

# OBELISK Trusted Archive - Integrační příručka

číslo verze 2.1.38

datum vytvoření 30.04.2024

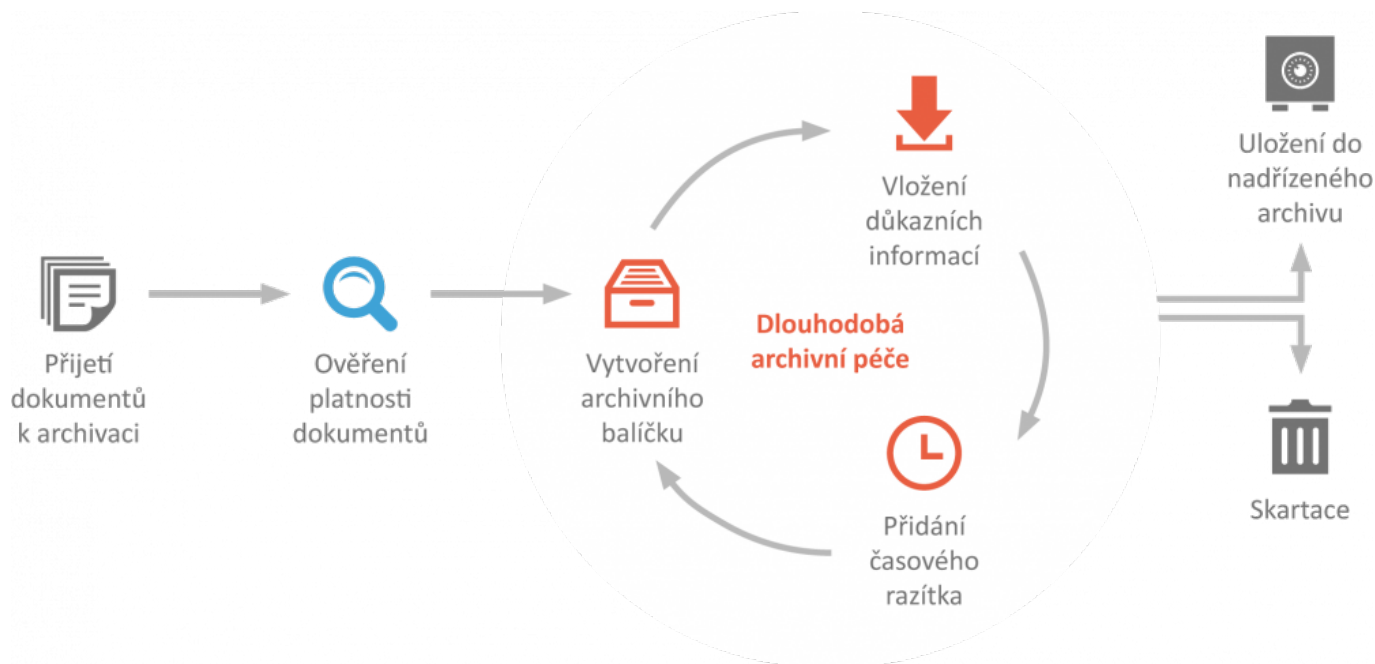
# Obsah

<b>Princip archivace</b>	1
<b>Úvod</b>	1
<b>Princip archivace</b>	2
<b>Přehled procesu zpracování OB-TA</b>	2
Přijetí dokumentu	2
Archivace dokumentu	2
Archivace souborů dokumentu	3
Vytváření a archivace balíčků	3
<b>Konfigurace</b>	5
<b>Kompozitní politiky příjmu obsahu</b>	5
<b>Politiky pro příjem do archivu a pro tvorbu balíčků</b>	5
<b>Webové služby - Úvod</b>	6
<b>Autentizace webové služby</b>	6
<b>Popis služby</b>	6
Parametry požadavku hlavičky volání	6
Parametry hlavičky odpovědi	7
<b>Webové služby - StoreDocument</b>	9
<b>StoreDocumentRequest</b>	9
Uložení nové verze dokumentu	10
<b>StoreDocumentResponse</b>	11
<b>Webové služby - Callback</b>	13
<b>Webové služby - UpdateMetadata</b>	14
<b>UpdateMetadataRequest</b>	14
Modifikace jednotlivých verzí dokumentu	14
<b>UpdateMetadataResponse</b>	14
<b>Webové služby - RetrieveDocument</b>	15
<b>RetrieveDocumentRequest</b>	15
<b>RetrieveDocumentResponse</b>	15
<b>Webové služby - RemoveDocument</b>	18
<b>RemoveDocumentRequest</b>	18
Mazání verzí dokumentu	18
<b>RemoveDocumentResponse</b>	18
<b>Webové služby - RepositoryFunction</b>	20
<b>RepositoryFunctionRequest</b>	20
<b>RepositoryFunctionResponse</b>	20
<b>Podporované funkce souborového úložiště</b>	20
<b>Důkazní materiál</b>	21
<b>Vyžádání přes GUI</b>	21
<b>Vyžádání requestem webové služby</b>	21
<b>Přílohy ke stažení - Webové služby</b>	23

# Princip archivace

## Úvod

Softwarové řešení pro důvěryhodné uchování elektronických dokumentů založené na aplikaci moderních ETSI standardů rozšířeného elektronického podpisu (AdES). Zajišťuje komplexní důvěryhodné služby pro archivní správu dokumentů v libovolném formátu.



