Citfin
S citem pro finance

Citfin API PSD2

Manuál a dokumentace pro třetí strany (TPP)

Citfin – Finanční trhy, a.s.

Citfin, spořitelní družstvo

Autor A && L soft, s. r. o.
Verze 1.0

Citfin
S citem pro finance

Obsah

1. Úvod **4**
1.1 Technické informace 4
1.1.1 Architektura 4
1.1.2 API 4
1.1.3 Řazení 4
1.1.4 Stránkování 4
1.2 Terminologie 5
2. Bezpečnost **6**
2.1 Základní bezpečnostní model 7
2.2 Šifrovaná komunikace TPP-Banka 7
2.3 Registrace TPP v E-Bankingu 7
2.4 Souhlas s přístupem TPP k účtům disponenta 8
2.5 Popis řešení 8
2.5.1 Hlavní vlastnosti API 8
2.5.2 Popis základních postupů 8
2.5.2.1 Registrace TPP v bance 8
2.5.2.2 Registrace klienta banky v aplikaci TPP - nastavení přístupu TPP k účtům disponenta 9
2.5.2.3 Autorizace platby klientem 10
2.6 Služby podporované v API PSD2 10
2.6.1 Služby oblasti AIS 11
2.6.2 Služby oblasti PIS 11
2.6.3 Služby oblasti CIS (Ověření dostatku postředků) 11
2.7 Dodatečné služby – automatická registrace TPP přes bankovní API 12
2.8 Popis metod používaných pro poskytovatele služeb (TPP) 12
2.8.1 Registrační resource (Enrollment) 12
2.8.1.1 Automatické vygenerování technický identifikátorů 12
2.8.1.2 Informace o registračních údajích aplikace 15
2.8.1.3 Změna registračních údajů 17
2.8.1.4 Smazání aplikace 18
2.8.1.5 Žádost o nový client_secret 19
2.8.2 Autentizace a Autorizace requestu (OAuth2) 20
2.8.2.1 OAuth2 Authorization Code Grant 20
2.8.2.1.1 Základní vlastnosti 20
2.8.2.1.2 Popis Code grant flow 20
2.8.2.1.3 Autentizační resource vystavené bankou 21
2.8.2.1.4 Získání tokenů (Get token resource) 22
2.8.2.1.5 Obnovení Access tokenu 23
2.9 Popis metod, které jsou k dispozici poskytovatelům služeb (TPP) přes PSD2 API 24
2.9.1 Služby pro AISP (Dotazy k účtům, přehled transakcí) 24
2.9.1.1 Předpoklady pro používání metod API pro AISP 24
2.9.1.2 Seznam metod používaných pro službu AISP 25
2.9.1.3 Definice hlavičky 25
2.9.1.4 AISP operace: Zůstatek na účtu 25
2.9.1.4.1 Definice typu ArrayOfAccountsInformationResponseBalance 26
2.9.1.5 AISP operace: Přehled transakcí 27
2.9.1.5.1 Definice typu ArrayOfAccountsTransactionResponseTransaction 28
2.9.1.5.2 Definice typu AccountsTransactionsResponseTransactionDetail 28
2.9.1.6 AISP operace: Account List 32
2.9.1.6.1 Definice typu ArrayOfAccountsInfo 33
2.9.2 Služby pro PISP (Vytvoření platby, Zjišťování stavu platby, Autorizace platby) 34

2.9.2.1 Předpoklady pro používání metod API pro službu PISP 34
2.9.2.2 Seznam metod používaných pro službu PISP 35
2.9.2.3 Definice hlavičky 35
2.9.2.4 PISP Operace: Dotaz na dostatek prostředků 36
2.9.2.5 PISP Operace: Nová platba (inicializace platby) 39
2.9.2.6 PISP operace: Status založené / iniciované platby 46
2.9.2.7 PISP operace: Info o založené / iniciované platbě 46
2.9.2.8 PISP operace: Smazání založené neautorizované platby 47
2.9.2.9 PISP Generování autorizačního ID 48
2.9.2.10 PISP Operace: Inicializace autorizace platby 49
2.9.3 Služba CISP (Ověření dostatečných prostředků na účtu) 51
2.9.3.1 Předpoklady pro používání metod API pro službu CISP 51
2.9.3.2 Seznam metod používaných pro službu CISP 51
2.9.3.3 Token pro CISP operaci 51
2.9.3.4 Definice hlavičky 51
2.9.3.5 CISP Operace: Dotaz na dostatek prostředků 52
2.9.3.6 Enum použitý v položce AuthenticationMethod 55
3. Zdroje 56

1. Úvod

Řešení API PSD2 v Citfin – Finanční trhy, a.s. a Citfin, spořitelní družstvo vychází z Českého standardu pro Open Banking (COBS) verze 2 (dále jako **COBS v2**) vydaného Českou Bankovní Asociací (ČBA). Ve specifikaci níže jsou popsány jednotlivá rozhraní i s uvedením případných odchylek od standardu. Pokud nejsou některé atributy či rozhraní popsány, pak nejsou ze strany banky podporovány.

1.1 Technické informace

1.1.1 Architektura

Pro komunikační rozhraní API se používá transportní protokol **REST** (Representational State Transfer).
Pro formát zápisu dat dotazu i odpovědi přes API je použit **JSON** (JavaScript Object Notation).
Pro autorizaci požadavků je použit **autentizační protokolu OAuth 2.0**.

1.1.2 API

Třetí strana bude své požadavky zasílat na vystavené endpointy.

Při provádění drobných změn v API, které jsou zpětně kompatibilní, se nebude zvyšovat verze API. Vyhrazujeme si právo provádět takovéto drobné změny bez předchozího upozornění, a proto by na tyto situace měla být aplikace třetí strany připravena. Jedná se zejména o:

- Rozšíření odpovědi o nové atributy bez změny struktury současných atributů
- Úprava chybových hlášek a výjimek, včetně jejich kódů
- Rozšíření volitelných parametrů (hlavička, URL parametr, tělo) požadavku

1.1.3 Řazení

Veškerá API vracejí záznamy implicitně v pořadí, v jakém jsou prezentována uživatelům prostřednictvím jiných elektronických kanálů banky (InternetBanking). Pokud při volání služeb API použijete volitelné parametry specifikující řazení dle COBS, budou rozhraním banky ignorovány.

1.1.4 Stránkování

U specifických API, které navracejí kolekce (Přehled transakcí), je možné požádat o stránkovaný seznam. Pro tento dotaz slouží query parametry page a size. Každý zdroj, který umožňuje požádat o stránkovaný seznam, má tuto vlastnost explicitně uvedenu.

Parametry dotazu na stránku

page - Požadované číslo stránky. Stránky jsou číslované od 0. Pokud parametr není uveden, vrací API první (nultou) stránku.
size - Požadovaný počet záznamů na stránce. Pokud parametr není uveden, API vrací celou kolekci.

Parametry stránkované odpovědi

pageNumber - Číslo aktuální stránky. První stránka má číslo 0.
pageCount - Celkový počet stránek.
pageSize - Počet záznamů na stránce. Tento parametr může odpovídat požadované hodnotě size z dotazu až na případy, kdy se jedná o poslední stránku, nebo v případě, kdy požadovaný rozsah stránek překročil maximální limit definovaný pro konkrétní API zdroj.

Filtrování

API nepodporuje filtrování s výjimkou přehledu „Přehled transakcí“

Pravidla pro filtrování

fromDate - datum od.
Může být max. 2 roky do historie. Pokud není vyplněno, použito se aktuální datum.
Typ: datum (volitelné, formát YYYY-MM-DD), příklad: 2018-04-28
toDate - datum do.
Pokud není vyplněno, použije se aktuální datum.
Typ: datum (volitelné, formát YYYY-MM-DD), příklad: 2018-05-16
fromDate nesmí být větší toDate.

Pokud budou v požadavku použity zde neuvedené parametry filtrování, budou rozhraním banky ignorovány.

1.2 Terminologie

Citfin – SD (dále CITFIN nebo banka)

Citfin, spořitelní družstvo

Citfin – FT (dále CITFIN nebo banka)

Citfin – Finanční trhy, a.s.

ASPSP

Account Servicing Payment Service Provider – poskytovatel platebních služeb, v tomto případě banka.

COBS

Zkratka pro Czech Open Banking Standard.

TPP

Third Party Provider – třetí strana, subjekt, poskytovatel platebních služeb.
Třetí stranou (TPP) může být instituce, která má udělenou licenci od ČNB nebo od jiné oficiální autority v rámci zemí EU. Banka, přes jejíž API chce TPP zasílat PSD2 požadavky, si tuto licenci může ověřit v seznamu dostupném na stránkách ČNB.

TPP se dělí na následující typy, přičemž TPP může mít licenci k provozování kombinace níže uvedených služeb:

- AISP (Poskytovatel služby informace o účtu)
- PISP (Poskytovatel služby iniciace platby)
- CISP (Vydavatel karetního platebního prostředku)

Consent

Souhlas klienta s poskytováním služeb skrze zprostředkovatele – TPP (služby typu AIS, PIS, CIS).
Klient udělením souhlasu povoluje přístup aplikace TPP ke svým účtům (v souhlasu specifikuje, ke kterým účtům uděluje pro TPP přístup a jaké služby (AIS, PIS, CIS) povoluje).

AISP

Account Information Service – poskytovatel služby informování o platebním účtu – na základě souhlasu klienta poskytuje TPP informace o platebním účtu a transakcích, které jsou vykonané na účtu klienta v bance. Například pokud má klient vedené účty ve vícero bankách, prostřednictvím třetí strany může vidět historii transakcí, případně i zůstatky na všech těchto účtech současně na jednom místě (přes aplikaci nebo portál TPP).

PISP
Payment Initation Service - Třetí strana může:
- za zákazníka iniciovat platby
- provádět odeslání iniciované platby ke zpracování (pokud předtím klient tuto platbu autorizoval)
- dotazovat se na stav zpracovávané platby
- dotázat se na dostatek platebních prostředků na účtu, ze kterého se má provést platba

CISP
Card-based Payment Instrument Issuer – poskytovatel platebních služeb vydávající platební prostředek (platební kartu). Existují TPP, které mohou vydat platební prostředek, který bude provázaný k platebnímu účtu v bance. TPP si pak bude moci ověřit, zdali má klient na bankovním účtu, ke kterému TPP vydala kartu, dostatek prostředků k zrealizování transakce kartou. Banka odpoví na dotaz TPP odpovědí ANO / NE.

API Gateway
- Poskytuje přístup TPP k službám banky.

Platební služby
Provedení platby (transakce) na základě autorizovaného požadavku.

Autentizační server (eCobra)
Aplikace autentizačního serveru, provozované v CITFIN.

IB
Zkratka pro elektronické bankovnictví, provozované v CITFIN.

PSD2
Payment Services Directive 2 - směrnice EU pro bankovní odvětví z roku 2015, která se obecně zabývá platebními službami. Směrnice vznikla, aby sjednotila poskytování platebních služeb v EU**.**

EV
Extended Validation certificate – certifikát s rozšířenou validací.

2. Bezpečnost

Autorizace požadavků na rozhraní je založena na autorizačním konceptu OAuth2 zabezpečeného tokenem. Klient poskytuje při každém volání API token (access_token) jako důkaz, že může přistupovat k požadovaným údajům. Rozhraní banky ověří použitý token ve vztahu k použitému rozhraní a teprve po úspěšném ověření tokenu a oprávnění z něj vyplývajících je provedena požadovaná operace. Token musí být uveden v hlavičce požadavku, např.:

Authorization: Bearer aT6oKuCt6i0plw26nxI7r32Lpi89bt

Parametry:

Parametr	Hodnota
Doba platnosti Refresh tokenu	90 dní
Doba platnosti Access tokenu	3600 sekund (1 hodina)
Doba platnosti vygenerovaného Code vygenerovaného doknočení požadavku /autorizace	10 minut
Doba platnosti client_secret	Neomezeno
Doba platnosti požadavku na autorizaci signid	5 minut (Přesměrování uživatele na Federovanou autorizaci banky je možné pouze po dobu platnosti signId. Po vypršení platnosti nutno zažádat o vygenerování nového signId pomocí metody - Generování autorizačního ID)

2.1 Základní bezpečnostní model

Základní bezpečnostní model pro přístup k API je založen na kombinaci níže uvedených bezpečnostních prvků (aby TPP mohla posílat požadavky přes API, musí být splněny všechny následující bezpečnostní prvky).

