**Rámcová dohoda o součinnosti**

Smluvní strany:

Město Český Krumlov

se sídlem: náměstí Svornosti 1, 381 01 Český Krumlov

zastoupené: Mgr. Dalibor Carda, starosta města

IČ, DIČ: 00245836, CZ00245836

bankovní spojení: Komerční banka a.s., číslo účtu: 221241/0100

(dále jen „**Město**“)

a

**ICZ a.s.**

se sídlem: Na hřebenech II 1718/10, Nusle, 140 00 Praha 4

jednající: Vlasta Šejvlová, na základě plné moci

IČ, DIČ 25145444, CZ699000372

společnost zapsaná v obchodním rejstříku vedeném Městským soudem v Praze, sp. zn. B 4840

bankovní spojení: UniCredit Bank Czech Republic and Slovakia, a.s.,

číslo účtu: 2109164825/2700

(dále jen „**Poskytovatel**“)

(Město a Poskytovatel společně též označováni dále jako „**smluvní strany**“)

Smluvní strany dnešního dne uzavřely tuto rámcovou dohodu o součinnosti v souladu s ustanovením § 1746 odst. 2 zákona č. 89/2012 Sb., občanského zákoníku, v platném znění (dále jen „**občanský zákoník**“ nebo „OZ“) (dále jen „Smlouva“).

Smluvní strany, vědomy si svých závazků v této Smlouvě obsažených a s úmyslem být touto Smlouvou vázány, dohodly se na následujícím znění Smlouvy:

PREAMBULE

Město je příjemcem dotace v rámci projektu „Rozvoj služeb eGovernmentu města Český Krumlov“, reg. č. CZ.06.3.05/0.0/0.0/16\_044/0006341 (dále jen „**projekt**“).

Projekt připraví prostředí pro sdílení elektronických služeb v rámci portálových prostředí včetně možnosti elektronického podání využitím elektronických formulářů, zajistí bezpečné a důvěryhodné prostředí pro výkon eGovernmentových služeb, doplní prvky pro bezpečnost a zajistí odpovídající úroveň síťové infrastruktury. Projekt si tak klade za cíl posílit infrastrukturu, konsolidovat ICT, podpořit standardizaci informačního systému města a zajistit potřebnou úroveň poskytování nejen elektronických služeb. Dojde tak k zefektivnění, zvýšení transparentnosti činnosti organizací Města, přístupnosti služeb a zajištění nezbytné bezpečnosti. Projekt zároveň vytváří podmínky, aby v dalším kroku po zefektivnění řízení procesů v rámci organizací Města došlo k významným úsporám na kapacitách či zdrojích Města.

Sdílí se elektronické služby, využívá se centrálních služeb eGovernmentu a doplní se infrastruktura pro uskutečnění záměrů. S tímto projektem budou sladěné další klíčové projekty Města, které spolu tvoří logický celek.

Projekt respektuje celostátní strategie a využívá jich ve svůj prospěch. Je s nimi komplementární a využívá centrálních služeb produkovaných státem. Je zároveň v souladu se strategickým dokumentem Digitální strategie města Český Krumlov, který udává základní směřování Města při využívání moderních technologií, určuje strategické priority a cíle a v neposlední řadě uvádí základní principy, kterými se Město řídí při pořizování a provozu informačních technologií.

Jedním z poptávaných řešení (mimo plnění této Smlouvy) je pořízení Centrální elektronické spisovny spočívající mimo jiné v integraci Centrální elektronické spisovny s eSSL ICZ e-spis® (dále jen „řešení“). V předchozí větě uvedené řešení tedy bude napojeno na stávající ICT systémy (ICT infrastrukturu) Města, aby mohlo být funkční, a současně aby mohlo naplnit cíle projektu.

Tato Smlouva je výsledkem zadávacího řízení s názvem „Rozvoj služeb eGovernmentu města Český Krumlov – část 2 – Integrace eSSL ICZ e-spis® “ vyhlášeného Městem.

1. PŘEDMĚT SMLOUVY
	1. Předmětem této Smlouvy je stanovení podmínek pro uzavírání dílčích smluv a/nebo objednávek (dále jen „**dílčích smluv**“), kdy na základě těchto dílčích smluv bude Poskytovatelem Městu poskytnuta součinnost k integraci eSSL ICZ e-spis® provozovaného a užívaného Městem s ICT řešením Centrální elektronické spisovny dodaným vítězným dodavatelem – vítězem veřejné zakázky „Rozvoj služeb eGovernmentu města Český Krumlov – část 10 – Centrální elektronická spisovna“ vyhlášeného Městem.
	2. Podrobný popis předmětu plnění je uveden v přílohách číslo 1 a 3 této Smlouvy a bude upřesněn na základě dílčích smluv
	3. Integrace proběhne v krocích a čase dle harmonogramu dle přílohy č. 2 Smlouvy a bude upřesněna na základě dílčích smluv.
2. CENA ZA PŘEDMĚT PLNĚNÍ
	1. Cena předmětu plnění jednotlivých dílčích smluv bude vypočtena jako násobek počtu hodin, za které Poskytovatel dané plnění na základě dílčí smlouvy poskytne a hodinové sazby za služby Poskytovatele, která činí 1 850,- Kč bez DPH. Maximální počet hodin za plnění dodané Poskytovatelem na základě všech dílčích smluv je limitován na 220 hodin a pokrývá zejména konfigurační a konzultační služby spojené s dodáním adekvátního API rozhraní eSSL e-spis, součinnost s dodavatelem řešení, na které je stávající ICT řešení připojováno a případné ostatní náklady spojené s dodáním API rozhraní systému eSSL ICZ e-spis®.
	2. Cena bude hrazena na základě daňového dokladu (faktury) vystavené Poskytovatelem s třiceti (30) denní splatností. Poskytovatel je povinen Městu řádně vyúčtovat poskytnutý předmět plnění na základě každé dílčí smlouvy nejpozději ve lhůtě 10 dnů od předání daného plnění na základě této Smlouvy, resp. dílčí smlouvy. Faktura bude obsahovat náležitosti daňového účetního dokladu, formou a obsahem odpovídat zákonu č. 563/1991 Sb., o účetnictví, ve znění pozd. předpisů, zákonu o DPH a bude mít náležitosti obchodní listiny dle § 435 občanského zákoníku, ve faktuře bude rovněž uvedena informace, že je vyhotovena na základě této Smlouvy a odkaz na příslušnou dílčí smlouvu. V případě, že daňový doklad (faktura) nebude obsahovat tyto náležitosti, bude Městem vrácen k opravení bez proplacení. V takovém případě lhůta splatnosti počíná běžet znovu ode dne doručení opraveného či nově vyhotoveného daňového dokladu (faktury).
	3. Každý daňový doklad (faktura) musí být označen názvem a číslem projektu „Rozvoj služeb eGovernmentu města Český Krumlov“, reg. č. CZ.06.3.05/0.0/0.0/16\_44/0006341.
3. DALŠÍ UJEDNÁNÍ
	1. Poskytovatel se jako osoba povinná dle ustanovení § 2 písm. e) zákona č. 320/2001 Sb., o finanční kontrole ve veřejné správě, v platném a účinném znění, zavazuje spolupůsobit při výkonu finanční kontroly, mj. umožnit všem subjektům oprávněným k výkonu kontroly přístup ke všem dokumentům, tedy i k těm částem nabídek, smluv a souvisejících dokumentů, které podléhají ochraně podle zvláštních právních předpisů (např. obchodní tajemství), a to za předpokladu, že budou splněny požadavky kladené právními předpisy; tuto povinnost rovněž zajistí Poskytovatel u případných poddodavatelů Poskytovatele.
	2. Smluvní strany jsou povinny uchovávat veškerou dokumentaci související s realizací projektu včetně účetních dokladů minimálně do konce roku 2030. Pokud je v platných právních předpisech stanovena lhůta delší, musí ji smluvní strany dodržet.
	3. Smluvní strany jsou povinny minimálně do konce roku 2030 poskytovat požadované informace a dokumentaci související s realizací projektu zaměstnancům nebo zmocněncům pověřených orgánů (Centru pro regionální rozvoj, Ministerstvu pro místní rozvoj ČR, Ministerstvu financí ČR, Evropské komise, Evropského účetního dvora, Nejvyššího kontrolního úřadu, příslušného orgánu finanční správy a dalších oprávněných orgánů státní správy) a jsou povinny vytvořit výše uvedeným osobám podmínky k provedení kontroly vztahující se k realizaci projektu a poskytnout jim při provádění kontroly součinnost.
	4. Smluvní pokuta pro případ prodlení Poskytovatele s provedením kteréhokoliv dílčího plnění na základě uzavření Dílčí smlouvy, činí za každý započatý kalendářní den prodlení 0,05 % z ceny dílčího plnění, s jehož realizací je Poskytovatel v prodlení. Výše smluvní pokuty však nepřesáhne částku odpovídající 30 % z ceny dílčího plnění.
	5. V případě prodlení Města s úhradou bezvadné faktury se sjednává úrok z prodlení ve výši 0,05 % z dlužné částky za každý započatý kalendářní den prodlení.
4. POSTUP PRO UZAVÍRÁNÍ DÍLČÍCH SMLUV
	1. Návrh na uzavření dílčí smlouvy zasílá Poskytovateli Město.
	2. Návrh dílčí smlouvy musí obsahovat alespoň: popis plnění ve smyslu Přílohy č. 1 této Smlouvy, počet hodin plnění, lhůtu, ve které má být předmět dílčí smlouvy realizován, další informace nezbytné pro plnění. Počet hodin specifikovaný v návrhu dílčí smlouvy Městem zohlední vždy příslušnou část plnění, na kterou se dílčí smlouva vztahuje, a to zejména s ohledem na: předpokládanou náročnost plnění dílčí smlouvy a jeho vazbu na ICT řešením Centrální elektronické spisovny dodaným vítězným dodavatelem – vítězem veřejné zakázky „Rozvoj služeb eGovernmentu města Český Krumlov – část 10 – Centrální elektronická spisovna“ vyhlášeného Městem.
	3. Návrh dílčí smlouvy musí být Poskytovatelem písemně potvrzen do 5 kalendářních dnů od jejího doručení Poskytovateli. Pokud Poskytovatel neakceptuje do 5 kalendářních dnů návrh dílčí smlouvy, musí být Poskytovatelem Městu zaslán ve stejné lhůtě protinávrh dílčí smlouvy k vyjádření s odůvodněním. Město buď protinávrh dílčí smlouvy písemně akceptuje, nebo písemně neakceptuje. V případě, že k akceptaci protinávrhu ze strany Města nedojde, Město může zaslat Poskytovateli nový návrh dílčí smlouvy ve smyslu ust. 4.1 této Smlouvy, nebo Město může dané plnění po Poskytovateli dále nepožadovat.
	4. Město si vyhrazuje možnost žádný návrh na uzavření dílčí smlouvy za dobu účinnosti této Smlouvy neučinit.
5. UVEŘEJNĚNÍ SMLOUVY V REGISTRU SMLUV

5.1 Poskytovatel (ICZ a.s.) tímto žádá Město, aby v souladu s ust. § 3 odst. 1 zákona č. 340/2015 Sb., o registru smluv (dále „zákon o registru smluv“) před uveřejněním smlouvy (objednávky) před uveřejněním smlouvy v registru smluv provedl znečitelnění:

* + - 1. veškerých osobních údajů obsažených ve smlouvě, a to zejména v rozsahu jméno, adresa, email, telefon, podpisy jednajících osob,

A aby, Město v souladu s ust. § 3 odst. 1 zákona o registru smluv, výše uvedené informace v registru smluv nezveřejnil.

5.2 Dále Poskytovatel žádá Město, aby v souladu s ust. § 3 odst. 2 písm. b) zákon o registru smluv nezveřejnil:

1. Přílohu č. 3 smlouvy

a další údaje nebo informace obsažené ve smlouvě (objednávce), které představují technickou předlohu, návod, výkres, projektovou dokumentaci, model, způsob výpočtu jednotkových cen, vzor a výpočet.

Pro vyloučení pochybností Poskytovatel (ICZ a.s.) uvádí, že výše uvedená omezení se vztahují také na jakékoliv jiné uveřejnění smlouvy (objednávky) jako např. uveřejnění smlouvy na webových stránkách zadavatele nebo na profilu zadavatele a poskytnutí informací při postupu podle zákona č. 106/1999 Sb., o svobodném přístupu k informacím.

1. ZÁVĚREČNÁ USTANOVENÍ
	1. Tato Smlouva může být měněna a doplňována pouze formou písemných dodatků podepsaných oběma smluvními stranami. Totéž platí pro dílčí smlouvy na základě této Smlouvy uzavřené.
	2. Smluvní strany se dohodly, že v případě že z výše uvedené veřejné zakázky „Rozvoj služeb eGovernmentu města Český Krumlov – část 10 – Centrální elektronická spisovna vyhlášené Městem nevzejde vítězný dodavatel (resp. s ním nebude uzavřena smlouva, resp. smlouvy na plnění předmětu veřejné zakázky „Rozvoj služeb eGovernmentu města Český Krumlov – část 10 – Centrální elektronická spisovna“ pozbyde tato Smlouva účinnosti.
	3. Město prohlašuje a potvrzuje, že uzavřením této smlouvy není nijak dotčena možnost účasti Poskytovatele (ICZ a.s.) v budoucích výběrových řízeních na obdobný předmět plnění vypsaných Městem.
	4. Tato Smlouva i dílčí smlouvy na základě této Smlouvy uzavřené se řídí právem České republiky. Tato Smlouva je vyhotovena ve třech originálech, z nichž Poskytovatel obdrží jeden originál a Město originály dva. Pokud oddělitelné ustanovení této Smlouvy je nebo se stane neplatným či nevynutitelným, nemá to vliv na platnost zbývajících ustanovení této Smlouvy. V takovém případě se smluvní strany zavazují uzavřít do 20 pracovních dnů od výzvy druhé ze stran této Smlouvy dodatek k této Smlouvě nahrazující oddělitelné ustanovení této Smlouvy, které je neplatné či nevynutitelné, platným a vynutitelným ustanovením odpovídajícím hospodářskému účelu takto nahrazovaného ustanovení.
	5. Smluvní strany po přečtení této Smlouvy prohlašují, že souhlasí s jejím obsahem, že tato Smlouva byla sepsána vážně, určitě, srozumitelně a na základě jejich pravé a svobodné vůle, na důkaz čehož připojují své podpisy.
	6. Uzavřením této Smlouvy se nahrazují veškerá případná předchozí právní jednání Smluvních stran ve věci Poskytovatelem dodávané integrace eSSL e-spis a užívaného Městem s ICT řešením Centrální elektronické spisovny dodaným vítězným dodavatelem – vítězem veřejné zakázky „Rozvoj služeb eGovernmentu města Český Krumlov – část 10 – Centrální elektronická spisovna“ vyhlášeného Městem.
	7. Tato smlouva nabývá účinnosti dnem uveřejnění v registru smluv. Uveřejnění v registru smluv provede Město a neprodleně o tom informuje Poskytovatele.
	8. Nedílnou součást Smlouvy tvoří tyto přílohy:

Příloha č. 1: Specifikace předmětu plnění

Příloha č. 2: Harmonogram

Příloha č. 3: Popis aplikačního rozhraní

|  |  |
| --- | --- |
| PoskytovatelV XXX dne \_\_\_\_\_\_\_\_\_\_\_\_\_ | MěstoV Českém Krumlově dne \_\_\_\_\_\_\_\_\_\_\_\_\_ |
| .........................................................................Za PoskytovateleJméno a příjmení podepisujícího | .........................................................................Město Český KrumlovMgr. Dalibor Carda, starosta |

**Příloha č. 1 Smlouvy o součinnosti
- Specifikace předmětu plnění**

Předmětem plnění (integrace) je napojení systému ICZ e-spis® na systém centrální elektronické spisovny s využitím služeb NS API systému ICZ e-spis®. Popis aplikačního rozhraní je uveden v Příloze č.3 Smlouvy.

Součástí předmětného plnění nebudou programové úpravy systému ICZ e-spis® a poskytnutých rozhraní na dodávaný systém Spisovny, v rámci nabízeného řešení není součástí migrace dat ze stávajících systémů.

Místem výkonu všech činností je sídlo Poskytovatele (ICZ a.s.):

 **Na hřebenech II 1718, 140 00 Praha 4-Nusle**

Jednotlivé práce pracovníků Poskytovatele (ICZ a.s.) budou probíhat:

Ve všední dny v čase mezi **9:00-15:00**

Poskytovatel je povinen Městu reagovat na jeho dotazy do **5 pracovních dnů.**

**Příloha č. 2 Smlouvy o součinnosti
- Harmonogram**

|  |  |
| --- | --- |
| **Fáze projektu** | **Délka trvání fáze (v týdnech)** |
| Zahájení projektu (podpis smlouvy) | **T** |
| Úvodní schůzka s dodavatelem | **T+1** |
| Vystavení rozhraní, poskytnutí dokumentací | **T+4** |
| Konzultace s dodavatelem | **T+6** |
| Nasazení testovacího prostředí | **T+7** |
| Nasazení produktivního prostředí | **T+8** |

Harmonogram předpokládá, že Město bude postupovat při realizaci tohoto projektu ve vzájemné důvěře a maximální součinnosti s Poskytovatelem (ICZ a.s.) se vstřícným vztahem k řešení vyskytnuvších se problémů a součinnost k jednotlivým výše uvedeným bodům poskytne do 2 pracovních dní.

Pro vybudování integračního můstku je potřeba:

* Vyplnění a předání konfiguračního dotazníku do 1 pracovního týdne od převzetí pro následnou konfiguraci spisové služby pro komunikaci s Portálem občana.
* Při analýze současného stavu integrace a při pracích vyplývajících z nestandardně provedených stávajících integrací (např. chybějící identifikace již zaevidovaných dokumentů v ISSD, apod.).
* Zajištění dispozice a ověření aktuálnosti verze ISSD dodavatele, která musí být kompatibilní s ICZ e-spis® v příslušné verzi provozované u zákazníka a to vč. podpory NSESSS komunikace
* Zajištění vzdáleného přístupu k testovací instanci ICZ e-spis ®.
* Sdělení e-mailových adres, na které budou zasílány k řešení notifikace o dávkách na rozhraní (např. Nejsou žádné příchozí dávky, Chyba v dávce)
* Zaslání testovacích dávek do ISSD a řešené chyb zjištěných v dávkách
* Plné otestování funkcionality ze strany integrovaného ISSD vůči rozhraní a včetně komunikace mezi ICZ e-spis® a ISSD.

**Příloha č. 3 – Popis aplikačního rozhraní**

**Obecné informace**

## Koncepce napojení spisové služby na digitální spisovnu

Digitální spisovna je obvykle integrována se systémy produkujícími dokumenty k uložení. Jedná se zejména o spisové služby, agendové systémy a systémy pro správu dokumentů. Tyto systémy komunikují s digitální spisovnou prostřednictvím dále popsaného aplikačního rozhraní. Rozhraní poskytuje externímu systému přístup k funkcím pro ukládání a výdej dokumentů a informace o skartačních řízeních.

Rozhraní obsahuje mimo jiné následující základní funkce:

* vstup balíčku (metadat a obsahu) určeného k uložení do spisovny ve formě SIP,
* vrácení stavu zpracování zaslaných balíčků,
* výdej metadat a obsahu balíčku ve formě DIP,
* vrácení seznamu skartačních řízení, jejich stavu a seznamu zařazených balíčků.

### Uložení obsahu do spisovny

Spisová služba nebo další zdrojový systém musí zajistit přípravu SIP balíčku a jeho odeslání do digitální spisovny. Spisová služba řídí procesu předávání do digitální spisovny včetně řešení chyb během přenosu a záznamu výsledku přenosu na základě informací poskytovaných digitální spisovnou.

Digitální spisovna akceptuje pouze stanovené datové formáty komponent dokumentů. Proto je nutné, aby datový formát komponent u dokumentů v digitální podobě předávaných do spisovny byl ve výstupním datovém formátu dle vyhlášky č. 191/2009 Sb., popř. v dalších schválených datových formátech. Spisová služba před předáním spisů/dokumentů do spisovny musí provádět případný převod do výstupního formátu (např. u doručených dokumentů ve formátu PDF jiné verze než PDF/A). Původní komponenty nebo komponenty analogového dokumentu vzniklé při digitalizaci mohou být předávány i v jiném formátu. Převod a popis vztahu komponent zajišťuje aplikace spisové služby před předání do spisovny.

