SUBSEMBLY
Banking Apps & APIs

BankAccessServer

FinTS XS2A PSD2 / SCA

Updatehinweise

Version 1.8.0

Subsembly GmbH

Hofmannstr. 7b
81379 München

http://subsembly.com
bas@subsembly.com

Stand: 10.09.2019

Vorgaben der PSD2 Richtline

Die PSD2 steht für Payment-Services-Directive 2 und ist eine EU-weite Regelung für Zahlungsdienstleister. In Deutschland ist diese Regelung als Gesetz zur Umsetzung der Zweiten Zahlungsdiensterichtlinie in nationales Recht umgesetzt. Alle Banken sind damit gesetzlich verpflichtet, bis zum Ablauf der Übergangsfrist am 14.09.2019 die Vorgaben aus der PSD2 zu erfüllen.

Die PSD2 umfasst eine Vielzahl neuer Regelungen, wobei nur einige davon die FinTS Schnittstelle betreffen und sich somit auf den BankAccessServer auswirken:

- Die Anforderungen an die verwendeten Sicherheitsverfahren wurden erhöht. Insbesondere wurden die Anforderungen an die 2-Faktor-Authentifizierung (sprich TAN-Verfahren) konkretisiert.
- Die Banken wurden verpflichtet, im Online-Banking mindestens alle 90 Tage eine starke Kundenauthentifizierung (SCA) durchführen.
- Die Banken müssen für Drittdienstleister (bankfremde Kontoinformationsdienste und Zahlungsauslösedienste) eine offene Schnittstelle (API) zur Verfügung stellen.

Um diese Anforderungen zu erfüllen, müssen die Banken alle Online-Banking-Systeme anpassen, teilweise alte TAN-Verfahren abschalten und neue TAN-Verfahren einführen.

Auch die FinTS-Schnittstelle der Banken wird von den Banken entsprechend erweitert und angepasst.

In der Vergangenheit war es einfach möglich, zwischen TAN-losen und TAN-pflichtigen Geschäftsvorfällen zu unterscheiden. So war die Anmeldung und der lesende Zugriff in der Regel nach der Angabe des Benutzernamens und der PIN möglich, ohne dass ein weiteres Sicherungsmerkmal (z.B. eine TAN / OTP) benötigt wurde. Weitere Sicherungsmerkmale waren prinzipiell nur für die Ausführung von Aufträgen notwendig. Für einen reinen Kontoinformationsdienst war es also seinerzeit nicht unbedingt notwendig, sich mit Sicherungsverfahren, TAN Medien und TAN Challenges auseinanderzusetzen.

Künftig - also ab dem 14.09.2019 - kann es bereits bei der Anmeldung und lesenden Zugriffen erforderlich sein, eine starke Kundenauthentifizierung vornehmen zu müssen. Je nach Kreditinstitut kann die SCA entweder bei jeder Anmeldung oder nur in bestimmten Abständen (maximal alle 90 Tage) notwendig werden.

Hierdurch sind auch bei der Nutzung der FinTS Schnittstelle über den BankAccessServer entsprechende Anpassungen notwendig, die nachfolgend beschrieben werden.

Bitte beachten Sie: Ab dem 14.09.2019 werden FinTS-Zugriffe nicht mehr funktionieren, welche nicht explizit an die durch die PSD2 vorgegebenen FinTS-Änderungen angepasst wurden.

Änderungen wird es auch beim Abruf von Konto-/Kreditkartendaten geben, die z.B. über WebScraping geholt werden. Beim Zugriff über die Online Banking Portale wird in der Regel ebefalls eine SCA notwendig, die konzeptionell über den XS2A Endpoint abgewickelt werden kann.

FinTS Requests / Responses

Die FinTS Requests des BankAccessServer haben folgende Struktur, die sich auch durch die erforderlichen SCA-Änderungen nicht verändert:

"**RequestOptions**": {
...
},
"**Connection**": {
...
},
"**Orders**": [
{
...
}
],
"**ResponseOptions**": {
...
}

Die PSD2/SCA relevanten Änderungen haben wir innerhalb der Connection vorgenommen und insbesondere den Aufbau der Orders unverändert gelassen.

Die nachfolgenden Szenarien können über die entsprechenden Connection-Settings abgebildet werden.

Erstmalige Anmeldung eines neuen Benutzers

Anmeldung über Benutzername/PIN, ohne dass das Sicherungsverfahren bekannt ist.

```
"Connection": {
        "LogOn": {
                "Contact": {
                        "BankCode": "{{BankCode}}",
                        "UserID": "{{UserID}}"
                },
                "Pin": "{{Pin}}"
        }
        "Suspend": true
}
```

Sofern der Benutzer mehr als ein TAN Verfahren besitzt, muss dieses nachfolgend spezifiziert werden, um eine eindeutige Auswahl zu treffen. Die aktuelle Session wird für folgende Aufrufe weiterverwendet und über den Service dargestellt. Alle weiteren Aufrufe müssen wiederum die zuvor gesendeten Orders beinhalten.

Die Anmeldung enthält in diesem Fall in der Response das Feld **NeedSecurityFunction**, das wiederum alle Sicherheitsverfahren des Kunden enthält.