- Šifrovaná komunikace mezi TPP a bankou (použití platného certifikátu na straně TPP i banky)
- Registrovaný ověřený platný záznam TPP v Internet Bankingu
- **Registrovaná aplikace TPP** v Internet Bankingu (s jedinečným client_id a client_secret)
- Existence platného souhlasu s definovaným přístupem **aplikace TPP** k účtům disponenta
- **Platný access token (navázaný na specifický souhlas, vytvořený disponentem) uváděný v hlavičce zaslaného požadavku přes API**

2.2 Šifrovaná komunikace TPP-Banka

Komunikace mezi klientským systémem a bankou předpokládá zabezpečení pomocí SSL protokolu s minimálně 128 bitovým šifrováním. Na straně banky i TPP musí být pro vytvoření zabezpečeného kanálu použit kvalifikovaný certifikát pro autentizaci webových serverů **dle eIDAS**. Použitý certifikát musí být vydán v souladu s ETSI TS 119 495 (Kvalifikované certifikáty a požadavky na politiku TSP podle směrnice o platebních službách (EU) 2015/2366).
Pro zabezpečení komunikační vrstvy je vyžadována verze TLS 1.2+.

2.3 Registrace TPP v E-Bankingu

Záznam každé TPP, která bude chtít zasílat požadavky přes API PSD2 vystavené bankou, musí existovat v databázi IB.

Vytváření nového záznamu TPP a aktualizace záznamů TPP již existujících v databázi e-Bankingu (ověření a registraci TPP) provádí správce E-Bankingu přes GUI intranetové části IB.

Správce e-Bankingu bude do databáze E-Bankingu zavádět jen TPP, které budou kontaktovat banku - po obdržení certifikátu správce IB ověří licenční číslo TPP a založí TPP databáze IB. Při registraci záznamu ověřeného TPP správce doplní do záznamu TPP celé licenční číslo včetně prefixu uvedené v certifikátu TPP.

Pokud existuje v databázi IB platný záznam TPP, musí TPP přes specifický endpoint vystaveného PSD2 API provést registrační flow v bance.
Při registračním flow si TPP v bance zaregistruje svou **aplikaci / multibank portál** (TPP může provozovat více aplikací – *poznámka: pokud TPP nabízí klientům více svých aplikací, musí každou svou PSD2 aplikaci zaregistrovat v bance*).
TPP obdrží ke každé zaregistrované aplikaci od banky technické identifikátory (client_id, client_secret).

2.4 Souhlas s přístupem TPP k účtům disponenta

Další podmínkou, která musí být splněna, aby TPP mohla zasílat požadavky přes API, je existující platný souhlas přístupu **aplikace TPP** k účtům disponenta. Souhlas vznikne na základě žádosti o přístup, kterou disponent sám vytvoří na straně banky a autorizuje ji svým autentizačním zařízením (správce IB nezasahuje do tohoto procesu).

Součástí uloženého souhlasu jsou položky:

- Vybraná aplikace TPP
- Seznam oprávnění ke službám (AISP, PISP, CISP), které povolil disponent pro aplikaci TPP.
- Seznam účtů, ke kterým disponent povolil přístup (v žádosti se nabízí pouze běžné účty, ke kterým má daný disponent nastavený v bance aktivní přístup a pokud má disponent ve vazbě na klienta, který je majitelem účtu, povolenou v bance službu PSD2)
- Datum "Platnost DO" vydaného souhlasu (platnost souhlasu je implicitně neomezená)

2.5 Popis řešení

2.5.1 Hlavní vlastnosti API

- ***API rozhraní podporuje:***
  - všechny mandatorní služby požadované v rámci COBS
  - nemandatorní služby používané pro automatizovanou registraci aplikace TPP
- ***API rozhraní:***
  - TPP budou využívat bankovní API, které je řešené jako webová služba (WS)
  - Pro komunikační rozhraní API je použit transportní protokol **REST** (Representational State Transfer).
  - Pro formát zápisu dat dotazu i odpovědi přes API je použit **JSON** (JavaScript Object Notation).
- ***Evidence aplikací TPP:*** Na každý záznam TPP může být navázáno 1…n aplikací TPP; aplikace si TPP registruje v bance při registračním flow.
- ***API rozhraní:*** E-Banking při každém požadavku obdrženém přes API provádí ověření TPP. Požadavek zaslaný z TPP přes API do banky obdrží požadovanou odpověď pouze při splnění všech následujících podmínek:
  - na základě certifikátu, který TPP používá při komunikaci je záznam TPP dohledán v tabulce TPP (TPP je dohledáváno na základě čísla licence TPP uvedeného v certifikátu (číslo licence včetně prefixu) – identické číslo licence musí být uvedeno i v záznamu TPP (v položce IdentifierInCertificate) v databázi IB)
  - dohledaný záznam TPP je platný,
  - typ použité metody odpovídá službě (AISP, PISP, CISP), která je povolená v dohledaném záznamu TPP v tabulce TPP
  - access kód použitý v požadavku je platný¨
  - na základě použitého access kódu (OAUTH protokol) uvedeného v požadavku, je dohledán platný souhlas ve vazbách Disponent-Aplikace TPP
  - účet, který je uveden v požadavku, je obsažen v souhlasu, který byl dohledán na základě access kódu (uvedeného v hlavičce požadavku)
  - aplikace TPP má v dohledaném souhlasu povolenu od disponenta službu (AISP, PISP, CISP), která odpovídá metodě použité v přijatém požadavku

2.5.2 Popis základních postupů

2.5.2.1 Registrace TPP v bance

Aby TPP mohla komunikovat přes API PSD2 banky, musí obdržet pro svou aplikaci od banky technické bezpečnostní prvky (client_id, client_secret), potřebné pro následné získání tokenu používaného v rámci OAuth 2.0. Tyto prvky může TPP získat až po registraci své aplikace v bance.

1. TPP provádí registraci své aplikace v bance přes API vystavené bankou s použitím specifických metod – viz kapitolu 2.8.1).

2. V okamžiku přijetí požadavku na registraci aplikace TPP přes PSD2 API vystavené bankou, proběhne na straně banky v systému elektronického bankovnictví ověření TPP. Ověření je prováděno na základě **ID licence** vydané národním regulátorem a **certifikátu daného subjektu**. Aby bylo možné požadavek provést, musí být splněno následující:
   - ID licence, uvedené v certifikátu, který TPP používá při komunikaci přes PSD2 API vystavené bankou, musí být dohledáno v záznamu TPP (v položce **IdentifierInCertificate** v databázi IB).
   - Záznam TPP, který byl dohledán na základě ID licence, musí být platný.

3. V případě, že ID licence obsažené v certifikátu použitého při komunikaci TPP přes API, není obsaženo v žádném záznamu TPP v databázi, může být postup následující:
   - TPP kontaktuje pracovníka banky, který provede manuální ověření (TPP bance předá svůj certifikát (bez tajné části) s potřebnými doklady, na základě kterých pracovník banky ověří danou TPP. Ověření záznamu TPP proběhne na základě Názvu TPP subjektu, **ID licence vydané národním regulátorem** a **certifikátu daného subjektu**. Po manuálním ověření pracovník banky vytvoří přes Intranetové rozhraní IB nový záznam TPP a doplní do databáze IB do záznamu TPP ID licence (obsažené v certifikátu TPP).
   - Po vytvoření nového záznamu TPP v databázi IB pracovníkem banky, TPP provede další pokus registrace své aplikace přes API.

4. Při registraci aplikace TPP pro komunikaci přes API banky jsou v elektronickém bankovnictví vygenerovány následující technické bezpečnostní prvky potřebné při autentizačním flow s použitím OAuth 2.0:
   - Identifikátor (client_id), který bude aplikace TPP při komunikaci přes API používat
   - secret kód (client_secret), (TPP nebude secret_code nikdy používat samostatně, vždy musí být použit v kombinaci s client_id – kombinace client_id a client_secret je obsaženo v requestu při výměně jednorázového autorizačního kódu za refresh a access token (Get token resource).

5. Vygenerované technické bezpečnostní prvky jsou předány TPP (TPP tyto technické bezpečnostní prvky obdrží při registraci přes API jako odpověď na požadavek registrace)

6. Od okamžiku vygenerování technických bezpečnostních prvků se název zaregistrované aplikace TPP bude nabízet klientům banky při vytváření souhlasů pro přístup TPP k účtům.

2.5.2.2 Registrace klienta banky v aplikaci TPP - nastavení přístupu TPP k účtům disponenta

Aby TPP mohla zasílat dotazy na účty disponenta nebo vytvářet za disponenta platbu a následně ji autorizovat, musí k tomu dát disponent souhlas. Následují kroky, které musí provést disponent.

1. Disponent může souhlas o přístup třetí strany ke svým běžným účtům vytvořit přes Centrální autorizační stránku, na kterou je přesměrován při aktivaci PSD2 přístupu přes aplikaci třetí strany.

2. Pokud je aplikace třetí strany již zaregistrována v bance, disponent klienta banky se může standardním způsobem přihlásit do IB a v sekci PSD2 vytvořit žádost o vytvoření souhlasu pro přístup specifické aplikace TPP ke svým účtům:
   - povolí pro danou aplikaci TPP požadované služby AIS / PIS / CIS
   - povolí přístup ke svým účtům (v nabídce se zobrazí jen běžné účty, ke kterým má daný disponent povolen **aktivní** přístup ve vazbě disponent klient a zároveň má v balíčku služeb k danému klientovi povolenu službu PSD2).
   - Disponent žádost o souhlas autorizuje svým autorizačním zařízením.

3. Klient banky si nainstaluje aplikaci TPP nebo přistupuje k portálu TPP.

4. Klient banky (uživatel aplikace) si v aplikaci / portálu vybere svou banku a spustí registrační workflow.

5. Klient je po žádosti o registraci v aplikaci TPP přesměrován na **autentizační frontend CITFIN (centrální autentizační stránku)** s využitím protokolu OAuth 2.0 s žádostí o autorizaci přístupu ke službám.

6. Klient se na centrální stránce standardním způsobem autentizuje (autentizuje se svým identifikátorem, heslem a kódem vygenerovaným na svém autentizačním zařízení).

7. Po autentizaci klienta IB zkontroluje, zda pro danou aplikaci a právě přihlášeného disponenta existuje ve vazbě **Disponent – aplikace TPP** platný souhlas (souhlas vytvořený na základě žádosti z IB nebo z Centrální autorizační stránky).
   - **Varianta - platný souhlas není dohledán:** zobrazí se stránka, ve které disponent klienta odsouhlasí přidělení přístupu dané aplikace TPP ke službám (AISP, PISP, CISP) a k běžným účtům, ke kterým má nastaven aktivní přístup (pokud je disponentem u vícero klientů, nabídnou se mu při vytváření souhlasu účty všech klientů, ke kterým **má přes specifického klienta nastaven aktivní přístup a zároveň má k danému klientovi v balíčku služeb povolenu službu PSD2**).
   - **Varianta - platný souhlas je dohledán**: v odpovědi protokolu OAuth získá aplikace TPP jednorázový autorizační kód, který následně zašle serveru TPP. Server TPP posléze kontaktuje /**token endpoint** vystavený na frontendu banky, aby tento jednorázový autorizační kód vyměnil za dvojici tokenů **access a refresh token**.

8. Aplikace TPP následně Access token používá při komunikaci s PSD2 API vystavený bankou. Vnitrobankovní systémy (API Gateway) požádají o ověření platnosti tokenu a získání příslušného scope (AIS/PIS/CIS) a příslušné uživatelské identity, ke které token patří.

2.5.2.3 Autorizace platby klientem

1. Pokud je přes aplikaci / portál TPP iniciována platba, obdrží TPP v odpovědi přes PSD2 API číslo, pod jakým se daná platba uložila na straně banky (orderId) a Identifikátor tokenu vygenerovaného pro daný autorizační proces konkrétní transakce (signId). Platnost signId je omezená (5 minut). Tuto platbu musí následně autorizovat klient banky přímo na straně banky.
2. Aplikace spustí inicializaci autorizace platby (POST /api/payments/{paymentId}/sign/{signId}). Spuštění autorizace procesu je umožněno pouze v případě, pokud je platný signId. Po zavolání této metody obsahující typ CODE odpovídající federované autorizaci (USERAGENT-REDIRECT) je odpovědí URL a parametry pro přesměrování na federovanou autorizační stránku (centrální autentizační stránku).
3. Třetí strana tyto parametry použije k přesměrování klienta na danou centrální autentizační stránku banky. Součástí tohoto přesměrování je i číslo platby (orderId), která má být klientem autorizována.
4. Logika centrální stránky ověří, zda aplikace, ze které byl disponent banky přesměrován je zaregistrována v IB (na základě **client_id** uvedené v URL adrese). Pokud client_id použité v požadavku není dohledáno v databázi IB v nějaké aplikaci TPP, je požadavek odmítnut (v odpovědi je vrácena chyba).
5. Pokud je aplikace TPP registrovaná a je validní i adresa redirect_uri použitá v požadavku (dané URI musí být uvedeno v záznamu zaregistrované aplikace TPP v IB), prochází klient po přesměrování prostřednictvím autentizačního procesu zajištěného bankou v principu SCA autentizací (tak jak je klientovi znám z prostředí IB).
6. Po autentizaci uživatele se uživateli zobrazí na základě orderID detail dané platby. Autorizací platby uživatel odsouhlasí provedení dané transakce.

