

# Subsembly BankAccessServer EBICS

## Lösungsbeschreibung

Version 1.2

---

Dieses Dokument ist urheberrechtlich geschützt. Eine Weitergabe des Dokuments oder Auszügen daraus darf, egal in welcher Form, nur mit ausdrücklicher Genehmigung der Subsembly GmbH erfolgen. Die Übergabe des Dokuments begründet keinen Anspruch auf Lizenz.

Es wurden alle Anstrengungen unternommen um die Richtigkeit des Dokuments sicher zu stellen. Subsembly GmbH übernimmt jedoch keine Garantie hinsichtlich der Richtigkeit oder Vollständigkeit. Die Tauglichkeit oder Eignung für einen bestimmten Zweck wird nicht gewährleistet. Die enthaltenen Informationen können ohne besondere Ankündigung geändert werden. Ein Rechtsanspruch ist hieraus nicht ableitbar.

Copyright © 2025 Subsembly GmbH.

## Inhaltsverzeichnis

|  |    |
|--|----|
| 1 Einführung.....                          | 4  |
| 2 Systemarchitektur.....                   | 6  |
| 3 EBICS Zugänge einrichten .....           | 8  |
| 4 Bankfachliche Funktionen.....            | 13 |
| 4.1 Umsatzabruf.....                       | 13 |
| 4.2 Zahlungsaufträge einreichen .....      | 16 |
| 4.3 VoP-Dokumente abrufen .....            | 18 |
| 4.4 VoP-Informationen in VEU .....         | 19 |
| 4.5 Abruf der Kundenprotokolle.....        | 19 |
| 4.6 EBICS Geschäftsvorfälle.....           | 19 |
| 5 EBICS Spooler.....                       | 21 |
| 6 Weitere Informationen.....               | 23 |
| 6.1 Subsembly BankAccessServer.....        | 23 |
| 6.2 Subsembly BankingVEU.....              | 23 |
| 6.3 Subsembly EBICS API.....               | 24 |
| 6.4 Subsembly EBICS Server.....            | 24 |
| 6.5 Subsembly EBICS Dummy.....             | 25 |
| 6.6 Subsembly BankingVEU App Server.....   | 25 |
| 6.7 Allgemeine Informationen zu EBICS..... | 26 |

**Versionshistorie**

| <b>Version</b> | <b>Datum</b> | <b>Autor</b> | <b>Beschreibung</b>                           |
|----------------|--------------|--------------|---|
| 1.0            | 05.10.2022   | Klaus Igel   | Initiale Dokumentation                        |
| 1.1            | 26.09.2024   | Klaus Igel   | Beschreibung weiterer EBICS Geschäftsvorfälle |
| 1.2            | 01.09.2025   | Klaus Igel   | Ergänzung um Verification of Payee            |

# 1 Einführung

Die EBICS-Schnittstelle wurde 2006 von den Deutschen Kreditwirtschaft zur Abwicklung des Zahlungsverkehrs mit Firmenkunden eingeführt. Die Kommunikation bei EBICS erfolgt über das Internet unter Verwendung asymmetrischer kryptographischer Verfahren.

EBICS kann bei nahezu allen Kreditinstituten und Kunden einheitlich genutzt werden und ermöglicht somit ein standardisiertes Multibanking, das für eine automatisierte Verwendung ohne PIN und TAN auskommt.

Aus fachlicher Sicht können über das EBICS Protokoll folgende Bereiche abgedeckt werden:

- Einrichtung und Administration von EBICS Zugängen
- Abrufen von Kontoinformationen, z.B. Kontoauszüge, Vormerkposten, Protokolle
- Ausführen von Zahlungen, z.B. Sammelüberweisungen und Sammellastschriften

EBICS ist für den Massenzahlungsverkehr hervorragend geeignet, da einerseits große Datenmengen problemlos verarbeitet sowie andererseits durch ein mehrstufiges Unterschriftskonzept und die standortunabhängige Auftragsfreigabe auch externe Dienstleister problemlos in die Abwicklung integriert werden können.

Ab Version 4.0.0 ist der BankAccessServer EBICS vollständig VoP-konform und ermöglicht eine flexible Steuerung, ob Zahlungen mit oder ohne Verification of Payee (VoP) ausgeführt werden. Anwender können dabei wählen, ob ein Auftrag direkt eingereicht wird (CCT/CIP) oder ob er zuvor einer VoP-Prüfung unterzogen wird (CTV/CIV). Der Geschäftsvorfall VPZ ermöglicht es, VoP-Dokumente gezielt abzurufen. Darüber hinaus enthalten VEU-Aufträge, sofern eine VoP-Prüfung durchgeführt wurde, automatisch auch das entsprechende Ergebnis der Prüfung.

