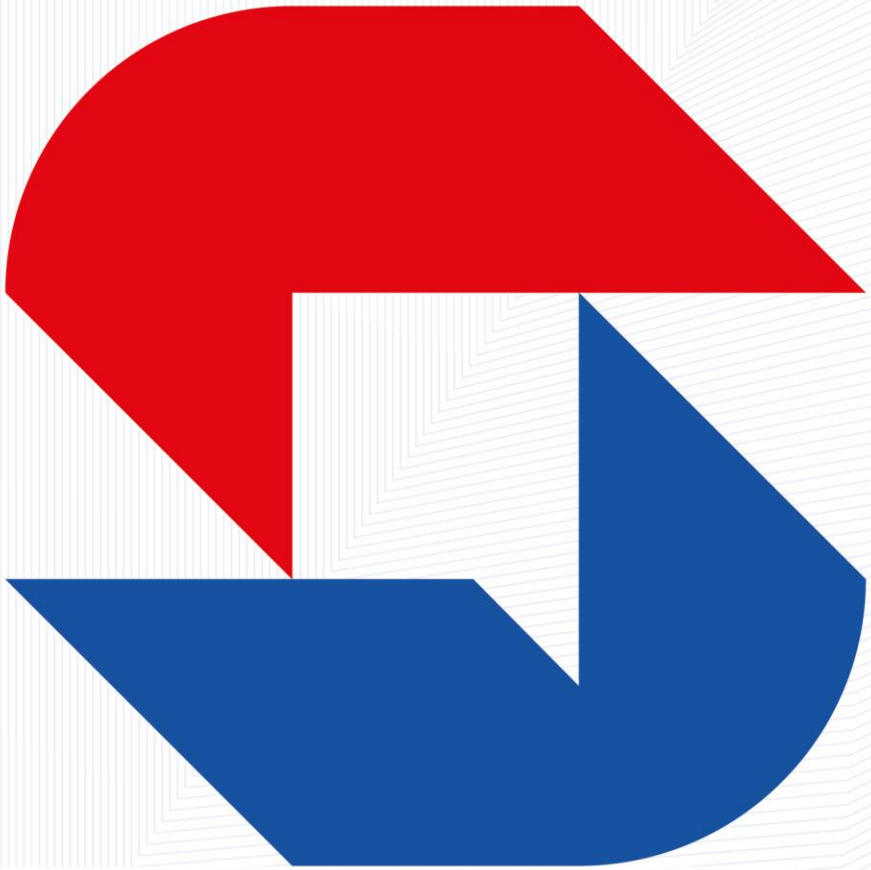
SPRÁVA STÁTNÍCH SLUŽEB VYTVÁŘEJÍCÍCH DŮVĚRU



Certifikační autorita EET v 2.0

Postupy získání pokladního certifikátu

Obsah

Úvod

Dokument popisuje postup pro získání pokladního certifikátu sloužícího k autentizaci datových zpráv s údaji o evidované tržbě zasílaných do systému EET 2.0 a také postup jeho obnovy.

Postup získání pokladního certifikátu je pro všechny pokladní systémy stejný. Pokladní certifikát získá poplatník na *Obslužném portálu EET v aplikaci Správa pokladních certifikátů EET*. Do *Obslužného portálu EET* lze vstoupit pouze z *portálu MOJE daně* resp. přihlašovací stránky aplikace Daňová informační schránka plus (DIS+) https://mojedane.gov.cz/pmd/home/prihlaseni-do-dis.

Obnovu pokladního certifikátu řeší pokladní systém automatizovaně, pokud má implementovanou podporu pro automatickou obnovu. V opačném případě se obnova řeší ručním vydáním nového pokladního certifikátu v aplikaci Správa pokladních certifikátů EET.

Postup vytvoření žádosti o pokladní certifikát

Vytvoření a zaslání žádosti o pokladní certifikát je jednotné pro všechny pokladní systémy bez ohledu na použitý operační systém a řešení pokladního systému.