OBELISK Trusted Archive (OB-TA) je důvěryhodným archivem, jehož funkce a základní principy jsou založeny na mezinárodně uznávaných technologických, procesních a legislativních standardech. Mezi hlavní z nich patří:

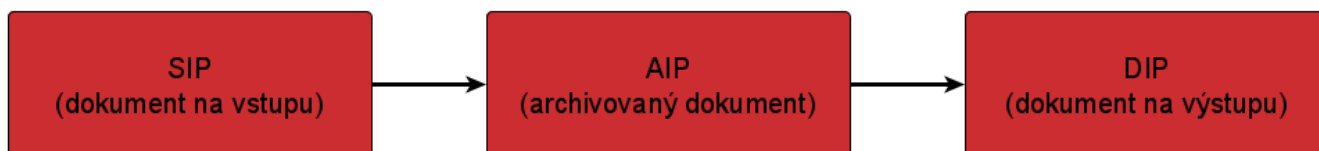
- Nařízení EU 910/2014 eIDAS, zejména pak články:
  - Čl. 26 - Požadavky na zaručené elektronické podpisy
  - Čl. 32 - Požadavky na ověřování platnosti kvalifikovaných elektronických podpisů
  - Čl. 33 - Kvalifikovaná služba ověřování platnosti kvalifikovaných elektronických podpisů
  - Čl. 34 - Kvalifikovaná služba uchovávání kvalifikovaných elektronických podpisů
  - Čl. 36 - Požadavky na zaručené elektronické pečeti
  - Čl. 40 - Ověřování platnosti a uchovávání kvalifikovaných elektronických pečetí \*
- Nečíslovaný seznam
- Referenční model ISO 14721:2012 Open Archival Information System (OAIS)
- Technologické normy ETSI:
  - CADES - ETSI EN 319 122 V1.1.1 (2016-04)
  - PAdES - ETSI EN 319 142 V1.1.1 (2016-04)
  - XAdES - ETSI EN 319 132 V1.1.1 (2016-04)
- Policy and security requirements for applications for signature creation and signature validation (ETSI TS 119 101)
- Procedures for Creation and Validation of AdES Digital Signatures (ETSI EN 319 102 V1.1.1)

Primárním účelem OB-TA je validace a důvěryhodná archivace elektronických dokumentů a jejich

bezpečnostních prvků v podobě elektronických podpisů, elektronických pečeti a elektronických časových razítek, za využití výše uvedených standardů. Validace elektronických bezpečnostních prvků probíhá v samostatném modulu OBELISK Validator. Vytváření elektronických pečeti a elektronicky zabezpečených dokumentů probíhá v samostatném modulu OBELISK Seal.

## Princip archivace

Základní princip archivace elektronických dokumentů je založen na implementaci modelu OAIS.



Dokument do OB-TA vstupuje v podobě SIP (Submission Information Package), tzn. je složen z dokumentu samotného a jeho metadat. V rámci zpracování v OB-TA je dokument uložen v AIP (Archival Information Package), který kromě vstupního dokumentu a jeho metadat zahrnuje dále detaily o ověření bezpečnostních prvků, archivní balíčky, do nichž je dokument (a jeho náležitosti) zařazen a metadata, která v rámci archivace vznikla. Na výstupu z OB-TA je dokument poskytován v DIP (Dissemination Information Package), kdy jsou např. v rámci důkazního materiálu poskytována veškerá data, která prokazují provedenou archivaci a existenci dokumentu v archivu. Veškeré činnosti OB-TA jsou řízeny detailním nastavením různých sad politik, které jsou popsány dále.

## Přehled procesu zpracování OB-TA

Tato část shrnuje proces péče o dokument od jeho přijetí do OB-TA až po dlouhodobé udržování platnosti archivních balíčků.

### Přijetí dokumentu

Dokument je přijat do OB-TA skrz standardní rozhraní modulu OB-IP. Pokud politika příjmu obsahu odkazuje politiku pro příjem dokumentu, je dokument (po jeho uložení do fyzických úložišť) předán do modulu OB-TA k archivaci s daným identifikátorem politiky pro příjem dokumentu k archivaci.

### Archivace dokumentu

Proces archivace zajišťuje zabezpečení všech bezpečnostních prvků, které archivovaný dokument může obsahovat, a jeho zafixování v čase pomocí důvěryhodné časové značky, aby bylo možné o něj dlouhodobě pečovat. Proces archivace lze nad dokumentem vyvolat manuálně i později po jeho uložení, a to opakovaně. Může tedy nastat situace, kdy je dokument archivován vícekrát (s různými politikami pro příjem dokumentu).

V rámci archivačního procesu je nejprve provedena kontrola, zda dokument není již s požadovanou politikou archivován<sup>1</sup>, a pokud ano, je dokument považován za archivovaný a archivační proces zde

končí. Pro kontrolu je důležitý identifikátor politiky příjmu dokumentu, nikoliv skutečné hodnoty jejich řídicích parametrů.

Následně dochází k přepočtu kryptografických otisků všech souborů dokumentu. Pokud tak určuje politika příjmu dokumentu (3.3.2), jsou dále extrahována metadata dokumentu a je nad nimi spuštěn archivační proces.

Nad všemi soubory dokumentu je vyvolán proces archivace souborů (viz dále) a po jejich dokončení je dokumentu nastavena časová značka nejbližšího vypršení některého z podpisů v souborech dokumentu – nejpozději k tomuto datu musí být provedeno přerazítkování souborů dokumentu se zvolenou politikou dlouhodobé péče.

