OBSAH

1 ÚVOD

Creditas API poskytuje služby přístupné na internetu přes HTTP protokol. Komunikace mezi klientským systémem a bankou vyžaduje zabezpečení pomocí SSL protokolu s minimálně 128 bitovým šifrováním. Konkrétně je požadovaná vzájemná (Two-Way) SSL autentizace a pro navázání spojení musí klientská aplikace použít kvalifikovaný certifikát pro autentizaci webových serverů dle eIDAS. Každá služba má své specifické URL a všechny služby jsou vystaveny metodou POST. Data na vstupu a výstupu volání jsou přenášená ve formátu JSON v těle zprávy. Přesná specifikace služeb je k dispozici ve formátu OpenAPI Specification 2.0 [link]. Pro využití jednotlivých služeb je při odeslání dotazu nutné v HTTP hlavičce vždy uvést platný bezpečnostní klíč typu "Bearer" v parametru "Authorization" (příklad - "Authorization": "Bearer 46a47afa6f1ccbcd7f…"). Bezpečnostní klíč je alfanumerický řetězec o délce 64 znaků, který je potřeba vygenerovat v internetovém bankovnictví nebo pomocí OAuth 2 protokolu.

2 BEZPEČNOSTNÍ MODEL

2.1 Bezpečnostní klíč vytvořený přes OAuth

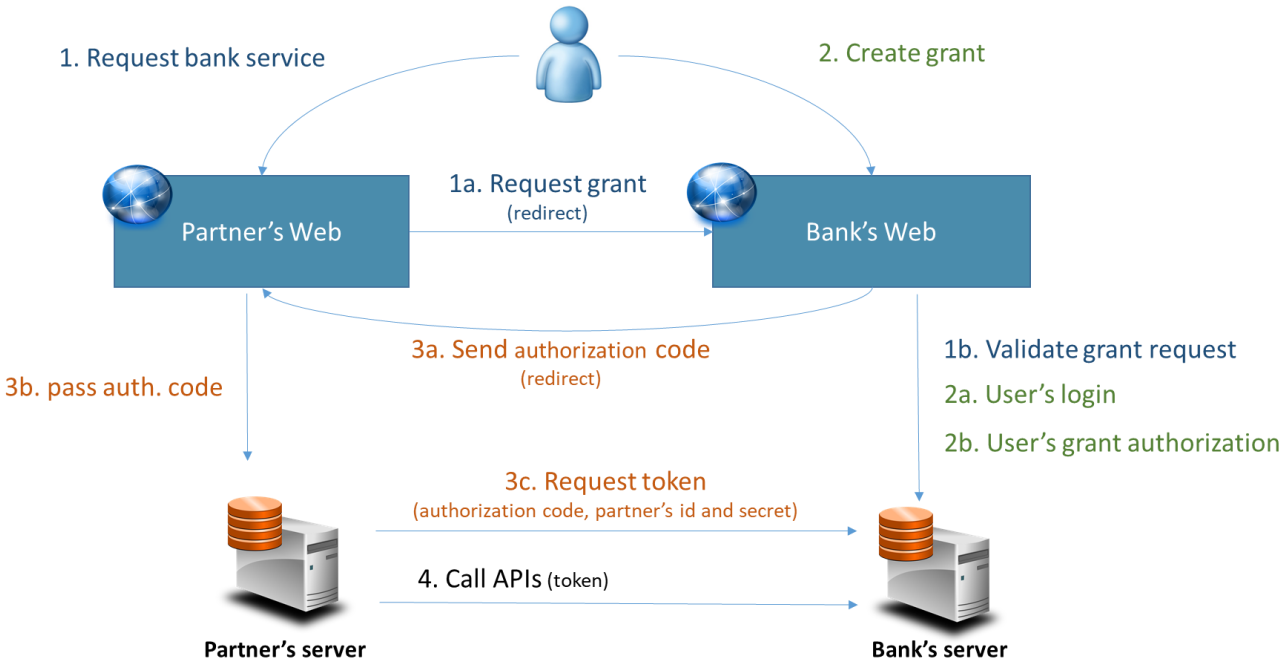
Základní a doporučený bezpečnostní model pro přístup k API je založený na protokolu OAuth 2. Bezpečnostní klíč vzniká udělením souhlasu uživatelem v procesu, kdy je uživatel přesměrován z partnerské aplikace do IB, kde se přihlásí, udělí souhlas, který autorizuje bezpečnostním prvkem a následně je přesměrován zpět do aplikace partnera. Creditas API poskytuje dle OAuth 2 specifikace následující “endpointy“:

1. https://api.creditas.cz/oam/authorize - pro vytvoření souhlasu přístupu k API
2. https://api.creditas.cz/oam/token - pro vygenerování bezpečnostního klíče (access a refresh token)
3. https://api.creditas.cz/oam/revoke - pro zneplatnění bezpečnostního klíče

Požadavek na udělení souhlasu musí obsahovat požadovaný OAuth “scope“. Povolené hodnoty pro definici “scope“ jsou:

- payment
- product_info
- balance_info
- transaction_info

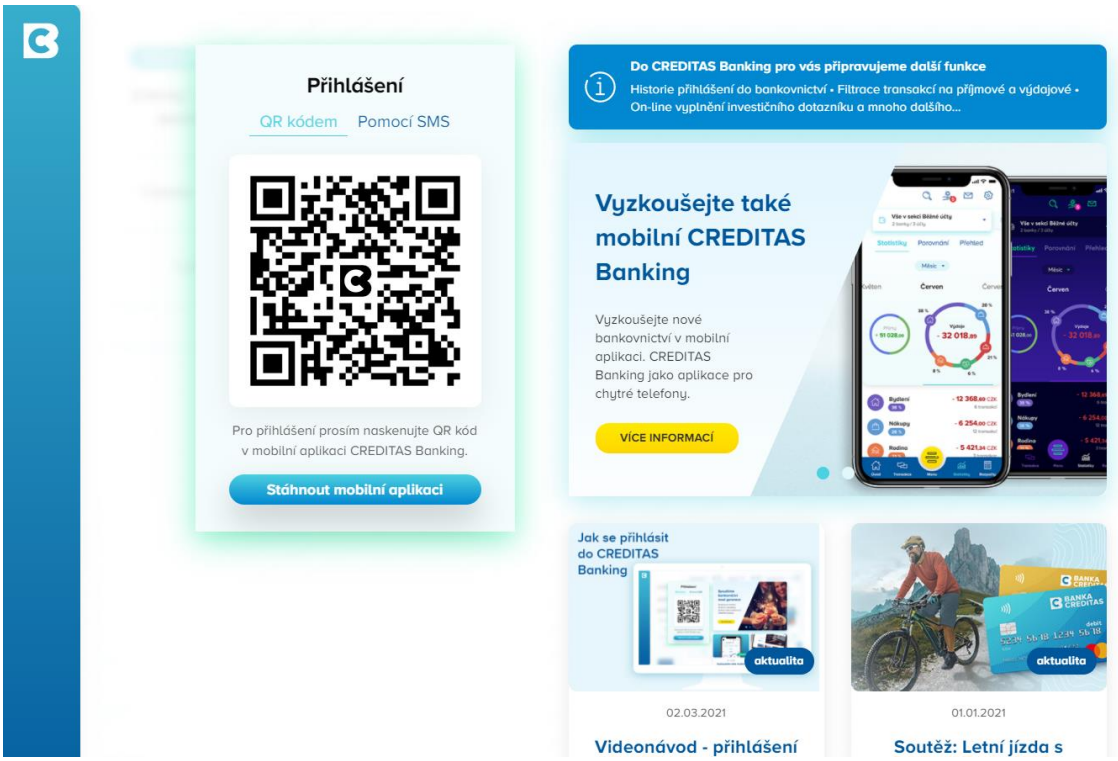
Následující schéma znázorňuje vytvoření souhlasu a vygenerování bezpečnostního klíče pomocí OAuth autorizačního kódu.