An dieser Stelle setzt auch der Subsembly BankAccessServer EBICS an und wendet sich insbesondere an Dienstleister, die Zahlungsverkehrslösungen in eigene (cloudbasierte) Produkte für ihre Kunden integrieren möchten.

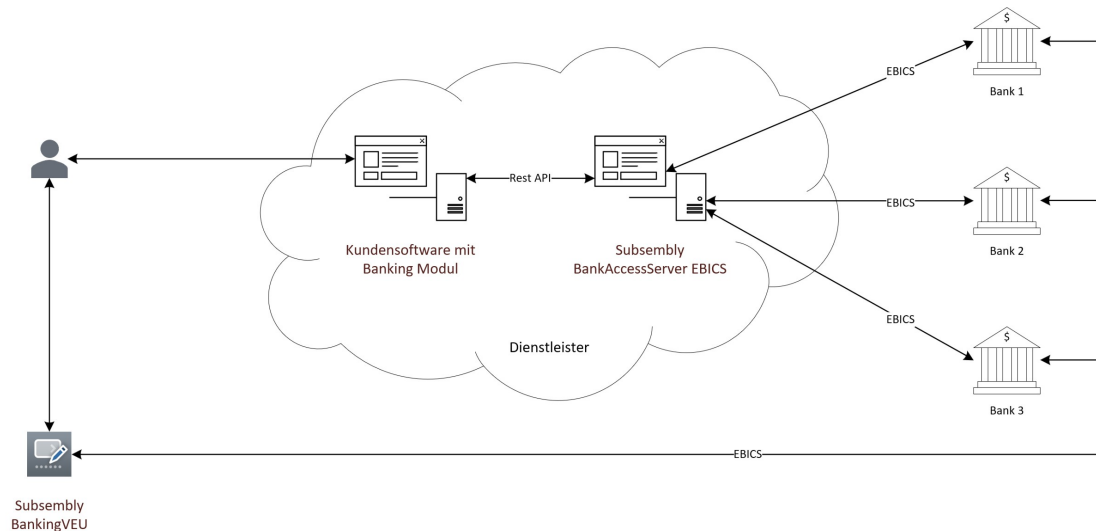
Die einfach zu verwendende REST API des BankAccessServers ermöglicht dabei die Einsatz in unterschiedlichsten Plattformen und Szenarien und befreit Entwickler davon, sich mit der Komplexität des EBICS Protokolls auseinandersetzen zu müssen.

Ein vollumfängliches Lösungsszenario wird auch durch die weiteren Produktbausteine der Subsembly abgedeckt.

- Mit der für Android und iOS verfügbaren App Subsembly **BankingVEU** können die über den BankAccessServer und andere Rechenzentren (z.B. Datev) vorbereiteten Zahlungsaufträge standortunabhängig freigegeben oder storniert werden.
- Der Subsembly **EBICS Dummy** ermöglicht es Entwicklern gegen eine bereitgestellte Testbank zu programmieren, ohne dass hierzu echte Bankzugänge benötigt werden. Dabei werden auch komplexe Vorgänge, wie die Einrichtung eines EBICS Zugangs, unterstützt.
- Der Subsembly **EBICS Server** ist ein plattformneutrales Servermodul für die Implementierung des EBICS Standards bei Kreditinstituten/Zahlungsdienstleistern.

## 2 Systemarchitektur

Das Zusammenspiel der einzelnen Systeme ist in folgendem Bild dargestellt.



In dem obigen Beispiel gibt es folgende Rollen:

- **Benutzer**, der die Softwarelösung des Dienstleisters verwendet und hierüber auch Banking-Funktionalität nutzt. Ferner gibt dieser über die Subsembly BankingVEU App zuvor eingestellte Zahlungsaufträge frei.
- **Dienstleister**, der die (cloudbasierte) Lösung seinen Kunden bereitstellt. In der Lösung werden Banking Funktionen, wie z.B. der Umsatzabruf oder die Einstellung von Zahlungsaufträgen, angeboten. Hierfür wird der Subsembly BankAccessServer EBICS angebunden, der die EBICS-seitige Kommunikation zu den einzelnen Banken übernimmt.
- **Bank**, bei der der Kunde (Benutzer) sein Konto führt. Die Bank ermöglicht den Kontozugriff über die EBICS Schnittstelle. Für die Bereitstellung des EBICS Zugriffs wird ein gesonderter Vertrag zwischen der Bank und dem Kunden geschlossen.