## Archivace souborů dokumentu

Každý archivovaný soubor je nejprve zařazen do balíčku, odkazuje-li politika pro příjem dokumentu na politiku pro vytváření archivního balíčku. Pokud politika specifikuje validační politiku, je nad souborem vyvolána validace. Ta může skončit několika způsoby:

1. Validace typu souboru není podporována a výsledkem procesu není validační report.
2. Soubor byl zvalidován, ale nebyly v něm nalezeny žádné elektronické podpisy.
3. Soubor byl zvalidován a všechny požadavky kladené na něj archivační politikou byly splněny.
4. Soubor byl zvalidován, ale nesplnil některý z požadavků kladený na něj archivační politikou.
5. Při validaci souboru došlo k chybě.

První dva výsledky značí, že není potřeba později opakovat validaci pro získání aktuálních revokačních dat prokazatelně vydaných po zajištění souboru časovým razítkem, protože soubor žádné elektronické podpisy neobsahuje.

V případě třetího výsledku záleží na úrovni elektronického podpisu – pokud obsahuje platné časové razítko, není třeba čekat na časovou fixaci souboru pomocí časového razítka XAdES balíčku; pokud je razítko navíc dostatečně staré, nevzniká ani potřeba opakovat později validaci pro získání revokačních dat, viz předchozí odstavec. V opačném případě je potřeba proces pozastavit až do zajištění souboru pomocí časového razítka XAdES balíčku.

Pokud validace skončí jedním ze dvou posledních výsledků, je opakována až do překročení maximálního povoleného počtu validací, který je určen politikou pro příjem dokumentu. Teprve při jeho překročení je proces archivace přerušen a správce aplikace musí po analýze problému rozhodnout, jak se takový soubor má dále zpracovat.

Po prokazatelném zajištění souboru v čase je (v případě nutnosti, viz předchozí text) provedena druhá validace, která slouží pro získání finálních revokačních dat (CRL, OCSP) a finálního validačního reportu, který obsahuje diagnostická data nutná k dlouhodobému udržování elektronických podpisů dokumentu. Tato data, společně s certifikáty použitými v rámci validace, mohou být archivována dle nastavení politiky pro příjem dokumentu.

## Vytváření a archivace balíčků

Archivní balíčky slouží k zajištění dlouhodobé péče o velké množství dokumentů při současné minimalizaci redundance dat, spotřeby výpočetních prostředků, časových razítek a dalších zdrojů.

Do balíčků jsou postupně zařazovány jednotlivé dokumenty; po naplnění balíčku nebo po vypršení lhůty pro uzavření balíčku je vytvořen podpis ve formátu XAdES, který podepisuje všechny do balíčku zařazené soubory. Podpis je dále zajištěn časovým razítkem, které ve výsledku fixuje v čase i všechny soubory zařazené do balíčku včetně všech elektronických podpisů a dalších bezpečnostních prvků, které mohou být v těchto souborech obsaženy.

Podoba archivních balíčků je určena nastavením politiky pro vytváření archivního balíčku. Dokumenty archivované s různě nastavenou balíčkovací politikou budou zařazeny do různých balíčků. Naproti tomu dokumenty archivované se stejnou balíčkovací politikou (shodovat se musí i hodnoty jejich parametrů) mohou být zařazeny společně do jednoho balíčku.

I soubory jednoho dokumentu mohou být zařazeny do více balíčků, a to například v případě, že v průběhu zařazování souborů je naplněna kapacita balíčku a musí být založen balíček nový.

# Konfigurace

Konfigurace modulu OBELISK Trusted Archive se provádí na několika úrovních:

- Kompozitní politiky příjmu obsahu
- Globální konfigurace modulu OB-TA
- Politiky pro příjem dokumentu do archivu
- Politiky pro vytváření archivních balíčků

Jednotlivé konfigurace jsou detailně rozebrány v následujícím textu.

## Kompozitní politiky příjmu obsahu

Všechny dokumenty vstupující do OBELISKu pro další zpracování (uložení, archivaci, apod.), jsou primárně zpracovávány tzv. kompozitními (sdruženými) politikami, které definují, jak se OBELISK k danému předanému dokumentu zachová. Kompozitní politiky jsou konfigurovány na cestě:  
root/modules/broker/policies

Popis služeb - viz: [Produkty - API-Index článků - Broker - Kompozitní politika uložení](#)

## Politiky pro příjem do archivu a pro tvorbu balíčků

Viz: [Produkty - API-index článků - Broker](#)

# Webové služby - Úvod

## Autentizace webových služeb

Autentizace standardně probíhá pomocí basic access authentication a http hlavičky authorization. Případně je možné podepsat XML SOAP zprávu pomocí WS-Security a systémovému účtu nakonfigurovat autentizační certifikát.

Endpoint může podporovat pouze jeden typ autentizace v daný moment.

## Popis služby

Technický způsob realizace v případě SOAP protokolu definují přiložené WSDL/XSD specifikace - v příloze.

Vstupní data (binární obsah) souborů dokumentu je předáván ve formě přílohy dle standardu MTOM (méně vhodná, ale podporovaná varianta je i BASE64 kódování v XML zprávě). Na výstupu je podporován pouze standard MTOM.

