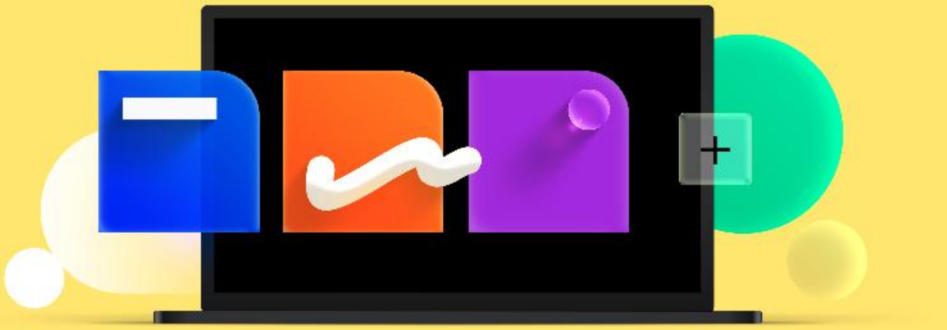

FormFlow
Web services

Last updated: 15.12.2025

Software602 a. s.
Hornokrčská 15
140 00 Praha 4

IČO: 63078236
DIČ: CZ63078236

www.602.cz
e-mail: info@602.cz
ID datové schránky: 7dcsfzg

Telefon: +420 222 011 602
Helpdesk: +420 222 011 254

Jednoduše.
Digitálně.
602

Contents

1 Introduction

1.1 Logging

Starting with version 10.2.3, web services and clients that are children of **FS\WSService** or **FS\WSClient** are logged by default. This can be turned off by setting the configuration parameter **XMLGW_CSV_LOG_WS_SERVER** or **XMLGW_CSV_LOG_WS_CLIENT** to false.

The log file is **XMLGW_LOGDIR/XMLGW_NAME_ws.RRRR-MM-DD.log.csv**.
The csv structure corresponds to the **FS\WSLogRec** class.

You can set runtime dumping in **config.php** using the following define:
define('XMLGW_WS_LOGDIR', XMLGW_LOGDIR . '/ws_log');
Not that the dump includes documents and other data that needs to be protected.

```
class WSLogRec
{
  /** @var string date of writing the request to 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 time of receiving data, in seconds. Cannot be determined when
in mode S.*/
  public $inDataRecvTime;
  /** @var int Number of bytes received. In mode S, unless XMLGW_WS_LOG_DIR is
set, the value of Content-Length from the request headers will be used.*/
  public $inDataRecvBytes;
  /** @var float Time of request handling before the processing itself using the
given web method logic, in seconds. Only logged in mode S.*/
  public $prepareTime;
  /** @var float Time of processing using the given web method logic, in
seconds. Only logged in mode S.*/
  public $handleTime;
  /** @var float Time (in seconds) of sending the soap response in mode S, or of
the soap request in mode C.*/
  public $outDataSentTime;
  /** @var int Number of sent bytes of response in mode S, or of the request in
mode C.*/
  public $outDataSentBytes;
  /** @var string 'S' for web service, 'C' for web service client. */
  public $mode;
  /** @var string relative endpoint in mode S or full url in mode C. */
  public $url;
  /** @var string pouzita metoda, musi podporovat implmentace dane webove
sluzby. Only logged in mode S.*/
  public $method;
  /** @var string If XMLGW_WS_LOG_DIR is set, relative path to file with
communication dump.*/
  public $dumpFile;
  /** @var string Id for business request binding. Can be entered in the X-
CorrelationId http header of the request or in the parameters of the specific ws
-
   * must support this. Only logged in mode S.*/
  public $correlationId;
  /** @var string Result of the method call. Must be supported by the given ws.
Only logged in mode S.*/
  public $result;
  /** @var string General exception.*/
 public $fatalError;
}
```

2 Document section

Web services (WS) are part of the standard FormFlow distribution, no additional configuration is needed. The ServiceDG web service has two endpoints:

- `/htdocs/soap/ServiceDGEndpoint.php` – this alternative will use authentication settings using the ws_authenticator nette service and is intended chiefly to serve the FormFlow Office Extension.
- `/htdocs/soap2/ServiceDGEndpoint.php` – alternative to be used in the case when an external system needs a different authentication model than FormFlow Office Extension does. E.g. SSO vs. basic authentication. If Windows authentication or basic Windows authentication is required, it is important for both the alternatives to set the authentication with /soap, /soap2 folders in IIS.

2.1 Change for ServiceDG when working with lists, starting with FormFlow version 10.2.2

Two optional parameters were added to the **meta** section in all cases: **value_code** and **value_name**. On input, **value_code** will take precedence over the original **value**. On output, **value_code** will always be used and only list items will use **value_name**.