**Beispielszenario:** Nehmen wir an, dass der Dienstleister eine cloudbasierte Software für Vereine anbietet, mit der die folgenden bankfachlichen Funktionen abgebildet werden können:

- Einreichung von Lastschriften zum Einzug von Mitgliedsbeiträgen
- Einreichung von Sammelüberweisungen für Gehaltszahlungen
- Umsatzabruf für den Abgleich der offenen Posten

Hierbei soll sichergestellt werden, dass der Dienstleister über seine Software lediglich die Aufträge bei der Bank einreichen und den Umsatzabruf vornehmen kann, jedoch soll keine Freigabe von Zahlungen möglich sein. Die Auftragsfreigabe würde somit weiterhin durch den Kunden erfolgen.

Sofern der Dienstleister nicht als Servicerechenzentrum (siehe: <https://die-dk.de/zahlungsverkehr/electronic-banking/datenaustausch-dfu/>) fungiert, kann der Kunde bei seiner Bank folgende EBICS Zugänge beantragen:

- EBICS Zugang mit reiner Transport-Berechtigung für den Dienstleister.
- EBICS Zugang mit Zeichnungsberechtigung (einzeln oder 4 Augen Prinzip) für „sich selbst“, also für entsprechend berechtigte Mitarbeiter.

Über die Software des Dienstleisters kann nunmehr der Zugang mit Transport-Berechtigung eingerichtet werden. Die EBICS-seitige Kommunikation mit der Bank erfolgt dabei über den BankAccessServer EBICS, der seinerseits über eine einfache Rest-Schnittstelle angesprochen werden kann. Der Einrichtungsprozess wird nachfolgend noch detailliert beschrieben.

Für die Auftragsfreigabe wird in der VEU App noch der entsprechende Zugang mit Zeichnungsberechtigung eingerichtet. Der Dienstleister ist in diesen Prozess nicht direkt involviert, kann jedoch nach der Einreichung von Zahlungsaufträgen entsprechende Push Nachrichten auslösen, so dass der App Benutzer aktiv über freizugebende Aufträge benachrichtigt wird.

### 3 EBICS Zugänge einrichten

Wie zuvor erwähnt richtet der Kunde über die angebotene Softwarelösung des Dienstleisters die EBICS Zugänge ein. Die Speicherung der Zugänge mit der reinen Transport-Berechtigung kann wahlweise im gesicherten Dateisystem durch den BankAccessServer oder direkt beim Dienstleister erfolgen, z.B. in einer relationalen Datenbank.

Nachdem der Kunde von seiner Bank die Teilnehmerdaten erhalten hat, kann der technische Einrichtungsprozess gestartet werden. Grundsätzlich müssen für die Einrichtung folgende Informationen vorliegen:

- Host-ID des EBICS-Banksystems
- URL des EBICS-Banksystems
- Teilnehmerdaten (Partner- und Kunden-Id)
- Vom Kontoinhaber unterschriebene Initialisierungsbriefe für die öffentlichen bankfachlichen Schlüssel (INI)
- Vom Kontoinhaber unterschriebene Initialisierungsbriefe für die öffentlichen Authentifikations- sowie Verschlüsselungsschlüssel (HIA)
- Informationen über das TLS-Serverzertifikat des EBICS-Banksystems
- Hashwerte der öffentlichen Schlüssel des EBICS-Banksystems

Der Prozess beginnt mit der **Initialisierung des Bankzugangs**, womit ein neuer EBICS Kontakt angelegt wird, der in allen folgenden EBICS Requests benötigt wird.

Über die „ContactOption“ kann festgelegt werden, wo die Speicherung erfolgen soll. Es stehen dabei die Werte „BAS“ für den BankAccessServer und „CON“ für die Speicherung in externen Systemen zur Verfügung. Sofern eine externe Verarbeitung von EBICS Kontakten erfolgen soll, muss aus der Antwort der Base64 kodierte Wert aus dem Feld „ContactSerialized“ gespeichert werden.