Všechny metody rozhraní poskytují synchronní odpověď (blokující zpracování). Některé operace iniciují spuštění asynchronního procesu (neblokující zpracování), na jehož výsledek zdrojový systém bude:

- nečeká, protože ho nepotřebuje znát (doručením synchronní odpovědi OBELISK přebírá zodpovědnost za vyřízení požadavku, příp. problémy řeší administrátor systému OBELISK),
- definovanou dobu synchronně čeká (viz dále parametr timetowait),
- aktivně se dotazuje (viz dále parametr processId), nebo
- bude uvědomen pomocí callbacku

V případě chyby při přijetí požadavku (chyba vstupů, neočekávané stavy systému, neošetřené výjimky) vrací metody chybu (SOAP fault, http status 500). Za vyřízení takových požadavků systém OBELISK zodpovědnost nemohl převzít a zdrojový systém musí požadavek zopakovat (příp. zopakovat korektně). V případě chyb provádění asynchronních procesů metoda vrací standardní odpověď, kde hlavička odpovědi obsahuje informace o stavu procesu. Zotavení z chyby může být provedeno automaticky nebo manuálně na straně systému OBELISK.

V následujících tabulkách jsou povinné atributy zpráv označeny tučným písmem. OBELISK Integration Platform nabízí tyto služby:

### Parametry požadavku hlavičky volání

Název	Typ	Význam
1. <b>processId</b>	Long	Identifikátor procesu zpracování vrácený systémem OBELISK v předchozím volání. Pokud je uveden, dostane klient v odpovědi stav zpracování požadavku s tímto id.



Název	Typ	Význam
2.timetowait	Long	Specifikace doby (v ms), po kterou chce klient čekat nejdéle na odpověď včetně vyřizování asynchronní části. Vždy se čeká na provedení synchronní části operace.
3.policy		Politika zpracování v OBELISK.
3.1.code	String	Kód politiky zpracování
3.2.arg	Array	Specifikace konkrétních hodnot argumentů politiky
3.2.1.key	String	Název parametru
3.2.2.dataType	Enum	Datový typ parametru, implicitně text
3.2.3.value	String	Hodnota parametru typu text nebo dlouhý text
3.2.4.valueNum	Number	Hodnota parametru typu číslo
3.2.5.valueDate	Date	Hodnota parametru typu datum
3.2.6.valueDateTime	DateTime	Hodnota parametru typu datum s časem

## Parametry hlavičky odpovědi

Název	Typ	Význam
processId	Long	Identifikátor procesu zpracování. Klient se pomocí tohoto id může zeptat na stav procesu
state	String	Kód identifikující stav procesu zpracování, nabývá hodnot: IN_PROGRESS, SUCCEEDED, BLOCKED, FAILED, FAILED_CONTINUE, TIMED_OUT
<b>timelasted</b>	Long	Přibližná doba (v ms), kterou trvalo vyřízení požadavku.
taskInfo	Array	Detailní informace o stavu jednotlivých částí procesu, pokud jsou k dispozici
taskInfo	Array	Rekurzivní definice (podproces)
<b>deliveryPhase</b>	Integer	Fáze zpracování procesu
<b>name</b>	String	Identifikace části procesu
description	String	Popis části procesu
<b>state</b>	String	Stav procesu
processedPhase	String	Již zpracované fáze této části procesu
logMessage	String	Zpráva, např. popis chyby
processingTime	Long	Doba posledního zpracování daného kroku procesu v ms.
processAfter	DateTime	Čas, kdy lze nejdříve začít se zpracováním daného kroku
retryCount	Integer	Počet pokusů o provedení daného kroku procesu
<b>lastChange</b>	DateTime	Čas poslední změny stavu kroku procesu
<b>createdAt</b>	DateTime	Čas vytvoření kroku procesu

## kód XML

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://www.sefira.cz/obb/api/storage/v1/">
  <soapenv:Header>
    <v1:ResponseHeader processId="675090" state="IN_PROGRESS"
timelasted="7670">
      <v1:taskinfo createdAt="2018-02-19T16:33:44+01:00"
description="Kompozitní uložení" lastChange="2018-02-19T16:33:47+01:00"
name="EvtcompositeStore" processedPhase="50" state="IN_PROGRESS">
        <v1:taskinfo createdAt="2018-02-19T16:33:45+01:00"
deliveryPhase="50" descriptions="Fyzické uložení dokumentu"
lastChange="2018-02-19T16:33:46+01:00"
name="EvtStoreDocumentRequest" processedPhase="20"
state="IN_PROGRESS">
          <v1:taskinfo createdAt="2018-02-19T16:33:46+01:00"
deliveryPhase="20" descriptions="fs_storeDocumentTask"
lastChange="2018-02-19T16:33:50+01:00"
name="fs_storeDocumentTask" state="IN_PROGRESS">
            </v1:taskinfo>
          </v1:taskinfo>
        </v1:taskinfo>
      </v1:ResponseHeader>
    </soapenv:Header>
  </soapenv:Envelope>
```

# Webové služby - StoreDocument

Tato operace zaeviduje dokument v systému OBELISK. Informace o dokumentu volitelně zahrnují binární obsah elektronického dokumentu, hodnoty metadatových položek a identifikace dokumentu v jiných úložištích/systémech. Výsledek synchronního volání obsahuje především identifikaci uloženého dokumentu ve jmenném prostoru OBB. Tento identifikátor může zdrojový systém (bezprostředně po obdržení) použít v dalších operacích tohoto rozhraní s daným dokumentem. Použitá politika případně definuje nastavenou retenci pro daný dokument. Na zaevidování dokumentu mohou být navázány další asynchronní procesy zpracování (validace, archivace, opatření pečetí a podobně). Ty do značné míry ovlivňují podobu odpovědi, viz dále.

## StoreDocumentRequest

Název	Typ	Význam
contents	Array	Pole se soubory, které se mají uložit
- name	String	Název souboru
- mime	String	MIME typ souboru
- data	Attachment	Binární obsah souboru (příloha, BASE64, part)
metadata	Array	Pole metadatových položek
- references	Array	Odkaz na dokument pomocí jedné či více referencí pro položku typu reference
- data type	Enum	Datový typ hodnoty metadatové položky, implicitně text
- <b>key</b>	String	Název metadatové položky
- value	String	Hodnota metadatové položky typu text nebo dlouhý text
- valueNum	Number	Hodnota metadatové položky typu číslo
- valueDate	Date	Hodnota metadatové položky typu datum
- valueDateTime	DateTime	Hodnota metadatové položky typu datum s časem
references	Array	Nepovinné identifikátory dokumentu v dalších registrovaných systémech
- <b>id</b>	String	Identifikace dokumentu
- idspace	String	Identifikace jmenného prostoru
- version	Long	Verze dokumentu
versionTo/references	Array	Nepovinné pole referencí identifikujících předešlou verzi dokumentu. Pokud je referencí více (typicky je specifikována pouze jedna), musí všechny identifikovat stejný dokument.

Příklad:

[kód XML](#)

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://www.sefira.cz/obb/api/storage/v1/">
  <soapenv:Body>
    <v1:StoreDocumentRequest>
      <v1:contents>
        <v1:content>
          <v1:name>document.pdf</v1:name>
          <v1:mime>application/pdf</v1:mime>
          <v1:data>cid:1111</v1:data>
        </v1:content>
      </v1:contents>
      <v1:metadata>
        <v1:meta key="CPS" value="895135413541"/>
        <v1:meta key="NAZEVDOK" value="Soudní řízení"/>
        <v1:meta datatype="NUMBER" key="CONST" value="6425"/>
      </v1:metadata>
    </v1:StoreDocumentRequest>
  </soapenv:Body>
</soapenv:Envelope>

