
API ROZHRANÍ SYSTÉMU DMS: WS_API_DMA

Projekt

**Datová integrace a implementace workflow procesů
(Projekt DMS)**



PROFINIT
new frontier group

Zhotovitel: Profinit, s. r. o.

Verze dokumentu: 3.4.0

Datum: 18. 5. 2015



Popis API rozhraní systému DMA

Projekt DMS

PROFINIT
new frontier group

Informace o dokumentu

Vedoucí projektu	Mgr. Ondřej Stibůrek	Vedoucí projektu	Mgr. Tomáš Mojžiš
Objednatel:		Zhotovitel:	
Projekt:	DMS – možnost přímého volání API pro technické účty	Verze dokumentu:	3.4.0
Fáze projektu:	Nasazeno v produkčním prostředí	Datum dokumentu:	18.05.2015

Historie verzí

Číslo verze	Datum verze	Vypracoval	Popis	Jméno souboru
1.0	30.07.2006	EXPRIT	Úvodní verze	
1.1	30.08.2006	EXPRIT	Doplnění verze 1.1	WS_API_DMA_060830.doc
1.2	21.09.2006	EXPRIT	Doplnění změn verze	WS_API_DMA_060921.doc
1.2	26.09.2006	EXPRIT	Doplnění hodnot číselníků	WS_API_DMA_060926.doc
1.3	12.10.2006	EXPRIT	Doplnění DIS Port	WS_API_DMA_061013.doc
1.4	6.12.2006	EXPRIT	Doplnění KNS port, změny login metody, další změny API	WS_API_DMA_061206.doc
1.5	13.3.2007	ICZ	Doplnění změn v ZDD portu	WS_API_DMA_070313.doc
1.6	02.04.2007	ICZ	Doplnění změn v ZDD a DMS portu	WS_API_DMA_070402.doc
1.7	12.06.2007	ICZ	Přidání popisu NumberGenerátoru (původně byl v samostatném dokumentu)	WS_API_DMA_070612.doc
1.7	20.06.2007	ICZ	Doplnění SysPort, úpravy BTS pro nový způsob řízení vytěžování OCR (P/O).	WS_API_DMA_070620.doc
1.8	7.10.2008	Empire s.r.o.	Nové operace pro získání skupiny OSSZ	WS_API_DMA_081007.doc
2.8	12.11.2009	SIS (Holan, Slepánek)	Úpravy související s nasazením Buildu DMS 2009, integrace s DIS systémem a Agendovým přístupem	WS_API_DMA_091112.doc
	26.1.2009	Langová	Jazyková korektura	WS_API_DMA_v2_9_100126.doc
2.9	12.2.2010	Slepánek	Zpracování připomínek	WS_API_DMA_v2_9_100212.doc
3.0	22.11.2010	SIS (Holan, Hrubý)	Úpravy související s přidáním funkce více těl	WS_API_DMA_v3.0_101122.doc
3.1	18.02.2011	SIS (Slepánek)	Úpravy související s povýšením verze IBM Content Manager 8.4.2 FP2	WS_API_DMA_v3.1_110218.doc
3.2	19.09.2012	Atos (Slepánek)	Aktualizace dokumentu	WS_API_DMA_v3.2_120919.doc
3.3	31.10.2013	Profinit (Mojžiš)	Aktualizace dokumentu, přidání nového portu ESB a metod	WS_API_DMA_v3.3_131031.doc
3.4.0	18.05.2015	Profinit	Aktualizace dokumentu, nový způsob přihlašování	CSSZ_DMS_WS_API_DMA_v3_4_0_20150518.doc



Obsah

1. ÚVOD	6
2. RELEASE NOTES	6
2.1. Úpravy rozhraní související s integrací s AAA Portálem	7
2.2. Změna zamykání dokumentu žádosti pro ZDD	7
3. SLUŽBY VYUŽÍVANÉ OBLASTÍ DMA	8
4. SLUŽBY ROZHRAŇÍ WS_API_DMA (POSKYTOVANÉ DMA)	8
4.1. Seznam služeb rozhraní WS_API_DMA	8
4.2. Chybová hlášení	12
4.3. Základní sekvence použití služeb pro práci s dokumenty	13
4.3.1. Založení dokumentu	14
4.3.2. Vyhledání dokumentu, zobrazení jeho metadat, zobrazení jeho těla (souboru)	14
4.3.3. Nahrazení těla dokumentu	15
4.3.4. Vytvoření vazby mezi dvěma dokumenty	16
4.3.5. Získání kódů OSSZ patřících do stejné skupiny jako vstupní OSSZ	16
4.4. Základní sekvence použití služeb pro práci s Novými žádostmi	18
4.4.1. vytvořit nové ZDD, uložit přílohy k ZDD	18
4.4.2. vyhledat ZDD k editaci, uložit změněné ZDD	18
4.4.3. vyhledat ZDD a k ní všechny přílohy	20
4.5. Základní sekvence použití služeb pro práci s Podáním DIS	21
4.5.1. Vytvořit nové podání DIS	21
4.5.2. Vyhledání podání DIS, zobrazení jeho metadat, zobrazení jeho těla (souboru)	21
4.5.3. Nahrazení těla podání DIS	22
4.6. Přímé volání API bez loginu pro technické účty	22
4.7. Služby API generátoru evidenčních čísel	23
4.7.1. Číselné řady	23
4.7.2. Dosud používané řady evidenčních čísel	24
4.7.3. API generátoru evidenčních čísel	24



4.8. Implementace v ČSSZ	24
5. PŘÍKLADY POUŽITÍ ROZHRANÍ WS_API_DMA	25
5.1. Login port	25
5.1.1. Metoda login	25
5.1.2. Metoda logout	25
5.1.3. Metoda refreshToken	25
5.1.4. Metoda getToken	25
5.2. DMS Port	26
5.2.1. Metoda search	26
5.2.2. Metoda retrieve	26
5.2.3. Metoda disable	27
5.2.4. Metoda enable	27
5.2.5. Metoda history	27
5.2.6. Metoda create	28
5.2.7. Metoda update	30
5.2.8. Metoda getOsszGroupCode	30
5.2.9. Metoda getOsszCodesByOsszGroupCode	31
5.3. KNS Port	31
SpusteniKNS	31
5.3.1. Metoda zmenaKlasifikace	31
5.3.2. Metoda resken	32
5.3.3. Metoda zmenaEvidencnihoCisla	32
5.3.4. Metoda zmenaZDV	32
5.3.5. Metoda zmenaPrav	32
5.3.6. Metoda nestandardniStav	33
5.4. Použití rozhraní generátoru evidenčních čísel	33
SequenceGenerator	33
Metoda getNextById	33
Metoda getNumbersById	33
5.5. GenericDMSPort	34
5.5.1. Metoda history	34
5.5.2. Metoda disable	34
5.5.3. Metoda enable	35
5.5.4. Metoda delete	35
5.5.5. Metoda getRelated	35
5.5.6. Metoda getItemTypes	35
5.5.7. Metoda getSearchAttributes	35
5.5.8. Metoda getItem	36
5.5.9. Metoda getItem	36
5.5.10. Metoda getItemInFolder	37
5.5.11. Metoda getItemInFolder	37
5.5.12. Metoda create	38
5.6. DIS Port	38
5.6.1. Metoda search	38
5.6.2. Metoda retrieve	39



5.6.3.	Metoda disable	40
5.6.4.	Metoda enable	40
5.6.5.	Metoda history	40
5.6.6.	Metoda create	41
5.6.7.	Metoda update	42
5.7.	ESB Port	43
5.7.1.	Metoda searchESB	43
5.7.2.	Metoda retrieveESB	43
5.7.3.	Metoda createESB	43
5.7.4.	Metoda updateESB	43
6.	POŽADAVKY NA OKOLNÍ SYSTÉMY PŘÍSTUPUJÍCÍ DO DMS	44
6.1.	Předpoklady implementace vzájemné komunikace	44
6.2.	Implementace	45
6.3.	Součinnost po nasazení do provozu	46
7.	PŘÍLOHA 1. SYSTÉM DMS - PŘEHLED	47
7.1.	Přehled	47
7.1.1.	Architektura IBM Content manager	47
7.1.2.	Implementace na ČSSZ – systém DMA	48
7.1.3.	Klientské aplikace DMA	48
7.2.	Dokument	49
7.2.1.	Logický datový model dokumentu	49
7.2.2.	Profil dokumentu – atributy dokumentů uložených v DMA	50
7.2.3.	Řízení práv přístupu k dokumentům	53
8.	PŘÍLOHA 2. ČÍSELNÍKY	54
9.	PŘÍLOHA 3. ČÍSELNÍK TYPŮ TĚL	57



1. Úvod

V tomto dokumentu je zdokumentováno rozhraní WS_API_DMA systému DMS. Jedná se o rozhraní a služby, které jsou využívány ostatními oblastmi pro ukládání vyhledávání a zobrazení dokumentů. Úplný popis rozhraní a služeb DMS je předmětem projektu DMS s názvem Datová integrace a implementace workflow procesů a dalších navazujících projektů ZPO, ZDD, ...

2. Release Notes

Verze API/ verze WASE rveru	Vydáno dne	Instalován o do vývojového dne	Instalová no do testovací ho/nově pilotního ČSSZ dne	Instalován o do produkce dne	Změny
0.7	30.1.2006				Pilot A
1.0	30.6.2006			30.6.2006	Doplnění pro pilot B – Primární verze
1.1	30.8.2006				Doplnění ZDD Proxy
1.2	21.9.2006		21.9.2006	Plán 15.10.2006 posunuto na 20.10.2006	Doplnění DMS_UpdateBody, DMS_deleteBody, změna návratových hodnot DMS_retrieve, doplnění vrácení názvu souboru
1.3	12.10.2006		13.10.2006	ne	Doplnění DIS Proxy
1.4	05.12.2006	06.12.2006 odpoledne	ne	únor.2007	1.Doplnění KNS portu pro kompenzaci nestandardních stavů, 2.Změny v souvislosti s Integrací do AAA Portálu – doplnění metody getToken (jako náhrada za Login), 3.Změna wsdl portu DMS – přidání atributu ID typuFormuláře a doplnění oprávnění! 4.Změna namespace ve wsdl, použití elementů ve wsdl
1.5	13.3.2007	13.3.2006	Plán 19.3.2007	13.4.2007	Přidání metod checkin, checkout pro ZDD
1.6 / 2.4.10	02.04.2007	2.4.2006	18.5.2007	25.5.2007	Přidání metod „getCount“ na získání počtu dokumentů odpovídajících zadaným kritériím pro ZDD, stejná metoda doplněna i pro DMS port.
1.7 / 2.4.11	6.6.2007	6.6.2006	12.11.2007	12.12.2007	Doplnění SYS Portu, úpravy BTS pro nový způsob řízení vytěžování OCR (P/O), rozšíření atributů vyhledávání dokumentů
1.8 / 2.5. 1	1.10.2008	1.10.2008	5.11.2008	5.12.2008	Rozšíření atributů vyhledávání, získání kodu skupiny OSSZ a vrácení seznamu OSSZ na základě skupiny OSSZ.
2.8.1	1.10.2009	1.10.2009	5.12.2009	12.11.2009	Implementace DISPortu, úpravy související s Agendovým přístupem.
3.0.0	16.09.2010	16.09.2010	4.11.2010	25.11.2010	Implementace funkce více těl
3.1.0	18.02.2011	18.10.2010	6.12.2010	22.01.2011	Povýšení verze IBM Content Manager 8.4.2 FP2



3.2.0	19.09.2012	19.09.2012	20.09.2012	01.10.2012	Aktualizace dokumentu
3.3.0	12.09.2013				Doplnění ESB Portu
3.4.0	18.05.2015				Alternativní způsob přihlášení

2.1. Úpravy rozhraní související s integrací s AAA Portálem

V souvislosti s přechodem autentikace uživatelů pomocí AAA Portálu se mění i komunikace s WS_API_DMA od externích aplikací (NEM, ZDD, a dalších).

Princip práce s DMA zůstává stejný – každá aplikace musí pro práci s DMA nejprve získat tzv. „token“, který pak posílá (jako identifikaci získané session) při použití všech metod pro práci s daty/dokumenty DMA.

Změna spočívá ve způsobu získání tokenu:

1. Každá aplikace komunikující s WS_API_DMA musí v hlavičce SOAP protokolu mít uvedeno Autorizační parametry uživatele, který operaci požaduje. (Náhradní řešení je, že aplikace používá své dohodnuté servisní Autorizační parametry – např. SRV_357). Pokud Autorizační parametry nebudou v SOAP hlavičce dostupné, WS_API_DMA volání odmítne s návratovým kódem: „exception.AuthorizationParameters.Missing“.

Konkrétní podoba parametru struktury Autorizačních parametrů:

Pro zajištění autorizace požadavku v rámci komunikace mezi servisními aplikacemi se bude používat struktura - UserNameToken, která se bude vkládat do SOAP hlavičky. Struktura - UserNameToken je definovaná ve specifikaci UsernameToken Profile 1.0 OASIS Standard 200401, March 2004:

```
<S11:Envelope xmlns:S11="..." xmlns:wsse="...">
  <S11:Header>
    ...
    <wsse:Security>
      <wsse:UsernameToken>
        <wsse:Username>iv-user</wsse:Username>
        <wsse:Password></wsse:Password>
        </wsse:UsernameToken>
      </wsse:Security>
    ...
  </S11:Header>
  ...
</S11:Envelope>
```

Kde,

namespace S11 - je <http://schemas.xmlsoap.org/soap/envelope/>

namespace WSSE - je <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd>

V této struktuře se bude nastavovat pouze parametr Username, parametr Password zůstane prázdný.

2. Pro získání tokenu volá aplikace metodu „getToken“ s parametrem serverID. Metoda vrací objekt token identifikující získanou session. (Před integrací s AAA Portálem se používala metoda login kde identifikační údaje (jméno a heslo) byly v parametrech volání metody). Tento objekt se v nezměněné podobě používá v dalších voláních metod WS_API_DMA. Pro ukončení session se i nadále používá metoda logout.

2.2. Změna zamykání dokumentu žádosti pro ZDD

Aplikace ZDD pracuje tak, že pokud přechází referent do editačního módu žádosti, je žádost pro něj aplikací ZDD zamknuta (metodou checkout). Během práce s žádostí může několikrát zavolat metodu update. Po tuto



dobu je žádost stále uzamčená. Odemknutí žádosti (metodou checkin) provede aplikace ZDD při opouštění editačního módu žádosti.

Upozornění: tento mechanismus je pouze pro ZDD. Pro ostatní dokumenty si nadále zamykání a odemykání řídí metody API a ne aplikace.

3. Služby využívané oblastí DMA

DMA využívá pro svoji potřebu služby z jiných oblastí IS uvedené v následujícím seznamu (kromě služeb infrastruktury).

Požadavky na služby ostatních modulů (VYUŽÍVÁ oblast DMS)				
Popis požadavku	Typ	Modul	Rozhraní	Kód služby
Autorizace uživatele	Čtení	Active Directory		
Informace o subjektu (dle RČ, VS, IČO)	Čtení	KE		
Detaily adresy (dle id adresy)	Čtení	KE		
Načtení datové věty ze ZDV	Čtení	ZDV	wszdv	
Získání URL adresy image (pouze pro image uložené v ZDV)	Čtení	Image Archiv		
Získání image pro zobrazení (pouze pro image uložené v ZDV)	Čtení	Image Archiv		
Předání URL adresy image pro podložení šablonou formuláře (pouze pro OCR formuláře)	Zápis	Image Server		
Spuštění procesu BTS – předání k vytěžení OCRFormuláře	Zápis	BizTalk Server		
Spuštění odborného workflow procesu	Zápis	General Interface BTS		

4. Služby rozhraní WS_API_DMA (poskytované DMA)

Technologie DMA poskytuje navenek služby, které mohou ostatní oblasti – odborné aplikace, resp. jejich uživatelé, využívat.

Tyto služby jsou implementovány v programovém rozhraní API s názvem WS_API_DMA. Základními funkcemi tohoto rozhraní je:

- Zaevidování dokumentu/ objektu (vytvoření nového profilu dokumentu)
- Vložení těla (nebo více těl) dokumentu/ objektu (souboru s dokumentem),
- vyhledání dokumentu/dokumentů podle hodnot atributů,
- získání dokumentu/ objektu a jeho profilu pro jeho zobrazení, tisk, ...(obecně prezentaci),
- změna profilu určitého dokumentu/ objektu (např. změna atributů, zneplatnění a další),
- změna těla dokumentu/ objektu.

4.1. Seznam služeb rozhraní WS_API_DMA

Následující tabulka uvádí přehled zveřejněných služeb WS_API_DMA.