O vydání prvního pokladního certifikátu požádá poplatník v aplikaci Správa pokladních certifikátů EET v prostředí prohlížeče. Do prohlížeče ani na zařízení (PC, smart zařízení), kde prohlížeč běží, není potřeba pro vytvoření žádosti o pokladní certifikát instalovat žádné dodatečné komponenty.

V rámci vytváření žádosti o pokladní certifikát poplatník vyplní v aplikaci Správa pokladních certifikátů EET nepovinnou položku „Označení certifikátu“ a nepovinnou položku „e-mail” pro zasílání upozornění týkající se pokladního certifikátu. Tlačítkem „Odeslat žádost“ požádá o vydání certifikátu.

Pokladní certifikát včetně páru RSA klíčů se generuje bezpečným způsobem v systému *CA EET*, tedy zcela mimo pokladní systém a prohlížeč.

Vydaný pokladní certifikát včetně soukromého klíče si poplatník stáhne v zabezpečené podobě z *aplikace* Správa pokladních certifikátů EET ve formátu PKCS#12 jako soubor s příponou .p12. Soubor .p12 je určený pro import do pokladního zařízení. Soubor .p12 obsahuje i certifikáty z certifikační cesty tj. kořenový certifikát CA a mezilehlý certifikát.

Soubor .p12 je proti zneužití chráněn heslem, které poplatník získá taktéž v aplikaci Správa pokladních certifikátů EET.

Soubor .p12 bude na *v aplikaci Správa pokladních certifikátů EET* nabízen ke stažení, dokud poplatník nepotvrdí jeho stažení, nejdéle však 30 dní od vydání pokladního certifikátu, Po stažení nebo uplynutí lhůty na stažení bude soubor .p12 v systému zrušen a nebude dále k dispozici.

Postup vytvoření žádosti o obnovu pokladního certifikátu

Pokladní certifikát může poplatník obnovit dvěma způsoby:

- Ručně v aplikaci Správa pokladních certifikátů EET vydáním nového pokladního certifikátu viz předchozí kapitola.
  nebo
- Automatizovaně z pokladního systému.

Požadavky na pokladní systém

Automatizovanou obnovu pokladního certifikátu realizuje pokladní systém, který má za tímto účelem implementované volání rozhraní *CA EET* pro automatizovanou obnovu.

Automatizovaná obnova pokladního certifikátu musí proběhnout vždy před koncem platnosti obnovovaného certifikátu. Doporučujeme obnovu realizovat 2-3 týdny před koncem platnosti pokladního certifikátu.

Žádost o obnovu pokladního certifikátu musí být podepsaná obnovovaným pokladním certifikátem.

Pro automatizovanou obnovu musí tedy pokladní systém:

- implementovat API *CA EET* popsané v **caeetapi_jwt.yml,**
- znát datum konce platnosti obnovovaného pokladního certifikátu,
- importovat .p12 soubor do úložiště klíčů, které využívá aplikace pro evidenci tržeb.

URL pro přístup k API jsou součástí definice API.

Autentizace k rozhraní

Přístup k rozhraní caeetapi je autentizován s použitím JWT tokenu, který se předává v HTTP hlavičce Authorization: Bearer <token>.

Formát tokenu je <base64url(protected-header)>.<base64url(payload)>.<base64url(signature)>.

Požadavky na JOSE protected header:

- alg: RS256
- typ: JWT (doporučeno)
- x5t#S256: base64url(SHA-256(DER))

Požadavky na payload:

- JWT payload musí být JSON objekt
- povinné claims: exp, iat
- exp a iat musí být typu NumericDate
- exp musí být větší než iat
- server odmítá tokeny, u kterých je exp nastaveno na více než 5 minut od iat
- minimální podporovaný payload je objekt obsahující exp a iat

Poznámka:

- Header x5c se nebude používat; klient neposílá celý certifikát v tokenu.
- Certifikát se identifikuje výhradně přes x5t#S256 a podle této hodnoty se vyhledává v databázi.