This change has been introduced to allow sending codes (not database **int ID**) and to be able to display list texts. The changes concern the following services:

- getDocument
- searchDocument
- setDocument.

Except list items or specific items (see below), the **value_code** will match **value**, so that the source system does not need to check whether it is a list item or not. If **value_code** is entered on input, it always takes precedence over **value**.

Changes on service input

- If a non-empty **value_code** is given on input, it will behave for non-list fields (and not special fields, see below) the same way as **value** has behaved until now, that is, the metadata value (**value_code** takes precedence over **value**)
- If a non-empty **value_code** is given in input and it is a list value (or a special value, see below), the **value_code** will represent the following data:
    - Class – class ID
    - Rank – rank ID
    - Status – status ID
    - Folder – folder SID
    - Other lists – list key.

Changes on service output

- The parameters **value_code** and **value_name** are added to the output (**value_name** is for the list items only).

Where **value_code** means:

- Class ID
- Rank ID
- Status ID
- Folder SID
- List item key
- For other items (text, date, etc.), the value will be the same as in **value.**

Value_name:

- class name
- rank name
- status name
- folder name
- list item name
- For other items (text, date, etc.) the element will not be given (or will be empty).

Impacts:

Higher data flow – the output includes data that is not always needed.

Solution:

Element **return_value_code_name_only** of *boolean* type will be added to the input:

- **true** – only value_code and value_name will be returned to the output, value will not be returned
- Unspecified or **false** or empty or unspecified – only value will be output, as until now.

Note: If **return_value_code_name_only** is unspecified or **false**, **value_code** and **value_name** for list metadata is always returned.

2.2 ServiceDG return statuses

Error code	Error text	Meaning
0	""	The WS call completed without errors
-1	`E_GENERAL`	General error
-2	`E_SQL`	General SQL error
-3	`E_UNKNOWN_FOLDER`	Unknown folder
-4	`E_UNAUTHORIZED`	Unauthorized request
-5	`E_INVALID_ARGUMENT`	Invalid argument
-6	`E_EMPTY_INPUT_FILE_ARRAY`	Empty input file array
-7	`E_INVALID_INPUT_FILE_BASE64`	Empty input file array in base64 format
-8	`E_EMPTY_MD5`	Empty MD5 checksum
-9	`E_SFLF`	Error when using external file store
-10	`E_EMPTY_EXTERNAL_SYSTEM_ID`	Empty external system ID

-11	E_UNKNOWN_MIME_TYPE	Unknown MIME type
-12	E_ASYNCHRONOUS_INSERTION_PERMANENTLY_FAILED	Asynchronous insertion failed permanently
-13	E_EMPTY_FILENAME_OF_MERGED_FILE	Empty filename of a merged file
-14	E_INVALID_FILENAME_OF_MERGED_FILE	Invalid filename of a merged file
-15	E_UNKNOWN_DOC_STATUS	Unknown document status
-16	E_NO_FILE_CONTENT	No file content
-17	E_AMBIGUOUS_FILE_CONTENT	Ambiguous file content
-18	E_NO_CONTRACT_NO	No contract number
-19	E_DOCUMENT_NOT_FOUND	Document not found
-20	E_DOCUMENT_BINARY_DATA_NOT_AVAILABLE	The binary data of the document is not available
-21	E_UNKNOWN_SID_FLDR	Unknown document folder SID
-22	E_MULTIPLE_FILES_AND_FILE_ECM_ID_SPECIFIED	A shallow copy from an existing dms document is requested, but merge is also requested.
-23	E_UNSPECIFIED_FOLDER	Unspecified folder
-24	E_UNSUFFICIENT_SPIS_ZNAK_METADATA	Unspecified file plan code
-25	E_NO_FILE_MARK	No file mark
-26	E_INVALID_FILE_MARK	Invalid file mark
-27	E_INVALID_DISCARD_CODE	Invalid discard code
-28	E_INVALID_DISCARD_PERIOD	Invalid discard period
-29	E_UNKNOWN_CONSTRAINT_TYPE	Unknown constraint type
-30	E_UNKNOWN_CONSTRAINT_OPERATOR	Unknown constraint operator