Příprava a předání SIP:

* Uzavřené spisy nebo vyřízené dokumenty lze předávat do spisovny přímo nebo v případě evidence ukládacích jednotek je uživatel nejprve zařadí do ukládacích jednotek a ty předá do spisovny. Impuls pro zahájení přenosu do spisovny vychází zpravidla od uživatele volbou příkazu či nastavením příznaku.
* Spisová služba musí rozpoznat objekty určené k uložení do spisovny. Je vhodné, aby s těmito objekty nebylo možné dále manipulovat.
* Spisová služba vygeneruje SIP balíček podle NSESS a prostřednictvím rozhraní DESA předá balíčky do spisovny. Spisová služba si uloží identifikátor verze balíčku přidělený spisovnou . *Tuto funkcionalitu zajišťuje samostatně funkce "submitpackage" rozhraní REST nebo funkce rozhrani SOAP - AsyncSubmitPackageStart a AsyncSubmitPackageEnd s předáním obsahu do vstupního adresáře (např. přes CIFS nebo FTP).*

Dokončení příjmu - kontrola uložení:

* Elektronická spisová služba se dotazuje na stav zpracování předaných balíčků. Digitální spisovna poskytuje službu aplikačního rozhraní pro získání seznamu změněných balíčků. Na základě těchto údajů lze provést aktualizaci údajů o stavu příjmu do spisovny ve spisové službě. *Tuto funkcionalitu zajišťují funkce rozhrani getPackageChanges a getPackageStatus .*
* U spisů/dokumentů, které prošly vstupní kontrolou a byly úspěšně uloženy, je možné provést ve spisové službě případné odstranění komponent a ponechání pouze hlaviček metadat.
* V případě neúspěšného předání (díky chybám) je nutné opravit chybu ve spisové službě na základě informace získané z digitální spisovny. Může se jednat o nepovolený datový formát, chybu v klasifikaci apod. Opravené objekty spisová služby opětovně předá do spisovny. *Tuto funkcionalitu zajišťují funkce rozhrani AsyncSubmitPackageStart a AsyncSubmitPackageEnd, s tím, že je plně nahrazena původní odeslaná podoba předaných dat novou verzí.*

### Získání uloženého obsahu ze spisovny

Kromě uživatelského rozhraní, spisovna poskytuje aplikační rozhraní pro získání uloženého spisu/dokumentu ve formě DIP.

* Spisová služba zprostředkuje výdej dokumentu/spisu uloženém v digitální spisovně. Pokud uživatel potřebuje získat dokument, musí příslušný dokument nejprve vyhledat ve spisové službě. Spisová služba na základě zadaného dotazu vyhledá odpovídající dokumenty, ověří přístup k nim pro přihlášeného uživatele a zobrazí seznam nalezených dokumentů.
* U nalezeného dokumentu/spisu si uživatel vyžádá obsah z uživatelského prostředí spisové služby. Spisová služba přes aplikační rozhraní elektronické spisovny vyžádá DIP balíček odpovídající požadavku uživatele na základě uloženého identifikátoru balíčku. *Tuto funkcionalitu zajišťují funkce rozhrani requestDIP a getDIPContent.*
* Spisová služba zprostředkuje zobrazení metadat a obsahu poskytnutého spisovnou.

### Zaznamenání výsledku skartačního řízení

Z digitální spisovny jsou dokumenty vyřazeny na základě skartačního řízení. Informaci o vyřazení je vhodné zpětně zaznamenat do elektronické spisové služby.

* Elektronická spisová služba se dotazuje na seznam balíčků, které byly skartovány nebo archivovány.
* Digitální spisovna poskytne seznam skartovaných a archivovaných balíčků a údajů o jejich vyřazení. Na základě těchto údajů lze provést aktualizaci údajů ve spisové službě. . *Tuto funkcionalitu zajišťuje funkce rozhrani getRDProcessPackages.*

### Využívání číselníků

V elektronické spisovně jsou využívány některé číselníky shodné se spisovou službou. Je vhodné, aby spisová služba prováděla automaticky aktualizaci základních číselníků, pokud dojde ke změně. Jedná se zejména o číselníky:

* spisový plán – číselník věcných skupiny
* skartační režimy
* uložení – fyzické spisovny

*Tuto funkcionalitu zajišťují funkce rozhraní uploadNomenclature a updateNomenclature*.

Jmenné prostory

V příkladech budu používat tyto prefixy pro jmenné prostory (namespace) :

|  |  |  |
| --- | --- | --- |
| **Prefix**  | **Namespace**  | **Popis**  |
| mets | http://www.loc.gov/METS/ | elementy obálky balíčku standardu METS  |
| premis | http://www.loc.gov/standards/premis/ | elementy technických metadat standardu PREMIS |
| nsesss | http://www.mvcr.cz/nsesss/v1  | formát národního standardu pro předávání do archivu v.1 |
| nsesss2 | http://www.mvcr.cz/nsesss/v2  | formát národního standardu pro předávání do archivu v.2 |
| nsext | <http://desa.i.cz/nsext/> | formát doplňující schéma národního standardu pro účely předávání dat do spisovny – zejména rozšíření o ukládací jednotky a definici podoby dokumentu. |
| Nsext2 | <http://desa.i.cz/nsext/>v2 | formát doplňující schéma národního standardu v.2 pro účely předávání dat do spisovny – zejména rozšíření o ukládací jednotky a definici podoby dokumentu. |

## Systém předávání a identifikace balíčků SIP

Rozhraní pro příjem balíčků SIP systému DESA umožňuje přijímat odděleně jednotlivé spisy, dokumenty, součásti a díly.

Systém DESA nyní umožňuje předávání struktury spisů a dokumentů dvěma způsoby:

a) Pro každou z těchto entit je na vstupní rozhraní DESA předáván samostatný balíček SIP. Vzájemný vztah evidován prostřednictvím odkazů na obsažené entity (např. dokumenty ve spisu).

Do systému DESA je nutné předávat balíčky obsahující vzájemné vazby v tomto pořadí: „dokumenty“, „díly“, „součásti“, „spisy“. pro zajištění kontroly vazeb a jejich správnou evidenci v provozní databázi (viz.. kapitola Posloupnost ukládání balíčků).

b) Je předán kompletní spis v jednom balíčku SIP. Tento balíček musí být ve struktuře METS definované standardem NSESSS 2 . Tyto balíčky SIP na příjmu rozdělí systém DESA tak, že každý dokument i spis je reprezentován samostatnými balíčky AIP.

Po předání do systému DESA je vrácen unikátní generovaný kód pro konkrétní verzi předaného balíčku SIP. Tento identifikátor slouží k následným operacím a sledování průběhu uložení balíčku do systému. V systému DESA je možné aktualizovat již uložené balíčky, např. přidávat dokumenty do otevřených spisů. V tomto případě je vrácen volající aplikaci nový kód balíčku, ale interně je původní verze nahrazena verzí novou na základě jednoznačného identifikátoru (spisová značka). Generované kódy balíčku by měly být uloženy v rámci volající aplikace u příslušných entit tak, aby bylo možné zajistit volání dalších funkcí aplikačního rozhraní DESA pro daný balíček.

## Formát balíčku SIP

Balíčky SIP jsou předávány jako množina souborů uložených v archivu komprimovaném metodou „zip“. Každý balíček obsahuje jeden soubor „mets.xml“. Dále jsou obsaženy v případě dokumentů všechny datové soubory, na něž odkazují metadata v sekci „fileSec“ souboru mets.xml. Datové soubory mohou být předávány ve formátech, které jsou povoleny v rámci konfigurace pro daného původce (nastavení číselníku formátů).

Názvy souborů ve formátu ZIP by měly používat kódování UTF8. Pokud není balíček předán uvedeným způsobem a obsahuje diakritiku, může dojít při příjmu k nesouladu názvu souboru s metadaty a s tím související chybě v obsahu balíčku.

Soubor „mets.xml“ obsahuje metadata balíčku ve formátu METS (Metadata Encoding and Transmission Standard).

Strukturu metadat v souboru METS znázorňuje následující schéma:



### Datový obsah – fileSec

Odkaz na datový obsah je umístěn v sekci „fileSec“. Každý dokument i jednotlivé komponenty mohou mít v závislosti na své původní podobě i následně provedených migracích několik podob téhož obsahu. Tyto **podoby jsou uloženy v oddělených sekcích „fileGrp“.** Logická struktura dokumentu (uložená ve „structMap“) přitom obsahuje jednotlivě pro každou logickou komponentu (pokud jsou definovány) odkaz na veškeré reprezentace téhož obsahu ve více sekcích („fileGrp“). Podstatné je **použití atributu USE u každé skupiny „fileGrp“.**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Akronym reprezentace****(Hodnota parametru USE* *pro reprezentaci)* | ***Název reprezentace*** | ***Poznámka*** | ***Způsob validace*** *podle PRONOM při příjmu* |
| Original  | **Obsah ve výstupním formátu**  | Jedná se o obsah v prvotní digitální podobě (digital born) nebo obsah konvertovaný do výstupního formátu | Musí odpovídat definovaným výstupním formátům a pro každý původem digitální nebo konvertovaný dokument **musí být přítomen pro každou komponentu dokumentu.** |
| Input | **Obsah v původním digitálním formátu** | Obsah v původním digitálním formátu před konverzí do výstupního formátu | Nemusí odpovídat definovaným výstupním formátům. **Reprezentace nemusí být přítomna, případně může být přítomna pro jednotlivé komponenty.** |
| Digitized | **Digitalizovaný obsah** | Digitalizovaný analogový dokument (nikoli konvertovaný) | Nemusí odpovídat definovaným výstupním formátům. **Reprezentace nemusí být přítomna, případně může být přítomna pro jednotlivé komponenty.** U původem digitálních dokumentů by neměla být přítomna. |
| Preview | **Obsah v náhledovém formátu** | Obsah po konverzi do náhledového formátu. | Nemusí odpovídat definovaným výstupním formátům. Použití náhledového formátu může být definováno v implementačním projektu DESA. V balíčku SIP se reprezentace může vyskytovat, ale **není na vstupu vyžadována**. |
| Migrated | **Obsah migrovaný do nového formátu** | Obsah migrovaný do nového formátu za účelem dlouhodobého zajištění zpřístupnění obsahu.  | Nemusí odpovídat definovaným výstupním formátům. Vzniká typicky na základě definovaných migračních procedur v rámci spisovny/archivu. V balíčku SIP se reprezentace může vyskytovat, ale **není na vstupu vyžadována**.. |

**V rámci vstupu komponent se validují tato pravidla:**

Pravidla pro metadata:

* Dokument typu *Digitalni* a *KonvertovanyDigitalni* musí mít u všech komponent jednu podobu v sekci „Original“.
* Dokument typu *KonvertovanyAnalogovy* musí mít u všech komponent jednu podobu v sekci „Original“.
* Spis typu *KonvertovanyAnalogovy* musí mít u všech dokumentů podobu *KonvertovanyAnalogovy*.
* Spis typu *Digitalni* a *KonvertovanyDigitalni* musí mít u všech dokumentů podobu *Digitalni* nebo *KonvertovanyDigitalni* .
* Díl typu *Digitalni* a *KonvertovanyDigitalni* musí mít u všech dokumentů podobu *Digitalni* nebo *KonvertovanyDigitalni* .
* Díl typu *KonvertovanyAnalogovy* musí mít u všech dokumentů podobu *KonvertovanyAnalogovy*.

Pravidla pro formát souboru:

* Všechny soubory v sekci „Original“ musí být v povoleném výstupním formátu (standardně detekce formátu knihovnou DROID).
* Soubory v ostatních sekcích jsou pouze rozpoznávány, nemusí mít povolený výstupní formát.

**příklad předávané sekce fileSec a příslušné structMap:**

- <!--

 technická metadata obsažených komponent - zejména původní název souboru a kontrolní součet na jeho binární podobou

  -->

[-](file:///C%3A%5CDATA%5Cprojekty%5CDES_projekt%5Ctesty%5Csipy_priklady%5Ckomplet_spis_v2%5CREC_SP2010_1035_2%5Cmets.xml##) <mets:fileSec>

- <!--

 digitální originály nebo autorizované konverze

  -->

[-](file:///C%3A%5CDATA%5Cprojekty%5CDES_projekt%5Ctesty%5Csipy_priklady%5Ckomplet_spis_v2%5CREC_SP2010_1035_2%5Cmets.xml##) <mets:fileGrp VERSDATE="2010-10-07T12:17:38.915+02:00" USE="Original" ID="GRP\_0001">

[-](file:///C%3A%5CDATA%5Cprojekty%5CDES_projekt%5Ctesty%5Csipy_priklady%5Ckomplet_spis_v2%5CREC_SP2010_1035_2%5Cmets.xml##) <mets:file CHECKSUMTYPE="SHA-1" ID="FILE\_0001" CHECKSUM="DPhCCTVNqCNZZjQqXTNpq2N4CmI=" SIZE="1468713" CREATED="2010-10-07T12:11:26.671+02:00">

  <mets:FLocat LOCTYPE="URN" xlin:href="Dokumentace\_schematu\_ess.pdf" xmlns:xlin="http://www.w3.org/1999/xlink" />

  </mets:file>

[-](file:///C%3A%5CDATA%5Cprojekty%5CDES_projekt%5Ctesty%5Csipy_priklady%5Ckomplet_spis_v2%5CREC_SP2010_1035_2%5Cmets.xml##) <mets:file CHECKSUMTYPE="SHA-1" ID="FILE\_0002" CHECKSUM="KtbwUd5tc2dpXfmctB4tZQNPbT4=" SIZE="70731" CREATED="2010-10-07T12:15:00.380+02:00">

  <mets:FLocat LOCTYPE="URN" xlin:href="planek\_budov.jpg" xmlns:xlin="http://www.w3.org/1999/xlink" />

  </mets:file>

[-](file:///C%3A%5CDATA%5Cprojekty%5CDES_projekt%5Ctesty%5Csipy_priklady%5Ckomplet_spis_v2%5CREC_SP2010_1035_2%5Cmets.xml##) <mets:file CHECKSUMTYPE="SHA-1" ID="FILE\_0003" CHECKSUM="dADBMHF8ae0PcIPAv5FAJzkOK+k=" SIZE="22939" CREATED="2010-10-07T12:15:50.631+02:00">

  <mets:FLocat LOCTYPE="URN" xlin:href="planek\_ortofoto\_m.jpg" xmlns:xlin="http://www.w3.org/1999/xlink" />

  </mets:file>

  </mets:fileGrp>

- <!--

 neautorizovaná digitální podoba - scan

  -->

[-](file:///C%3A%5CDATA%5Cprojekty%5CDES_projekt%5Ctesty%5Csipy_priklady%5Ckomplet_spis_v2%5CREC_SP2010_1035_2%5Cmets.xml##) <mets:fileGrp VERSDATE="2010-10-07T12:17:38.915+02:00" USE="Digitized" ID="GRP\_0002">

[-](file:///C%3A%5CDATA%5Cprojekty%5CDES_projekt%5Ctesty%5Csipy_priklady%5Ckomplet_spis_v2%5CREC_SP2010_1035_2%5Cmets.xml##) <mets:file CHECKSUMTYPE="SHA-1" ID="FILE\_0004" CHECKSUM="DPhCCTVNqCNZZjQqXTNpq2N4CmI=" SIZE="1468713" CREATED="2010-10-07T12:11:26.671+02:00">

  <mets:FLocat LOCTYPE="URN" xlin:href="Dokumentace\_schematu\_ess.pdf" xmlns:xlin="http://www.w3.org/1999/xlink" />

  </mets:file>

  </mets:fileGrp>

- <!--

 původní zdrojová (pracovní) podoba dokumentu

  -->

[-](file:///C%3A%5CDATA%5Cprojekty%5CDES_projekt%5Ctesty%5Csipy_priklady%5Ckomplet_spis_v2%5CREC_SP2010_1035_2%5Cmets.xml##) <mets:fileGrp VERSDATE="2010-10-07T12:17:38.915+02:00" USE="Input" ID="GRP\_0003">

[-](file:///C%3A%5CDATA%5Cprojekty%5CDES_projekt%5Ctesty%5Csipy_priklady%5Ckomplet_spis_v2%5CREC_SP2010_1035_2%5Cmets.xml##) <mets:file CHECKSUMTYPE="SHA-1" ID="FILE\_0005" CHECKSUM="kdhfdfdsqCN=ZjQqXdfpq2N4Cmty" SIZE="1332093" CREATED="2010-10-03T12:11:26.671+02:00">

  <mets:FLocat LOCTYPE="URN" xlin:href="Dokumentace\_schematu\_ess.doc" xmlns:xlin="http://www.w3.org/1999/xlink" />

  </mets:file>

  </mets:fileGrp>

  </mets:fileSec>

- <!--

 strukturální metadata HYBRIDNÍHO dokumentu - dokument (RECORD) a odkaz na metadata dokumentu v nsess

  -->

[-](file:///C%3A%5CDATA%5Cprojekty%5CDES_projekt%5Ctesty%5Csipy_priklady%5Ckomplet_spis_v2%5CREC_SP2010_1035_2%5Cmets.xml##) <!--

 variantně může obsahovat komponenty (DOCUMENT), zde existují 2 - text smlouvy a příloha.

 Příloha přitom obsahuje 2 komponenty - obrázky.

  -->

[-](file:///C%3A%5CDATA%5Cprojekty%5CDES_projekt%5Ctesty%5Csipy_priklady%5Ckomplet_spis_v2%5CREC_SP2010_1035_2%5Cmets.xml##) <mets:structMap TYPE="LOGICAL">

[-](file:///C%3A%5CDATA%5Cprojekty%5CDES_projekt%5Ctesty%5Csipy_priklady%5Ckomplet_spis_v2%5CREC_SP2010_1035_2%5Cmets.xml##) <mets:div DMDID="DM\_0001" LABEL="Smlouva o pronájmu" ORDER="1" TYPE="record">

- <!--

 digitální dokument ve výstupním formátu, současné obsahuje odkaz na dokument ve zdrojovém formátu FILE\_0005(nevaliduje se)

  -->

[-](file:///C%3A%5CDATA%5Cprojekty%5CDES_projekt%5Ctesty%5Csipy_priklady%5Ckomplet_spis_v2%5CREC_SP2010_1035_2%5Cmets.xml##) <mets:div ORDER="1" LABEL="text smlouvy" TYPE="document">

  <mets:fptr FILEID="FILE\_0001" />

  <mets:fptr FILEID="FILE\_0005" />

  </mets:div>

- <!--

 digitální příloha ve dvou souborech tvořících jednu komponentu, např. dvě strany přílohy

  -->

[-](file:///C%3A%5CDATA%5Cprojekty%5CDES_projekt%5Ctesty%5Csipy_priklady%5Ckomplet_spis_v2%5CREC_SP2010_1035_2%5Cmets.xml##) <mets:div ORDER="2" LABEL="příloha 1 - plánek budovy" TYPE="document">

  <mets:fptr FILEID="FILE\_0002" />

  <mets:fptr FILEID="FILE\_0003" />

  </mets:div>

- <!--

 analogová příloha - neobsahuje buď žádný digitální obsah, nebo pouze digitalizovanou neautorizovanou kopii (v tomto případě)

  -->

[-](file:///C%3A%5CDATA%5Cprojekty%5CDES_projekt%5Ctesty%5Csipy_priklady%5Ckomplet_spis_v2%5CREC_SP2010_1035_2%5Cmets.xml##) <mets:div ORDER="3" LABEL="příloha 2 - výpis z katastru" TYPE="document">

  <mets:fptr FILEID="FILE\_0004" />

  </mets:div>

  </mets:div>

  </mets:structMap>

  </mets:mets>

### Strukturální metadata - structMap

V této kapitole je popsána struktura balíčku SIP systému DESA, pokud není předáván přímo ve formě kompletního spisu nebo dokumentu ve struktuře definované standardem NSESSS 2.

Strukturální metadata se vyskytují v dokumentovém i spisovém SIP. Hodnota atributu TYPE u kořenového elementu <div> určuje typ balíčku. Může nabývat těchto hodnot:

* “record” = dokument
* “file” = spis
* “part” = část spisu
* “volume” = díl spisu

Balíček spisu zpravidla ve strukturálních metadatech obsahuje odkazy na obsažené balíčky dokumentů. Tyto odkazy jsou předávány prostřednictvím obsaženého elementu "div", který musí obsahovat element "mptr" s odkazem na dokument. Element "mptr" pro definici vazby mezi spisem a obsaženými dokumenty musí být definován následovně:

**LOCTYPE** = "other"

**OTHERLOCTYPE**="ERMS\_ID" (pro odkaz identifikátorem spisové služby) nebo "internal\_reference" (pro odkaz číslem jednacím)

**xlink:href**=*"identifikátor spisové služby nebo číslo jednací"* (vlastní identifikátor odkazu)

Následují příklady metadat pro jednotlivé typy balíčků SIP.