Endpointy pro obnovu pokladního certifikátu

- POST /request/renew – Podání žádosti o obnovu pokladního certifikátu

  Endpoint je určen pro automatizované nebo integrační volání autorizované pomocí JWT tokenu v hlavičce Authorization. Token musí být podepsán obnovovaným certifikátem a musí splňovat profil popsaný ve .**bearerJwtRs256.**

  Obnovovaný certifikát musí být v okamžiku volání platný. Při úspěšném zpracování endpoint vrací identifikátor založené žádosti .**reqId.**

  Stav žádosti je následně možné zjišťovat přes ./**request/{reqId}/status.**

- GET /request/{reqId}/status – Získání aktuálního stavu žádosti o pokladní certifikát

  Endpoint vrací stav žádosti identifikované parametrem **reqId**. Odpověď používá schéma **RequestStatusDTO**, které rozlišuje jednotlivé stavy **INPROCESS, ISSUED, DELIVERING, FINISHED a REJECTED.**

  Pokud je žádost ve stavu **INPROCESS**, odpověď obsahuje doporučený interval pro další polling v poli **pollAfterSeconds** a současně i v hlavičce **Retry-After.**

  Endpoint je dostupný pro integrační volání pomocí JWT. Podpisový certifikát použitý pro JWT musí být platný.

- POST /request/{reqId}/claim-download – Stažení PKCS#12 a hesla

  Endpoint je možné volat pouze pro žádost ve stavu **ISSUED** nebo **DELIVERING**. Stav **ISSUED** znamená, že certifikát je vydán a připraven k prvnímu stažení. Stav **DELIVERING** znamená, že PKCS#12 a heslo již byly zpřístupněny ke stažení, ale převzetí dosud nebylo potvrzeno. V tomto stavu je možné data znovu získat až do potvrzení přes **/request/{reqId}/ack-download.**

  Při úspěšném volání vrací endpoint objekt **Pkcs12DTO** obsahující PKCS#12 v base64, heslo pro jeho otevření a metadata vydaného certifikátu.

  Pokud nedojde ke stažení PKCS#12 do 30 dnů od vydání certifikátu server odstraní dočasně držená data pro stažení a žádost přejde do ukončeného stavu workflow. Po uplynutí této lhůty není možné PKCS#12 a heslo znovu získat přes **/request/{reqId}/claim-download.**

  Endpoint je dostupný pro integrační volání pomocí JWT.

  Vrácené údaje jsou citlivé, volající aplikace by je neměla ukládat do cache ani logů. Po úspěšném převzetí má volající aplikace potvrdit dokončení přes **/request/{reqId}/ack-download.**

- POST /request/{reqId}/ack-download - Potvrzení stažení PKCS#12

  Endpoint je možné volat pouze pro žádost ve stavu **DELIVERING** . Stav **DELIVERING** znamená, že PKCS#12 a heslo již byly klientovi zpřístupněny ke stažení, ale převzetí dosud nebylo potvrzeno.

  Po úspěšném potvrzení server odstraní dočasně držená data pro stažení a žádost přejde do ukončeného stavu workflow. Po tomto potvrzení již není možné PKCS#12 a heslo znovu získat přes **/request/{reqId}/claim-download.**

  Pokud nedojde k potvrzení stažení PKCS#12 do 30 dnů od vydání certifikátu, server odstraní dočasně držená data pro stažení a žádost přejde do ukončeného stavu workflow. Po uplynutí této lhůty není možné PKCS#12 a heslo znovu získat přes **/request/{reqId}/claim-download.**

  Endpoint je dostupný pro integrační volání pomocí JWT.

- GET /request/not-finished – Získání všech neukončených žádostí pro EIČ

  Endpoint vrací seznam identifikátorů žádostí **reqId**, které jsou ve stavech **INPROCESS**, **ISSUED** nebo **DELIVERING.**

  Endpoint je dostupný pro integrační volání pomocí JWT. Podpisový certifikát použitý pro JWT musí být platný.