```

## Uložení nové verze dokumentu

Nová verze dokumentu nepřebírá nic ze stávající verze, vzniká jako nový dokument ve verzi N+1 podle zadání ve StoreDocument requestu.

Při odeslání následující části XML requestu vznikne verze N+1 k dokumentu OBB\_123:

```

<v1:versionTo>
  <v1:references>
    <v1:reference>
      <v1:id>OBB_123</v1:id>
      <v1:idspace>OBB</v1:idspace>
      <v1:version>1</v1:version> <!-- nepovinné -->
    </v1:reference>
  </v1:references>
</v1:versionTo>

```

Pro uložení nové verze existujícího dokumentu je možné zadat číslo verze v referenci <versionTo>, není to však nutné. Číslo neoznačuje verzi, která je právě ukládána, ale číslo existující verze dokumentu, která je již v OBELISKu uložená a ke které se bude vytvářet nová verze. Číslo verze nově vznikajícího dokumentu je inkrementální a vždy vzniká jako verze N+1, kde N je číslo nejnovější (poslední) verze a to i když je v requestu zmíněno, že dokument vychází ze starší revize.

## StoreDocumentResponse

Název	Typ	Význam
<b>references</b>	Array	Pole identifikací, které jsou k dokumentu přiřazeny. Vždy obsahuje alespoň jednu identifikaci, a to ve jmenném prostoru OBB. Pořadí identifikací v různých jmenných prostorech není obecně definováno.
- <b>id</b>	String	Identifikace dokumentu v externím systému
- <b>idspace</b>	String	Identifikace jmenného prostoru
- <b>version</b>	Long	Verze dokumentu
- <b>date</b>	DateTime	Časová značka vytvoření identifikace
<b>contents</b>	Array	Pole s uloženými soubory
- <b>name</b>	String	Název souboru
- <b>processingInfo</b>	Any	Dodatečné informace o zpracování

Příklad:

[kód XML](#)

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://www.sefira.cz/obb/api/storage/v1/"
  <soapenv:Header>
    <v1:RequestHeader processId="22038061" timelasted="505"
state="IN_PROGRESS"/>
  </soapenv:Header>
  <soapenv:Body>
    <v1:StoreDocumentResponse>
      <v1:references>
        <v1:reference>
          <v1:id>OBB_127507421</v1:id>
          <v1:idspace>OBB</v1:idspace>
        </v1:reference>
      </v1:references>
      <v1:contents>
        <v1:content>
          <v1:name>example.pdf</v1:name>
          <v1:processingInfo>
            <v1:validation>
              <v1:result>TOTAL_PASSED</v1:result>
              <v1:report>
                <v1:id>OBB_127508952</v1:id>
                <v1:idspace>OBB</v1:idspace>
                <v1:date>2016-06-28T13:15:21.000+02:00</v1:date>
              </v1:report>
            </v1:validation>
          </v1:processingInfo>
        </v1:content>
      </v1:contents>
    </v1:StoreDocumentResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

```
</v1:content>  
</v1:contents>  
</v1:StoreDocumentResponse>  
</soapenv:Body>  
</soapenv:Envelope>
```

K uvedenému výstupu mohou být doplněny další (tzv. `processingInfo`, míra podrobností je opět řízena politikou, např. vkládání kompletního validačního reportu) v případě sdružených operací, např. validace, opatření dokumentu pečeti apod. V takovém případě je vhodné politikou aktivovat vkládání `taskInfo` hlavičky, která signalizuje, v jaké fázi zpracování se dokument v okamžiku zaslání odpovědi nachází a jaké výstupy mohou být očekávány. Konkrétní interpretace hlavičky závisí na konkrétním nastavení politiky a procesů zpracování. Při změně stavu zpracování může být v závislosti na konkrétní konfiguraci procesu aktivován asynchronní callback se shodnou strukturou doplňkových informací (`processingInfo`); tím je možné u dlouhotrvajících procesů zajistit automatickou notifikaci klienta při změně stavu bez nutnosti opakovaného volání služby s identifikací tohoto procesu. V případě, že klient dostane při synchronním zpracování již úplnou informaci, je věcí politiky, zda se v takovémto případě zasílá callback či nikoliv.