Seznam služeb rozhraní SDD::DMS (POSKYTUJE modul DMS)				
Port Type	Služba	Její činnost	parametry	poznámka
Login	login	Provede přihlášení uživatele do systému.	serverID, jméno, heslo	Po integraci s AAAPortálem nahrazena metodou



Popis API rozhraní systému DMA

Projekt DMS

PROFINIT
new frontier group

				getToken. Dále nepoužívat, časem bude odstraněna!
Login	logout	Provede odhlášení uživatele ze systému	Vstupuje Token vytvořen předcházejícím voláním metody login	
Login	refreshToken	Obnoví token		
Login	getToken	Vrátí objekt user token	serverID	Po integraci s AAAPortálem nahrazuje metodu login
DMS	search	Vyhledá dokumenty odpovídající vyhledávacím kritériím		
DMS	retrieve	Získá dokument specifikovaný jeho UUID		
DMS	disable	Zneplatní dokument		
DMS	enable	Zruší zneplatnění dokumentu		
DMS	history	Získá výpis historie dokumentu		
DMS	create	Provede operaci evidence dokumentu		
DMS	update	Provede změny v dokumentu		
DMS	getRelated	Vrátí uuid souvisejících dokumentů		
DMS	createRelation	Vytvoří link mezi dvěma dokumenty		
DMS	deleteRelation	Odstraní link mezi dvěma dokumenty		
DMS	uploadBody	Přidá tělo dokumentu. Pokud má tělo vyplněno uuid těla provede se update již založeného těla.		
DMS	deleteBody	Odstraní tělo dokumentu		
DMS	getCount	Vrátí počet dokumentů, které odpovídají vyhledávacím kritériím	Vyhledávací kritéria jsou obdobná jako pro metodu search	je třeba aplikačně zajistit, aby se tyto metody nespouštěly často, protože hodně zatěžují databázi. Určitě by neměly být dostupné běžným uživatelům, kdykoli.
DMS	getOsszGroupCode	Vrátí kod skupiny OSSZ na základě OSSZ	Kod OSSZ	
DMS	getOsszCodesByOsszGroupCode	Získá seznam OSSZ na základě kodu skupiny OSSZ	Kod Skupiny OSSZ	
ZDD	search	Vyhledá ZDD odpovídající vyhledávacím kritériím		



Popis API rozhraní systému DMA

Projekt DMS

PROFINIT
new frontier group

ZDD	retrieve	Získá ZDD specifikovaný jeho UUID		
ZDD	create	Provede operaci evidence ZDD		Vytvořené ZDD je uzamčeno!
ZDD	update	Provede změny v ZDD dokumentu		ZDD musí být před voláním metody uzamčen!
ZDD	disable	Zneplatní dokument		
ZDD	enable	Zruší zneplatnění dokumentu		
ZDD	history	Získá výpis historie dokumentu		
ZDD	createLink	Vytvoří link ze žádosti k příloze		
ZDD	deleteLink	Smaže link ze žádosti k příloze		
ZDD	getPrilohy	Na základě uuid žádosti vrátí seznam UUID příloh žádosti (pro dotažení detailů příloh použijte serivce DMS)		
ZDD	getZadost	Na základě uuid přílohy vrátí uuid žádosti		
ZDD	checkout	Uzamkne zadost pro uživatele		
ZDD	checkin	Odemkne žádost		
ZDD	getZDDCount	Vrátí počet žádostí, které odpovídají vyhledávacím kritériím	Vyhledávací kritéria jsou obdobná jako pro metodu search	je třeba aplikačně zajistit, aby se tyto metody nespouštěly často, protože hodně zatěžují databázi. Určitě by neměly být dostupné běžným uživatelům, kdykoli.
BTS	POLToDMS	Uložit výsledek OCR do DMS		
DIS	createDIS	Provede operaci evidence podání DIS		
DIS	deleteBodyDIS	Odstraní z dokumentu podáníDISUUID telo teloUUID		
DIS	disableDIS	Zneplatní podání DIS		
DIS	enableDIS	Zruší zneplatnění podání DIS		
DIS	historyDIS	Získá výpis historie podání DIS		
DIS	retrieveDIS	Získá podání specifikované jeho UUID		
DIS	searchDIS	Vyhledá podání DIS odpovídající vyhledávacím kritériím		



Popis API rozhraní systému DMA

Projekt DMS

PROFINIT
new frontier group

DIS	updateDIS	Provede změny v podání DIS		
DIS	uploadBodyDIS	Přidá těla podání DIS. Pokud má tělo vyplněno uuid těla provede se update již založeného těla.		
KNS	zmenaKlasifikace	Spustí proces pro změna agendy a, nebo typu dokumentu	uuid nebo evidenční číslo, ageda, typ	
KNS	resken	Spustí proces resken	uuid nebo evidenční číslo	
KNS	zmenaEvidencnih oCisla	Spustí proces na změnu evidenčního čísla	uuid nebo evidenční číslo	
KNS	zmenaZDV	Spustí proces na změnu ZDV	uuid nebo evidenční číslo	
KNS	zmenaPrav	Spustí proces změny přístupových práv	uuid nebo evidenční číslo	
KNS	nestandardniStav	Spustí proces pro obecnou chybu	uuid nebo evidenční číslo, poznámka	
sequenceG enerator	getNextById	Vygeneruje nové číslo	id číselné řady	
sequenceG enerator	getNumbersById	Zarezervuje daný počet čísel. Vráť první a poslední zarezervované číslo	id číselné řady, požadovaný počet čísel	
sequenceG enerator	getNextByName	Vygeneruje nové číslo	id aplikace, název číselné řady	Tato služba je použitelná pouze pro DMA aplikace (není veřejná). Zde uvedeno pouze pro úplnost.
sequenceG enerator	getSequenceById	Vráť definici číselné řady	id číselné řady	Tato služba je použitelná pouze pro DMA aplikace (není veřejná). Zde uvedeno pouze pro úplnost.
sequenceG enerator	getSequenceByNa me	Vráť definici číselné řady	id aplikace, název číselné řady	Tato služba je použitelná pouze pro DMA aplikace (není veřejná). Zde uvedeno pouze pro úplnost.
sequenceG enerator	getSequenceNam es	Vráť jména číselných řad definovaných pro aplikaci	id aplikace	Tato služba je použitelná pouze pro DMA aplikace (není veřejná). Zde uvedeno pouze pro úplnost.
sequenceG enerator	getSequences	Vráť jména číselných řad definovaných pro aplikaci	id aplikace	Tato služba je použitelná pouze pro DMA aplikace (není veřejná)
sequenceG enerator	registerSequence	Zaregistruje novou číselnou řadu	id aplikace, název číselné řady, vzor,	Tato služba je použitelná pouze



			délka, perioda resetování, id číselné řady	pro DMA aplikace (není veřejná). Zde uvedeno pouze pro úplnost.
SYS	getCentralImageUrl	Vrátí url na image na centrálním resource manageru	Uuid těla obsahujícího obrázek	sdf
SYS	getLoggedInUsers Count	Vrátí počet aktuálně přihlášených uživatelů		
ESB	createESB	Provede operaci evidence dokumentu		Stejně jako na portu DMS, přidány nové atributy
ESB	updateESB	Provede změny v dokumentu		Stejně jako na portu DMS, přidány nové atributy
ESB	searchESB	Vyhledá dokumenty odpovídající vyhledávacím kritériím		Stejně jako na portu DMS, přidány nové atributy
ESB	retrieveESB	Získá dokument specifikovaný jeho UUID		Stejně jako na portu DMS, přidány nové atributy

4.2. Chybová hlášení

Tabulka obsahuje význam chybových hlášení vrácených WS_API_DMA.

Chybová hlášení		
Hláška	Popis	Možné příčiny
exception.lock.pessimistic	Chyba pesimistického zámku	Objekt je uzamčen jiným uživatelem než který se jej pokouší editovat.
exception.lock.optimistic	Chyba optimistického zámku	Dokument, žádost ZDD i podání DIS obsahují jako jeden z atributů datum změny. Pokud při volání metody update objekt obsahuje jinou hodnotu než jaká je uložena v CM je vrácena tato chybová hláška.
exception.document.invalid-ec	Neplatné evidenční číslo	Zadané evidenční číslo již existuje
exception.document-type.unknown	Neznámý typ dokumentu	Neplatný typ typu dokumentu - není v číselníku
exception.document-agenda- type.unknown	Neplatná kombinace agendy a typu	
exception.document.invalid-uuid	Neplatné UUID dokumentu	Objekt se zadaným UUID neexistuje
exception.integration-ke.nonunique-id	Nejednoznačný výsledek vyhledávání v KE	V této verzi nepoužito
exception.security.invalid-token	Neplatný security token	Vypršel časový limit pro použití user tokenu
exception.document.invalid-sha1	Neplatná sha1	Tělo se zadanou SHA1 sumou už



		existuje. Kontroluje se jen u objektů typu Dokument založených přes DMSPort.
exception.document.cannot-upload-body	Dokument už obsahuje tělo nebo není ve stavu kdy může dostat tělo	Aby bylo možné přidat tělo k dokumentu, musí být ve stavu 10 – čeká na tělo. Kontroluje se jen u objektů typu Dokument založených přes DMSPort.
exception.document-kind.unknown	Neznámý druh dokumentu	CM je nedostupný, atribut obsahuje příliš dlouhé data, typ dat je neplatný, atd.
exception.document-agenda-type.locked		Nad dokumentem je již spuštěn proces změny agendy a typu – počkejte se změnou agendy a typu až tento proces doběhne
exception.document-change-date.unknown		Datum změny dokumentu musí být vyplněn – vyplňuje jej DMS
exception.generic.cm	Chyba CM	
exception.generic	Obecná chyba	
exception.AutorizationParameters.Missing	V SOAP hlavičce nejsou správně nastaveny autorizační parametry	

4.3. Základní sekvence použití služeb pro práci s dokumenty

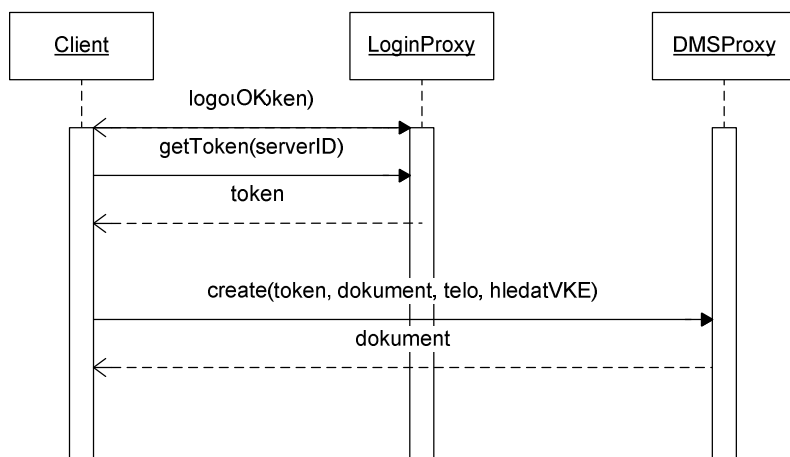
V následujících podkapitolkách jsou uvedeny příklady sekvencí použití služeb pro typické/časté operace s dokumenty.

Poznámky:

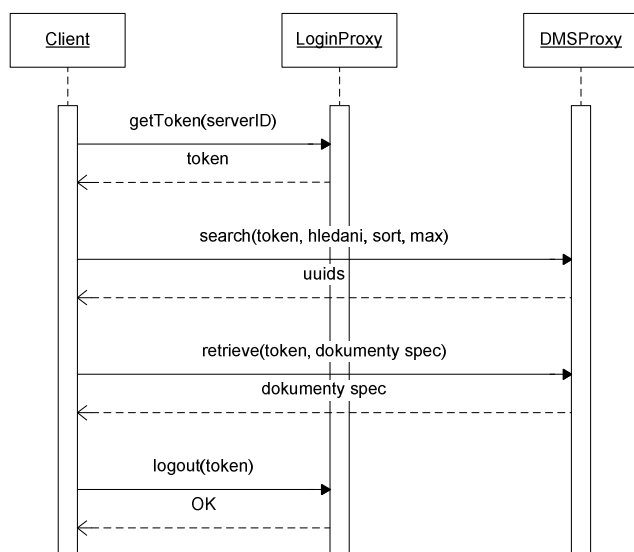
1. Mezi operacemi login a logout může být volání více operací – např. pro vložení několika dokumentů není třeba a není žádoucí po jednom každém provádět logout a další login.
2. Platnost tokenu získaného operací login a používaného pro autorizaci ve všech dalších operacích je časově omezena (nyní na 2 hodiny). Po vypršení se token stane neplatným. Vypršení je možno prodloužit použitím operace refreshToken. Automaticky se token obnovuje po provedení jakékoli operace ve které se daný token použije.