API Aufruf:

```
{
  "RequestOptions": {
    "AccessToken": "<AccessToken>"
  },
  "Orders": [
    {
      "Contact": {
        "HostId": "<wird von der Bank bereitgestellt>",
        "PartnerId": "<wird von der Bank bereitgestellt>",
        "UserId": "<wird von der Bank bereitgestellt>",
        "HostURL": "<wird von der Bank bereitgestellt>"
      },
      "KeyPassword": "<Schlüsselwort des Teilnehmers>",
      "ContactOption": "BAS|CON",
      "Type": "EbicsAdminInitRequest"
    }
  ]
}
```

Die folgenden Aufrufe verwenden die Syntax für die im BankAccessServer gespeicherten Kontakte:

```
"Connection": {
  "ContactRef": {
    "HostId": "<wird von der Bank bereitgestellt>",
    "PartnerId": "<wird von der Bank bereitgestellt>",
    "UserId": "<wird von der Bank bereitgestellt>",
    "HostURL": "<wird von der Bank bereitgestellt>"
  },
  "KeyPassword": "<Schlüsselwort des Teilnehmers>"
}
```

Alternativ kann bei einer Speicherung in externen Systeme auch bei jedem Request der komplette EBICS Kontakt mitgeschickt werden:

```
"Connection": {
  "ContactSerialized": "<Base64 kodierter EBICS Kontakt>",
  "KeyPassword": "<Schlüsselwort des Teilnehmers>"
}
```

Nach dem Erhalt der Bankparameter vom beteiligten Kreditinstitut erfolgt die Initialisierung des Teilnehmer auf dem EBICS-System. Die Initialisierung erfolgt mit den Auftragsarten "INI" und "HIA", womit der Signatur- und Authentifizierungsschlüssel an die Bank übertragen wird.

API Aufruf:

```
{
  "RequestOptions": {
    "AccessToken": "<AccessToken>"
  },
  "Connection": {
    "ContactRef": {
      "HostId": "<wird von der Bank bereitgestellt>",
      "PartnerId": "<wird von der Bank bereitgestellt>",
      "UserId": "<wird von der Bank bereitgestellt>",
      "HostURL": "<wird von der Bank bereitgestellt>"
    },
    "KeyPassword": "<Schlüsselpasswort des Teilnehmers>"
  },
  "Orders": [
    {
      "Type": "EbicsAdminHIARequest"
    },
    {
      "Type": "EbicsAdminINIRequest"
    }
  ]
}
```

Anschließend muss ein vom Kunden unterschriebener Initialisierungsbrief für die öffentlichen bankfachlichen Schlüssel (INI) und die öffentlichen Authentifikations- sowie Verschlüsselungsschlüssel (HIA) an die Bank gesendet werden.

Die Erstellung des Briefs im PDF Format kann über den BankAccessServer EBICS durch folgenden Request vorgenommen werden:

API Aufruf:

```
{
  "RequestOptions": {
    "AccessToken": "<AccessToken>"
  },
  "Connection": {
    "ContactRef": {
      "HostId": "<wird von der Bank bereitgestellt>",
      "PartnerId": "<wird von der Bank bereitgestellt>",
      "UserId": "<wird von der Bank bereitgestellt>",
      "HostURL": "<wird von der Bank bereitgestellt>"
    },
    "KeyPassword": "<Schlüsselpassword des Teilnehmers>"
  },
  "Orders": [
    {
      "Type": "EbicsAdminGenerateINILetterRequest"
    }
  ]
}
```

Nach Zusendung und Prüfung des Initialisierungsbrief nimmt die Bank umgehend die Freischaltung vor. Nach dem erfolgreichen Abruf der öffentlichen Bankschlüssel über die Auftragsart "HPB" kann der Kunde die vereinbarten EBICS Geschäftsvorfälle bei seiner Bank nutzen.

API Aufruf:

```
{
  "RequestOptions": {
    "AccessToken": "<AccessToken>"
  },
  "Connection": {
    "ContactRef": {
      "HostId": "<wird von der Bank bereitgestellt>",
      "PartnerId": "<wird von der Bank bereitgestellt>",
      "UserId": "<wird von der Bank bereitgestellt>",
      "HostURL": "<wird von der Bank bereitgestellt>"
    },
    "KeyPassword": "<Schlüsselpasswort des Teilnehmers>"
  },
  "Orders": [
    {
      "Type": "EbicsAdminHPBRequest"
    }
  ]
}
```

Die Antwort auf den Request enthält auch die Hashwerte der Bankschlüssel, so dass an dieser Stelle ein Vergleich mit den zuvor von der Bank erhaltenen Werte erfolgen kann.

## 4 Bankfachliche Funktionen

In diesem Kapitel wird die Ausführung ausgewählter bankfachlicher Operationen über den BankAccessServer beschrieben.