***Dokumentový (record) SIP*** *se dvěma soubory:*

 <mets:structMap TYPE="LOGICAL">

 <mets:div DMDID="MD\_0001" LABEL="Dokument 1" TYPE="record" ORDER="1">

 <mets:div ORDER="1">

 <mets:fptr FILEID="FILE\_0001"/>

 <mets:fptr FILEID="FILE\_0002"/>

 </mets:div>

 </mets:div>

 </mets:structMap>

***Dokumentový (record) SIP******se dvěma částmi*** *(document) a třemi soubory:*

 <mets:structMap TYPE="LOGICAL">

 <mets:div DMDID="MD\_0001" LABEL="Dokument 2" TYPE="record" ORDER="1">

 <mets:div LABEL="Hlavní dokument" TYPE="document" ORDER="1">

 <mets:fptr FILEID="FILE\_0001"/>

 <mets:fptr FILEID="FILE\_0002"/>

 </mets:div>

 <mets:div LABEL="Příloha 1" TYPE="document" ORDER="2">

 <mets:fptr FILEID="FILE\_0003"/>

 </mets:div>

 </mets:div>

 </mets:structMap>

***Spisový (file) SIP*** *s odkazem na dva dokumentové (record) SIPy. Odkaz je představován jednoznačnou položkou popisných metadat u dokumentového SIP v rámci původce (číslo jednací).. V okamžiku přípravy SIP není k dispozici identifikátor přidělovaný systémem DESA, ten je doplněn při transformaci na AIP jako další položka „mptr“ :*

 <mets:structMap TYPE="LOGICAL">

 <mets:div DMDID="MD\_0001" LABEL="Spis 1" TYPE="file" ORDER="1">

 <mets:div ORDER="1" TYPE="record" LABEL="Dokument 1">

 <METS:mptr LOCTYPE="OTHER" OTHERLOCTYPE="internal\_reference" xlink:href="DOC1">

 </mets:div>

 <mets:div ORDER="2" TYPE="record" LABEL="Dokument 2">

 <METS:mptr LOCTYPE="OTHER" OTHERLOCTYPE="internal\_reference" xlink:href="DOC2">

 </mets:div>

 </mets:div>

 </mets:structMap>

### Popisná metadata

Popisná metadata se předávají v sekci dmdSec struktury METS. Základním formátem popisných metadat je předávací formát NSESS (lze použít metadata ve formátu NSESSS v.1 nebo NSESSS v.2). Balíček přitom může obsahovat i jiné typy popisných metadat. Příslušný kořenový element musí být prostřednictvím interního identifikátoru odkazován v rámci strukturálních metadat (viz. předchozí kapitola).

#### Specifická popisná metadata DESA

V rámci NSESS používá DESA specifická popisná metadata ve volitelné **sekci   <nsesss:JineUdaje> v rámci <nsesss:EvidencniUdaje>** příslušné sekce popisných metadat dokumentu nebo spisu. Tato metedata doplňují standardní údaje předávacího formátu, jsou používána pro účely správy spisovny a nejsou primárně určena pro přenos do Národního Archivu. Tato metadata jsou definována jmenným prostorem <http://desa.i.cz/nsext/>, který je obsažen v příloze (desa\_nsext.xsd).



**Podoba obsahu dokumentu** (PodobaObsahu) slouží k určení původu/zpodobnění dokumentu (ve vztahu digitální/listinný) a může nabývat následujících hodnot: Typ podoby obsahu (PodobaObsahu) slouží k určení podoby/zpodobnění dokumentu (ve vztahu digitální/listinný) a může nabývat následujících hodnot:

|  |  |
| --- | --- |
| *Typ podoby dokumentu* | *Komentář* |
| Digitalni | Původem digitální dokument nebo spis – „Digital Born“. Musí mít povinně kompletní digitální obsah ve výstupním formátu. |
| KonvertovanyDigitalni | Digitální dokument nebo spis, který má evidovánu i konvertovanou analogovou podobu. Musí mít povinně kompletní digitální obsah ve výstupním formátu. |
| Analogovy | Analogový dokument nebo spis, může mít volitelně digitalizovanou podobu v SIP, která nemá status originálu a není určena k předání do archivu. |
| KonvertovanyAnalogovy | Analogový dokument nebo spis s kompletní digitalizovanou podobu autorizovanou konverzí ve výstupním formátu.  |
| HybridniDA | Analogový dokument nebo spis, který má některé části v digitální podobě (původem digitální nebo konvertované) – jde například o některé digitální dokumenty v analogovém spisu nebo některé digitální komponenty v analogovém dokumentu.  |

Zařazení dokumentu podle podoby (digitální, analogová) bude součástí rozšíření předávacího formátu ve schématu nsext, kde budou umístěna pro spis nebo dokument metadata o ukládací jednotce a umístění v sekci JineUdaje. Může být uvedeno pro spis i dokument. Pokud není přítomna entita “EvidencniUdajeDESA”, považuje se podoba za digitální.

Příklad předání podoby dokumentu:

*<nsesss:JineUdaje>*

 *<nsext:EvidencniUdajeExtDESA xmlns:nsext="http://desa.i.cz/nsext/">*

 *<nsext:PodobaObsahu>*

 *<nsext:Typ>HybridniDA</nsext:Typ>*

 *<nsext:Komentar>Dokument s elektronickou přílohou</nsext:Komentar>*

 *</nsext: PodobaObsahu >*

**Umístění** může být evidováno pro spis, díl nebo dokument. Může obsahovat identifikaci a parametry spisovny a ukládací jednotky. Pro identifikátory se při importu ze spisové služby použije hodnota atributu *zdroj="ERMS".* Na základě identifikátorů ERMS a kódu jsou v rámci DESA verifikovány, resp. zakládány a aktualizovany číselník spisoven a ukládacích jednotek v rámci původce.

Schéma metadat k umístění a ukládací jednotce v okamžiku umístění do spisovny:



**Rozšíření možnosti deklarace věcné skupiny** - Vzhledem k tomu, že podle národního standardu může mít (resp. měl by mít) každý archivní objekt vlastní unikátní spisový znak zahrnující nejen položku spisového plánu, ale i identifikaci vlastního objektu, je v rámci rozšiřujícího schématu přidána položka *Klasifikace/VecnaSkupina* , která určuje přiřazení objektu (spis, dokument, díl, součást) k číselníku spisového plánu.

Systém DESA přitom nevyžaduje s každým dokumentem předávat plnou hiearchi metadat spisového plánu (MaterskeEntity). Proto **systém DESA umožňuje na vstupu tyto 3 varianty deklarace věcné skupiny**:

1. Pokud je (plně podle standardu) uvedena **hiearchie entit v entitě Trideni/MaterskeEntity** včetně položky spisového plánu, použije se plně určený spisový znak z mateřských entit.
2. Pokud není uvedena hiearchie mateřských entit, použije se **"PlneUrcenySpisovyZnak" položky *Klasifikace/VecnaSkupina* ze schématu NSEXT.** *Úplnost hiearchie mateřských entit v předávacím formátu přitom záměrně na příjmu nevalidujeme (= možnost zjednodušeného předání klasifikace bez rozsáhlých duplicit).*
3. Pokud není uvedena žádná hodnota pro věcnou skupinu (viz. předchozí body) a archivní objekt (dokument, spis) nemá vlastní specifický spisová znak (nýbrž tento odpovídá položce spisového plánu), použije se pro mapování na položku spisového plánu **plně určený spisový znak ze schématu NSESS –element *Trideni/PlneUrcenySpisovyZnak***).

###  Technická metadata

Technická metadata k předávaným digitálním objektům (komponentám) jsou ve struktuře standardu PREMIS (<http://www.loc.gov/standards/premis/>). Příslušný kořenový element je prostřednictvím interního identifikátoru odkazován v rámci sekce fileSec struktury METS.

Předávání metadat ve formátu PREMIS není povinnou součástí balíčku SIP, tato metadata jsou v tom případě založena modulem vstupu systému DESA. Pokud jsou metadata PREMIS objektů (komponent) součástí SIPu, jsou následně při příjmu doplněna, zejména o určení formátu souboru.

## Posloupnost ukládání balíčků a vzájemné závislosti entit

Pro ukládání balíčků platí následující **pravidla pro vzájemné vztahy předávaných hierarchicky uspořádaných entit**:

* Do systému se předávají uzavřené dokumenty v samostatných balíčcích -„dokumentový SIP“ v definované posloupnosti s balíčky „dílu“, „součásti“ nebo „spisu“ – spisové SIP, které v rámci struktury METS odkazují na již předané související entity. V případě seskupení obsahujících dokumenty se tedy předávají odkazy pouze na již uzavřené dokumenty.
* Pokud je dokument zařazen do „dílu“, „součásti“ a „spisu“, je nutné **po** odeslání dokumentu do systému DESA odeslat i balíček SIP s metadaty této nadřazené entity. Tímto je dokument evidován ve správné struktuře a bude zajištěna správná funkcionalita skartačního řízení z hlediska kontroly závislostí.
* Je nutné zajistit aktualizací některých změn provedených ve zdrojové aplikaci a souvisejících s entitami uloženými v DESA. Aktualizace bude provedena na základě zaslaného aktuálního balíčeku s metadaty příslušné entity do DESA:
	+ Pokud budou změněna metadata spisu jehož součásti či dokumenty jsou již uzavřeny a uloženy v DESA. Pozn. odeslání SIP balíčku spisu při změně metadat je nutné v případě, že bude požadováno udržování aktuálních metadat o spisu v DESA.
	+ Při uzavření spisu jehož součásti či dokumenty jsou již uzavřeny a uloženy v DESA.
	+ Pokud nastane změna v zařazení (např. přetříděním) příslušné entity (spisu, součásti, dílu, dokumentu).
	+ Pokud je uzavřené seskupení nebo uzavřený dokument (či více dokumentů současně) vyřazeno z „dílu“, „součásti“  nebo  „spisu“,  je nutné do spisovny odeslat aktuální balíček SIP s metadaty této entity.
* V rámci DESA jsou balíčky a jejich aktualizace ve standardní konfiguraci párovány na základě čísla jednacího a spisové značky.  Tyto identifikátory nelze ve zdrojovém systému měnit po odeslání metadat spisu do DESA.
* V rámci skartačního řízení se budou validovat vzájemné vazby dokumentů a spisů s ohledem na jejich skartační znaky a lhůty.

**Řešení příslušných chybových stavů během příjmu do DESA:**

* Pokud v rámci příjmu balíčku nebo vstupní kontroly balíčku dokumentu v systému DESA nastane chyba , není přijat ani nadřazený balíček  „dílu“, „součásti“ a „spisu“ a je vrácena příslušná chyba pro tyto nadřazené balíčky – ve volajícím systému je možné tuto chybu v rámci hierarchie zobrazit. Ostatní balíčky dokumentů obsažených ve spisu, které prošly kontrolou se ukládají do archivního úložiště.
* Pokud je chybu v příjmu balíčku „dokumentu“ v rámci „dílu“, „součásti“  nebo  „spisu“ nutné řešit na úrovni volající aplikace (nejde o technickou chybu řešitelnou na úrovni administrace DESA), je nutné po zaslání balíčku „dokumentu“ zaslat opakovaně i balíček „dílu“, „součásti“  nebo  „spisu“.  Pozn. stejně jako u prvotního vložení nezáleží ani u tohoto opravného předání na tom, zda se jedná o otevřený nebo uzavřený spis.

Pro ukládání balíčků platí následující **pravidla pro předávání křížových odkazů.** Tento postup se použije i pro předávání  hierarchicky strukturovaných spisů(daňové spisy, priorace spisů):

* Souvislosti mezi entitami (spisy, díly, součástmi a dokumenty) ukládanými do DESA je možné předávat v rámci metadat v struktuře NSESS (evidenční údaje – souvislosti – křížový odkaz).
* Souvislosti se evidují zejména pro účely validace v rámci skartačního řízení.
* Lze předávat křížové odkazy i na otevřené spisy.
* V případě odkazu z otevřeného spisu, dílu nebo součásti na obsažený uzavřený dokument (resp. díl, součást) by měl být do DESA předán balíček s metadaty spisu, dílu nebo součásti  obsahující tyto odkazy, aby byly tato vazba registrována pro účely skartačního řízení, viz. dále.
* Pokud nastane změna v křížových odkazech příslušné entity (spisu, součásti, dílu, dokumentu), je nutné předat aktualizovaný balíček s metadaty příslušné entity do DESA.
* Souvislosti se uloží do databáze DESA a pokud je odkazovaná entita systémem již přijata, dojde ke spárování a zanesení této vazby do provozních metadat systému. Pokud odkazovaná entita dosud přijata není, dojde k párování až při jejím zařazování do skartačního řízení – jestliže je v tomto okamžiku již v DESA evidována, dojde ke spárování a zanesení této vazby do provozních metadat systému.
* Systém DESA bude blokovat skartaci a archivaci v případě existence křížového odkazu na entitu, která není zpracována v rámci stejného skartačního řízení, nebo není vůbec evidována v DESA (např. otevřený dokument v odesílající aplikaci). Verifikace odkazů (kontrola konfliktů) se provede v rámci přípravy skartační návrhu.

## Import a export číselníků

Číselníky systému DESA mohou být evidovány společně nebo odděleně pro jednotlivé původce, vždy pro určité období platnosti. Aktuální verze aplikace umožňuje import číselníků a export číselníků prostřednictvím administračního uživatelského rozhraní a export číselníků prostřednictvím API. Číselníky jsou importovány ve formátu XML.

K dispozici jsou tyto typy číselníků:

|  |  |  |
| --- | --- | --- |
| *Akronym* | *Název* | *Poznámka* |
| ACollect  | Sbírka  | Doplňkové členění dokumentů u některých původců |
| Afund  | Fond | Doplňkové členění dokumentů u některých původců |
| CreCntrl  | Řízení důvěryhodnosti dokumentu | Umožňuje odlišné zacházení s dokumenty při řízení důvěryhodnosti (např. časová razítka jen pro vybrané množiny dokumentů) |
| Loc  | Lokace  | Umístění fyzické kopie |
| Pronom  | Typy souborů | Typy povolených formátů souborů podle číselníku PRONOM |
| **RdCntrl**  | **Skartační znak a lhůta**  |  |
| **RecCl**  | **Položka klasifikace**  | **Položka spisového plánu** |
| RecType | Typ dokumentu  |  |
| LocDetail | Lokační objekt | Doplňující lokační objekty (např. regály, boxy) |
| OrgUnit | Organizační jednotka |  |

Popis XML struktury číselníků je v samostatné kapitole.

# Přehled operací

## Operace vstupního rozhraní

* Odeslání balíčku (rozhraní REST) – submitpackage
* Zjištění stavu balíčku (rozhraní REST)
* Zahájení přenosu (alternativní rozhraní SOAP) - asyncSubmitPackageStart
* Ukončení přenosu (alternativní rozhraní SOAP) - asyncSubmitPackageEnd
* Aktualizace meatadat balíčku - updatePackageMetadata
* Informace o zpracování balíčků - getPackageChanges
* Informace o stavu balíčku - getPackageStatus
* Získání obsahu číselníků – getNomenclatures
* Vytvoření procesu vyřazování (skartačního řízení) - createRDProcess
* Přidání balíčků do procesu vyřazování (skartačního řízení) - addRDProcessPackages
* Obebrání balíčků z procesu vyřazování (skartačního řízení) - removeRDProcessPackages
* Změna parametrů balíčku v rámci vyřazování - updateRDProcessPackage
* Změna stavu procesu vyřazování (skartačního řízení) - setRDProcessState
* Seznam konfliktů ve skartačním řízení - getRDProcessConflicts
* Změna umístění ukládací jednotky - changeSUnitLocation
* Nastavení parametrů ukládací jednotky - setupSUnit
* Nastavení (předdefinování) křížových odkazů - setPackageReference
* Zrušení předdefinovaných křížových odkazů - removePackageReference
* Získání seznamu křížových odkazů – getReferences
* Nastavení aktuálního zpracovatele dokumentu – setPackageProcessor
* Získání seznamu AIP obsažených v balíčku – getSIPContent
* Označení balíčků k aplikaci časového razítka – selectPackagesForTimeStamp
* Nastavit spouštěcí událost – updateExternalEvent
* Autentizovat uživatele DESA – authenticateUser (\*)

## Operace výstupního rozhraní

* Vyhledání balíčku podle identifikátoru - findPackages
* Vyžádání výstupního balíčku - requestDIP,
* Schválení výdeje DIP - resolveDIPRequest
* Vyžádání zápůjčky - requestLoan
* Schválení zápůjčky - resolveLoanRequest
* Výdej zápůjčky – deliverLoan
* Vrácení zápůjčky - returnLoan
* Vyžádání nahlížení - requestView
* Schválení nahlížení - resolveViewRequest
* Vyžádání manipulace - requestManipulation
* Zahájení manipulace - startManipulation
* Ukončení manipulace - finishManipulation
* Informace o stavu balíčku DIP – getDIPStatus
* Získání obsahu DIP – getDIPContent
* Získání seznamu DIP, zápůjček, nahlížení, manipulace - getDIPList
* Seznam skartovaných a archivovaných balíčků – getRDProcessPackages
* Stav balíčku ve skartaci – getRDProcessPackage
	1. Stav skartačního řízení – getRDProcessState
	2. Operace pro administraci
* Nahrání číselníku - uploadNomenclature
* Aktualizace číselníku - updateNomenclature
* Nahrání nebo aktualizace externího číselníku (NSESSS2) - uploadExternalNomenclature
* Aktualizace nebo založení uživatele - setupUser
* Nastavení rolí uživatele - setupUserRoles
* Nastavení povolených spisoven uživatele - setupUserLocations
* Mazání uživatele - deleteUser
* Seznam uživatelů - getUsers

V následující části jsou popsány detailně jednotlivé operace. U každé operace je zobrazena definice jejího:

* vstupu včetně povinných a volitelných elementů
* výstupu
* příklad práce s operací.

# Operace vstupního rozhraní

## Operace POST "submitpackage" rozhraní REST

Operace rozhraní je volána při předávání balíčku SIP z aplikace původce. Tato operace slouží spolu s funkcí „AsyncSubmitPackageEnd“ pro předání metadat a obsahu v balíčku SIP do spisovny.

Vlastní obsah balíčku je předáván v komprimované podobě formátu ZIP definované struktury (mets.xml+datové soubory).

Použití této metody je preferováno pro jednoduchost použití a konfigurace před předáváním prostřednictvím rozhraní SOAP s předáním obsahu souborovým protokolem (CIFS/FTP) = metody AsyncSubmitPackageStart a AsyncSubmitPackageEnd (od verze 2.15 nebude předávání souborovým protokolem podporováno).

Pro účely předávání je možné volitelně nastavit validaci kontrolního součtu, který je předáván v hlavičce.

Pro „fileHashAlg„ je možno použít hodnoty: "MD5“,“SHA-1","SHA-256","SHA-384", "SHA-512".

**Vstup:**

**metoda POST na rozhraní - "**/rest/sipsubmission/submitpackage"

povinné parametry:

* parametr userName type string - přihlašovací jméno uživatele, jehož jménem je balíček odesílán
* parametr producerCode type string - kód/zkratka původce
* parametr producerSipId - type string - označení balíčku z odesílající aplikace, používá se v procesu příjmu

nepovinné parametry:

* aipVersionUUID – předgenerovaný unikátní kód UUID pro verzi balíčku, není možné zaslat opakovaně stejnou hodnotu. Standardně se nepředává, ale je součástí opdovědi.
* parametr fileHashAlg type string - standardní kód hashovacího algoritmu
* parametr fileHash type string - hash odeslaného obsahu pro kontrolu při příjmu
* parametr fileNameEncoding - type string (default: UTF-8) - kódování názvů souborů v rámci ZIP, alternativně lze použít např. "CP437"

hlavička requestu:

* Accept-Language: cs - (cs je default, používá se pro lokalizaci chybových zpráv)

data-binary: obsah balíčku komprimovaný metodou ZIP jako obsah requestu POST

**Výstup**:

hlavička responsu:

* X-DEA-AipVersionId - vygenerované ID verze balíčku další identifikací balíčku v ostatních funkcích API

**Příklad volání (příkazový řádek utility curl):**

curl -v0 -u ws@homol:ws -H "Accept-Language: cs" --data-binary @"/tmp/test.sip" -X POST "<http://localhost:8080/dea-frontend/rest/sipsubmission/submitpackage?userName=superAdmin&producerCode=homol&producerSipId=test>"

## Operace HEAD rozhraní REST

Operace rozhraní je volána pro zjištění stavu zpracování balíčku.

**Vstup:**

**metoda HEAD na rozhraní - "**/rest/sipsubmission/{ID verze balíčku} "

URL requestu:

* {ID verze balíčku}: Unikátní identifikátor verze balíčku, vrácený metodou POST v parametru „X-DEA-AipVersionId“

povinné parametry:

* parametr userName type string - přihlašovací jméno uživatele, jehož jménem je balíček odesílán
* parametr producerCode type string - kód/zkratka původce

hlavička requestu:

* Accept-Language: cs - (cs je default, používá se pro lokalizaci chybových zpráv)

**Výstup**:

hlavička responsu:

* X-DEA-PackageStateCode: Stav zpracování balíčku SIP

**Možné stavy zpracování balíčku:**

Pole „packageStateCode“ může obsahovat jednu z  hodnot následující tabulky podle aktuálního stavu zpracování.

**Červeně uvedené** - chybové stavy, kde je nutná oprava metadat nebo binárního obsahu vstupujícího balíčku prostřednictvím aktualizace balíčku odesílající aplikací. V případě stavu AI\_INVALID je možno obsah ve vyjímečných případech upravit i v rámci administračních funkcí DESA (není určeno pro koncové uživatele).