2.6 Služby podporované v API PSD2

Řešení PSD2 umožňuje třetí straně používat přes vystavené API služby PSD2 – viz následující tabulky:

2.6.1 Služby oblasti AIS

AIS	Popis
Zůstatek na účtu (JSON)	prostřednictvím této služby dostane klientem autorizovaná třetí strana přehled zůstatků bankovního účtu klienta vedeného v dané bance
Přehled transakcí (JSON)	prostřednictvím této služby dostane klientem autorizovaná třetí strana přehled transakcí
Seznam platebních účtů klienta (JSON)	služba na požadavek vrátí seznam účtů, ke kterým klient dává souhlas s konkrétním TPP (nikoliv seznam všech klientských účtů) bez zůstatků

2.6.2 Služby oblasti PIS

PIS	Popis
Dotaz na dostatek prostředků (JSON)	Ověření dostatečného zůstatku na účtu plátce
Nová platba (iniciace platby) (JSON)	prostřednictvím této služby klientem autorizovaná třetí strana iniciuje (vytvoří) **jeden** non eCommerce platební příkaz z bankovního účtu klienta ve formátu JSON TPP následně použije na daný příkaz metodu na inicializaci autorizace platby - klient je přesměrován na centrální autentizační stránku banky a zde daný příkaz autorizuje svým autentizačním zařízením (TOKEN).
Status založené/iniciované platby (JSON)	zjišťování stavu platebního příkazu
Smazání založené neautorizované platby (JSON)	Služba umožňující zrušit platbu, která ješte nebyla autorizovaná a která byla vytvořena prostřednictvím služby PISP Nová platba (iniciace platby)
Autorizace platby (JSON)	Prostřednictvím této sluby může třetí strana spustit workflow autorizace platby iniciované přes aplikaci třetí strany
Generování autorizačního ID (JSON)	Prostřednictvím této metody může třetí strana zažádat o vygenerování nového SignId. Používá se v případech kdy: › Platnost původního požadavku na autorizaci (např. vzniklý jako výstup z volání metody Inicializace platby) vypršela. Platnost požadavků na autorizaci je omezena na 5 minut od jejich založení. › Autorizace požadavku uživatelem nebyla provedena z vůle uživatele - odmítnut pokyn autorizovat - a má zájem o opakovaný pokud o autorizaci › Z technických důvodů na straně TPP aplikací *Poznámka: založením nového autorizačního zůstávají v platnosti původní dosud platné a nevyužité požadavky na autorizaci.*

2.6.3 Služby oblasti CIS (Ověření dostatku postředků)

Služba směrnicí PSD2 definovaná jako informace o dostatku prostředků poskytovaná pro providery CISP.

CIS	Popis
Dotaz na dostatek prostředků (JSON)	Ověření dostatečného zůstatku na účtu plátce

2.7 Dodatečné služby – automatická registrace TPP přes bankovní API

Součástí řešení je i implementace níže uvedených metod, které jsou v rámci COBS volitelné. Tyto metody TPP umožňují automatickou registraci aplikace / změny v registraci aplikace přes bankovní API.

Přiřazování technických bezpečnostních prvků TPP (Enroll)	Popis
Registrace aplikace TPP (JSON)	prostřednictvím této služby TPP s platným certifikátem a licenčním číslem provede automatickou registraci své aplikace v bance a v odpovědi obdrží k registrované aplikaci technické bezpečnostní prvky (client_id a client_secret)
Informace o registračních údajích aplikace (JSON)	Prostřednictvím této metody může TPP požádat o přehled registračních údajů pro konkrétní aplikaci
Změna registrace aplikace (JSON)	prostřednictvím této služby bude TPP moci provést změnu registračních údajů
Smazání registrace aplikace (JSON)	prostřednictvím této služby bude TPP moci zrušit registraci aplikace
Žádost o vygenerování nového client_secret	prostřednictvím této služby bude TPP moci požádat o vygenerování nového client_secret

2.8 Popis metod používaných pro poskytovatele služeb (TPP)

2.8.1 Registrační resource (Enrollment)

Následující kapitoly popisují metody, pomocí nichž TPP žádá o registraci své aplikace v bance, případně může provést změny nebo odregistrování své aplikace.

2.8.1.1 Automatické vygenerování technický identifikátorů

Pro zavolání resource je potřeba:

- **použít platný certifikát**.

Výstupem jsou parametry **client_id** a **client_secret**, které TPP potřebuje pro následné získání dvojice tokenů **access_token** a **refresh_token**.

Endpoint: POST https://api.bankservis.cz/api**/oauth2/register**

Request			
Atribut	**Mandatory**	**Typ / povolené hodnoty**	**Popis**
application_type	Ano	string	Typ aplikace, která bude používat client_id. (jsou povolené jen typy Web, native).
redirect_uris	Ano	Array of strings e.g. URL [Max 3x 2047 B]	Výčet URL kam je na konci přesměrováno flow autentizace. Autorizační request musí obsahovat právě jedno z těchto zaregistrovaných URI v přesném formátu.
client_name	Ano	String [Max 255 B]	Jméno TPP aplikace
client_name#en-US	Ne	String [Max 1024 B]	Jméno TPP aplikace v příslušném jazyce / kódování.
logo_uri	Ne	URI [Max 2047 B]	URI loga aplikace (resp. místo odkud je možné ho při registraci stáhnout)
contact	Ne	string [Max 320 B]	E-mail jako kontakt na zodpovědnou osobu na straně klientské aplikace.
scopes	Ne	Array of strings [Max 10x 255 B]	Pole aplikací požadovaných scopes. Při registraci jsou scopes validovány proti obsahu použitého certifikátu a proti scopes uvedených v záznamu TPP, který v té době již musí existovat v databázi IB.

Response			
Atribut	**Mandatory**	**Typ**	**Popis**
client_id	Ano	String	Aplikaci přiřazené **client_id**. Toto ID je používáno při spuštění autentizačního procesu a při komunikačním procesu (výměně jednorázového code za dvojici tokenů access_token a refresh_token a při obnovení tokenu).
client_secret	Ano	String	**Client_secret** - password / token vydaný bankou (ASPSP) pro TPP aplikaci (**client_id**)
client_secret_expires_at	Ne	DateTime	Defaultní hodnota je 0 (client_secret nikdy neexpiruje). V opačném případě je uvedena hodnota v sekundách od data 1970-01-01T0:0:0Z
api_key	Ne	String	API klíč, který aplikace používá při komunikaci s API banky. **API klíč není v tomto řešení bankou podporován** (v odpovědi v položce uvedeno „NOT_PROVIDED“)
application_type	Ano	String	Typ aplikace, která bude používat client_id. (jsou povolené jen typy Web, native).
redirect_uris	Ano	Array of strings e.g. URL [Max 3x 2047 B]	Výčet URL kam je na konci přesměrováno flow autentizace.
client_name	Ano	String [Max 255 B]	Jméno TPP aplikace
client_name#en-US	Ne	String [Max 1024 B]	Jméno TPP aplikace v příslušném jazyce / kódování.
logo_uri	Ne	URI [Max 2047 B]	URI loga aplikace
contact	Ne	string [Max 320 B]	E-mail jako kontakt na zodpovědnou osobu na straně klientské aplikace.
scopes	Ne	Array of strings [Max 10x 255 B]	Pole požadovaných scopes.

Chybové kódy		
HTTP Status	Error kód	Popis
400	**invalid_request**	Nevalidní request. V dotazu chybí povinné pole nebo je v nevhodném / nevalidním formátu.
400	**invalid_scope**	Nevalidní scope v požadavku.
400	**invalid_redirect_uri**	Hodnota jednoho nebo více redirect uri není validní.
401	**unauthorized_client**	TPP není oprávněný provádět tento dotaz.
401	**access_denied**	Autorizační server odmítl přístup.
403	**insufficient_scope**	Např. nedostatečné oprávnění pro použití požadovaného scope.
500, 503	**server_error**	Chyba autorizačního serveru.

Příklad použití viz zdroj [7] kapitola 1.4.1.1.

2.8.1.2 Informace o registračních údajích aplikace

Zavoláním tohoto resource může TPP požádat o přehled registračních údajů pro konkrétní aplikaci.
Pro zavolání resource je potřeba:

- **použít platný certifikát**
- **použít client_id, které je vydáno k tomuto TPP**.

Výstupem je přehled registračních údajů.

Endpoint: GET https://api.bankservis.cz/api**/oauth2/register/{client_id}**

Response			
Atribut	**Povinný**	**Typ**	**Popis**
client_id	A	String	Jedinečný identifikátor **client_id** přiřazený aplikaci TPP bankou.
client_secret	A	String	**Client_secret** - password / token vydaný bankou (ASPSP) pro TPP aplikaci (**client_id**)
client_secret_expires_at	N	DateTime	Defaultní hodnota je 0 (client_id nikdy neexpiruje). V opačném případě je uvedena hodnota v sekundách od data 1970-01-01T0:0:0Z
api_key	N	String	API klíč, který aplikace používá při komunikaci s API banky. **API klíč není v tomto řešení bankou podporován** (v odpovědi v položce uvedeno „NOT_PROVIDED“)
application_type	A	String	Typ aplikace, která používá client_id (povoleny jsou hodnoty web, native)
redirect_uris	A	Array of strings e.g. URL	Výčet URL, kam je na konci přesměrováno flow autentizace. Autorizační request musí obsahovat právě jedno z těchto zaregistrovaných URI v přesném formátu.
client_name	A	String	Jméno TPP aplikace
client_name#en-US	N	String	Jméno TPP aplikace v příslušném jazyce / kódování.
logo_uri	N	URI	URI loga aplikace
contact	N	string	Seznam E-mail adres, kontakty na zodpovědnou osobu na straně TPP aplikace.
scopes	N	Array of strings [Max 10x 255 B]	Pole požadovaných scopes.

Chybové kódy		
HTTP Status	**Error kód**	**Popis**
400	**invalid_request**	Nevalidní request. V dotazu chybí povinné pole nebo je v nevhodném / nevalidním formátu.
401	**invalid_client**	Nevalidní client_id.
401	**unauthorized_client**	TPP není oprávněný provádět tento dotaz.
401	**access_denied**	Autorizační server odmítl přístup.
403	**insufficient_scope**	Např. nedostatečné oprávnění pro použití požadovaného scope.
500, 503	**server_error**	Chyba autorizačního serveru.

Příklad použití viz zdroj [7] 1.4.1.2.

2.8.1.3 Změna registračních údajů

Zavoláním tohoto resource může TPP požádat o změnu registračních údajů pro konkrétní aplikaci.
Pro zavolání resource je potřeba:

- **použít platný certifikát**
- **použít client_id, které je vydáno k tomuto TPP**.

Výstupem je přehled změněných údajů.

Endpoint: PUT https://api.bankservis.cz/api/**oauth2/register/{client_id}**

Request			
Atribut	**Povinný**	**Typ**	**Popis**
application_type	A	string	Typ aplikace, která bude používat client_id (povoleny jsou hodnoty web, native)
redirect_uris	A	Array of strings e.g. URL [Max 3x 2047 B]	Výčet URL kam je na konci přesměrováno flow autentizace. Autorizační request musí obsahovat právě jedno z těchto zaregistrovaných URI v přesném formátu.
client_name	A	String [Max 255 B]	Jméno TPP aplikace
client_name#en-US	N	String [Max 1024 B]	Jméno TPP aplikace v příslušném jazyce / kódování.
client_type	A	String	OAuth definuje dva typy klientů (Confidential / Public). ASPSP (banka) podporuje pouze typ **Confidental**.
logo_uri	N	URI [Max 2047 B]	URI loga aplikace (resp. místo odkud je možné ho při registraci stáhnout)
contact	N	string [Max 320 B]	E-mail jako kontakt na zodpovědnou osobu na straně klientské aplikace.
scopes	N	Array of strings [Max 10x 255 B]	Pole aplikací požadovaných scopes. Při registraci jsou scopes validovány proti obsahu použitého certifikátu a proti scopes uvedených v záznamu TPP, který v té době již musí existovat v databázi IB (záznam TPP je vytvořen při aktualizaci dat z NBS).

Response			
Atribut	**Povinný**	**Typ**	**Popis**
client_id	A	String	Jedinečný identifikátor **client_id** přiřazený aplikaci TPP bankou.
application_type	A	String	Typ aplikace, která bude používat client_id
redirect_uris	A	Array of strings e.g. URL	Výčet URL kam je na konci přesměrováno flow autentizace.
client_name	A	String	Jméno TPP aplikace
client_name#en-US	N	String	Jméno TPP aplikace v příslušném jazyce / kódování.
logo_uri	N	URI	URI loga aplikace
contact	N	string	Seznam E-mail adres, kontakty na zodpovědnou osobu na straně TPP aplikace.
scopes	N	Array of strings	Pole požadovaných scopes.