### 4.1 Umsatzabruf

EBICS-seitig werden die Abholung von Tagesauszügen und Vormerkposten unterstützt. In der Vereinbarung mit der Bank wird festgelegt, welche Auftragsarten beim Abruf verwendet werden können.

In der Praxis kann der Abruf bei einer Bank im CAMT und/oder SWIFT Format erfolgen. Hierzu stehen die folgenden EBICS-Auftragsarten zur Verfügung:

- C52 - Abholen der Vormerkposten im CAMT Format
- C53 - Abholen der Kontoauszüge im CAMT Format
- STA - Abholen der Kontoauszüge im SWIFT MT940 Format
- VMK - Abholen der Vormerkposten im SWIFT MT942 Format

Der BankAccessServer EBICS kann wahlweise die Daten im Rohformat, also exakt wie von der Bank bereitgestellt, oder in der JSON Notation zurückliefern. Zusätzlich stehen interne Konverter zur Verfügung, so dass beispielsweise eine Client Anwendung die Daten immer im CAMT Format anfordern kann, auch wenn die Bank selbst nur das SWIFT Format unterstützt.

Die Antwort für die Request-Typen EbicsOrderC52Request, EbicsOrderC53Request, EbicsOrderSTARrequest, EbicsOrderVMKRequest enthält die jeweiligen Daten in der JSON-Notation.

API Aufruf zum Abholen der Tagesauszüge im CAMT Format / JSON Notation:

```
{
  "RequestOptions": {
    "AccessToken": "<AccessToken>"
  },
  "Connection": {
    "ContactRef": {
      "HostId": "<wird von der Bank bereitgestellt>",
      "PartnerId": "<wird von der Bank bereitgestellt>",
      "UserId": "<wird von der Bank bereitgestellt>",
      "HostURL": "<wird von der Bank bereitgestellt>"
    },
    "KeyPassword": "<Schlüsselwort des Teilnehmers>"
  },
  "Orders": [
    {
      "DateTimeFrom": "2022-09-20",
      "Type": "EbicsOrderC53Request"
    }
  ]
}
```

Der generische EbicsOrderGenericRequest stellt hingegen in der Antwort die Rohdaten bereit.

API Aufruf zum Abholen der Tagesauszüge über den generischen Download Request:

```
{
  "RequestOptions": {
    "AccessToken": "<AccessToken>"
  },
  "Connection": {
    "ContactRef": {
      "HostId": "<wird von der Bank bereitgestellt>",
      "PartnerId": "<wird von der Bank bereitgestellt>",
      "UserId": "<wird von der Bank bereitgestellt>",
      "HostURL": "<wird von der Bank bereitgestellt>"
    },
    "KeyPassword": "<Schlüsselpasswort des Teilnehmers>"
  },
  "Orders": [
    {
      "Type": "EbicsOrderGenericRequest",
      "OrderType": "C53",
      "Upload": false,
      "OrderAttribute": "DZHNN",
      "OrderParams": {
        "Type": "EbicsStandardOrderParamsRS",
        "DateTimeFrom": "2022-09-20"
      }
    }
  ]
}
```

## 4.2 Zahlungsaufträge einreichen

Der BankAccessServer EBICS ermöglicht die Einreichung von Sammelaufträgen (Sammellastschriften/Sammelüberweisungen) und bietet hierzu für eine bestmögliche Unterstützung der Clientsoftware unterschiedliche Requests an:

- Der **EbicsOrderSepaRequest** nimmt alle gültigen Sepa Dokumente wahlweise im JSON oder XML-Format entgegen. Sofern die Clientsoftware bereits gültige Sepa Dokumente im XML Format erzeugt, können diese einfach als Base64 kodierte Auftragsdaten übergeben werden.
- Bei dem generische **EbicsOrderGenericRequest** **müssen** neben der EBICS Auftragsart (z.B. CCT) ebenfalls die Auftragsdaten im XML Format (Base64 kodiert) übermittelt werden.

VoP-Steuerung (EBICS):

- Direkt-Einreichung ohne VoP: OrderType = CCT (SEPA Credit Transfer), CIP (Instant)
- Einreichung mit VoP-Prüfung: OrderType = CTV (SEPA Credit Transfer with VoP), CIV (Instant mit VoP)
- Die Ergebnisse der VoP-Prüfung können über den OrderType VPZ abgerufen werden bzw. stehen im Rahmen der VEU-Aufträge zur Verfügung.