**Oranžově uvedené -** chybové stavy, které označují interní problém při zpracování v DESA. V tomto případě se nepředpokládá žádná činnost uživatele odesílající aplikace (oprava metadat nebo obsahu). Problém je k řešení na straně administrace nebo dodavatele DESA, případně může být potřeba spolupráce s administrátorem nebo dodavatelem odesílající aplikace při řešení nestandardní situace (např. potřeba opětovného zaslání balíčků).

**Zeleně uvedené –** finální nechybový stav, balíček je uložen. V tomto okamžiku je možné smazat obsah dokumentu v odesílající aplikaci.

Tabulka stavů:

|  |  |
| --- | --- |
| **Zkratka stavu** | **Název stavu** |
| AI\_RECEIVED | Balíček přijat pro zpracování |
| AI\_DECODE\_SIP | Balíček je dekódován ze zdrojového archivního formátu |
| AI\_DECOMPOSE\_SIP | Balíček se dělí ze vstupního SIP na jednotlivé AIP (pouze v případě spisu předávaného včetně dokumentů) |
| AI\_AV\_INP\_OK | Vstupní antivirová kontrola v pořádku, čeká v karanténě |
| AI\_FORM\_OK  | Kontrola formátu balíčku v pořádku |
| AI\_MIG\_WAIT | Balíček čeká na konverzi formátů souborů (komponent). |
| **AI\_INFECTED** | **Balíček neprošel antivirovou kontrolou** |
| **AI\_QA\_ERR** | **Interní chyba při kontrole formátu nebo antivirové kontrole** |
| AI\_IC\_OK | Vstupní zpracování balíčku ukončeno |
| AI\_EDIT | Balíček s neúplným nebo chybným obsahem připraven pro úpravy obsahu archivářem / administrátorem |
| **AI\_REJECT** | **Uložení balíčku AIP zamítnuto z důvodu neopravitelných chyb v obsahu, balíček vrácen původci k opravě** |
| **AI\_INVALID** | **Uložení balíčku AIP nemožné z důvodu chyb v obsahu, balíček může být doplněn archivářem nebo vrácen původci k opravě podle nastavení systému** |
| AI\_DM\_OK | Uložení metadat do provozního systému  |
| AI\_AS\_REP | Čeká na opakovaný pokus ukládání do archivního úložiště |
| **AI\_AS\_OK** | **Uložení do archivního úložiště v pořádku** |
| AI\_ACC\_REP | Čeká na opakovaný pokus o uložení do přístupového modulu |
| **AI\_ACC\_OK** | **Uložení do přístupového modulu v pořádku, balíček zpracován** |
| **AI\_ERROR** | **Interní chyba při zpracování nebo ukládání** |

**Příklad volání (příkazový řádek utility curl):**

curl -v0 -u ws@homol:ws -H "Accept-Language: cs" -X HEAD "http://localhost:8080/dea-frontend/rest/sipsubmission/4a988c0f-9004-44f8-bb8e-6d9b499459aa?userName=superAdmin&producerCode=homol"

## Operace AsyncSubmitPackageStart (nebude podporováno)

Operace rozhraní je volána při předávání balíčku SIP z aplikace původce. Tato operace slouží spolu s funkcí „AsyncSubmitPackageEnd“ pro předání metadat a obsahu v balíčku SIP do spisovny.

Vlastní obsah balíčku je předáván v komprimované podobě formátu ZIP definované struktury (mets.xml+datové soubory). Předání obsahu SIP probíhá na definované umístění ve vstupním diskovém prostoru systému DESA. Balíčky se předávají do **příslušného podadresáře totožného s přihlašovacím jménem předávajícího uživatele v základním adresáři** (podle konfigurace DESA) **a pod jménem odvozeným od návratové hodnoty "idSIPVersion" funkce AsyncSubmitPackageStart takto: "*sip\_root*/*login*/*idSipVersion*.sip".**

Předání probíhá libovolným protokolem pro přenos souborů podle konkrétní implementace (např. FTP, CIFS). Objem dat balíčku SIP je limitován použitou infrastrukturou a nezávisí na vlastnostech protokolu SOAP. Kořenový adresář pro předávání balíčků je definován v konfiguraci DESA.

Místo použití této metody je preferováno rozhraní REST.

**Vstup:**

**asyncSubmitPackage**

* producerID - type int *nebo* producerCode type string
* userLogin - type string
* producerSIPID - type string
* fileSize - type int
* fileCheckSum - type string

**Výstup**:

**asyncSubmitPackageStartResponse**

* idSIPVersion type string
* producerSIPID type string

**Příklad:**

**Vstup:**

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">

 <S:Body>

 <ns2:asyncSubmitPackageStart xmlns:ns2="http://i.cz/dea/schemas/SIPSubmission/types">

 <producerCode>Puvodce1</producerCode>

 <userLogin>admin</userLogin>

 <producerSIPID>D123</producerSIPID>

 <fileSize>6532</fileSize>

 <fileCheckSum>quASPc94TKLwsNqzNwzuN1epElc=</fileCheckSum>

 </ns2:asyncSubmitPackageStart>

 </S:Body>

</S:Envelope>

**Výstup (v případě úspěšného zahájení přenosu):**

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

 <soap:Body>

 <ns2:asyncSubmitPackageStartResponse xmlns:ns2="http://i.cz/dea/schemas/SIPSubmission/types">

 <idSIPVersion>4721787c-3301-4cdb-8b26-8cc90e0b08dc</idSIPVersion>

 <producerSIPID> D123</producerSIPID>

 </ns2:asyncSubmitPackageStartResponse>

 </soap:Body>

</soap:Envelope>

## Operace AsyncSubmitPackageEnd (nebude podporováno)

Operace rozhraní je volána po předání obsahu balíčku SIP z aplikace původce. Tato operace slouží spolu s funkcí „AsyncSubmitPackageStart“ pro předání metadat a obsahu v balíčku SIP do spisovny. Musí být volána vždy po ukončení přenosu dat do vstupního adresáře spisovny, jinak není příjem balíčku dokončen, a odeslaný obsah není považován za přijatý.

Místo použití této metody je preferováno rozhraní REST.

**Vstup:**

**asyncSubmitPackageEnd**

* producerID - type int *nebo* producerCode type string
* userLogin - type string
* idSIPVersion type string

**Výstup**:

**asyncSubmitPackageEndResponse**

* retCode type string

**Příklad:**

**Vstup:**

 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">

 <S:Body>

 <ns2:asyncSubmitPackageEnd xmlns:ns2="http://i.cz/dea/schemas/SIPSubmission/types">

 <producerCode>Puvodce1</producerCode>

 <userLogin>admin</userLogin>

 <idSIPVersion>97736b56-7ce7-4daa-89b1-8bd003ae54a5</idSIPVersion>

 </ns2:asyncSubmitPackageEnd>

 </S:Body>

 </S:Envelope>

**Výstup (v případě úspěšného ukončení přenosu):**

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

 <ns2:asyncSubmitPackageEndResponse xmlns:ns2="http://i.cz/dea/schemas/SIPSubmission/types">

 <retCode>OK</retCode>

 </ns2:asyncSubmitPackageEndResponse>

</soap:Body>

</soap:Envelope>

## Operace getPackageChanges

Operace vrací seznam balíčků SIP, u nichž se změnil stav příjmu a zpracování v intervalu počínajícím zadaným časem. Tato funkce slouží zejména k periodickému ověřování stavu zpracování balíčků v rámci systému spisovny, např. pro ověření uložení balíčku do archivního úložiště.

**Vstup:**

**getPackageChanges**

* producerID - type int *nebo* producerCode type string
* userLogin - type string
* startByTime - type dateTime

**Výstup**:

**getPackageChangesResponse**

* retCode type string
* changeList type ArrayPackageChanges
	+ item - optional, unbounded; type PackageChanges

idSIPVersion - type string

producerSIPID - type string

**Příklad:**

**Vstup:**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

 <soapenv:Header/>

 <soapenv:Body>

 <typ:getPackageChanges xmlns:typ="http://i.cz/dea/schemas/SIPSubmission/types">

 <producerID>5</producerID>

 <userLogin>admin</userLogin>

 <startByTime>2008-10-29T12:01:05</startByTime>

 </typ:getPackageChanges>

 </soapenv:Body>

</soapenv:Envelope>

**Výstup:**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

 <soapenv:Header/>

 <soapenv:Body>

 <typ:getPackageChangesResponse xmlns:typ="http://i.cz/dea/schemas/SIPSubmission/types">

 <changeList>

 <item>

 <idSIPVersion>4721787c-3301-4cdb-8b26-8cc90e0b08dc</idSIPVersion>

 <producerSIPID>D123</producerSIPID>

 </item>

 <item>

 <idSIPVersion>97736b56-7ce7-4daa-89b1-8bd003ae54a5</idSIPVersion>

 <producerSIPID>D456</producerSIPID>

 </item>

 </changeList>

 </typ:getPackageChangesResponse>

 </soapenv:Body>

</soapenv:Envelope>

## Operace getPackageStatus

Operace vrací aktuální stav zpracování daného balíčku SIP.

**Vstup:**

**getPackageStatus**

* producerID - type int *nebo* producerCode type string
* userLogin - type string
* idSIPVersion - type string

**Výstup**:

**getPackageStatusResponse**

* + idSIPVersion - type string
	+ producerSIPID - type string
	+ packageStateCode - type string
	+ packageStateText - type string item - optional, unbounded; type PackageChanges

**Příklad:**

**Vstup:**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

 <soapenv:Header/>

 <soapenv:Body>

 <typ:getPackageStatus xmlns:typ="http://i.cz/dea/schemas/SIPSubmission/types">

 <producerID>5</producerID>

 <userLogin>admin</userLogin>

 <idSIPVersion>4721787c-3301-4cdb-8b26-8cc90e0b08dc</idSIPVersion>

 </typ:getPackageStatus>

 </soapenv:Body>

</soapenv:Envelope>

**Výstup:**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

 <soapenv:Header/>

 <soapenv:Body>

 <typ:getPackageStatusResponse xmlns:typ="http://i.cz/dea/schemas/SIPSubmission/types">

 <idSIPVersion>4721787c-3301-4cdb-8b26-8cc90e0b08dc</idSIPVersion>

 <producerSIPID>D123</producerSIPID>

 < packageStateCode > AI\_AS\_OK </packageStateCode >

 < packageStateText > Uložení do archivního úložiště v pořádku </packageStateText >

 </typ:getPackageStatusResponse>

 </soapenv:Body>

</soapenv:Envelope>

**Možné stavy zpracování balíčku:**

Pole „packageStateCode“ může obsahovat jednu z  hodnot následující tabulky podle aktuálního stavu zpracování.

**Červeně uvedené** - chybové stavy, kde je nutná oprava metadat nebo binárního obsahu vstupujícího balíčku prostřednictvím aktualizace balíčku odesílající aplikací. V případě stavu AI\_INVALID je možno obsah ve vyjímečných případech upravit i v rámci administračních funkcí DESA (není určeno pro koncové uživatele).

**Oranžově uvedené -** chybové stavy, které označují interní problém při zpracování v DESA. V tomto případě se nepředpokládá žádná činnost uživatele odesílající aplikace (oprava metadat nebo obsahu). Problém je k řešení na straně administrace nebo dodavatele DESA, případně může být potřeba spolupráce s administrátorem nebo dodavatelem odesílající aplikace při řešení nestandardní situace (např. potřeba opětovného zaslání balíčků).

**Zeleně uvedené –** finální nechybový stav, balíček je uložen. V tomto okamžiku je možné smazat obsah dokumentu v odesílající aplikaci.

Tabulka stavů:

|  |  |
| --- | --- |
| **Zkratka stavu** | **Název stavu** |
| AI\_RECEIVED | Balíček přijat pro zpracování |
| AI\_AV\_INP\_OK | Vstupní antivirová kontrola v pořádku, čeká v karanténě |
| AI\_FORM\_OK | Kontrola formátu balíčku v pořádku |
| **AI\_INFECTED** | **Balíček neprošel antivirovou kontrolou** |
| **AI\_QA\_ERR** | **Interní chyba při kontrole formátu nebo antivirové kontrole** |
| AI\_IC\_OK | Vstupní zpracování balíčku ukončeno |
| AI\_EDIT | Balíček s neúplným nebo chybným obsahem připraven pro úpravy obsahu archivářem / administrátorem |
| **AI\_REJECT** | **Uložení balíčku AIP zamítnuto z důvodu neopravitelných chyb v obsahu, balíček vrácen původci k opravě** |
| **AI\_INVALID** | **Uložení balíčku AIP nemožné z důvodu chyb v obsahu, balíček může být doplněn archivářem nebo vrácen původci k opravě podle nastavení systému** |
| AI\_DM\_OK | Uložení metadat do provozního systému  |
| AI\_AS\_REP | Čeká na opakovaný pokus ukládání do archivního úložiště |
| **AI\_AS\_OK** | **Uložení do archivního úložiště v pořádku** |
| AI\_ACC\_REP | Čeká na opakovaný pokus o uložení do přístupového modulu |
| **AI\_ACC\_OK** | **Uložení do přístupového modulu v pořádku, balíček zpracován** |
| **AI\_ERROR** | **Interní chyba při zpracování nebo ukládání** |

## Operace updatePackageMetadata

Operace slouží k aktualizaci popisných metadat balíčku prostřednictvím zaslané šablony XSLT. Předaná šablona může provádět pouze modifikaci metadat umístěných v sekci dmdSec původního souboru mets.xml.

Aktualizovaný balíček je uložen jako nová verze s tím, že původní binární obsah není duplikovaně uložen.

**Vstup:**

**updatePackageMetadata**

* producerID - type int *nebo* producerCode type string
* userLogin - type string
* producerSIPID- type string = uživatelské označení balíčku
* idUpdateSIPVersion - type string = identifikátor balíčku před aktualizací
* xsl - type base64Binary = šablona pro transformaci metadat

**Výstup**:

**updatePackageMetadataResponse**

* idSIPVersion - type string = identifikátor aktualizovaného balíčku
* producerSIPID- type string = uživatelské označení balíčku

**Příklad:**

Nahrazují se kompletní popisná metadata v elementu nsesss2:Dokument.

**Vstup:**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:typ="http://i.cz/dea/schemas/SIPSubmission/types">

 <soapenv:Header/>

 <soapenv:Body>

 <typ:updatePackageMetadata>

 <producerCode> Puvodce1</producerCode>

 <userLogin> admin </userLogin>

 <producerSIPID> D123</producerSIPID>

 <idUpdateSIPVersion>4721787c-3301-4cdb-8b26-8cc90e0b08dc</idUpdateSIPVersion>

 <xsl>

***zakódovaná aktualizace XSLT ve formě base64:***

*<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:nsesss2="http://www.mvcr.cz/nsesss/v2" version="1.0">*

 *<xsl:template match="/nsesss2:Dokument">*

 *<nsesss2:Dokument ID="RECORD\_MCP1ESaf85d4">*

 *.......*

 *</nsesss2:Dokument>*

 *</xsl:template>*

*</xsl:stylesheet>*

 </xsl>

 </typ:updatePackageMetadata>

 </soapenv:Body>

</soapenv:Envelope>

**Výstup:**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:typ="http://i.cz/dea/schemas/SIPSubmission/types">

 <soapenv:Header/>

 <soapenv:Body>

 <typ:updatePackageMetadataResponse>

 <idSIPVersion>ed21787c-3301-4cdb-8b26-8cc90e0b0815 </idSIPVersion>

 <producerSIPID> D123</producerSIPID>

 </typ:updatePackageMetadataResponse>

 </soapenv:Body>

</soapenv:Envelope>

## Operace getNomenclatures

Operace rozhraní je volána po předání číselníků evidovaných v rámci daného původce k danému datu v systému spisovny volající aplikaci. Slouží jako zdroj hodnot pro správné plnění obsahu vstupních balíčků SIP.

Číselníky jsou předávány ve formátu XML souboru, který obsahuje jednotlivé tabulky, jejich záznamy a hodnoty polí tabulky (viz. popis číselníků).

Pokud není uveden v požadavku seznam číselníků (nomenclatureList), vrací se všechny číselníky, které jsou k dispozici.

**Vstup:**

**getNomenclatures**

* producerID - type int *nebo* producerCode type string
* userLogin type string
* nomenlatureList type NomenclatureListType
	+ nomenclatureAcronyme - optional, unbounded; type string
* date type date

**Výstup**:

**getNomenclaturesResponse**

* nomenclaturesData type base64Binary

**Příklad:**

**Vstup:**

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">

 <S:Header/>

 <S:Body>

 <typ:getNomenclatures xmlns:typ="http://i.cz/dea/schemas/SIPSubmission/types">

 <producerID>5</producerID>

 <userLogin>admin</userLogin>

 <nomenlatureList>

 <!—akronymy číselníků -->

 <nomenclatureAcronyme>RdCntrl</nomenclatureAcronyme>

 </nomenlatureList>

 <date>2007-11-21</date>

 </typ:getNomenclatures>

 </S:Body>

</S:Envelope>

**Výstup:**

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

 <ns2: getNomenclaturesResponse xmlns:ns2="http://i.cz/dea/schemas/SIPSubmission/types">

 <nomenclaturesData >

PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4NCjxnZW46bm9t

ZW5jbGF0dXJlcyB4bWxuczpnZW49Imh0dHA6Ly9kZWEuaS5jei9ub21lbi94bWxi

ZWFucy9nZW4iPg0KICAgDQogIDxnZW46cmVjLWNscz4NCiAgICA8Z2VuOnRpdGxl

PlBvbG+ea2Ega2xhc2lmaWthY2U8L2dlbjp0aXRsZT4NCiAgICA8Z2VuOmRlc2Ny

PktsYXNpZmlrYWNlIGRva3VtZW50dSBwb2RsZSBzcGlzb3bpaG8gcGzhbnU8L2dl

bjpkZXNjcj4NCiAgICA8Z2VuOnJlYy1jbD4NCiAgICAgIDxnZW46ZnVsbHlRY2M+

MS4yMy4zNDwvZ2VuOmZ1bGx5UWNjPg0KICAgICAgPGdlbjp0aXRsZT5O4WplbW7t

IHNtbG91dnk8L2dlbjp0aXRsZT4NCiAgICAgIDxnZW46ZGVzY3I+VpplY2hueSBz

bWxvdXZ5IHT9a2Fq7WPtIHNlIHByb27ham11PC9nZW46ZGVzY3I+DQogICAgICA8

Z2VuOmFnPkExNTwvZ2VuOmFnPg0KICAgICAgPGdlbjpjbGFzc0NvZGU+MzQ8L2dl

bjpjbGFzc0NvZGU+DQogICAgICA8Z2VuOmNyZWNudHJsPmNyZS1jbnRybHMtMDwv

Z2VuOmNyZWNudHJsPg0KICAgICAgPGdlbjpnZW4+UzEwPC9nZW46Z2VuPg0KICAg

ICAgPGdlbjprZXl3PnNtbG91dnk8L2dlbjprZXl3Pg0KICAgICAgPGdlbjpsb2M+

Vk9EMS0xMjMtMTwvZ2VuOmxvYz4NCiAgICAgIDxnZW46YWNsPnB1YmxpY19hbGw8

L2dlbjphY2w+DQogICAgPC9nZW46cmVjLWNsPg0KICAgIDxnZW46cmVjLWNsPg0K

ICAgICAgPGdlbjpmdWxseVFjYz4xLjI1PC9nZW46ZnVsbHlRY2M+DQogICAgICA8

Z2VuOnRpdGxlPkZha3R1cnk8L2dlbjp0aXRsZT4NCiAgICAgIDxnZW46ZGVzY3I+

VnlkYW7pIGZha3R1cnkgb3JnYW5pemFjZTwvZ2VuOmRlc2NyPg0KICAgICAgPGdl

bjphZz5TMTA8L2dlbjphZz4NCiAgICAgIDxnZW46Y2xhc3NDb2RlPjI1PC9nZW46

Y2xhc3NDb2RlPg0KICAgICAgPGdlbjpjcmVjbnRybD5jcmUtY250cmxzLTE8L2dl

bjpjcmVjbnRybD4NCiAgICAgIDxnZW46Z2VuPkExNTwvZ2VuOmdlbj4NCiAgICAg

IDxnZW46a2V5dz5mYWt0dXJ5PC9nZW46a2V5dz4NCiAgICAgIDxnZW46bG9jPlZP

RDEtMTIzLTI8L2dlbjpsb2M+DQogICAgICA8Z2VuOmFjbD5wdWJsaWNfYWxsPC9n

ZW46YWNsPg0KICAgIDwvZ2VuOnJlYy1jbD4NCiAgPC9nZW46cmVjLWNscz4gIA0K

ICANCjwvZ2VuOm5vbWVuY2xhdHVyZXM+

</nomenclaturesData>

 </ns2: getNomenclaturesResponse >

</soap:Body>

</soap:Envelope>

**Výstup – dekódovaný obsah položky „nomenclaturesData“:**

<?xml version="1.0" encoding="UTF-8"?>

<gen:nomenclatures xmlns:gen="http://dea.i.cz/nomen/xmlbeans/gen">

 <gen:rec-cls>

 <gen:title>Položka klasifikace</gen:title>

 <gen:descr>Klasifikace dokumentu podle spisového plánu</gen:descr>

 <gen:rec-cl>

 <gen:fullyQcc>1.23.34</gen:fullyQcc>

 <gen:title>Nájemní smlouvy</gen:title>

 <gen:descr>Všechny smlouvy týkající se pronájmu</gen:descr>

 <gen:ag>A15</gen:ag>

 <gen:classCode>34</gen:classCode>

 <gen:crecntrl>cre-cntrls-0</gen:crecntrl>

 <gen:gen>S10</gen:gen>

 <gen:keyw>smlouvy</gen:keyw>

 <gen:loc>VOD1-123-1</gen:loc>

 <gen:acl>public\_all</gen:acl>

 </gen:rec-cl>

 <gen:rec-cl>

 <gen:fullyQcc>1.25</gen:fullyQcc>

 <gen:title>Faktury</gen:title>

 <gen:descr>Vydané faktury organizace</gen:descr>

 <gen:ag>S10</gen:ag>

 <gen:classCode>25</gen:classCode>

 <gen:crecntrl>cre-cntrls-1</gen:crecntrl>

 <gen:gen>A15</gen:gen>

 <gen:keyw>faktury</gen:keyw>

 <gen:loc>VOD1-123-2</gen:loc>

 <gen:acl>public\_all</gen:acl>

 </gen:rec-cl>

 </gen:rec-cls>

</gen:nomenclatures>

## Operace createRDProcess

Operace rozhraní je volána pro založení procesu vyřazování, pokud je vyřazování řízeno externí aplikací. Veškeré další operace v rámci tohoto procesu se projeví v systému DESA stejně, jako když je vyřazování spravováno uživatelem přímo prostřednictvím uživatelského rozhraní.