4.3.1. Založení dokumentu

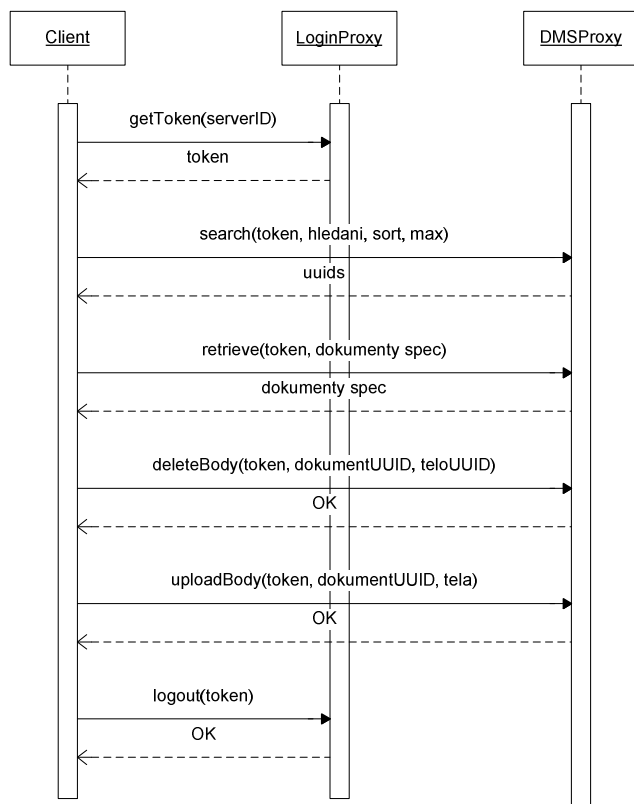


4.3.2. Vyhledání dokumentu, zobrazení jeho metadat, zobrazení jeho těla (souboru)



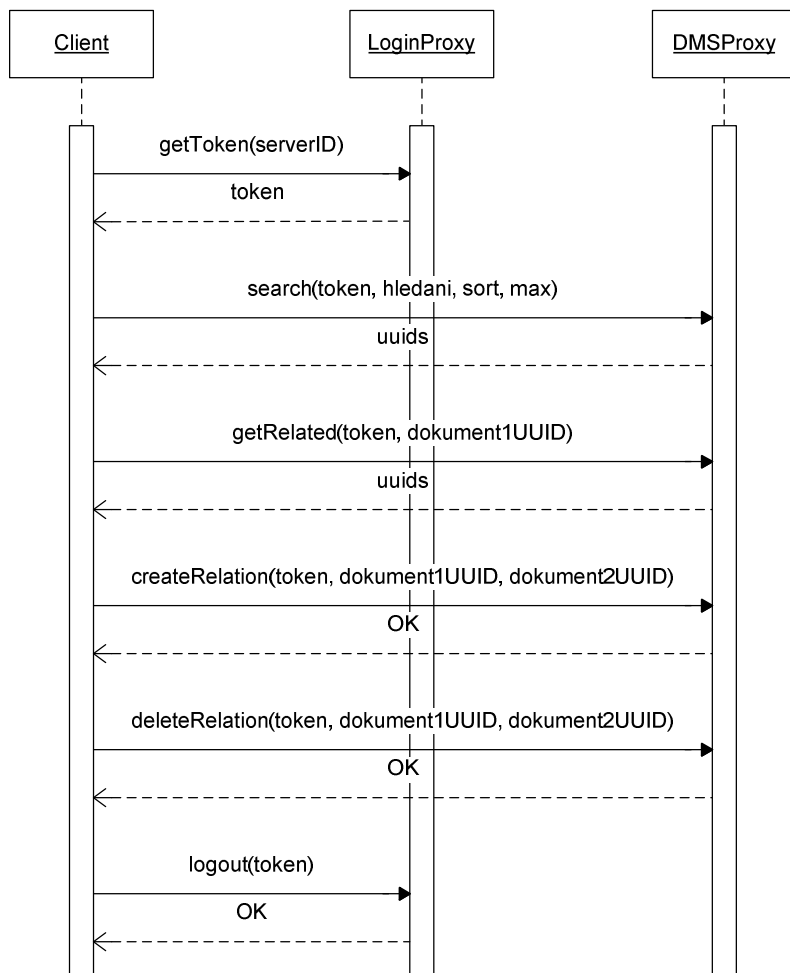


4.3.3. Nahrazení těla dokumentu



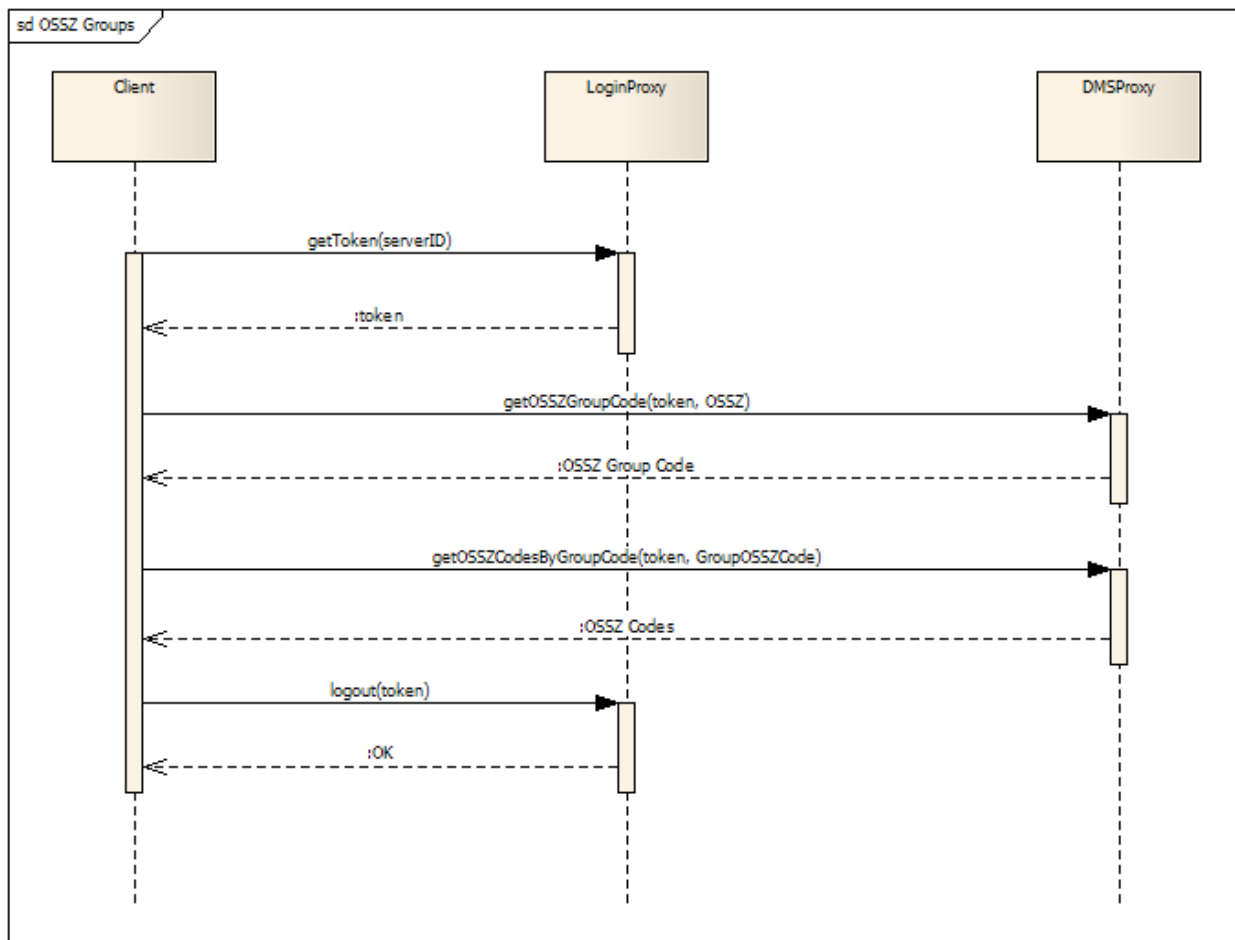


4.3.4. Vytvoření vazby mezi dvěma dokumenty



4.3.5. Získání kódů OSSZ patřících do stejné skupiny jako vstupní OSSZ

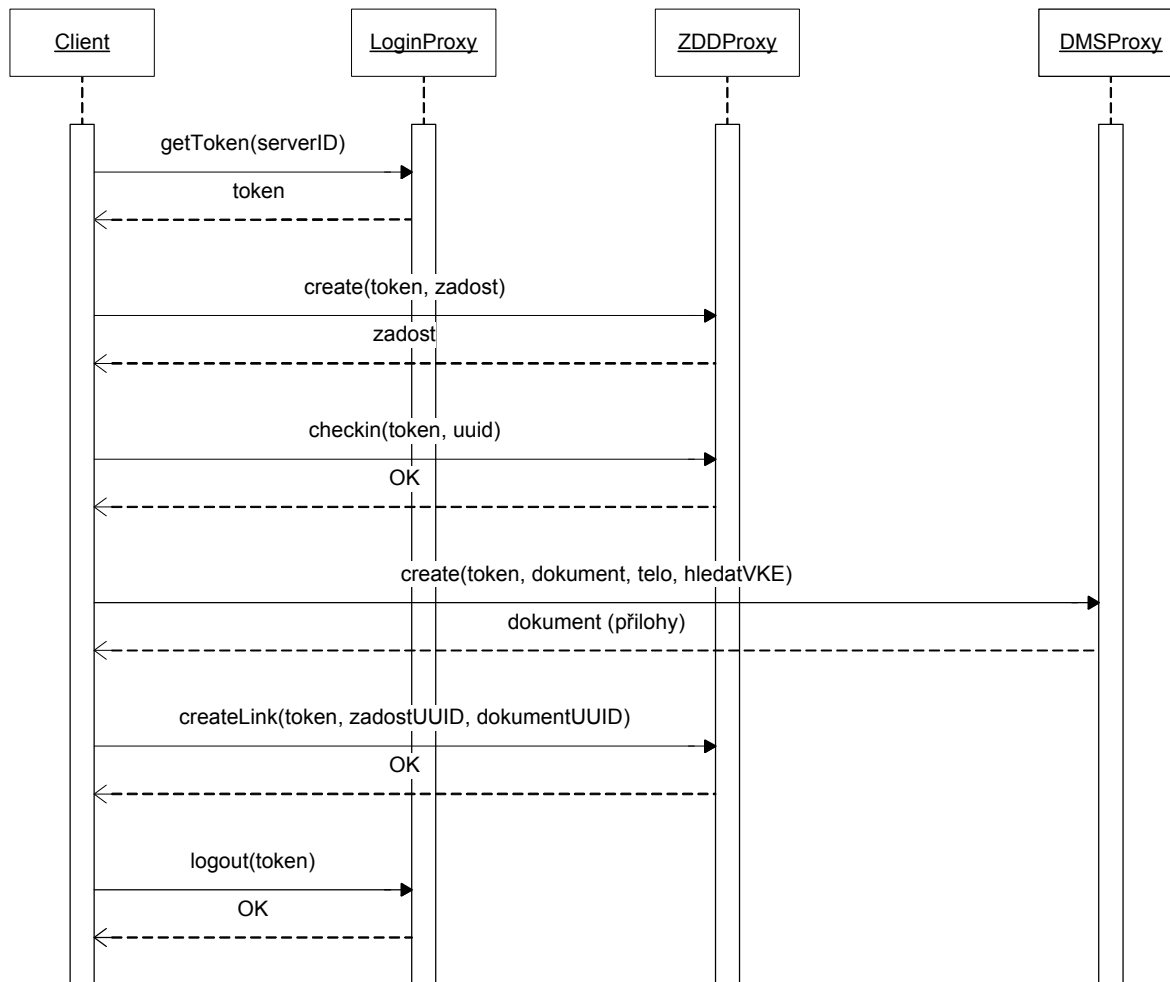
Postup získá kody OSSZ, které patří do stejné skupiny, jako vstupní OSSZ.





4.4. Základní sekvence použití služeb pro práci s Novými žádostmi

4.4.1. vytvořit nové ZDD, uložit přílohy k ZDD



4.4.2. vyhledat ZDD k editaci, uložit změněné ZDD

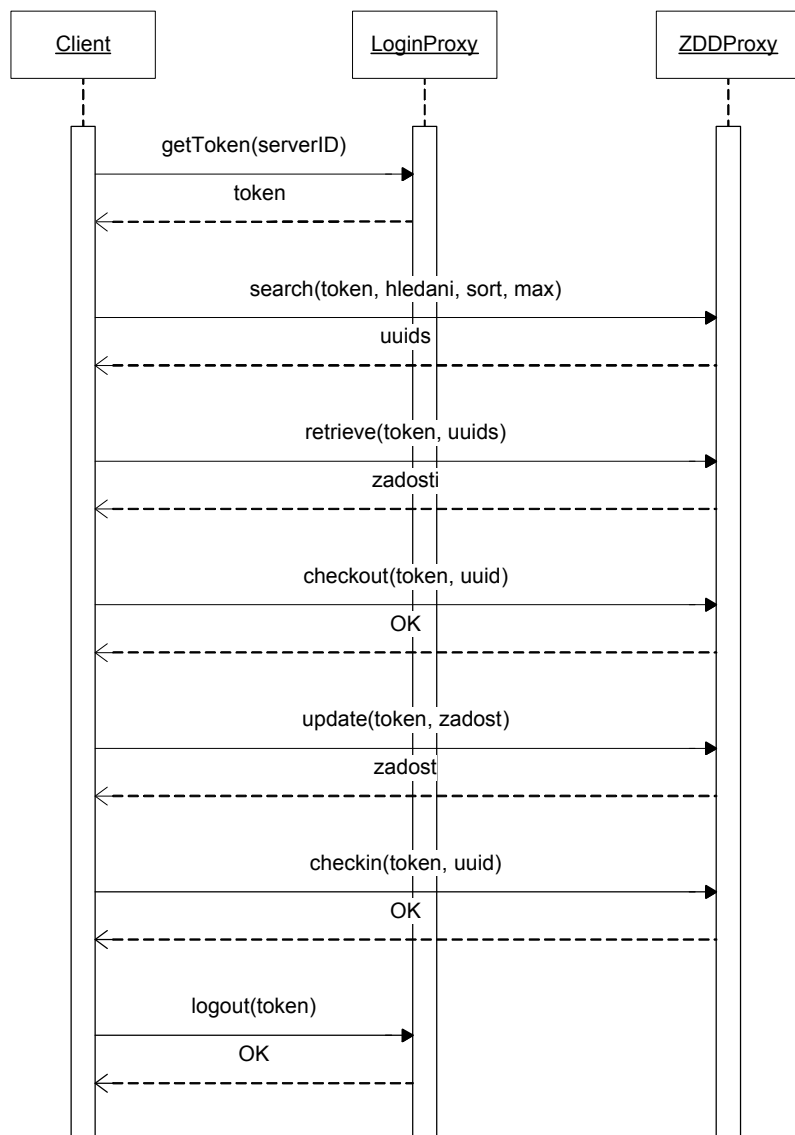
Aplikace ZDD pracuje tak, že pokud přechází referent do editačního módu žádosti, je žádost pro něj checkoutnuta. Během práce s žádostí může několikrát zavolat metodu update. Po tuto dobu je nutné, aby byla žádost stále uzamčená. Odemknutí žádosti se provádí při opouštění editačního módu žádosti.



Popis API rozhraní systému DMA

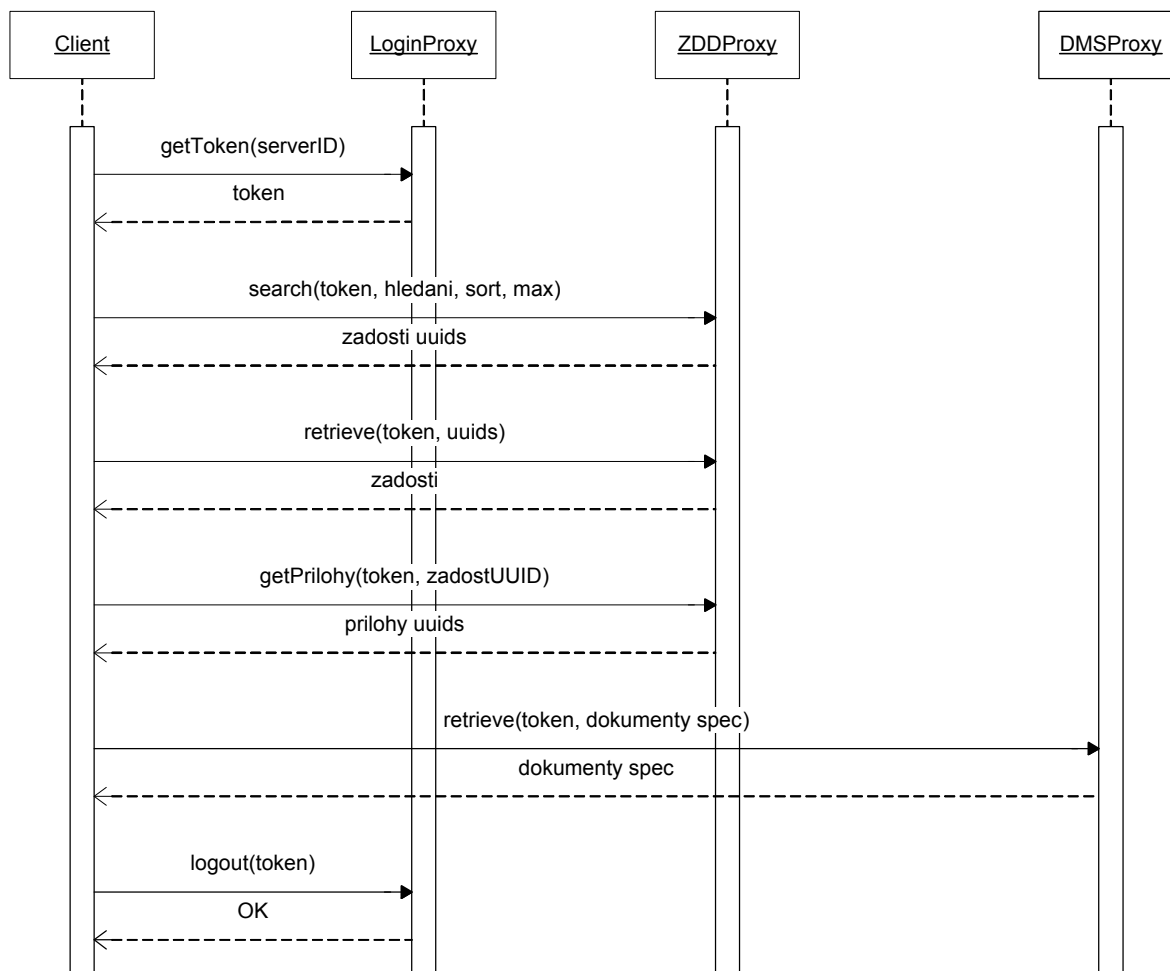
Projekt DMS

PROFINIT
new frontier group





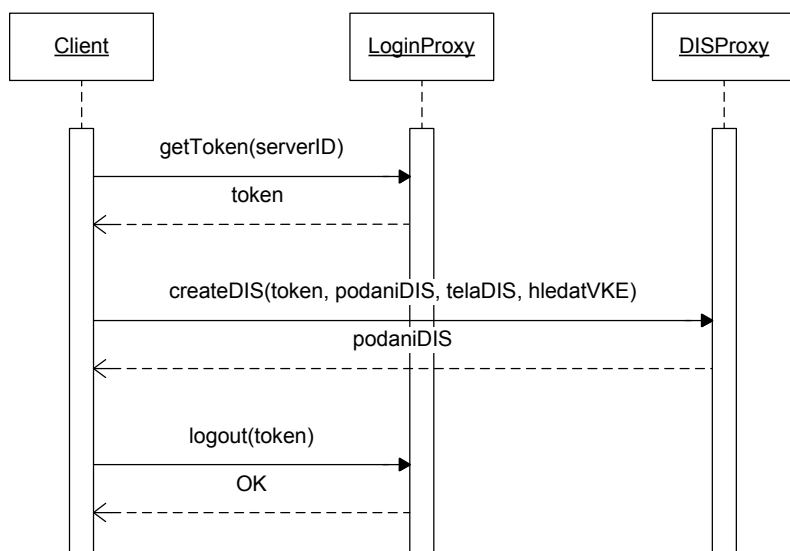
4.4.3. vyhledat ZDD a k ní všechny přílohy



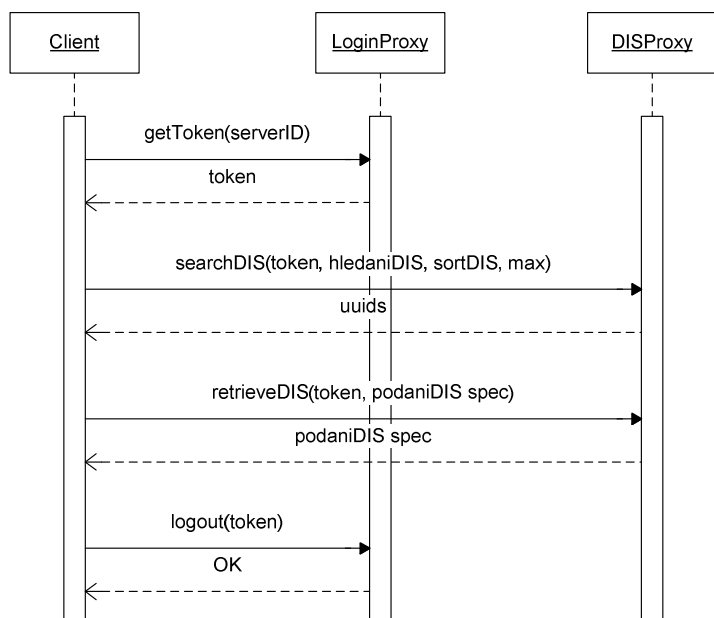


4.5. Základní sekvence použití služeb pro práci s Podáním DIS

4.5.1. Vytvořit nové podání DIS

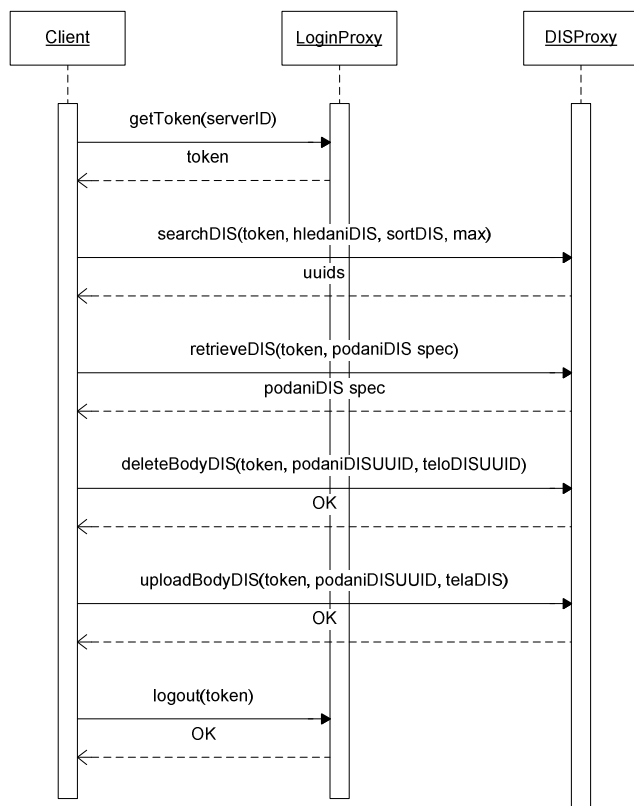


4.5.2. Vyhledání podání DIS, zobrazení jeho metadat, zobrazení jeho těla (souboru)





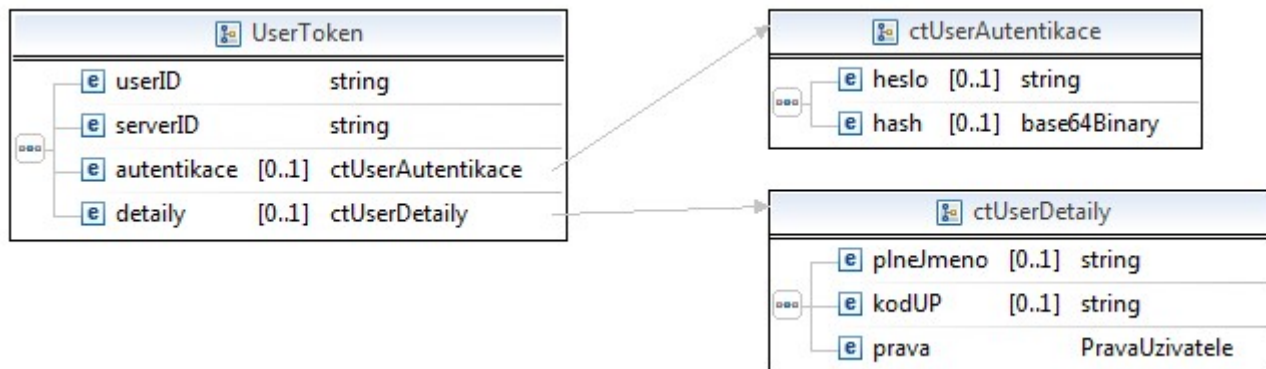
4.5.3. Nahrazení těla podání DIS



4.6. Přímé volání API bez loginu pro technické účty

Pro **technické účty**, které nepoužívají pro přihlášení pomocí AAA, je možné volat API rozhraní přímo bez nutnosti volat nejdříve službu login.

Každá metoda API obsahuje autorizaci pomocí struktury UserToken (viz níže). Jedná se o stejnou strukturu, kterou používají technické účty pro přihlášení v metodě login.



Základní použití spočívá v tom, že pro přihlášení se vyplní jméno a heslo a pro autorizaci v API se používá identifikace session pomocí přiděleného hashe získaného z loginu.

Alternativně je pro technické účty možné nevolat login a zasílat jméno a heslo přímo do volání API. V takovém případě dojde k automatickému přihlášení. Toto přihlášení si server chvíli (v současnosti 1 minutu) drží otevřené a použije se automaticky v případě dalšího volání stejného uživatele.



4.7. Služby API generátoru evidenčních čísel

4.7.1. Číselné řady

Správa číselných řad evidenčních čísel umožňuje definovat různé číselné řady evidenčních čísel a formát čísla generovaný Generátorem (modul NumberGenerator).

