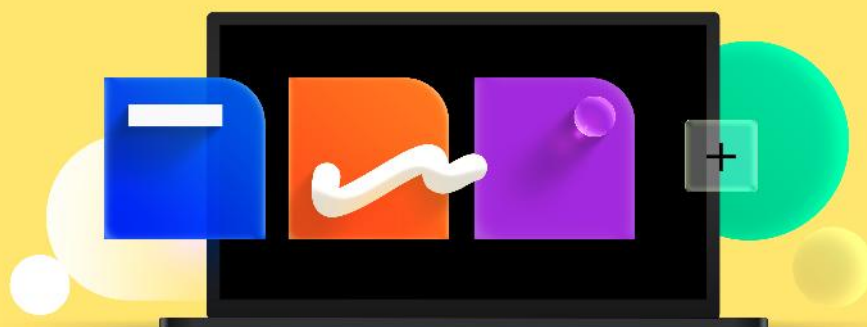




FormFlow Webové služby

Aktualizováno: 15.12.2025

Obsah

1	Úvod	3
1.1	Logování	3
2	Dokumentová část	5
2.1	Změna pro službu ServiceDG pro práci s číselníky od verze FormFlow 10.2.2	5
2.2	Návratové stavy služby ServiceDG	7
2.3	Služba ServiceDG, getDocument	10
2.4	Služba ServiceDG, setDocument	13
2.5	Služba ServiceDG, searchDocument	18
2.6	Služba ServiceDG, createDocumentVersion	23
2.7	Služba ServiceDG, lockDocumentByUser	24
2.8	Služba ServiceDG, createDocumentRelation	25
2.9	Služba ServiceDG, createDocumentRelationCollection	26
2.10	Služba ServiceDG, deleteDocumentRelation	28
2.11	Služba ServiceDG, searchDocumentRelation	29
2.12	Služba ServiceDG, getDocumentVersions	31
2.13	DzhpService, insertDocument	33
3	Elektronická spisová služba (ESS)	35
3.1	Autentizace webových služeb ESS	35
3.2	Instalace	35
3.3	Profily	36
3.4	Obecné datové struktury	36
3.5	Seznam položek čísla jednacího: CjList	43
3.6	Seznam příloh: DocumentList	43
3.7	Vložení dokumentu do spisu – ESSAddDocumentToSpis	43
4	Práce s dokumenty v LTV	60
4.1	Služba Service602LTV, computeHashDocument	60
4.2	Služba Service602LTV, LtvAnalyze	62
4.3	Služba Service602LTV, LtvDeleteTenant	64
4.4	Služba Service602LTV, LtvGetDocumentInfo	65
4.5	Služba Service602LTV, LtvInsert	66
4.6	Služba Service602LTV, LtvRegisterTenant	67
4.7	Služba Service602LTV, LtvUnregister	68
4.8	Služba Service602LTV, SignDocument	69
4.9	Služba Service602LTV, TimestampPdfDocument	71
5	Návratové stavy WS	72
5.1	Úspěšné zpracování	72
5.2	Rozsah 100–199 – chyby konfigurace a volání	72
5.3	Rozsah 200–299 - chyby běhu	73
5.4	Rozsah 300–399 – databázové chyby	75

1 Úvod

1.1 Logování

Od verze 10.2.3 se implicitně loguje provoz webových služeb a klientů, které jsou potomky **FS\WSService**, resp. **FS\WSCient**. Toto lze vypnout nastavením konfigurační proměnné **XMLGW_CSV_LOG_WS_SERVER**, resp. **XMLGW_CSV_LOG_WS_CLIENT** na hodnotu false.

Loguje se do **XMLGW_LOGDIR/XMLGW_NAME_ws.RRRR-MM-DD.log.csv**.

Struktura csv odpovídá třídě **FS\WSLogRec**.

Dumpování provozu je možno nastavit v config.php pomocí define

```
define('XMLGW_WS_LOGDIR', XMLGW_LOGDIR . '/ws_log');
```

Pozor, v dumpu jsou i obsažené dokumenty a jiná data, která vyžadují ochranu.

```
class WSLogRec
{
    /** @var string datum zapisu requestu do csv */
    public $date;
    /** @var int */
    public $appServerId;
    /** @var int */
    public $pid;
    /** @var int */
    public $threadId;
    /** @var string */
    public $srcIp;
    /** @var int */
    public $id_user;
    /** @var string */
    public $userAgent;
    /** @var float Cas v sekundach prijeti dat. V modu S nelze zjistit.*/
    public $inDataRecvTime;
    /** @var int Poce prijatych bajtu. V modu S, pokud neni nastaveno
XMLGW_WS_LOG_DIR, se pouzije hodnota Content-Length z hlavicek pozadavku.*/
    public $inDataRecvBytes;
    /** @var float Cas v sekundach vyrizovani pozadavku pred samotnym zpracovanim
logikou dane webove metody. Zapisovano jen v modu S.*/
    public $prepareTime;
    /** @var float Cas v sekundach zpracovanim logikou dane webove metody.
Zapisovano jen v modu S.*/
    public $handleTime;
    /** @var float Cas v sekundach odesilani soap odpovedi v modu S, resp. soap
pozadavku v modu C.*/
    public $outDataSentTime;
    /** @var int Pocet odeslanych bajtu odpovedi v modu S, resp. pozadavku v modu
C.*/
    public $outDataSentBytes;
    /** @var string 'S' pro webovou sluzbu, 'C' pro klientskou webovou sluzbu. */
    public $mode;
    /** @var string relativni endpoint v modu S, anebo plne url v modu C. */
    public $url;
    /** @var string pouzita metoda, musi podporovat implementace dane webove
sluzby. Zapisovano jen v modu S.*/
    public $method;
    /** @var string Pokud nastaveno XMLGW_WS_LOG_DIR, relativni cesta k souboru s
dumpem komunikace.*/
    public $dumpFile;
```

```
/** @var string Id pro business svazani pozadavku. Muze byt bude vyplneno v
http hlavicece pozadavku X-CorrelationId anebo v parametrech kontrektu ws -
* toto musi podporovat. Zapisovano jen v modu S.*/
public $correlationId;
/** @var string Vysledek volani metody. Musi podporovat konkretni ws.
Zapisovano jen v modu S.*/
public $result;
/** @var string Obecna vyjimka.*/
public $fatalError;
}
```

2 Dokumentová část

Webové služby (WS) jsou součástí standardní distribuce FormFlow, není potřeba další konfigurace. Webová služba ServiceDG má dva koncové body:

- `/htdocs/soap/ServiceDGEndpoint.php` – pro tuto variantu se použije nastavení autentizace pomocí nette služby `ws_authenticator` a je primárně určena pro obsluhu FormFlow Office Extension.
- `/htdocs/soap2/ServiceDGEndpoint.php` – alternativa, která se využije v případě, že cizí systém potřebuje jiný model autentizace než FormFlow Office Extension. Např. SSO vs. basic autentizace. Pokud se požaduje Windows autentizace nebo základní ověření Windows, je důležité pro obě alternativy nastavit autentizaci s adresáři `/soap`, `/soap2` v IIS.

2.1 Změna pro službu ServiceDG pro práci s číselníky od verze FormFlow 10.2.2

Ve všech případech se přidaly do sekce **meta** dva nepovinné parametry: **value_code** a **value_name**. Na vstupu bude **value_code** mít přednost před původním **value**, na výstupu se bude využívat vždy **value_code** a jen u číselníkových položek se bude používat **value_name**.

Tato změna je zde zavedena z důvodu možnosti zasílání kódů (nikoliv databázových **int ID**) a také z důvodu možnosti zobrazit texty číselníků. Změny se týkají služeb:

- `getDocument`
- `searchDocument`
- `setDocument`.

Pokud se nebude jednat o číselníkovou položku nebo specifickou položku (viz dále), bude **value_code** shodné s **value**, aby zdrojový systém nemusel řešit, zda se jedná či nejedná o číselníkovou položku. Pokud bude na vstupu vyplněno **value_code** má vždy přednost před **value**.

Změny na vstupu služeb

- Pokud bude na vstupu uveden neprázdný **value_code**, bude se u nečíselníkových (a speciálních polí viz níže) chovat stejně, jako se nyní chová **value**, tedy hodnota metadata (**value_code** má přednost před **value**)
- Pokud bude na vstupu uveden neprázdný **value_code** a bude se jednat o číselníkovou hodnotu (nebo speciální hodnotu viz dále), bude **value_code** reprezentovat tato data:
 - Třída – kód třídy
 - Klasifikace – kód klasifikace
 - Stav – kód stavu
 - Složka – SID složky
 - Ostatní číselníky – klíč v číselníku.

Změny na výstupu služeb

- Na výstup se přidají parametry **value_code** a **value_name** (**value_name** jen u číselníkových položek).

Kde **value_code**:

- Kód třídy
- Kód klasifikace
- Kód stavu
- Sid složky
- Klíč položky v číselníku
- U ostatních položek (text, datum atp.) bude hodnota stejná jako ve **value**.

Value_name:

- název třídy
- název klasifikace
- název stavu
- název Složky
- Název položky číselníku
- U ostatních položek (text, datum atp.) se element nebude uvádět (případně bude prázdný).

Dopady:

Vyšší datový tok – na výstup se předávají data, která nejsou vždy potřeba.

Řešení:

Na vstup bude přidán element **return_value_code_name_only**, který bude typu *boolean*:

- **true** – bude se vracet na výstup jen value_code a value_name, nebude se vracet value
- Neuvedeno nebo **false** nebo prázdné nebo neuvedeno – bude se vracet jen value tak jako doposud.

Pozn. Pokud je **return_value_code_name_only** neuvedeno nebo **false**, tak pro číselníková metadata je **value_code** a **value_name** vráceno vždy.

2.2 Návrátové stavy služby ServiceDG

Kód chyby	Chybový text	Význam
0	""	Volání WS proběhlo bez chyby
-1	E_GENERAL	Obecná chyba
-2	E_SQL	Obecná sql chyba
-3	E_UNKNOWN_FOLDER	Neznámý adresář
-4	E_UNAUTHORIZED	Neoprávněný požadavek
-5	E_INVALID_ARGUMENT	Neplatný argument
-6	E_EMPTY_INPUT_FILE_ARRAY	Prázdný vstupní pole soubor
-7	E_INVALID_INPUT_FILE_BASE64	Prázdný vstupní pole soubor ve tvaru base64
-8	E_EMPTY_MD5	Prázdný MD5 součet
-9	E_SFLF	Chyba při práci s externím úložištěm souborů
-10	E_EMPTY_EXTERNAL_SYSTEM_ID	Prázdný ID externího systému
-11	E_UNKNOWN_MIME_TYPE	Neznámý MimeType
-12	E_ASYNCHRONOUS_INSERTION_PERMANENTLY_FAILED	Asynchronní vložení trvale selhalo
-13	E_EMPTY_FILENAME_OF_MERGED_FILE	Prázdný název sloučeného souboru
-14	E_INVALID_FILENAME_OF_MERGED_FILE	Nesprávný název sloučeného souboru
-15	E_UNKNOWN_DOC_STATUS	Neznámý dokumentový status
-16	E_NO_FILE_CONTENT	Žádný obsah souboru
-17	E_AMBIGUOUS_FILE_CONTENT	Nejednoznačný obsah souboru
-18	E_NO_CONTRACT_NO	Žádná smlouva číslo
-19	E_DOCUMENT_NOT_FOUND	Dokument nebyl nalezen

-20	E_DOCUMENT_BINARY_DATA_NOT_AVAILABLE	Binární data dokumentu nejsou dostupná
-21	E_UNKNOWN_SID_FLDR	Neznámý SID dokumentové složky
-22	E_MULTIPLE_FILES_AND_FILE_ECM_ID_SPECIFIED	Požaduje se mělká kopie z existujícího dms dokumentu, ale požaduje se současně slévání.
-23	E_UNSPECIFIED_FOLDER	Nespecifikovaná složka
-24	E_UNINSUFFICIENT_SPIS_ZNAK_METADATA	Nespecifikovaný spisový znak
-25	E_NO_FILE_MARK	Žádná značka souboru
-26	E_INVALID_FILE_MARK	Neplatná značka souboru
-27	E_INVALID_DISCARD_CODE	Neplatný kód zrušení
-28	E_INVALID_DISCARD_PERIOD	Nesprávná perioda zrušení
-29	E_UNKNOWN_CONSTRAINT_TYPE	Neznámý typ omezení
-30	E_UNKNOWN_CONSTRAINT_OPERATOR	Neznámý operátor omezení
-31	E_DZHP_INSERT_SOME_FAILED	Chyba na vstupu z DZHP
-32	E_SQL_DEADLOCK	SQL DeadLOCK
-33	E_INPUT_BINARY_DATA_MUST_BE_SET	Soubor na vstupu, data musí být vyplněna
-34	E_INPUT_DOCUMENT_RELATION_IS_EMPTY	Dokument na vstupu, vztah je prázdný
-35	E_INPUT_DOCUMENT_RELATION_PRIMARY_ID_DOCU_IS_EMPTY	Dokument na vstupu, ID_DOCU je prázdné
-36	E_INPUT_DOCUMENT_RELATION_RELATED_ID_DOCU_IS_EMPTY	Není specifikováno ID_DOCU pro vytvoření vazby
-37	E_INPUT_DOCUMENT_RELATION_RELATED_NAME_IS_EMPTY	Není specifikován název dokumentu pro vazbu
-38	E_INPUT_SEARCH_DOCUMENT_RELATION_IS_EMPTY	Hledaná vazba není specifikována

-39	E_INVALID_SEARCH_DOCUMENT_RELATION_DIRECTION_VALUE	Neplatná hodnota směru vazby dokumentů
-40	E_DOCUMENT_RELATION_DOESNT_EXIST	Specifikovaná vazba neexistuje
-41	E_EMPTY_REQUEST	Prázdný požadavek
-42	E_DOCUMENT_IN_SPIS_CANNOT_BE_DELETED	Dokument vložený do spisu nemůže být smazán
-43		Nevyužito
-44	E_CANNOT_WRITE_FILE_TO_UNC_PATH	Nelze zapsat soubor do specifikované unc cesty
-45	E_CANNOT_FIND_VERSION	Nenalezena verze dokumentu
-46	E_CANNOT_GET_VERSION_BINARY	Nenalezen binární soubor verze
-47	E_REQUIRED_METADATA_IS_EMPTY	Požadovaná metadata jsou prázdná
-48	E_UNKNOWN_ID_CERT_DOCU	Neznámý obsluhovaný subjekt ID_CERT_DOCU
-49	E_UNKNOWN_ID_CERT_DOCU_IS_MISSING	Je zadán spisový znak, ale nebyl zadán obsluhovaný subjekt
-50	E_UNKNOWN_FILE_MARK_DOCU	Neznámá značka dokumentu
-51	E_UNKNOWN_ID_ARIS_DOCU	Požaduje se překlad kódu vkladatele, ale není definováno ID_ARIS_DOCU, id externího systému
-52	E_FILE_MARK_DOCU_IS_MISSING	Značka dokumentu je ztracena
-53	E_INVALID_SHRED_MARK_DOCU	Neplatný skartační znak
-54	E_SHRED_TIME_DOCU_IS_MISSING	Je zadán skartační znak, ale nebyla zadána skartační lhůta
-55	E_EMPTY_ID_ARDE_VALUE_NAME	Požaduje se založení nového vkladatele, ale položka value_name je prázdná.