1. Uživatel v partnerské aplikaci chce přistoupit k službě poskytované bankou
   a. Aplikace nemá pro uživatele platný bezpečnostní klíč a přesměruje ho na stránky banky s požadavkem na udělení souhlasu pro přístup k API. Na to použije OAuth endpoint “authorize“. Příklad volání dle rfc6749#section-4.1.1: *https://api.creditas.cz/oam/authorize?response_type=code&client_id=394002CRDTS&redirect_uri=https://example.com/client&scope=transaction_info product_info balance_info payment&state=3h5sd*
   b. OAM Server ověří náležitosti grant requestu a vyvolá uživatelský use case na udělení souhlasu

2. Udělení souhlasu

   a. Uživatel se přihlásí pomocí svých přihlašovacích údajů do internetového bankovnictví



   b. Uživatel povolí přístup partnerské aplikaci, který autorizuje svými bankovními bezpečnostními prvky.

3. Vytvoření bezpečnostního klíče
   a. OAM server vygeneruje autorizační kód, který je zaslán v redirect URL, která přesměruje uživatele nazpátek do partnerské aplikace. Příklad redirect URL dle rfc6749#section-4.1.2: *HTTP/1.1 302 Found Location: https://example.com/client?code=SplxlOBeZQQYbYS6WxSbIA&state=3h5sd*
   b. Autorizační kód je předán z prohlížeče na server partnera
   c. Server partnera si vyžádá bezpečnostní klíč zasláním autorizačního kódu na “token“ endpoint dle rfc6749#section-4.1.3: Parametry jsou v uvedeny v tele http dotazu ve formátu "application/x-www-form-urlencoded". Příklad parametrů:

grant_type= authorization_code
code= SplxlOBeZQQYbYS6WxSbIA
redirect_uri= https://example.com/client
client_id= 394002CRDTS
client_secret= a0d1194eaf3833cbe58623be27f164eaf862a11b

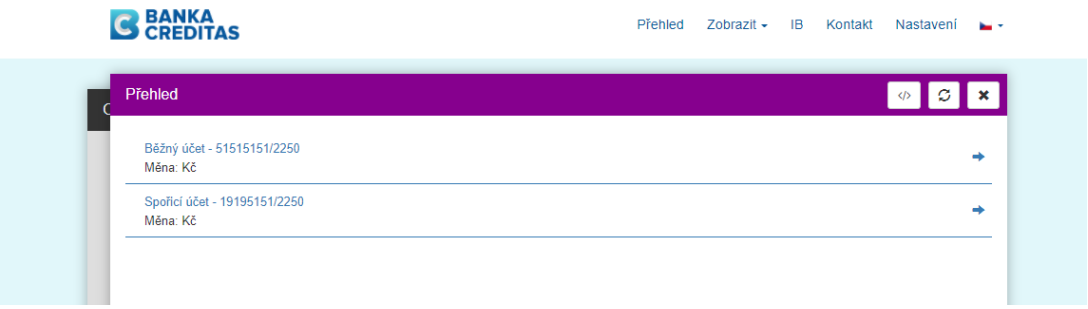
Server pak vrátí bezpečnostní klíč (access a refresh token) v odpovědi dle rfc6749#section-4.1.4. Příklad odpovědi:

HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache

{

```
"access_token":"2YotnFZFEjr1zCsicMWpAA",
 "token_type":"bearer",
 "expires_in":3600,
 "refresh_token":"tGzv3JOkF0XG5Qx2TlKWIA",
}
```


4. Partner server pak použije bezpečnostní klíč pro volání požadovaných bankovních API vystavených na OAM serveru.



2.2 Ručně generovaný bezpečnostní klíč

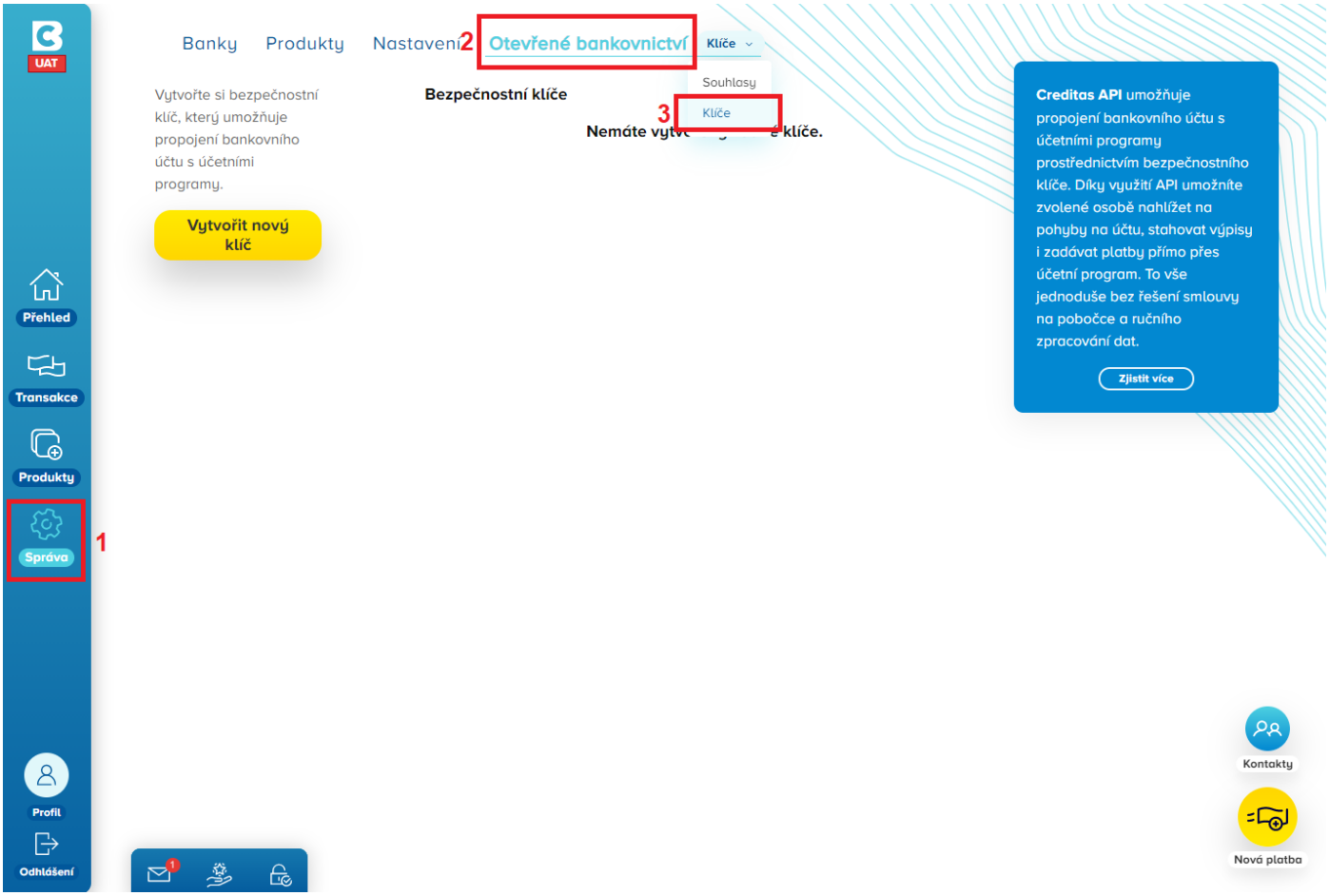
Bezpečnostní klíč vygenerovaný ručně je omezen na použití pro jeden konkrétní účet. Pokud přes API chcete přistupovat k více účtům, tak je potřeba vygenerovat pro každý účet samostatný bezpečnostní klíč. Vygenerovat jej může v internetovém bankovnictví každý klient Banky CREDITAS s potřebným oprávněním.