Interně má každá číselná řada následujícími parametry:

- Název - název řady,
- id číselné řady – ID používané pro specifikaci řady při volání API Generátoru evidenčních čísel
- Poslední změna – ukazuje datum, kdy bylo z této řady naposled generováno číslo
- Formát - formát generovaného čísla. Generované číslo smí obsahovat pouze číslice. Celkový počet číslic evidenčního čísla nesmí přesáhnout 15. Může obsahovat prefixy, sufixy a hodnoty {year}, {month}, {day}, {id}, nahrazované aktuální hodnotou roku {year}, měsíce {month}, dne {day} a číslem generovaným z numerické řady {id}.
- Reset – určuje periodu, kdy se provede nulování čísla numerické řady {year}, {month}, {day},
- Délka – určuje počet znaků čísla numerické řady.
- Poslední číslo – ukazuje hodnotu posledního použitého čísla numerické řady

Konvence pro nastavení řad Evidenčních čísel

Atribut Evidenční číslo je v DMS u jednotlivých objektů kontrolován na unikátnost.

Aby nedocházelo k duplicitám v generovaných číslech navrhuje se používat následující konvence pro nastavení řad evidenčních čísel:

Použití	formát	Celkový počet míst	
Evid.čísla používaná na štítcích s čárovými kódy - pro rok 2006	KKKyyyy1234567	14	kde kkk je kód ÚP, yyyy je čtyřčíslí letopočtu a 1234567 je sedmimístné číslo.
Evid.čísla používaná na štítcích s čárovými kódy - pro následující roky	KKKyy123456789	14	kde kkk je kód ÚP, yy je dvojčíslí letopočtu a 123456789 je devítimístné číslo
Evid.čísla pro Nové žádosti – formulář. Číslo generuje NumGen ze řady ZAKL	yyyymm1234567	13	Číslo generuje NumGen z řady ZAKL
Evid.čísla pro Nové žádosti – image vytištěné žádosti	9yyyymm1234567	14	Použije se stejné číslo jako pro formulář a na začátek se doplní 9 tak aby evid.číslí na image mělo délku 14 číslic
Evid.čísla pro import z hromadné digitalizace	1xxxxxxxxxxxxxx 2xxxxxxxxxxxxxx 3xxxxxxxxxxxxxx	15	Patnáctimístné řady začínající číslem 1, nebo 2, nebo 3. Číslo generuje datová pumpa při importu.
Evid.čísla pro dokumenty z jiných aplikací pro velký počet dokumentů v jednom roce	8Ayyyy123456789	15	Kde 8 je příznak že se jedná o řadu pro aplikaci, A je kód aplikace, yyyy je čtyřčíslí letopočtu a 123456789 je devítimístné číslo ID. Umožňuje rozlišit až 10 aplikací a evidovat až miliardu dokumentů za jeden rok. Tuto konvenci je nutno použít pro nové řady Evid.čísel
Evid.čísla pro dokumenty z jiných aplikací – do 100 milionů dokumentů v roce	7ayyyy12345678	15	Kde 7 je příznak že se jedná o řadu pro aplikaci, aa je kód aplikace, yyyy je čtyřčíslí letopočtu a 12345678 je osmimístné číslo ID. Umožňuje rozlišit až 99 aplikací a evidovat



až 100 milionů dokumentů za jeden rok.
Tuto konvenci je nutno použít pro nové
řady Evid.čísels

Při tomto způsobu použití formátu evidenčního čísla se jednotlivé řady vzájemně nepřekrývají.

4.7.2. Dosud používané řady evidenčních čísel

Název	ID číselné řady	Formát	Reset	Délka	Poznámka
ZAKL	1	{year}{month}{id}	month	7	Základní řada evidenčních čísel
EXK	3	709{year}{id}	year	8	Řada evidenčních čísel pro projekt Exekuce (výstupní dokumenty ATR_EXK)
CERE	4	751{year}{id}	year	8	Řada evidenčních čísel pro statistiky a reporty CERE
DIS	5	761{year}{id}	year	8	Řada evidenčních čísel pro podání z DIS portálu.
INS	6	790{year}{id}	year	8	Řada evidenčních čísel pro projekt Insolvenční řízení (výstupní dokumenty ATR_INS)

4.7.3. API generátoru evidenčních čísel

API generátoru evidenčních čísel poskytuje navenek dvě služby, které mohou ostatní oblasti využívat pro získání dalšího čísla ze specifikované řady, resp. Pro zarezervování několika dalších čísel z řady.

Port Type	Služba	Její činnost	parametry	Poznámka
sequenceGenerator	getNextById	Vygeneruje nové číslo	id číselné řady	
sequenceGenerator	getNumbersById	Zarezervuje daný počet čísel. Vrátil první a poslední zarezervované číslo	id číselné řady, požadovaný počet čísel	

4.8. Implementace v ČSSZ

Adresy ve vývojovém prostředí Atos:

<http://172.20.2.15:9080/WAServer/services>

<http://172.20.2.15:9080/numberGenerator>

Adresy v pilotním/školicím prostředí ČSSZ:

<http://t-dmsapi.appv.cssz.cz/WAServer/services/...>

<http://t-dmsapi.appv.cssz.cz/numberGenerator>

Adresy na produkčním prostředí jsou:



<http://dmsapi.appv.cssz.cz/WAServer/services/...>
<http://dmsapi3.app.cssz.cz/numberGenerator>

Upozornění:

Služby BTSPort, ZDDPort, KEPort, DISPort jsou specializované a nejsou určeny k běžnému použití.

5. Příklady použití rozhraní WS_API_DMA

5.1. Login port

Port pro získání user tokenu s detailem uživatele, pro odhlášení uživatele a pro obnovení platnosti tokenu.

// tridy vygenerovane WSDL2Java

```
LoginProxy loginProxy = new LoginProxy();  
DMSProxy dmsProxy = new DMSProxy();
```

5.1.1. Metoda login

Provede přihlášení uživatele do systému. Vracený user token obsahuje přístupová práva, územní pracoviště, plné jméno a identifikaci uživatele pro práci s CM (hash). Platnost hashe je časově omezena – uživatel je po určité době automaticky odhlášen.

Upozornění: Po integraci DMS s AAA portálem je tato metoda nahrazena metodou getToken– od verze API 1.4 (cca od 15.12.2006. Dále nepoužívat, bude odstraněna!!!

```
private UserToken prihlasit() {  
    UserToken token = new UserToken(); // vytvori sea nový UserToken  
    token.setUserID("klara"); // uzivatelske jmeno  
    token.setServerID("icmnlbdb"); // server, na který se pripojuje  
    token.setAutentikace(new CtUserAutentikace()); // vytvori se nový objekt autentikace  
    token.getAutentikace().setHeslo("klara"); // nastavi se heslo pro pripojeni  
    token = loginProxy.login(token); // zavola se metoda login  
    return token;  
}
```

5.1.2. Metoda logout

Provede odhlášení uživatele ze systému.

```
private void odhlasit(UserToken token) throws Exception {  
    loginProxy.logout(token); // pouzije se token vygenerovany prihlasenim  
}
```

5.1.3. Metoda refreshToken

Obnoví platnost tokenu uživatele. Platnost tokenu se obnoví při každé operaci s user tokenem, každá metoda jejíž vstupní parametr obsahuje user token obnoví jeho platnost.

```
private void obnovitToken(UserToken token) throws Exception {  
    loginProxy.refreshToken(token); // pouzije se token vygenerovany prihlasenim  
}
```

5.1.4. Metoda getToken

Metoda vygeneruje user token objekt. Prihlasovací údaje se zjistí z Autorizačních parametrů ze SOAP hlavičky. Používá se místo metody login po integraci DMS s AAA portálem – od verze API 1.4 (cca od 15.12.2006).



```
private UserToken getToken(String serverID) throws Exception {  
    return loginProxy.getToken(serverID);  
}
```

5.2. DMS Port

Port pro práci s objekty typu Dokument.

5.2.1. Metoda search

Vyhledá uuid dokumentů odpovídající zadaným kritériím. Objekt DokumentHledani specifikuje vyhledávací kritéria, mezi jednotlivými kritérii platí vztah AND.
Atributy číslo jednací, číslo spisu, název, spisový znak, jméno odesílatele, jméno osoby, příjmení osoby, název organizace mohou být specifikovány pouze začátkem, ostatní musí být vypsány přesně.
Parametr maximální počet dokumentů nemusí být vyplněn – v tom případě metoda vrátí maximálně 500 dokumentů. Pokud tento parametr == 0, potom se vrátí všechny dokumenty které odpovídají zadaným kritériím – tato varianta však pro svou náročnost není doporučována.
Na dotažení detailu dokumentu se používá metod retrieve.

```
private void hledatDokument(UserToken token) throws Exception {  
    DokumentHledani dokument = new DokumentHledani();//objekt pro specifikovani hledani  
    dokument.setEvidencniCislo("1234567891");  
  
    DokumentSort docSort = new DokumentSort();  
    docSort.setSortType(CtSortType.ASC);  
    docSort.setSortAttribute(CtDocumentSortAttribute.agenda);  
  
    HledaniDokumentuRequest request = new HledaniDokumentuRequest(token, dokument,  
        docSort, new Integer(100));//vrati 100 uuid dokumentu  
  
    String[] dokumenty = dmsProxy.search(request).getUuid();  
}
```

5.2.2. Metoda retrieve

Metoda vrátí seznam detailů dokumentů specifikovaných pomocí uuid dokumentu. Metoda umožňuje nastavení, jestli se mají spolu s profily dokumentů dotáhnout i profily těl. Metoda vrátí pouze url adresu těla dokumentu – body je používáno pouze jako vstupní atribut při vytváření těla.

```
private void dotahniDetaily(UserToken token, String uuids[]) throws Exception {  
    CtRetrieveSpec[] specs = new CtRetrieveSpec[uuids.length]; // vytvoreni objektu requestu  
    for (int i = 0; i < specs.length; i++) {  
        specs[i] = new CtRetrieveSpec();  
        specs[i].setUUID(uuids[i]);  
        specs[i].setDotahnoutTelo(true);// priznak, jestli ma byt dotahnute i telo dokumentu  
    }  
    RetrieveRequest request = new RetrieveRequest(token, specs); // nastavi se UserToken ziskany pri prihlaseni  
    CtRetrievedDokument[] response = dmsProxy.retrieve(request).getDokument();  
    // vykona operaci retrieve  
  
    // response.getTelo() - telo dokumentu, ak bolo vyžadovane  
    // response.getDokument() - samotny objekt Dokument s naplnenymi detailami  
    for (int i = 0; i < response.length; i++) {  
        System.out.println("Retrieved doc. ec: "  
            + response[i].getProfil().getEvidencniCislo() + " typ:"
```



```
+ response[i].getProfil().getTypDokumentu() + ", url:"  
+ response[i].getTelo(0).getUrl();
```

```
}  
}
```

Response nově (verze API 3.0.0.) vedle atributu *telo* obsahuje i atribut *tela*. Tento atribut obsahuje všechna případná obecná těla daného dokumentu. (.NET)

```
Např: textBoxResultsDMS.Text += " pocet obecných tel: " + RDOC[0].tela.Length.ToString();  
foreach (DMSPort.TeloDokumentuObecne tdo in RDOC[0].tela)  
{  
    textBoxResultsDMS.Text += " typ a uuid ob tela : " + tdo.typ + " uuid: " + tdo.uuid;  
}
```

V případě, že externí aplikace přímo požaduje stažení těla pomocí vrácené URL je zapotřebí provést na straně volajícího serveru nezbytné nastavení prostředí OS tak, aby část vráceného URL byla korektně přeložena a URL přeposláno na konkrétní aplikační server DMS.

Pro jednotlivá prostředí tedy musí být v souboru „**hosts**“ (pro OS Win např.: C:\WINDOWS\system32\drivers\etc) nastaven překlad:

Integrační prostředí ČSSZ:
10.204.203.180 i-dms.cssz.cz

Testovací prostředí ČSSZ:
10.202.203.64 t-dms.cssz.cz

Produkční prostředí ČSSZ:
10.200.203.25 dms.cssz.cz

5.2.3. Metoda disable

Metoda zneplatní dokument specifikovaný svým uuid. Nastaví dokumentu atribut isNeplatny = true.

```
public void zneplatneni(UserToken token, String uuid) throws Exception {  
    UuidRequest uuidRequest = new UuidRequest();  
    uuidRequest.setToken(token);  
    uuidRequest.setUuid(uuid);  
    dmsProxy.disable(uuidRequest);  
}
```

5.2.4. Metoda enable

Metoda zruší zneplatnění dokumentu specifikovaného svým uuid. Nastaví dokumentu atribut isNeplatny = false.

```
public void obnoveni(UserToken token, String uuid) throws Exception {  
    UuidRequest uuidRequest = new UuidRequest();  
    uuidRequest.setToken(token);  
    uuidRequest.setUuid(uuid);  
    dmsProxy.enable(uuidRequest);  
}
```

5.2.5. Metoda history

Metoda vrací historii daného dokumentu ve formě XML.
Formát XML:



<list>

<history date="yyyy-MM-dd HH:mm:ss.SSSS" userName="" kod="" detail=""/>

</list>

Date obsahuje datum a čas změny, username obsahuje jméno uživatele, který provedl změnu (detailní jméno, nejedná se o login)

Kód může nabývat těchto hodnot:

Kód	Popis
history.message.createProfil	Založení profilu
history.message.updateProfil	Změna profilu
history.message.deleteProfil	Smazání profilu
history.message.toRecycleBinProfil	Vyhození profilu do koše
history.message.restorefromRecycleBinProfil	Obnovení profilu z koše
history.message.addContent	Přidání těla
history.message.updateContent	Změna těla
history.message.deleteContent	Odstranění těla
history.message.createRecord	Založení záznamu
history.message.createLink	Založení související
history.message.deleteLin	Odstranění související
history.message.toFolder	Přidání do složky
history.message.removeFromFolder	Odstranění ze složky
history.message.enableProfil	Obnovení profilu
history.message.disableProfil	Zneplatnění profilu
history.message.processStart	Spuštění procesu
history.message.processStop	Zastavení procesu

V detailu jsou atributy, které se změnily, ve formátu
navezSkupiny.navezAtributu=novaHodnota;...

```
public String historie(UserToken token, String uuid) throws Exception {  
    UuidRequest uuidRequest = new UuidRequest();  
    uuidRequest.setToken(token);  
    uuidRequest.setUuid(uuid);  
    return (String)dmsProxy.history(uuidRequest);  
}
```

5.2.6. Metoda create

Metoda zakládá dokument v CM. K dokumentu je možno rovnou přidat i tělo (základní i obecné) a metoda také specifikuje, jestli se má dokument spojit s kmenovými evidencemi nebo jestli se má předpokládat, že pokus o spojení s KE už byl proveden.

K jenomu dokumentu je možno ukládat 1 tělo základní a doporučeno ukládat maximálně 15 těl obecných.

Dokument musí mít vyplněné atributy agenda, typ, druh.

Pokud není specifikován atribut evidenční číslo, je vygenerováno.

```
private String createDokument(UserToken token) throws Exception {  
    Dokument dokument = new Dokument();// vytvoří se nový objekt Dokument  
    dokument.setPodani(new PodaniAtributy());  
    dokument.setDruhDokumentu(CtDruhDokumentu.PodaniPapirove);  
  
    // základní vlastnosti dokumentu  
    dokument.setTypDokumentu("101");  
    dokument.setAgenda("NEM");
```



```
dokument.setNazev("Testovací podání XXX");
dokument.setPoznamka("toto je test");
dokument.setNeplatny(false);

// nastavení známých informací o klientovi
dokument.setIco("222");
dokument.setOssz("227");
dokument.setRodneCislo("9999999999");
dokument.setVariabilniSymbol("222");

// nastavi se detaily o podání
dokument.getPodani().setPodani(new CtVstup());
// napr. aktuální datum
dokument.getPodani().getPodani().setDatum(Calendar.getInstance().getTime());
// a způsob podání
dokument.getPodani().getPodani().setZpusob(new Short((short) 1)); // z číselníka - 1 = Papír

// nastavení detailu o doručení
dokument.getPodani().setDoruceni(new CtVstup());
// napr. aktuální datum
dokument.getPodani().getDoruceni().setDatum(Calendar.getInstance().getTime());
// a způsob doručení
dokument.getPodani().getDoruceni().setZpusob(new Short((short) 3)); // z číselníku - 3 = Poštou

// podání se rozšíří o informace specifické pro papírové podání
dokument.getPodani().setPapirove(new PapirovePodani());
dokument.getPodani().getPapirove().setAdresaOdesilatele(
    "Jahodková chalupka 12, Rozpravkovo");
dokument.getPodani().getPapirove().setJmenoOdesilatele("Baba Jaga");
dokument.getPodani().getPapirove().setPocetListu(new Short((short) 1));
dokument.getPodani().getPapirove().setPocetPriloh(new Short((short) 1));
dokument.getPodani().getPapirove().setPscOdesilatele("14554");
dokument.getPodani().getPapirove().setMestoOdesilatele("htrhw");

// objekt pro tělo dokumentu, pokud se nejake přikládá
TeloDokumentu telo = new TeloDokumentu();
telo.setMime("image/tiff"); // MIME type - musí být korektně nastavený
telo.setPoznamka("test import"); // volitelná poznámka pro tělo
telo.setUrl("http://image.server.cc/obrazek.tif"); // URL z kterého má být tělo dotahnuto
// telo.setBody(new byte[] {1,2,3,4,5}); - použití když tělo existuje jako bytearray;
// může být nastaveno pouze jedno - URL nebo Body; nemohou být nastaveny současně

// příklad vložení obecného těla (.NET)
DMSPort.TeloDokumentuObecne telo8 = new DMSPort.TeloDokumentuObecne();
telo8.body = readByteArrayFromFile(@"D:\dev\csv.csv");
telo8.datumSpecified = true;
telo8.datum = DateTime.Now;
telo8.nazev = "PROTOKOL.doc";
telo8.poznamka = "A toto je moje poznámka!";
telo8.typSpecified = true;
telo8.typ = 6; // povinný atribut viz. číselník typů těl
telo8.sha2 = "SHA SHA ";
```



```
// vytvoří se request objekt
CreateRequest request = new CreateRequest();
// obsahuje objekt Dokument
request.setDokument(dokument);
request.setToken(token); // použije se token vygenerovaný při přihlášení
request.setHledatVKE(Boolean.TRUE);
request.setTelo(telo);

// případně ještě navíc pro tělo obecné (.NET)

dokumentCR.telaObecne = new DMSPort.TeloDokumentuObecne[1];
dokumentCR.telaObecne[0] = telo8;

// zavola se metoda na vytvoreni dokumentu
dokument = dmsProxy.create(request);
// system vrati dokument s vyplnenymi atributmi
System.out.println(dokument.getUuid());
return dokument.getUuid();
}
```

5.2.7. Metoda update

Metoda provádí změnu profilu dokumentu v CM. Metoda umožňuje specifikovat, jestli se má dokument spojit s kmenovými evidencemi nebo jestli se má předpokládat, že pokus o spojení s KE už byl proveden. Musí být použit objekt Dokument, který byl předtím vytažen z CM pomocí metody retrieve. Atribut datum změny se používá pro zjišťování, jestli dokument nebyl mezi voláním metody retrieve a update v databázi změněn.

```
private void upravit(Dokument dokument) throws Exception {
    dokument.setNazev("Zmeneny nazev");
    UpdateRequest uReq = new UpdateRequest(token, dokument, Boolean.TRUE);
    Dokument novyDokument = dmsProxy.update(uReq);
}
```

5.2.8. Metoda getOsszGroupCode

Metoda vyhledá kód skupiny okresních správ, do které náleží vstupní okresní správa. Metoda vychází z předpokladu, že každá okresní správa náleží právě do jedné skupiny OSSZ.

```
protected String getOsszGroupCode(String ossz) throws Exception {
    GetOsszGroupCodeRequest groupCodeRequest =
        new GetOsszGroupCodeRequest(token, ossz);
    GetOsszGroupCodeResponse resp = dmsProxy.getOsszGroupCode(groupCodeRequest); return
    resp.getOsszGroupCode();
}
```



5.2.9. Metoda `getOsszCodesByOsszGroupCode`

Metoda vyhledá všechny okresní správy, které náleží do dané skupiny okresních správ. Návrátovou hodnotou je pole Stringů, metoda vychází z předpokladu, že každá okresní správa náleží právě do jedné skupiny OSSZ.

```
protected String[] getOsszCodes(String groupCode) throws Exception {  
    GetOsszCodesByOsszGroupCodeRequest osszCodesRequest  
        = new GetOsszCodesByOsszGroupCodeRequest(token, groupCode);  
    GetOsszCodesByOsszGroupCodeResponse resp =  
        dmsProxy.getOsszCodesByOsszGroupCode(osszCodesRequest);  
  
    return resp.getOsszMap();  
}
```

5.3. KNS Port

SpusteniKNS

Port pro spuštění procesů pro opravu nestandardních stavů (KNS – kompenzace nestandardních stavů).