API Aufrufe:

**EbicsOrderSepaRequest** im JSON Format:

```
{
  "RequestOptions": {
    "AccessToken": "<AccessToken>"
  },
  "Connection": {
    "ContactRef": {
      ...
    },
    "KeyPassword": "<Schlüsselwort des Teilnehmers>"
  },
  "Orders": [
    {
```

```
    "Type": "EbicsOrderSepaRequest",
    "SepaDocument": {
      <SepaDocument im JSON Format>
    }
  }
]
```

### **EbicsOrderSepaRequest** im XML Format

```
{
  "RequestOptions": {
    "AccessToken": "<AccessToken>"
  },
  "Connection": {
    "ContactRef": {
      ...
    },
    "KeyPassword": "<Schlüsselwort des Teilnehmers>"
  },
  "Orders": [
    {
      "Type": "EbicsOrderSepaRequest",
      "SepaDocumentSerialized": "SepaDocument im XML Format, Base64 encoded"
    }
  ]
}
```

## EbicsOrderGenericRequest

```
{
  "RequestOptions": {
    "AccessToken": "<AccessToken>"
  },
  "Connection": {
    "ContactRef": {
      ...
    },
    "KeyPassword": "<Schlüsselpasswort des Teilnehmers>"
  },
  "Orders": [
    {
      "Type": "EbicsOrderGenericRequest",
      "OrderType": "CCT",
      "Upload": true,
      "OrderAttribute": "OZHNN",
      "OrderData": "SepaDocument im XML Format, Base64 encoded",
      "OrderParams": {
        "Type": "EbicsStandardOrderParamsRS"
      }
    }
  ]
}
```

## 4.3 VoP-Dokumente abrufen

Mit dem Geschäftsvorfall VPZ können VoP-Dokumente (pain.002-basiert) gezielt abgefragt werden.

```
"Orders": [
  {
    "Type": "EbicsOrderVPZRequest",
    "DateTimeFrom": "{{StartDate}}"
  }
]
```

Die Antwort enthält gruppen- und transaktionsbezogene VoP-Status (z. B. GroupStatus = RVCM; RCVC/RVMC/RVNM pro Transaktion).

## 4.4 VoP-Informationen in VEU

Bei Aufträgen, die mit CTV/CIV eingereicht wurden, enthalten VEU-Details (HVT) die VoP-Ergebnisse pro Posten.

Beispielauszug (gekürzt): "TypedDescriptions": [{"Type": "VoPStatus", "Description": "RVCM" }],  
"VopStatusCode": "RVCM", optional "VopCloseMatchName".

## 4.5 Abruf der Kundenprotokolle

Von einer erfolgreichen Übertragung der eingereichten Aufträge kann erst ausgegangen werden, wenn über das Kundenprotokoll die erfolgreiche Einlieferung und Unterschriftenprüfung angezeigt wird. Somit empfiehlt es sich, den Abruf des Kundenprotokolls zeitnah nach der Ausführung eines Auftrags durchzuführen.

Für eine eindeutige Identifikation des Geschäftsvorfalles wird in der jeweiligen Antwort eine "EbicsOrderID" bereitgestellt, die Ihrerseits auch Einträge im Protokoll kennzeichnet. Je nach Institut kann das Protokoll im Text- (PTK) oder XML-Format (HAC) abgerufen werden.

## 4.6 EBICS Geschäftsvorfälle

Der BankAccessServer EBICS ist grundsätzlich in der Lage alle zwischen dem Kunden und seiner Bank vereinbarten Geschäftsvorfälle durchzuführen. Wie in den beiden vorherigen Abschnitten bereits angesprochen, unterstützt der BankAccessServer die wichtigsten Auftragsarten in Form von

individuellen Requests mit einem entsprechendem Mapping der Klassen und Bereitstellung der Daten im JSON Format.

Alle weiteren EBICS Auftragsarten können über den EbicsOrderGenericRequest unter Angabe des OrderTypes, der Information ob es sich um einen Abhol- oder Sendeauftrag handelt (Upload: true | false) und der jeweiligen Auftragsdaten/-parameter durchgeführt werden.