Chybové kódy		
HTTP Status	**Error kód**	**Popis**
400	**invalid_request**	Nevalidní request. V dotazu chybí povinné pole nebo je v nevhodném / nevalidním formátu.
400	**invalid_scope**	Nevalidní scope v požadavku.
400	**invalid_redirect_uri**	Hodnota jednoho nebo více redirect uri není validní.
401	**invalid_client**	Nevalidní client_id.
401	**unauthorized_client**	TPP není oprávněný provádět tento dotaz.
401	**access_denied**	Autorizační server odmítl přístup.
403	**insufficient_scope**	Např. nedostatečné oprávnění pro použití požadovaného scope.
500, 503	**server_error**	Chyba autorizačního serveru.

Příklad použití viz zdroj [7] 1.4.1.3.

2.8.1.4 Smazání aplikace

Zavoláním tohoto resource může TPP požádat o smazání údajů a přístupu konkrétní aplikaci.
Pro zavolání resource je potřeba:

- **použít platný certifikát**
- **použít platné client_id, které je vydáno tomuto TPP**.

Výstupem je potvrzení o smazání.

Endpoint: DELETE https://api.bankservis.cz/api**/oauth2/register/{client_id}**

Pokud se smazání aplikace provede, je vrácena odpověď HTTP 204 jako úspěšná odezva na smazání záznamu aplikace s konkrétním client_id).

Chybové kódy		
HTTP Status	Error kód	Popis
400	**invalid_request**	Nevalidní request. V dotazu chybí povinné pole nebo je v nevhodném / nevalidním formátu.
401	**invalid_client**	Nevalidní client_id.
401	**unauthorized_client**	TPP není oprávněný provádět tento dotaz.
401	**access_denied**	Autorizační server odmítl přístup.
500, 503	**server_error**	Chyba autorizačního serveru.

Příklad použití viz zdroj [7] kapitola 1.4.1.4.

2.8.1.5 Žádost o nový client_secret

Zavoláním tohoto resource může TPP požádat o vydání nového client_secret.
Pro zavolání resource je potřeba použít:

- **platný certifikát**
- **platné client_id, které je vydáno tomuto TPP**.

Původní client_secret bude tímto requestem zrušen.

Endpoint: POST https://api.bankservis.cz/api/**oauth2/register/{client_id}/renewSecret**

Response			
Atribut	Povinný	Typ	Popis
client_id	A	String	Aplikaci přiřazené **client_id**.
client_secret	A	String	**Nový Client_secret** - password / token vydaný bankou (ASPSP) pro TPP aplikaci (**client_id**)
client_secret_expires_at	N	DateTime	Defaultní hodnota je 0 (client_secret nikdy neexpiruje). V opačném případě je uvedena hodnota v sekundách od data 1970-01-01T0:0:0Z

Chybové kódy		
HTTP Status	Error kód	Popis
400	**invalid_request**	Nevalidní request. V dotazu chybí povinné pole nebo je v nevhodném / nevalidním formátu.
401	**invalid_client**	Nevalidní client_id.
401	**unauthorized_client**	TPP není oprávněný provádět tento dotaz.
401	**access_denied**	Autorizační server odmítl přístup.
500, 503	**server_error**	Chyba autorizačního serveru.

Příklad použití viz zdroj [7] kapitola 1.4.1.5.

2.8.2 Autentizace a Autorizace requestu (OAuth2)

Autorizace requestu je založena na autorizačním flow konceptu OAuth2 zabezpečeného tokenem – aplikace pouze zkontroluje platnost tokenu použitého v hlavičce požadavku, které TPP poskytuje pro každé volání jako důkaz, že může přistupovat k požadovaným údajům.
V rámci těchto API je autorizační token považován za krátkodobý a bezstavový prvek, který musí být použit v každém volání API, které požaduje autorizaci requestu.

Základem řešení je použití OAuth2 otevřeného protokolu pro vystavování autorizačních tokenů – **je podporovaný jen autorizační framework Authorization code grant**. Varianta **Client Credentials Grant flow**, není v implementovaném řešení podporována.

2.8.2.1 OAuth2 Authorization Code Grant

V rámci protokolu OAuth2 se v případě autorizačního frameworku Authorization code grant jedná o způsob, jak partnerské aplikaci vydat refresh token i access token jako výsledek identifikace a autentizace uživatele.
Krátkodobý access token partnerská aplikace používá pro komunikaci s API banky a po jeho expiraci může použít refresh token pro vyžádání nového access tokenu.

2.8.2.1.1 Základní vlastnosti

- access token je vydáván jako krátkodobý (3600 s)
- access token je vydáván **pro konkrétní aplikaci a konkrétního uživatele** (**je navázan na souhlas vytvořený uživatelem - disponentem**), pro jinou aplikaci ho není možné úspěšně použít
- refresh token není možné přímo použít pro komunikaci s API, má dlouhou platnost (v případě PSD2 90 dní)
- banka a aplikace (TPP) spolu sdílí společné „tajemství“ (client secret)
- výsledkem identifikace a autentizace uživatele je jednorázový code, který aplikace třetí strany s použitím client secret vymění za refresh token a access token
- samotný jednorázový code bez znalosti client secret není možné použít

2.8.2.1.2 Popis Code grant flow

Podmínka použití flow:

- aplikace TPP má od banky přiděleno vlastní jedinečné **client_id** a zná pro dané **client_id** i **client secret**
- při vydání **client_id** a **client_secret** banka získá informaci o **redirect uri** – tedy o URL, kam má přesměrovat uživatele po úspěšné autentizaci

Jednotlivé kroky code grant flow:

1. Aplikace TPP zavolá /auth resource banky a následně je uživatel (klient banky) přesměrován na centrální autentizační stránku pro provedení identifikace a autentizace uživatele (klienta banky)

2. Probíhá identifikace a autentizace klienta – tyto kroky jsou plně v režii banky

3. Po úspěšné autentizaci banka vygeneruje **code** a přesměrovává s ním uživatele na URI, které bylo součástí požadavku /auth (redirect_uri)

4. TPP použije resource /token pro získání refresh_tokenu a access_tokenu. Při volání tohoto resource TPP předává bance v požadavku dvojici client_id a client_secret a hodnotu code, který dostala v odpovědi předchozího požadavku /auth.

5. Aplikace TPP používá při komunikaci na API banky v případech, kdy je to nutné, v hlavičce požadavku, získaný access_token

6. Banka interně provádí ověření access_tokenu. Při tomto ověření získává identitu uživatele, na základě jehož autentizace byl access token vydán.

2.8.2.1.3 Autentizační resource vystavené bankou

Pokud neexistuje platná dvojice tokenů (access_token a refresh_token), musí TPP vytvořit Autorizační požadavek, na základě kterého je klient banky z aplikace přesměrován na centrální autentizační stránku banky, kde daný požadavek následně autorizuje (viz 2.5.2.2). Požadavek je typu Oauth 2.0 **Authorization Code Grant.**

Endpoint: GET https://api.bankservis.cz/oauth2/auth

Request			
Atribut	**Mandatory**	**Typ**	**Popis**
response_type	Ano	**code**	Povinný parametr. Hodnotou parametru je určeno, jaký typ autentizačního flow je požadováno. V tomto případě se jedná o **code grant**. Pro autentizační proces to znamená, že výsledkem tohoto požadavku bude jednorázový **auth_code, který TPP následně pomocí dalšího požadavku (metodou token) zamění za dvojici tokenů access_token a refresh_token**
client_id	Ano	String	Jedinečný identifikátor, který banka vygenerovala pro aplikaci TPP
redirect_uri	Ano	URL	URL kam je na konci přesměrováno flow autentizace. Toto URL je stanoveno již při vydání client_id a v rámci autentizace je tento parametr validován proti URL zavedenému k client_id v záznamu aplikace registrované v bance. Hodnota se musí shodovat s jednou z hodnot uvedených v záznamu registrované aplikace.
Scope	Ano	String	Jedná se o pole aplikací požadovaných scope (oprávnění). V případě PSD2 to mohou být role AISP, PISP, CISP. Např. pokud je TPP držitelem více oprávnění, může zde pro svoji aplikaci požádat jen o jedno z nich nebo více. Pokud je použito více typů scope, jsou odděleny mezerou.
state	Ano	Libovolný string [min 128 bits]	Parametrem se zvyšuje bezpečnost komunikace při přesměrování. Chrání před útoky CSRF a předává informace z aplikace prostřednictvím toku autentizace.

Response			
Atribut	**Mandatory**	**Typ**	**Popis**
Code	Ano	String	Jednorázový Autorizační kód
State	Ano	String	Hodnota atributu předaného z TPP požadavku

Chybové kódy

› Chybové kódy jsou definovány podle [1] RFC 6749, kapitola 4.1.2.1

Příklad URL na autentizaci:
https://api.bankservis.cz/oauth2/auth?state=profil&redirect_uri=https://www.mypfm.cz/start&client_id=MyPFM&response_type=code_grand&scope=aisp

2.8.2.1.4 Získání tokenů (Get token resource)

Pokud TPP na základě předchozího požadavku (viz kapitolu 2.8.2.1.3) obdrží autorizační kód (**code**) a string uvedený v položce **state** je validní (hodnota state je v odpovědi shodná s hodnotou state, která byla uvedena v požadavku), může TPP zažádat o přístupové tokeny z ASPSP pomocí autorizačního kódu. TPP zašle společně s tímto autorizačním kódem (který musí být uveden v těle požadavku) **i client_id a client_secret.**

Endpoint: POST https://api.bankservis.cz/oauth2**/token**

Request			
Atribut	**Mandatory**	**Typ**	**Popis**
code	Ano	string	Autorizační code navrácený z autentizačního flow (code grant)
client_id	Ano	String	Id aplikace TPP
client_secret	Ano	String	Bezpečnostní kód, vydeaný bankou pro aplikaci (client_id) TPP
redirect_uri	Ano	URL	URL redirectu shodné s URL předaném v autentizačním requestu
grant_type	Ano	**authorization_code**	Podle stávající definice/zvyklosti OAuth2 bude tato hodnota **authorization_code**, pokud dochází k výměně code za dvojici tokenů access_token a refresh_token.

Response			
Atribut	**Mandatory**	**Typ**	**Popis**
access_token	Ano	string	Krátkodobý (v některých případech jednorázový) token (platnost tokenu je 3600s), který je možné znovu vygenerovat použitím refresh_tokenu. Tento token slouží k autorizaci requestu na API.
expires_in	Ano	number	Zbývající čas do expirace access_tokenu – ve vteřinách.
refresh_token	Ano	String	Dlouhodobý token (platnost 90 dní) vydaný na základě výměny za jednorázový code.
token_type	Ano	String	Typ tokenu “Bearer”
acr	Ne	Number	Úroveň ověření. Nabývá hodnot 0 až 4. Default 3. Hodnota „0“ odpovídá nonSCA.
scope	Ne	String	Seznam Scope oddělených mezerou, pro které je token vydán (položka navíc oproti COBS).

Chybové kódy

- Chybové kódy jsou definovány podle [1] RFC 6749, kapitola 5.2

Příklad použití viz zdroj [7] kapitola 1.4.4. 2a Get token resource

2.8.2.1.5 Obnovení Access tokenu

TPP může po expiraci access_tokenu prostřednictvím refresh tokenu požádat o nový. Pro to je možné použít resouce „Získání tokenů“ s níže uvedenými parametry.

Endpoint: POST https://api.bankservis.cz/oauth2**/token**

Request			
Atribut	**Mandatory**	**Typ**	**Popis**
client_id	Ne	string	ID aplikace TPP
grant_type	Ano	**refresh_token**	Podle stávající definice/zvyklosti OAuth2 bude tato hodnota **refresh_token**, pokud dochází k obnovení access_tokenu na základě refresh_token.
refresh_token	Ano	String	Validní **refresh_token, za který se provádí výměna access_tokenu**

Response			
Atribut	**Mandatory**	**Typ**	**Popis**
access_token	Ano	string	Krátkodobý (v některých případech jednorázový) token (platnost tokenu je 3600s), který je možné znovu vygenerovat použitím refresh_tokenu. Tento token slouží k autorizaci requestu na API.
token_type	Ano	String	Typ tokenu “Bearer”
expires_in	Ano	number	Zbývající čas do expirace access_tokenu – ve vteřinách.
Acr	Ne	number	Úroveň ověření. Nabývá hodnot 0 až 4. Default 3 nebo 4. Hodnota „0“ automaticky odpovídá nonSCA. Hodnoty 1 až 4 odpovídají hodnotám definovaným normou ISO 29115.

Chybové kódy

- Chybové kódy jsou definovány podle [1] RFC 6749, kapitola 5.2

Příklad použití viz zdroj [7] kapitola 5.2.4.