-31	E_DZHP_INSERT_SOME_FAILED	Error on input from DZHP
-32	E_SQL_DEADLOCK	SQL deadlock
-33	E_INPUT_BINARY_DATA_MUST_BE_SET	File on input, the data must be specified
-34	E_INPUT_DOCUMENT_RELATION_IS_EMPTY	Document on input, the relation is empty
-35	E_INPUT_DOCUMENT_RELATION_PRIMARTY_ID_DOCU_IS_EMPTY	Document on input, ID_DOCU is empty
-36	E_INPUT_DOCUMENT_RELATION_RELATED_ID_DOCU_IS_EMPTY	No ID_DOCU is specified to create a relation
-37	E_INPUT_DOCUMENT_RELATION_RELATED_NAME_IS_EMPTY	The document name for relation is not specified
-38	E_INPUT_SEARCH_DOCUMENT_RELATION_IS_EMPTY	The relation to search is not specified
-39	E_INVALID_SEARCH_DOCUMENT_RELATION_DIRECTION_VALUE	Invalid value of document relation direction
-40	E_DOCUMENT_RELATION_DOESNT_EXIST	The relation specified does not exist
-41	E_EMPTY_REQUEST	Empty request
-42	E_DOCUMENT_IN_SPIS_CANNOT_BE_DELETED	A document placed in a record cannot be deleted
-43		Not in use
-44	E_CANNOT_WRITE_FILE_TO_UNC_PATH	Cannot write a file to the unc path specified
-45	E_CANNOT_FIND_VERSION	Document version not found
-46	E_CANNOT_GET_VERSION_BINARY	Binary file for the version not found
-47	E_REQUIRED_METADATA_IS_EMPTY	The required metadata is empty
-48	E_UNKNOWN_ID_CERT_DOCU	Unknown ID_CERT_DOCU entity

-49	E_UNKNOWN_ID_CERT_DOCU_IS_MISSING	A file plan code is specified, but no entity has been specified
-50	E_UNKNOWN_FILE_MARK_DOCU	Unknown document mark
-51	E_UNKNOWN_ID_ARIS_DOCU	A conversion of submitter ID is requested, but ID_ARIS_DOCU, external system id, is not specified
-52	E_FILE_MARK_DOCU_IS_MISSING	A document mark is missing
-53	E_INVALID_SHRED_MARK_DOCU	Invalid retention code
-54	E_SHRED_TIME_DOCU_IS_MISSING	A retention code is specified, but no retention period has been specified
-55	E_EMPTY_ID_ARDE_VALUE_NAME	Requested to create a new submitter, but the value_name entry is empty.

2.3 ServiceDG, getDocument

This method is used to get a single document based on its **id** or two values (**external_id_docu**, **external_system_id_docu**).

Input:

```
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, FF version 10.2.2 and up
  public $return_xml_datetime; // boolean, FF version 10.2.2 and up
  public $fileUncPath; // string, FF version 10.2.3 and up
  public $version_dver; //int, 10.2.2.0.3 and up
  public $id_dver; // int, 10.2.2.0.3 and up
  public $correlatinId // Optional, 10.2.3 and up
}
```

- **$id_docu**: document indentifier.
- **$external_id_docu**: external document id (DG_DOCU > EXTERNAL_ID_DOCU).
- **$external_system_id_docu**: external system id (DG_DOCU > EXTERNAL_SYSTEM_ID_DOCU).
- **$fileRequired**: specifies whether the binary content of the current document version is requested. It is possible to only request metadata.
- $**metaList**: is the list of requested metadata.
- **$getAllMetadaData**: is an alternative to **$metaList**, to get all metadata without having to know the whole list.
- **$tryLockFile**: obsolete, do not use
- **$return_value_code_name_only**: metadata values in the **MetaValue** structure will not be in the **$value** item but in **$value_code**. For lists, classes, ranks and statuses the **$value_code** will contain the key/list ID and **$value_name** will contain the list item name.
- **$return_xml_datetime:** If true, the values of date and datetime data types will be returned as string in ISO8601 format, otherwise formatted as Y-m-d H:i:s (php specification).
- **$fileUncPath**: if the parameter is specified, the output service will not return the document's binary contents but will save it to the requested UNC path.
    - The UNC path must be write-accessible for the DMS.
    - The file saved to the UNC path will not be deleted by the DMS, this is done by the service consumer.
    - The UNC path will contain the whole path, that is, including the file name.
- **$version_dver;** Optional. From 10.2.2.0.3 upwards, the version order number. If specified, the returned binary will contain the binary of the requested version specified here. Id_docu must be specified.

- **$id_dver**; Optional. From 10.2.2.0.3 upwards, system-wide unique version number. If specified, the returned binary will contain the contents of this version, id_docu then does not need to be specified. id_dver takes precedence over id_docu and ($external_id_docu, $external_system_id_docu).
- **$correlationId**: Optional. Allows grouping of identification of several calls in FormFlow logs.

Output:

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

- **$resultText**: status code, 0 is OK. The code list is given in FS\DG\WS\WSStatus.
- **$restulText**: status message, typically an error description.
- **$metaValueList**: list of metadata with their values if requested. The values are always of the string type.
- **$file**: structure containing the binary data of the current document version, if requested.

Additional information:

If transferring information of a date or date and time type, the ISO 8601 format is expected.