Postup vytvoření bezpečnostního klíče v IB:

1. Přihlaste se standardně do internetového bankovnictví https://banking.creditas.cz

2. Zvolte záložku **Správa > Otevřené bankovnictví > Klíče**

Po přihlášení do vašeho bankovnictví na adrese https://banking.creditas.cz prosím v hlavním menu zvolte položku **Správa**.
V horní navigaci pak zvolte položku **Otevřené bankovnictví** a položku **Klíče**.

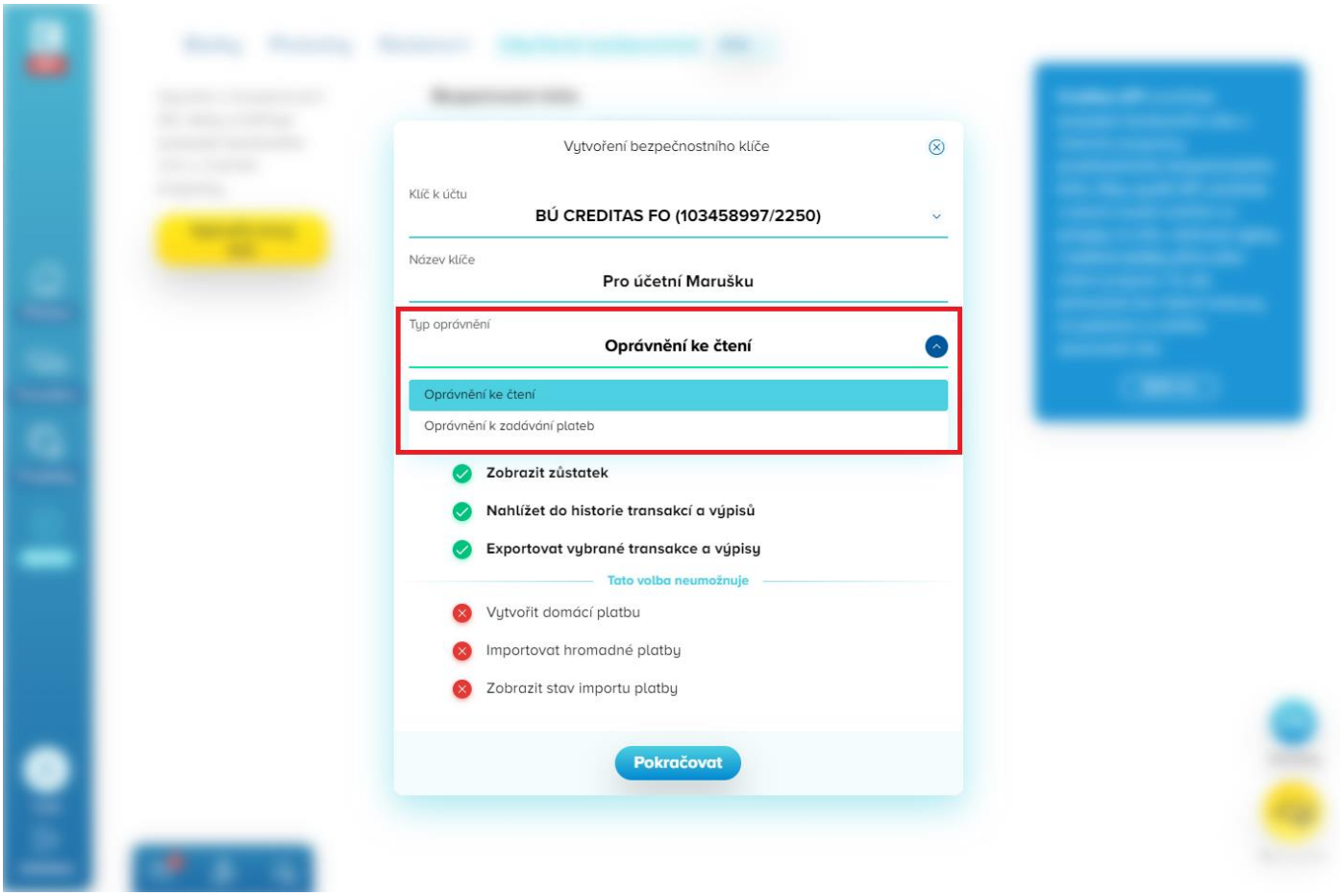


3. Vytvořte **nový klíč**

Vytvořte si bezpečnostní klíč / token, který umožňuje propojení bankovního účtu s účetními programy.

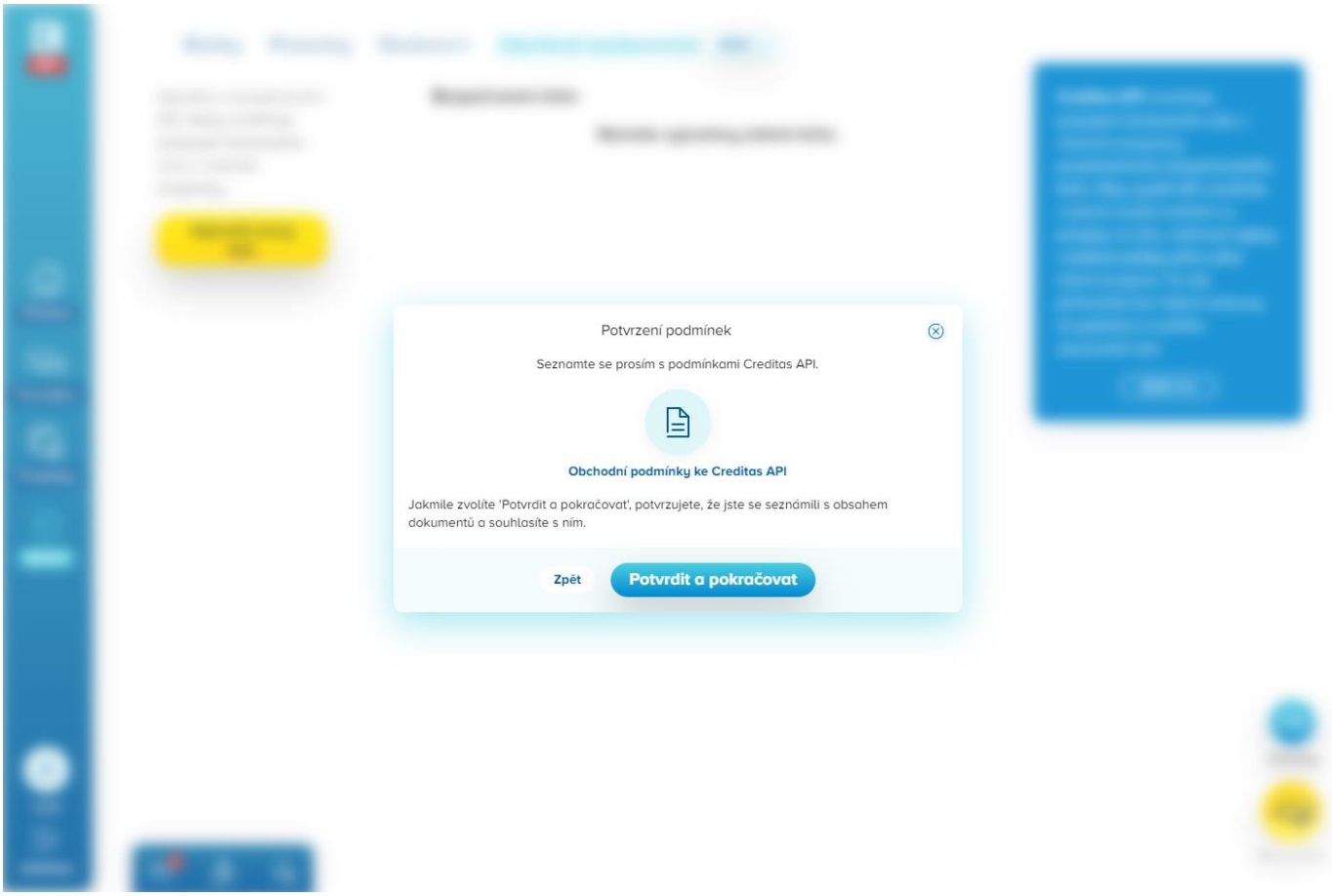
4. Nastavte parametry klíče a vyberte oprávnění