// tridy vygenerovane WSDL2Java

SpusteniKNSProxy **spusteniKNS** = new SpusteniKNSProxy();

V objektu `SpecifikaceDokumentu` stačí vyplnit pouze jeden z parametrů `uuid`, `evidenční číslo`. Pokud jsou vyplněny oba, aplikace hledá pouze parametr `uuid` a atribut `evidenční číslo` ignoruje. Pokud se nepodaří dokument vyhledat, vrátí aplikace chybu s kódem 1 (Dokument se zadaným UUID neexistuje) nebo s kódem 2 (Dokument se zadaným evidenčním číslem neexistuje).

Poznámka: Specifikace podle `uuid` je rychlejší.

```
SpecifikaceDokumentu specifikaceDokumentu = new SpecifikaceDokumentu();  
specifikaceDokumentu.setUuid(uuid);  
specifikaceDokumentu.setEvidencniCislo(evidencniCislo);
```

Pokud dojde k problémům během spuštění procesu, vrátí aplikace chybu s kódem 4 (Systemova chyba(popis chyby))

5.3.1. Metoda `zmenaKlasifikace`

Používá se v případě, že uživatel/aplikace zjistí, že dokument byl klasifikován špatnou agendou nebo typem dokumentu.

Metoda spustí interní DMS proces změna klasifikace nad zadaným dokumentem. Pokud je správně vyplněna hodnota agendy a typu dokumentu, nastaví DMS proces přímo tyto nové hodnoty do profilu dokumentu (bez interakce uživatele). V případě, že je zadána neplatná kombinace agendy a typu, nebo agenda a typ nejsou vyplněny, zobrazí se tento dokument ve worklistu uživateli zodpovědnému za KNS, který je požádán o volbu agendy a typu podle platného číselníku klasifikace.

```
private ReturnValue zmenaKlasifikace(SpecifikaceDokumentu specifikaceDokumentu, String agenda, String  
typDokumentu) {  
    ZmenaKlasifikace zmenaKlasifikaceRequest = new ZmenaKlasifikace();  
    zmenaKlasifikaceRequest.setSpecifikaceDokumentu(specifikaceDokumentu);  
    zmenaKlasifikaceRequest.setAgenda(agenda);
```




```
zmenaKlasifikaceRequest.setTypDokumentu(typDokumentu);
```

```
ReturnValue result = spusteniKNS.zmenaKlasifikace(zmenaKlasifikaceRequest);  
return result;  
}
```

5.3.2. Metoda resken

Používá se, pokud uživatel/aplikace zjistil, že naskenovaný image je nečitelný. Metoda spustí interní DMS proces, který vyžádá resken zadaného dokumentu. Vlastní kompenzaci provede uživatel zodpovědný za kompenzaci na příslušné OSSZ.

```
private ReturnValue resken(SpecifikaceDokumentu specifikaceDokumentu) {  
    ReturnValue result = spusteniKNS.resken(specifikaceDokumentu);  
    return result;  
}
```

5.3.3. Metoda zmenaEvidencnihoCisla

Používá se, pokud uživatel/aplikace zjistí nesoulad mezi evidenčním číslem v profilu dokumentu a evid.číslem na štítku s čárovým kódem nalepeném na papíru (neskenovaném na image). Metoda spustí interní DMS proces kontroly evidenčního čísla zadaného dokumentu. Vlastní kompenzaci provede uživatel zodpovědný za kompenzaci na příslušném OSSZ.

```
private ReturnValue zmenaEvidencnihoCisla(SpecifikaceDokumentu specifikaceDokumentu) {  
    ReturnValue result = spusteniKNS.zmenaEvidencnihoCisla(specifikaceDokumentu);  
    return result;  
}
```

5.3.4. Metoda zmenaZDV

Používá se, pokud uživatel/aplikace zjistí nesoulad uložené datové věty v ZDV. Metoda spustí interní DMS proces změny datové věty ze ZDV. Vlastní kompenzaci provede uživatel zodpovědný za kompenzaci na příslušném OSSZ.

```
private ReturnValue zmenaZDV(SpecifikaceDokumentu specifikaceDokumentu) {  
    ReturnValue result = spusteniKNS.zmenaZDV(specifikaceDokumentu);  
    return result;  
}
```

5.3.5. Metoda zmenaPrav

Používá se, pokud uživatel/aplikace vyžaduje změnit práva přístupu. Metoda spustí interní DMS proces, který vyžádá změnu práv dokumentu. Vlastní kompenzaci provede uživatel zodpovědný za kompenzaci na příslušném OSSZ.

```
private ReturnValue zmenaPrav(SpecifikaceDokumentu specifikaceDokumentu, String prava) {  
    ZmenaPrav zmenaPrav = new ZmenaPrav();  
    zmenaPrav.setSpecifikaceDokumentu(specifikaceDokumentu);  
    zmenaPrav.setPrava(prava);  
  
    ReturnValue result = spusteniKNS.zmenaPrav(zmenaPrav);  
    return result;  
}
```




}

5.3.6. Metoda nestandardniStav

Metoda spustí interní DMS proces pro řešení jiného nestandardního stavu dokumentu – předem nespecifikovaného. Umožňuje předat poznámku ke kompenzaci. Vlastní kompenzaci provede uživatel zodpovědný za kompenzaci na příslušném OSSZ podle svého uvážení, problému a stavu dokumentu: zneplatnění, editace profilu, resken, bez činnosti.

```
private ReturnValue nestandardniStav(SpecifikaceDokumentu specifikaceDokumentu, String poznamka) {  
    NestandardniStavRequest nestandardniStav = new NestandardniStavRequest();  
    nestandardniStav.setSpecifikaceDokumentu(specifikaceDokumentu);  
    nestandardniStav.setPoznamka(poznamka);  
  
    ReturnValue result = spusteniKNS.nestandardniStav(nestandardniStav);  
    return result;  
}
```

5.4. Použití rozhraní generátoru evidenčních čísel

Implementace generátoru evidenčních čísel poskytuje navenek dvě služby, které mohou ostatní oblasti využívat pro získání dalšího čísla ze specifikované řady evidenčních čísel, resp. pro zarezervování několika dalších čísel ze specifikované řady.

SequenceGenerator

Port pro generování číselné řady.

// *tridy vygenerovane WSDL2Java*

SequenceGeneratorProxy **sequenceGenerator** = new SequenceGeneratorProxy();

Metoda getNextById

Metoda vygeneruje nové číslo číselné řady.

```
private String getNextById(int idCislnRady) {  
    return sequenceGenerator.getNextById(idCislnRady);  
}
```

Metoda getNumbersById

Metoda vygeneruje zadaný počet nových čísel číselné řady. Vrátí první a poslední číslo.

```
private void getNumbersById(int idCislnRady, int pocet) {  
    SequenceIdAndCount request = new SequenceIdAndCount();  
    request.setSequenceId(idCislnRady);  
    request.setCount(pocet);  
  
    GeneratedNumbers generated = sequenceGenerator.getNumbersById(request);  
    System.out.println("prvni cislo=" + generated.getFirstNumber() +  
        " posledni cislo=" + generated.getLastNumber());  
}
```



5.5. GenericDMSPort

5.5.1. Metoda history

Metoda vrací historii daného dokumentu ve formě XML.

Formát XML:

<list>

<history date="yyyy-MM-dd HH:mm:ss.SSSS" userName="" kod="" detail=""/>

</list>

Date obsahuje datum a čas změny, username obsahuje jméno uživatele které provedl změnu (detailní jméno, nejedná se o login)

Kód může nabývat těchto hodnot:

Kód	Popis
history.message.createProfil	Založení profilu
history.message.updateProfil	Změna profilu
history.message.deleteProfil	Smazání profilu
history.message.toRecycleBinProfil	Vyhození profilu do koše
history.message.restorefromRecycleBinProfil	Obnovení profilu z koše
history.message.addContent	Přidání těla
history.message.updateContent	Změna těla
history.message.deleteContent	Odstranění těla
history.message.createRecord	Založení záznamu
history.message.createLink	Založení související
history.message.deleteLin	Odstranění související
history.message.toFolder	Přidání do složky
history.message.removeFromFolder	Odstranění ze složky
history.message.enableProfil	Obnovení profilu
history.message.disableProfil	Zneplatnění profilu
history.message.processStart	Spuštění procesu
history.message.processStop	Zastavení procesu

V detailu jsou atributy, které se změnily, ve formátu
nazevSkupiny.nazevAtributu=novaHodnota;...

```
public String historie(UserToken token, String uuid) throws Exception {  
    UuidRequest uuidRequest = new UuidRequest();  
    uuidRequest.setToken(token);  
    uuidRequest.setUuid(uuid);  
    return (HistoryResponse)genDmsProxy.history(uuidRequest);  
}
```

5.5.2. Metoda disable

Metoda zneplatní dokument specifikovaný svým uuid. Nastaví dokumentu atribut isNeplatny = true.

```
public void zneplatneni(UserToken token, String uuid) throws Exception {  
    UuidRequest uuidRequest = new UuidRequest();  
    uuidRequest.setToken(token);  
    uuidRequest.setUuid(uuid);  
    genDmsProxy.disable(uuidRequest);  
}
```



5.5.3. Metoda enable

Metoda zruší zneplatnění dokumentu specifikovaného svým uuid. Nastaví dokumentu atribut isNeplatny = false.

```
public void obnoveni(UserToken token, String uuid) throws Exception {  
    UuidRequest uuidRequest = new UuidRequest();  
    uuidRequest.setToken(token);  
    uuidRequest.setUuid(uuid);  
    genDmsProxy.enable(uuidRequest);  
}
```

5.5.4. Metoda delete

Metoda odstraní dokument specifikovaného svým uuid.

```
UUIDRequest request = new UUIDRequest(token, uuid);  
GenericResponse response = genDmsProxy.delete(request);  
  
return response.getStatus();
```

5.5.5. Metoda getRelated

Metoda vrátí seznam uuid záznamů, které jsou se vstupním záznamem slinkovány určitým typem linku.

```
GetLinkedRequest request = new GetLinkedRequest(  
    token, uuid, "CZ_SOUVISEJICI");  
UUIDsResponse response = genDmsProxy.getLinked(request);
```

5.5.6. Metoda getItemTypes

Metoda vrátí seznam itemtypů, které jsou viditelné pro uživatele identifikovaného tokenem.

```
GenericTokenRequest request = new GenericTokenRequest(token);  
ItemTypesResponse response = genDmsProxy.getItemTypes(request);  
for (int i = 0; i < response.getItemTypeName().length; i++) {  
    System.out.println(response.getItemTypeName(i));  
}
```

5.5.7. Metoda getSearchAttributes

Metoda vrátí seznam atributů a jejich jmen, které obsahuje item type, který je součástí vstupního parametru.

```
SearchAttributesRequest request = new SearchAttributesRequest(  
    token, itemtype);  
SearchAttributeResponse response = genDmsProxy  
    .getSearchAttributes(request);  
for (int i = 0; i < response.getAttributes().length; i++) {  
    System.out.println(  
        response.getAttributes(i).getName() + " "  
        + response.getAttributes(i).getType());  
}
```



5.5.8. Metoda getItem

Metoda vrátí záznam z CMOD, který je specifikovaný svým uuid. Pokud uživatel zadá, že chce vrátit i těla k záznamu, musí zadat typ vazby, který je mezi tělem a záznamem. Např. CZ_SKEN, ICMBASE a pod.

```
getItemRequest request = new getItemRequest(token, uuid, false, null);
ItemResponse response = genDmsProxy.getItem(request);
System.out.println(response.getCmObject().getItem().getAcl());
System.out.println(response.getCmObject().getItem().getItemType());
System.out.println(response.getCmObject().getItem().getUuid());

System.out.println(response.getCmObject()
    .getItem().getAttribute().length);
System.out.println(response.getCmObject()
    .getItem().getPrava().isProhlizetSoubor());
return response.getCmObject();
```

5.5.9. Metoda.getItems

Metoda vrátí seznam záznamů z CMOD, který je specifikovaný atributy a itemTypem. Pokud uživatel zadá, že chce vrátit i těla k záznamu, musí zadat typ vazby, který je mezi tělem a záznamem. Např. CZ_SKEN, ICMBASE a pod.

```
cz.cssz.dms.schemas.genericDMS._1.SearchRequest request = new
cz.cssz.dms.schemas.genericDMS._1.SearchRequest(token, "CZD_ZDD ", attributes, null, new Integer("5"),
bodies, "ICMBASE");
ItemsResponse response = genDmsProxy.getItems(request);
if (response.getCmObjects() == null)
    return null;
System.out.println(response.getCmObjects().length);
CmObject cmObject = null;
for (int i = 0; i < response.getCmObjects().length; i++) {
    cmObject = response.getCmObjects(i);
    System.out.println(cmObject.getItem().getItemType());
    System.out.println(cmObject.getItem().getUuid());
    System.out.println(cmObject.getItem().getAttribute().length);
    System.out.println(cmObject.getItem().getPrava().isProhlizetSoubor());
    if (bodies) {
        System.out.println(cmObject.getBodies(0).getBodyUrl());
        System.out.println(cmObject.getBodies(0).getMimeType());
    }
    CMAAttribute[] attrs = cmObject.getItem().getAttribute();
    System.out.println("Atributu " + attrs.length);
    for (int e = 0; e < attrs.length; e++) {
        if (attrs[e] == null)
            continue;
        System.out.println(attrs[e].getName());
        //System.out.println(attrs[e].getType());
        System.out.println(attrs[e].getValue());
    }
}
return response.getCmObjects();
```



5.5.10. Metoda `getItemsInFolder`

Metoda vrátí seznam uuid záznamů z CMOD, který je ve složce specifikované uuid.

```
cz.cssz.dms.schemas.genericDMS._1.UUIDRequest request = new
    cz.cssz.dms.schemas.genericDMS._1.UUIDRequest(token, uuid);
GetItemsInFolderResponse response = genDmsProxy.getItemsInFolder(request);
if (response.getCmObjects() == null)
    return null;
System.out.println(response.getCmObjects().length);
CMObject cmObject = null;
for (int i = 0; i < response.getCmObjects().length; i++) {
    cmObject = response.getCmObjects(i);
    System.out.println(cmObject.getItem().getItemType());
    System.out.println(cmObject.getItem().getUuid());
    System.out.println(cmObject.getItem().getAttribute().length);
    if (bodies) {
        System.out.println(cmObject.getBodies(0).getBodyUrl());
        System.out.println(cmObject.getBodies(0).getMimeType());
    }
    CMAAttribute[] attrs = cmObject.getItem().getAttribute();
    System.out.println("Atributu " + attrs.length);
    for (int e = 0; e < attrs.length; e++) {
        if (attrs[e] == null)
            continue;
        System.out.println(attrs[e].getName());
        System.out.println(attrs[e].getValue());
    }
}
return response.getCmObjects();
```

5.5.11. Metoda `getItemsInFolder`

Metoda vrátí záznam, kterému odpovídají metadatům folderu, ve kterém se nachází záznam specifikovaný uuid.

```
cz.cssz.dms.schemas.genericDMS._1.UUIDRequest request = new
    cz.cssz.dms.schemas.genericDMS._1.UUIDRequest(token, uuid);
GetFolderForItemResponse response = genDmsProxy.getFolderForItem(request);
if (response.getCmItem() == null)
    return null;
System.out.println(response.getCmItem().getItemType());
System.out.println(response.getCmItem().getUuid());
System.out.println(response.getCmItem().getAttribute().length);
System.out.println(response.getCmItem()
    .getPrava().isProhlizetSoubor());
CMAAttribute[] attrs = response.getCmItem().getAttribute();
System.out.println("Atributu " + attrs.length);
for (int e = 0; e < attrs.length; e++) {
    if (attrs[e] == null)
        continue;
    System.out.println(attrs[e].getName());
    //System.out.println(attrs[e].getType());
    System.out.println(attrs[e].getValue());
}
```



5.5.12. Metoda create

Metoda vytvoří záznam v CM ve vstupním itemType a naplní jej atributy.

```
CMAttribute[] attributes = genAttributesTest(itemType);
for (int i = 0; i < attributes.length; i++) {
    if (attributes[i].getName()
        .indexOf("detailyKlienta.czk_rodneCislo") != -1) {
        attributes[i].setValue("16611223344");
    }
    if (attributes[i].getName()
        .indexOf("spolecneAtributy.czg_ed") != -1) {
        attributes[i].setValue("200933221111");
    }
    if (attributes[i].getName().indexOf("czx_dUkZprac") != -1) {
        attributes[i].setValue("2008-12-02");
        attributes[i].setSearchOperator(CtSearchOperatorAttribute.OD);
    }
}

CMItem cmItem = new CMItem(null, null, null, attributes, "CZD_ZDD");

cz.cssz.dms.schemas.genericDMS._1.CreateRequest request = new
    cz.cssz.dms.schemas.genericDMS._1.CreateRequest(token, cmItem);
CreateResponse response = genDmsProxy.create(request);
if (response.getCmObject() == null)
    return null;
CMObject cmObject = response.getCmObject();
System.out.println(cmObject.getItem().getItemType());
System.out.println(cmObject.getItem().getUuid());
System.out.println(cmObject.getItem().getAttribute().length);
    System.out.println(cmObject.getItem().getPrava().isProhlizetSoubor());
CMAttribute[] attrs = cmObject.getItem().getAttribute();
for (int i = 0; i < attrs.length; i++) {
    System.out.println(attrs[i].getName() + " "
        + attrs[i].getType() + " " + attrs[i].getValue());
}
return response.getCmObject();
}
```

5.6. DIS Port

Port pro práci s objekty typu PodaniDIS.