2.3 Služba ServiceDG, getDocument

Tato metoda slouží pro získání jednoho dokumentu na základě jeho **id** nebo dvojice hodnot (**external_id_docu**, **external_system_id_docu**).

Vstup:

```
class GetDocumentReq {  
    public $id_docu; // int  
    public $external_id_docu; // string  
    public $external_system_id_docu; // string  
    public $fileRequired; // boolean  
    public $metaList; // MetaSpecList  
    public $getAllMetaData; // boolean  
    public $tryLockFile; // boolean  
  
    public $return_value_code_name_only; // boolean, od verze FF 10.2.2  
    public $return_xml_datetime; // boolean, od verze FF 10.2.2  
    public $fileUncPath; // string, od verze FF 10.2.3  
    public $version_dver; //int, od 10.2.2.0.3  
    public $id_dver; // int, od 10.2.2.0.3  
    public $correlatinId // Nepovinné, od 10.2.3  
}
```

- **\$id_docu**: identifikátor dokumentu.
- **\$external_id_docu**: externí id dokumentu (DG_DOCU > EXTERNAL_ID_DOCU).
- **\$external_system_id_docu**: id externího systému (DG_DOCU > EXTERNAL_SYSTEM_ID_DOCU).
- **\$fileRequired**: určuje, zda je požadován binární obsah aktuální verze dokumentu. Lze totiž požadovat jen metadata.
- **\$metaList**: je seznam požadovaných metadat.
- **\$getAllMetadaData**: je alternativou k **\$metaList**, aby se dodala všechna metadata, aniž by bylo nutné znát jejich výčet.
- **\$tryLockFile**: zastaralé, nepoužívat
- **\$return_value_code_name_only**: hodnoty metadat ve struktuře **MetaValue** nebudou v položce **\$value**, ale ve **\$value_code**. Pro číselníky, třídy, klasifikace a stavy bude ve **\$value_code** klíč/kód číselníku a ve **\$value_name** název položky číselníku.
- **\$return_xml_datetime**: Pokud true, hodnoty datových typů date a datetime budou vráceny jako řetězec ve formátu ISO8601, jinak ve formátu Y-m-d H:i:s (php specifikace).
- **\$fileUncPath**: pokud je parametr vyplněn, služba na výstupu nebude vracet binární obsah dokumentu, ale uloží ho na požadovanou UNC cestu.
 - UNC cesta musí být pro DMS dostupná pro zápis.
 - Soubor uložený na UNC cestu DMS po sobě nemaže, toto provádí konzument služby.
 - UNC cesta bude obsahovat celou cestu, tzn. i s názvem souboru.
- **\$version_dver**; Nepovinné. Od 10.2.2.0.3 pořadové číslo verze. Pokud uvedeno, ve vráceném bináru bude binár požadované verze dle tohoto pořadového čísla verze. Id_docu musí být vyplněno.

- **\$id_dver**; Nepovinné. Od 10.2.2.0.3 systémově unikátní číslo verze. Pokud uvedeno, ve vráceném bináru bude obsah této verze, id_docu pak nemusí být vyplněno. id_dver má přednost před id_docu a (\$external_id_docu, \$external_system_id_docu).
- **\$correlationId**: Nepovinné. Pro seskupení identifikace několika volání do logů FormFlow.

Výstup:

```
class GetDocumentResp {
    public $result; // int
    public $resultText; // string
    public $metaValueList; // MetaValueList
    public $file; // DocumentFile
}
```

- **\$resultText**: stavový kód, 0 je OK. Číselník kódů je ve výčtu FS\DG\WS\WSStatus.
- **\$resultText**: stavová zpráva, typicky chybový text.
- **\$metaValueList**: seznam metadat s jejich hodnotami, pokud bylo požadováno. Hodnoty jsou vždy typu string.
- **\$file**: struktura obsahující binární data aktuální verze dokumentu, pokud bylo požadováno.

Doplňující informace:

Pokud jsou přenášeny informace typu datum či datum a čas, je očekáván formát ISO 8601.

Metodu lze zavolat i bez požadavku na binární data a metadata, pak se jedná o pouhý test existence dokumentu.

```
class MetaValue extends MetaSpec
{
    /* @var string IN/OUT. */
    public $value;

    /** @var string IN/OUT. Je možné použít v případě výtce pro klíč položky výtce. */
    public $value_code;

    /** @var string OUT. Je navracena v případě výtce hodnota položky výtce. */
    public $value_name;
}
```

Příklad volání – požadavek:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://dg.software602.cz/">
  <SOAP-ENV:Body>
    <ns1:getDocument>
      <req>
        <id_docu>175388</id_docu>
        <fileRequired>true</fileRequired>
      </req>
    </ns1:getDocument>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```

        <metaList/>
        <getAllMetaData>true</getAllMetaData>
        <return_value_code_name_only>true</return_value_code_name_only>
        <return_xml_datetime>true</return_xml_datetime>
    </req>
</ns1:getDocument>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Příklad volání – odpověď:

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://dg.software602.cz/">
  <SOAP-ENV:Body>
    <ns1:getDocumentResponse>
      <return>
        <result>0</result>
        <metaValueList>
          <meta>
            <name_meta>ID_FLDR_DOCU</name_meta>
            <dbtable_meta>DG_DOCU</dbtable_meta>
            <dbcolumn_meta>ID_FLDR_DOCU</dbcolumn_meta>
            <value />
            <value_code>rsts1</value_code>
            <value_name>Klientská</value_name>
          </meta>
          <meta>
            <name_meta>ID_CLAS_DOCU</name_meta>
            <dbtable_meta>DG_DOCU</dbtable_meta>
            <dbcolumn_meta>ID_CLAS_DOCU</dbcolumn_meta>
            <value />
            <value_code>OIA</value_code>
            <value_name>OIA</value_name>
          </meta>
          ...
        </metaValueList>
        <file>
          <fileContent>base64 obsah</fileContent>
          <fileName>1.pdf</fileName>
          <mimeType>application/pdf</mimeType>
          <extension>pdf</extension>
        </file>
      </return>
    </ns1:getDocumentResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

2.4 Služba ServiceDG, setDocument

Tato metoda slouží pro vytvoření dokumentu, pokud není `id_docu` vyplněno. Lze také použít pro jeho aktualizaci, pokud vyplněno je.

Vstup:

```
class SetDocumentReq {
    /* @var int */
    public $id_docu;
    /* @var int */
    public $id_fldr;
    /* @var string */
    public $folder;
    /* @var MetaValueList */
    public $metaValueList;
    /* @var DocumentFile[] */
    public $fileList;
    /* @var boolean */
    public $tryLockFile;
    /* @var string*/
    public $formFlowHashId;
    /** @var string Nepovinne. Od 10.2.3 */
    public $correlationId;
    /** @var boolean Nepovinne Od 10.2.2.0.4. Pokud true a jeste dokument existuje
, tak se ignoruje */
    public $createNewVersion;
    /** @var boolean */
    public $createUnknownEnumItems;
}
```

- **\$id_docu:** id dokumentu se zadává při aktualizaci metadata nebo binárního obsahu aktuální verze dokumentu. Při vytváření nového dokumentu musí být prázdné.
- **\$folder:** unikátní jméno složky (SID_FLDR), využívá se pouze v případě vytváření nového dokumentu. Přesun do složky není implementován.
- **\$id_fldr:** unikátní int id složky. Lze použít místo \$folder.
- **\$metaValueList:** seznam metadat pro vytvářený nebo aktualizovaný dokument.
- **\$fileList:** seznam binárních souborů. Nyní akceptujeme pouze 0 nebo 1 soubor. (hodnoty větší než 1 jsou připraveny pro budoucí použití, pokud bude třeba). Ve struktuře **DocumentFile** je nutné uvést název souboru s příponou nebo **IANA mime** typ.
- **\$tryLockFile:** Pokusí se zamknout dokument pro vlastníka dokumentu

- **\$formFlowHashId**: Identifikace instance FormFlow, příprava pro FormFlow Office Extension, nechat nevyplněno.
- **\$correlationId**: Zde možnos vyplnit svůj identifikátor pro seskupení idenfikace více volání.
- **\$createNewVersion**: Pokud true, vytvoří dokumentu novou verzi, do které vloží posílaná binár.
- **\$createUnknownEnumItems**: Od 10.2.4 založí nové položky obecného číselníku dle value_code a value_name, viz níže.

```
class MetaValueList
{
    /* @var MetaValue[] */
    public $meta;
}

class MetaValue extends MetaSpec
{
    /* @var string IN/OUT. */
    public $value;

    /** @var string IN/OUT. Je mozne pouzit v pripade vycitu pro klic polozky vycitu. */
    public $value_code;

    /** @var string OUT. Je navracena v pripade vycitu hodnota polozky vycitu. */
    public $value_name;
}
```

Poznámky:

- Pokud se požaduje zařazení dokumentu do LTD údržby, tak se nastaví metadato (jedna položka seznamu **\$metaValueList**) s name_meta = “LTD_ACCEPT_DOCU“ a value=“1“. Výchozí hodnota je 0 (nesmí být NULL).
- Od verze FF 10.2.2 je možné specifikovat hodnotu číselníku jejich řetězcovým kódem tak, že se do položky **\$metaValue->value_code** uvede tento klíč. Pokud je **\$value_code** uvedeno, má přednost před **\$value**. Viz popis třídy **MetaValue** v metodě **getDocument**.
- Od verze FF 10.2.4 je možné specifikovat value_name a při neznámé položce obecného číselníku a paramentru \$req->createUnknownEnumItems = 1 je tato číselníková položka vytvořena.

Systémové číselníky:

- Do value_code lze psát řetězcové kódy pro:

name_meta	popis
ID_CLAS_DOCU	Třída
ID_RANK_DOCU	Klasifikace
ID_STAT_DOCU	Stav
ID_USER_DOCU	Autor
OFFICER_ID_USER_DOCU	Referent
WARRANTOR_ID_GROU_DOCU	Garant
CURRENT_DVER_ID_MIME_DOCU	Mime typ dle IANA
ID_ARDE_DOCU	Archivoval (Modul DocsMaster, od 10.2.4)
ID_CERT_DOCU	Obsluhovaný subjekt (od 10.2.4)
ID_ARMP_DOCU	Způsob vyřízení

Modul archiv:

- Od 10.2.4. Pro vkládání do složek archivu je na těchto metadatech vyžadována správnost hodnoty, pokud je na tomto metadatu v dané složce nastaveno IS_REQUIRED_MIN_WS_MGUI na 1. A navíc pro některá metadata je vyžadováno vyplnění dalších metadata, která jsou prerekvizitami. To platí i pro metodu searchDocument.

Metadato	Prerekvizity
ID_CERT_DOCU	-
FILE_MARK_DOCU	ID_CERT_DOCU
EXTERNAL_SYSTEM_ID_DOCU	-
ID_ARDE_DOCU	EXTERNAL_SYSTEM_ID_DOCU
SHRED_MARK_DOCU	ID_CERT_DOCU, FILE_MARK_DOCU, SHRED_TIME_DOCU – pozn. Musí dát dohromady jak čtveřice platnou kombinaci.

- ID_ARDE_DOCU – Pokud je ve value_code posílána neznámá položka pro FF, je povinné specifikovat value_name a pak se takový vkladatel založí bez ohledu na hodnotu \$req->createUnknownEnumItems

Příklad:

```
<meta>

    <dbtable_meta>DG_DOCU</dbtable_meta>
    <dbcolumn_meta>ID_ARDE_DOCU</dbcolumn_meta>
    <value_code>ESS</value_code>
    <value_name>Jan Vkladatel</value_name>

</meta>
```

- ID_ARMP_DOCU – Tento číselník nemá value_name

Struktura DocumentFile:

```
class DocumentFile
{
    /* @var base64Binary */
    public $fileContent;
    /* @var string */
    public $fileUncPath;
    /* @var string */
    public $fileName;
    /* @var string */
    public $mimeType;
    /* @var string */
    public $id;
    /* @var string */
    public $checksumAlg;
    /* @var string */
    public $checksumValue;
    /* @var string */
    public $extension;
}
```

- **fileUncPath:** Náhrada base64 obsahu pro vstup (setDocument) a výstup (getDocument)
- Pokud je vyplněno **checksumAlg** a **checksumValue**, pak se při vkládání ověří heš vstupního bináru. List algoritmů dle php fce hash(.).
- **fileName:** jméno verze dokumentu.
- **mimeType:** IANA mime typ.
- **id:** nevyužito
- **extension:** přípona bináru, pouze pro výstup ve fci getDocument(.)

Výstup:

```
class SetDocumentResp {
    public $result; // int
    public $resultText; // string
    public $id_docu; // int
    public $version_dver; // int, 10.2.2.0.3
    public $id_dver; // int, 10.2.2.0.3
}
```

- **\$resultText**: int stavový kód WSSStatus.
- **\$resultTextText**: stavová zpráva, typicky chybový text.
- **\$id_docu**: ID dokumentu.
- **\$version_dver**: Od 10.2.2.0.3 doplňující informace o založené nebo aktualizované verzi – pořadové číslo verze.
- **\$id_dver**: Od 10.2.2.0.3 doplňující informace o založené nebo aktualizované verzi – unikátní číslo verze.