Proces vyřazování je nutno označit unikátním označením. Lze evidovat současně více těchto procesů, každý má nezávislé stavy. Do procesu vyřazování je možné zařazovat balíčky splňující určená pravidla (metodami addRDProcessPackages, removeRDProcessPackages). V jednom čase nelze daný balíček zařadit do více procesů vyřazování současně. V určitých případech může být během času balíček zařazen do více procesů, ale postupně v jiném čase (např. storno skartačního řízení nebo provedená archivace bez mazání ve vlastním úložišti a následné interní vyřazení balíčku v jiném procesu).

Stavy celého procesu (jako schvalování, storno) se řídí metodou setRDProcessState.

**Vstup:**

**CreateRDProcessRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string = login pod kterým je volána služba
* acronym - type string = unikátní označení procesu
* type - type RDProcessType = typ procesu vyřazování
* determinationDate - type date (nutné pro skartační řízení) = rozhodné datum pro vyřazování (z hlediska skartační lhůty)
* deleteDiscarded - type boolean (volitelně) = po schválení skartačního řízení vymazat vyřazené balíčky z vlastního úložiště (standardně zapnuto)
* deleteRemoved - type boolean (volitelně) = po schválení skartačního řízení vymazat archivované balíčky z vlastního úložiště (standardně vypnuto)

Typy procesu vyřazování:

|  |  |
| --- | --- |
| **Zkratka v parametru** | **Popis** |
| RDT\_RD | Skartační řízení - skartace a archivace vůči příslušnému archivu. |
| RDT\_INT | Interní vyřazování - proces výmazu balíčků, které již v minulosti prošly skartačním řízením, ale kopie byla ponechána v úložišti. Není předmětem komunikace s příslušným archivem. |

**Výstup**:

**CreateRDProcessResponse**

* status - type string

Možné návratové kódy - status:

|  |  |
| --- | --- |
| **Zkratka v parametru** | **Popis** |
| OK  | Proces vyřazování založen. |
| ERR\_DUPLICATE\_ACRONYME | Duplicitní označení, proces nezaložen. |

## Operace addRDProcessPackages

Operace rozhraní je volána pro přidání balíčků do procesu vyřazování. V některých stavech již nelze s balíčky v procesu vyřazování manipulovat - např. schválené nebo strornované skartační řízení.

**Vstup:**

**AddRDProcessPackagesRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string = login pod kterým je volána služba
* acronym - type string = unikátní označení procesu vyřazování
* idSIPVersions - type ArraySIPVersion = seznam přidávaných balíčků
	+ - idSIPVersion - optional, unbounded; type string = identifikátor verze balíčku při předání do DESA

**Výstup**:

**CreateRDProcessResponse**

* status - type string
* failedSIPVersions - type ArrayFailedSIPVersion = seznam balíčků, které se nepodařilo přidat
	+ - idSIPVersion - type string = identifikátor verze balíčku při předání do DESA
		- code - type string = kód chyby při přidávání balíčku

Možné návratové kódy - parametr "status":

|  |  |
| --- | --- |
| **Zkratka v parametru** | **Popis** |
| OK  | Přidání provedeno. |
| ERR\_RD\_PROCESS\_NOT\_EXIST | Proces vyřazovaní s daným označením neexistuje. |
| ERR\_RD\_PROCESS\_INCORRECT\_STATE | Proces vyřazovaní s daným označením v aktuálním stavu neumožňuje manipulovat s obsaženými balíčky. |

Možné návratové kódy - parametr "code" pro jednotlivé balíčky:

|  |  |
| --- | --- |
| **Zkratka v parametru** | **Popis** |
| ADD\_RD\_OTHER\_PROCESS | Balíček je zařazen do jiného procesu vyřazování. |
| ADD\_RD\_BAD\_PERIOD | Skartační lhůta balíčku nevyhovuje pro zařazení do skartačního řízení. |
| ADD\_RD\_BAD\_STATE | Stav balíčku nevyhovuje zařazení do skartačního řízení |

## Operace removeRDProcessPackages

Operace rozhraní je volána pro odebrání balíčků z procesu vyřazování. V některých stavech již nelze s balíčky v procesu vyřazování manipulovat - např. schválené nebo strornované skartační řízení.

**Vstup:**

**RemoveRDProcessPackagesRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string = login pod kterým je volána služba
* acronym - type string = unikátní označení procesu vyřazování
* idSIPVersions - type ArraySIPVersion = seznam odebíraných balíčků
	+ - idSIPVersion - optional, unbounded; type string = identifikátor verze balíčku při předání do DESA

**Výstup**:

**RemoveRDProcessPackagesResponse**

* status - type string
* failedSIPVersions - type ArrayFailedSIPVersion = seznam balíčků, které se nepodařilo přidat
	+ - idSIPVersion - type string = identifikátor verze balíčku při předání do DESA
		- code - type string = kód chyby při odebírání balíčku

Možné návratové kódy - parametr "status":

|  |  |
| --- | --- |
| **Zkratka v parametru** | **Popis** |
| OK  | Přidání provedeno. |
| ERR\_RD\_PROCESS\_NOT\_EXIST | Proces vyřazovaní s daným označením neexistuje. |
| ERR\_RD\_PROCESS\_INCORRECT\_STATE | Proces vyřazovaní s daným označením v aktuálním stavu neumožňuje manipulovat s obsaženými balíčky. |

Možné návratové kódy - parametr "code" pro jednotlivé balíčky:

|  |  |
| --- | --- |
| **Zkratka v parametru** | **Popis** |
| REMOVE\_RD\_NOT\_EXISTS | Vyřazovaný balíček není zařazen v zadaném procesu vyřazování. |

## Operace updateRDProcessPackage

Operace rozhraní je volána pro nastavení parametrů vyřazování balíčku v rámci skartačního řízení. Umožňuje nastavit nový skartační režim nebo skartační lhůtu, případně označit skartaci balíčku za odloženou.

Odložit vyřazení má smysl pouze s nastavením nové delší skartační lhůty.

**Vstup:**

**UpdateRDProcessPackageRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string = login pod kterým je volána služba
* acronym - type string = unikátní označení procesu vyřazování
* idSIPVersion - type string = identifikátor verze balíčku při předání do DESA
* packageAction - type RDAipProcessAction = nový skartační režim balíčku
* discardPeriod - type int = nová skartační lhůta balíčku v letech

Typy nového skartačního režimu:

|  |  |
| --- | --- |
| **Zkratka v parametru** | **Popis** |
| RD\_ARCHIVE | Archivovat (A). |
| RD\_DISCARD | Skartovat (S). |
| RD\_DECIDE | Rozhodnout o vyřazení (V). |
| RD\_DELAYED | Odložit vyřazení. |

**Výstup**:

**UpdateRDProcessPackageResponse**

* status - type string

Možné návratové kódy - parametr "status":

|  |  |
| --- | --- |
| **Zkratka v parametru** | **Popis** |
| OK  | Změna provedena. |
| ERR\_RD\_PROCESS\_NOT\_EXIST | Proces vyřazovaní s daným označením neexistuje. |
| ERR\_RD\_PROCESS\_INCORRECT\_STATE | Proces vyřazovaní s daným označením v aktuálním stavu neumožňuje manipulovat s obsaženými balíčky. |
| ERR\_RD\_AIP\_INCORRECT\_PERIOD | Nově nastavená lhůta nesmí být krátší než původně nastavená skartační lhůta. |

## Operace updateRDProcessState

Operace rozhraní je volána pro změnu stavu vyřazování. Změna stavu zahrnuje i spuštění navrženého procesu skartace a je nevratná !

Před provedením operace se v případě přechodu do stavu schvalování nebo akceptace procesu automaticky provede kontrola na předdefinované konflikty, které mohou nastat pro jednotlivé balíčky zahrnuté do skartačního řízení. V případě, že dojde ke zjištění konfliktů (např. nesouhlasné skartační lhůty pro spis a dokument, existující zápůjčka dokumentu atp.), změna stavu se neprovede a zjištěné konflikty je možné získat metodou "getRDProcessConflicts".

Stavy je nutné nastavovat postupně v pořadí RDP\_SELECT , RDP\_REVIEW, RDP\_ACCEPED. Neschválený proces je možné stornovat.

**Vstup:**

**UpdateRDProcessStateRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string = login pod kterým je volána služba
* acronym - type string = unikátní označení procesu vyřazování
* processState - type RDProcessState = nový stav procesu vyřazování

Stavy vyřazování, které je možné nastavit:

|  |  |
| --- | --- |
| **Zkratka v parametru** | **Popis** |
| RDP\_SELECT | Příprava procesu vyřazování. Toto je stav po založení procesu. V tomto stavu lze s balíčky v rámci procesu libovolně manipulovat. |
| RDP\_REVIEW | Skartační řízení ve stavu schvalování. V tomto stavu lze s balíčky v rámci procesu manipulovat, ale před přechodem do stavu schvalování se provede kontrola na konflikty pro zařazené balíčky. |
| RDP\_ACCEPTED | Vyřazování je schváleno. V tomto stavu nelze s balíčky v rámci procesu manipulovat. Nastavení tohoto stavu vede ke spuštění vyřazování zařazených balíčků a automatickému přechodu do dalšího stavu - RDP\_PROCEED. Schválené vyřazování nelze stornovat. Nastavení tohoto stavu je nevratné ! |
| RDP\_CANCEL | Skartační řízení je stornováno. V tomto stavu nelze s balíčky v rámci procesu manipulovat. Po stornování jsou balíčky uvolněny z tohoto procesu vyřazování a je možno s nimi zacházet jako by nikdy nebyly předmětem tohoto procesu vyřazování. |

**Výstup**:

**UpdateRDProcessStateResponse**

* status - type string

Možné návratové kódy - parametr "status":

|  |  |
| --- | --- |
| **Zkratka v parametru** | **Popis** |
| OK | Změna provedena. |
| ERR\_RD\_PROCESS\_NOT\_EXIST | Proces vyřazovaní s daným označením neexistuje. |
| ERR\_RD\_PROCESS\_INCORRECT\_STATE | Proces vyřazovaní s daným označením v aktuálním stavu neumožňuje nastavit nový uvedený stav. |
| ERR\_RD\_PROCESS\_PENDING\_CONFLICTS | Existují konflikty pro zařazené balíčky, zabraňující nastavení nového stavu procesu vyřazování. |

## Operace getRDProcessConflicts

Operace rozhraní je volána pro zjištění konfliktů u zařazených balíčků, které zabraňují provedení procesu vyřazování. Je možné ji volat po neúspěšném nastavení nového stavu procesu vyřazování nebo kdykoli v průběhu přípravy procesu vyřazování pro informování uživatele o existujících problémech. Volání operace iniciuje nové vygenerování seznamu konfliktů k aktuálnímu času.

**Vstup:**

**getRDProcessConflictsRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string = login pod kterým je volána služba
* acronym - type string = unikátní označení procesu vyřazování

Stavy vyřazování, které je možné nastavit:

|  |  |
| --- | --- |
| **Zkratka v parametru** | **Popis** |
| RDP\_SELECT | Příprava procesu vyřazování. Toto je stav po založení procesu. V tomto stavu lze s balíčky v rámci procesu libovolně manipulovat. |
| RDP\_REVIEW | Skartační řízení ve stavu schvalování. V tomto stavu lze s balíčky v rámci procesu manipulovat, ale před přechodem do stavu schvalování se provede kontrola na konflikty pro zařazené balíčky. |
| RDP\_ACCEPTED | Vyřazování je schváleno. V tomto stavu nelze s balíčky v rámci procesu manipulovat. Nastavení tohoto stavu vede ke spuštění vyřazování zařazených balíčků a automatickému přechodu do dalšího stavu - RDP\_PROCEED. Schválené vyřazování nelze stornovat. Nastavení tohoto stavu je nevratné ! |
| RDP\_CANCEL | Skartační řízení je stornováno. V tomto stavu nelze s balíčky v rámci procesu manipulovat. Po stornování jsou balíčky uvolněny z tohoto procesu vyřazování a je možno s nimi zacházet jako by nikdy nebyly předmětem tohoto procesu vyřazování. |

**Výstup**:

**getRDProcessConflictsResponse**

* status - type string
* conflicts - type ArrayRDProcessConflicts = seznam balíčků s konfliktem
	+ idSIPVersion - type string = identifikace balíčku
	+ code - type RDConflictCode = kód konfliktu

Možné návratové kódy - parametr "status":

|  |  |
| --- | --- |
| **Zkratka v parametru** | **Popis** |
| OK | Změna provedena. |
| ERR\_RD\_PROCESS\_NOT\_EXIST | Proces vyřazovaní s daným označením neexistuje. |

Možné kódy konfliktů před provedením vyřazování - parametr "code":

C\_ACTION\_V = Skartační znak není A nebo S

C\_BORROWED = Dokument je zapůjčen

C\_CONTENT\_DELAYED = Některé obsažené dokumenty nebo součásti mají odloženou skartaci

C\_CONTENT\_RDPROC = Všechny obsažené dokumenty nebo součásti nejsou předmětem aktuálního skartačního řízení

C\_PARTS\_ACTION = Obsahuje součásti s odlišným skartačním znakem

C\_PARTS\_PERIOD = Obsahuje součásti s delší skartační lhůtou

C\_PART\_ACTION = Skartační znak objektu se liší od součásti

C\_PART\_PERIOD = Skartační lhůta objektu je kratší než skartační lhůta součásti

C\_RECS\_ACTION = Obsahuje dokumenty s odlišným skartačním znakem

C\_RECS\_PERIOD = Obsahuje dokumenty s delší skartační lhůtou

C\_REF\_PERIOD = Objekt je odkazován prostřednictvím křížového odkazu ve spisu nebo dokumentu s delší skartační lhůtou

C\_REF\_RDPROC = Objekt je odkazován prostřednictvím křížového odkazu ve spisu nebo dokumentu, který není předmětem aktuálního skartačního řízení

C\_SUBFILE\_ACTION = Skartační znak objektu se liší od spisu

C\_SUBFILE\_PERIOD = Skartační lhůta objektu je kratší než skartační lhůta spisu

C\_VOLS\_ACTION = Obsahuje díly s odlišným skartačním znakem

C\_VOLS\_PERIOD = Obsahuje díly s delší skartační lhůtou

C\_VOLUME\_ACTION = Skartační znak objektu se liší od dílu

C\_VOLUME\_PERIOD = Skartační lhůta objektu je kratší než skartační lhůta dílu

## Operace changeSUnitLocation

Operace rozhraní je volána pro nastavení umístění ukládací jednotky do nové lokace / spisovny.Ukládací jednotka je označena uživatelským identifikátorem nebo idebtifikátorem ze zdrojového systému.

**Vstup:**

**ChangeSUnitLocatioRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string = login pod kterým je volána služba
* identifier - type string *nebo* erms\_uid type string = unikátní označení ukládací jednotky, uživatelský identifikátor nebo identifikátor zdrojového systému
* locationAcr - type string = akronym lokality/spisovny odpovídající číselníku
* description - type string = doplňující popis umístění

**Výstup**:

**ChangeSUnitLocationResponse**

* status - type string

Možné návratové kódy - parametr "status":

|  |  |
| --- | --- |
| **Zkratka v parametru** | **Popis** |
| SUNIT\_SUCCESSFULLY\_UPDATED | Umístění ukládací jednotky úspěšně změněno |
| SUNIT\_UPDATE\_FAILED | Chyba při změně umístění |

## Operace setupSUnit

Operace rozhraní je volána pro nastavení parametrů ukládací jednotky. U existující ukládací jednotky se mění pouze parametry uvedené v requestu, u ostatních zůstavají původní nastavené hodnoty. Pokud není ukládací jednotka podle identifikátoru nalezena, zakládá se nová.

**Vstup:**

**setupSUnitRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string = login pod kterým je volána služba
* identifier - type string *nebo* erms\_uid type string = unikátní označení ukládací jednotky, uživatelský identifikátor nebo identifikátor zdrojového systému. Lze zadat i oba identifikátory. Primárně se pro vyhledání stávající ukládací jednotky použije "erms\_uid".
* title - type string = název ukládací jednotky
* description - type string = popis ukládací jednotky
* type - string = typ ukládací jednotky (fyzická podoba)
* classificationFullyQcc - string = plně určený spisový znak dle spisového plánu organizace
* retentionAndDispositionControlAcronym - string = skartační znak
* author = autor (vlastník) ukládací jednotky
	+ fullname - type string = plné jméno
	+ identifier - type string = identifikátor
	+ address - type string = adresa
	+ email - type string = e-mailová adresa
	+ organisationIdentifier - type string = identifikátor organizace
	+ organisationalUnit - type string = organizační jednotka
	+ position - type string = funkční místo
* dateCreated - date = datum vytvoření
* dateClosed - date = datum uzavření
* unitsCount - int = počet fyzických jednotek v rámci ukládacích jednotek
* itemsCount - int = deklarovaný počet všech obsažených balíčků AIP
* locationAcr - type string = akronym lokality/spisovny odpovídající číselníku
* locationDescription - type string = doplňující popis umístění

**Výstup**:

**SetupSUnitResponse**

* status - type string

Možné návratové kódy - parametr "status":

|  |  |
| --- | --- |
| **Zkratka v parametru** | **Popis** |
| SUNIT\_SUCCESSFULLY\_UPDATED | Parametry ukládací jednotky úspěšně změněny |
| SUNIT\_UPDATE\_FAILED | Chyba při aktualizaci |
| SUNIT\_SUCCESSFULLY\_CREATED | Ukládací jednotka založena |
| SUNIT\_CREATE\_FAILED | Chyba při zakládání ukládací jednotky |

## Operace setPackageReference

Operace se použije při nastavování křížových odkazů mezi jednotlivými balíčky (spis-spis, dokument-spis, dokument-dokument). Je možné definovat křížový odkaz i mezi spisy/dokumenty, které dosud nejsou uloženy ve spisovně. Tedy zdroj ani cíl křížového odkazu nemusí být v okamžiku předání uložen ve spisovně.

Odkazy jsou definovány prostřednictvím externích identifikátorů (erms\_uid). Tyto identifikátory jsou jedinečné v rámci původce (napříč odesílajícími aplikacemi). Pro odkazy je alternativně možno použít i identifikátor balíčku systému DESA, který je vrácen předávající aplikaci při předání balíčku SIP.

