transactionStatus" : "RJCT"  
}
```

In diesem Beispiel wurde die Zahlung leider abgelehnt.

2.2 ABFRAGE EINER KONTOÜBERSICHT

Der Kunde möchte seine Kontostände über einen längeren Zeitraum über einen Drittanbieter abfragen

2.2.1 REQUEST – POST /CONSENTS

Zunächst wird die Einwilligung des Benutzers benötigt. Dazu wird ein Request an diesen Endpunkt geschickt.

```
POST https://.../api/v1/consents
accept: application/json
content-type: application/json
psu-id: <Login-ID des Endkunden>
x-request-id: <UUID für diesen Request>

{
  "access" : {
    "allPsd2" : "allAccounts"
  },
  "recurringIndicator" : true,
  "validUntil" : <Zeitpunkt in der Zukunft - Formatbeispiel: "2019-07-13">,
  "frequencyPerDay" : 1,
  "combinedServiceIndicator" : false
}
```

2.2.2 RESPONSE – POST /CONSENTS

```
201
ASPSP-SCA-Approach: EMBEDDED
Content-Type: application/json
Location: /api/v1/consents/<Consent-ID>

{
  "consentStatus" : "received",
  "consentId" : "<Consent-ID>/",
  "_links" : [ {
    "self" : {
      "href" : "/api/v1/consents/<Consent-ID>"
    }
  }, {
    "status" : {
      "href" : "/api/v1/consents/<Consent-ID>/status"
    }
  }, {
    "startAuthorisationWithPsuAuthentication" : {
      "href" : "<URL für Authorisierung>"
    }
  } ]
}
```

Die Antwort enthält die URL zum Starten der Autorisierung.

2.2.3 REQUEST – POST /CONSENTS/{CONSENTID}/AUTHORISATIONS

Mit dieser URL kann nun die Autorisierung gestartet werden.

```
POST https://.../api/v1/consents/<Consent-ID>/authorisations
accept: application/json
content-type: application/json
x-request-id: <UUID für diesen Request>

{ }
```

2.2.4 RESPONSE – POST /CONSENTS/{CONSENTID}/AUTHORISATIONS

```
201
ASPSP-SCA-Approach: EMBEDDED
Content-Type: application/json
x-request-id: <UUID für diesen Request>

{
  "scaStatus" : "psuIdentified",
  "authorisationId" : "<Authorisation-ID>",
  "_links" : [ {
    "updatePsuAuthentication" : {
      "href" : "<URL für Autorisierung>"
    }
  } ]
}
```

In der Antwort ist die URL enthalten, an der die Autorisierung fortgesetzt wird.

2.2.5 REQUEST – PUT /CONSENTS/{CONSENTID}/AUTHORISATIONS

Beim Embedded-Verfahren ist zunächst das Kundenpasswort zu übergeben.

```
PUT https://.../api/v1/consents/<Consent-ID>/authorisations/<Authorisation-ID>
accept: application/json
content-type: application/json
psu-id: <Login-ID des Endkunden> (Optional)
x-request-id: <UUID für diesen Request>

{
  "psuData" : {
    "password" : "<Password des Endkunden>"
  }
}
```

2.2.6 RESPONSE – PUT /CONSENTS/{CONSENTID}/AUTHORISATIONS

```
200
ASPSP-SCA-Approach: EMBEDDED
Content-Type: application/json
```

```
x-request-id: <UUID für diesen Request>

{
  "scaStatus" : "psuAuthenticated",
  "chosenScaMethod" : {
    "authenticationType" : "SMS_OTP",
    "authenticationVersion" : "1",
    "authenticationMethodId" : "Mtan",
    "name" : "Mtan an die registrierte Handynummer",
    "explanation" : "Generiert eine Mtan und verschickt diese an die registrierte Handynummer"
  },
  "challengeData" : {
    "otpMaxLength" : 6,
    "otpFormat" : "integer"
  },
  "_links" : [ ]
}
```

Durch den Aufruf wird der Versand einer TAN ausgelöst.

2.2.7 REQUEST – PUT /CONSENTS/{CONSENTID}/AUTHORISATIONS

Nachdem der Kunde die TAN empfangen und eingegeben hat, kann diese an die Schnittstelle übergeben werden.

```
PUT https://.../api/v1/consents/<Consent-ID>/authorisations/<Authorisation-ID>
accept: application/json
content-type: application/json
psu-id: <Login-ID des Endkunden> (Optional)
x-request-id: <UUID für diesen Request>

{
  "scaAuthenticationData" : "<Vom Endkunden eingegebene TAN>"
}
```

2.2.8 RESPONSE – PUT /CONSENTS/{CONSENTID}/AUTHORISATIONS

```
200
Content-Type: application/json
x-request-id: <UUID für diesen Request>

{
  "scaStatus" : "finalised",
  "_links" : [ ]
}
```

Ist die TAN richtig, so ist die Autorisierung abgeschlossen und die Einwilligung des Kunden gültig.

2.2.9 REQUEST – GET /ACCOUNTS

Es können nun die Konten des Kunden abgerufen werden.

```
GET https://.../api/v1/accounts?withBalance=true
accept: application/json
```

consent-id: <Consent-ID>
x-request-id: <UUID für diesen Request>

2.2.10 RESPONSE – GET /ACCOUNTS

Content-Type: application/json
x-request-id: <UUID für diesen Request>

```
{ "accounts" : [ { ... } ] }
```

Die Antwort enthält die Kundenkonten. Diese Anfrage kann bis zum Ablauf der Einwilligung einmal täglich abgesetzt werden.