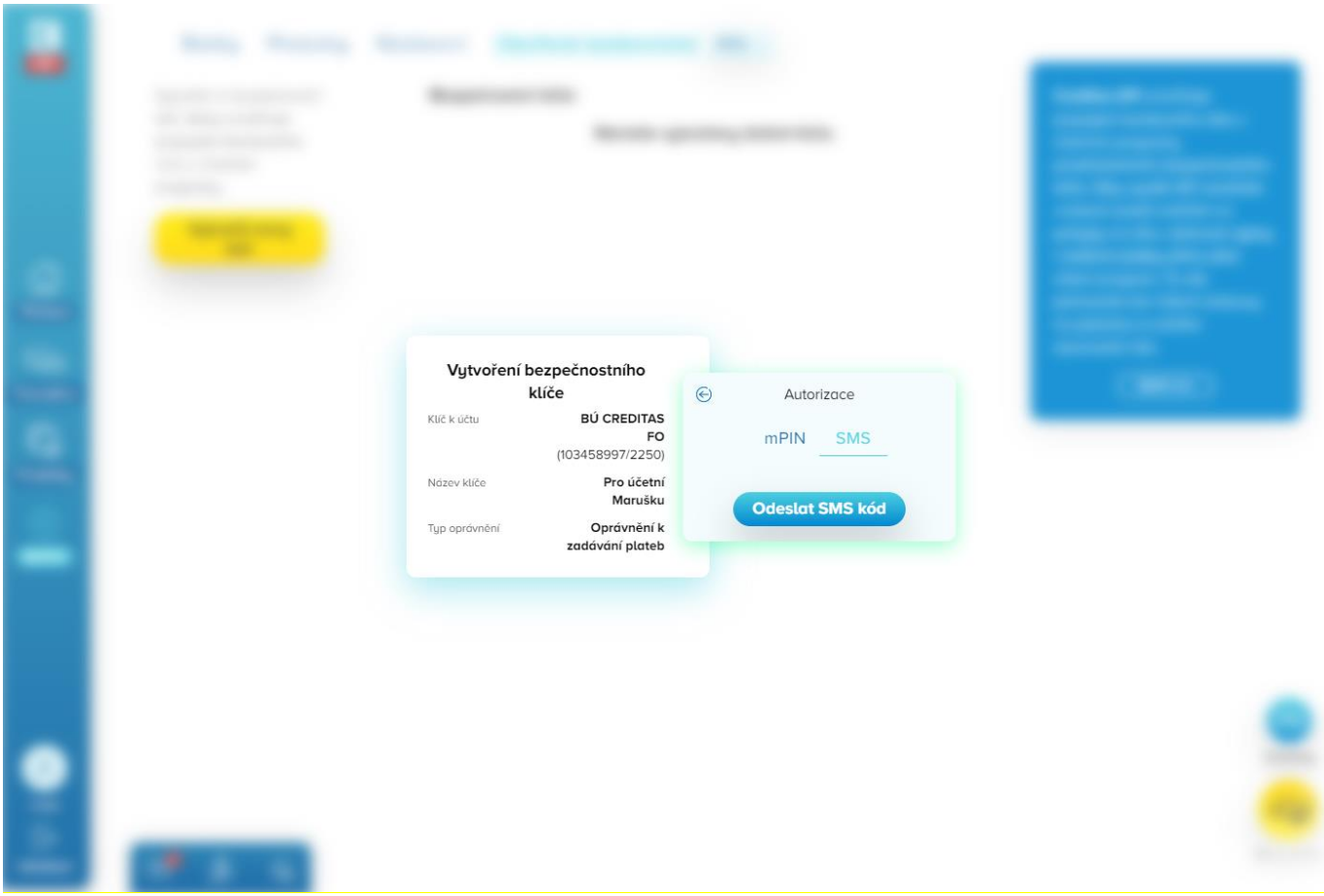
Nastavte prosím bezpečnostní klíč. Zvolte účet, ke kterému bude vytvořený, název a typ oprávnění.



5. Potvrďte podmínky a autorizujte vytvoření klíče

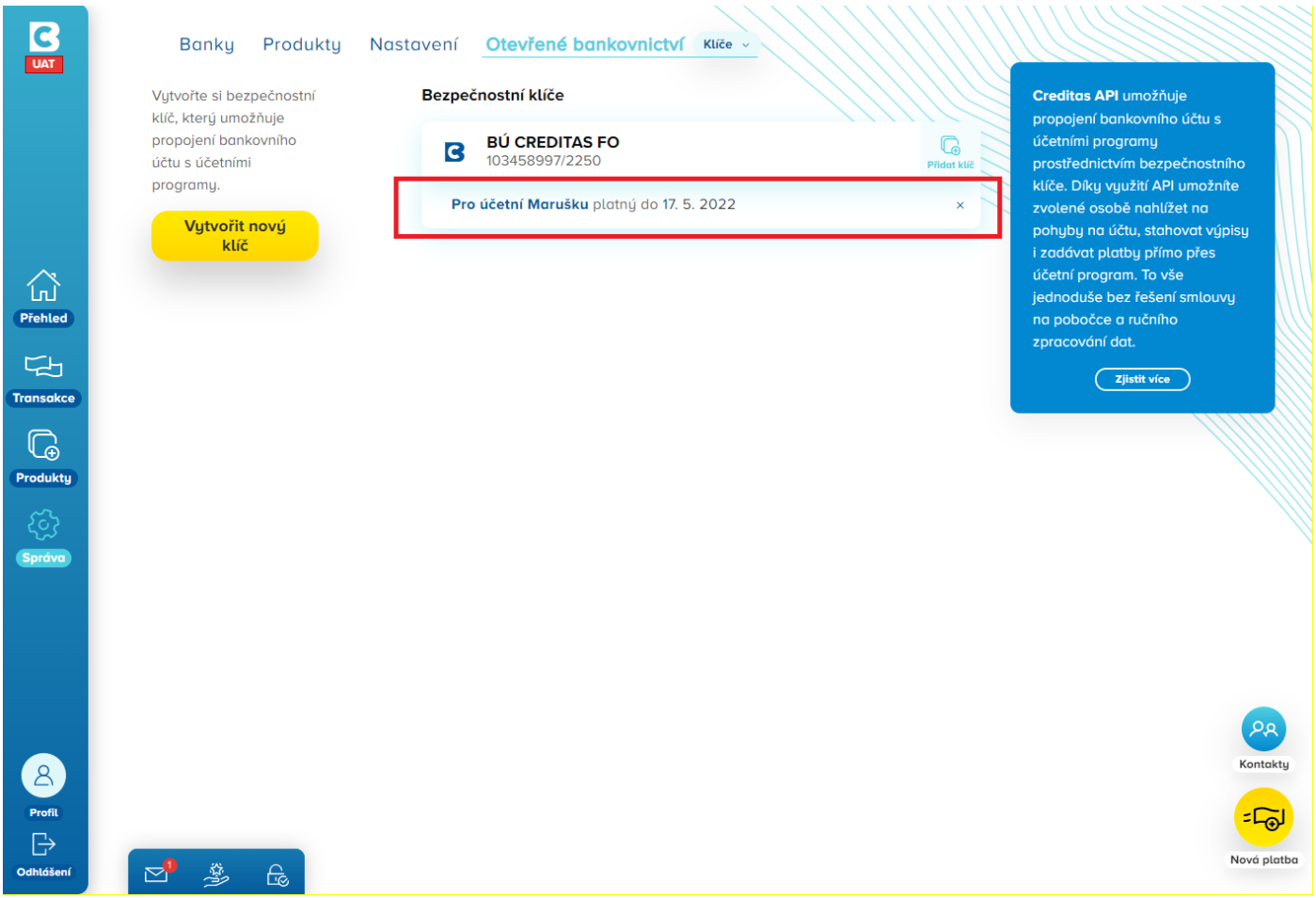
Potvrďte prosím podmínky pro Creditas API a autorizujte vytvoření nového klíče.