Podrobnosti o `processingInfo` jsou v sekci týkající se callbacku.

# Webové služby - Callback

Standardní produktový callback obsahuje v hlavičce požadavku prvky processId, state a taskInfo, stejně, jako jsou popsány v ostatních webových službách. Pro spárování callbacku s požadavkem, který callback inicioval, lze použít buď hodnotu processId, nebo libovolnou z identifikací dokumentu v těle callback požadavku.

Samotné tělo callback požadavku obsahuje následující informace:

Název	Typ	Význam
<b>references</b>	Array	Pole identifikací, které jsou k dokumentu přiřazeny
<b>id</b>	String	Identifikace dokumentu
<b>idspace</b>	String	Identifikace jmenného prostoru
version	Long	Verze dokumentu
<b>date</b>	DateTime	Časová značka vytvoření identifikace
contents	Array	Pole s informacemi o souborech dokumentu
name	String	Název souboru
mime	String	MIME typ dokumentu
processingInfo / validation		Informace o ověření elektronických podpisů dokumentu
<b>result</b>	Enum	Souhrnný výsledek ověření elektronických podpisů souboru.
report		Odkaz na dokument obsahující výsledek ověření (val. report), příp. i s vloženým obsahem reportu.
additionalInfo	Array	Sada prvků klíč-hodnota s doplňujícími informacemi o validaci.

# Webové služby - UpdateMetadata

Slouží k aktualizaci hodnot metadatových položek zvoleného dokumentu a/nebo aktualizaci retence (vyplývá z použité politiky). V těle volání metody systém předává identifikátor dokumentu a nové hodnoty metadatových položek.

## UpdateMetadataRequest

Název	Typ	Význam
<b>id</b>	String	Identifikace dokumentu
idspace	String	Identifikace jmenného prostoru
version	Long	Verze dokumentu - pokud není vyplněno, modifikuje se nejnovější verze
metadata	Array	Pole metadatových položek
references	Array	Odkaz na dokument pomocí jedné či více referencí pro položku typu reference
dataType	Enum	Datový typ hodnoty metadatové položky, implicitně text
<b>key</b>	String	Název metadatové položky
value	String	Hodnota metadatové položky typu text nebo dlouhý text
valueNum	Number	Hodnota metadatové položky typu číslo
valueDate	Date	Hodnota metadatové položky typu datum
valueDateTime	DateTime	Hodnota metadatové položky typu datum s časem

## Modifikace jednotlivých verzí dokumentu

Specifikací verze a identifikátoru dokumentu, lze modifikovat jednotlivé verze dokumentů. Pokud není číslo verze vyplněno, modifikuje se vždy nejnovější (poslední) verze.

## UpdateMetadataResponse

Nenese žádnou informaci.



# Webové služby - RetrieveDocument

Slouží pro získání uloženého dokumentu, nebo jeho metadata. Metadata a případně binární obsah souborů jsou volajícimu systému vráceny typicky v synchronní odpovědi.

## RetrieveDocumentRequest

Název	Typ	Význam
id	String	Identifikace dokumentu
idspace	String	Identifikace jmenového prostoru
version	Long	Identifikace verze
metadataOnly	Boolean	Příznak, zda jsou požadována pouze metadata
contentNumber	Long	Pořadové číslo souboru s obsahem, který se má vyzvednout. Pokud není určeno, jsou vyzvednuty všechny soubory.

Příklad:

[kód XML](#)

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://www.sefira.cz/obb/api/storage/v1/">
  <soapenv:Body>
    <v1:RetrieveDocumentRequest>
      <v1:id>OBB_12050742</v1:id>
      <v1:idspace>OBB</v1:idspace>
    </v1:RetrieveDocumentRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

## RetrieveDocumentResponse

Název	Typ	Význam
metadata	Array	Pole metadatových položek
dataType	Enum	Datový typ hodnoty metadatové položky
key	String	Název metadatové položky
references	Array	Odkaz na dokument pomocí jedné či více referencí pro položku typu reference.
value	String	Hodnota metadatové položky typu text nebo dlouhý text

Název	Typ	Význam
valueNum	Number	Hodnota metadatové položky typu číslo
valueDate	Date	Hodnota metadatové položky typu datum
valueDateTime	DateTime	Hodnota metadatové položky typu datum s časem
<b>references</b>	Array	Pole identifikací, které jsou k dokumentu přiřazeny
<b>id</b>	String	Identifikace dokumentu v externím systému
<b>idspace</b>	String	Identifikace jmenného prostoru
version	Long	Verze dokumentu
<b>date</b>	DateTime	Časová značka vytvoření identifikace
<b>contentCount</b>	Long	Počet evidovaných souborů
<b>contents</b>	Array	Soubor s daty
name	String	Název souboru
mime	String	MIME typ souboru
data	Attachment	Soubor formou přílohy

Příklad:

[kód XML](#)