Der Anwender kann nunmehr die entsprechende Security Function auswählen und den Request wie folgt absenden:

```
"Connection": {
        "SetSecurityFunction": {
               "Service": "{{Service}}",
              "SecurityFunction": {{SelectedSecurityFunction}}
        },
       "Suspend": true
}
```

Die Auswahl der Security Function erfolgt über den dreistelligen numerischen Wert, z.B. 901.

Weiterhin kann es notwendig sein ein TAN Medium zu spezifizieren. Das kann z.B. notwendig sein wenn der Kunde ein SMS oder App TAN Verfahren auf mehreren Endgeräten nutzt. In diesem Fall enthält die Antwort auf die Anmeldung das Feld **NeedTanMediaName**, das die verfügbaren TAN-Medien beinhaltet.

Das zu verwendende TAN Medium wird wie folgt festgelegt und muss wiederum die zuvor gesendeten Orders beinhalten.

```
"Connection": {
        "SetTanMediaName": {
               "Service": "{{Service}}",
              "TanMediaName": {{SelectedTanMediaName}}
        },
       "Suspend": true
}
```

Sofern im Rahmen der Anmeldung eine TAN notwendig ist, enthält die Antwort das Feld **NeedTan** mit der zugehörigen TAN-Challenge.

```
"Challenge": "Die mobileTAN zu diesem Auftrag wird als SMS an Ihre
Mobil-Telefonnummer gesendet, die bei Ihrer Bank registriert ist.",
```

Eine TAN kann wie folgt übertragen werden und muss wiederum die zuvor gesendeten Orders beinhalten.:

```
"Connection": {
         "SendTan": {
                 "Service": "{{Service}}",
                "Tan": {{TAN}}
         },
        "Suspend": true
}
```

Wie dargestellt, kann eine SCA basierte Anmeldung im Worst Case aus vier Schritten bestehen. Somit sollte bei wiederkehrenden Benutzer Anmeldungen immer mit einem vollständigen Kontakt vorgenommen werden.

Beispiel:

```
"Connection": {
        "LogOn": {
                "ContactSerialized": "{{SerializedContact}}",
                     "Pin": "{{Pin}}"
                 },
        "Suspend": true
}
```

Der serialisierte FinTS Kontakt kann in der Response zurückgegeben werden, in dem in den Response Options der Wert Contact auf true gesetzt wird.

```
"ResponseOptions": {
        "Trace": false,
        "Contact": true
}
```

Bei Instituten, die nicht bei jedem Login eine SCA verlangen, können über diesen Weg unnötige Schritte vermieden werden.

Nach einer Anmeldung ist im Verlauf der Session immer sinnvoll, über den Service (entspricht dem aktiven FinTS Dialog) zu gehen und diesen am Ende der Session sauber zu beenden. Der Service wird immer dann zurückgeliefert, wenn in der Connection **Suspend=true** angegeben wird.

```
"Connection": {
        "Resume": {
                "Service": "{{Service}}"
        },
        "Suspend": true
},
```

Der FinTS Dialog wird beendet, indem Suspend auf false gesetzt wird.

Weiterhin ist zu beachten, dass alle FinTS Anfragen mit einer gültigen Produktkennung / Version vorgenommen werden.

Hintergrund: Zur Erfüllung der PSD2-Vorgaben bzgl. der Transparenz über die kundenseitig eingesetzte Software hat die Deutsche Kreditwirtschaft einen **Prozess zur Registrierung von FinTS-Produkten** etabliert, um gegenüber den Kunden Informationen zur FinTS-Nutzung nachweisen zu können. Ab dem **01.12.2018 sollen** zugeteilte FinTS-Registrierungsnummern in der Dialoginitialisierung im Element Produktbezeichnung gesendet werden. **Vor dem 1.12.2018 ist die Verwendung nicht erlaubt**.

Weitere Informationen zur FinTS Produktregistrierung finden Sie auf:
https://www.hbci-zka.de/register/register_faq.htm

Hierzu kann, sofern der BankAccessServer immer mit derselben Kennung arbeitet, der in den appsettings konfigurierte Wert verwendet werden.

```
"BasConfig": {
        ...
        "ProductName": "",
        "ProductVersion": ""
}
```

Sofern dynamische Produktkennungen verwendet werden sollen, können diese innerhalb der Connection -> LogOn spezifiziert werden.