Potvrzení podmínek
Seznamte se prosím s podmínkami Creditas API.
Obchodní podmínky ke Creditas API
Jakmile zvolíte 'Potvrdit a pokračovat', potvrzujete, že jste se seznámili s obsahem dokumentů a souhlasíte s ním.
Zpět
Potvrdit a pokračovat


Vytvoření bezpečnostního klíče
Klíč k účtu
BÚ CREDITAS FO (103458997/2250)
Název klíče
Pro účetní Marušku
Typ oprávnění
Oprávnění k zadávání plateb
Autorizace
mPIN
SMS
Odeslat SMS kód

6. Hotový bezpečnostní klíč je připraven

Po úspěšné autorizaci uvidíte v seznamu bezpečnostních klíčů nový klíč.

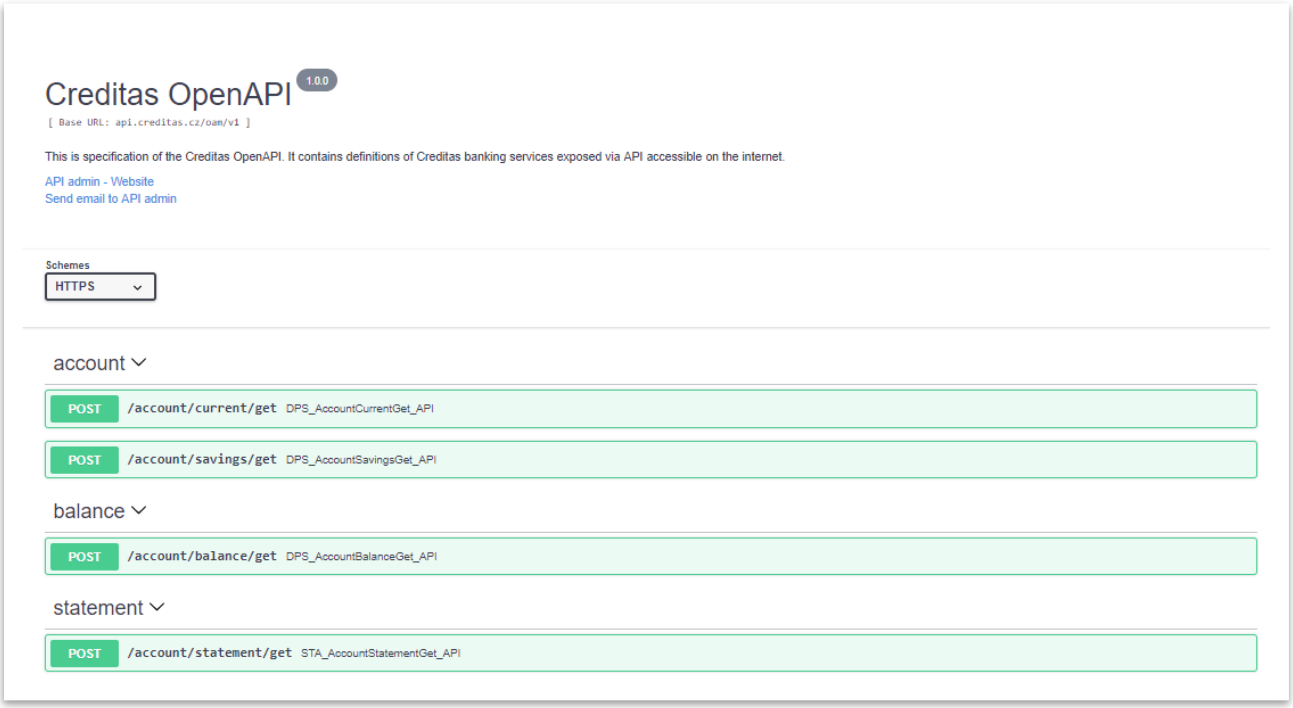


Zrušení bezpečnostního klíče v IB:
Zrušení vygenerovaného klíče provedete obdobným postupem, jakým jste klíč vygenerovali. Po přihlášení do internetového bankovnictví a přechodu do záložky Správa > Otevřené bankovnictví > Klíče je u každého vygenerovaného klíče zobrazena ikona křížku. Volbou tohoto tlačítka deaktivujete funkčnost spojenou s klíčem a odstraníte jej ze seznamu klíčů k účtu.

3 SLUŽBY CREDITAS API

3.1 Specifikace služeb

Technický popis jednotlivých API vystavených v rámci Creditas API je dostupný online na portálu SwaggerHub [link], který kromě uchování specifikace spolupracuje s řadou vývojových platforem a umožnuje do nich vygenerovat strukturu API. Jsou zde umístěny i ukázkové příklady volání a při vyplnění korektních údajů je možné i API provolat a rychle takto služby vyzkoušet.



3.2 Identifikátor účtu

Jako jednoznačnou referencí konkrétního účtu přes API je nutné vždy použít systémový identifikátor účtu (accountId). V případě ručně generovaného bezpečnostního klíče je tento identifikátor možné získat na seznamu aktivních klíčů v sekci „Nastavení Creditas API“, a to v detailu konkrétního účtu.

3.3 Oprávnění ke službám – ručně generovaný bezpečnostní klíč

Bezpečnostní klíč je svázán s daným uživatelem, který udělil přístup v procesu ručně generovaného klíče v internetovém bankovnictví. Uživatelem se myslí klient banky, fyzická osoba s aktivovaným přístupem do internetového bankovnictví. Při volání jednotlivých služeb je proto dostupná funkčnost v rozsahu platných dispozičních oprávnění dané osoby. Při udělování přístupu k API má uživatel možnost dále omezit rozsah oprávnění pro daný token. K dispozici jsou tyto dva typy oprávnění:

- Oprávnění ke čtení (automatické)
- Oprávnění k zadávání plateb (rozšířené)

Oprávnění ke čtení

Adresa služby	Popis služby
/account/current/get	Informace o běžném účtu
/account/savings/get	Informace o spořícím účtu
/account/balance/get	Informace o zůstatku na účtu
/account/transaction/search	Vyhledávaní v transakcích účtu
/account/transaction/export	Export transakcí
/account/statement/list	Seznam dostupných výpisů k účtu
/account/statement/get	Stažení výpisu

Oprávnění k zadávání plateb

Adresa služby	Popis služby
/payment/import	Hromadný import platebních příkazů
/payment/import/status/get	Informace o stavu importu platebních příkazů
/payment/domestic/create	Vytvoření platebního příkazu pro domácí platbu

3.4 Oprávnění ke službám – Bezpečnostní klíč vytvořený přes OAuth

Aplikace třetí strany si vždy vyžádá potřebný rozsah oprávnění a klient při udělení souhlasu může ještě udělenou sadu oprávnění omezit. Při udělení souhlasu je možné aplikaci přidělit tato oprávnění:

- Získávat informace o účtech (product_info)
- Získávat informace o zůstatku (balance_info)
- Přistupovat do historie transakcí (transaction_info)
- Umožnit realizaci platby (payment)

Oprávnění se aplikují na všechny účty, které klient pro aplikaci v rámci souhlasu zpřístupnil. Každé oprávnění dává možnost, přistoupit ke konkrétní sadě informaci nebo funkčnosti. S tato sadou je většinou svázána i množina služeb, které tuto oblast obsluhují.