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://www.sefira.cz/obb/api/storage/v1/"
  <soapenv:Header>
    <v1:RequestHeader processId="22038061" timelasted="505"
state="IN_PROGRESS"/>
  </soapenv:Header>
  <soapenv:Body>
    <v1:RetrieveDocumentResponse>
      <v1:metadata>
        <v1:meta key="CPS" dataType="TEXT" value="895135413541"/>
        <v1:meta key="NAZEVDOK" dataType="TEXT" value="Soudní
řízení"/>
        <v1:meta datatype="CONST" value="6425"/>
      </v1:metadata>
      <v1:references>
        <v1:reference>
          <v1:id>0BB_127507421</v1:id>
          <v1:idspace>0BB</v1:idspace>v1:idspace>
        <v1:date>2016-06-28T13:15:21.000+02:00</v1:date>v1:date>
        <v1:reference>
        <v1:reference>
          <v1:id>LKJHGG</v1:id>
          <v1:idspace>centeraPoolA</v1:idspace>
        <v1:date>2016-06-28T13:15:21.000+02:00</v1:date>v1:date>
```

```
<v1:reference>
</v1:references>
</v1:contents>
  <v1:content>
    <v1:name >example.pdf</v1:name>v1:name>
    <v1:mimo>application/pdf<v1:mimo>
    <v1:data>inc:Include
      xmlns:inc=http://www.w3.org/2004/08/xop/include"
      href="cid:1"/></v1:data>
  </v1:content>
</v1:contents>
<v1:RetrieveDocumentResponse>
</soapenv:Body>
</soapenv:Envelope>
```

# Webové služby - RemoveDocument

Metoda zajišťuje mazání (skartaci) dokumentu z úložišť příp. vyřazení dokumentu z evidence v systému OBELISK. V těle volání metody je předána identifikace dokumentu. Použitá politika určuje úložné systémy, ze kterých se má dokument smazat. Použitá politika může blíže specifikovat režim odstranění dokumentu např. z garantovaného úložiště (privilegovaný režim, komentář aj.).

## RemoveDocumentRequest

Název	Typ	Význam
id	String	Identifikace dokumentu
idspace	String	Identifikace jmenného prostoru
version	Long	Verze dokumentu - pokud není vyplněno, maže se nejnovější verze

Příklad:

[kód XML](#)

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://www.sefira.cz/obb/api/storage/v1/">
  <soapenv:Body>
    <v1:RemoveDocumentRequest>
      <v1:id>0BB_2913737</v1:id>
      <v1:idspace>0BB</v1:idspace>
      <v1:version>1</v1:version><!-- nepovinné -->
    </v1:RemoveDocumentRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

## Mazání verzí dokumentu

Smazat lze libovolnou verzi dokumentu. V případě, že se verze dokumentu nezmění, maže se vždy nejnovější (poslední) verze

- Pokud je smazána nejnovější (poslední) verze N a následně uložena další verze, bude její číslo znovu N.
- Pokud je smazána verze č. 2 v řadě deseti verzí, vznikne ve verzích „díra“.

## RemoveDocumentResponse

Nenese žádnou informaci.

# Webové služby - RepositoryFunction

Slouží k vyvolání speciální funkce úložiště nebo k řízení podobných vlastností na úrovni systému OBELISK. Může se např. jednat o změnu retenční doby, vyvolání litigation hold, apod. V těle volání metody je předáván identifikátor funkce a parametry volání.

## RepositoryFunctionRequest

Název	Typ	Význam
<b>id</b>	String	Identifikace dokumentu
idspace	String	Identifikace jmenného prostoru
version	Long	Verze dokumentu
<b>function</b>	String	Identifikace funkce
parameters	Array	Parametry volání
<b>key</b>	String	Název parametru
dataType	Enum	Datový typ parametrů, implicitně text
value	String	Hodnota parametru typu text nebo dlouhý text
valueNum	Number	Hodnota parametru typu číslo
valueDate	Date	Hodnota parametru typu datum
valueDateTime	DateTime	Hodnota parametru typu datum s časem

## RepositoryFunctionResponse

Nenese žádnou informaci

## Podporované funkce souborového úložiště

- Funkce **triggerEvent** – vyvolá EBR událost s předanými parametry retence. Politika operace případně specifikuje:
  - Parametr **retentionPeriod** – Délka retenční doby, která se má nastavit po vyvolání události (v sekundách).
- Funkce **setLitigationHold** – nastaví/vypne konkrétní litigation hold:
  - Parametr **holdName** – název litigation hold, který se aplikuje. Musí být vyplněn.
  - Parametr **holdValue** – true = zapne litigation hold, false = vypne litigation hold.

# Důkazní materiál

OBELISK Trusted Archive na vyžádání generuje důkazní materiál v podobě DIP (Dissemination Information Package), který v ZIP souboru zahrnuje veškerá data, která prokazují provedenou archivaci a existenci dokumentu v archivu:

- Vlastní dokumenty
- Metadata dokumentů
- Validační reporty
- Validační politiky
- Archivní balíčky
- Statická validační data (certifikáty)
- Dynamická validační data (CRL, ORS - OCSP odpovědi, TSL).

Sestavení a výdej důkazního materiálu se řídí příslušnou politikou, která je v produktu definována a pro standardní použití ji není třeba měnit. Lze vyžádat důkazní materiál jak pro jednotlivé dokumenty, tak pro archivní balíčky. Důkazní materiál lze vyžádat prostřednictvím GUI nebo webovou službou RetrieveVerification.

## Vyžádání přes GUI

Tlačítko Sestavit důkazní materiál je k dispozici pro dokumenty na výsledkových sadách Obsah archivu - Vložené dokumenty a Obsah archivu - Úplný obsah, pro archivní balíčky ve výsledkové sadě Obsah archivu - Archivní balíčky. Je nutno použít zaškrťovací tlačítko.

## Vyžádání requestem webové služby

Služba očekává volání na adrese

<https://hostname/obelisk-broker/services/ArchiveServiceAccess>. Politika volání se liší od kompozitní politiky. Jako parametr volání lze použít jak identifikátor OBBID, tak jiný používaný identifikátor.

Příklad volání:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://www.sefira.cz/obb/api/storage/v1/"
xmlns:v11="http://www.sefira.cz/obb/api/archive/v1/">
  <soapenv:Header>
    <v1:RequestHeader timetowait="100000">
      <v1:policy code="polArchEvidence-default"></v1:policy>
    </v1:RequestHeader>
  </soapenv:Header>
  <soapenv:Body>
    <v11:RetrieveVerificationRequest>
      <v11:requestedDocuments>
        <!--Zero or more repetitions!-->
      </v11:requestedDocuments>
    </v11:RetrieveVerificationRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