  Detail stavu jednotlivých žádostí je možné následně získat přes **/request/{reqId}/status.**

Příklad sekvence volání

Endpointy je nutné volat v následujícím pořadí:

- **POST /request/renew**
- Periodicky volat **GET /request/{reqId}/status**.
  - Volání opakovat dle hodnoty v pollAfterSeconds v těle odpovědi nebo dle HTTP hlavičky Retry-After. Volání opakovat, dokud je stav žádosti INPROCCESS. Při změně stavu žádosti na stav ISSUED pokračovat voláním POST z další odrážky.
- **POST /request/{reqId}/claim-download**
- Po uložení PCSK#12 a hesla volat **POST /request/{reqId}/ack-download.**

Ochrana soukromého klíče

Soukromý klíč musí být chráněn proti zcizení a zneužití, neboť právě soukromý klíč slouží k vytváření elektronických podpisů. Ochrana soukromého klíče a certifikátu pro evidenci tržeb před jeho zneužitím je zákonnou povinností poplatníka.

Poplatník si tedy musí chránit soukromý klíč k pokladnímu certifikátu. Musí si bezpečně uložit soubor .p12. Musí si bezpečně zaznamenat heslo k otevření souboru .p12.

Soubor .p12 obsahuje pokladní certifikát a k němu příslušný soukromý klíč a certifikáty z certifikační cesty. Soubor .p12 je zabezpečen heslem. Heslo má délku 8 znaků, obsahuje velká a malá písmena a číslice.

Kvůli zpětné kompatibilitě se staršími pokladními systémy je soubor .p12 generován v legacy formátu, kde je soukromý klíč chráněn šifrováním pomocí algoritmu 3DES-CBC.

Upozornění:

U implementací běžících nad OpenSSL 3 je očekávané, že standardní načtení výše popsaného .p12 souboru může selhat, pokud není dostupný OpenSSL legacy provider.

Doporučené postupy jsou následující:

- Pokud aplikace pracuje přímo se staženým .p12 souborem, je možné pro daný proces povolit OpenSSL legacy provider. Doporučujeme jej povolit cíleně pouze pro prostředí nebo proces, který tento legacy PKCS#12 formát skutečně potřebuje, a ponechat zároveň aktivní i default provider.
- Z dlouhodobého provozního hlediska je vhodnější po stažení certifikát bezpečně importovat nebo převést do formátu používaného konkrétní aplikací, například PEM nebo nově zabalený PKCS#12 s modernější ochranou. Tento převod musí proběhnout v důvěryhodném prostředí, s odpovídajícím zabezpečením privátního klíče. Převod nemění vlastní certifikát ani klíčový pár, mění se pouze lokální způsob uložení.
- Jako vhodnou variantu doporučujeme také uložení certifikátu a privátního klíče do zabezpečeného úložiště tajemství, například vaultu, key vaultu nebo obdobného enterprise secret management řešení. V takovém případě by aplikace neměla dlouhodobě pracovat se souborem .p12 uloženým na filesystému, ale měla by certifikát/klíč získávat řízeně z vaultu, s omezeným přístupem, auditováním a rotací přístupových oprávnění. Pokud použité řešení podporuje přímo podpisové operace nad privátním klíčem, je vhodné preferovat variantu, kdy privátní klíč vault neopouští. Pokud to aplikace nebo knihovna pro XML podpis nepodporuje, je potřeba zajistit alespoň bezpečné dočasné načtení klíče s minimálními oprávněními a bez ukládání nešifrované kopie na disk.

Z bezpečnostního pohledu nedoporučujeme privátní klíč předávat mimo systém provozovatele ani jej ponechávat v nešifrované podobě. Pokud je prováděn převod do PEM nebo nového PKCS#12, měl by být proveden pouze lokálně, s omezenými oprávněními k souborům a bezpečnou likvidací dočasných souborů. Preferovaným řešením je bezpečné uložení v odpovídajícím vaultu nebo secret management systému.

Seznam příloh

Soubor s přílohou	Obsah
caeetapi_jwt.yml	endpointy pro automatizovanou obnovu