Získávat informace o účtech

Adresa PSD2 služby	Popis služby
/aisp/account/list	Informace o běžném účtu

Adresa premium služby	Popis služby
/account/list	Seznam účtů
/account/current/get	Informace o běžném účtu
/account/savings/get	Informace o spořicím účtu
/account/termdeposit/get	Informace o termínovaném vkladu

Získávat informace o zůstatku

Adresa PSD2 služby	Popis služby
/aisp/account/balance/get	Informace o zůstatku na účtu
/cisp/account/balance/check	Kontrola zůstatku na účtu - CISP
/pisp/account/balance/check	Kontrola zůstatku na účtu - PISP

Adresa premium služby	Popis služby
/account/balance/get	Informace o zůstatku na účtu

Přistupovat do historie transakcí

Adresa PSD2 služby	Popis služby
/aisp/account/transaction/list	Vyhledávaní v transakcích účtu

Adresa premium služby	Popis služby
/account/transaction/search	Vyhledávaní v transakcích účtu
/account/transaction/export	Export transakcí
/account/statement/list	Seznam dostupných výpisů k účtu
/account/statement/get	Stažení výpisu

Umožnit realizaci platby

Adresa PSD2 služby	Popis služby
/pisp/payment/domestic/create	Vytvoření platebního příkazu pro domácí platbu
/pisp/payment/sepa/create	Vytvoření platebního příkazu pro SEPA platbu
/pisp/payment/foreign/create	Vytvoření platebního příkazu pro zahraniční platbu
/pisp/payment/status/get	Informace o stavu platebního příkazu

Adresa premium služby	Popis služby
/payment/import	Hromadný import platebních příkazů
/payment/import/status/get	Informace o stavu importu platebních příkazů
/payment/domestic/create	Vytvoření platebního příkazu pro domácí platbu

3.5 Import plateb

Hromadný import platebních příkazů podporuje stejné formáty, které umožnuje internetové bankovnictví Banky CREDITAS. Všechny tyto importní soubory jsou do API předávány v base64 kódování.

Formáty pro domácí příkazy

Formát	Popis
ABO (kpc)	Standardizovaný formát pro import domácích platebních příkazů. Podrobný popis formátu naleznete na [link].
CSV	Bankovní formát pro import domácích platebních příkazů. Podrobný popis formátu naleznete na [link].

Formáty pro zahraniční příkazy

Formát	Popis
XML (pain.001.001.03)	Formát pro import zahraničních platebních příkazů založený na normě ISO 20022 (pain.001.001.03). Podrobný popis formátu naleznete na [link].
MT101	Formát pro import zahraničních platebních příkazů založený na MT101 zprávě definované SWIFT asociací. Podrobný popis formátu naleznete na [link].

3.6 Export transakční historie a výpisy z účtu

Formáty pro export transakční historie

Formát	Popis
PDF	Standardizovaný formát vhodný pro tisk.
XML (camt.053.001.02)	Formát založený na standardu ČBA pro XML výpisy vycházející z normy ISO 20022 (camt.053.001.02). Podrobný popis formátu od ČBA naleznete na [link].
CSV	Bankovní formát pro export transakční historie. Podrobný popis formátu naleznete na [link].

Formáty výpisů z účtu

Formát	Popis
PDF	Standardizovaný formát vhodný pro tisk.
ABO (gpc)	Standardizovaný formát pro import domácích platebních příkazů. Podrobný popis formátu naleznete na [link].

4 OŠETŘENÍ CHYB (ERROR HANDLING)

Volání Open API může mít za následek tři typy aplikačních chyb:

- Deklarované chyby – Jsou specifikované rozhraním konkrétní API operace, jejich popis je vždy uveden v dokumentaci dané API operace.
- Neočekávané chyby – Pokud nastane chyba, která není deklarovaná, server vrátí neočekávanou API výjimku.
- Bezpečnostní chyby – V případě, že request neobsahuje validní klíč pro přístup k dané API, tak server vrátí tuto chybu.
- Chyby autorizace transakce – Chyby indikující problém v procesu autentizace uživatele vyžádané při konkrétním volání API.

4.1 Deklarované chyby

Server vrátí http kód 500 a v těle odpovědi je error struktura, kde “name“ definuje typ deklarované chyby. Response type je application/json. Specifický typ deklarované chyby je validační chyba - SYS_ValidationExc, která v error struktuře obsahuje element data obsahující kolekci validationResult. Každý záznam kolekce obsahuje validationErrorCode a validationErrorMessage.


```
{
    "name": "SYS_ValidationExc",
    "message": "validation exception with the following errors (1):\n Payment order can't be p
    "data": {
        "validationResult": {
            "validationErrorCode": "VAERR2006",
            "validationErrorMessage": "Payment order can't be processed with requested due dat
        }
    }
}
```


4.2 Neočekávané chyby

Server vrátí http kód 500 a v těle odpovědi je error struktura, kde „name“ má hodnotu SYS_UnexpectedExc. Response type je application/json.

```
{
    "name": "SYS_UnexpectedExc",
    "message": "Unexpected exception occurred"
}
```

4.3 Bezpečnostní chyby

Server vrátí http kód 500 a v těle odpovědi je error struktura, kde „name“ má hodnotu OAM_SecurityExc. Response type je application/json.

```
{
    "name": "OAM_SecurityExc",
    "message": "Access denied"
}
```

4.4 Chyby autorizace transakce

Server vrátí http kód 500 a v těle odpovědi je error struktura, kde „name“ má hodnotu OAM_TransactionAuthorizationExc. Response type je application/json.