Příklad volání – požadavek:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://dg.software602.cz/">
  <SOAP-ENV:Body>
    <ns1:setDocument>
      <req>
        <id_fldr>3</id_fldr>
        <metaValueList>
          <meta>
            <dbtable_meta>DG_DOCU</dbtable_meta>
            <dbcolumn_meta>NAME_DOCU</dbcolumn_meta>
            <value>1_~_5c00f37d69e323.47435298.pdf</value>
          </meta>
          <meta>
            <dbtable_meta>DG_DOCU</dbtable_meta>
            <dbcolumn_meta>ID_CLAS_DOCU</dbcolumn_meta>
            <value>melaByBytPrepsana hodnotou value_code</value>
            <value_code>OIA</value_code>
          </meta>
          <meta>
            <dbtable_meta>DG_DOCU</dbtable_meta>
            <dbcolumn_meta>CJ_DOCU</dbcolumn_meta>
            <value />
            <value_code>Nejake CJ</value_code>
          </meta>
          <meta>
            <dbtable_meta>DG_DOCU</dbtable_meta>
            <dbcolumn_meta>CREATED_TS_DOCU</dbcolumn_meta>
            <value />
            <value_code>2018-11-30T09:28:25+01:00</value_code>
          </meta>
          <meta>
            <dbtable_meta>DG_DOCU</dbtable_meta>
            <dbcolumn_meta>LTD_ACCEPT_DOCU</dbcolumn_meta>
            <value>1</value>
          </meta>
          <meta>
            <dbtable_meta>DG_DOCU</dbtable_meta>
            <dbcolumn_meta>ID_ARDE_DOCU</dbcolumn_meta>
            <value_code>ESS</value_code>
            <value_name>Jan Vkladatel</value_name>
          </meta>
        </metaValueList>
        <fileList>
          <fileContent>base64 data</fileContent>
          <fileName>1.pdf</fileName>
        </fileList>
      </req>
    </ns1:setDocument>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Příklad volání – odpověď:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://dg.software602.cz/">
  <SOAP-ENV:Body>
    <ns1:setDocumentResponse>
      <return>
        <result>0</result>
        <resultText></resultText>
        <id_docu>175389</id_docu>
      </return>
    </ns1:setDocumentResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

2.5 Služba ServiceDG, searchDocument

Tato metoda slouží pro vyhledání dokumentu za podmínek stanovených v parametru **\$criteria**. Je nutno specifikovat metadata požadovaná ve výstupu.

Vstup:

```
class SearchDocumentReq
{ /* @var string */
  public $folder;
  /* @var MetaOperand[] */
  public $criteria;
  /* @var int */
  public $pageNo;
  /* @var int */
  public $pageSize;
  /* @var MetaSpec[] */
  public $metaList;
  /* @var MetaSpec */
  public $orderByMeta;
  /* @var boolean */
  public $orderAsc;
  /** @var boolean */
  public $return_value_code_name_only;
  /** @var boolean */
  public $return_xml_datetime;
  /** @var string Nepovinne. Od 10.2.3 */
  public $correlationId;
}
```

- **\$folder**: id (string id) složky.
- **\$criteria**: seznam podmínek vyhledání.
- **\$pageNo**: číslo stránky od 1, pokud není uvedeno, dosadí se hodnota 1.
- **\$pageSize**: velikost stránky, pokud není uvedeno, dosadí se hodnota 200.
- **\$metaList**: seznam sloupců ve výstupu.
- **\$orderByMeta**: sloupec, podle kterého třídit; pokud není uveden, netřídí se.
- **\$orderAsc**: pokud neuvedeno nebo je hodnota **true**, třídí se vzestupně; jinak sestupně.
- **\$return_value_code_name_only**: Pokud true, tak výstup MetaDataValue bude obsahovat prázdný atribut value pro úsporu přenesených dat. A data budou ve value_code a v případě číselníku ještě ve value_name; jinak se bude vracet hodnota jen ve value. Vychází hodnota je neuvedeno, tedy false.
- **\$return_xml_datetime**: Pokud true, metadata typu datetime jsou vracena ve xml formátu ISO8601.
- **\$correlationId**: Nepovinné. Pro seskupení identifikace několika volání do logů FormFlow.

Vstup:

```
class MetaOperand{
    public $type; // string
    public $criteria; // MetaOperand
    public $metaSpec; // MetaSpec
    public $operator; // string
    public $metaValue; // string
    public $metaCode; // string
}
```

- **\$type**: typ logického spojení: 'AND', 'OR'.
- **\$criteria**: podvýraz, v sql realizován (). V podvýrazu může být řetězec podpodmínek. Pokud je **\$criteria** neprázdné, další položky nesmí být uvedeny.
- **\$metaSpec**: označení sloupce, podle kterého se hledá.
- **\$operator**: =, >, >=, <, <=, <>, !=, startsWith, endsWith, isEmpty, isNotEmpty.
- **\$metaValue**: literál – hodnota metadata.
- **\$metaCode**: parameter byl zaveden od verze FF 10.2.2 pro možnost specifikace kódu číselníkové položky. Má přednost před **\$metaValue**. U nečíselníkových položek může být v **\$metaCode** uvedena běžná hodnota metadata.

Poznámky:

- **isEmpty** a **isNotEmpty** jsou unárními operátory od verze FF 10.2.2, tedy **metaValue** se ponechá **null**, a vyhledá metadata se sloupcem s hodnotou "" nebo **NULL**.
- Pro operátory **startsWith**, **endsWith** nelze použít znaky _ a % v **\$metaValue** nebo **\$metaCode** jako speciální znaky pro sql výraz **LIKE**. Tyto znaky jsou escapovány.
- Pokud se dotazujeme na specifická metadata ve složkách modulu archiv (od 10.2.4), je nutné v podmínce specifikovat všechna prerekvizitní metadata - viz metoda setDocument, tabulka v odstavci "Modul archiv".

Výstup:

```
class searchDocumentResponse{
    public $return; // SearchDocumentResp
}
```

- **\$return**: nalezený dokument

Výstup:

```
class SearchDocumentResp{
    public $document; // Document
    public $result; // int
    public $resultText; // string
}
```

- **\$document**: nalezená stránka dokumentů
- **\$resultText**: int stavový kód WSStatus.
- **\$resultTextText**: stavová zpráva, typicky chybový text.

Příklad volání – požadavek:

V níže uvedeném příkladu je realizovaná podmínka, která v sql vypadá následovně. Jde tedy o komplexnější příklad s užitím podvýrazu.

```
(DG_DOCU.NAME_DOCU LIKE :p1_NAME_DOCU ESCAPE '~')
AND (
    (DG_DOCU.ID_CLAS_DOCU = :p2_ID_CLAS_DOCU)
    OR (DG_DOCU.ID_CLAS_DOCU = :p3_ID_CLAS_DOCU)
)
AND (
    DG_DOCU.CJ_DOCU IS NOT NULL
    AND DG_DOCU.CJ_DOCU <> ''
)
```

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://dg.software602.cz/">
  <SOAP-ENV:Body>
    <ns1:searchDocument>
      <req>
        <folder>rsts1</folder>
        <criteria>
          <type>AND</type>
          <metaSpec>
            <name_meta>NAME_DOCU</name_meta>
          </metaSpec>
          <operator>endsWith</operator>
          <metaValue>_~_5c00f37d69e323.47435298.pdf</metaValue>
        </criteria>
        <criteria>
          <type>AND</type>
          <criteria>
            <type>OR</type>
            <metaSpec>
              <name_meta>ID_CLAS_DOCU</name_meta>
            </metaSpec>
            <operator>=</operator>
            <metaValue>Mela by byt prepsana elementem classCode</metaValue>
            <metaCode>OIA</metaCode>
          </criteria>
          <criteria>
            <type>OR</type>
```

```

        <metaSpec>
          <name_meta>ID_CLAS_DOCU</name_meta>
        </metaSpec>
        <operator>=</operator>
        <metaValue>Mela by byt prepsana elementem classCode</metaValue>
        <metaCode>OXA</metaCode>
      </criteria>
    </citeria>
    <criteria>
      <type>AND</type>
      <metaSpec>
        <name_meta>CJ_DOCU</name_meta>
      </metaSpec>
      <operator>isNotEmpty</operator>
    </criteria>
    <pageNo>1</pageNo>
    <pageSize>5</pageSize>
    <metaList>
      <name_meta>ID_DOCU</name_meta>
    </metaList>
    <metaList>
      <name_meta>NAME_DOCU</name_meta>
    </metaList>
    <metaList>
      <name_meta>ID_FLDR_DOCU</name_meta>
    </metaList>
    <metaList>
      <name_meta>ID_CLAS_DOCU</name_meta>
    </metaList>
    <metaList>
      <name_meta>ID_RANK_DOCU</name_meta>
    </metaList>
    <metaList>
      <name_meta>ID_STAT_DOCU</name_meta>
    </metaList>
    <metaList>
      <name_meta>CREATED_TS_DOCU</name_meta>
    </metaList>
    <orderByMeta>
      <name_meta>CREATED_TS_DOCU</name_meta>
    </orderByMeta>
    <orderAsc>false</orderAsc>
    <return_value_code_name_only>true</return_value_code_name_only>
    <return_xml_datetime>true</return_xml_datetime>
  </req>
</ns1:searchDocument>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Příklad volání – odpověď:

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns1="http://dg.software602.cz/">
  <SOAP-ENV:Body>
    <ns1:searchDocumentResponse>
      <return>
        <document>
          <metaList>
            <meta>

```

```

        <name_meta>ID_DOCU</name_meta>
        <value />
        <value_code>175389</value_code>
    </meta>
    <meta>
        <name_meta>NAME_DOCU</name_meta>
        <value />
        <value_code>1_~_5c00f37d69e323.47435298.pdf</value_code>
    </meta>
    <meta>
        <name_meta>ID_FLDR_DOCU</name_meta>
        <value />
        <value_code>rsts1</value_code>
        <value_name>Klientská</value_name>
    </meta>
    <meta>
        <name_meta>ID_CLAS_DOCU</name_meta>
        <value />
        <value_code>OIA</value_code>
        <value_name>OIA</value_name>
    </meta>
    <meta>
        <name_meta>ID_RANK_DOCU</name_meta>
        <value />
        <value_code>interni</value_code>
        <value_name>Interní</value_name>
    </meta>
    <meta>
        <name_meta>ID_STAT_DOCU</name_meta>
        <value />
        <value_code>N</value_code>
        <value_name>N</value_name>
    </meta>
    <meta>
        <name_meta>CREATED_TS_DOCU</name_meta>
        <value />
        <value_code>2018-11-30T09:28:25+01:00</value_code>
    </meta>
</metaList>
</document>
<result>0</result>
</return>
</ns1:searchDocumentResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

2.6 Služba ServiceDG, createDocumentVersion

Služba vytvářející novou verzi dokumentu.

Vstup:

```
class CreateDocumentVersionReq{  
    public $id_docu; //int  
}
```

- **\$id_docu:** int ID dokumentu.

Výstup:

```
class CreateDocumentVersionResp{  
    public $result;  
    public $resultText;  
}
```

Příklad volání – požadavek:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:ns1="http://dg.software602.cz/">  
  <SOAP-ENV:Body>  
    <ns1:createDocumentVersion>  
      <req>  
        <id_docu>175391</id_docu>  
      </req>  
    </ns1:createDocumentVersion>  
  </SOAP-ENV:Body>  
</SOAP-ENV:Envelope>
```

Příklad volání – odpověď:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:ns1="http://dg.software602.cz/">  
  <SOAP-ENV:Body>  
    <ns1:createDocumentVersionResponse>  
      <return>  
        <result>0</result>  
        <resultText>Created a new version with id_dver: 194577.</resultText>  
      </return>  
    </ns1:createDocumentVersionResponse>  
  </SOAP-ENV:Body></SOAP-ENV:Envelope>
```

2.7 Služba ServiceDG, lockDocumentByUser

Metoda pro zamčení uživatelským zámekem.

Vstup:

```
class LockDocumentByUserReq{  
    public $id_docu; // string  
    public $doLock; // boolean  
}
```

- **\$id_docu:** id dokumentu, který se bude zamykat/odemykat.
- **\$doLock:** pokud true, požadováno zamčení; jinak odemčení.

Výstup:

```
class LockDocumentByUserResp{  
    public $result; // int  
    public $resultText; // string  
}
```

V případě úspěchu je **\$resultText** = 0, jinak nenula společně s chybovou hláškou.

Příklad volání – požadavek:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:ns1="http://dg.software602.cz/">  
  <SOAP-ENV:Body>  
    <ns1:lockDocumentByUser>  
      <req>  
        <id_docu>175409</id_docu>  
        <doLock>true</doLock>  
      </req>  
    </ns1:lockDocumentByUser>  
  </SOAP-ENV:Body>  
</SOAP-ENV:Envelope>
```

Příklad volání – odpověď:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:ns1="http://dg.software602.cz/">  
  <SOAP-ENV:Body>  
    <ns1:lockDocumentByUserResponse>  
      <return>  
        <result>0</result>  
      </return>  
    </ns1:lockDocumentByUserResponse>  
  </SOAP-ENV:Body>  
</SOAP-ENV:Envelope>
```

2.8 Služba ServiceDG, createDocumentRelation

Metoda pro vytvoření dokumentové relace.

Vstup:

```
class CreateDocumentRelationReq
{
    /* @var DocumentRelation */
    public $documentRelation;
}

class DocumentRelation
{
    /* @var string */
    public $primary_id_docu;
    /* @var string */
    public $related_id_docu;
    /* @var string */
    public $relation_name;
}
```

Entita **DocumentRelation** obsahuje id primárního dokumentu, id vztaženého dokumentu a název relace.