Spezielle Testeinstellungen für die starke Kundenauthentifizierung:

Innerhalb des Connection->LogOn->Contact-StrongCustomerAuthentication kann das SCA Verhalten definiert werden. Folgende Werte sind zulässig:

- Default - keine SCA bis 14.09.2019, danach mit SCA
- RequestSCA - SCA aktiviert (hilfreich um im Vorfeld gegen Server mit SCA testen zu können)
- NoSCA - ggfs. sinnvoll wenn nach dem 14.09.2019 Server noch kein SCA aktiviert haben

```
 "Connection": {
        "LogOn": {
            "Contact": {
                "BankCode": "{{BankCode}}",
                "UserID": "{{UserID}}",
                "StrongCustomerAuthentication":"RequestSCA"
            },
            "Pin": "{{Pin}}"
        },
        "Suspend": true
    }
```

XS2A Requests / Responses

Die grundsätzliche Struktur der XS2A hat sich auch die Einführung der PSD2 / SCA nicht verändert und sieht wie folgt aus:

Die grundsätzliche Struktur des Requests entspricht folgendem Aufbau, wobei nicht alle gezeigten Elemente verpflichtend oder in dieser Kombination gültig sind.

```
{
    "RequestOptions": {
    },

    "Connection": {
        "Service": {
        },
        "Resume": {
        },
        "Credentials": {
        },
        "Suspend": ...
    },
    "Orders": [
    ],
    "ResponseOptions": {
    }
}
```

Die typische Anmeldung mit UserID / Password für den Xs2aContactInfoRequest sieht wie folgt aus:

```
"Connection": {
        "Service": {
            "Account": {{Xs2aAcctCRDC}}
        },
        "Credentials": {
            "UserID": "{{Xs2aUserIDCRDC}}",
            "Password": "{{Xs2aPasswordCRDC}}"
        }
    }
```

Sofern der Anbieter keine SCA erfordert beinhaltet die Antwort bereits die Xs2aContactInfoResponse.

Ist eine SCA erforderlich, so gibt es zwei Antworttypen, die beide eine Session-Referenz beinhalten:

1. NeedQueryResponse - es ist eine weitere Auswahl für die Art der SCA, z.B. des TAN Mediums, erforderlich
2. NeedChallengeResponse - Informationen zur geforderten SCA/TAN Challenge

Beispielablauf: NeedQueryResponse mit Auswahl des TAN2go Endgeräts

```
"NeedQueryResponse": {
        "QueryFieldName": "TanMediaIndex",
        "QueryHeading": "TAN2go Endgerät auswählen",
        "QueryPrompt": "Bitte senden Sie die TAN2go an folgendes Endgerät:",
        "QueryChoices": [
            {
                "Value": "0",
                "Text": "GooglePixel",
                "Default": false
            },
            {
                "Value": "1",
                "Text": "Iphone",
                "Default": false
            }
        ]
    },
    "Session": "63d0b022-8a60-4106-9d32-4394a7bc52ee"
```

In diesem Fall muss im folgenden Request der ausgewählte Wert, der Feldname und die zurückgelieferte Session angegeben werden

```
"Connection": {
        "SetQueryResponse": {
            "Session": "{{xs2aSession}}"
        },
        "Credentials": {
            "UserID": "{{Xs2aUserIDCRDC}}",
            "Password": "{{Xs2aPasswordCRDC}}",
            "QueryResponseFieldName": "TanMediaIndex",
            "QueryResponseFieldValue":"1"
        }
    }
```

Die Antwort enthält nunmehr den ChallengeType, sofern verfügbar die ChallengeData (z.B. bei ChipTAN), den DisplayText mit Hinweisen für den Benutzer sowie die Session Referenz.

```
"NeedChallengeResponse": {
       "ChallengeType": "SimpleTAN",
       "DisplayText": "Bitte bestätigen Sie die Anmeldung mit der TAN aus Ihrer
DKB-TAN2go-App."
   }
```

Im letzten Schritt erfolgt die Übertragung der ChallengeResponse, auf die dann mit der Xs2aContactInfo Response geantwortet wird.

```
"Connection": {
        "SetChallengeResponse": {
            "Session": "{{xs2aSession}}"
        },
        "Credentials": {
             "ChallengeResponse": "187876"
        },
        "Suspend": false
    }
```

Weitere Informationen

Subsembly BankAccessServer

Webseite: https://subsembly.com/bank-access-server.html

Produktinformationen: https://subsembly.com/download/BankAccessServer.pdf

API Spezifikation: https://subsembly.com/download/BankAccessServerClientInterface.pdf

Installation: https://subsembly.com/download/BankAccessServerInstallation.pdf

Subsembly Banking APIs / SDKs

FinTS API: https://subsembly.com/sepa-api.html

Ebics API: https://subsembly.com/ebics-api.html

SEPA API: https://subsembly.com/sepa-api.html

XS2A API: https://subsembly.com/xs2a-api.html

Spezifikationen

Subsembly Payments Datenformate (SUPA): https://subsembly.com/supa.html

Deutsche Kreditwirtschaft / EBICS:
https://die-dk.de/zahlungsverkehr/electronic-banking/dfu-verfahren-ebics/

Deutsche Kreditwirtschaft / FinTS:

https://die-dk.de/zahlungsverkehr/electronic-banking/fints/

Deutsche Kreditwirtschaft / PSD2 Kontoschnittstelle:

https://die-dk.de/zahlungsverkehr/electronic-banking/psd2-kontoschnittstelle/

NextGenPSD2 Access to Account Interoperability Framework

PSD2 Access to Bank Accounts

https://www.berlin-group.org/psd2-access-to-bank-accounts

Laufzeitumgebung

Microsoft .NET Core: https://dotnet.microsoft.com/download/dotnet-core

Codegenerierung

Swagger CodeGen: http://swagger.io/swagger-codegen/