2.9 Popis metod, které jsou k dispozici poskytovatelům služeb (TPP) přes PSD2 API

2.9.1 Služby pro AISP (Dotazy k účtům, přehled transakcí)

Kapitola definuje seznam metod poskytovaných pro AISP.

2.9.1.1 Předpoklady pro používání metod API pro AISP

a/ je vyžadováno použití certifikátu TPP - TPP je na základě licenčního čísla (včetně použitého prefixu) uvedeného v certifikátu, který TPP používá při komunikaci dohledáno v databázi IB v tabulce TPP – identické licenční číslo musí být uvedeno v záznamu TPP v položce **IdentifierInCertificate** v databázi IB
b/ dohledaný záznam TPP je platný,

c/ TPP má v záznamu v databázi IB povolenu službu AISP
d/ registrovaná aplikace TPP má povolenou službu AISP
e/ v certifikátu, který používá TPP při komunikaci je uvedena služba AISP
f/ TPP použil v hlavičce požadavku access_token (vygenerovaný v kontextu “OAuth2 Authorization Code Grant”), na jehož základě je na straně banky dohledán ve vazbách Disponent-aplikace TPP platný souhlas vytvořený disponentem.
g/ je vyžadována autorizace klientem (aplikace TPP má v dohledaném souhlasu povolenu od disponenta službu AISP)

2.9.1.2 Seznam metod používaných pro službu AISP

Endpoint	Metoda	Popis
/api/v1/**accounts/{id}/balance**	GET	Zůstatek na účtu - prostřednictvím této služby dostane disponentem autorizovaná třetí strana přehled zůstatků bankovního účtu disponenta vedeného v dané bance
/api/v1/**accounts/{id}/transactions**	GET	Přehled transakcí- prostřednictvím této služby dostane disponentem autorizovaná třetí strana přehled transakcí
/api/v2/accounts	GET	Seznam platebních účtů klienta - služba na požadavek vrátí seznam účtů, které disponent uvedl v souhlasu k používání s konkrétním TPP (nikoliv seznam všech účtů disponenta) bez zůstatků

2.9.1.3 Definice hlavičky

Hlavička pro Request

Attribute	Mandatory	Typ	Popis
Content-Type	Ano	String	Definuje typ média MIME zdroje. Například aplikace / json nebo application / x-www-form-urlencoded (zdroje OAuth2 / auth)
Authorization	Ano	String	Typ autorizace definovaný dle RFC 6750 - The OAuth 2.0 Authorization Framework: Bearer Token Usage
TPP-Name	Ano	Text	Jméno providera (třetí strany), který vytvořil požadavek
TPP-Identification	Ne	Text	Identifikace (licenční číslo) provider (třetí strany), který vytvořil požadavek

Hlavička pro Response

Attribute	Mandatory	Typ	Popis
Content-Type	Ano	String	application/json nebo application/xml

2.9.1.4 AISP operace: Zůstatek na účtu

Zůstatek konkrétního účtu klienta podle referenčního id účtu.

Endpoint: GET https://api.bankservis.cz/api/**v1/accounts/{id}/balance**

Query parametry requestu

Metoda: accounts/{id}/balance			
Název atributu	Formát	Mandatory	Poznámka
Id	Text	Ano	API Identifikátor platebního účtu z odpovědi na dotaz o přehledu účtů. Účet musí být obsažen v dohledaném souhlasu od disponenta

Response

Metoda: accounts/{id}/balance			
Název atributu	Formát	Mandatory	Poznámka
balances	typ:ArrayOfAccountsInformationResponseBalance	Ano	Pole zůstatků

Chybové kódy		
HTTP Status	Error kód	Popis
400	**parameter_missing**	Chybí povinný parametr.
400	**parameter_invalid**	Nevalidní hodnota vstupního parametru.
401	**UNAUTHORISED**	Chybějící access token = uživatel není autentizován Chybějící certifikát = provider není autentizován
403	**FORBIDDEN**	Autentizace neplatným certifikátem nebo expirovaným access tokenem, příp. volání, které neodpovídá licenci třetí strany.
404	**ID_NOT_FOUND**	Neplatné nebo neznámé ID účtu
500, 503	**server_error**	Chyba autorizačního serveru.
Použití ostatních http status kódů a chybových kódů dle [1] RFC 6749, kapitola 5.2		

2.9.1.4.1 Definice typu ArrayOfAccountsInformationResponseBalance

	Metoda: accounts/{id}/balance - typ: ArrayOfAccountsInformationResponseBalance				
Název atributu atributu (každý sloupec představuje jednu úroveň v JSON struktuře)			Formát	Mandatory	Poznámka
amount	value		decimal (2 desetinná místa)	Ano	Hodnota zůstatku
	currency		String (3)	Ano	Kód měny zůstatku podle ISO 4217 - 3 velká písmena
creditDebitIndicator			enum	Ano	Zkratka Indikátoru Kredit / Debet CRDT (Kredit) DBIT (Debet)
date	dateTime		dateTime	Ano	Datum aktualizace zůstatku
type	codeOrProprietary	code	enum	Ano	Typ zůstatku CLAV (disponibilní zůstatek) CLBD (aktuální zůstatek)

2.9.1.5 AISP operace: Přehled transakcí

Prostřednictvím této služby dostane disponentem autorizovaná třetí strana přehled transakcí provedených na bankovním účtu zákazníka v rámci zadaného termínu. Historie transakcí zahrnuje pouze transakce, které ovlivňují zůstatek (rezervace, zaúčtované transakce). Transakce jsou řazeny od nejnovější po nejstarší.

Endpoint: POST https://api.bankservis.cz/api/**v1** **/accounts/{id}/transactions{?fromDate,toDate,page,size}**

Query parametry requestu

Metoda: /accounts/{id}/transactions{?fromDate,toDate,page,size}			
Název atributu	Formát	Mandatory	Poznámka
id	String	Ano	API Identifikátor platebního účtu z odpovědi na dotaz o přehledu účtů Účet musí být obsažen v dohledaném souhlasu od disponenta
fromDate	dateTime	Ne	Datum počátku období pro historii transakcí. Výchozí hodnota je aktuální den
toDate	dateTime	Ne	Datum konce období pro historii transakcí. Banka umožňuje zobrazit transakce staré max. 2roky Výchozí hodnota je aktuální den.
page	integer	Ne	Pořadové číslo stránky s ohledem na velikost stránky pro záznamovou sadu. Výchozí hodnota je 0 (první stránka).
pageSize	integer	Ne	Počet záznamů zahrnutých na jedné stránce pro zobrazení. Výchozí hodnota je 50 záznamů. Maximální povolená hodnota je **100** záznamů na stránku.

Response

Metoda: /accounts/{id}/transactions			
Název atributu	Formát	Mandatory	Poznámka
pageCount	integer	Ne	Celkový počet stránek
Transactions	typ: ArrayOfAccountsTransactionsResponseTransaction	Ano	Pole transakcí

Chybové kódy		
HTTP Status	**Error kód**	**Popis**
400	**parameter_missing**	Chybí povinný parametr.
400	**parameter_invalid**	Nevalidní hodnota vstupního parametru.
400	**DT01**	[InvalidDate] Neplatné datum
401	**UNAUTHORISED**	Chybějící access token = uživatel není autentizován Chybějící certifikát = provider není autentizován
403	**FORBIDDEN**	Autentizace neplatným certifikátem nebo expirovaným access tokenem, příp. volání, které neodpovídá licenci třetí strany.
404	**ID_NOT_FOUND**	Neplatné nebo neznámé ID účtu
404	**PAGE_NOT_FOUND**	Dotaz na neexistující stránku
500, 503	**server_error**	Chyba autorizačního serveru.

Příklad použití viz zdroj [7] kapitola 5.2.6.

2.9.1.5.1 Definice typu ArrayOfAccountsTransactionResponseTransaction

Metoda: accounts/transactions - typ: ArrayOfAccountsTransactionsResponseTransaction					
Název atributu (každý sloupec představuje jednu úroveň v JSON struktuře)			Formát	Mandatory	Poznámka
entryReference			String (35)	Ne	bankou přidělený jednoznačný identifikátor transakce
amount	**value**		decimal (2 desetinná místa)	Ano	Hodnota částky transakce.
	currency		String (3)	Ano	Měna částky transakce podle ISO 4217 - 3 velká písmena
creditDebitIndicator			enum	Ano	Zkratka Indikátoru Kredit / Debet CRDT (Kredit) DBIT (Debet)
reversalIndicator			boolean	Ne	Příznak určuje, zda se jedná o reverzní transakci (storno) true: Jedná se o storno false: Nejedná se o storno
status			enum	Ano	Stav položky (odepsané nebo připsané platby) na účtu z pohledu banky. Ve výpise se zobrazují pouze: - zaúčtované položky (BOOK), - blokované položky, (PDNG).
bookingDate	**date**		dateTime	Ano	Datum zpracování/zaúčtování platby bankou ve formátu ISODate, resp. ISODateTime, tj. YYYYMM-DD, popř. YYYY-MM-DDThh:mm:ss.sTZD.
valueDate	**date**		dateTime	Ano	Datum splatnosti/valuty platby ve formátu ISODate, resp. ISODateTime, tj. YYYYMM-DD, popř. YYYY-MM-DDThh:mm:ss.sTZD.
bankTransactionCode	proprietary	**code**	String	Ano	Kód kategorie typu transakce ze seznamu kódů SBA.
		issuer		Ano	Identifikace vydavatele číselníku kódů bankovních transakcí, která nabývá hodnoty CBA (CBA = česká bankovní asociace)

entryDetails	transactionDetails		typ:AccountsTransactionsResponseTransactionDetail	Ano	Položky detailu transakce (detail obratu). Každý detail obratu začíná touto dvojící:
					"entryDetails": {
					"transactionDetails": {

2.9.1.5.2 Definice typu AccountsTransactionsResponseTransactionDetail

Metoda: accounts/transactions - typ: ArrayOfAccountsTransactionsResponseTransaction							
Název atributu (každý sloupec představuje jednu úroveň v JSON struktuře)					Formát	Mandatory	Poznámka
additionalTransactionInformation					Text (500)	Ne	Popis bankovní transakce
amountDetails	instructedAmount	**amount**	**value**		Decimal (2 desetinná místa)	Ano, pokud je použit „InstructedAmounts“	Hodnota částky transakce.
			currency		String (3)		Měna částky transakce podle ISO 4217 - 3 velká písmena
	counterValueAmount	**amount**	**value**		Decimal (2 desetinná místa)	Ano, pokud je použi „CounterValueAmount“	Konečná částka a měna platby, která byla klientem požadována převést.
			currency		String (3)		
		currencyExchange	**sourceCurrency**		String (3)	Ano, pokud je použit „currencyExchange“	Měna účtu klienta (původní měna).
			targetCurrency		String (3)	Ne	Měna platby (cílová měna).
			exchangeRate		Decimal (2 desetinná místa)	Ano, pokud je použit „currencyExchange“	Použitý směnný kurz pro konverzi z instruované měny na měnu cílového účtu.
references	accountServicerReference				String (35)	Ne	Jedinečné ID transakce generované bankou (jedná se např. i číslo, pod kterým byla platba uložena v IB).
	clearingSystemReference				String (35)	Ne	Bankou definovaná číselníková hodnota identifikující typ platby nebo používaný název typu platby. U karetních transakcí - identifikace karetní asociace.
	chequeNumber				String (35)	Ne	Používáno u karetních transakcí Číslo karty ve formátu **** **** **** 1111
	endToEndIdentification				String (35)	Ne	Jedinečná identifikace zadaná klientem iniciujícím platbu, která slouží pro nezaměnitelnou (zde může být vyplněný např. variabilní symbol.)

	paymentInformationIdentification				String (35)	Ne	Další/jiná bankovní reference přiřazená platbě přidělené bankou, u plateb z platebních karet může být doplněno sekvenční číslo platební karty, příp. zde může být vyplněný specifický symbol.
	mandateIdentification				String (35)	Ne	Identifikace platby zadaná třetí stranou, příp. zde může být vyplněný konstantní symbol (pro SEPA inkasa uvedeno Unique Mandate Reference pro dané SEPA inkaso, jako povinné pole)
	messageIdentification				String (35)	Ne	ID platby (převzatá identifikace platby zadaná klientem při jejím iniciování)
relatedAgents	creditorAgent	financialInstitutionIdentification	**Bic**		String (11)	Ano, pokud je použit „creditorAgent“ a není použit „memberIdentification“	BIC / SWIFT kód banky příjemce
			clearingSystemMemberIdentification	**memberIdentification**	String (35)	Ano, pokud je použit „creditorAgent“ a není použit „Bic“	kód banky příjemce dle číselníku bank vedeného u ČNB
	debtorAgent	financialInstitutionIdentification	**Bic**		String (11)	Ano, pokud je použit „debtorAgent“ a není použit „memberIdentification“	BIC / SWIFT kód banky plátce
			clearingSystemMemberIdentification	**memberIdentification**	String (35)	Ano, pokud je použit „debtorAgent“ a není použit „Bic“	kód banky plátce dle číselníku bank vedeného u ČNB
relatedDates	acceptanceDateTime				Date	Ne	Datum zadání transakce (datum přijetí transakce v bance).
relatedParties	creditor	name			String (70)	Ne	Jméno příjemce.