In der nachfolgenden Tabelle sind einige relevante Geschäftsvorfälle auf Basis EBICS 2.5 aufgelistet:

| <b>Geschäftsvorfall</b>                   | <b>Upload / Download</b> | <b>Auftragsart</b> |
|---|--------------------------|--------------------|
| PDF Tagesauszüge abrufen                  | D                        | BKA                |
| SEPA-Basis-Lastschrift einreichen         | U                        | CDD, CDC           |
| SEPA-Firmen-Lastschrift einreichen        | U                        | CDB, C2C           |
| SEPA-Überweisungen ohne VoP-Prüfung       | U                        | CCT, CCC           |
| SEPA-Echtzeitüberweisung ohne VoP-Prüfung | U                        | CIP                |
| SEPA-Überweisungen mit VoP-Prüfung        | U                        | CTV                |
| SEPA-Echtzeitüberweisung mit VoP-Prüfung  | U                        | CIV                |
| VoP-Dokumente abrufen                     | D                        | VPZ                |
| Tagesauszüge abrufen                      | D                        | C53, STA           |
| Vormerkposten abrufen                     | D                        | C52, VMK           |
| Kundenprotokoll abrufen                   | D                        | HAC, PTK           |
| VEU Status abrufen                        | D                        | HVD                |
| VEU Unterschrift hinzufügen               | U                        | HVE                |
| VEU Storno                                | U                        | HVS                |
| VEU Transaktionsdetails abrufen           | D                        | HVT                |
| VEU Übersicht abholen                     | D                        | HVU                |
| Senden der Teilnehmerschlüssel            | U                        | HIA, INI           |
| Abholen der öffentlichen Bankschlüssel    | D                        | HPB                |
| Teilnehmerdaten abrufen                   | D                        | HTD                |

## 5 EBICS Spooler

Bei den in den vorherigen Kapiteln beschriebenen EBICS Requests handelte es sich um synchron ausgeführte Aufrufe, bei denen ein Client immer solange warten muss, bis der Server die Antwort liefert.

Da EBICS Zugänge in der Praxis häufig für das Senden sehr großer Zahlungsaufträge oder dem Abruf sehr vieler Umsätze verwendet werden, kann die Ausführung auch eine längere Zeit in Anspruch nehmen und in der Folge zu HTTP Timeouts führen.

Für die Ausführung umfangreicher Aufträge bietet der BankAccessServer EBICS zusätzlich eine asynchrone Variante an, bei der die eigentlichen Aufträge im Hintergrund über den EBICS Spooler ausgeführt werden.

Bei der Verwendung des Ebics Spoolers werden die empfangenen / zu sendenden Dateien in der konfigurierten Verzeichnisstruktur gespeichert.

Der Ablauf sieht in diesem Fall wie folgt aus:

### Umsatzabruf:

- Der Client ruft die Tagesauszüge oder Vormerkposten ab und erhält in der Antwort eine EBICS Spooler Id.
- Im Hintergrund ruft der Ebics Spooler die Umsatzdaten ab und stellt die CAMT Dateien bereit.
- Der Client kann mit dem `EbicsOrderStatusSpoolerRequest` für die Ebics Spooler Id den Auftragsstatus abrufen.
- Im Anschluss kann mit dem `EbicsOrderGetFileNamesSpoolerRequest` eine Liste der bereitgestellten CAMT Dateien abgerufen werden.
- Der Abruf kann wahlweise im XML oder JSON Format über den `EbicsOrderGetFilesSpoolerRequest` erfolgen.

**Senden von Aufträgen:**

- Übertragen der Zahlungsdatei zum BankAccessServer EBICS via `EbicsOrderSubmitSpoolerRequest`.
- Der `EbicsOrderUploadSpoolerRequest` startet die Übertragung zum Bankserver und liefert in der Antwort eine EBICS Spooler Id zurück.
- Der Client kann mit dem `EbicsOrderStatusSpoolerRequest` für die Ebics Spooler Id den Auftragsstatus abrufen.
- Für einen ausgeführten Auftrag kann im Anschluss mit dem `EbicsOrderGetFileNamesSpoolerRequest` eine Liste der verarbeiteten Dateien abgerufen werden.
- Bei Bedarf Abruf der Inhalte wahlweise im XML oder JSON Format über den `EbicsOrderGetFilesSpoolerRequest`.

## 6 Weitere Informationen

Dieses Dokument gibt eine kurze Übersicht über die Einsatzmöglichkeiten des Subsembly BankAccessServer EBICS.

Nachfolgend haben wir weiterführende Informationen aufgeführt, die für die Arbeit mit dem BankAccessServer im Speziellen und mit EBICS im Allgemeinen von Interesse sein können.

### 6.1 Subsembly BankAccessServer

Der Subsembly BankAccessServer bietet neben EBICS weitere Endpoints für verschiedene Bankschnittstellen an, so dass in Summe mit allen für den jeweiligen Einsatzzweck benötigten Bankdaten gearbeitet werden kann. Hier seien z.B. das FinTS Protokoll, die PSD2 Schnittstellen, Screen Scraper oder native Banking Interfaces genannt.