5.6.1. Metoda search

Vrátí seznam uuid dokumentů odpovídající zadaným kritériím. Objekt PodaniDISHledani specifikuje vyhledávací kritéria, mezi jednotlivými kritérii platí vztah AND.

Parametr maximální počet dokumentů nemusí být vyplněn – v tom případě metoda vrátí maximálně 500 dokumentů. Pokud tento parametr = 0, potom se vrátí všechny dokumenty, které odpovídají zadaným kritériím – tato varianta však pro svou náročnost není doporučována.

Na dotažení detailu dokumentu se používá metoda retrieve.

```
private void hledatDIS(UserToken token) throws Exception {
```



```
PodaniDISHledani hledaniDis = new PodaniDISHledani (); //objekt pro specifikovani hledani  
hledaniDis.setEvidencniCislo("1234567891");
```

```
SearchDISRequest request = new SearchDISRequest (token, hledaniDis,  
null, new Integer(100)); //vrati 100 uuid dokumentu
```

```
SearchDISResponse podani = disProxy.search(request).getUuid();  
String uuids[] = searchDISResponse.getUuid();  
}
```

5.6.2. Metoda retrieve

Metoda vrátí seznam detailů dokumentů specifikovaných pomocí uuid dokumentu. Metoda umožňuje nastavení, jestli se mají spolu s profily dokumentů dotáhnout, profily těl nebo seznam všech rodných čísel náležejících k záznamu (vícenásobný atribut záznamu). Metoda vrací pouze url adresu těla dokumentu – body je používáno pouze jako vstupní atribut při vytváření těla.

```
private void dotahnoutDIS(String[] uuids) throws Exception {  
    CtRetrieveDISSpec[] specs = new CtRetrieveDISSpec[uuids.length];  
    for (int i = 0; i < specs.length; i++) {  
        specs[i] = new CtRetrieveDISSpec(uuids[i], true, true);  
    }  
    RetrieveDISRequest request = new RetrieveDISRequest(token, specs);  
    RetrieveDISResponse retrieveDISResponse = disProxy.retrieveDIS(request);  
    if (retrieveDISResponse != null) {  
        CtRetrievePodaniDIS[] podaniDIS = retrieveDISResponse.getPodaniDIS();  
        for (int i = 0; i < podaniDIS.length; i++) {  
            System.out.println("Retrieved dis. ec: "  
                + podaniDIS[i].getProfil().getEvidencniCislo() + "  
                + podaniDIS[i].getProfil().getTypDokumentu() + "  
                + podaniDIS[i].getProfil().getZmena().getDatum().getTimeInMillis());  
            System.out.println("ACL " + podaniDIS[i].getProfil().getAcl());  
  
            if (podaniDIS[i].getProfil().getFormularData() instanceof  
PodaniFormularData[]) {  
                for (int j = 0; j < podaniDIS[i].getProfil().getFormularData().length;  
j++) {  
                    System.out.println(j + " " +  
podaniDIS[i].getProfil().getFormularData(j).getRodneCislo());  
                }  
            }  
        }  
    }  
}
```

Těla DIS jsou od API verze 3.0.0 ukládána i dotahována jako *obecná těla* (např. v .NET):

```
foreach (DISPort.TeloDokumentuObecne tdo in retpoddis[0].tela)  
{  
    textBoxDIS.Text += tdo.typ + " " +  
    tdo.poznamka + " " + tdo.nazev + " " +  
    tdo.datum + " " + tdo.sha2 + " " + tdo.url + " " + tdo.uuid + " " +  
    tdo.body.ToString()+Environment.NewLine;  
}
```




5.6.3. Metoda disable

Metoda zneplatní dokument specifikovaný svým uuid. Nastaví dokumentu atribut isNeplatny = true.

```
public void zneplatneni(UserToken token, String uuid) throws Exception {  
    UuidDISRequest uuidRequest = new UuidDISRequest();  
    uuidRequest.setToken(token);  
    uuidRequest.setUuid(uuid);  
    disProxy.disableDIS(uuidRequest);  
}
```

5.6.4. Metoda enable

Metoda zruší zneplatnění dokumentu specifikovaného svým uuid. Nastaví dokumentu atribut isNeplatny = false.

```
public void obnoveni(UserToken token, String uuid) throws Exception {  
    UuidDISRequest uuidRequest = new UuidDISRequest();  
    uuidRequest.setToken(token);  
    uuidRequest.setUuid(uuid);  
    disProxy.enableDIS(uuidRequest);  
}
```

5.6.5. Metoda history

Metoda vrací historii daného dokumentu ve formě XML.

Formát XML:

```
<list>  
    <history date="yyyy-MM-dd HH:mm:ss.SSSS" userName="" kod="" detail=""/>  
</list>
```

Date obsahuje datum a čas změny, username obsahuje jméno uživatele, který provedl změnu (detailní jméno, nejedná se o login)

Kód může nabývat těchto hodnot:

Kód	Popis
history.message.createProfil	Založení profilu
history.message.updateProfil	Změna profilu
history.message.deleteProfil	Smazání profilu
history.message.toRecycleBinProfil	Vyhození profilu do koše
history.message.restorefromRecycleBinProfil	Obnovení profilu z koše
history.message.addContent	Přidání těla
history.message.updateContent	Změna těla
history.message.deleteContent	Odstranění těla
history.message.createRecord	Založení záznamu
history.message.createLink	Založení související
history.message.deleteLin	Odstranění související
history.message.toFolder	Přidání do složky
history.message.removeFromFolder	Odstranění ze složky
history.message.enableProfil	Obnovení profilu
history.message.disableProfil	Zneplatnění profilu



history.message.processStart	Spuštění procesu
history.message.processStop	Zastavení procesu

V detailu jsou atributy, které se změnily, ve formátu
navezSkupiny.navezAtributu=novaHodnota;...

```
public String historie(UserToken token, String uuid) throws Exception {  
    UuidRequest uuidRequest = new UuidRequest();  
    uuidRequest.setToken(token);  
    uuidRequest.setUuid(uuid);  
    return (String)disProxy.historyDIS(uuidRequest);  
}
```

5.6.6. Metoda create

Metoda zakládá dokument v CM. K dokumentu je možno rovnou přidat i tělo a metoda také specifikuje, jestli se má dokument spojit s kmenovými evidencemi nebo jestli se má předpokládat, že pokus o spojení s KE už byl proveden.

Dokument musí mít vyplněné atributy agenda, typ, druh.

Pokud není specifikován atribut evidenční číslo, je toto vygenerováno.

```
private String createPodaniDIS(String url1, String url2) throws Exception {  
    PodaniDIS podaniDIS = new PodaniDIS();  
    //Nastaveni atributu  
    podaniDIS.setDruhDokumentu(CtDruhDokumentu.PodaniElektronicke);  
    podaniDIS.setNazev("Testovací zadost");  
    podaniDIS.setPoznamka("toto je test");  
    podaniDIS.setNeplatny(false);  
    podaniDIS.setCisloJednaci("3214321");  
    podaniDIS.setCisloSpisu("66");  
    podaniDIS.setMistoUlozeni("TADY");  
    podaniDIS.setNarokovyPodklad(new Boolean(false));  
    podaniDIS.setPodtypDokumentu("7787382432");  
    podaniDIS.setVariabilniSymbol("444444");  
    podaniDIS.setTypDokumentu("01");  
    podaniDIS.setAgenda("REG");  
    RMKlasifikace klasifikace = new RMKlasifikace();  
    klasifikace.setSkartacniLhuta(new Short((short) 20));  
    klasifikace.setSkartacniZnak("A");  
    klasifikace.setSpisovyZnak("101");  
    podaniDIS.setKlasifikace(klasifikace);  
    podaniDIS.setOssz("118");  
    podaniDIS.setPodani(new PodaniAtributy());  
    podaniDIS.getPodani().setIdZdv(new BigDecimal(454564));  
    podaniDIS.setPodaniDis(new PodaniDISAtributy());  
    podaniDIS.getPodaniDis().setDatumPotvrzeni(new Date(new  
Long("1248782000000").longValue()));  
    podaniDIS.getPodaniDis().setPotvrdil("potvrdil");  
    podaniDIS.getPodaniDis().setIdCSSZ("idCSSZ");  
    podaniDIS.getPodaniDis().setIdPVS("idPVS");  
    podaniDIS.getPodani().setPodani(new CtVstup());  
    podaniDIS.getPodani().getPodani().setDatum(new Date(new  
Long("1248783000000").longValue()));  
    podaniDIS.getPodani().getPodani().setZpusob(new Short((short) 6));  
    podaniDIS.getPodani().setDoruceni(new CtVstup());  
}
```



```
podaniDIS.getPodani().setElektronicke(new ElektronickePodani());
podaniDIS.getPodani().getElektronicke().setZpusobAutorizace("RUCNE");
podaniDIS.getPodani().getDoruceni().setZpusob(new Short((short) 8));
podaniDIS.getPodani().getDoruceni().setDatum(new Date(new
Long("1248784000000").longValue()));
podaniDIS.getPodani().setVyrizeni(new CtVyrizeni());
podaniDIS.getPodani().getVyrizeni().setZpusob("DOBRE");
podaniDIS.getPodani().getVyrizeni().setDatum(new Date(new
Long("1248794000000").longValue()));
podaniDIS.getPodani().setVstupniZpracovani(new CtZpracovani());
podaniDIS.getPodani().getVstupniZpracovani().setZpusob(new Short((short) 7));
podaniDIS.getPodani().getVstupniZpracovani().setStav(new Short((short) 1));
podaniDIS.getPodani().getVstupniZpracovani().setDatumUkonceni(new Date(new
Long("1248774000000").longValue()));
podaniDIS.getPodani().getVyrizeni().setDatum(new Date(new
Long("1248794000000").longValue()));

podaniDIS.getPodani().setDatumZTextu(new Date(new Long("1248784000000").longValue()));

//Pridani seznamu rodnych cisel zaznamu
podaniDIS.setFormularData(
    new PodaniFormularData[]{
        new PodaniFormularData("8811222010"),
        new PodaniFormularData("8811222011")
    }); //Seznam tel zaznamu DIS se nově (API 3.0.0) vyplňuje jako tělo
obecné(.NET)
DISPort.TeloDokumentuObecne telo1 = new DISPort.TeloDokumentuObecne();
telo1.body = readByteArrayFromFile(@"c:\51 - DIScripted.xml");
telo1.datumSpecified = true;
telo1.datum = DateTime.Now;
telo1.nazev = "PROTOKOL.doc";
telo1.poznamka = "A toto je moje poznámka!";
telo1.typSpecified = true;
telo1.typ = 51; // povinný atribut z platného číselníku
telo1.sha2 = "SHA SHA";

DISPort.TeloDokumentuObecne[] sadatel = new DISPort.TeloDokumentuObecne[1];
sadatel[0] = telo1;

CreateDISRequest request = new CreateDISRequest();
request.setPodaniDIS(podaniDIS);
request.setToken(token);
request.setTelo(sadatel);
//Vytvoreni
podaniDIS = disProxy.createDIS(request);
System.out.println(podaniDIS.getUuid());
return podaniDIS.getUuid();
}
```

5.6.7. Metoda update

Metoda provádí změnu profilu podání DIS v CM. Metoda umožňuje specifikovat, jestli se má dokument spojit s kmenovými evidencemi nebo jestli se má předpokládat, že pokus o spojení s KE už byl proveden.



Musí být použit objekt PodaniDIS, který byl předtím vytažen z CM pomocí metody retrieve. Atribut datum změny se používá pro zjišťování, jestli dokument nebyl mezi voláním metody retrieve a update v databázi změněn.

```
private void upravit(PodaniDIS podaniDIS) throws Exception {  
    podaniDIS.setNazev("Zmeneny nazev");  
    UpdateDISRequest uReq = new UpdateDISRequest (token, podaniDIS, Boolean.TRUE);  
    PodaniDIS novePodaniDIS = disProxy.update(uReq);  
}
```

5.7. ESB Port

Port pro práci s objekty typu PodaniESB. Použití metod je stejné jako v případě portu DMS, pouze byly rozšířeny atributy pro podání dokumentu a pro vyhledávání.

5.7.1. Metoda searchESB

Vyhledá uuid dokumentů odpovídající zadaným kritériím. Objekt PodaniESBHledani specifikuje vyhledávací kritéria, mezi jednotlivými kritérii platí vztah AND. Všechny informace platí jako v případě metody search na portu DMS (viz kapitola 5.2.1 Metoda search).

5.7.2. Metoda retrieveESB

Metoda vrátí seznam detailů dokumentů specifikovaných pomocí uuid dokumentu. Všechny informace platí jako v případě metody retrieve na portu DMS (viz kapitola 5.2.2 Metoda retrieve).

5.7.3. Metoda createESB

Metoda zakládá dokument v CM. K dokumentu je možno rovnou přidat i tělo (základní i obecné) a metoda také specifikuje, jestli se má dokument spojit s kmenovými evidencemi nebo jestli se má předpokládat, že pokus o spojení s KE už byl proveden. Objekt PodaniESB definuje atributy dokumentu.

Všechny informace platí jako v případě metody create na portu DMS (viz kapitola 5.2.6 Metoda create).

5.7.4. Metoda updateESB

Metoda provádí změnu profilu dokumentu v CM. Metoda umožňuje specifikovat, jestli se má dokument spojit s kmenovými evidencemi nebo jestli se má předpokládat, že pokus o spojení s KE už byl proveden. Objekt PodaniESB definuje atributy dokumentu.

Musí být použit objekt PodaniESB, který byl předtím vytažen z CM pomocí metody retrieve. Atribut datum změny se používá pro zjišťování, jestli dokument nebyl mezi voláním metody retrieve a update v databázi změněn.

Všechny informace platí jako v případě metody update na portu DMS (viz kapitola 5.2.7 Metoda update).



6. Požadavky na okolní systémy přistupující do DMS

6.1. Předpoklady implementace vzájemné komunikace

Během přípravy využívání DMS (včetně ATV) jako úložiště pro dokumenty z přistupující aplikace je nutné si uvědomit několik důležitých faktů, které pomohou oběma stranám formulovat představu o využívání úložiště DMS.

Prvním krokem je kontaktování garanta za oblast DMS na ČSSZ z důvodů objasnění základních otázek vzájemné komunikace propojovaných systémů. Zejména se jedná o následující témata :

- Který z archivů má být použit pro ukládání dokumentů
Data je možné ukládat do DMS nebo do ATV (archiv tiskových výstupů). DMS tým poskytne potřebnou podporu při rozhodnutí, který z archivů je vhodnější.
- Požadavky na konfiguraci číselníků
Ukládané dokumenty jsou věcně členěny na jednotlivé agendy a typy. Je třeba upřesnit, zda požadované agendy či typy dokumentů již v číselnících existují, nebo zda je nutné zavést je nově. Aktuální verze číselníků agend a typů dokumentů je uložena v úložišti číselníků a za aktuálnost tohoto číselníku zodpovídá garant za oblast DMS na ČSSZ.
- Problematika objemu ukládaných či poskytovaných dokumentů do/z DMS
Jaká množství dokumentů plánuje aplikace vytvářet/číst v rámci daného časového úseku ? Jaký datový objem bude přistupující aplikace vytvářet/číst ? Bude se jednat o kontinuální či dávkové operace ? DMS tým doporučí postup ukládání dokumentů v souvislosti s vytižením DMS a s povahou vytvářených dokumentů.
- Shoda na obecných attributech
Jsou existující atributy ve standardním datovém modelu DMS či ATV dostatečné ? Atributy je možné rozšířit, či některé z nich personifikovat pro použití dané aplikace.
- Zjištění režimu práce s DMS
Bude aplikace provádět zápis objektů do DMS, čtení dat z DMS nebo oboje ?
- Využití služby numberGeneratoru systému DMS
Každý dokument ukládaný do DMS musí mít jako povinný atribut vyplněno tzv. evidenční číslo (EČ). Bez tohoto unikátního identifikátoru nelze dokument uložit. Evidenční číslo pro nový dokument lze získat voláním specifické služby DMS – numberGenerator. Druhou možností při zakládání dokumentu v DMS je nevyplňovat pole tohoto atributu. V takovém případě dojde k automatickému vygenerování tohoto EČ systémem DMS. V rámci implementace je tedy nutné rozhodnout, jaký způsob je v daném případě metodicky vhodnější.
- Představa o využití uživatelských účtů
DMS je omezeno pomocí interně definovaných uživatelských účtů. Z povahy aplikace je nutné vysledovat, zda přístup do DMS bude probíhat pomocí pojmenovaných uživatelů (loginname referentů) nebo pomocí servisního účtu. Jednotlivé akce založení/čtení dokumentů by měli provádět pojmenovaní uživatelé nebo, v opodstatněných případech, servisní účet. V případě použití servisního účtu je nutné zvážit bezpečnostní dopady tohoto přístupu. Zřízení servisního účtu podléhá schválení garantem oblasti DMS a dále odborem 52 na ČSSZ.



Na základě shody nad výše uvedenými tématy lze vypracovat a schválit harmonogram implementace a nasazení dohodnuté komunikace mezi jednotlivými systémy a to ve všech prostředích ČSSZ (integračním, testovacím/školícím, produkčním).

6.2. Implementace

Všechny okolní aplikace využívají k přístupu do DMS metody DMS API. Přístup je možný v režimu čtení (do DMS se neukládá, pouze se zobrazují profily a např. těla dokumentů) nebo zápisu. V každém případě však přístup k dokumentům podléhá řízení podle vnitřních přístupových práv ACL. Obecně platí, že uživatelé mohou měnit metadata a zobrazovat těla (např. obrázky) pouze u dokumentů, které místně přísluší dané OSSZ. Referenti z jiných ÚOJ mají k dispozici pouze pasivní náhled na metadata dokumentů. Druhým pohledem na zpřístupnění dokumentů uživatelům je tzv. agendový přístup (definuje povolené a zakázané agendy na dvou úrovních - čtení a editace). Až na výjimky (specifické metody DMS) je nutné začít jakoukoliv práci s dokumenty přihlášením do DMS. Po ukončení operací s dokumenty je potřeba se z DMS odhlásit.