	creditorAccount	identification	**iban**		IBAN2007Identifier	Ano, pokud není zadán „Other“	IBAN příjemce
			other	**identification**	String (34)	Ano, pokud není zadán „iban“	číslo účtu příjemce v národním (CZ) formátu
	debtor	**Name**			String	Ne	Jméno plátce
	debtorAccount	Identification	**iban**		IBAN2007Identifier	Ano, pokud není zadán „Other“	Jedinečná identifikace účtu plátce (IBAN).
			other	**identification**	String (34)	Ano, pokud není zadán „iban“	číslo účtu v národním (CZ) formátu
	tradingParty	Identification			String (35)	Ne	Jedinečná identifikace třetí strany. Pro karetní transakce je zde uváděno ID obchodníka.
		merchantCode			String (4)	Ne	Kód kódu obchodníka (MCC) koordinovaný společností MasterCard a Visa.
		name			String	Ne	Jméno třetí strany. Pro karetní transakce je zde uváděno jméno obchodníka.
remittanceInformation	unstructured				Text(140)	Ne	Text pro příjemce transakce
	structured	creditorReferenceInformation	**reference**		String (35)	Ne	Kolekce symbolů platby. První 2 znaky každé hodnoty v poli definují typ symbolu. Za dvojtečkou následuje hodnota symbolu (max. 10 čísel). Např. "reference": ["VS:123"] znamená variabilní symbol=123 Pokud v platbě nebyl variabilní, specifický nebo konstantní symbol vyplněn, pak zůstane celá struktura Structured prázdná.

2.9.1.6 AISP operace: Account List

Endpoint: GET https://api.bankservis.cz/api/**v1/accounts**

Request

Tělo požadavku neobsahuje žádné atributy.

Response (pokud nedojde při zpracování požadavku k chybě)

Metoda: accounts			
Název atributu	Formát	Mandatory	Poznámka
accounts	typ: ArrayOfAccountInfo	Ano	Pole zůstatků

Chybové kódy		
HTTP Status	**Error kód**	**Popis**
400	**parameter_invalid**	Nevalidní hodnota vstupního parametru.
401	**unauthorised**	Chybějící access token = uživatel není autentizován Chybějící certifikát = provider není autentizován
403	**forbidden**	Autentizace neplatným certifikátem nebo expirovaným access tokenem, příp. volání, které neodpovídá licenci třetí strany.
500, 503	**server_error**	Chyba autorizačního serveru.
Použití ostatních http status kódů a chybových kódů dle [1] RFC 6749, kapitola 5.2		

Příklad použití viz zdroj [7] kapitola 3.1.3.

2.9.1.6.1 Definice typu ArrayOfAccountsInfo

Metoda: accounts - typ: ArrayOfAccountInfo				
Název atributu (každý sloupec představuje jednu úroveň v JSON struktuře)		Formát	Mandatory	Poznámka
id		Text	Ano	Jedinečný API Identifikátor účtu disponenta
identification	iban	String (34)	Ne	IBAN účtu disponenta (odchylka od standardu COBS, položka IBAN není uváděna, pokud se jedná o číslo účtu klientů FT (finanční trhy))
	other	String	Ne	číslo účtu v národním (CZ) formátu
currency		String (3)	Ano	Měna účtu (kód měny podle ISO 4217 - 3 velká písmena)
nameI18N		String	Ne	název účtu nebo uživatelské pojmenování účtu, pokud je dostupné
productI18N		String	Ne	Název produktu
servicer	bankCode	Text	Ne	Kód banky
	countryCode	String (2)	Ne	Země banky podle ISO 3166 (2 znaky)
	Bic	String (35)	Ne	BIC kód banky (ASPSP)

2.9.2 Služby pro PISP (Vytvoření platby, Zjišťování stavu platby, Autorizace platby)

Kapitola definuje seznam metod poskytovaných pro PISP.

Upozornění: prostřednictvím popsaného API může TPP aplikace obsluhovat POUZE ty pokyny, které byly zadány jí samotnou.

2.9.2.1 Předpoklady pro používání metod API pro službu PISP

a/ je vyžadováno použití certifikátu TPP - TPP je na základě licenčního čísla (včetně použitého prefixu) uvedeného v certifikátu, který TPP používá při komunikaci dohledáno v databázi IB v tabulce TPP – identické licenční číslo musí být uvedeno v záznamu TPP v položce **IdentifierInCertificate** v databázi IB
b/ dohledaný záznam TPP je platný,
c/ TPP má v záznamu v databázi IB povolenu službu PISP
d/ registrovaná aplikace TPP má povolenou službu PISP
e/ v certifikátu, který používá TPP při komunikaci je uvedena služba PISP
f/ TPP použil v hlavičce požadavku access_token (vygenerovaný v kontextu “OAuth2 Authorization Code Grant”), na jehož základě je na straně banky dohledán ve vazbách Disponent-aplikace TPP platný souhlas vytvořený disponentem.
g/ je vyžadována autorizace klientem (aplikace TPP má v dohledaném souhlasu povolenu od disponenta službu PISP)

2.9.2.2 Seznam metod používaných pro službu PISP

Endpoint	Metoda	Popis
/api/v1/**accounts/balanceCheck**	POST	**Dotaz na dostatek prostředků** - prostřednictvím této metody si TPP může ověřit, zdali má klient na bankovním účtu, ke kterému TPP vydala kartu, dostatek prostředků k zrealizování transakce kartou
/api/v1/**payments**	POST	**Nová platba (inicializace platby)** - prostřednictvím této metody disponentem autorizovaná třetí strana iniciuje (vytvoří) jeden non eCommerce platební příkaz z bankovního účtu disponenta. TPP musí následně musí inciovat autorizaci platby (workflow viz kapitolu 2.5.2.3)
/api/v1/**payments/{paymentId}**	GET	**Info o založené/iniciované platbě** - Poskytne detaily o platbě založené prostřednictvím rozhraní „Inicializace platby“
/api/v1/**payments/{paymentId}/status**	GET	**Dotaz na status založené/iniciované platby –** prostřednictvím této služby je TPP umožněno zjišťování stavu platebního příkazu
/api/v1/**payments/{paymentId}**	DELETE	**Smazání založené neautorizované platby -** prostřednictvím této služby je umožněno zrušení platby, která ještě nebyla autorizovaná a která byla vytvořena prostřednictvím služby PISP Nová platba (iniciace platby)
/api//v1/**payments/{paymentId}/sign**	POST	**Generování autorizačního ID -** Vygeneruje nové SignId.
/api//v1/**payments/{paymentId}/sign/{signId}**	POST	**Inicializace autorizace platby** – na základě zavolání této metody třetí strana získá v odpovědi URL pro provedení Federované autorizace vybraného příkazu k úhradě. V rámci řešení je podporována pouze Federovaná autorizace platby – přešměrování klienta na centrální autoentizační stránku banky, kde klient následně platbu po příhlášení autorizuje svým autorizačním zařízením.

2.9.2.3 Definice hlavičky

Hlavička pro Request

Attribute	Mandatory	Typ	Popis
Content-Type	Ano	String	Definuje typ média MIME zdroje. Například aplikace / json nebo application / x-www-form-urlencoded (zdroje OAuth2 / auth)
Authorization	Ano	String	Typ autorizace definovaný dle RFC 6750 - The OAuth 2.0 Authorization Framework: Bearer Token Usage
TPP-Name	Ano	Text	Jméno providera (třetí strany), který vytvořil požadavek
TPP-Identification	Ne	Text	Identifikace (licenční číslo) provider (třetí strany), který vytvořil požadavek

Hlavička pro Response

Attribute	Mandatory	Typ	Popis
Content-Type	Ano	String	application/json nebo application/xml

2.9.2.4 **PISP Operace: Dotaz na dostatek prostředků**

Prostřednictvím této metody si TPP může ověřit, zdali má klient na bankovním účtu, ke kterému TPP vydala kartu, dostatek prostředků k zrealizování transakce kartou

Endpoint: POST https://api.bankservis.cz/api/v1/payments/balanceCheck

Request

	Metoda: accounts/balanceCheck					
Název atributu				Formát	Mandatory	Poznámka
exchangeIdentification				String (18)	Ano	Jednoznačná identifikace dotazu
Card	cardHolderName				Ne	jméno držitele karty
	maskedPAN				Ano, pokud je použit „Card“	maskované číslo karty
debtorAccount	identification	iban		IBAN2007Identifier	Ne	IBAN účtu - položka je nepovinná, neboť pro účty klientů typu FT nelze IBAN používat Účet musí být obsažen v dohledaném souhlasu od disponenta
		other	Identification	String (34)	Ano, pokud není použit „iban“	Odchylka od COBS – nová položka
	currency			String (3)	Ne	Měna účtu plátce podle ISO 4217
authenticationMethod				Enum	Ne	Metoda ověření klienta Povolené hodnoty enumu – viz kapitolu 2.9.3.6
Merchant	identification			String (35)	Ano, pokud je použita úroveň 1„merchant“	Identifikace obchodníka
	type				Ne	Typ obchodníka
	shortName			String (35)	Ano, pokud je použita úroveň 1„merchant“	Název obchodníka
	commonName			String (70)	Ano, pokud je použita úroveň 1„merchant“	Jméno obchodníka tak, jak je uvedeno na potvrzení o platbě
	address			Text (140)	Ne	Adresa obchodníka
	countryCode			String (2)	Ne	Dvouznakový kód země obchodníka podle normy ISO3166.
	merchantCategoryCode			String (4)	Ano, pokud je použita úroveň 1„merchant“	Kód obchodníka v návaznosti na typ obchodu podle normy ISO 18245
transactionDetails	totalAmoun			Decimal (2 desetinná místa)	Ano	Hodnota částky transakce.
	currency			String (3)	Ano	Měna částky transakce podle ISO 4217 - 3 velká písmena

Response (pokud nedojde při zpracování požadavku k chybě)

Metoda: accounts/balanceCheck			
Název atributu	Formát	Mandatory	Poznámka
responseIdentification	Integer	Ano	Jednoznačná identifikace odpovědi na dotaz na dostatek prostředků (ze strany ASPSP)
exchangeIdentification	String (18)	Ano	Zopakovaná identifikace dotazu na dostatek prostředků ze strany vydavatele karty
response	Enum	Ano	Výsledek volání. Může nabývat následujících hodnot APPR (dostatečné finanční prostředky na účtu) DECL (nedostatečné prostředky na účtu)

Chybové kódy		
HTTP Status	**Error kód**	**Popis**
400	**field_missing**	Chybí povinný parametr.
400	**field_invalid**	Nevalidní hodnota vstupního parametru.
400	**AC02**	[InvalidDebtorAccountNumber] – nevalidní identifikátoru účtu v obsahu požadavku.
400	**AC09**	[InvalidAccountCurrency] – uvedena nevalidní měna požadovaného účtu.
401	**Unauthorised**	Chybějící certifikát
403	**Forbidden**	Volání metody, která neodpovídá licenci, nebo neplatný certifikát.
403	**AG01**	[TransactionForbidden] – neexistující souhlas s přístupem k danému typu operace.
400	**AM12**	[InvalidAmount] – chybně zadaná částka.
400, 50x	**NARR**	Narrative – obecný důvod pro odmítnutí platby, s doplněním informace o chybě.
500, 503	**server_error**	Chyba autorizačního serveru.

2.9.2.5 **PISP Operace: Nová platba (inicializace platby)**

Operace umožňuje inicializaci jedné platby ve struktuře JSON.
PISP odešle přes API požadavek obsahující platbu založenou na struktuře JSON.
Odesláním tohoto požadavku se na straně banky vytvoří platební příkaz, který se vztahuje k obchodní transakci mezi PSU a providerem (TPP typu PISP).