Unter der URL <https://subsembly.com/bank-access-server.html> finden Sie neben der API Dokumentation, Produktbeschreibung auch Informationen zur Installation und zum Betrieb des Produkts.

Details und Beispiele zu CCT/CIP vs. CTV/CIV, VPZ sowie VoP-Statuscodes finden Sie hier:

VoP-Dokumentation DE: [https://subsembly.com/download/BankAccessServerVoP\\_de.pdf](https://subsembly.com/download/BankAccessServerVoP_de.pdf)

VoP-Dokumentation EN: [https://subsembly.com/download/BankAccessServerVoP\\_en.pdf](https://subsembly.com/download/BankAccessServerVoP_en.pdf)

### 6.2 Subsembly BankingVEU

Mit der für Android und iOS verfügbaren App Subsembly BankingVEU können die über den BankAccessServer und andere Rechenzentren (z.B. Datev) vorbereiteten Zahlungsaufträge standortunabhängig freigegeben oder storniert werden. Darüber hinaus besteht die Möglichkeit der

Push Benachrichtigung bei der Einreichung von Zahlungen über den Subsembly BankAccessServer EBICS oder BankingZV.

Eine Lösungsbeschreibung steht unter der URL: <https://subsembly.com/bankingveu.html> bereit. Neben der Nutzung der Standard-App bieten wir die Erstellung entsprechender White Label Versionen in Ihrem Layout und gewünschtem Funktionsumfang an.

## 6.3 Subsembly EBICS API

Die Subsembly EBICS API ist eine .NET Klassenbibliothek zur einfachen Integration von E-Banking in beliebige Windows Anwendungen. Eine detaillierte Beschreibung finden Sie hier: <https://subsembly.com/ebics-api.html>

Die EBICS API wird intern auch vom BankAccessServer EBICS verwendet und ist unter .NET Core und dem vollständigen .NET Framework ab Version 4.5 lauffähig.

## 6.4 Subsembly EBICS Server

Der Subsembly EBICS Server ist ein plattformneutrales, vollständig auf Basis von .NET Core realisiertes, Servermodul für die Implementierung des EBICS Standards bei Kreditinstituten oder anderen Anbietern, welche einen sicheren Dateitransfer benötigen.

Siehe: <https://subsembly.com/ebics-server.html>

## 6.5 Subsembly EBICS Dummy

Der Subsembly EBICS Dummy ist ein von der Subsembly GmbH betriebener EBICS Server mit Dummy-Backend für Entwicklung, Test und Validierung von EBICS Kundensystemprodukten. Mit Hilfe des Subsembly EBICS Dummy können Entwickler von EBICS Kundenprodukten implementieren und testen.

Das Verhalten unter verschiedene Testszenarien kann wiederholt und ohne Risiko ausführlich getestet werden. Der Zugang zum Subsembly EBICS Dummy steht allen Benutzern des BankAccessServers EBICS kostenfrei zur Verfügung, so dass insbesondere auch der Prozess der Erstellung von EBICS Kontakten ausführlich vor der Verwendung mit echten Bankkonten getestet werden kann.

Der genaue Leistungsumfang ist hier beschrieben:

<https://subsembly.com/download/EBICS.Server.Dummy.pdf>

## 6.6 Subsembly BankingVEU App Server

Über den BankingVEU App Server ist es möglich, dass die BankingVEU App auf dem Smartphone Push-Benachrichtigungen bei neuen VEU-Aufträgen erhält. Der Anwender wird somit aktiv über alle neuen VEU-Aufträge informiert und kann diese dann umgehend freigeben.

Im Zusammenspiel mit dem BankAccessServer EBICS ist es somit möglich, Push-Benachrichtigungen nach der erfolgreichen Einreichung eines unsignierten Auftrags zu erzeugen.

Der BankingVEU App Server wird von der Subsembly GmbH als Cloud Service Lösung betrieben, kann jedoch bei Bedarf auch als On-Premise Lösung zur Verfügung gestellt werden.

## 6.7 Allgemeine Informationen zu EBICS

Das EBICS Verfahren ist auf den Seiten der Deutschen Kreditwirtschaft (DK) <https://die-dk.de/zahlungsverkehr/electronic-banking/dfu-verfahren-ebics/> und <https://www.ebics.de/de/startseite> ausführlich beschrieben.