Posloupnost akcí by tedy pokaždé měla být následující :

- přihlášení k DMS (login) – metoda *getToken*,
- práce s dokumenty pomocí zvolených metod – např. založení dokumentu pomocí metody *create*, získání informací o dokumentu pomocí metody *retrieve*,
- odhlášení z DMS (logout) – metoda *logout*.

V případě hromadného (dávkového) zakládání dokumentů nebo jiné opakující se práce s dokumenty je velmi vhodné omezit doprovodné loginy a logouty do DMS. Pokud servisní účet či pojmenovaný uživatel provádí více operací (například zakládá více dokumentů) je žádoucí provést tyto operace v rámci jediného přihlášení (loginu) do DMS (například pokud se jedná o dávkové zpracování, lze na volající straně sloučit požadavky jednotlivých uživatelů podle jejich loginu). Tím se výrazně sníží zátěž DMS a zvýší se rychlost požadovaných operací.

Uživatelé, jejichž účty budou akce v DMS provádět a budou se do DMS připojovat, musí v DMS existovat. V DMS mohou pojmenované uživatelské účty vzniknout pouze v případě přidělení příslušných logických rolí pro aplikace DMS v ISIM GUI. Servisní účty zakládá na vyžádání a po schválení garanta za oblast DMS na ČSSZ administrace DMS.

V průběhu implementace řešení využívajícího DMS jako úložiště dokumentů je nutná spolupráce vyvíjející strany s vývojovým týmem DMS zejména na :

- poskytnutí adres DMS API v daných prostředích a ověření dostupnosti těchto adres z volající strany,
- ověření dostupnosti adresy pro vyzvedávání dokumentů DMS v daných prostředích (dokumenty jsou z DMS předávány pomocí URL odkazu),
- API DMS je dostupné ve vývojovém prostředí SIS a dále ve všech (IP,TP,PP) prostředích ČSSZ,
- poskytnutí dokumentace k DMS API a případné konzultace,
- testování úplnosti a správnosti vyplňování metadat zakládaných profilů dokumentů,
- ověření správnosti postupu při volání operací metodami DMS,
- zpřístupnění aplikací DMS pro případnou kontrolu vytvářených i čtených dokumentů,
- provedení a vyhodnocení zátěžových a funkčních testů v testovacím prostředí v režimu simulace vytvoření reálné zátěže, která bude probíhat v produkčním prostředí.

Požadavek na implementaci http hlavičky do všech volání API DMS (request):

Na síťové vrstvě loadbalanceru ACE je pro požadavky směřované na API DMS zajištěna tzv. „session persistence“. Všechny aplikace používající přihlášení pomocí tokenu a přistupující k API DMS musí mít ve svých requestech jednoznačný identifikátor, který se vkládá do HTTP hlavičky a pomocí něhož je zajištěno na síťové



úrovni přesměrování všech požadavků stále na stejný server. Pro API DMS je nutno implementovat identifikátor:

- XX-CSSZDMSSessionHeader

Pro komunikaci na aplikační úrovni se využívá protokol HTTP 1.0. Pokud je využíván protokol HTTP 1.1. je třeba zajistit, aby v rámci jednoho TCP spojení byl použit pouze jeden konkrétní identifikátor.

V rámci implementace je vždy hodnota tohoto identifikátoru zvolena po dohodě s týmem DMS (např. loginname, EČ apod.).

Pokud aplikace pro přihlášení nepoužívá tokeny ale přímé volání API se jménem a heslem technického účtu, nemusejí zasílat session hlavičku. Tím dojde k rozležení požadavků mezi více fyzických serverů.

6.3. Součinnost po nasazení do provozu

Po nasazení řešení do produkčního prostředí se předpokládá zvýšený dohled ze strany administrace DMS i ze strany administrace přistupující aplikace. Je očekávána bezodkladná součinnost s garantem okolního systému komunikujícího s DMS pro případné okamžité vyřešení problémů, které se mohou objevit v ostrém provozu po nasazení do produkčního prostředí.

Požadavky na oblasti, které komunikují se systémem DMS, jsou podrobněji definovány v metodické příručce IN1_Ukladani_vstupnich_dat_a_dokumentu.



7. Příloha 1. Systém DMS - přehled

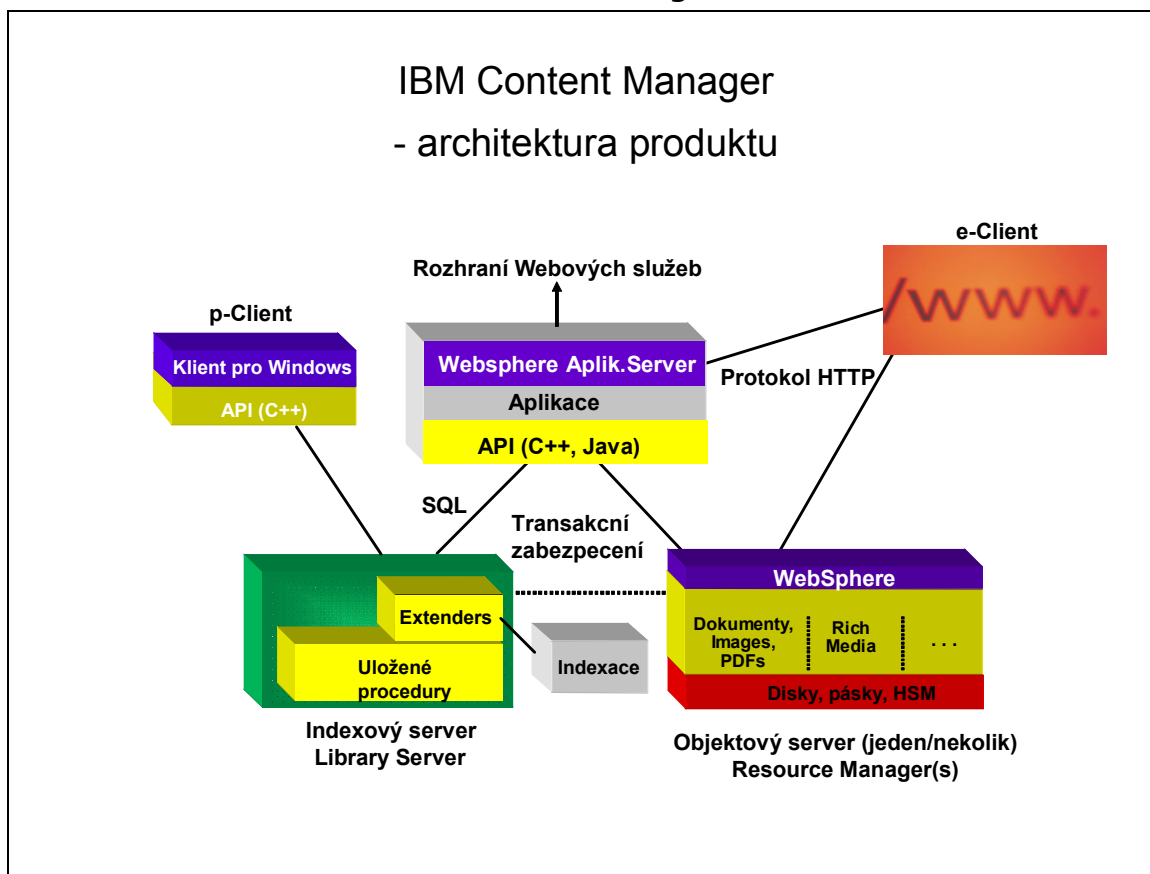
7.1. Přehled

Pro systém DMS na ČSSZ byl implementován produkt IBM Content Manager (DB2 Content Manager for Multiplatforms). Tato konkrétní implementace je nazývána Document Management Archiv (DMA).

Základní funkce DMA jsou:

- Centrální úložiště dokumentů s funkcemi dlouhodobé archivace vysokých počtů dokumentů.
- Univerzální správa dokumentů pro libovolný typ dokumentu, resp. objektu.
- Metadata (profil dokumentu) jako údaje popisující dokument (kategorizace) a jeho životní cyklus.
- Multikriteriální vyhledávání.
- Vícekanálový přístup z klientů a aplikací
- Zpřístupnění dokumentů prostřednictvím počítačové sítě (Intranetu).
- Řízení přístupových oprávnění k objektům (dokumenty, složky, procesy).
- Zajištění životního cyklu dokumentů a jeho sledování a logování.

7.1.1. Architektura IBM Content manager

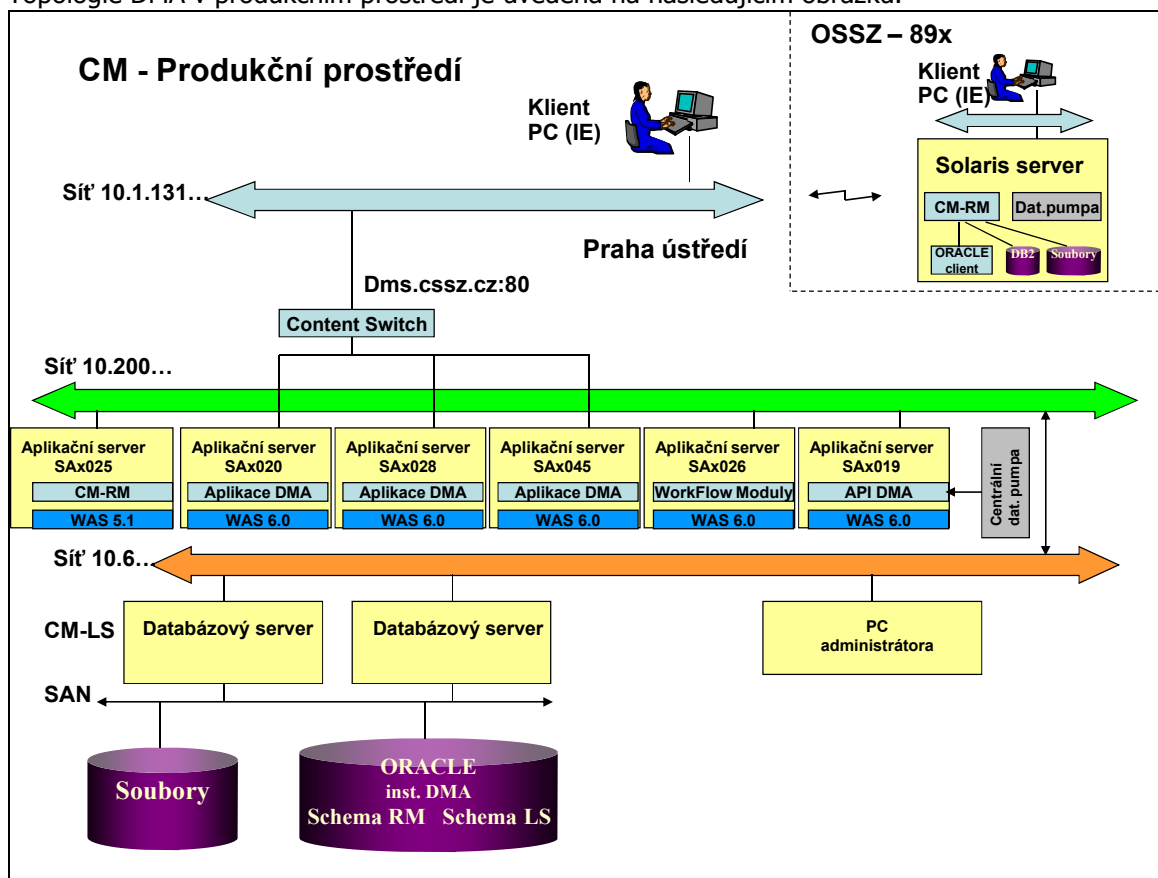


Architektura systému IBM Content Manager je vícevrstvá. Skládá se z jednoho centrálního indexového serveru (Library Server), jednoho nebo více objektových serverů (Resource Manager) a klientských pracovních stanic. Content Manager využívá relační databázi (v ČSSZ Oracle) pro ukládání indexových informací (metadat) o uložených dokumentech (objektech). Vlastní dokumenty (objekty) jsou uloženy na Objektových serverech v připojených diskových polích, nebo externích velkokapacitních knihovnách.



7.1.2. Implementace na ČSSZ – systém DMA

Topologie DMA v produkčním prostředí je uvedena na následujícím obrázku.



7.1.3. Klientské aplikace DMA

DMA má k dispozici následující klientské aplikace (stav platný k 18.02.2011)

Evidence podání

Pro evidenci podání je vytvořen klient Evidence podání.. Klient umožňuje zaevidování papírových i elektronických dokumentů (podání – i vlastních), které provádí uživatel (např. pracovník podatelny nebo referent přijímající podání).

Aktivní klient

Aktivní klient je používán pro vyhledávání a zobrazování naskenovaných dokumentů, opravu profilu dokumentů (indexů) a doplňování poznámek.

Univerzální klient

Univerzální klient DMA je určen pro prohlížení veškerých dokumentů uložených v DMA. Dokumenty je možno vyhledávat podle různých kritérií jako je evidenční číslo dokumentu, rodné číslo, variabilní symbol, číslo spisu, příslušnost k OSSZ, datum příjmu a podle kombinací těchto parametrů. Klient umožňuje zobrazení profilu (metadat) dokumentu, zobrazení vlastního dokumentu, jeho historie, zobrazení souvisejících dokumentů – např. ležících ve spisu. V případě verzování dokumentu umožňuje univerzální klient DMA i zobrazení předchozích verzí dokumentu.



Klient Administrátor

Systém Administration Client je určen pro nastavování parametrů IBM CM. Pomocí něj je možno nastavovat parametry Library serveru, Resource managerů, uživatele a jejich oprávnění, nastavovat datový model pro ukládání dokumentů (objektů) a nastavovat interní workflow systém (Document routing).

7.2. Dokument

7.2.1. Logický datový model dokumentu

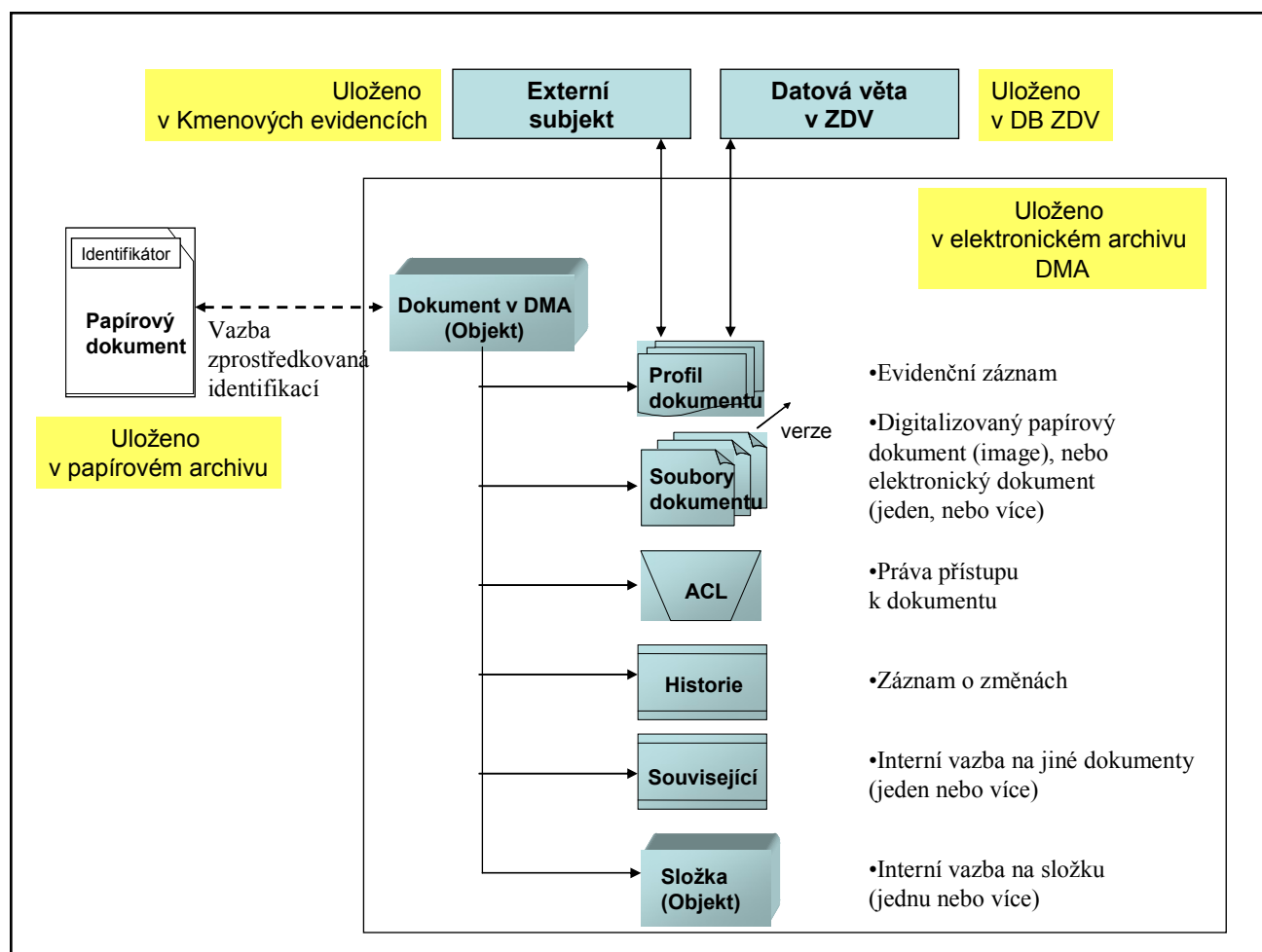
Každý dokument je v DMA uložen jako objekt s povinnou strukturovanou částí – tzv. profilem (metadata) – a vlastním souborem dokumentu (tělo dokumentu). Tělo dokumentu představuje obecně binární soubor v libovolném formátu (např.: .doc, .xls, .pdf, .dgv, .xml, html, .tif a další). Datový DMA model umožňuje ukládání proměnného, nespecifikovaného počtu těl k jednomu zaevidovanému dokumentu. Jednotlivá těla mohou být různého typu (tif, pdf, doc, apod.).

V DMS je vytvořen samostatný ItemType CZPSKEN, kam se zpravidla ukládá naskenovaný obraz dokumentu ve formátu MultiPage TIFF.

Pro ukládání více těl k jednomu dokumentu je v DMS vytvořen ItemType CZPBODY.

Pro potřeby ukládání těl z DIS je v DMS vytvořen ItemType DISBODY.