Endpoint: POST https://api.bankservis.cz/api**/v1/payments**

Domácí platba k úhradě - Request

		Metoda: payments				
Název atributu				Formát	Mandatory	Poznámka
paymentIdentification	instructionIdentification			String (35)	Ano	identifikace instrukce v aplikaci třetí strany
paymentTypeInformation	instructionPriority			string	Ne	priorita instrukce. Pokud není vyplněno, je použita Normální priorita instrukce. (NORM = Normální, HIGH = Expresní)
amount	instructedAmount	value		Decimal (2 desetinná místa)	Ano	Čáska v instrukci
		currency		String (3)	Ano	Měna převodu
requestedExecutionDate				ISODate	Ano	Požadované datum provedení. Oproti COBS se zde jedná o povinnou položku
debtorAccount	identification	iban		IBAN2007Identifier	Ne	Číslo účtu plátce Odchýlení od COBS - položka je nepovinná, neboť pro účty klientů typu FT nelze IBAN používat
		other	identification	String (34)	Ano, pokud není použita položka **iban**	Číslo účtu plátce v lokálním formátu BBAN
	currency			String (3) CurrencyCode ISO 4217	Ne	Měna účtu plátce
creditorAccount	identification	iban		IBAN2007Identifier	Ne	Číslo účtu příjemce Odchýlení odCOBS - položka je nepovinná
		other	identification	String (34)	Ano, pokud není použita položka **iban**	Číslo účtu příjemce v lokálním formátu BBAN
	currency			String (3) CurrencyCode ISO 4217	Ne	Měna účtu příjemce
remittanceInformation	unstructured			Text (140)	Ne	Nestrukturovaná zpráva pro příjemce
	structured	creditorReferenceInformation	reference	String	Ne	Kolekce symbolů platby. První 2 znaky každé hodnoty v poli definují typ symbolu (VS, KS, SS). Za dvojtečkou následuje hodnota symbolu (max. 10 čísel). Např. "reference": ["VS:123"] znamená variabilní symbol=123

SEPA příkaz k úhradě - Request

		Metoda: payments				
Název atributu				Formát	Mandatory	Poznámka
paymentIdentification	instructionIdentification			String (35)	Ano	identifikace instrukce v aplikaci třetí strany
	endToEndIdentification			String (35)	Ano	identifikace domluvená mezi plátcem a příjemcem
paymentTypeInformation	instructionPriority			string	Ne	priorita instrukce. Pokud není vyplněno, je použita Normální priorita instrukce. (NORM = Normální)
amount	instructedAmount	value		Decimal (2 desetinná místa)	Ano	Čáska v instrukci
		currency		String (3)	Ano	Měna převodu (musí být EUR)
requestedExecutionDate				ISODate	Ano	Požadované datum provedení. Oproti COBS se zde jedná o povinnou položku
debtorAccount	identification	iban		IBAN2007Identifier	Ne	IBAN účtu plátce; odchylka od COBS - položka je nepovinná, neboť pro klienty typu FT nelze IBAN používat
		other	identification	String (34)	Ano, pokud není použita položka **iban**	Číslo účtu plátce v lokálním formátu BBAN
	currency			String (3) CurrencyCode ISO 4217	Ne	Měna účtu plátce
creditorAccount	identification	iban		IBAN2007Identifier	Ano	IBAN účtu příjemce
creditorAgent	financialInstitutionIdentification	bic		String (11)	Ano	BIC / SWIFT kód banky příjemce
creditor	name			String (70)	Ano	Jméno příjemce
	postalAddress	streetName		String (70)	Ne	Poštovní adresa (ulice)
		buildingNumber		String (16)	Ne	Poštovní adresa (číslo budovy)
		townName		String (35)	Ne	Poštovní adresa (město)
		postCode		String (16)	Ne	Poštovní adresa (PSČ)
		country		String (2)	Ne	Poštovní adresa (kód země podle ISO3166)

remittanceInformation	unstructured			Text (140)	Ne	Nestrukturovaná zpráva pro příjemce

Zahraniční platba - Request

		Metoda: payments				
Název atributu				Formát	Mandatory	Poznámka
paymentIdentification	instructionIdentification			String (35)	Ano	identifikace instrukce v aplikaci třetí strany
paymentTypeInformation	instructionPriority			string	Ne	priorita instrukce. Pokud není vyplněno, je použita Normální priorita instrukce. (NORM = Normální)
amount	instructedAmount	value		Decimal (2 desetinná místa)	Ano	Čáska v instrukci
		currency		String (3)	Ano	Měna převodu (musí být EUR)
requestedExecutionDate				ISODate	Ano	Požadované datum provedení. Oproti COBS se zde jedná o povinnou položku
chargeBearer				Enum	Ne	Plátce poplatků
debtorAccount	identification	iban		IBAN2007Identifier	Ne	IBAN účtu plátce Odchýlení od COBS - položka je nepovinná
		other	identification	String (34)	Ano, pokud není použita položka **iban**	Číslo účtu plátce v lokálním formátu
	currency			String (3) CurrencyCode ISO 4217	Ne	Měna účtu plátce
creditorAccount	identification	iban		IBAN2007Identifier	Ano	IBAN účtu příjemce
		other	identification	String (34)	Ano	Jiný formát čísla účtu
creditorAgent	financialInstitutionIdentification	bic		String (11)	Ano	BIC / SWIFT kód banky příjemce
creditor	name			String (70)	Ano	Jméno příjemce
	postalAddress	streetName		String (70)	Ne	Poštovní adresa (ulice)
		buildingNumber		String (16)	Ne	Poštovní adresa (číslo budovy)
		townName		String (35)	Ne	Poštovní adresa (město)
		postCode		String (16)	Ne	Poštovní adresa (PSČ)
		country		String (2)	Ne	Poštovní adresa (kód země podle ISO3166)

remittanceInformation	unstructured			Text (140)	Ne	Nestrukturovaná zpráva pro příjemce

Response pro všechny výše uvedené typy plateb (pokud nedojde při zpracování požadavku k chybě)

Struktura výstupu je stejná jako vstupu, a navíc jsou vráceny následující hodnoty:

			Metoda: payments/standard/iso		
Název atributu			Formát	Mandatory	Poznámka
paymentIdentification	transactionIdentification		String	Ano	Číslo příkazu vytvořeného v databázi E-Bankingu, v dalších dotazech se používá jako **{paymentId}** na vstupu
paymentTypeInformation	serviceLevel	code	String	Ano	Typ zadané platby DMCT = Domácí příkaz k úhradě ESCT = SEPA platba XBCT = Zahraniční platba EXCT = Zahraniční platba v rámci EHP NXCT = Zahraniční platba mimo EHP
signInfo	signId		String	Ne	Identifikátor autorizačního procesu konkrétní transakce.
	state		Enum	Ano	Status příkazu Status může nabývat následujících hodnot: ACTC - Authentication and syntactical and semantical validation are successful (v IB bude mít vytvořený příkaz status „K podpisu“)

Chybové kódy		
HTTP Status	**Error kód**	**Popis**
400	**field_missing**	Chybí povinný parametr.
400	**filed_invalid**	Nevalidní hodnota vstupního parametru.
400	**AC02**	[InvalidDebtorAccountNumber] – nevalidní identifikátoru účtu v obsahu požadavku.
400	**AC03**	[InvalidCreditorAccountNumber] - číslo účtu příjemce uvedeno v nevalidním formátu (poznámka: validováno pouze pro in-house platby).
400	**AC10**	[InvalidDebtorAccountCurrency] – uvedená měna účtu plátce neodpovídá měně účtu klienta pro dané číslo účtu vedené v bance (měna účtu je nepovinná).
400	**AM11**	[InvalidTransactionCurrency] – v požadavku je uvedena neobchodovaná/nepodporovaná měna.
400	**AM12**	[InvalidAmount] – chybně zadaná částka.
400	**FF01**	[Invalid File Format] – nevalidní JSON formát, či jiný technický problém se zpracování dotazu.
400	**BE19**	[InvalidChargeBearerCode] - neplatný typ poplatku pro daný typ transakce.
400	**DT01**	[InvalidDate] – Chybné datum splatnosti.
400, 50x	**NARR**	Narrative – obecný důvod pro odmítnutí platby, s doplněním informace o chybě.
400	**RC07**	[InvalidCreditorBICIdentifier] – neplatný SWIFT / BIC kód banky příjemce.
400	**RC10**	[InvalidCreditorClearingSystemMemberIdentifier] - neplatná identifikace kódu banky příjemce.
403		[TransactionForbidden] – neexistující souhlas s přístupem k PISP operaci.
500, 503	**server_error**	Chyba autorizačního serveru.

2.9.2.6 PISP operace: Status založené / iniciované platby

Operace poskytuje informace o stavu zpracování přijaté platební transakce na základě parametru {paymentId}.

Endpoint: GET https://api.bankservis.cz/api/v1/payments/{paymentId}/status

Vstupní URI parametry

- paymentId - identifikátor platby v e-Bankingu, typ: string (povinné)

Request

Tělo požadavku neobsahuje žádné atributy.

Response

Metoda: payments/{paymentId}/status			
Název atributu	Formát	Mandatory	Poznámka
instructionStatus	Enum	Ano	Status příkazu Status může nabývat následujících hodnot: › RJCT (Odmítnuto - Rejected) › PDNG (Autorizováno - Authorized) › ACTC (K podpisu - WaitingForSignatures) › ACSP (Zpracovává se - InProgress, Exportováno - Exported) › ACSC (Akceptováno bankovním systémem) › ACCR (Platba zrušená klientem) › OTHR (rezerva)

Chybové kódy		
HTTP Status	Error kód	Popis
400	parameter_missing	Chybí povinný parametr.
400	parameter_invalid	Nevalidní hodnota vstupního parametru.
500, 503	server_error	Chyba autorizačního serveru.

2.9.2.7 PISP operace: Info o založené / iniciované platbě

Resource pro zobrazení informace přijaté platební transakce na základě parametru **{paymentId}**. Jedná se o platbu, která byla přijata k autorizaci, ale ještě nebyla klientem autorizována. Resource pracuje pouze s transakcemi založenými prostřednictvím konkrétního providera.

Endpoint: GET https://api.bankservis.cz/api/v1/payments/{paymentId}

Vstupní URI parametry

- paymentId - identifikátor platby v e-Bankingu, typ: string (povinné)

Request

Tělo požadavku neobsahuje žádné atributy.

Response

Výstupem zprávy je informace o založené nebo již iniciované platbě. Proto seznam elementů odpovídá elementům z resource (request+response) Nová platba. Viz kapitolu 2.9.2.5.

Chybové kódy		
HTTP Status	Error kód	Popis
401	UNAUTHORISED	Nevalidní/chybějící certifikát = provider není autentizován.
501	NOT_IMPLEMENTED	Neimplementovaná metoda
404	TRANSACTION_MISSING	Volání metody, která neodpovídá licenci, nebo neplatný certifikát.
500, 503	server_error	Chyba autorizačního serveru.

2.9.2.8 PISP operace: Smazání založené neautorizované platby

Operace umožnuje zrušit platbu platbu, která byla inciována prostřednictvím identického providera typu PISP (třetí strany) pomocí služby “Nová platba (iniciace platby)”. Platbu je možné zrušit, dokud platba není autorizována klientem.

Endpoint: DELETE https://api.bankservis.cz/api**/v1/payments/{paymentId}**

Vstupní URI parametry

- paymentId - identifikátor platby v e-Bankingu, typ: string (povinné)

Request

Tělo požadavku neobsahuje žádné atributy.

Response (pokud nedojde při zpracování požadavku k chybě)

Response (HTTP 204) - zadaná platba smazána

Chybové kódy		
HTTP Status	Error kód	Popis
401	UNAUTHORISED	Nevalidní/chybějící certifikát = provider není autentizován.
501	NOT_IMPLEMENTED	Neimplementovaná metoda
404	TRANSACTION_MISSING	Volání metody, která neodpovídá licenci, nebo neplatný certifikát.
500, 503	server_error	Chyba autorizačního serveru.

2.9.2.9 PISP Generování autorizačního ID

Vygeneruje nový požadavek na vygenerování signId pro autorizaci platby. Používá se v případech kdy:

- Platnost původního požadavku na autorizaci (např. vzniklý jako výstup z volání IF-200) vypršela. Platnost požadavků na autorizaci je omezena na 5 minut od jejich založení.
- Autorizace požadavku uživatelem nebyla provedena z vůle uživatele - odmítnut pokyn autorizovat - a má zájem o opakovaný pokus o autorizaci
- Z technických důvodů na straně TPP aplikací

Poznámka: založením nového autorizačního zůstávají v platnosti původní dosud platné a nevyužité požadavky na autorizaci.

Endpoint: POST https://api.bankservis.cz/api/**v1/payments/{paymentId}/sign**

Vstupní URI parametry

- paymentId - identifikátor platby v e-Bankingu, typ: string (povinné)

Request

Tělo požadavku neobsahuje žádné atributy.