Chyba bude vrácena pokud:

* definovaný odkaz je již zaevidován
* neexistuje odkazující nebo odkazované id balíčku (netýká se definice odkazu externím identifikátorem)

**Vstup:**

**SetPackageReferenceRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string
* referring

erms\_uid *nebo* idSIPVersion - type string

* referenced

erms\_uid *nebo* idSIPVersion - type string

* specification - type string

**Výstup**:

**SetPackageReferenceResponse**

* status type string

**Příklad:**

**Vstup:**

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">

 <S:Body>

 <ns2:setPackageReferenceRequest xmlns:ns2="http://i.cz/dea/schemas/SIPSubmission/types">

 <producerCode>Puvodce1</producerCode>

 <userLogin>admin</userLogin>

 <referring><idSIPVersion>97736b56-7ce7-4daa-89b1-8bd003ae54a5</idSIPVersion></referring >

 <referenced><erms\_uid>SPS\_123445667</erms\_uid ></ referenced >

 <specification>priorace</specification>

 </ns2: setPackageReferenceRequest >

 </S:Body>

 </S:Envelope>

**Výstup (v případě úspěšného volání):**

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

 <soap:Body>

 <ns2: setPackageReferenceResponse xmlns:ns2="http://i.cz/dea/schemas/SIPSubmission/types">

 <status>OK</status >

 </ns2: setPackageReferenceResponse>

 </soap:Body>

</soap:Envelope>

Možné návratové kódy - parametr "status":

|  |  |
| --- | --- |
| **Zkratka v parametru** | **Popis** |
| OK | Operace proběhla bez chyby |
| REFERENCE\_ALREADY\_EXISTS | Odkaz již existuje |
| REFERRING\_ID\_NOT\_EXISTS | Odkazující identifikátor neexistuje |
| REFERED\_ID\_NOT\_EXISTS | Odkazovaný identifikátor neexistuje |

## Operace removePackageReference

Operace se použije při stornování křížových odkazů mezi jednotlivými balíčky (spis-spis, dokument-spis, dokument-dokument). Je možné zrušit křížový odkaz i mezi spisy/dokumenty, které dosud nejsou uloženy ve spisovně.

Odkazy jsou definovány prostřednictvím externích identifikátorů (erms\_uid). Tyto identifikátory jsou jedinečné v rámci původce (napříč odesílajícími aplikacemi). Pro odkazy je alternativně možno použít i identifikátor balíčku systému DESA, který je vrácen předávající aplikaci při předání balíčku SIP.

V případě, že definovaný odkaz je již předán v rámci metadat balíčku, zrušení odkazu není možné.

Chyba bude vrácena pokud:

* definovaný odkaz není zaevidován
* definovaný odkaz je součástí metadat balíčku
* neexistuje odkazující nebo odkazované id balíčku (netýká se definice odkazu externím identifikátorem)

**Vstup:**

**RemovePackageReferenceRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string
* referring

erms\_uid *nebo* idSIPVersion - type string

* referenced

erms\_uid *nebo* idSIPVersion - type string

**Výstup**:

**RemovePackageReferenceResponse**

* status type string

**Příklad:**

**Vstup:**

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">

 <S:Body>

 <ns2:removePackageReferenceRequest xmlns:ns2="http://i.cz/dea/schemas/SIPSubmission/types">

 <producerCode>Puvodce1</producerCode>

 <userLogin>admin</userLogin>

 <referring><erms\_uid >AP1\_1234567</erms\_uid ></referring >

 <referenced><erms\_uid>SPS\_123445667</erms\_uid ></ referenced >

 </ns2: removePackageReferenceRequest >

 </S:Body>

 </S:Envelope>

**Výstup (v případě úspěšného volání):**

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

 <soap:Body>

 <ns2: setPackageReferenceResponse xmlns:ns2="http://i.cz/dea/schemas/SIPSubmission/types">

 <status>OK</status >

 </ns2: setPackageReferenceResponse>

 </soap:Body>

</soap:Envelope>

Možné návratové kódy - parametr "status":

|  |  |
| --- | --- |
| **Zkratka v parametru** | **Popis** |
| OK | Operace proběhla bez chyby |
| REFERENCE\_NOT\_EXISTS | Odkaz neexistuje |
| REFERENCE\_IN\_METADATA | Odkaz je trvale definován v metadatech |
| REFERRING\_ID\_NOT\_EXISTS | Odkazující identifikátor neexistuje |
| REFERED\_ID\_NOT\_EXISTS | Odkazovaný identifikátor neexistuje |

## Operace getReferences

Operace vrací již existující křížové odkazy mezi dokumenty a spisy. Je možné získat buď odkazující balíčky nebo odkazované balíčky k balíčku s předaným identifikátorem ve formě seznamu s doplňujícími údaji.

**Vstup:**

**GetReferencesRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string
* referring

erms\_uid *nebo* idSIPVersion - type string

* *nebo* referenced

erms\_uid *nebo* idSIPVersion - type string

**Výstup**:

**GetReferencesResponse**

* status type string
* references
	+ **referring *nebo* referenced**
	+ erms\_uid - externí identifikátor ze zdrojového systému - type string
	+ idSIPVersion - identifikátor balíčku - type string
	+ classificationFullyQcc - odkazovaná položka spisového plánu - type string
	+ specification - popis typu odkazu - type string
	+ isPredefined - odkaz není definovaný v metadatech balíčku - type boolean
	+ predefinedTime - čas externí evidence odkazu - type dateTime

**Příklad:**

**Vstup:**

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">

 <S:Body>

 <ns2:getReferencesRequest xmlns:ns2="http://i.cz/dea/schemas/SIPSubmission/types">

 <producerCode>Puvodce1</producerCode>

 <userLogin>admin</userLogin>

 <referring><erms\_uid >AP1\_1234567</erms\_uid ></referring >

 </ns2: getReferencesRequest >

 </S:Body>

 </S:Envelope>

**Výstup (v případě úspěšného volání):**

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

 <soap:Body>

 <ns2: getReferencesResponse xmlns:ns2="http://i.cz/dea/schemas/SIPSubmission/types">

 <status>OK</status >

 <references>

 <referenced>

 <erms\_uid >SPS\_1234567</erms\_uid >

 <specification >priorace</ specification >

 <isPredefined >true</ isPredefined >

 <predefinedTime > 2002-05-30T09:00:00 </predefinedTime >

 </referenced>

 <referenced>

 <erms\_uid >SPS\_1234568</erms\_uid >

 <idSIPversion >97736b56-7ce7-4daa-89b1-8bd003ae54a5</ idSIPversion>

 <specification >odkaz</ specification >

 <isPredefined >true</ isPredefined >

 <predefinedTime > 2002-05-30T10:00:00 </predefinedTime >

 </referenced>

 </ references >

 </ns2: getReferencesResponse >

 </soap:Body>

</soap:Envelope>

## Operace setPackageProcessor

Operace se použije při nastavování aktuálního zpracovatele spisu nebo dokumentu. Aktuální zpracovatel se nastavuje prostřednictvím identifikátoru funkčního místa ze spisové služby a slouží pro validaci oprávnění pro některé funkce spisovny, např. schvalování výdeje balíčků a schvalování skartačních operací jednotlivých balíčků, pokud je tato funkce v aplikaci povolena.

Tyto identifikátory jsou jedinečné v rámci původce (napříč odesílajícími aplikacemi). Identifikátory musí být naplněny v číselníku funkčních míst.

Nastavení aktuálního zpracovatele nemění údaje o původním zpracovateli v uloženém balíčku.

Parametr „package“ udává identifikátor balíčku buď jeko generovaný identifikátor spisovny vrácený při vstupu balíčku nebo jako identifikátor spisové služby uvedený v balíčku.

**Vstup:**

**SetPackageProcessorRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string
* package

erms\_uid *nebo* idSIPVersion - type string

* positionID - type string

**Výstup**:

**SetPackageProcessorResponse**

* status type string

**Příklad:**

**Vstup:**

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">

 <S:Body>

 <ns2:SetPackageProcessorRequest xmlns:ns2="http://i.cz/dea/schemas/SIPSubmission/types">

 <producerCode>Puvodce1</producerCode>

 <userLogin>admin</userLogin>

 <package><idSIPVersion>97736b56-7ce7-4daa-89b1-8bd003ae54a5</idSIPVersion></package >

 < positionID>FM\_123445667</positionID >

 </ns2: SetPackageProcessorRequest >

 </S:Body>

 </S:Envelope>

**Výstup (v případě úspěšného volání):**

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

 <soap:Body>

 <ns2: SetPackageProcessorResponse xmlns:ns2="http://i.cz/dea/schemas/SIPSubmission/types">

 <status>OK</status >

 </ns2: SetPackageProcessorResponse>

 </soap:Body>

</soap:Envelope>

Možné návratové kódy - parametr "status":

|  |  |
| --- | --- |
| **Zkratka v parametru** | **Popis** |
| OK | Operace proběhla bez chyby |
| PACKAGE\_NOT\_EXISTS | Balíček neexistuje |
| POSITION\_ID\_NOT\_EXISTS | Identifikátor funkčního místa zpracovatele neexistuje |

## Operace updateExternalEvent

Operace se použije pro nastavení datumu externí spouštěcí události. Nastavení datumu má efekt pouze u dokumentů a spisů, jejichž skartační režim nastavený v rámci balíčku má typ skartační události nastavený jako externí. U ostaních balíčků se nepoužije.

Údaj určuje, kdy začíná běžet skartační lhůta pro dokument nebo spis. Dokud není údaj nastaven, balíček není vybrán do skartačního řízení.

Na vstupu se balíček označí identifikátorem verze AIP nebo externím identifikátorem.

**Vstup:**

**updateExternalEvent**

* producerID - type int *nebo* producerCode type string
* userLogin - type string
* package

externalId *nebo* aipVersionId - type string

* externalEventDate - type date

**Výstup**:

**updateExternalEventResponse**

* status type string

**Příklad:**

**Vstup:**

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">

 <S:Body>

 <ns2: updateExternalEventRequest xmlns:ns2="http://i.cz/dea/schemas/SIPSubmission/types">

 <producerCode>Puvodce1</producerCode>

 <userLogin>admin</userLogin>

 <package><aipVersionId >97736b56-7ce7-4daa-89b1-8bd003ae54a5</aipVersionId></package >

 <externalEventDate>2018-10-26</externalEventDate >

 </ns2:updateExternalEventRequest >

 </S:Body>

 </S:Envelope>

**Výstup (v případě úspěšného volání):**

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

 <soap:Body>

 <ns2: updateExternalEventResponse xmlns:ns2="http://i.cz/dea/schemas/SIPSubmission/types">

 <status>OK</status >

 </ns2: updateExternalEventResponse>

 </soap:Body>

</soap:Envelope>

## Operace selectPackagesForTimeStamp

Operace rozhraní je volána pro přidání balíčků do procesu časového razítkování. Nastavení má vliv pouze na balíčky, které se standardně dle konfigurace nerazítkují, ale aplikace časového razítka je takto vynucena. Volání rozhraní nespustí proces časového razítkování, ale pouze označí balíčky tak, aby v rámci nastávající úlohy vytváření časových razítek došlo k jejich vytvoření.

Pro aplikaci časových razítek je podmínkou správné nastavení konfigurace autority a povolení funkcionality v konfiguraci.

**Vstup:**

**SelectPackagesForTimeStampRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string = login pod kterým je volána služba
* idSIPVersions - type ArraySIPVersion = seznam přidávaných balíčků
	+ - idSIPVersion - optional, unbounded; type string = identifikátor verze balíčku při předání do DESA

**Výstup**:

**CreateRDProcessResponse**

* status - type string

## Operace getSIPContent

Operace vrací obsah balíčku SIP. Používá, pokud balíček tvoří celý spis, obsahující více dokumentů, z nichž DESA ukládá samostatné balíčky AIP. Pro manipulaci s těmito balíčky je nutné získat identifikátor prostřednictvím této metody, pokud je volající aplikace nezná.

**Vstup:**

**getSIPContent**

* producerID - type int *nebo* producerCode type string
* userLogin - type string
* idSIPVersion - type string

**Výstup**:

**getSIPContentResponse**

* packageList - type ArraySIPContent = seznam AIP v balíčku SIP
	+ item - SIPContentPackage
		- idSIPVersion - type string = identifikátor verze balíčku AIP při předání do DESA
		- erms\_uid - type string = identifikátor externího systému
		- producerIdentifier - type string = uživatelský identifikátor z externího systému
		- packageStateCode - type string = stav balíčku

# Operace výstupního rozhraní

## Operace findPackages

Operace pro vyhledávání balíčků uložených ve spisovně podle některého z identifikátorů. Funkce vrací seznam nalezených identifikátorů, zejména idSIPVersion, který lze použít při volání dalších funkcí API.

Funkce se používá v rámci externího systému, který není zdrojem balíčků spisovny, ale může dokument identifikovat jiným způsobem, nebo při uživatelském vyhledávání v rámci externího systému.

Parametr "returnDetails" určuje, zda se mají vrátit i další pole mimo idSIPVersion, erms\_uid a producerIdentifier.

**Vstup:**

**FindPackagesRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string
* filterFields (libovolné z uvedených filtračních podmínek)

idSIPVersion - identifikátor balíčku DESA - type string

erms\_uid - externí identifikátor (typicky identifikátor spisové služby) - type string

producerIdentifier - číslo jednací / spisová značka - type string

title - věc/název - type string

metadataFulltext – hledání metadat v plnotextovém indexu, pokud je nakonfigurován - type string

contentFulltext – hledání obsahu v plnotextovém indexu, pokud je nakonfigurován - type string

* maxItemCount - volitelně - maximální počet vrácených záznamů - type int
* returnDetails - vrátit detaily pro uživatele, nejen identifikátory - type boolean

**Výstup**:

**FindPackagesResponse**

* packageList (seznam)
	+ item
		- idSIPVersion - identifikátor balíčku - type string
		- erms\_uid - externí identifikátor ze zdrojového systému - type string
		- producerIdentifier - číslo jednací / spisová značka - type string
		- title - věc/název balíčku - type string
		- classificationFullyQcc - položka spisového plánu - type string
		- fulltextHighlightItems – seznam nalezených výskytu při fulltextovém vyhledávání obsahu
			* fulltextHighlightItem
				+ fileName – název souboru z balíčku
				+ highlight – nalezený text
		- timeArchived - čas uložení do spisovny - type dateTime
* limitedItemCount - výsledek byl omezen na uvedený počet záznamů - int

**Příklad:**

**Vstup:**

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">

 <S:Body>

 <ns2:findPackagesRequest xmlns:ns2="http://i.cz/dea/schemas/SIPSubmission/types">

 <producerCode>Puvodce1</producerCode>

 <userLogin>admin</userLogin>

 <filterFields><erms\_uid >AP1\_1234567</erms\_uid ></ filterFields>

 </ns2: findPackagesRequest >

 </S:Body>

 </S:Envelope>

**Výstup (v případě úspěšného volání):**

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

 <soap:Body>

 <ns2: findPackagesResponse xmlns:ns2="http://i.cz/dea/schemas/SIPSubmission/types">

 <packageList>

 <item>

 <idSIPversion >97736b56-7ce7-4daa-89b1-8bd003ae54a5</ idSIPversion>

 <erms\_uid > AP1\_1234567</erms\_uid >

 <producerIdentifier >DOK124/2014</ specification >

 <title>Dokument 1 - žádost</ title >

 <classificationFullyQcc>1.2.3</ classificationFullyQcc >

 <timeArchived > 2002-05-30T09:00:00 </ timeArchived >

 </ item >

 </ packageList >

 </ns2: findPackagesResponse>

 </soap:Body>

</soap:Envelope>

## Operace requestDIP

Operace zakládá požadavek na výdej balíčku DIP. Vstupem je množina identifikátorů balíčků, kterou zasílá volající aplikace. Dále je identifikován uživatel, jehož jménem je žádáno o vydání balíčku, a důvod požadavku. Systémový účet použitý pro spuštění operace musí mít oprávnění pro výdej balíčků v rámci původce uživatele, pro nějž se balíček vydává.

Alternativně se předávají údaje o externím uživateli, který žádá o výdej obsahu a je finálním příjemce vydaného digitálního obsahu.

Vrací se identifikátor balíčku DIP a informace o stavu přípravy DIP. Pokud je balíček připraven k výdeji ihned po zaslání požadavku (synchronní operace) je touto funkcí vrácen stav DIP\_READY. Informace o stavech vrací funkce getDIPStatus. V rámci dokumentace této funkce je uveden seznam možných stavů. Vlastní obsah balíčku je dostupný prostřednictvím operace getDIPContent.

**Vstup:**

**requestDIP**

* producerID - type int *nebo* producerCode type string
* userLogin - type string = požadavek na výdej pro interního uživatele s daným login jménem

*nebo*

* externalSubject - type typeDIPExternalSubject = požadavek na výdej pro externí subjekt
	+ internalUserLogin - type string = login interrního uživatele, který žádá o výdej pro externí subjekt
	+ name - type string = jméno osoby
	+ surname - type string = příjmení osoby
	+ identifier - type string = identifikátor (např. RČ, OP)
	+ address - type string = adresa
* userReason – type string
* packageList type ArrayPackageID
	+ item - optional, unbounded; type PackageID
		- idSIPVersion - type string

**Výstup**:

**getPackageStatusResponse**

* + idDIP - type string
	+ DIPState - type string

**Příklad:**

**Vstup:**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

 <soapenv:Header/>

 <soapenv:Body>

 <typ:requestDIP xmlns:typ=„http://i.cz/dea/schemas/SIPOutputInterface/types“>

 <producerID>5</producerID>

 <userLogin>jnovak</userLogin>

 <userReason>Dohledání informací k případu.</userReason>

 <packageList>

 <item>

 <idSIPVersion>4721787c-3301-4cdb-8b26-8cc90e0b08dc</idSIPVersion>

 </item>

 <item>

 <idSIPVersion>97736b56-7ce7-4daa-89b1-8bd003ae54a5</idSIPVersion>

 </item>

 </packageList>

 </typ: requestDIP >

 </soapenv:Body>

</soapenv:Envelope>

**Výstup:**

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

 <soap:Body>

 <ns2: requestDIPResponse xmlns:ns2="http://i.cz/dea/schemas/SIPOutputInterface/types">

 <idDIP>269898</idDIP>

 <DIPState>DIP\_CONFIRM</ DIPState>

 </ns2: requestDIPResponse>

 </soap:Body>

</soap:Envelope>

## Operace resolveDIPRequest

Operace umožňuje schválit již evidovanou žádost o výdej balíčku.

**Vstup:**

**ResolveDIPRequestRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string = požadavek na výdej pro interního uživatele s daným login jménem
* idDIP - type string = identifikátor DIP
* result - type ApprovalRequestResultType = způsob vyřízení APPROVED nebo DECLINED

**Výstup**:

**ResolveDIPRequestResponse**

* idDIP- type string
* DIPState - type string

## Operace requestLoan

Operace zakládá požadavek na zápůjčku fyzických nebo hybridních dokumentů. Pro čistě elektronické dokumenty nemá smysl. Od operace requestDIP se liší možností evidovat vrácení fyzické zápůjčky dokumentu. Ostatní funkcionalita je analogická. Pokud nemá dokument žádný digitální obsah, není vůbec nutné vyzvedávat obsah prostřednictvím metody getDIPContent.

Vstupem je množina identifikátorů balíčků, kterou zasílá volající aplikace. Dále je identifikován uživatel, jehož jménem je žádáno o vydání balíčku, a důvod požadavku. Systémový účet použitý pro spuštění operace musí mít oprávnění pro výdej balíčků v rámci původce uživatele, pro nějž se balíček vydává.

Alternativně se předávají údaje o externím uživateli, který žádá o výdej obsahu a je finálním příjemce vydaného digitálního obsahu.

Vrací se identifikátor balíčku DIP a informace o stavu přípravy DIP. Pokud je balíček připraven k výdeji ihned po zaslání požadavku (synchronní operace) je touto funkcí vrácen stav DIP\_READY. Informace o stavech vrací funkce getDIPStatus. V rámci dokumentace této funkce je uveden seznam možných stavů. Vlastní digitální obsah balíčku a metadata jsou dostupné prostřednictvím operace getDIPContent.

**Vstup:**

**requestLoanRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string = požadavek na výdej pro interního uživatele s daným login jménem

*nebo*

* externalSubject - type typeDIPExternalSubject = požadavek na výdej pro externí subjekt
	+ internalUserLogin - type string = login interrního uživatele, který žádá o výdej pro externí subjekt
	+ name - type string = jméno osoby
	+ surname - type string = příjmení osoby
	+ identifier - type string = identifikátor (např. RČ, OP)
	+ address - type string = adresa
* userReason – type string
* sendByEmail - type boolean = zaslat digitální obsah emailem
* packageList type ArrayPackageID
	+ item - optional, unbounded; type PackageID
		- idSIPVersion - type string
* prepareOnly – type boolean = zápůjčka je pouze rezervována, nikoli ihned vydána, výdej se eviduje další operací deliverLoan
* deliveryType - type DeliveryType = typ doručení
* deliveryServiceSpecification - type string = specifikace doručení

**Výstup**:

**requestLoanResponse**

 idDIP - type string

* + DIPState - type string

## Operace resolveLoanRequest

Operace umožňuje schválit již evidovanou žádost o zápůjčku.

**Vstup:**

 **ResolveLoanRequestRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string = požadavek na výdej pro interního uživatele s daným login jménem
* idDIP - type string = identifikátor zápůjčky
* result - type ApprovalRequestResultType = způsob vyřízení APPROVED nebo DECLINED

**Výstup**:

**ResolveLoanRequestResponse**

* idDIP- type string
* DIPState - type string

## Operace deliverLoan

Operace umožňuje evidovat výdej zápůjčky, která již byla registrována a nebyla vydána přímo při založení.