Výstup:

```
class CreateDocumentRelationResp
{
    /* @var int */
    public $result;
    /* @var string */
    public $resultText;
}
```

Příklad volání – požadavek:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://dg.software602.cz/">
  <SOAP-ENV:Body>
    <ns1:createDocumentRelation>
      <req>
        <documentRelation>
          <primary_id_docu>175397</primary_id_docu>
          <related_id_docu>175395</related_id_docu>
          <relation_name>vychozi_vazba</relation_name>
        </documentRelation>
      </req>
    </ns1:createDocumentRelation>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```

    </ns1:createDocumentRelation>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Příklad volání – odpověď:

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://dg.software602.cz/">
  <SOAP-ENV:Body>
    <ns1:createDocumentRelationResponse>
      <return>
        <result>0</result>
      </return>
    </ns1:createDocumentRelationResponse>
  </SOAP-ENV:Body></SOAP-ENV:Envelope>

```

2.9 Služba ServiceDG, createDocumentRelationCollection

Metoda pro hromadné vytvoření primární dokumentové relace. Pokud je jakékoliv vytvoření relace odmítnuto z důvodu chyby, jsou všechny doposud vytvořené relace zrušeny a celý požadavek je nutno opakovat.

Vstup:

```

class CreateDocumentRelationCollectionReq
{
    /* @var string */
    public $primary_id_docu;
    /* @var string[] */
    public $related_id_docu;
    /* @var string */
    public $relation_name;
}

```

- **\$primary_id_docu** – int id docu, které bude primární v relaci.
- **\$related_id_docu** – int id docu, které bude vztažené v relaci.
- **\$relation_name** – kód typu vazby (CODE_TREL).

Výstup:

```

class CreateDocumentRelationCollectionResp
{
    /* @var int */
    public $result;
    /* @var string */
    public $resultText;
}

```

- **\$resultText** – int stav, 0 je OK.
- **\$resultTextTest** – string stavové hlášení.

Příklad volání – požadavek:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://dg.software602.cz/">
  <SOAP-ENV:Body>
    <ns1:createDocumentRelationCollection>
      <req>
        <primary_id_docu>175404</primary_id_docu>
        <related_id_docu>175405</related_id_docu>
        <related_id_docu>175406</related_id_docu>
        <relation_name>vychozi_vazba</relation_name>
      </req>
    </ns1:createDocumentRelationCollection>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Příklad volání – odpověď:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://dg.software602.cz/">
  <SOAP-ENV:Body>
    <ns1:createDocumentRelationCollectionResponse>
      <return>
        <result>0</result>
      </return>
    </ns1:createDocumentRelationCollectionResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

2.10 Služba ServiceDG, deleteDocumentRelation

Metoda pro smazání dokumentové relace.

Vstup:

```
class DeleteDocumentRelationReq
{
    /* @var DocumentRelation */
    public $documentRelation;
}
```

Výstup:

```
class DeleteDocumentRelationResp
{
    /* @var int */
    public $result;
    /* @var string */
    public $resultText;
}
```

- result 0 je stav OK.

Příklad volání – požadavek:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://dg.software602.cz/">
  <SOAP-ENV:Body>
    <ns1:deleteDocumentRelation>
      <req>
        <documentRelation>
          <primary_id_docu>175401</primary_id_docu>
          <related_id_docu>175402</related_id_docu>
          <relation_name>vychozi_vazba</relation_name>
        </documentRelation>
      </req>
    </ns1:deleteDocumentRelation>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Příklad volání – odpověď:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://dg.software602.cz/">
  <SOAP-ENV:Body>
    <ns1:deleteDocumentRelationResponse>
      <return>
        <result>0</result>
      </return>
    </ns1:deleteDocumentRelationResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

2.11 Služba ServiceDG, searchDocumentRelation

Metoda pro vyhledání dokumentových relací.

Vstup:

```
class SearchDocumentRelationReq
{
    /* @var string */
    public $id_docu;
    /* @var string */
    public $relation_name;
    // 1: id_docu is primary and returned is related
    // -1: id_docu related and returned is primary
    // 0: both -1 and 1
    /* @var int */
    public $direction;
}
```

- **id_docu**, jež má figurovat v relaci
- **relation_name** – kód typu relace (CODE_TREL). Může být prázdný, pak se vyhledávají relace bez ohledu na typ.
- **direction** – směr:
 - 1: id_docu má být primární v relaci,
 - 1 id_docu má být vztažené v relaci,
 - 0 / prázdné: vyhledání se provede bez ohledu na směr,

Výstup:

```
class SearchDocumentRelationResp
{
    /* @var int */
    public $result;
    /* @var string */
    public $resultText;
    /* @var DocumentRelation[] */
    public $documentRelation;
}
```

- **result 0** je stav OK
- je vráceno pole **DocumentRelation[]**.

Příklad volání – požadavek:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://dg.software602.cz/">
  <SOAP-ENV:Body>
    <ns1:searchDocumentRelation>
      <req>
        <id_docu>175395</id_docu>
      </req>
    </ns1:searchDocumentRelation>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Příklad volání – odpověď:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://dg.software602.cz/">
  <SOAP-ENV:Body>
    <ns1:searchDocumentRelationResponse>
      <return>
        <result>0</result>
        <documentRelation>
          <primary_id_docu>175395</primary_id_docu>
          <related_id_docu>175396</related_id_docu>
          <relation_name>vychozi_vazba</relation_name>
        </documentRelation>
        <documentRelation>
          <primary_id_docu>175397</primary_id_docu>
          <related_id_docu>175395</related_id_docu>
          <relation_name>vychozi_vazba</relation_name>
        </documentRelation>
      </return>
    </ns1:searchDocumentRelationResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

2.12 Služba ServiceDG, getDocumentVersions

Metoda pro vyhledání verzí dokumentu, od 10.2.2.0.3

Vstup:

```
class GetDocumentVersionsReq
{
    /* @var string */
    public $id_docu;
    /** @var boolean */
    public $fileRequired;
    /** @var int */
    public $versionNumber;
}
```

- **id_docu**, id dokumentu
- **fileRequired**, určení, zda se má ve výstup vrátit binár verze.
- **versionNumber** – nepovinné, omezení množiny vrácených verzí na jednu podle pořadového čísla verze

Výstup:

```
class GetDocumentVersionsResp
{
    /* @var int */
    public $result;
    /* @var string */
    public $resultText;
    /* @var DocumentVersion [] */
    public $documentVersions;
}

class DocumentVersion
{
    /** @var int */
    public $id_dver;
    /** @var int */
    public $id_docu_dver ;
    /** @var int */
    public $version_dver;
    /** @var int string iana mime typ*/
```

```

public $mime_dver;
/** @var string */
public $filename_dver;
/** @var string xs:datetime */
public $created_ts_dver;
/** @var string xs:datetime */
public $modified_ts_dver;
/** @var int */
public $created_by_id_user_dver;
/** @var int */
public $modified_by_id_user_dver;
/** @var string binary string*/
public $fileContent;
}

```

- **result 0** je stav OK
- je vráceno pole **DocumentVersion[]**.

Příklad volání – požadavek:

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://dg.software602.cz/">
  <SOAP-ENV:Body>
    <ns1:getDocumentVersions>
      <req>
        <id_docu>195525</id_docu>
        <fileRequired>true</fileRequired>
      </req>
    </ns1:getDocumentVersions>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Příklad volání – odpověď:

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://dg.software602.cz/">
  <SOAP-ENV:Body>
    <ns1:getDocumentVersionsResponse>
      <return>
        <result>0</result>
        <documentVersions>
          <id_dver>214738</id_dver>
          <id_docu_dver>195525</id_docu_dver>
          <version_dver>1</version_dver>
          <mime_dver>text/plain</mime_dver>
          <filename_dver>CreatellVersionsAndListThem.2019-08-
13T13:24:42+02:00.version1.txt</filename_dver>
          <created_ts_dver>2019-08-13T13:24:46+02:00</created_ts_dver>
          <modified_ts_dver>2019-08-13T13:24:46+02:00</modified_ts_dver>
          <created_by_id_user_dver>1</created_by_id_user_dver>
          <modified_by_id_user_dver>1</modified_by_id_user_dver>

```

```

<fileContent>Q3JlYXRlMTFWZXJzaW9uc0FuZExpc3RUaGVtLjIwMTktMDgtMTNUMTM6MjQ6NDIrMDI6MDAuc21vbjE=
</fileContent>
</documentVersions>
<documentVersions>
  <id_dver>214739</id_dver>
  <id_docu_dver>195525</id_docu_dver>
  <version_dver>2</version_dver>
  <mime_dver>text/plain</mime_dver>
  <filename_dver>Create11VersionsAndListThem.2019-08-13T13:24:42+02:00.version2.txt</filename_dver>
  <created_ts_dver>2019-08-13T13:24:48+02:00</created_ts_dver>
  <modified_ts_dver>2019-08-13T13:24:49+02:00</modified_ts_dver>
  <created_by_id_user_dver>1</created_by_id_user_dver>
  <modified_by_id_user_dver>1</modified_by_id_user_dver>
</documentVersions>
</return>
</ns1:getDocumentVersionsResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

2.13 DzhpService, insertDocument

Metoda pro vložení dms dokumentu do spisu

Vstup:

```

class InsertDocumentReq
{
  /* @var int[] */
  public $id_docu_array;
  /* @var int */
  public $id_ufrm_spis;
  /* @var string */
  public $spisova_znacka;
  /* @var string */
  public $source;
}

```

- **id_docu_array**, pole id dokumentů
- **id_ufrm_spis**, id spisu
- **spisova_znacka** – spisová značka
- **source**, id zdrojového systému

Výstup:

```
class insertDocumentResponse
{
    /* @var InsertDocumentResp */
    public $return;
}

class InsertDocumentResp
{
    /* @var int */
    public $result;
    /* @var string */
    public $resultText;
    /* @var InsertDocumentInfoItem[] */
    public $info_array;
}

class InsertDocumentInfoItem
{
    /* @var int */
    public $id_docu;
    /* @var boolean */
    public $result;
    /* @var boolean */
    public $really_inserted;
    /* @var string */
    public $msg;
}
```

- **result 0** je stav OK
- je vráceno pole **InsertDocumentInfoItem []**.

3 Elektronická spisová služba (ESS)

Webové služby vystavené *Elektronickou spisovou službou* umožňují v omezené míře využívat funkce ESS bez nutnosti přímé obsluhy ESS prostřednictvím *FormFlow*.

Webová služba spisové služby má tři koncové body, které jsou si rovny:

- /htdocs/dzhp/ws.php
- /htdocs/soap/ServiceESSEndpoint.php
- /htdocs/soap2/ServiceESSEndpoint.php

3.1 Autentizace webových služeb ESS

Uživatel, pod kterým se běh služby vykonává, může být v případě tzv. otevřeného systému specifikován v profilu (viz dále). Při volání WS se jako součást volání předávají povinné elementy **UserName** a **Password**, v případě instalace FFS s podporou domén také nepovinný element **Domain** (viz wsdl, typ tns:AuthHeader). Pro většinu instalací se ale využívá basic autentizace, tedy od verze 10.2.4.0.5 již není třeba ve volání uvádět UserName a Password.

Vstup:

```
class AuthHeader{  
    public $UserName; // string  
    public $Password; // string  
    public $Domain; // string  
}
```

- **\$UserName**: uživatelské jméno.
- **\$Password**: uživatelské heslo.
- **\$Domain**: doména.

Při volání WS FFS ověřuje, zda daný uživatel existuje, může se přihlásit do FFS a má právo používat formulář uvedený v profilu. Pokud jsou tyto podmínky splněny, WS pokračuje v běhu. Jinak vrací chybu (viz návratové stavy).

3.2 Instalace

WS jsou součástí standardní distribuce ESS a není nutné je zvlášť instalovat. Pro jejich používání je však nutné je nakonfigurovat. Nastavení WS se provádí až po konfiguraci FFS a po nastavení celé ESS.

Konfigurace WS spočívá ve vytvoření komunikačních profilů. Ty se zapisují do souboru `/dzhp/ws_profile.php`, jehož strukturu můžete převzít z distribučního (vzorového) souboru `/dzhp/ws_profile.php.distrib`.

Jednotlivé profily se vytváří jako položky konfiguračního pole, číslo profilu (klíč v poli `$config_ws_profile`) je uváděno ve volání služby. Pro jednotlivé operace WS můžete připravit jeden nebo i více profilů. Díky tomu lze provést drobné změny některých konfiguračních parametrů WS bez nutnosti měnit volající systémy. Ty také nemusí vždy přenášet veškeré konfigurační informace, ale místo toho stačí, když uvedou číslo profilu.

3.3 Profily

Každé volání WS obsahuje jako povinný parametr číslo používaného profilu (`$config_ws_profile[1]`). Profil je pevná konfigurace na straně aplikačního serveru. Ta obsahuje zejména:

- `id_serv_subj` – ID obsluhovaného subjektu, tj. id subjektu, který bude vlastníkem všech vytvořených formulářů a do jehož spisové služby se formuláře zařazují (`'id_serv_subj'=>4,`).
- `MSG_IN_id_fmve` – ID formuláře pro vytvoření MSG_IN (`'MSG_IN_id_fmve'=>7,`).
- `MSG_OUT_id_fmve` – ID formuláře pro vytvoření MSG_OUT (`'MSG_OUT_id_fmve'=>8,`).
- `SPIS_id_fmve` – ID formuláře pro vytvoření SPIS (`'SPIS_id_fmve'=>4,`).
- `SPIS_cj_id` – číselná řada pro vytvoření spisové značky spisu (`'SPIS_cj_id'=>4,`).
- `MSG_IN_can_use_default' => 0` – zakáže používat defaultní hodnoty při vytváření MSG_IN.
- `MSG_OUT_can_use_default' => 0` – zakáže používat defaultní hodnoty při vytváření MSG_OUT.
- `SPIS_can_use_default' => 0` – zakáže používat defaultní hodnoty při vytváření SPIS.
- `login_user` – volitelný parametr. Při jeho zadání se nehledí na autentizaci a celý běh služby se vykonává pod daným uživatelem. Doporučuje se využívat autentizace a tento parametr nevyplňovat. V případě FFS 6.0 s podporou domén a vyšších verzí se vyplňuje hodnota z DB sloupceku `login_user ('login_user'=>user,`).

Příklad:

```
$config_ws_profile[1] = array('login_user'=>user, 'id_serv_subj'=>4, 'MSG_IN_id_fmve'=>7, 'MSG_OUT_id_fmve'=>8, 'SPIS_id_fmve'=>4, 'SPIS_cj_id'=>4);
```

3.4 Obecné datové struktury

Jelikož operace prováděné WS nejsou triviální a pro úspěšné vykonání vyžadují velké množství informací, jsou tyto strukturovány do logických celků. Takto vzniká sada obecných datových struktur, které jsou opakovaně používány při různých voláních.

3.4.1 Struktura klíč – hodnota: KeyValuePair

Pro obecné datové struktury využívané zejména v pluginech (viz plugin) se využívá triviální datová struktura **Klíč-hodnota**. Obsahuje vždy textovou reprezentaci klíče a textovou reprezentaci přiřazené hodnoty.

Vstup:

```
class KeyValuePair{
    public $Key; // string
    public $Value; // string
}
```

- **\$Key**: Klíč, ke kterému se váže hodnota.
- **\$Value**: Přiřazená hodnota.

3.4.2 Struktura dat pro plugin: PluginDataStruct

Každé volání WS obsahuje jeden element typu **PluginDataStruct**, do kterého je možné vložit 0 až N elementů typu **KeyValuePair** a tak předat případnému pluginu další data nad rámec běžného volání ws.

Vstup:

```
class PluginDataStruct{
    public $Item; // KeyValuePair
}
```

- **\$Item**: 0 až N struktur typu klíč-hodnota.

Příklad plnění PluginData pro WS ESSCreateMsgOut:

```
// PluginData
$out_param[0]['PluginData'][0]['Key']='IdMsg';
    $out_param[0]['PluginData'][0]['Value']='4983568';
    $out_param[0]['PluginData'][1]['Key']='KLIENT_1';
    $out_param[0]['PluginData'][1]['Value']='1111178992;Smlouva 111';
    $out_param[0]['PluginData'][2]['Key']='KLIENT_2';
    $out_param[0]['PluginData'][2]['Value']='2222278992;Smlouva 222';
    $out_param[0]['PluginData'][3]['Key']='KLIENT_3';
    $out_param[0]['PluginData'][3]['Value']='3333378992;Smlouva 333';
```

Pokud je klientů více, musí **Key** začínat stringem **KLIENT** a pak už může být cokoli, aby klíče byly unikátní. **VALUE** u jednoho klienta má strukturu: **RC ; CONTRAT_NO**

KEY IdMsg je tam jen jednou a ve **VALUE** je **ID_DZ**, na kterou se odpovídá.

Aby došlo k automatickému odeslání MSG_OUT do výstupní fronty, musí být nastaven element **ws SendType = 1**;

3.4.3 Struktura spisu: FileStruct

Struktura popisuje metadata jednoho spisu, tj. spisová a skartační data a informace o subjektech.

Vstup:

```
class FileStruct
{
    public $id_spis; // int
    public $spis_name; // string
    public $spis_znak; // string
    public $skart_znak; // string
    public $skart_lhuta; // int
    public $spisova_znacka; // string
    public $id_firma; // int
    public $adr_name; // string
    public $adr_address; // string
}
```

- **\$id_spis**: ID spisu v interní databázi (ID_UFRM).
- **\$spis_name**: Název spisu.
- **\$spis_znak**: Spisový znak.
- **\$skart_znak**: Skartační znak.
- **\$skart_lhuta**: Skartační lhůta.
- **\$spisova_znacka**: Spisová značka.
- **\$id_firma**: ID subjektu v interní databázi.
- **\$adr_name**: Název subjektu složený.
- **\$adr_address**: Adresa subjektu složená.

3.4.4 Struktura spisů: Files

Jednoduchá struktura popisující 0 až N spisů. Obsahuje položky typu **FileStruct**.

Vstup:

```
class Files{
    public $File; // FileStruct
}
```

- **\$File**: 0 až N struktur popisujících spisy.

3.4.5 Seznam struktur položek: ItemListStruct

Struktura popisuje obecnou strukturu položky ESS, tj. dokument nebo spis.

Vstup:

```
class ItemListStruct{
    public $ListItem; // ItemStruct, povinný
}
```

- **\$ListItem**: 0 až N struktur typu klíč-hodnota.

3.4.6 Struktura položky: ItemStruct

Struktura popisuje obecnou položku ESS, tj. dokument nebo spis.

Vstup:

```
class ItemStruct{
    public $type; // int
    public $name; // string
    public $id_doc; // int
    public $cj; // string
    public $jid; // string
    public $autor_id; // int
    public $autor_name; // string
    public $date_created; // dateTime
    public $date_filled; // dateTime
    public $spis_id; // int
    public $spis_name; // int
    public $spis_jid; // string
    public $archived; // int
}
```

- **\$type:** 0 = spis, 1 = dokument.
- **\$name:** název spisu / anotace dokumentu.
- **\$id_doc:** vnitřní identifikátor záznamu (ID_UFRM).
- **\$cj:** přidělené číslo jednací.
- **\$jid:** přidělený JID.
- **\$autor_id:** ID uživatele – autora formuláře.
- **\$autor_name:** čitelné jméno autora (Jméno Příjmení).
- **\$date_created:** Datum a čas založení spisu/dokumentu.
- **\$date_filled:** datum a čas poslední změny.
- **\$spis_id:** interní ID (ID_UFRM) nadřazeného spisu (je-li spis nebo dokument zařazen ve spisu).
- **\$spis_name:** název nadřazeného spisu (je-li spis nebo dokument zařazen ve spisu).
- **\$spis_jid:** JID nadřazeného spisu (je-li spis nebo dokument zařazen ve spisu).
- **\$archived:** 1 = spis / dokument je archivován, 0 = spis / dokument není archivován.

3.4.7 Struktura položky: CjStruct

Struktura čísla jednacího.

Vstup:

```
class CjStruct{
    public $cj_name; // string
    public $cj_is_used; // int
    public $cj_num; // int
    public $cj_num_digit; // int
    public $cj_rein_month; // int
    public $cj_num_after_rein; // int
    public $cj_current_year; // int
    public $cj_current_month; // int
    public $cj_num_digit_answer; // int
    public $cj_mask; // string
    public $cj_mask_answer; // string
    public $cj_doc_mask; // string
    public $cj_doc_num_digit; // int
    public $cj_typ; // int
    public $cj_for_dok; // int
    public $cj_for_spis; // int
    public $cj_for_tlog; // int
    public $cj_in; // string
    public $cj_out; // string
}
```

3.4.8 Struktura adresy: AddressStruct

Struktura popisuje adresu použitelnou ve webové službě FSS převážně pro identifikaci původce příchozí zprávy nebo příjemce odchozí zprávy. Povinnost jednotlivých elementů je závislá na způsobu přijetí/odeslání dané zprávy, pokud je element povinný, je u něj uvedeno 1, 2 nebo 3 (nebo jejich kombinace) ve významu povinný pro papírové odeslání (1), elektronické odeslání emailem (2) nebo odeslání pomocí systému ISDS (3).

Vstup:

```
class AddressStruct{
    public $Type; // int
    public $TypeAdr; // int
    public $IdFirma; // int
    public $IdDS; // string
    public $Email; // string
    public $Salutation; // string
    public $Name; // string
    public $SubName; // string
    public $Representative; // string
    public $Street; // string
    public $RedNumber; // string
    public $BlueNumber; // string
    public $ZipCode; // string
    public $City; // string
    public $State; // string
    public $OrgUnitName; // string
    public $OrgUnitNumber; // int
    public $DeliveryMethod; // string
}
```

- **\$Type**: 1=DZ, 2=klasická pošta, 3=e-mail.
- **\$TypeAdr**: 1=zástupce před firmou, 2=zástupce za firmou.
- **\$IdFirma**: Má přednost před ostatním, aktualizace adresáře se neprovádí.
- **\$IdDS**: ID datové schránky.
- **\$Email**: e-mailové spojení příjemce.
- **\$Salutation**: oslovení.
- **\$Name**: název příjemce.
- **\$SubName**: rozšiřující název příjemce.
- **\$Representative**: jméno zástupce společnosti.
- **\$Street**: ulice (bez čísel popisných a orientačních).
- **\$RedNumber**: číslo popisné.
- **\$BlueNumber**: číslo orientační.
- **\$ZipCode**: PSČ.
- **\$City**: název obce.
- **\$State**: kód státu.
- **\$OrgUnitName**: jméno organizační jednotky.
- **\$OrgUnitNumber**: číslo organizační jednotky.
- **\$DeliveryMethod**: ID nebo název způsobu doručení.

3.4.9 Struktura jedné přílohy: DocumentStruct

Struktura popisující dokument. Dokumentem může být jak odkaz do SFLS, tak přímo vložená data ve formátu base64.

Vstup:

```
class DocumentStruct{
    public $Description; // string
    public $Data; // base64Binary
    public $Filename; // string
    public $MimeType; // string
    public $InternalId; // string
}
```

- **\$Description:** popis dokumentu.
 - **\$Data:** buď APX link, nebo data. Vždy v base64 formátu.
 - **\$Filename:** jméno souboru.
 - **\$MimeType:** Mime type dle zvyklostí FFS, tedy například „pdf“.
- \$InternalId:** interní odkaz na dokument (SFLF) ve WS spisové služby, tento element může obsahovat XG_SFLF.ID_SFLF nebo XG_SFLF.ID. Pokud je tento element vyplněn, má přednost před elementem "Data" ve stejné struktuře. Ve WS ESS byla implementována podpora ASiC formulářů MSG_IN, MSG_OUT a SPIS.

3.4.10 Struktura více příloh: Document

Struktura popisující dokument. Dokumentem může být jak odkaz do SFLS, tak přímo vložená data ve formátu base64.

Vstup:

```
class Document{
    public $DocumentList; // DocumentList, povinný
    public $NumPhysicalSheets; // int
    public $NumAttachmentSheets; // int
    public $Category; // int
}
```

- **\$DocumentList:** Přílohy dokumentů (komponenty).
- **\$NumPhysicalSheets:** Počet fyzických listů dokumentu.
- **\$NumAttachmentSheets:** Počet fyzických listů příloh.
- **\$Category:** Číslo kategorie dokumentu.

3.5 Seznam položek čísla jednacího: CjList

Vstup:

```
class CjList{
    public $OneCj; // CjStruct
}
```

- **\$OneCj**: Jedno číslo jednací.

3.6 Seznam příloh: DocumentList

Struktura popisující dokument. Dokumentem může být jak odkaz do SFLS, tak přímo vložená data ve formátu base64.

Vstup:

```
class DocumentList{
    public $Document; // DocumentStruct, povinný
}
```

- **\$Document**: Seznam příloh.

3.7 Vložení dokumentu do spisu – ESSAddDocumentToSpis

Tato metoda slouží k vložení dokumentu do spisu.

Vstup:

```
class ESSAddDocumentToSpisRequest{
    public $profileID; // int
    public $IdSubject; // int
    public $IdUserTo; // int
    public $UserEmail; // string
    public $FileId; // string
    public $IdUfrmSpis; // int
    public $Jid; // string
    public $Cj; // string
    public $IdDoc; // string
}
```

- **\$profileID**: identifikuje profil, který tato WS používá (ws_profile.php).
- **\$IdSubject**: ID obsluhovaného subjektu zadávaný přes WS request, jinak se získá z ws_profile (id_serv_subj).
- **\$IdUserTo**: ID uživatele, který vkládá dokument do spisu.
- **\$UserEmail**: e-mail uživatele, který vkládá dokument do spisu.
- **\$FileId**: spisová značka spisu, do kterého má být dokument vložen.
- **\$IdUfrmSpis**: id_ufrm spisu, do kterého má být dokument vložen.
- **\$Jid**: přidělený JID.
- **\$Cj**: č.j. dokumentu.
- **\$IdDoc**: ID_UFRM dokumentu, který bude vkládán.

Výstup:

```
class ESSAddDocumentToSpisResponse{
    public $ResultCode; // int
    public $ErrMsg; // string
    public $IdDoc; // int
    public $IdUfrmSpis; // int
}
```

- **\$resultTextCode**: číslo návratového stavu, viz číselník návratových stavů.
- **\$ErrMsg**: popis v případě chyby.
- **\$IdDoc**: ID_UFRM dokumentu, který byl vložen do spisu.
- **\$IdUfrmSpis**: ID_UFRM spisu, do kterého byl dokument vložen.

3.7.1 Příchozí dokument ESSCreateMsgIn

Struktura popisuje požadavek na vytvoření příchozího dokumentu.

Vstup:

```
class ESSCreateMsgInRequest {
    public $profileID; // int
    public $IdSubject; // int
    public $IdSubjekt; // int
    public $IdFmve; // int
    public $IdUserTo; // int
    public $UserEmail; // string
    public $DocumentName; // string
    public $MailBody; // string
    public $SendType; // int
    public $FinishingMethod; // string
    public $ClassificationCode; // string
    public $ShreddingMark; // string
    public $ShreddingPeriod; // int
}
```

```

public $ShreddingMode; // string
public $PhysicalArchive; // string
public $ExpeditionNote; // string
public $SenderRefNumber; // string
public $SenderId; // string
public $RecipientRefNumber; // string
public $RecipientIdent; // string
public $SenderCj; // string
public $SenderSpisZnak; // string
public $RecipientCj; // string
public $RecipientSpisZnak; // string
public $IdCategory; // int
public $Category; // string
public $companyType; // string
public $DeliveryDate; // string
public $DeliveryTime; // string
public $DeliveryMethod; // string
public $SerialNumber; // string
public $PostOfOrigin; // string
public $NumPhysicalSheets; // integer
public $NumAttachementSheets; // integer
public $IdFirma; // integer
public $FileId; // string
public $Sender; // AddressStruct
public $DocumentList; // DocumentList
public $Documents; // Document
public $PluginData; // PluginDataStruct
}

```

- **\$profileID**: ID obsluhovaného subjektu zadávaný přes WS request, jinak se získá z ws_profile.php.
- **\$IdSubject**: starý element – ID obsluhovaného subjektu zadávaný přes WS request, jinak se získá z ws_profile (id_serv_subj).
- **\$IdSubjekt**: starý element – ID obsluhovaného subjektu zadávaný přes WS request, jinak se získá z ws_profile (id_serv_subj).
- **\$IdFmve**: ID_FMVE šablony formuláře MSG_IN, jinak se získá z ws_profile (MSG_IN_id_fmve).
- **\$idUserTo**: ID uživatele, který je autorem formuláře.
- **\$UserEmail**: E-mail uživatele, který je autorem formuláře.
- **\$DocumentName**: název dokumentu MsgIn.

- **\$MailBod**: obsah mailové zprávy.
= 1: MSG_IN jde do archivu, pakliže jsou splněny všechny náležitosti ESS. Podle Sender.Type se určí typ příchozí zásilky. WS podporuje hodnoty 2 = klasická pošta a 3 = e-mailová zásilka.
- **\$FinishingMethod**: způsob vyřízení ID nebo název.
- **\$ClassificationCode**: spisový znak.
- **\$ShreddingMark**: skartační znak.
- **\$ShreddingPeriod**: skartační lhůta.
- **\$ShreddingMode**: název skartačního režimu.
- **\$PhysicalArchive**: fyzické uložení dokumentu.
- **\$ExpeditionNote**: poznámka.
- **\$SenderRefNumber**: číslo jednacích odesílatele SenderRefNumber.
- **\$SenderIdent**: spisová značka odesílatele SenderIdent.
- **\$RecipientRefNumber**: číslo jednacích příjemce RecipientRefNumber.
- **\$RecipientIdent**: spisová značka příjemce RecipientIdent.
- **\$SenderCj**: starý element – číslo jednacích odesílatele SenderRefNumber.
- **\$SenderSpisZnak**: starý element – spisová značka odesílatele SenderIdent.
- **\$RecipientCj**: starý element – číslo jednacích příjemce RecipientRefNumber.
- **\$RecipientSpisZnak**: starý element – spisová značka příjemce RecipientIdent.
- **\$IdCategory**: ID kategorie DZHP_MSG_TYP.ID_TYP.
- **\$Category**: název kategorie DZHP_MSG_TYP.MSG_TYP.
- **\$companyType**: název kategorie DZHP_MSG_TYP.MSG_TYP.
- **\$DeliveryDate**: datum doručení.
- **\$DeliveryTime**: čas doručení.
- **\$DeliveryMethod**: ID nebo název způsobu doručení.
- **\$SerialNumber**: podací číslo.
- **\$PostOfOrigin**: podací pošta.
- **\$NumPhysicalSheets**: Počet fyzických listů dokumentu.
- **\$NumAttachementSheets**: Počet fyzických listů příloh.
- **\$IdFirma**: ID odesílatele = DZHP_FIRMA.ID_FIRMA má přednost před strukturou Sender, nelze tedy editovat položky v adresáři.
- **\$FileId**: spisová značka spisu, do kterého má být dokument vložen.
- **\$Sender**: adresa odesílatele.
- **\$DocumentList**: přílohy (komponenty) dokumentu.
- **\$Documents**: původní množina příloh – už nepoužívat.
- **\$PostOfOrigin**: podací pošta.
- **\$PluginData**: PluginDataStruct.

Výstup:

```
class ESSCreateMsgInResponse{
    public $ResultCode; // int
    public $ErrMsg; // string
    public $Jid; // string
    public $Cj; // string
    public $RefNumber; // string
    public $DocIdentifier; // string
    public $IdUfrm; // int
    public $IdUfrmSpis; // int
    public $InfoMsg; // string
    public $Temp; // string
    public $PluginData; // PluginDataStruct
}
```

- **\$resultTextCode**: Číslo návratového stavu, viz číselník návratových stavů.
- **\$ErrMsg**: Popis v případě chyby.
- **\$Jid**: Přidělený JID.
- **\$Cj**: Přidělené číslo jednací.
- **\$RefNumber**: Přidělené číslo jednací.
- **\$DocIdentifier**: ID_UFRM vytvořeného MSG_IN.
- **\$IdUfrm**: ID_UFRM vytvořeného MSG_IN.
- **\$IdUfrmSpis**: ID_UFRM spisu, do kterého byl nově vytvořený dokument vložen.
- **\$InfoMsg**: info, pokud se nepodařilo vložit dokument do spisu.
- **\$Temp**: pro zobrazení outWsParam.
- **\$PluginData**: PluginDataStruct.

3.7.2 Odchozí dokument – ESSCreateMsgOut

Struktura popisuje požadavek na vytvoření odchozího dokumentu.

Vstup:

```
class ESSCreateMsgOutRequest
{
    public $profileID; // int
    public $IdSubject; // int
    public $IdSubjekt; // int
    public $IdFmve; // int
    public $IdUserTo; // int
    public $UserEmail; // string
    public $DocumentName; // string
}
```

```

public $MailBody; // string
public $Type; // int
public $SendType; // int
public $FinishingMethod; // string
public $ClassificationCode; // string
public $ShreddingMark; // string
public $ShreddingPeriod; // int
public $ShreddingMode; // string
public $PhysicalArchive; // string
public $ExpeditionNote; // string
public $SenderRefNumber; // string
public $SenderId; // string
public $RecipientRefNumber; // string
public $RecipientIdent; // string
public $SenderCj; // string
public $SenderSpisZnak; // string
public $RecipientCj; // string
public $RecipientSpisZnak; // string
public $IdCategory; // int
public $Category; // string
public $companyType; // string
public $NumPhysicalSheets; // integer
public $NumAttachementSheets; // integer
public $FileId; // string
public $Recipient; // AddressStruct
public $Recipients; // AddressList
public $DocumentList; // DocumentList
public $Documents; // Document
public $PluginData; // PluginDataStruct}

```

- **\$profileID**: Identifikuje profil, který tato WS používá.
- **\$IdSubject**: ID obsluhovaného subjektu zadávaný přes WS request, jinak se získá z ws_profile (id_serv_subj).
- **\$IdSubjekt**: starý element – ID obsluhovaného subjektu zadávaný přes WS request, jinak se získá z ws_profile (id_serv_subj).
- **\$IdFmve**: ID_FMVE šablony formuláře MSG_IN, jinak se získá z ws_profile (MSG_IN_id_fmve).
- **\$idUserTo**: ID uživatele, který je autorem formuláře.
- **\$UserEmail**: E-mail uživatele, který je autorem formuláře.
- **\$DocumentName**: název dokumentu MsgOut (annotation).

- **\$MailBody**: Obsah mailové zprávy.
- **\$Type**: element v `AddressStruct` `DZHP_DENIK_ADR.TYP_MSG` 1=DZ, 2=klasik, 3=mail.
- **\$SendType** = 1: MSG_OUT jde do výstupní fronty k odeslání, pakliže jsou splněny všechny náležitosti ESS.
- **\$ClassificationCode**: spisový znak.
- **\$ShreddingMark**: skartační znak.
- **\$ShreddingPeriod**: skartační lhůta.
- **\$ShreddingMode**: název skartačního režimu.
- **\$PhysicalArchive**: fyzické uložení dokumentu.
- **\$SenderRefNumber**: číslo jednacích odesílatele `SenderRefNumber`.
- **\$SenderId**: spisová značka odesílatele `SenderId`.
- **\$RecipientRefNumber**: číslo jednacích příjemce `RecipientRefNumber`.
- **\$RecipientId**: spisová značka příjemce `RecipientId`.
- **\$SenderCj**: starý element – číslo jednacích odesílatele `SenderRefNumber`.
- **\$SenderSpisZnak**: starý element – spisová značka odesílatele `SenderId`.
- **\$RecipientCj**: starý element – číslo jednacích příjemce `RecipientRefNumber`.
- **\$RecipientSpisZnak**: starý element – spisová značka příjemce `RecipientId`.
- **\$IdCategory**: ID kategorie `DZHP_MSG_TYP.ID_TYP`.
- **\$Category**: název kategorie `DZHP_MSG_TYP.MSG_TYP`.
- **\$companyType**: název kategorie `DZHP_MSG_TYP.MSG_TYP`.
- **\$NumPhysicalSheets**: Počet fyzických listů dokumentu.
- **\$NumAttachementSheets**: Počet fyzických listů příloh.
- **\$FileId**: spisová značka spisu, do kterého chceme vložit dokument MSG_OUT.
- **\$Recipient**: jeden příjemce.
- **\$Recipients**: vícero příjemců.
- **\$DocumentList**: přílohy (komponenty) dokumentu.
- **\$Documents**: původní množina příloh – již nepoužívat!
- **\$PluginData**: `PluginDataStruct`.

Výstup:

```
class ESSCreateMsgOutResponse{
    public $ResultCode; // int
    public $ErrMsg; // string
    public $Jid; // string
    public $Cj; // string
    public $RefNumber; // string
    public $DocIdentifier; // string
    public $IdUfrm; // int
    public $IdUfrmSpis; // int
    public $InfoMsg; // string
    public $Temp; // string
    public $PluginData; // PluginDataStruct
}
```

- **\$resultTextCode**: číslo návratového stavu, viz číselník návratových stavů.
- **\$ErrMsg**: popis v případě chyby.
- **\$Jid**: přidělený JID.
- **\$Cj**: přidělené číslo jednací.
- **\$RefNumber**: přidělené číslo jednací.
- **\$DocIdentifier**: ID_UFRM vytvořeného MSG_IN.
- **\$IdUfrm**: ID_UFRM vytvořeného MSG_IN.
- **\$IdUfrmSpis**: ID_UFRM spisu, do kterého byl nově vytvořený dokument vložen.
- **\$InfoMsg**: info, pokud se nepodařilo vložit dokument do spisu.
- **\$Temp**: pro zobrazení outWsParam.
- **\$PluginData**: PluginDataStruct.

3.7.3 Vytvořit spis – ESSCreateSpis

Vstup:

```
class ESSCreateSpisRequest{
    public $profileID; // int
    public $IdSubject; // int
    public $FileType; // int
    public $IdUserTo; // string
    public $UserEmail; // string
    public $FileName; // string
    public $FileDescription; // string
    public $CjId; // int
    public $CjName; // string
    public $FinishingMethod; // string
    public $ClassificationCode; // string
    public $ShreddingMark; // string
    public $ShreddingPeriod; // int
    public $IdFirma; // int
    public $Subject; // AddressStruct
    public $DocumentList; // DocumentList
    public $PluginData; // PluginDataStruct
}
```

- **\$profileID**: identifikuje profil, který tato WS používá.
- **\$IdSubject**: ID obsluhovaného subjektu zadávaný přes WS request, jinak se získá z ws_profile (id_serv_subj).
- **\$FileType**: typ spisu, zatím se nepoužívá.
- **\$IdUserTo**: ID uživatele, který dostane spis na stůl.
- **\$UserEmail**: e-mail uživatele, který dostane spis na stůl.

- **\$FileName**: název spisu.
- **\$FileDescription**: popis spisu.
- **\$CjId**: CJ_ID číselné řady pro spisovou značku.
- **\$CjName**: název číselné řady pro spisovou značku.
- **\$FinishingMethod**: způsob vyřízení ID nebo název.
- **\$ClassificationCode**: spisový znak.
- **\$ShreddingMark**: skartační znak.
- **\$ShreddingPeriod**: skartační lhůta.
- **\$IdFirma**: DZHP_FIRMA.ID_FIRMA dotčeného subjektu. Má přednost před adresou dotčeného subjektu.
- **\$Subject**: adresa dotčeného subjektu.
- **\$DocumentList**: přílohy (komponenty) dokumentu.
- **\$PluginData**: PluginDataStruct.

Výstup:

```
class ESSCreateSpisResponse{
    public $ResultCode; // int
    public $ErrMsg; // string
    public $RefNumber; // string
    public $DocIdentifier; // string
    public $IdUfrm; // int
    public $Temp; // string
    public $PluginData; // PluginDataStruct
}
```

- **\$ResultTextCode**: číslo návratového stavu, viz číselník návratových stavů.
- **\$ErrMsg**: popis v případě chyby.
- **\$RefNumber**: přidělené číslo jednací.
- **\$DocIdentifier**: ID_UFRM vytvořeného MSG_IN.
- **\$IdUfrm**: ID_UFRM vytvořeného MSG_IN.
- **\$Temp**: pro zobrazení outWsParam.
- **\$PluginData**: PluginDataStruct.

3.7.4 Získat číslo jednací – ESSGetCj

Vstup:

```
class ESSGetCjRequest
{
    public $profileID; // int
    public $IdSubject; // string
    public $idCert; // int
    public $PluginData; // PluginDataStruct
}
```

- **\$profileID**: IDENTIFIKUJE profil, který tato WS používá.
- **\$IdSubject**: ID obsluhovaného subjektu zadávaný přes WS request, jinak se získá z ws_profile (id_serv_subj).
- **\$idCert**: ID obsluhovaného subjektu.
- **\$PluginData**: PluginDataStruct.

Výstup:

```
class ESSGetCjResponse{
    public $ResultCode; // int
    public $ErrMsg; // string
    public $Cjs; // CjList
    public $PluginData; // PluginDataStruct
}
```

- **\$resultTextCode**: číslo návratového stavu, viz číselník návratových stavů.
- **\$ErrMsg**: popis v případě chyby.
- **\$Cjs**: seznam čísel jednací.
- **\$PluginData**: PluginDataStruct.

3.7.5 Získat seznam – ESSGetList

Vstup:

```
class ESSGetListRequest{
    public $profileID; // int
    public $IdSubject; // int
    public $IdUser; // int
    public $UserEmail; // string
    public $RequestFor; // int
    public $Start; // int
    public $Limit; // int
    public $Jid; // string
    public $Cj; // string
    public $Name; // string
    public $CreatedDateFrom; // dateTime
    public $CreatedDateTo; // dateTime
    public $ModificationDateFrom; // dateTime
    public $ModificationDateTo; // dateTime
    public $DeliveryDateFrom; // dateTime
    public $DeliveryDateTo; // dateTime
    public $PodaciCislo; // string
    public $PodaciPosta; // string
    public $DeliveryMethod; // int
    public $IdFirma; // string
    public $SpisZnak; // string
    public $SkartZnak; // string
    public $SkartLhuta; // int
    public $ClassificationCode; // string
    public $ShreddingMark; // string
    public $ShreddingPeriod; // int
    public $SenderCj; // string
    public $SenderSpisZnacka; // string
    public $SenderRefNumber; // string
    public $SenderId; // string
    public $RecipientRefNumber; // string
    public $RecipientIdent; // string
    public $Archived; // int
    public $PluginData; // PluginDataStruct
}
```

- **\$profileID**: identifikuje profil, který tato WS používá.
- **\$IdSubject**: ID obsluhovaného subjektu zadávaný přes WS request, jinak se získá z ws_profile (id_serv_subj).
- **\$IdUser**: ID uživatele.
- **\$UserEmail**: e-mail uživatele, který dostane spis na stůl.
- **\$RequestFor**: 0=spisy, 1=dokumenty.
- **\$Start**: od 1.
- **\$Limit**: počet záznamů v odpovědi.
- **\$Jid**: přidělený JID.
- **\$Name**: jméno.
- **\$CreatedDateFrom**: vytvořeno od.
- **\$CreatedDateTo**: vytvořeno do.
- **\$ModificationDateFrom**: upraveno od.
- **\$ModificationDateTo**: upraveno do.
- **\$DeliveryDateFrom**: doručeno od.
- **\$DeliveryDateTo**: doručeno do.
- **\$PodaciCislo**: podací číslo.
- **\$PodaciPosta**: podací pošta.
- **\$DeliveryMethod**: ID způsobu doručení.
- **\$IdFirma**: DZHP_FIRMA.ID_FIRMA Má přednost před ostatním, update adresáře se neprovádí.
- **\$ClassificationCode**: spisový znak.
- **\$ShreddingMark**: skartační znak.
- **\$ShreddingPeriod**: skartační lhůta.
- **\$SenderRefNumber**: číslo jednacích odesílatele SenderRefNumber.
- **\$SenderId**: spisová značka odesílatele SenderIdent.
- **\$RecipientRefNumber**: číslo jednacích příjemce RecipientRefNumber.
- **\$RecipientId**: spisová značka příjemce RecipientIdent.
- **\$Archived**: 0=rozpracované, 1=archivované, nic=rozpracované+archivované.
- **\$PluginData**: PluginDataStruct.

Výstup:

```
class ESSGetListResponse{
    public $ResultCode; // int
    public $ErrMsg; // string
    public $Items; // ItemListStruct
    public $PluginData; // PluginDataStruct
}
```

- **\$resultTextCode**: číslo návratového stavu, viz číselník návratových stavů.
- **\$ErrMsg**: popis v případě chyby.
- **\$Items**: ItemListStruct.
- **\$PluginData**: PluginDataStruct.

3.7.6 ESSSetCategory

Vstup:

```
class ESSSetCategoryRequest{
    public $profileID; // int
    public $IdSubject; // int
    public $IdUserTo; // int
    public $UserEmail; // string
    public $IdCategory; // int
    public $Category; // string
    public $IdDoc; // int
    public $Jid; // string
    public $Cj; // string
}
```

- **\$profileID**: ID obsluhovaného subjektu zadávaný přes WS request, jinak se získá z `ws_profile.php`.
- **\$IdSubject**: starý element – ID obsluhovaného subjektu zadávaný přes WS request, jinak se získá z `ws_profile (id_serv_subj)`.
- **\$IdUserTo**: ID uživatele, který mění kategorii.
- **\$UserEmail**: E-mail uživatele, který mění kategorii.
- **\$IdCategory**: ID kategorie.
- **\$Category**: název kategorie.
- **\$IdDoc**: ID_UFRM dokumentu.
- **\$Jid**: přidělený JID.
- **\$Cj**: přidělené číslo jednací.

Výstup:

```
class ESSSetCategoryResponse{
    public $ResultCode; // int
    public $ErrMsg; // string
}
```

- **\$resultTextCode**: číslo návratového stavu, viz číselník návratových stavů.
- **\$ErrMsg**: popis v případě chyby.

3.7.7 Získání konkrétního ESS dokumentu – ESSGetDocument

Vstup:

```
class ESSGetDocumentRequest{
    public $profileID; // int
    public $IdSubject; // int
    public $IdUserTo; // int
    public $UserEmail; // string
    public $IdDoc; // int
    public $Jid; // string
    public $Cj; // string
}
```

- **\$profileID**: ID konfigurace WS (**config_ws_profile**), která může obsahovat ID obsluhovaného subjektu a ID uživatele
- **\$IdSubject**: ID obsluhovaného subjektu zadávaný přes WS request, jinak se získá z config_ws_profile (**id_serv_subj**).
- **\$IdUserTo**: ID uživatele, který dokument stahuje. Musí mít na dokument právo.
- **\$UserEmail**: E-mail uživatele, který dokument stahuje. Musí mít na dokument právo.
- **\$IdDoc**: ID_UFRM dokumentu.
- **\$Jid**: JID dokumentu.
- **\$Cj**: Číslo jednacích dokumentu.
Musí být zadáno **IdDoc** nebo **Jid** nebo **Cj**

Výstup:

```
class ESSGetDocumentResponse{
    public $Result; // Výstup, 0 = O.K., vše ostatní nějaká chyba
    public $ErrMsg; // Popis chyby
    public $InfoMsg; // Informace
    public $id_cert; // ID obsluhovaného subjektu
    public $id_user; // ID_USER uživatele, který dokument vyřizuje nebo vyřídil
    public $id_ufrm; // ID_UFRM nalezeného dokumentu
    public $Type; // Typ nalezeného dokumentu (1=příchozí, 2=odchozí)
    public $DocumentName; // Název nalezeného dokumentu
    public $Cj; // Č.j. nalezeného dokumentu
    public $Jid; // JID nalezeného dokumentu
    public $CreateTimestamp; // Datum a čas vytvoření nalezeného dokumentu
    public $ModificationTimestamp; // Datum a čas poslední modifikace dokumentu
    public $Category; // Kategorie nalezeného dokumentu
}
```

```

public $FinishingMethod; // Vyřízení nalezeného dokumentu
public $ExecutionMethod; // Popis spouštěcích událostí nalezeného dokumentu
public $SpisZnak; // Spisový znak nalezeného dokumentu
public $SkartZnak; // Skartační znak nalezeného dokumentu
public $SkartLhuta; // Skartační lhůta nalezeného dokumentu
public $NumPhysicalSheets; // Počet listů nalezeného dokumentu
public $NumAttachementSheets; // Počet listů příloh nalezeného dokumentu
public $PhysicalArchive; // Místo uložení nalezeného dokumentu
public $SenderRefNumber; // Č.j. odesílatele
public $SenderId; // Spisová značka odesílatele
public $RecipientRefNumber; // Č.j. příjemce
public $RecipientId; // Spisová značka příjemce
public $ToHands; // K rukám
public $PersonalDelivery; // Do vlastních rukou
public $AllowSubstDelivery; // Povoleno doručení fikcí
public $DocumentData; // Celé ZFO formuláře
public $SenderRecipientList; // Odesílatel/příjemci dokumentu - struktura
public $InSpis; // Do kterých spisů je nalezený dokument vložen - struktura
public $CommentList; // Seznam komentářů nalezeného dokumentu - struktura
public $DocumentList; // Přílohy dokumentu - struktura
}

```

Struktura \$SenderRecipientList - Odesílatel/příjemci dokumentu

```

class SenderRecipientList {
    public $SenderRecipient; // Jeden odesílatel/příjemce
}

class SenderRecipient {
    public $DeliveryTimestamp; // Datum a čas doručení
    public $SerialNumber; // Podací číslo
    public $PostOfOrigin; // Podací pošta
    public $TypeMsg; // Typ zprávy (1=DZ, 2=klasická poštovní zásilka, 3=email)
    public $DeliveryMethod; // Způsob doručení
    public $IdDS; // ID datové schránky odesílatele/příjemce
    public $Email; // E-mail odesílatele/příjemce
    public $IdFirma; // ID odesílatele/příjemce v adresáři
    public $IdKontakt; // ID kontaktu odesílatele/příjemce v adresáři
}

```

```

    public $AddrName; // Název odesílatele/příjemce
    public $AddrAddress; // Adresa odesílatele/příjemce
}

```

Struktura InSpis - V jakých spisech se nalezeny dokument nachází

```

class InSpis {
    public $OneSpis; // Jeden spis
}

```

Struktura \$OneSpis - jeden spis

```

class OneSpis {
    public $id_ufrm_spis; // DZHP_SPIS.ID_UFRM spisu
    public $spis_name; // Název spisu
    public $spis_znak; // Spisový znak
    public $skart_znak; // Skartační znak
    public $spisova_znacka; // Spisová značka
    public $id_firma; // ID dotčeného subjektu v adresáři
    public $adr_name; // Název dotčeného subjektu
    public $adr_address; // Adresa dotčeného subjektu
}

```

Struktura \$CommentList - Seznam komentářů nalezeného dokumentu

```

class CommentList {
    public $CommentItem; // Jeden komentář - struktura
}

```

Struktura \$CommentItem - Jeden komentář

```

class CommentItem {
    public $Comment; //Jeden komentář
    public $DocumentList; // Seznam příloh jednoho komentáře - struktura
}

```

Struktura \$DocumentList - Seznam souborů vložených do dokumentů nebo komentáře

```

class DocumentList {
    public $Document; // Jeden příložený soubor - struktura
}

```

Struktura \$Document - Jeden soubor

```

class Document {

```

```

public $Description; // Popis souboru
public $Filename; // Název souboru
public $MimeType; // Mime type souboru
public $Data; // Binární data souboru
public $InternalId; // ID nebo ID_SFLF z tabulky XG_SFLF
public $Ltd; // LTD - určuje, zda se bude soubor LTD udržovat - hodnoty 0,1,NUL
}

```

- **\$resultTextCode**: číslo návratového stavu, viz číselník návratových stavů.

3.7.8 Zjištění statusu odeslané Datové zprávy – ESSCheckSend

Vstup:

```

class ESSCheckSendRequest{
    public $DocIdentifier; // int
    public $IdUfrm; // int
}

```

- **\$DocIdentifier** nebo **IdUfrm**: ID_UFRM odeslaného MSG_OUT.

Výstup:

```

class ESSCheckSendResponse{
    public $ResultCode; // < 0 nějaká chyba, která je popsána v $ErrMsg
        1 = dokument je rozpracovaný
        2 = dokument je odeslán všem adresátům
        3 = dokument je dodán všem adresátům
        4 = dokument je doručen všem adresátům
    public $ErrMsg; // Popis chyby
    public $InfoMsg; // Informace
}

```

4 Práce s dokumenty v LTV

4.1 Služba Service602LTV, computeHashDocument

Tato metoda slouží pro získání výpočtu hash jednoho dokumentu na základě jeho **binárního obsahu**.

Vstup:

```
class ComputeHashDocument
{
    public $request; /* @var request */
    public $input; /* @var base64Binary */
    public $externalId; /* @var externalId */
    public $url; /* @var string */
    public $hashType; /* @var string */
}

class request
{
    public $system; /* @var system */
    public $requestId; /* @var string requestId = OPERATION_ID */
    public $businessRequestId; /* @var string businessRequestId externalId. */
}
```

- **\$input**: binární obsah v base64 tvaru.
- **\$externalId**: volitelně, binární soubor bude stažen a uložen pomocí sfil external_id.
- **\$url**: relativní endpoint v módu S, anebo plné url v módu C.
- **\$hashType**: výchozí hodnota "SHA-256" else "SHA-512" or "MD5".
- **\$system**: identifikace systému, požaduje vyplněnou hodnotu z TIF hlavičky: RQ_APP_ID.
- **\$requestId**: identifikace systému, požaduje vyplněnou hodnotu z TIF hlavičky: OPERATION_ID.
- **\$businessRequestId**: identifikace požadavku businessRequestId (nepoužívá se pro TIF).

Výstup:

```
class ComputeHashDocumentResponse
{
    public $result; /* @var result */
    public $hashDocument; /* @var base64Binary */
}
class result
{
    public $resultCode; /* @var resultCode */
    public $errorCode; /* @var errorCode */
    public $errorDescription; /* @var errorDescription */
    public $requestId; /* @var requestId */
    public $businessRequestId; /* @var businessRequestId */
}
```

- **\$resultCode**: stavový kód, 0 je OK. Číselník kódů je ve výčtu `\FS\XG\WS\Service602LTV\SvcStatus.php`.
- **\$errorCode**: číselný kód chyby nebo varování.
- **\$errorDescription**: popis chyby nebo varování.
- **\$requestId**: identifikace systému, požaduje vyplněnou hodnotu z TIF hlavičky: OPERATION_ID.
- **\$businessRequestId**: označení korelačního id, využívá volající.

4.2 Služba Service602LTV, LtvAnalyze

Tato metoda vrátí ve své odpovědi konvertovaný soubor do PDF. V požadavku lze určit, zda má být výsledné PDF elektronicky podepsané a zda má být použitý podpis vizualizovaný či nikoliv.

Vstup:

```
class LtvAnalyze
{
    public $request; /* @var request */
    public $input; /* @var base64Binary */
    public $externalId; /* @var externalId */
    public $fileType; /* @var string */
    public $bAllowUnknownCertificateAuthority; /* @var boolean */
    public $revocationChecking; /* @var revocationChecking */
    public $bReturnValidationMaterial; /* @var boolean */
    public $visibleSignChecking; /* @var boolean */
    public $vsPosX; /* @var double */
    public $vsPosY; /* @var double */
    public $vsSizeX; /* @var double */
    public $vsSizeY; /* @var double */
    public $vsPage; /* @var vsPage */

    public $validationMode; /** @var int|null validationMode = 1: pouze všechny platné kvalifikované podpisy jsou akceptovány. https://gitlab.602.cz/sofa/sofa/-/issues/735 */

    public $lang; /* @var string|null cs-CZ, en-US, ... pro identifikaci jazyka - v plánu, zatím se nepoužívá. */

    public $idLtdConf; /* @var ?int GL735: What tenant is calling this analysis for the purpose of evidence of possible added time stamp.*/

    public $guidLtdConf; /* @var ?string GL735: What tenant is calling this analysis for the purpose of evidence of possible added time stamp. UUID variant.*/
}
```

- **\$request:** technické parametry – vyplněno z TIF hlavičky.
- **\$input:** binární obsah v base64 tvaru.
- **\$externalId:** volitelně, binární soubor bude stažen a uložen pomocí `$external_id`.
- **\$fileType:** PDF nebo XML.
- **\$bAllowUnknownCertificateAuthority:** povolení neznámé certifikační autority (nekontrolují se seznamy CRL, pouze podpis a integrita dokumentu).
- **\$revocationChecking:** zda zkontrolovat seznam CRL, uvádí pouze poslední podpis ve stromě podpisu (EndCertificateOnly), přičemž se vynechá kořenový certifikát (ExcludeRoot) bez kontroly CRL (None). Ve výchozím nastavení je vše zaškrtnuto [ExcludeRoot;EndCertificateOnly;None].
- **\$bReturnValidationMaterial:** zda má výstup vrátit seznamy CRL, výchozí stav nevrací.
- **\$visibleSignChecking:** pokud je nastaveno na true, pak PKI zkontrolujte, zda jsou všechny viditelné znaky v povolené oblasti definované v `vsPosX`, `vsPosY`, `vsSizeX`, `vsSizeY`, `vsPage`.
- **\$vsPosX:** souřadnice pozice X.
- **\$vsPosY:** souřadnice pozice Y.

- **\$vsSizeX**: šířka X.
- **\$vsSizeY**: šířka Y.
- **\$vsPage**: číslo stránky na kterou se má vizualizace umístit.
- **\$validationMode**: pokud je nastaveno na 1, je globalStatus platný pouze v případě, že jsou platné všechny podpisy. Pokud není žádné koncové časové razítko, přidá proces jedno, pokud má dokument předpoklad být platný.
- **\$lang**: v jakém jazyku se vrací zprávy. Plánovaná funkce, zatím se nepoužívá.
- **\$idLtdConf**: jaký tenant (konfigurace) bude použit pro danou analýzu za účelem evidence možného přidaného časového razítka.
- **\$guidLtdConf**: jaký tenant (konfigurace) bude použit pro danou analýzu za účelem evidence možného přidaného časového razítka (UUID variant).

Výstup:

```
class LtvAnalyzeResponse
{
    public $result; /* @var result */
    public $docType; /* @var string @deprecated */
    public $expiration; /* @var dateTime */
    public $docDataHash; /* @var base64Binary */
    public $docDataHashAlg; /* @var string */
    public $currSignHashAlg; /* @var string */
    public $globalStatus; /* @var globalStatus */
    public $sigInfos; /* @var sigInfos */
    public $docMimeType; /* @var string mime type */
    public $newExternalId; /* @var string|null nový binar, který vznikl pro $validationMode = 1. - Pdf dokumenty nemající jako poslední razítko je třeba nejprve o razit GL735 */
    public $globalStatusMessage; /* @var string|null Stručný txt popis o validaci certifikátu */
}
```

- **\$result**: 0=OK, 4=warning, 8=error, C=critical [hodnoty 0;4;8;C].
- **\$docType**: typ dokumentu.
- **\$expiration**: datum vypršení platnosti dokumentu, týká se všech podpisů a časových razítek.
- **\$docDataHash**: hash dokumentu.
- **\$docDataHashAlg**: hash algoritmu.
- **\$currSignHashAlg**: hash algoritmu posledního podpisu nebo časového razítka.
- **\$globalStatus**: globální stav platnosti dokumentu. VAROVÁNÍ znamená platný podpis v tuto chvíli, ale neověřený po období upozornění nebo chybí časové razítko. [výsledky: OK; WARNING; ERROR; NO_SIGNATURE].
- **\$sigInfos**: pole informací o podpisech.
- **\$docMimeType**: nahrazení docType úplným řetězcem mime type string.
- **\$newExternalId**: viz input validationMode=1.
- **\$globalStatusMessage**: krátké txt shrnutí výsledku ověření certifikátu. Vyplněno, když je validationMode = 1.

4.3 Služba Service602LTV, LtvDeleteTenant

Neveřejná ws.

Vstup:

```
class LtvDeleteTenant
{
    /**
     * @var Tenant $tenant
     */
    public $tenant;
}
```

Výstup:

```
class LtvDeleteTenantResponse
{
    /**
     * @var string $errorCode
     */
    public $errorCode;

    /**
     * @var string $errorDescription
     */
    public $errorDescription;
}
```

4.4 Služba Service602LTV, ItvGetDocumentInfo

Tato metoda slouží pro získání informací o konkrétním dokumentu, který je zařazen do dlouhodobé údržby.

Vstup:

```
class ItvGetDocumentInfo
{
    public $tenant; / * @var Tenant $tenant
    public $externalId; / * @var string $externalId
}
```

- **\$tenant:**
 - **\$idLtdConf:** jaký tenant (konfigurace) bude použit pro danou analýzu za účelem evidence možného přidaného časového razítka (UUID variant).
 - **\$guidLtdConf:** jaký tenant (konfigurace) bude použit pro danou analýzu za účelem evidence možného přidaného časového razítka (UUID variant).
- **\$externalId:** volitelně, binární soubor bude stažen / uložen pomocí sflf external_id.

Výstup:

```
class ItvGetDocumentInfoResponse
{
    public $errorCode; / * @var string $errorCode
    public $errorDescription; / * @var string $errorDescription
    public $externalFileId; / * @var string */
    public $externalBlobId; / * @var string */
    public $state; /* @var string */
    public $expirationDate; /* @var string */
}
```

- **\$errorCode:** číselný kód chyby nebo varování.
- **\$errorDescription:** popis chyby nebo varování.
- **\$externalFileId:** identifikátor souboru v externím systému.
- **\$externalBlobId:** informace o souboru v externím systému uloženém v Azure Blob Storage.
- **\$state:** celkový stav dokumentu.
- **\$expirationDate:** datum expirace.

4.5 Služba Service602LTV, LtvInsert

Služba pro zaregistrování dokumentu do dlouhodobé údržby.

Vstup:

```
class LtvInsert
{
    public $request; /* @var request */
    public $input; /* @var base64Binary */
    public $externalId; /* @var externalId */
    public $fileType; /* @var string */
    public $idLtdConf; /* @var idLtdConf */
    public $fileName; /* @var fileName */
    public $guidLtdConf; /* @var string Alternativa k $idLtdConf, sloupec XG_LCON
TENANT_IDENT_LCON*/
    public $ruleException; /* @var boolean*/
    public $baMode; /* @var int @see BaMode */
    public $baArchivedAt; /* @var null|string Timestamp of callers document archi
vation - in its application. If value is not empty, operation will re-register a
ll known documents with the same id and force to be sealed.*/
}
```

- **\$request:** technické parametry – vyplněno z TIF hlavičky.
- **\$input:** binární obsah v base64 tvaru.
- **\$externalId:** volitelně, binární soubor bude stažen / uložen pomocí sfil external_id.
- **\$fileType:** PDF nebo XML.
- **\$idLtdConf:** jaký tenant (konfigurace) bude použit pro danou analýzu za účelem evidence možného přidaného časového razítka.
- **\$fileName:** název vkládaného souboru.
- **\$guidLtdConf:** jaký tenant (konfigurace) bude použit pro danou analýzu za účelem evidence možného přidaného časového razítka (UUID variant).
- **\$ruleException:** umožňuje udržovat dokumenty, jejichž některé podpisy jsou neplatné.
- **\$baMode:** N/A nebo 0 znamená klasické vkládání do DDA.
1 – pouze jednou ověří časové razítko nebo se zaregistruje, aby se předešlo jeho expiraci. Vhodné pro rozpracované dokumenty.
2 - plně archivováno do účtovaného archivu.

Výstup:

```
class LtvInsertResponse
{
    public $result; /* @var result */
}
```

- **\$result:** technické parametry mapované z TIF hlavičky.
- **\$resultCode:** stavový kód, 0=OK, 4=warning, 8=error, C=critical [0;4;8;C].

4.6 Služba Service602LTV, LtvRegisterTenant

Neveřejná ws. Tato metoda slouží pro registraci dokumentu do databáze Secusign.

Vstup:

```
class LtvRegisterTenant
{
    public $tenantName; /* @var string $tenantName Jmeno firmy
    public $tenantIdent; /* @var string $tenantIdent UUID identifikator firmy
    public $sealConfig; /* @var SealConfig $sealConfig
    public $azureBlobTenantConfig; /* @var AzureBlobTenantConfig $azureBlobTenantConfig
    public $tenantBIN; /* @var string $tenantBIN IC */
    public $tenantBModel; /* @var int $tenantBModel 1 - SofaBased, 2 - */
}
```

Výstup:

```
class LtvInsertResponse
{
    public $result; /* @var result */
}
```

4.7 Služba Service602LTV, LtvUnregister

Tato metoda slouží pro odregistrování LTV udržovaného dokumentu z databáze SecuSign.

Vstup:

```
class LtvUnregister
{
    public $tenant; /* @var Tenant $tenant
    public $externalId; /* @var string $externalId
}
```

- **\$tenant:**
 - **\$idLtdConf:** jaký tenant (konfigurace) bude použit pro danou analýzu za účelem evidence možného přidaného časového razítka (UUID variant).
 - **\$guidLtdConf:** jaký tenant (konfigurace) bude použit pro danou analýzu za účelem evidence možného přidaného časového razítka (UUID variant).
- **\$externalId:** volitelně, binární soubor bude stažen / uložen pomocí sflf external_id.

Výstup:

```
class LtvUnregisterResponse
{
    public $errorCode; /* @var string $errorCode
    public $errorDescription; /* @var string $errorDescription
}
```

- **\$errorCode:** číselný kód chyby nebo varování.
- **\$errorDescription:** popis chyby nebo varování.

4.8 Služba Service602LTV, SignDocument

Tato metoda slouží pro opatření předloženého dokumentu kvalifikovaným certifikátem.

Vstup:

```
class SignDocument
{
    public $request; /* @var request correlationId je $request->businessRequestId
    */
    public $input; /* @var base64Binary */
    public $externalId; /* @var externalId. Optionally binary will be downloaded a
    nd stored by sflf external_id. */
    public $id_docu; /* @var int. Optionally binary will be downloaded as current
    document version with id_docu and stored as a new version. */
    public $url; /* @var string */
    public $uri; /* @var string. Alternative source of file. EP40151. */
    public $fileType; /* @var string */
    public $addTimestamp; /* @var boolean */
    public $signatureType; /* @var signatureType */
    public $signatureID; /* @var signatureID */
    public $userData; /* @var userData */
    public $userDataVisible; /* @var userDataVisible */
    public $visibleSign; /* @var boolean */
    public $signNumber; /* @var integer */
    public $signReason; /* @var signReason */
    public $signLocation; /* @var signLocation */
    public $signComment; /* @var signComment */
    public $signImage; /* @var base64Binary */
    public $signPosX; /* @var double */
    public $signPosY; /* @var double */
    public $signSizeX; /* @var double */
    public $signSizeY; /* @var double */
    public $signPage; /* @var signPage */
    public $customValues; /* @var customValues */
}
```

- **\$request:** technické parametry – vyplněno z TIF hlavičky.
- **\$input:** binární obsah v base64 tvaru.
- **\$externalId:** volitelně, binární soubor bude stažen a uložen pomocí sflf external_id.
- **\$id_docu:** identifikátor dokumentu.

- **\$url**: relativní endpoint v módu S, anebo plné url v módu C.
- **\$uri**: alternativní zdroj souboru.
- **\$fileType**: PDF nebo XML.
- **\$addTimestamp**: False / True, nastavení, když potřebujete přidat časové razítko spolu s podpisem.
- **\$signatureType**: může obsahovat hodnoty "DEFAULT" [DEFAULT; ENVELOPED; DETACHED; ENCAPSULATE].
- **\$signatureID**: ID certifikátu, který má být použit pro podepsání.
- **\$userData**: uživatelská data, která budou použita v informacích o podpisu.
- **\$userDataVisible**: uživatelská data, která budou použita ve vizualizaci u podpisu.
- **\$visibleSign**: True=Vizuální podpis, False nebo prázdná hodnota = standardní způsob podepsání.
- **\$signNumber**: číslo podpisu (očekává se 1 až 5). Po vyplnění se z konfigurace použijí parametry signReason, signlocation, signComment, signImage, signPosX, signPosY, signSizeX, signSizeY.
- **\$signReason**: důvod podepsání.
- **\$signLocation**: podepsáno v
- **\$signComment**: komentář k podpisu.
- **\$signImage**: obrázek použitý k podpisu.
- **\$signPosX**: X pozice od levého horního rohu pro oblast vizuálního podpisu.
- **\$signPosY**: Y pozice od levého horního rohu pro oblast vizuálního podpisu.
- **\$signSizeX**: X velikost objektu vizuálního podpisu v milimetrech.
- **\$signSizeY**: Y velikost objektu vizuálního podpisu v milimetrech.
- **\$signPage**: číslo stránky kam bude umístěn podpis, hodnota MAXINT = poslední stránka.
- **\$customValues**: vlastní hodnoty.

Výstup:

```
{
    public $result; /* @var result */
    /* @var base64Binary */ public $output;
    public $uri; /* @var string. Alternative uri of signed file. EP40151. */
}
```

- **\$result**: 0=OK, 4=warning, 8=error, C=critical [hodnoty 0;4;8;C].
 - **\$errorCode**: číselný kód chyby nebo varování.
 - **\$errorDescription**: popis chyby nebo varování.

4.9 Služba Service602LTV, TimestampPdfDocument

Tato metoda slouží pro opečetění předloženého dokumentu kvalifikovanou pečeti.

Vstup:

```
class TimestampPdfDocument
{
    public $request; /* @var request */
    public $input; /* @var base64Binary */
    public $externalId; /* @var externalId. Optionally binary will be downloaded and stored by sflf external_id. */
    public $id_docu; /* @var int. Optionally binary will be downloaded as current document version with id_docu and stored as a new version. */
    public $uri; /* @var string. Alternative source of file. EP40151. */
}
```

- **\$request:** technické parametry – vyplněno z TIF hlavičky.
- **\$input:** binární obsah v base64 tvaru.
- **\$externalId:** volitelně, binární soubor bude stažen a uložen pomocí sflf external_id.
- **\$id_docu:** identifikátor dokumentu.
- **\$uri:** parametr URI se předává na pluginu, které následně provede stažení souboru z externího zdroje a vrátí soubor.

Výstup:

```
class TimestampPdfDocumentResponse
{
    public $result; /* @var result */
    public $output; /* @var base64Binary */
    public $uri; /* @var string. Alternative uri of timestamped file. EP40151. */
}
```

- **\$result:** 0=OK, 4=warning, 8=error, C=critical [hodnoty 0;4;8;C].
- **\$output:** binární obsah v base64 tvaru.
- **\$uri:** alternativní URI časového razítka z externího zdroje.

5 Návrátové stavy WS

5.1 Úspěšné zpracování

Kód chyby	Chybový text	Význam
0	""	Volání WS proběhlo bez chyby

5.2 Rozsah 100–199 – chyby konfigurace a volání

Kód chyby	Chybový text	Význam
100		Blíže nespecifikovaná chyba konfigurace
101		Profil uvedený ve volání WS není na serveru konfigurovaný, neznámý profil
102		Definice profilu je neúplná, neobsahuje ID_FMVE formuláře
103		Definice profilu je neúplná, neobsahuje identifikaci obsluhovaného subjektu
104		Definice profilu je neúplná, neobsahuje definici SFLF / DMS
105		WS není autorizovaná
106		SFLF error

5.3 Rozsah 200–299 - chyby běhu

Jsou nejčastěji vyvolány chybným voláním služby.

Kód chyby	Chybový text	Význam
200		Blíže nespecifikovaná běhová chyba
201		Nezdařilo se stáhnout šablonu formuláře k vyplnění
202		Zadaný obsluhovaný subjekt nebyl nalezen a nelze ho použít
203		Obsluhovaný subjekt nemá zadaný výchozí spisový znak
204		Obsluhovaný subjekt nemá nastavený výchozí způsob vyřízení
205		Chyba autentizace, neplatný uživatel
206		Nezdařilo se přidělit ČJ/JID
207		Zadaná spisová značka není unikátní
208		Nezdařilo se přiřadit spisovou značku
209		Volání služby je neúplné, chybí vyžadovaná data, nejčastěji identifikace nebo adresa protistrany
210		Nezdařilo se nahrát soubory do SFLF
211		Volání služby neobsahuje žádnou přílohu
212		Volání služby neobsahuje název (anotaci) příloh
213		Metadata přiložených souborů nejsou kompletní
214		Vložení dokumentu do spisu selhalo
215		Nemohu vyhledat spis pro vložení
216		Spis nenalezen
217		Dokument nenalezen
218		Nemohu určit typ dokumentu
219		Omezení "od" nebo "do" nesplňuje požadavky. Nejčastěji od > do, od není definováno, atd.

220		Požadujete příliš mnoho výsledků (limit je 1000 na jedno volání WS)
221		Chyba běhu služby
222		Zadané ID kategorie není platné, kategorie neexistuje nebo je zakázána
223		Zadaný název kategorie není platný, kategorie neexistuje nebo je zakázána
224		Uživatel nemá právo k tomuto objektu
225		Uživatel nenalezen
226		Šablona formuláře nenalezena
227		Uživatel nemá právo k šabloně formuláře
228		ČJ nebylo nalezeno
229		Uživatel nemá právo k ID ČJ
230		Chyba při vytváření ČJ
231		Uživatel nemá právo k tomuto dokumentu
232		Chyba při vytváření JID
233		Chybná data
234		Dokument je již uzavřený
235		XML data jsou prázdná

5.4 Rozsah 300–399 – databázové chyby

Toto jsou stavy pomáhající v ladění a počátečním nastavení. Při správném použití by k jejich vyvolání nemělo dojít.

Kód chyby	Chybový text	Význam
300, 399		Blíže nespecifikovaná chyba databáze
301		Chyba manipulace s tabulkou XG_UFRM
302		Chyba manipulace s tabulkou XG_UFSS
303		Chyba manipulace s tabulkou XG_UFDF
304		Chyba zakládání záznamu v tabulce DZHP_SPIS
305		Chyba aktualizace záznamu v tabulce DZHP_SPIS
306		Chyba zakládání záznamu v deníku
307		Chyba zakládání záznamu v adresáři
308		Chyba modifikace záznamu v deníku
309		Chyba modifikace záznamu v adresáři
310		Chyba zjištění id ufrm
311		Chyba načtení seznamu kategorií dokumentů
399		Chyba databáze