Response

	Metoda: payments/{paymentId}/status			
Název atributu		Formát	Mandatory	Poznámka
scenarios		Pole podporovaných autorizačních metod	Ano	kolekce možných způsobů autorizace platby, v tomto případě se vrátí pouze jeden typ autorizace platby - Federovaná autorizace, resp. přesměrování uživatele na stránku banky, kde proběhne autorizace [USERAGENT-REDIRECT]
signInfo	state	string	Ano	Status autorizace příkazu Status v tomto případě může mít pouze hodnotu: › OPEN = Nově vytvořený požadavek na autorizaci (příkaz má interní status k podpisu a signId je platný)
	signId	String	Ano	Jedinečný identifikátor tokenu vygenerovaného pro autorizaci transakce (od vygenerování platí 5 minut)

Chybové kódy		
HTTP Status	**Error kód**	**Popis**
401	**UNAUTHORISED**	Nevalidní/chybějící certifikát = provider není autentizován.
403	**FORBIDDEN**	Nevalidní/chybějící certifikát = provider není autentizován
501	**NOT_IMPLEMENTED**	Neimplementovaná metoda
404	**TRANSACTION_MISSING**	Volání metody, která neodpovídá licenci, nebo neplatný certifikát.
500, 503	**server_error**	Chyba autorizačního serveru.

2.9.2.10 PISP Operace: Inicializace autorizace platby

Tento resource je určen ke startu autorizační metody z vybraného scénáře.
V řešení PSD2 pro Citfin je podporována pouze Federovaná autorizace.
Vstupem je JSON objekt obsahující požadovaný typ autorizační metody - CODE a všechny elementy specifické pro tento krok.
Výstupem tohoto resource je přehled hodnot potřebných pro dokončení autorizace.
Odpovědí pro CODE odpovídající federované autorizaci bude odpovědí URL a parametry pro přesměrování na federovanou autorizační stránku.

Endpoint: POST https://api.bankservis.cz/api**/v1/payments/{paymentId}/sign/{signId}**

Vstupní URI parametry

- paymentId - identifikátor platby v e-Bankingu, typ: string (povinné)
- signId - identifikátor požadavku na autorizaci, typ: string (povinné)

Request

Metoda: payments/{paymentId}/sign/{signId}			
Název atributu	Formát	Mandatory	Poznámka
authorizationType	String	Ano	Vybraný způsob autorizace platby podporovaný bankou Je podporováný pouzenásledující způsob: USERAGENT-REDIRECT = Federovaná autorizace, resp. přesměrování uživatele na stránku banky, kde proběhne autorizace

Response (pokud nedojde při zpracování požadavku k chybě)

	Metoda: payments/{paymentId}/sign/{signId}			
Název atributu		Formát	Mandatory	Poznámka
authorizationType		String	Ano	Vybraný způsob autorizace platby. Další pole odpovědi popisují tento způsob vybrané autorizace USERAGENT-REDIRECT = Federovaná autorizace, resp. přesměrování uživatele na stránku banky, kde proběhne autorizace
href	id	String	Ne	Není podporováno
	url	string	Ano	URI adresa pro autorizaci konkrétního pokynu. Upozornění: kompletní URL adresu pro autorizaci požadavku sestavíte spojením URI specifické pro vystavené bankovní API a tímto vráceným URI
method		string	Ano	HTTP metoda pro přesměrování na Federovanou autorizaci (GET)
signInfo	State	String	Ano	Status autorizace příkazu Status v tomto případě může mít pouze hodnotu: **OPEN** = Nově vytvořený požadavek na autorizaci (příkaz má interní status k podpisu a signId je platný) **DONE** = Autorizace úspěšně provedena (příkaz změnil status) **EXPIRED** = příkaz je k podpisu, ale uživatel neprovedl autorizaci pokynu v době platnosti autorizačního požadavku. Pro opakování autorizace je nutné vytvoření nového požadavku na vygenerování nového SignID a opakovat autorizaci.
	SignId	String	Ano	Jedinečný identifikátor aktuální autorizace transakce

Chybové kódy		
HTTP Status	Error kód	Popis
400	AUTH_LIMIT_EXCEE DED	Vypršela platnost SignId, v okamžiku, kdy banka přijala požadavek na inicializaci autorizace
401	UNAUTHORISED	Nevalidní/chybějící certifikát = provider není autentizován.
403	FORBIDDEN	Nevalidní/chybějící certifikát = provider není autentizován
404	ID_NOT_FOUND	Požadované id neexistuje
500, 503	server_error	Chyba autorizačního serveru.

2.9.3 Služba CISP (Ověření dostatečných prostředků na účtu)

Kapitola definuje seznam metod poskytovaných pro CISP.

2.9.3.1 Předpoklady pro používání metod API pro službu CISP

a/ je vyžadováno použití certifikátu TPP - TPP je na základě licenčního čísla (včetně použitého prefixu) uvedeného v certifikátu, který TPP používá při komunikaci dohledáno v databázi IB v tabulce TPP – identické licenční číslo musí být uvedeno v záznamu TPP v položce **IdentifierInCertificate** v databázi IB
b/ dohledaný záznam TPP je platný,
c/ TPP má v záznamu v databázi IB povolenu službu CISP
d/ registrovaná aplikace TPP má povolenou službu CISP
e/ v certifikátu, který používá TPP při komunikaci je uvedena služba CISP
f/ TPP použil v hlavičce požadavku access_token (vygenerovaný v kontextu “OAuth2 Authorization Code Grant”), na jehož základě je na straně banky dohledán ve vazbách Disponent-aplikace TPP platný souhlas vytvořený disponentem.
g/ je vyžadována autorizace klientem (aplikace TPP má v dohledaném souhlasu povolenu od disponenta službu CISP)

2.9.3.2 Seznam metod používaných pro službu CISP

Endpoint	Metoda	Popis
/api/v1/**accounts/balanceCheck**	POST	Dotaz na dostatek prostředků - prostřednictvím této metody si TPP může ověřit, zdali má klient na bankovním účtu, ke kterému TPP vydala kartu, dostatek prostředků k zrealizování transakce kartou

2.9.3.3 Token pro CISP operaci

Pro CISP operaci bude používán access_token získaný na základě autorizačního resource **Authorization Code Grant** popsaného v kapitole 2.8.2.1.3 nebo případně viz [1], kapitola 4.1.

Generování access_tokenu na základě Client Credentials Grant flow **není v řešení podporováno.**

2.9.3.4 Definice hlavičky

Hlavička pro Request

Attribute	Mandatory	Typ	Popis
Content-Type	Ano	String	Definuje typ média MIME zdroje. Například aplikace / json nebo application / x-www-form-urlencoded (zdroje OAuth2 / auth)
Authorization	Ano	String	Typ autorizace definovaný dle RFC 6750 - The OAuth 2.0 Authorization Framework: Bearer Token Usage
TPP-Name	Ano	Text	Jméno providera (třetí strany), který vytvořil požadavek
TPP-Identification	Ne	Text	Identifikace (licenční číslo) provider (třetí strany), který vytvořil požadavek

Hlavička pro Response

Attribute	Mandatory	Typ	Popis
Content-Type	Ano	String	application/json nebo application/xml

2.9.3.5 CISP Operace: Dotaz na dostatek prostředků

Prostřednictvím této metody si TPP může ověřit, zdali má klient na bankovním účtu, ke kterému TPP vydala kartu, dostatek prostředků k zrealizování transakce kartou.

Dotaz na dostatek prostředků se provádí oproti disponibilnímu zůstatku na účtu (debtorAccount)

Endpoint: POST https://api.bankservis.cz/api/v1/payments/balanceCheck

Request

	Metoda: accounts/balanceCheck					
Název atributu				Formát	Mandatory	Poznámka
exchangeIdentification				String (18)	Ano	Jednoznačná identifikace dotazu
card	cardHolderName				Ne	jméno držitele karty
	maskedPAN				Ano, pokud je použit „Card“	maskované číslo karty
debtorAccount	identification	iban		IBAN2007Identifier	Ne	jednoznačná identifikace účtu plátce, na němž se zjišťuje dostatek prostředků IBAN účtu - položka je nepovinná, neboť pro účty klientů typu FT nelze IBAN používat Účet musí být obsažen v dohledaném souhlasu od disponenta
		other	Identification	String (34)	Ano, pokud není použit „iban“	Odchylka od COBS – nová položka
	currency			String (3)	Ne	Měna účtu plátce podle ISO 4217
authenticationMethod				Enum	Ne	Metoda ověření klienta Povolené hodnoty enumu – viz kapitolu 2.9.3.6
merchant	identification			String (35)	Ano, pokud je použita úroveň 1„merchant“	Identifikace obchodníka
	type				Ne	Typ obchodníka
	shortName			String (35)	Ano, pokud je použita úroveň 1„merchant“	Název obchodníka
	commonName			String (70)	Ano, pokud je použita úroveň 1„merchant“	Jméno obchodníka tak, jak je uvedeno na potvrzení o platbě
	address			Text (140)	Ne	Adresa obchodníka
	countryCode			String (2)	Ne	Dvouznakový kód země obchodníka podle normy ISO3166.

	merchantCategoryCode			String (4)	Ano, pokud je použita úroveň 1„merchant“	Kód obchodníka v návaznosti na typ obchodu podle normy ISO 18245
transactionDetails	totalAmoun			Decimal (2 desetinná místa)	Ano	Hodnota částky transakce.
	currency			String (3)	Ano	Měna částky transakce podle ISO 4217 - 3 velká písmena

Response (pokud nedojde při zpracování požadavku k chybě)

Metoda: accounts/balanceCheck			
Název atributu	Formát	Mandatory	Poznámka
responseIdentification	Integer	Ano	Jednoznačná identifikace odpovědi na dotaz na dostatek prostředků (ze strany ASPSP)
exchangeIdentification	String (18)	Ano	Zopakovaná identifikace dotazu na dostatek prostředků ze strany vydavatele karty
response	Enum	Ano	Výsledek volání. Může nabývat následujících hodnot **APPR** (dostatečné finanční prostředky na účtu) **DECL** (nedostatečné prostředky na účtu)

Chybové kódy		
HTTP Status	Error kód	Popis
400	**field_missing**	Chybí povinný parametr.
400	**field_invalid**	Nevalidní hodnota vstupního parametru.
400	**AC02**	[InvalidDebtorAccountNumber] – nevalidní identifikátoru účtu v obsahu požadavku.
400	**AC09**	[InvalidAccountCurrency] – uvedena nevalidní měna požadovaného účtu.
401	**Unauthorised**	Chybějící certifikát
403	**Forbidden**	Volání metody, která neodpovídá licenci, nebo neplatný certifikát.
403	**AG01**	[TransactionForbidden] – neexistující souhlas s přístupem k danému typu operace.
400	**AM12**	[InvalidAmount] – chybně zadaná částka.
400, 50x	**NARR**	Narrative – obecný důvod pro odmítnutí platby, s doplněním informace o chybě.
500, 503	**server_error**	Chyba autorizačního serveru.

2.9.3.6 **Enum použitý v položce AuthenticationMethod**

Zkratka	Význam
NPIN	On-line PIN authentication (PersonalIdentification Number)
PPSG	Handwritten paper signature.
PSWD	Authentication by a password
SCRT	Electronic commerce transaction secured with the X.509 certificate of a customer
SCNL	Channel-encrypted transaction
SNCT	Secure electronic transaction without cardholder certificate
CPSG	Electronic signature capture (handwritten signature);
ADDB	Cardholder billing address verification
BIOM	Biometric authentication of the cardholder
CDHI	Cardholder data provided for verification, for instance social security number, driver license number, passport number
CRYP	Verification of a cryptogram generated by a chip card or another device, for instance ARQC (Authorisation Request Cryptogram).
CSCV	Verification of Card Security Code
PSVE	Authentication based on statistical cardholder behaviour
CSEC	Authentication performed during a secure electronic commerce transaction
ADDS	Cardholder shipping address verification
TOKP	Verification or authentication related to the use of a payment token, for instance the validation of the authorised use of a token

3. Zdroje

1. *RFC 6749 - The OAuth 2.0 Authorization Framework*, [online]. The Internet Engineering Task Force, October 2012. WWW: https://tools.ietf.org/html/rfc6749

2. *RFC 6750 - The OAuth 2.0 Authorization Framework: Bearer Token Usage,* [online]. The Internet Engineering Task Force, October 2012. WWW: https://tools.ietf.org/html/rfc6750

3. *RFC 7636 - Proof Key for Code Exchange by OAuth Public Clients,* [online]. The Internet Engineering Task Force, September 2015. WWW: https://tools.ietf.org/html/rfc7636

4. *RFC 7519 - JSON Web Token (JWT)*, [online]. The Internet Engineering Task Force, May 2015. WWW: https://tools.ietf.org/html/rfc7519

5. *RFC 7515 - JSON Web Signature (JWS)*, [online]. The Internet Engineering Task Force, May 2015. WWW: https://tools.ietf.org/html/rfc7515

6. *ISO 20022 Financial Services - Universal financial industry message scheme,* [online]. International Organization for Standardization. WWW: https://www.iso20022.org/

7. *Czech Open Banking Standard*, dokument. WWW: https://www.czech-ba.cz/sites/default/files/cobs_rulebook_v02.0-final_vnejsi_web.en_.pdf