```

    <v1:requestedDocument>
      <v1:id>OBB_35121</v1:id>
      <!--Optional:-->
      <v1:idspace>OBB</v1:idspace>
    </v1:requestedDocument>
  </v1:requestedDocuments>
</v1:RetrieveVerificationRequest>
</soapenv:Body>
</soapenv:Envelope>

```

Odpověď (v příloze ZIP soubor s důkazním materiálem):

```

<soapenv:Envelope xmlns:oba="http://www.sefira.cz/obb/api/archive/v1/"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://www.sefira.cz/obb/api/storage/v1/">
  <soapenv:Header>
    <v1:ResponseHeader processId="23891" state="SUCCEEDED"
timelasted="1232"/>
  </soapenv:Header>
  <soapenv:Body>
    <oba:RetrieveVerificationResponse>
      <v1:metadata/>
      <v1:references/>
      <v1:contentCount>1</v1:contentCount>
      <v1:contents>
        <v1:content>
          <v1:name>evidence.zip</v1:name>
          <v1:mime>application/zip</v1:mime>
          <v1:data fileSize="165993">
            <inc:Include href="cid:32262/1"
xmlns:inc="http://www.w3.org/2004/08/xop/include"/>
          </v1:data>
        </v1:content>
      </v1:contents>
    </oba:RetrieveVerificationResponse>
  </soapenv:Body>
</soapenv:Envelope>

```



# Přílohy ke stažení - Webové služby

Přílohy jsou dostupné online na následujících adresách:

<b>Příklady komunikace</b>	<b>Adresa ke stažení</b>
Store document	<a href="https://doc.obelisk.eu/public/lib/exe/fetch.php?media=download:doc:v38:ob-ds:store_document-examples.zip">https://doc.obelisk.eu/public/lib/exe/fetch.php?media=download:doc:v38:ob-ds:store_document-examples.zip</a>
Retrieve document	<a href="https://doc.obelisk.eu/public/lib/exe/fetch.php?media=download:doc:v38:ob-ds:retrieve_document-examples.zip">https://doc.obelisk.eu/public/lib/exe/fetch.php?media=download:doc:v38:ob-ds:retrieve_document-examples.zip</a>
Update metadata	<a href="https://doc.obelisk.eu/public/lib/exe/fetch.php?media=download:doc:v38:ob-ds:update_metadata-examples.zip">https://doc.obelisk.eu/public/lib/exe/fetch.php?media=download:doc:v38:ob-ds:update_metadata-examples.zip</a>
Repository function	<a href="https://doc.obelisk.eu/public/lib/exe/fetch.php?media=download:doc:v38:ob-ds:repository_function-examples.zip">https://doc.obelisk.eu/public/lib/exe/fetch.php?media=download:doc:v38:ob-ds:repository_function-examples.zip</a>
Remove document	<a href="https://doc.obelisk.eu/public/lib/exe/fetch.php?media=download:doc:v38:ob-ds:remove_document-examples.zip">https://doc.obelisk.eu/public/lib/exe/fetch.php?media=download:doc:v38:ob-ds:remove_document-examples.zip</a>
Document batch	<a href="https://doc.obelisk.eu/public/lib/exe/fetch.php?media=download:doc:v38:ob-ds:document_batch-examples.zip">https://doc.obelisk.eu/public/lib/exe/fetch.php?media=download:doc:v38:ob-ds:document_batch-examples.zip</a>
Asynchronous document batch	<a href="https://doc.obelisk.eu/public/lib/exe/fetch.php?media=download:doc:v38:ob-ds:async_document_batch-examples.zip">https://doc.obelisk.eu/public/lib/exe/fetch.php?media=download:doc:v38:ob-ds:async_document_batch-examples.zip</a>
Batch info	<a href="https://doc.obelisk.eu/public/lib/exe/fetch.php?media=download:doc:v38:ob-ds:batch_info-examples.zip">https://doc.obelisk.eu/public/lib/exe/fetch.php?media=download:doc:v38:ob-ds:batch_info-examples.zip</a>
Callback	<a href="https://doc.obelisk.eu/public/lib/exe/fetch.php?media=download:doc:v38:ob-ds:callback-examples.zip">https://doc.obelisk.eu/public/lib/exe/fetch.php?media=download:doc:v38:ob-ds:callback-examples.zip</a>
<b>Definice webové služby</b>	<b>Adresa ke stažení</b>
WSDL a XSD	<a href="https://doc.obelisk.eu/public/lib/exe/fetch.php?media=download:doc:v38:ob-ds:storage-wsdl.zip">https://doc.obelisk.eu/public/lib/exe/fetch.php?media=download:doc:v38:ob-ds:storage-wsdl.zip</a>



SEFIRA spol. s r.o.  
Antala Staška 2027/77  
140 00 Praha  
Česká republika  
[www.sefira.cz](http://www.sefira.cz)