Kromě toho má každý dokument svou historii, nastavení přístupových práv a další informace. Logický datový model dokumentu v DMA a logické vazby dokumentu na ostatní objekty ukazuje následující obrázek.





7.2.2. Profil dokumentu – atributy dokumentů uložených v DMA

Následující tabulka obsahuje všechny atributy dokumentu, uloženého v DMA.

Červeným písmem jsou zvýrazněny atributy, které je nutno vyplňovat při evidenci nového dokladu do DMA.

Jedná se o povinné atributy, bez jejichž vyplnění není možno založit nový záznam v DMA.

Modrým písmem jsou zvýrazněny atributy, které je nutno vyplnit (pokud jsou známe) v okamžiku evidence nového dokladu do DMA. Jedná se ale o nepovinné atributy, bez jejichž vyplnění je možné nový záznam v DMA založit.

Jméno atributu	Popis atributu	Typ	Délka	Vyhledávat podle položky	Povinné	Poznámka
zakladniAtributy	Skupina základní atributy					
czg_nazev	Název nebo „věc“	AN	255	ano	ano	
czg_poznamka	Poznámka	AN	255	ne		
czg_isDeleted	Je zneplatněn	L	1	ano	ano	automaticky nastavuje API

spolecneAtributy	Skupina společné atributy					
czg_datumVytvoreni	Datum vytvoření	D		ano		automaticky nastavuje CM
czg_vytvoril	Vytvořil	ID uživatele		ano		automaticky nastavuje CM
czg_datumVyjmuti	Datum vyjmutí	D		ne		automaticky nastavuje CM
czg_vyjmul	Vyjmul k editaci	ID uživatele		ne		automaticky nastavuje CM
czg_datumZmeny	Datum změny	D		ne		automaticky nastavuje CM
czg_zmenil	Změnil	ID uživatele		ne		automaticky nastavuje CM
czg_itemType	Typ objektu	Number		ano	ano	automaticky nastavuje API
czg_acl	Nastavení práv přístupu			ne	ano	automaticky nastavuje API
czg_ed	Evidenční číslo dokumentu	Number	32	ano	ano	Ize použít max 15 čísel
czg_cj	Číslo jednací	AN	100	ano		
czg_cspis	Číslo spisu	AN	50	ano		
czg_typ	Typ dokumentu	Short	0-10000	ano	ano	číselník
czg_podtypy	Podtypy dokumentu	AN	255	ano		
czg_agenda	Příslušnost dokumentu k agendě	Short	0-10000	ano	ano	číselník
czg_stav	Údaj pro řízení životního cyklu	Short	0-10000	ano	ano	číselník, API nastavuje defaultní hodnotu
czg_mistoUloz	Specifikace úložného místa	AN	100	ne		
czg_spisZnak	Podle spisového a skartačního plánu	AN	20	po		automaticky nastavuje API podle číselníku typů dokumentů
czg_skartZnak	Podle spisového a skartačního plánu	AN	1	ne		automaticky nastavuje API podle číselníku typů dokumentů
czg_skartLhuta	Podle spisového a skartačního plánu	Short	0-1000	ne		automaticky nastavuje API podle číselníku typů dokumentů
czg_druh	Druh dokumentu (podání pap, podání el., vlastní pap. Vlastní el.)	Short	0-100	ano	ano	číselník, automaticky nastavuje API
czg_dDorVyd	Datum doručení / vydání (podání / vlastní)	D		ano	ano	
czg_dVyřízení	Datum vyřízení	D		ne		



Popis API rozhraní systému DMA

Projekt DMS

PROFINIT
new frontier group

Jméno atributu	Popis atributu	Typ	Délka	Vyhledávat podle položky	Povinné	Poznámka
czg_zVyrizeni	Způsob vyřízení	AN	255	ne		
czg_pOdbZprac	Proces odb.zpracování	Short	0-32000	ne		automaticky nastavuje API
czg_dSpOdbZprac	Datum spuštění procesu odborného zpracování	D		ne		automaticky nastavuje API
czg_cPrOdbZprac	Číslo instance procesu odborného zpracování	AN	30	ne		automaticky nastavuje API
czo_typFormul	Id typu formuláře	AN	20	ne	ne	

detailyKlienta	Skupina DetailyKlienta					
czk_ossz	Kód ÚP kde bylo přijato	Short	0-10000	ano	ano	číselník
czk_varSymbol	Identifikace organizace v rámci ČSSZ	N	16	ano		
czk_rodneCislo	Rodné číslo	N	16	ano		
czk_ico	IČO	N	21	ano		
czk_izdv	Identifikátor záznamu v ZDV	N	16	ne		
czk_narokovyP	Je to nárokový podklad	L	1	ne		automaticky nastavuje API podle číselníku typů dokumentů
czk_ikeos	Identifikátor záznamu v KE	OID	25	ne		OID podle osoby
czk_ikeor	Identifikátor záznamu v KE	OID	25	ne		OID podle organizace
czk_jmenoOs	Jméno osoby z KE	AN	50	ano		hodnota z KE
czk_prijmeniOs	Příjmení osoby z KE	AN	50	ano		hodnota z KE
czk_nazevOr	Název organizace z KE	AN	255	ano		hodnota z KE

obecnePodani	Skupina Obecné Podání					
czd_zPodani	Způsob podání	Short	0-1000	ne		číselník
czd_zDoruceni	Způsob doručení	Short	0-1000	ano		číselník
czd_zVstZprac	Způsob vstup. zpracování	Short	0-1000	ne		číselník
czd_sVstZprac	Stav vstup. zpracování	Short	0-1000	ne		číselník
czd_dUkVstZprac	Ukončení vstup. zpracování	D		ne		
czd_dPodani	Datum podání	D		ne		zatím není sledováno
czd_obdobi	Období dokumentu	AN	15	ne		
czd_dzTextu	Datum uvedené v textu dokumentu	D		ne		

vlastniAtributy	Skupina Vlastní dokumenty					
czl_podepsal	Podepsal	AN	50	ne		
czl_dPodpisu	Datum podpisu	D		ne		

papirovePodani	Skupina Papírové podání					
czo_cdz	Číslo doporučené zásilky - Pošta	AN	50	ne		
czo_jmeno	Z obálky: Jméno a Příjmení / Název organizace	AN	80	ne		
czo_adresa	Z obálky: Adresa	AN	80	ne		
czo_pocetListu	Počet listů	Short	0-1000	ne	ano	povinné při skenování
czo_pocetPriloh	Počet příloh	Short	0-1000	ne		zatím není sledováno
czo_psc	Z obálky: PSČ	AN	6	ne		



Popis API rozhraní systému DMA

Projekt DMS

PROFINIT
new frontier group

Jméno atributu	Popis atributu	Typ	Délka	Vyhledávat podle položky	Povinné	Poznámka
czo_mesto	Z obálky: Město	AN	50	ne		

elektronickePodani	Skupina Elektronické podání					
czo_autorizace	Způsob autorizace	AN	100	ne		
Skupina DIS	Skupina DIS					
czx_dPotvrzeni	datum potvrzení	D		ne	ne	
czx_potvrdil	potvrzeno kým	AN	50	ne	ne	
czx_iCSSZ	identifikátor CSSZ	AN	32	ano	ano	
czx_iPVS	identifikátor PVS	AN	32	ano	ano	
DIScripted	zasifrovaná data podání	Část dokumentu	xml	ne	ne	
DISuncrypted	rozsifrovaná data podání	Část dokumentu	xml	ne	ne	
DISprotokol	protokol o zpracování	Část dokumentu	xml	ne	ne	
Nezařazené atributy						
czk_zic	ZIČ	N	21	ano		
czk_varSymbol8	Variabilní symbol 8	N	16	ano		
Tělo dokumentu						
CZPSKEN	Tělo dokumentu					
mimetype	MimeType	AN		ne	ano	Lze ukládat 1 tělo typu CZPSKEN k jednomu dokumentu v DMS
uuid	UUID identifikující jednoznačně objekt těla	AN		ne		automaticky nastavuje CM
czo_nazev	Název souboru	AN	255	ne		Jako název souboru je doporučeno ukládat název těla (dokumentu) , např. PROTOKOL.doc
url	URL adresa, z které se dá získat binární obsah těla dokumentu.	AN		ne		
body	Binární obsah těla dokumentu.	Binary		ne		Pouze vstupní parametr - na výstupu není nastavován
czo_sha1	Hash tiffu	AN	50	ne		
czo_subdivision	Pracoviště ČSSZ	AN	20	ne		
czo_nazSken	Název skeneru	AN	15	ne		
czo_porVeSpisu	Pořadové číslo ve spisu	AN	15	ne		
czo_dSken	Datum skenování	D		ne		
czo_pcSVlozky	Pořadové číslo spisové vložky	AN	15	ne		
czo_cdavky	Číslo dávky	AN	20	ne		
czo_pocetStran	Počet stran	N	0-10000	ne		
CZPBODY	Obecné tělo dokumentu					Doporučeno ukládat max 15 těl typu CZPBODY k jednomu dokumentu DMS
czo_dBody	Datum	D		ne	ne	
czo_nazev	Název souboru	AN	255	ne	ne	
czo_sha2	Hash	AN	255	ne	ne	



Jméno atributu	Popis atributu	Typ	Délka	Vyhledávat podle položky	Povinné	Poznámka
czo_typ	Typ těla	N	0-10000	ne	ano	Číselník typů těl - Ident.kód
czo_poznamka	Poznámka	AN	255	ne	ne	
DISBODY	Obecné tělo dokumentu					Doporučeno ukládat max 15 těl typu DISBODY k jednomu dokumentu DMS
czo_dBody	Datum	D		ne	ne	
czo_nazev	Název souboru	AN	255	ne	ne	
czo_sha2	Hash	AN	255	ne	ne	
czo_typ	Typ těla	N	0-10000	ne	ano	Číselník typů těl - Ident.kód
czo_poznamka	Poznámka	AN	255	ne	ne	
esbPodani	Skupina Elektronické podání					
czx_iDS	ID datové schránky	AN	7	Ano	Ne	
czx_iDZ	ID datové zprávy	AN	20	Ano	Ne	
czx_iESSUID	Identifikátor podání v ESS	AN	50	Ano	Ne	
czx_iESSPrilohy	Identifikace přílohy DZ v ESS	AN	50	Ne	Ne	
czx_email	E-mail odesílatele/příjemce	AN	254	Ne	Ne	
univerzalniIdentifikatory	Skupina Univerzální identifikátory					
czx_iXX01	Univerzální identifikátor 01	AN	255			
czx_iXX02	Univerzální identifikátor 02	AN	255			
czx_iXX03	Univerzální identifikátor 03	N	32			
czx_iXX04	Univerzální identifikátor 04	N	16			

7.2.3. Řízení práv přístupu k dokumentům

Přístup uživatelů k dokumentům uloženým v DMA je možný jak z DMS aplikací (např. Aktivní klient, nebo Evidence podání), tak přímo z odborných aplikací. Veškerý přístup k dokumentům uloženým v DMA je plně podřízen kontrole práv přístupu nezávisle na tom, ze které aplikace k nim uživatel přistupuje. Práva přístupu na objekt dokumentu jsou definována pomocí mechanismu Access Control List (ACL). Tento mechanismus umožňuje ke každému dokumentu definovat různá práva přístupu (vyhledat, číst, editovat, ...) pro různé uživatele, či jejich skupiny.

Správa přístupu, která je integrální částí CM, je založena na následujících pravidlech:

- Každý uživatel má administrátorem přiřazenu základní sadu oprávnění (Privilege set), které definuje jeho maximální práva ke všem objektům CM,
- Každý objekt datové struktury má přiřazen tzv. Access Control List (ACL), který určuje maximální přístupová práva k tomuto objektu pro uživatele či skupiny.
- ACL nemůže poskytnout uživateli větší oprávnění k objektu než má definováno ve své základní sadě. Naopak, ACL může uživateli odebrat určité oprávnění k objektu, i když ho uživatel v základní sadě má k dispozici.

Informace o nastavení uživatelských profilů i nastavení ACL jsou uloženy do relační databáze indexového serveru (LS).



8. Příloha 2. Číselníky

Následující tabulky uvádějí strukturu a hodnoty použitých číselníků.

DRUH DOKUMENTU

Položka	Popis
PodaniPapírove	Papírové podání (např. žádost 357, nebo vytištěná žádost ZDD)
PodaniElektronicke	Elektronické podání (např. DIS)
InterniDokumentPapírovy	Vlastní papírový dokument
InterniDokumentElektronicky	Vlastní elektronický dokument
DokumentZDV	Dokument ZDV
PodaniHlasove	Hlasové podání
TMP	Systémový druh dokumentu

AGENDA

Položka	Typ	Popis
KOD	N	Kód
NAZEV	AN	Název agendy

Kód	Název agendy
10	Dobrovolné důchodové pojištění
20	Důchodové listy

Hodnoty jsou v samostatné tabulce

TYP

Položka	Typ	Popis
KOD	N	Kód
NAZEV	AN	Název typu
CIS_RADA	LOV	Číselná řada
I_AGENDA	LOV	Agenda
I_SPZ	AN	Spisový znak
PRISTUP	LOV	ACL
czk_narokovyP	L	Je to nárokový podklad
I_WF	LOV	Odborný proces

Kód	Název typu	Číselná řada	Agenda	Spisový znak	nárokový podklad	Odborný proces
10	Evidenční list	ZAKL	REG	4601.1	ano	
Hodnoty jsou v samostatné tabulce						
		



OSSZ

Položka	Typ	Popis
KOD	N	Kód
NAZEV	AN	Název OSSZ

Kód	Název
110	Praha 10
772	Brno - město

Hodnoty jsou v samostatné tabulce

STAV

Položka	Typ	Popis
KOD	N	Kód
NAZEV	AN	Název stavu

Kód	Název stavu
10	Čeká na tělo/image
100	Vstup kompletní
500	archivován
600	ke skartaci
1000 -	
1299	rezervováno pro DIS: podání ve zpracování v DIS
1001	systemu podání předáno do cílových
1002	systemu
1101	podání po výzvě nedoplněno
1102	stornované podání
1103	zamítnuté podání

ZPŮSOB PODÁNÍ

Položka	Typ	Popis
KOD	AN	Kód
NAZEV	AN	Název způsobu podání

Kód	Název způsobu podání
1	písemné
2	elektronické
3	ústní

ZPŮSOB DORUČENÍ

Položka	Typ	Popis
KOD	AN	Kód
NAZEV	AN	Název způsobu doručení



Kód	Název způsobu doručení
1	neuvedeno
2	osobně
3	poštou
4	kurýrem
5	e-mail
6	PVS
7	faxem
8	telefonicky
9	e-podatelnou
10	poštou doporučeně
11	paměťové médium
12	přetypováno do formuláře
13	DIS vyjma PVS
14	VREP
15	DS

ZPŮSOB VSTUPNÍHO ZPRACOVÁNÍ

Položka	Typ	Popis
KOD	N	Kód
NAZEV	AN	Název způsobu vstup. zprac.

Kód	Název způsobu vstup. zprac.
1	neurčeno
100	skenování
200	zpracování na DIS
300	OCR vytěžování na centrálním prac.
400	pouze evidence

STAV VSTUPNÍHO ZPRACOVÁNÍ

Položka	Typ	Popis
KOD	N	Kód
NAZEV	AN	Název stavu vstup. zprac.

Kód	Název stavu vstup. zprac.
1xx	Evidováno na OSSZ
2xx	Naskenováno na OSSZ
3xx	Předáno BTS
4xx	Převzato OCR
5xx	Chyba na OCR
6xx	DV uložena do ZDV

Poznámka: Stav vstupního zpracování pro účely vytěžování OCR (při způsobVstupZprac=300) je třiciferné číslo, kde nejvyšší řád udává přímo hodnotu stavu, prostřední řád funguje jako čítač pro počet kolikrát je spuštěn proces změny typu a nejnižší řád kolikrát je spuštěn požadavek na resken. (Pozn.: implementačně je zajištěno že pokud tento počet dosáhne 9, vynuluje se) Pokud proběhlo vytěžení úspěšně, stav vstupního zpracování dokumentu (nejvyšší řád) se nastaví na hodnotu 6 (uloženo do ZDV). Při požadavku na resken nebo na změnu typu se nastaví na hodnotu 5 (chyba na OCR).

ZPŮSOB Autorizace



Položka	Typ	Popis
KOD	N	Kód
NAZEV	AN	Název způsobu autorizace

Kód	Název způsobu doručení
1	osobní doručení
2	podpisem na průvodním dopise
3	ověřením elektronického podpisu
301	Kvalifikovaný (qualified)
302	čssz
303	ossz
304	Jiný (other)
305	Neznámý (unknown)

9. Příloha 3. Číselník typů těl

Aktuální číselník platných typů těl k 02/2011

Ident.kód	Název	Popis	MimeType
1	DokumentTIFF	Dokument ve formátu TIFF	image/tiff
2	DokumentPDF	Dokument ve formátu PDF	application/pdf
3	DokumentExcel	Dokument ve formátu MS Excel	application/vnd.ms-excel
4	DokumentWord	Dokument ve formátu MS Word	application/msword
5	DokumentTXT	Dokument ve formátu TXT	text/plain
6	DokumentCSV	Dokument ve formátu CSV	text/csv
7	DokumentXML	Dokument ve formátu XML	text/xml
8	DokumentJPEG	Dokument ve formátu JPEG	image/jpeg
9	DokumentHTML	Dokument ve formátu HTML	text/html
10	DokumentExcel2007	Dokument ve formátu MS Excel 2007	application/vnd.openxmlformats-officedocument.spreadsheetml.sheet
11	DokumentWord2007	Dokument ve formátu MS Word 2007	application/vnd.openxmlformats-officedocument.wordprocessingml.document
12	DokumentZIP	Dokument ve formátu ZIP	application/zip
13	DokumentRTF	Dokument ve formátu RTF	application/rtf
51	DIScrypted	Zašifrovaná data podání z DIS (GovTalk)	text/xml
52	DISuncrypted	Rozšifrovaná data podání z DIS (podání)	text/xml
53	DISprotokol	Protokol o zpracování z DIS	text/xml
54	DISresponse	Nezašifrovaná odpověď DISu (data odpovědi)	text/xml
55	DISresponseencrypted	Zašifrovaná odpověď DISu (GovTalk)	text/xml
81	ISDSzprava	Originál celé zprávy z ISDS (ZFO)	application/vnd.software602.filler.form+xml-zip
82	ISDSodpoved	Odpověď zasláná na ISDS (ZFO)	application/vnd.software602.filler.form+xml-zip
83	ISDSdoručenka	Doručenka (ZFO)	application/vnd.software602.filler.form+xml-zip



Popis API rozhraní systému DMA

Projekt DMS

PROFINIT
new frontier group

99	DokumentTMP	Neznámý formát dokumentu	text/plain
----	-------------	--------------------------	------------