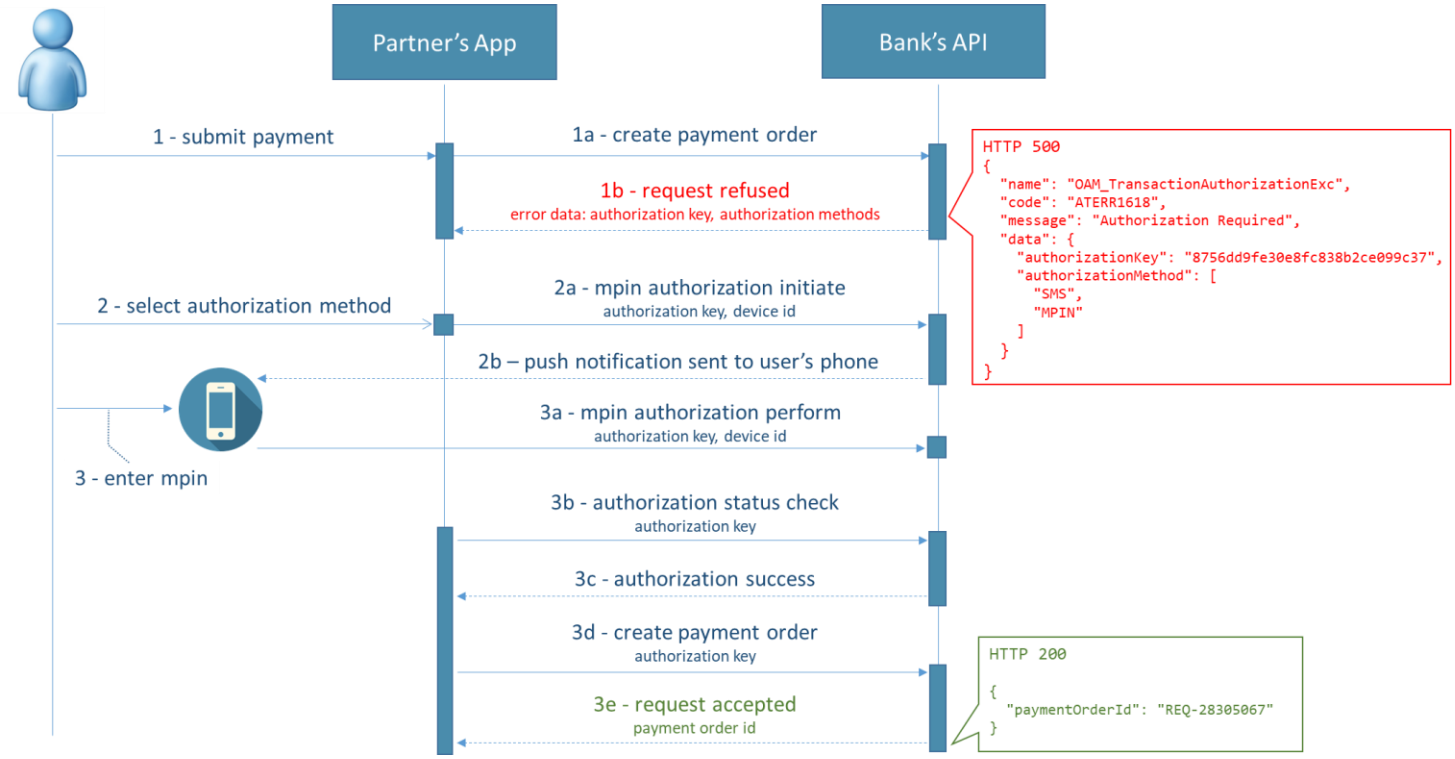
```
{
    "name": "OAM_TransactionAuthorizationExc",
    "code": "ATERR1618",
    "message": "Authorization Required",
    "data": {
        "authorizationKey": "8756dd9fe30e8fc838b2ce099c37db35b277b8be6a993844a4f480580d325b8f",
        "authorizationMethod": [
            "SMS",
            "MPIN"
        ]
    }
}
```

5 AUTORIZACE TRANSAKCÍ

Každý dotaz na API vyžaduje v http hlavičce platný access token. Access token je bezpečnostní klíč vygenerovaný uživatelem v IB nebo je vytvořen automaticky pomocí OAuth 2 protokolu. Oba bezpečnostní modely na získání bezpečnostního klíče jsou popsány v kapitole 2. Pro ověření, že bezpečnostní klíč třetí strana nezneužije, musí API v určitých případech ověřit, že dané volání je výsledkem uživatelské akce a vyžádá si autorizaci volání pomocí jednoho z bezpečnostních prvků, které používá v IB jako je Mobilní PIN nebo SMS OTP. V současné verzi je toto vyžadováno pro platební transakce:

- /pisp/payment/domestic/create
- /pisp/payment/foreign/create
- /pisp/payment/sepa/create

Princip autorizace transakce je, že partnerská aplikace pošle dotaz na zadání platby. Server buď dotaz akceptuje a platební příkaz zpracuje anebo požadavek zamítne a v odpovědi vrátí chybu, která obsahuje autorizační klíč, který je potřebné použít v procesu autentizace uživatele vybraným bezpečnostním prvkem. Po úspěšném ověření pak partnerská aplikace opakuje původní dotaz na zadání platby a v hlavičce pošle autorizační klíč. Server ověří platnost autorizačního klíče, zkontroluje, že vstupní data pro zadání platby se shodují s původním dotazem a následně platební příkaz zpracuje. Následující schéma popisuje proces autorizace platby za pomocí bezpečnostního prvku MPIN.