**Vstup:**

 **DeliverLoanRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string = požadavek na výdej pro interního uživatele s daným login jménem
* idDIP - type string = identifikátor zápůjčky

**Výstup**:

**DeliverLoanResponse**

* idDIP- type string
* DIPState - type string

## Operace returnLoan

Operace označuje předchozí zápůjčku fyzických nebo hybridních dokumentů za vrácenou. Pro čistě elektronické dokumenty nemá smysl.

Vstupem je množina identifikátor zápůjčky (idDIP) získaný metodou requestLoan. Vrací se informace o výsledném stavu DIP.

**Vstup:**

**requestLoanRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string = požadavek na výdej pro interního uživatele s daným login jménem
* idDIP - type string = identifikátor zápůjčky
* returnDate - type date = datum vrácení zápůjčky

**Výstup**:

**requestLoanResponse**

* DIPState - type string

## Operace requestView

Operace zakládá požadavek na nahlížení do fyzických nebo elektronických dokumentů. Od operace requestDIP se liší možností evidence nahlížení i do fyzického dokumentu. Ostatní funkcionalita je analogická. Pokud nemá dokument žádný digitální obsah, není vůbec nutné vyzvedávat obsah prostřednictvím metody getDIPContent.

V případě digitálního obsahu je možné použít uživatelského rozhraní DESA pro externí uživatele (např. informační kiosky, studovny) s přístupem prostřednictvím kódu pro nahlížení, který je součástí návratových hodnot této metody.

Vstupem je množina identifikátorů balíčků, kterou zasílá volající aplikace. Dále je identifikován uživatel, jehož jménem je žádáno o vydání balíčku, a důvod požadavku. Systémový účet použitý pro spuštění operace musí mít oprávnění pro výdej balíčků v rámci původce uživatele, pro nějž se balíček vydává.

Alternativně se předávají údaje o externím uživateli, který žádá o výdej obsahu a je finálním příjemce vydaného digitálního obsahu.

Vrací se identifikátor balíčku DIP a informace o stavu přípravy DIP. Pokud je balíček připraven k výdeji ihned po zaslání požadavku (synchronní operace) je touto funkcí vrácen stav DIP\_READY. Informace o stavech vrací funkce getDIPStatus. V rámci dokumentace této funkce je uveden seznam možných stavů. Vlastní digitální obsah balíčku a metadata jsou dostupné prostřednictvím operace getDIPContent.

**Vstup:**

**requestViewRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string = požadavek na výdej pro interního uživatele s daným login jménem

*nebo*

* externalSubject - type typeDIPExternalSubject = požadavek na výdej pro externí subjekt
	+ internalUserLogin - type string = login interrního uživatele, který žádá o výdej pro externí subjekt
	+ name - type string = jméno osoby
	+ surname - type string = příjmení osoby
	+ identifier - type string = identifikátor (např. RČ, OP)
	+ address - type string = adresa
* userReason – type string
* packageList type ArrayPackageID
	+ item - optional, unbounded; type PackageID
		- idSIPVersion - type string

**Výstup**:

**requestViewResponse**

* idDIP - type string = identifikátor DIP
	+ DIPState - type string
	+ viewCode - type string - kód pro nahlížení na obsah DIP pro externí přístup uživatelů k DESA.

## Operace resolveViewRequest

Operace umožňuje schválit již evidovanou žádost o nahlížení.

**Vstup:**

 **ResolveViewRequestRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string = požadavek na výdej pro interního uživatele s daným login jménem
* idDIP - type string = identifikátor nahlížení
* result - type ApprovalRequestResultType = způsob vyřízení APPROVED nebo DECLINED

**Výstup**:

**ResolveViewRequestResponse**

* idDIP- type string
* DIPState - type string

## Operace requestManipulation

Operace zakládá požadavek na fyzickou manipulaci fyzických nebo hybridních dokumentů ve spisovně. Pro čistě elektronické dokumenty nemá smysl.

Vstupem je množina identifikátorů balíčků, kterou zasílá volající aplikace. Dále je identifikován uživatel, jehož jménem je žádáno o vydání balíčku, a důvod požadavku. Systémový účet použitý pro spuštění operace musí mít oprávnění pro výdej balíčků v rámci původce uživatele, pro nějž se balíček vydává.

Vrací se identifikátor balíčku DIP a informace o stavu přípravy DIP. Pokud je balíček připraven k manipulaci ihned po zaslání požadavku (synchronní operace) je touto funkcí vrácen stav DIP\_READY. Informace o stavech vrací funkce getDIPStatus. V rámci dokumentace této funkce je uveden seznam možných stavů. Vlastní digitální obsah balíčku a metadata jsou dostupné prostřednictvím operace getDIPContent.

**Vstup:**

**RequestManipulationRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string = požadavek na výdej pro interního uživatele s daným login jménem

*nebo*

* externalSubject - type typeDIPExternalSubject = požadavek na výdej pro externí subjekt
	+ internalUserLogin - type string = login interrního uživatele, který žádá o výdej pro externí subjekt
	+ name - type string = jméno osoby
	+ surname - type string = příjmení osoby
	+ identifier - type string = identifikátor (např. RČ, OP)
	+ address - type string = adresa
* userReason – type string
* packageList type ArrayPackageID
	+ item - optional, unbounded; type PackageID
		- idSIPVersion - type string
* prepareOnly – type boolean = není proveden přímo výdej j manipulaci a pouze je zaevidována žádost

**Výstup**:

**RequestManipulationResponse**

 idDIP - type string = identifikátor manipulace

* + DIPState - type string

## Operace startManipulation

Operace umožňuje evidovat zahájení manipulace, která již byla registrována a nebyla zahájena přímo při založení.

**Vstup:**

 **StartManipulationRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string = požadavek na interního uživatele s daným login jménem
* idDIP - type string = identifikátor manipulace

**Výstup**:

**StartManipulationResponse**

* idDIP- type string
* DIPState - type string

## Operace finishManipulation

Operace umožňuje evidovat ukončení manipulace, která již byla zahájena.

**Vstup:**

**FinishManipulationRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string = požadavek na interního uživatele s daným login jménem
* idDIP - type string = identifikátor manipulace

**Výstup**:

**FinishManipulationResponse**

* idDIP- type string
* DIPState - type string

## Operace getDIPStatus

Operace vrací stav přípravy balíčku DIP se zadaným identifikátorem. Funkci je nutné volat, pokud není balíček připraven k výdeji obratem po zaslání požadavku, jde tedy o asynchronní proces výdeje balíčku z důvodu nutnosti schválení výdeje nebo z technických důvodů na straně archivního úložiště systému..

**Vstup:**

**getDIPStatus**

* producerID - type int *nebo* producerCode type string
* userLogin - type string
* idDIP – type string

**Výstup**:

**getDIPStatusResponse**

* + DIPState - type string

**Stavy balíčku DIP**:

|  |  |
| --- | --- |
| **Zkratka stavu** | **Název stavu** |
| DIP\_CREATED | Žádost o výdej balíčku vytvořena |
| DIP\_CONFIRM | Čeká se na schválení obsahu balíčku |
| **DIP\_READY** | **Balíček připraven k odeslání** |
| DIP\_SENT | Balíček odeslán |
| DIP\_REJECT | Zamítnut výdej balíčku |

**Příklad:**

**Vstup:**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

 <soapenv:Header/>

 <soapenv:Body>

 <typ:getDIPStatus xmlns:typ="http://i.cz/dea/schemas/SIPOutputInterface/types">

 <producerID>5</producerID>

 <userLogin>jnovak</userLogin>

 <idDIP>269898</idDIP>

 </typ: getDIPStatus>

 </soapenv:Body>

</soapenv:Envelope>

**Výstup:**

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

 <soap:Body>

 <ns2: getDIPStatusResponse xmlns:ns2="http://i.cz/dea/schemas/SIPOutputInterface/types">

 <DIPState>DIP\_READY</producerSIPID>

 </ns2: getDIPStatusResponse>

 </soap:Body>

</soap:Envelope>

## Operace getDIPContent

Operace vrací obsah balíčku DIP, pokud je připraven k výdeji. Výstupní balíček DIP je ve formátu ZIP, který jednotlivě obsahuje obálky mets.xml a obsah datových souborů (komponent) ke všem dokumentům, které jsou zahrnuty v balíčku DIP. Jednotlivé zařazené balíčky jsou obsaženy v podadresářích pojmenovaných identifikátorem verze balíčku.

Obsah balíčku DIP může na základě konfigurace a oprávnění uživatele obsahovat omezenou množinu metadat a datových souborů (komponent) oproti datům uleženým v balíčcích AIP archivního úložiště systému DESA.

**Vstup:**

**getDIPContent**

* producerID - type int *nebo* producerCode type string
* userLogin - type string
* idDIP – type string

**Výstup**:

**getDIPStatusResponse**

* + DIPContent - type base64Binary

**Příklad:**

**Vstup:**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

 <soapenv:Header/>

 <soapenv:Body>

 <typ:getDIPContent xmlns:typ="http://i.cz/dea/schemas/SIPOutputInterface/types">

 <producerID>5</producerID>

 <userLogin>jnovak</userLogin>

 <idDIP>269898</idDIP>

 </typ: getDIPContent >

 </soapenv:Body>

</soapenv:Envelope>

**Výstup:**

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

 <soap:Body>

 <ns2: getDIPContentResponse xmlns:ns2="http://i.cz/dea/schemas/SIPOutputInterface/types">

 <DIPContent>

 PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4NCjxnZW46bm9t

 ZW5jbGF0dXJlcyB4bWxuczpnZW49Imh0dHA6Ly9kZWEuaS5jei9ub21lbi94bWxi

 ZWFucy9nZW4iPg0KICAgDQogIDxnZW46cmVjLWNscz4NCiAgICA8Z2VuOnRpdGxl

 PlBvbG+ea2Ega2xhc2lmaWthY2U8L2dlbjp0aXRsZT4NCiAgICA8Z2VuOmRlc2Ny

 PktsYXNpZmlrYWNlIGRva3VtZW50dSBwb2RsZSBzcGlzb3bpaG8gcGzhbnU8L2dl

 bjpkZXNjcj4NCiAgICA8Z2VuOnJlYy1jbD4NCiAgICAgIDxnZW46ZnVsbHlRY2M

 .........

 </DIPContent>

 </ns2: getDIPContentResponse>

 </soap:Body>

</soap:Envelope>

## Operace getDIPList

Operace vrací seznam evidováných zápůjček, nahlížení, žádostí o výdej DIP a evidovaných manipulací ve fyzické spisovně.

Výstupem je seznam s identifikátorem balíčku DIP, jeho typem a aktuálním stavem.

**Vstup:**

**getDIPListRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string
* startByTime - type dateTime = čas poslední změny stavu, na který se filtruje vrácený seznam.

**Výstup**:

**getDIPListResponse**

* dipList type DipList
	+ dip - optional, unbounded; type Dip

idDIP - type string

type – type DipType = typ balíčku

DIPState – type string

stateLastChanged - type dateTime = čas poslední změny stavu

userLogin *nebo* externalSubject = subjekt žadatele

userReason – type string = důvod žádosti

deliveryType - DeliveryType = způsob doručení

deliveryServiceSpecification – type string = specifikace způsovu doručení

packageList - ArrayPackageID = seznam obsažených balíčků AIP, označených identifikátorem idAIPVersion, který je balíčku přidělen při odeslání do spisovny.

**Typy žádosti (type)**:

|  |  |
| --- | --- |
| **Zkratka stavu** | **Název stavu** |
| dip | Výdej digitálního obsahu balíčku - DIP. |
| loan | Zápůjčka fyzického dokumentu. |
| view | Nahlížení na digitální nebo fyzický obsah či poskytnutí kopie. |
| manipulation | Manipulace s dokumentem ve fyzické spisovně. |

**Stavy balíčku DIP (DIPState)**:

|  |  |
| --- | --- |
| **Zkratka stavu** | **Název stavu** |
| DIP\_CREATED | Žádost o výdej balíčku vytvořena |
| DIP\_CONFIRM | Čeká se na schválení obsahu balíčku |
| **DIP\_READY** | **Balíček připraven k odeslání** |
| DIP\_SENT | Balíček odeslán |
| DIP\_REJECT | Zamítnut výdej balíčku |

## Operace getRDProcessPackages

Operace vrací seznam balíčků, které byly skartovány, archivovány nebo zařazeny do skartačního řízení. Vstupem je označení původce a časové období, do nějž spadá provedení operací nad balíčky uvedenými na výstupu funkce.

Výstupem je seznam balíčků s identifikátorem balíčku, stavem v rámci skartačního řízení a aktuálním stavem skartačního řízení, do nějž byl balíček zahrnut. Současně se vrací čas provedení archivace nebo skartace.

**Vstup:**

**getRDProcessPackages**

* producerID - type int *nebo* producerCode type string
* userLogin - type string
* startByTime - type dateTime

**Výstup**:

**getRDProcessPackagesResponse**

* packageList type ArrayRDPackages
	+ item - optional, unbounded; type RDPackage

idSIPVersion - type string

RDState – type string

RDProcState – type string

time - type dateTime

**Stavy balíčku ve skartačním řízení (RDState)**:

|  |  |
| --- | --- |
| **Zkratka stavu** | **Název stavu** |
| RD\_DISCARD | Balíček je navržen ke skartaci |
| RD\_ARCHIVE | Balíček je navržen k archivaci |
| RD\_DELAYED | Skartace nebo archivace odložena |
| RD\_DISCARDED | Balíček byl skartován – obsah není k dispozici |
| RD\_ARCHIVED | Balíček byl úspěšně předán do archivu – obsah je dosud k dispozici |
| RD\_A\_DISCARDED  | Balíček archivován - obsah není k dispozici |
| RD\_D\_AVAILABLE   | Balíček skartovaný - obsah dosud k dispozici |
| RD\_DECIDE  | Balíček s volbou skartační operace |
| RD\_RELOCATE | Balíček je navržen k rozluce |
| RD\_ RELOCATED | Rozluka byla provedena |
| RD\_ R\_DISCARDED | Rozluka byla provedena, balíček není k dispozici |

**Stavy skartačního řízení (RDProcState)**:

|  |  |
| --- | --- |
| **Zkratka stavu** | **Název stavu** |
| RDP\_SELECT | Skartační řízení v přípravě |
| RDP\_REVIEW | Skartační řízení ve fázi schvalování |
| RDP\_CANCEL | Skartační řízení stornováno |
| RDP\_ACCEPTED | Schváleny balíčky k archivaci a skartaci |
| RDP\_PROCEED | Probíhá skartace a archivace |
| RDP\_FINISHED | Skartace a archivace ukončena |

**Příklad:**

**Vstup:**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

 <soapenv:Header/>

 <soapenv:Body>

 <typ:getRDProcessPackages xmlns:typ="http://i.cz/dea/schemas/SIPSubmission/types">

 <producerID>5</producerID>

 <userLogin>admin</userLogin>

 <startByTime>2008-10-29T12:01:05</startByTime>

 </typ: getRDProcessPackages >

 </soapenv:Body>

</soapenv:Envelope>

**Výstup:**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

 <soapenv:Header/>

 <soapenv:Body>

 <typ: getRDProcessPackagesResponse xmlns:typ="http://i.cz/dea/schemas/SIPOutputInterface/types">

 <packageList>

 <item>

 <idSIPVersion>9821787c97736b56-7ce7-6886-89b1-8bd003ae54a5</idSIPVersion>

 <rdState>RD\_DISCARDED</rdState>

 <rdProcState>RDP\_FINISHED</ rdProcState>

 <time>2009-11-29T14:00:02</time>

 </item>

 <item>

 <idSIPVersion>4721787c97736b56-7ce7-4daa-89b1-8bd003ae54a5</idSIPVersion>

 <rdState>RD\_ARCHIVED</rdState>

 <rdProcState>RDP\_FINISHED</ rdProcState>

 <time>2009-11-29T14:00:05</time>

 </item>

 </ packageList>

 </typ: getRDProcessPackagesResponse >

 </soapenv:Body>

</soapenv:Envelope>

## Operace getRDProcessState

Operace vrací stav skartačního řízení jako celku. Současně se vrací čas poslední změny stavu skartačního řízení a další jeho parametry nastavené při založení.

**Vstup:**

**getRDProcessStateRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string
* startByTime - type dateTime = minimální datum poslední změny stavu, na nějž se filtrují výsledná skartační řízení
* acronym - type string = akronym skartačního řízení, nemusí být uveden, pak se vrátí stav všech evidovaných s ohledem na parametr startByTime

**Výstup**:

**getRDProcessStateResponse**

* RDProcesses type ArrayRDProcesses
	+ item - optional, unbounded; type RDProcess

acronym - type string = akronym skartačního řízení

RDProcState – type string = stav skartačního řízení

time – type dateTime = čas poslední změny stavu skartačního řízení

type - RDProcessType

determinationDate - date = rozhodný datum

**Stavy skartačního řízení (RDProcState)**:

|  |  |
| --- | --- |
| **Zkratka stavu** | **Název stavu** |
| RDP\_SELECT | Skartační řízení v přípravě |
| RDP\_REVIEW | Skartační řízení ve fázi schvalování |
| RDP\_CANCEL | Skartační řízení stornováno |
| RDP\_ACCEPTED | Schváleny balíčky k archivaci a skartaci |
| RDP\_PROCEED | Probíhá skartace a archivace |
| RDP\_FINISHED | Skartace a archivace ukončena |

**Typy skartačního řízení (RDProcessType)**:

|  |  |
| --- | --- |
| **Zkratka stavu** | **Název stavu** |
| RDT\_INT | Interní skartace, u spisovny je možné až po proběhnutí řádného skartačního řízení, pokud jsou některé digitální dokumenty ponechány v kopii i u původce. |
| RDT\_RD | Řádné skartační řízení. |
| RDT\_REL | Spisová rozluka - předání do jiné organizace. |

## Operace getRDProcessPackage

Operace vrací stav skartace pro jednotlivý balíček.

**Vstup:**

**getRDProcessPackage**

* producerID - type int *nebo* producerCode type string
* userLogin - type string
* acronym – type string = označení skartačního řízení
* idSIPVersion - type string = identifikátor balíčku ke skartaci

**Výstup**:

**getRDProcessPackageResponse**

* packageAction – type RDAipProcessAction
* discardAction - type string
* approvalState– type RDAipProcessApproval
* approvalDate - type dateTime

**Stavy balíčku ve skartačním řízení (RDAipProcessAction)**:

|  |  |
| --- | --- |
| **Zkratka stavu** | **Název stavu** |
| RD\_DISCARD | Balíček je navržen ke skartaci |
| RD\_ARCHIVE | Balíček je navržen k archivaci |
| RD\_DELAYED | Skartace nebo archivace odložena |
| RD\_DISCARDED | Balíček byl skartován – obsah není k dispozici |
| RD\_ARCHIVED | Balíček byl úspěšně předán do archivu – obsah je dosud k dispozici |
| RD\_A\_DISCARDED  | Balíček archivován - obsah není k dispozici |
| RD\_D\_AVAILABLE   | Balíček skartovaný - obsah dosud k dispozici |
| RD\_DECIDE  | Balíček s volbou skartační operace |
| RD\_RELOCATE | Balíček je navržen k rozluce |
| RD\_ RELOCATED | Rozluka byla provedena |
| RD\_ R\_DISCARDED | Rozluka byla provedena, balíček není k dispozici |

**Stavy skartačního řízení (RDAipProcessApproval)**:

|  |  |
| --- | --- |
| **Zkratka stavu** | **Název stavu** |
| RDAIP\_APPROVED | Skartace schválena zpracovatelem |

# Operace administračního rozhraní

## Operace uploadNomenclature

Operace rozhraní je volána při nahrávání číselníků evidovaných v rámci daného původce k danému datu v systému spisovny. Číselníky jsou předávány ve formátu XML souboru, který obsahuje jednotlivé tabulky, jejich záznamy a hodnoty polí tabulky (viz. popis číselníků). Může být plněna pouze vybraná množina číselníků, není nutné plnit všechny současně. Tato funkce naplní kompletně novou instanci číselníku - neaktualizuje stávající položky.

Číselníky je nutné nahrát ve správném pořadí - např. spisový plán závisející na skartačních režimech musí být uložen až po číselníku skartačních režimů.

**Vstup:**

**uploadNomenclatureRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string
* validityFrom - type dateTime
* validityTo - optional type dateTime
* contentXML - type base64Binary

**Výstup**:

**uploadNomenclatureResponse**

* importProblems - type ArrayImportProblem

 item - type ImportProblemStatus

 csType - type string

 text - type string

**Příklad:**

**Vstup:**

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">

 <S:Header/>

 <S:Body>

 <typ: uploadNomenclature xmlns:typ="http://i.cz/dea/schemas/DEAAdmin/types">

 <producerID>5</producerID>

 <userLogin>admin</userLogin>

 < validityFrom 1900-01-01T00:00:00</ validityFrom >

 < contentXML >

PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4NCjxnZW46bm9t

ZW5jbGF0dXJlcyB4bWxuczpnZW49Imh0dHA6Ly9kZWEuaS5jei9ub21lbi94bWxi

 .....