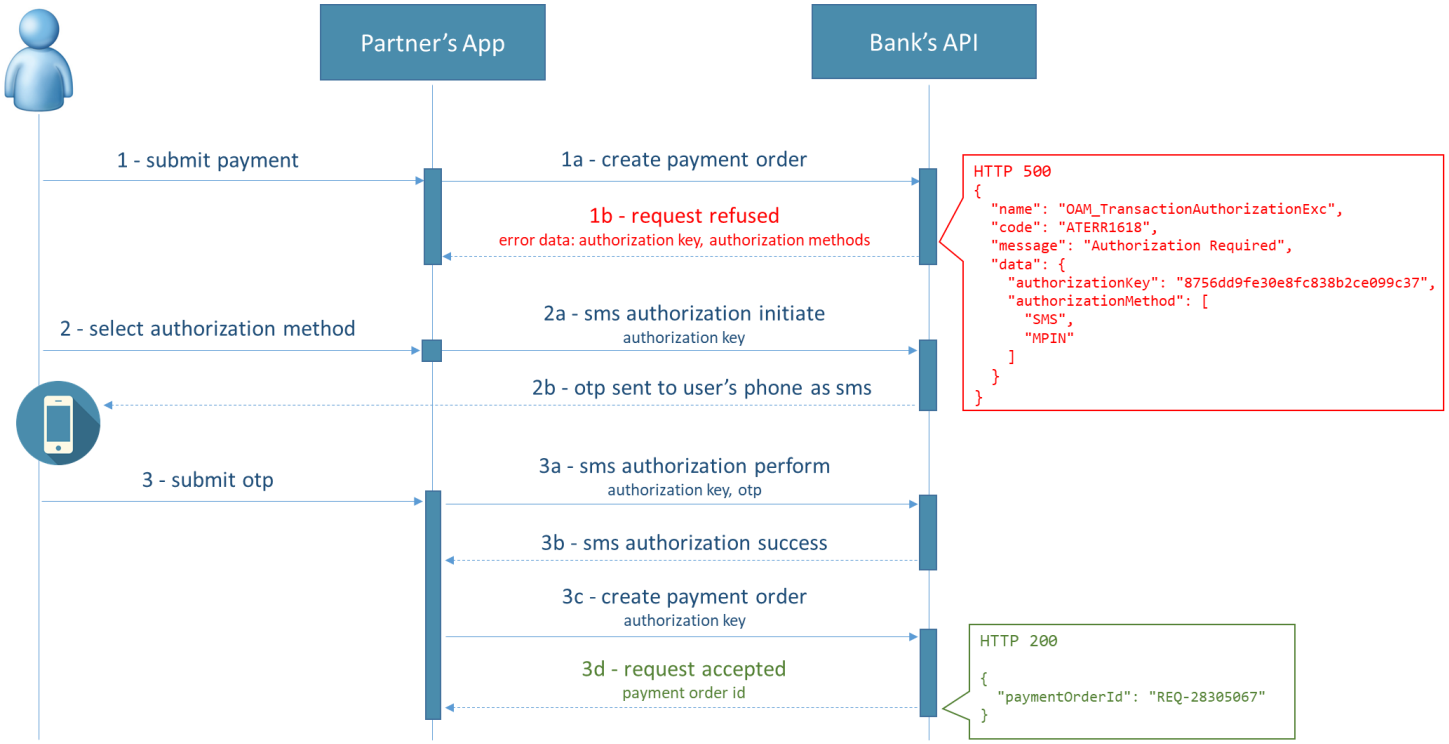


1. Uživatel potvrdí zadání platby v partnerské aplikaci
    a. Partnerská aplikace pošle dotaz na vytvoření platebního příkazu na bankovní API
    b. API pošle odpověď s http status kódem 500 a v těle odpovědi obsahuje chybovou strukturu s názvem chyby *OAM_TransactionAuthorizationExc* a chybovým kódem *ATERR1618*, v sekci data je pak uveden autorizační klíč a dostupné autorizační metody.
2. Uživatel vybere autorizační metodu MPIN
    a. Partnerská aplikace zavolá na inicializaci MPIN autorizace */authorization/mpinesign/initiate*. V http hlavičce *Authorization-Key* pošle hodnotu autorizačního klíče a v těle zprávy identifikátor zařízení, na kterém chce uživatel autorizovat transakci. Seznam dostupných zařízení poskytuje operace */authorization/mpinesign/device/list*.
    b. Server pošle požadavek na autorizaci formou push notifikace na vybrané zařízení.
3. Uživatel v mobilní aplikaci Creditas autorizuje požadavek zadáním MPINu.
    a. Mobilní aplikace ověří MPIN a pošle dotaz na sever pro vyhodnocení autorizace.
    b. Partnerská aplikace nic netuší o komunikaci mezi mobilní aplikací a bankovním API a na výsledek autorizace čeká tak, že periodicky ověřuje stav voláním operace */authorization/status/get* (v budoucí verzi API bude možné použít callback).
    c. Pokud autorizace byla úspěšně dokončena, server vrátí stav *COMPLETE*.
    d. Partnerská aplikace opakuje dotaz na zadání platby, ale v hlavičce již uvede autorizační klíč.
    e. Server ověří platnost autorizace, požadavek na platební příkaz zpracuje a vrátí v odpovědi číslo platebního příkazu.

Důležité: Autorizační klíč funguje nezávisle na bezpečnostním klíči. **Bezpečnostní klíč je nutné v dotazech uvádět vždy**, protože se jedná o OAuth access token, který reprezentuje souhlas uživatele s přístupem k API a v http hlavičce se uvádí ve tvaru *Authorization:Bearer 1604944855050* nebo *Authorization-Bearer: 1604944855050* (druhý uvedený tvar přestane být časem podporován). Autorizační klíč se používá jenom v případě, že konkrétní volání si vyžádá dodatečnou autorizaci pro ověření identity uživatele. Autorizační klíč se uvádí v http hlavičce ve tvaru *Authorization-Key: 98904565959*

Následující schéma zobrazuje proces použití autorizační metody SMS OTP. Funguje obdobně jako MPIN, ale používá specifické operace pro inicializaci bezpečnostního prvku aj jeho ověření

- */authorization/smsotp/initiate* – vygeneruje a pošle jednorázové heslo na bezpečnostní telefonní číslo uživatele
- */authorization/smsotp/perform* – ověří jednorázové heslo a vyhodnotí autorizaci



V případě SMS OTP není nutné periodicky ověřovat výsledek autorizace, protože jednorázové heslo zadá uživatel přímo do partnerské aplikace a ta zavolá operaci */authorization/smsotp/perform*, která synchronně vrátí výsledek autorizace. SMS OTP je momentálně v experimentální modu a není možné ho použít pro produkční řešení.

6 SEZNAM API

Operation	Operation name (Swagger)	Operation Kind	Access type	PSD2 Scope	Authorization Code Grant Allowed	Implicit Grant Allowed	Client Credentials Grant Allowed	Transferable Grant Allowed
PAY_PispPaymentStatusGet_API	Get payment status	BUSINESS	PSD2	PISP	Y	Y	N	N
PAY_PispPaymentSepaCreate_API	Create SEPA payment order	BUSINESS	PSD2	PISP	Y	Y	N	N
PAY_PispPaymentForeignCreate_API	Create foreign payment order	BUSINESS	PSD2	PISP	Y	Y	N	N
PAY_PispPaymentDomesticCreate_API	Create domestic payment order	BUSINESS	PSD2	PISP	Y	Y	N	N
OAM_ClientRegistrationCreate_API	Create client's application registration	OTHER	PSD2		N	N	N	N

DPS_PispAccountBalanceCheck_API	Check balance for amount	BUSINESS	PSD2	PISP	Y	Y	N	N
DPS_CispAccountBalanceCheck_API	Check balance for amount	BUSINESS	PSD2	CISP	Y	Y	Y	N
DPS_AispAccountTransactionList_API	Get list of account transactions	BUSINESS	PSD2	AISP	Y	Y	N	N
DPS_AispAccountList_API	Get list of accounts	BUSINESS	PSD2	AISP	Y	Y	N	N
DPS_AispAccountBalanceGet_API	Get account balance	BUSINESS	PSD2	AISP	Y	Y	N	N
AUT_SmsOtpPerform_API	Perform SMS authorization	AUTHORIZATION	PSD2		Y	Y	N	N
AUT_SmsOtpInitiate_API	Initiate SMS authorization	AUTHORIZATION	PSD2		Y	Y	N	N
AUT_RedirectIbInitiate_API	Initiate redirect authorization	AUTHORIZATION	PSD2		Y	Y	N	N
AUT_MpinEsignInitiate_API	Initiate MpinEsign authorization	AUTHORIZATION	PSD2		Y	Y	N	N
AUT_MpinEsignDeviceList_API	Get device list for MpinEsign	AUTHORIZATION	PSD2		Y	Y	N	N
AUT_AuthorizationStatusGet_API	Get authorization status	AUTHORIZATION	PSD2		Y	Y	N	N
STA_AccountStatementList_API	Get list of available account satements	BUSINESS	Premium		Y	Y	N	Y
STA_AccountStatementGet_API	Download account statement	BUSINESS	Premium		Y	Y	N	Y
PAY_PaymentStatusGet_API	Get payment import status	BUSINESS	Premium		Y	Y	N	Y
PAY_PaymentSepaCreate_API	Create SEPA payment order	BUSINESS	Premium		Y	Y	N	N
PAY_PaymentSearch_API	Search payments	BUSINESS	Premium		Y	Y	N	N
PAY_PaymentImportStatusGet_API	Get payment import status	BUSINESS	Premium		Y	Y	N	Y

PAY_PaymentImport_API	Import payment order(s)	BUSINESS	Premium	Y	Y	N	Y
PAY_PaymentForeignCreate_API	Create foreign payment order	BUSINESS	Premium	Y	Y	N	N
PAY_PaymentDomesticCreate_API	Create domestic payment order	BUSINESS	Premium	Y	Y	N	Y
LNS_LoanList_API	Get loan list	BUSINESS	Premium	Y	Y	N	N
LNS_LoanGet_API	Get loan	BUSINESS	Premium	Y	Y	N	N
DPS_AccountTransactionSearch_API	Search account transactions	BUSINESS	Premium	Y	Y	N	Y
DPS_AccountTransactionExport_API	Export account transactions	BUSINESS	Premium	Y	Y	N	Y
DPS_AccountTermDepositGet_API	Get term deposit account	BUSINESS	Premium	Y	Y	N	N
DPS_AccountSavingsGet_API	Get savings account	BUSINESS	Premium	Y	Y	N	Y
DPS_AccountList_API	Get account list	BUSINESS	Premium	Y	Y	N	N
DPS_AccountCurrentGet_API	Get current account	BUSINESS	Premium	Y	Y	N	Y
DPS_AccountBalanceGet_API	Get account balance	BUSINESS	Premium	Y	Y	N	Y
CRD_CardList_API	Get card list	BUSINESS	Premium	Y	Y	N	N
CRD_CardDebitGet_API	Get debit card	BUSINESS	Premium	Y	Y	N	N