 </ contentXML >

 </typ: uploadNomenclature >

 </S:Body>

</S:Envelope>

**Výstup:**

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

 <ns2: uploadNomenclatureResponse xmlns:ns2="http://i.cz/dea/schemas/DEAAdmin/types">

 < importProblems >

 </ importProblems>

 </ns2: uploadNomenclatureResponse >

</soap:Body>

</soap:Envelope>

## Operace updateNomenclature

Operace rozhraní je volána při aktualizaci číselníků evidovaných v rámci daného původce k danému datu v systému spisovny. Číselníky jsou předávány ve formátu XML souboru, který obsahuje jednotlivé tabulky, jejich záznamy a hodnoty polí tabulky (viz. popis číselníků). Může být plněna pouze vybraná množina číselníků, není nutné plnit všechny současně.

Číselníky je nutné aktualizovat ve správném pořadí - např. spisový plán závisející na skartačních režimech musí být uložen až po číselníku skartačních režimů.

**Vstup:**

**updateNomenclatureRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string
* validityFrom - type dateTime
* contentXML - type base64Binary

**Výstup**:

**updateNomenclatureResponse**

* importProblems - type ArrayImportProblem

 item - type ImportProblemStatus

 csType - type string

 text - type string

**Příklad:**

**Vstup:**

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">

 <S:Header/>

 <S:Body>

 <typ: updateNomenclature xmlns:typ="http://i.cz/dea/schemas/DEAAdmin/types">

 <producerID>5</producerID>

 <userLogin>admin</userLogin>

 < validityFrom 1900-01-01T00:00:00</ validityFrom >

 < contentXML >

PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4NCjxnZW46bm9t

ZW5jbGF0dXJlcyB4bWxuczpnZW49Imh0dHA6Ly9kZWEuaS5jei9ub21lbi94bWxi

 .....

 </ contentXML >

 </typ: updateNomenclature >

 </S:Body>

</S:Envelope>

**Výstup:**

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

 <ns2: updateNomenclatureResponse xmlns:ns2="http://i.cz/dea/schemas/DEAAdmin/types">

 < importProblems >

 </ importProblems>

 </ns2: updateNomenclatureResponse >

</soap:Body>

</soap:Envelope>

## Operace uploadExternalNomenclature

Operace rozhraní je volána při nahrávání nebo aktualizaci číselníku spisových plánů a skartačních režimů evidovaných v rámci daného původce k danému datu v systému spisovny. Číselníky jsou předávány ve formátu XML souboru dle standardu NSESSS2. V případě opakování vět číselníku skartačních režimů se stejným identifikátorem s použije hodnota uvedená u první věcné skupiny se stejným skartačním režimem.

Platnost číselníku i platnost položek je aplikována dle předaného XML souboru. Je možné nahrát číselník s libovolnou platností.

 V případě, že datum otevření předaného číselníku je shodné s uloženým číselníkem, jsou aktualizovány jeho položky. V případě, že předaný číselník má nižší hodnotu data uzavření než stávající číselník, bude doba platnosti zkrácena (slouží např. pro ukončení platnosti stávajícího číselníku a následný import nového číselníku.

Platnost importovaného číselníku se nesmí překrývat s platností již existujícího číselníku.

**Vstup:**

**uploadExternalNomenclatureRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string
* contentXML - type base64Binary

**Výstup**:

**uploadExternalNomenclatureResponse**

* importProblems - type ArrayImportProblem

 item - type ImportProblemStatus

 csType - type string

 text - type string

**Příklad:**

**Vstup:**

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">

 <S:Header/>

 <S:Body>

 <typ: uploadExternalNomenclature xmlns:typ="http://i.cz/dea/schemas/DEAAdmin/types">

 <producerID>5</producerID>

 <userLogin>admin</userLogin>

 < contentXML >

PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4NCjxnZW46bm9t

ZW5jbGF0dXJlcyB4bWxuczpnZW49Imh0dHA6Ly9kZWEuaS5jei9ub21lbi94bWxi

 .....

 </ contentXML >

 </typ: uploadExternalNomenclature >

 </S:Body>

</S:Envelope>

**Výstup:**

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

 <ns2: uploadExternalNomenclatureResponse xmlns:ns2="http://i.cz/dea/schemas/DEAAdmin/types">

 < importProblems >

 </ importProblems>

 </ns2: uploadNomenclatureResponse >

</soap:Body>

</soap:Envelope>

## Operace setupUser

Operace rozhraní je volána při vložení nebo aktualizaci údajů o uživateli systému.

**Vstup:**

**setupUserRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string
* name - type string
* surname - type string
* email - type string
* login - type string
* passwordHash - type string
* isSuperAdmin - type boolean, default false
* hashAlg - type string

**Výstup**:

**setupUserResponse**

* status - type string

**Příklad:**

**Vstup:**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:typ="http://i.cz/dea/schemas/DEAAdmin/types">

 <soapenv:Header/>

 <soapenv:Body>

 <typ:setupUser>

 <producerCode>puvodce1</producerCode>

 <userLogin>admin</userLogin>

 <name>Franta</name>

 <surname>Novák</surname>

 <email>fnovak@test.cz</email>

 <login>fnovak</login>

 <passwordHash> 937b1b31e7fcb21f3125cf8915ec1cf0647222b9dbbbbbc4d0917a9ae76548a1</passwordHash>

 <hashAlg>SHA-256</hashAlg>

 </typ:setupUser>

 </soapenv:Body>

</soapenv:Envelope>

**Výstup:**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:typ="http://i.cz/dea/schemas/DEAAdmin/types">

 <soapenv:Header/>

 <soapenv:Body>

 <typ:setupUserResponse>

 <status>OK</status>

 </typ:setupUserResponse>

 </soapenv:Body>

</soapenv:Envelope>

## Operace deleteUser

Operace rozhraní je volána při rušení uživatele ze systému.

**Vstup:**

**deleteUserRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string = login pod kterým je volána služba
* login - type string = login rušeného uživatele

**Výstup**:

**deleteUserResponse**

* status - type string

**Příklad:**

**Vstup:**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:typ="http://i.cz/dea/schemas/DEAAdmin/types">

 <soapenv:Header/>

 <soapenv:Body>

 <typ:deleteUser>

 <producerCode>puvodce1</producerCode>

 <userLogin>admin</userLogin>

 <login>fnovak</login>

 </typ:deleteUser>

 </soapenv:Body>

</soapenv:Envelope>

**Výstup:**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:typ="http://i.cz/dea/schemas/DEAAdmin/types">

 <soapenv:Header/>

 <soapenv:Body>

 <typ:deleteUserResponse>

 <status>OK</status>

 </typ:deleteUserResponse>

 </soapenv:Body>

</soapenv:Envelope>

## Operace setUserRoles

Operace rozhraní je volána pro nastavení rolí uživatele. Role uživatele se kompletně nahradí seznamem předaných rolí.

**Vstup:**

**setUserRolesRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string = login pod kterým je volána služba
* login - type string = login aktualizovaného uživatele
* roles - seznam rolí uživatele

**Výstup**:

**setUserRolesResponse**

* status - type string

**Příklad:**

**Vstup:**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:typ="http://i.cz/dea/schemas/DEAAdmin/types">

 <soapenv:Header/>

 <soapenv:Body>

 <typ:setUserRoles>

 <producerCode>puvodce1</producerCode>

 <userLogin>admin</userLogin>

 <login>fnovak</login>

 <roles>

 <item>

 <RoleAcr>producer\_admin</RoleAcr>

 </item>

 <item>

 <RoleAcr>producer\_archivist</RoleAcr>

 </item>

 </roles>

 </typ:setUserRoles>

 </soapenv:Body>

</soapenv:Envelope>

**Výstup:**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:typ="http://i.cz/dea/schemas/DEAAdmin/types">

 <soapenv:Header/>

 <soapenv:Body>

 <typ:setUserRolesResponse>

 <status>OK</status>

 </typ:setUserRolesResponse>

 </soapenv:Body>

</soapenv:Envelope>

## Operace setUserLocations

Operace rozhraní je volána pro nastavení povolených spisoven uživatele, pokud je aplikace v režimu omezeného přístupu k doukumentům dle spisoven v rámci původce. Spisovny uživatele se kompletně nahradí seznamem předaných spisoven.

**Vstup:**

**setUserLocationsRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string = login pod kterým je volána služba
* login - type string = login aktualizovaného uživatele
* roles - seznam rolí uživatele

**Výstup**:

**setUserLocationsResponse**

* status - type string

**Příklad:**

**Vstup:**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:typ="http://i.cz/dea/schemas/DEAAdmin/types">

 <soapenv:Header/>

 <soapenv:Body>

 <typ:setUserLocations>

 <producerCode>puvodce1</producerCode>

 <userLogin>admin</userLogin>

 <login>fnovak</login>

 <locations>

 <item>

 <LocationAcr>spisovna1</LocationAcr>

 </item>

 </locations>

 </typ:setUserLocations>

 </soapenv:Body>

</soapenv:Envelope>

**Výstup:**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:typ="http://i.cz/dea/schemas/DEAAdmin/types">

 <soapenv:Header/>

 <soapenv:Body>

 <typ:setUserLocationsResponse>

 <status>OK</status>

 </typ:setUserLocationsResponse>

 </soapenv:Body>

</soapenv:Envelope>

## Operace getUsers

Operace rozhraní je volána pro získání seznamu uživatelů a informací o uživatelích.

**Vstup:**

**getUsersRequest**

* producerID - type int *nebo* producerCode type string
* userLogin - type string = login pod kterým je volána služba

**Výstup**:

**getUsersResponse**

* userList - type ArrayUsers

**Příklad:**

**Vstup:**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:typ="http://i.cz/dea/schemas/DEAAdmin/types">

 <soapenv:Header/>

 <soapenv:Body>

 <typ:getUsers>

 <producerCode>puvodce1</producerCode>

 <userLogin>admin</userLogin>

 </typ:getUsers>

 </soapenv:Body>

</soapenv:Envelope>

**Výstup:**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:typ="http://i.cz/dea/schemas/DEAAdmin/types">

 <soapenv:Header/>

 <soapenv:Body>

 <typ:getUsersResponse>

 <status>OK</status>

 <users>

 <userList>

 <name>Franta</name>

 <surname>Novák</surname>

 <email>fnovak@test.cz</email>

 <login>fnovak</login>

 <passwordHash> 937b1b31e7fcb21f3125cf8915ec1cf0647222b9dbbbbbc4d0917a9ae76548a1</passwordHash>

 <hashAlg>SHA-256</hashAlg>

 <roles>

 <item>

 <RoleAcr>producer\_admin</RoleAcr>

 </item>

 <item>

 <RoleAcr>producer\_archivist</RoleAcr>

 </item>

 </roles>

 <locations>

 <item>

 <LocationAcr>spisovna1</LocationAcr>

 </item>

 </locations>

 </userList>

 </users>

 </typ:getUsersResponse>

 </soapenv:Body>

</soapenv:Envelope>

# Formát číselníků

Číselníky jsou exportovány a importovány prostřednictvím administračního uživatelského rozhraní. Mají XML formát. Vstupní soubor může obsahovat více číselníků zároveň. V případě použití odkazů na položky ostatních číselníků, musejí být odkazované číselníky importovány první (týká se odkazů na jiné číselníky ve spisovém plánu).

Dále uvedený popis neplatí pro číselníky ve formátu dle standardu NSESSS2, které se importují funkcí uploadExternalNomenclature.

Základní struktura XML souboru s obsahem číselníků:



Kořen každého číselníku obsahuje název číselníku a popis a dále jednotlivé záznamy číselníku. Název a popis má pouze informativní význam, Do systému se importují jednotlivé záznamy. Následující schéma platí pro číselník „Spisový plán“, analogickou strukturu mají i ostatní typy číselníků.



## Typy a pole číselníků

|  |  |
| --- | --- |
| *Název číselníku:* | **Spisový plán** |
| *Element (v XML struktuře):* | **rec-cls** | *Element položky (v XML struktuře):* | **rec-cl** |
| Označení "nomenclatureAcronyme" pro metodu "getNomenclatures"  | **RecCl** |  |  |
| *Pole číselníku:* | *Element pole (v XML):* | *Poznámka:* |
| Plně kvalifikovaný spisový znak | **fullyQcc** | Jsou importovány i položky (větve) v rámci struktury, které nejsou přímo využity ke klasifikaci doumentů.  |
| Název | **title** | Název položky |
| Popis | **descr** | Dopňující popis. Není nutné plnit. |
| Skartační znak pro interní archiv | **ag** | Není nutné využívat, slouží v případě interní archivace. Odkazuje na číselník skartačních režimů – phodnota pole ACR. |
| Jednoduchý spisový znak | **classCode** | Poslední část spisového znaku označující aktuální položku. |
| Stupeň důvěryhodnosti dokumentu | **crecntrl** | Použito, pokud není u dokumentu dané klasifikace uvedeno jinak. Není nutné využívat. Odkazuje na číselník stupně důvěryhodnosti – hodnota pole ACR. |
| Skartační znak | **gen** | Odkazuje na číselník skartačních režimů – hodnota pole ACR. |
| Standardní klíčová slova | **keyw** | Použito, pokud není u dokumentu dané klasifikace uvedeno jinak. Není nutné využívat. |
| Standardní umístění analogové podoby dokumentu | **loc** | Použito, pokud není u dokumentu dané klasifikace uvedeno jinak. Není nutné využívat. Odkazuje na číselník umístění – hodnota pole FULLYQCC. |
| Nadřazený spisový znak | **parent** | Odkaz na nadřazený spisový znak, pokud nejde o první úroveň spisového plánu. |
| Přístupová oprávnění | **acl** | Použito, pokud není u dokumentu dané klasifikace uvedeno jinak. Není nutné využívat, slouží pro specifické řízení přístupu k jednotlivě označeným dokumentům na základě role uživatele. Odkazuje na zkratku ACL oprávnění definovanou v konfiguraci systému. |

|  |  |
| --- | --- |
| *Název číselníku:* | **Typ dokumentu** |
| *Element (v XML struktuře):* | **rec-types** | *Element položky (v XML struktuře):* | **rec-type** |
| Označení "nomenclatureAcronyme" pro metodu "getNomenclatures"  | **RecType** |  |  |
| *Pole číselníku:* | *Element pole (v XML):* | *Poznámka:* |
| Název | **title** |  |
| Popis | **descr** | Nepovinné. |

|  |  |
| --- | --- |
| *Název číselníku:* | **Sbírka** |
| *Element (v XML struktuře):* | **a-collects** | *Element položky (v XML struktuře):* | **a-collect** |
| Označení "nomenclatureAcronyme" pro metodu "getNomenclatures"  | **ACollect** |  |  |
| *Pole číselníku:* | *Element pole (v XML):* | *Poznámka:* |
| Název | **title** |  |
| Popis | **descr** |  |
| Specifická popisná metadata | **defBlockXsd** | Odkaz na XSD definici použitou pro plnění specifických popisných metadat dané sbírky. Nemusí se používat. |

|  |  |
| --- | --- |
| *Název číselníku:* | **Fond** |
| *Element (v XML struktuře):* | **a-funds** | *Element položky (v XML struktuře):* | **a-fund** |
| Označení "nomenclatureAcronyme" pro metodu "getNomenclatures"  | **Afund** |  |  |
| *Pole číselníku:* | *Element pole (v XML):* | *Poznámka:* |
| Název | **title** |  |
| Popis | **descr** |  |
| Specifická popisná metadata | **defBlockXsd** | Odkaz na XSD definici použitou pro plnění specifických popisných metadat daného fondu. Nemusí se používat. |

|  |  |
| --- | --- |
| *Název číselníku:* | **Stupeň důvěryhodnosti** |
| *Element (v XML struktuře):* | **cre-cntrls** | *Element položky (v XML struktuře):* | **cre-cntrl** |
| Označení "nomenclatureAcronyme" pro metodu "getNomenclatures"  | **CreCntrl** |  |  |
| *Pole číselníku:* | *Element pole (v XML):* | *Poznámka:* |
| Název | **title** |  |
| Popis | **descr** | Nepovinné. |
| Zkratka | **acr** | Specifické hodnoty jsou použité v konfiguraci systému pro řízení spoučtění funkcí k zajištění důvěryhodnosti. |

|  |  |
| --- | --- |
| *Název číselníku:* | **Umístění analogové podoby dokumentu** |
| *Element (v XML struktuře):* | **Locs** | *Element položky (v XML struktuře):* | **loc** |
| Označení "nomenclatureAcronyme" pro metodu "getNomenclatures"  | **Loc** |  |  |
| *Pole číselníku:* | *Element pole (v XML):* | *Poznámka:* |
| Úplná zkratka umístění | **fullyQcc** |  |
| Název | **title** |  |
| Popis | **descr** | Nepovinné. |

|  |  |
| --- | --- |
| *Název číselníku:* | **Skartační režim** |
| *Element (v XML struktuře):* | **rd-cntrls** | *Element položky (v XML struktuře):* | **rd-cntrl** |
| Označení "nomenclatureAcronyme" pro metodu "getNomenclatures"  | **RdCntrl** |  |  |
| *Pole číselníku:* | *Element pole (v XML):* | *Poznámka:* |
| Kód skartačního režimu | **acr** | Pokud není režim označen jinak, obsahuje skartační znak a lhůu (např. „A10“,“V15“,“S5“)  |
| Název | **title** | Pokud není režim nazván jinak, obsahuje skartační znak a lhůu (např. „A10“,“V15“,“S5“) |
| Popis | **descr** | Není nutné používat. |
| Skartační znak | **action** | Jeden ze znaků A/S/V |
| Externí skartační událost | **userPeriod** | Obsahuje hodnotu true, pokud se lhůta nepočítá na základě data uzavření, ale na základě jiné skutečnosti. |
| Právní důvod | **mand** | Právní důvod aplikace skartačního režimu. Není nutné používat. |
| Skartační lhůta | **period** | Skartační lhůta v celých letech. |
| Věcný důvod | **reas** | Věcný důvod aplikace skartačního režimu. Není nutné používat. |
| Spouštěcí událost – typ | **eventtype** | Typ spouštěcí události:* „DateClosed“
* „DateCreated“
* „DateProcessed“
* „Custom“
 |
| Spouštěcí údálost – popis | **eventdescr** | Popis spouštěcí události. Vyplnění je povinné, pokud je typ spouštěcí události „Custom“. |
| Dědění | **inherit** | Obsahuje hodnotu „true“, pokud se režim dědí v rámci hierarchie entit spisovny. |

|  |  |
| --- | --- |
| *Název číselníku:* | **Formáty souborů (komponent)** |
| *Element (v XML struktuře):* | **Pronoms** | *Element položky (v XML struktuře):* | **Pronom** |
| Označení "nomenclatureAcronyme" pro metodu "getNomenclatures"  | **Pronom** |  |  |
| *Pole číselníku:* | *Element pole (v XML):* | *Poznámka:* |
| Unikátní identifikátor | **puid** | Přiděluje PRONOM, označuje konkrétní verzi formátu. |
| Název | **title** |  |
| Popis | **descr** |  |
| Typ MIME | **mime** | Není unikátní. |
| Přípona souboru | **suffix** | Není unikátní. |
| Doporučené použití dle PRONOM | **state** |  |

|  |  |
| --- | --- |
| *Název číselníku:* | **Organizační jednotky** |
| *Element (v XML struktuře):* | **org-units** | *Element položky (v XML struktuře):* | **org-unit** |
| Označení "nomenclatureAcronyme" pro metodu "getNomenclatures"  | **OrgUnit** |  |  |
| *Pole číselníku:* | *Element pole (v XML):* | *Poznámka:* |
| Kód organizační jednotky | **acr** | Kompletní kód org. jednotky |
| Název | **title** | Název |
| Popis | **descr** | Popis |
| Nadřízený identifikátor | **parentErmsUid** | Identifikátor zdrojového systému |
| Identifikátor | **ermsUid** | Jedinečný identifikátor zdrojového systému |

|  |  |
| --- | --- |
| *Název číselníku:* | **Lokační objekty**  |
| *Element (v XML struktuře):* | **loc-details** | *Element položky (v XML struktuře):* | **loc-detail** |
| Označení "nomenclatureAcronyme" pro metodu "getNomenclatures"  | **LocDetail** |  |  |
| *Pole číselníku:* | *Element pole (v XML):* | *Poznámka:* |
| Název | **title** | Název |
| Popis | **descr** | Popis |
| Nadřazený lokační objekt | **parent** | Plný kód nadřazeného objektu |
| Kód | **code** | Kód lokačního objektu bez kódu nadřazeného objektu |
| Plný kód | **full-code** | Plný kód objektu |
| Spisovna/lokace | **loc** | Kód lokace |
| Typ objektu | **type** | Typ objektu |
| Rozměr - délka | **space-length** |  |
| Rozměr - výška | **space-heighth** |  |
| Rozměr - hloubka | **space-depth** |  |