The method can be called without requesting binary data or metadata, then it only means a test of the existence of the document.

```
class MetaValue extends MetaSpec
{
  /* @var string IN/OUT. */
  public $value;
  /** @var string IN/OUT. Can be used for lists to get a list item's key.*/
  public $value_code;
  /** @var string OUT. For lists returns the value of a list item. */
  public $value_name;
}
```

Call example – request:

```
<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>
        <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>
```

Call example – response:

```
<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>Client</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 content</fileContent>
          <fileName>1.pdf</fileName>
          <mimeType>application/pdf</mimeType>
          <extension>pdf</extension>
        </file>
      </return>
    </ns1:getDocumentResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

2.4 ServiceDG, setDocument

This method is used to create a document if **id_docu** is not specified. If specified, it can be used to update a document.

Input:

```
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 Optional. Version 10.2.3 and up */
  public $correlationId;
  /** @var boolean Optional Od 10.2.2.0.4. If true and the document still exists, it is ignored */
  public $createNewVersion;
  /** @var boolean */
  public $createUnknownEnumItems;
  }
```

- **$id_docu**: document id should be specified when updating the metadata or binary contents of the current version of a document. When creating a new document, it must be empty.
- **$folder**: unique folder name (SID_FLDR), used only when creating a new document. Move to a folder is not implemented.
- **$id_fldr**: unique folder int id. Can be used instead of $folder.
- **$metaValueList**: list of metadata for the document being created or updated.
- **$fileList**: list of binary files. Currently only accepting 0 or 1 file. (values higher than 1 are reserved for future use, if needed). The **DocumentFile** structure must contain the file name including extension or **IANA mime** type.
- **$tryLockFile**: Attempts to lock the document for the document owner
- **$formFlowHashId**: FormFlow instance identification, preparation for FormFlow Office Extension, leave empty.

- **$correlationId**: An option to enter own identifier for grouping the identification of more than one call.
- **$createNewVersion:** If true, creates a new version of the document and places the submitted binary in it.
- **$createUnknownEnumItems:** Starting with version 10.2.4, creates new entries of a general list based on value_code and value_name, see below.

```
class MetaValueList
{
  /* @var MetaValue[] */
  public $meta;
}
class MetaValue extends MetaSpec
{
  /* @var string IN/OUT. */
  public $value;
  /** @var string IN/OUT. Can be used for lists to get a list item's key.*/
  public $value_code;
  /** @var string OUT. For lists returns the value of a list item. */
  public $value_name;
}
```

Notes:

- If requesting to include the document in LTD maintenance, set a metadatum (one item in list **$metaValueList**) with name_meta = **"$LTD_ACCEPT_DOCU**" and **$value**="1". The default value is 0 (must not be NULL).
- Starting with FF version 10.2.2, it is possible to specify a list value using its string code by entering this key in item **$metaValue->value_code**. If **$value_code** is specified, it takes precedence over **$value**. See description of the **MetaValue** class in **getDocument** method.
- Starting with FF version 10.2.4, it is possible to specify value_name and if it is an unknown entry in a general list and the parameter $req->createUnknownEnumItems = 1 is specified, this list item is created.

System lists:

- You can use the following string codes in value_code:

name_meta	description
ID_CLAS_DOCU	Class
ID_RANK_DOCU	Rank
ID_STAT_DOCU	Status
ID_USER_DOCU	Author
OFFICER_ID_USER_DOCU	Officer
WARRANTOR_ID_GROU_DOCU	Warrantor
CURRENT_DVER_ID_MIME_DOCU	Mime type according to IANA
ID_ARDE_DOCU	Archived by (The DocsMaster module, version 10.2.4 and up)
ID_CERT_DOCU	Entity (version 10.2.4 and up)
ID_ARMP_DOCU	Processing method

The archive module:

- Since version 10.2.4. When inserting in archive folders, this metadata requires a correct value, if this metadata item in the given folder has IS_REQUIRED_MIN_WS_MGUI set to 1. Additionally, some metadata require other metadata (prerequisites) to be filled in. This also applies to the searchDocument method.

Metadata item	Prerequisites
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 – note: The combination of these four must be a valid one.

- ID_ARDE_DOCU – If an item unknown to FF is submitted in value_code, it is mandatory to specify value_name and this submitter is then created regardless of the value of $req->createUnknownEnumItems

  Example:

```
<meta>
                <dbtable_meta>DG_DOCU</dbtable_meta>
                <dbcolumn_meta>ID_ARDE_DOCU</dbcolumn_meta>
                <value_code>ESS</value_code>
                <value_name>John Submitter</value_name>
</meta>
```

- ID_ARMP_DOCU – This list does not have a value_name

DocumentFile structure:

```
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:** Replacement of base64 contents for input (setDocument) and output (getDocument)
- If **checkSumAlg** and **checkSumValue** are specified, then the hash of the input binary is checked on input. Algorithm list according to the php hash(.) function.
- **fileName:** document version name.
- **mimeType:** IANA mime type.
- **id:** not in use
- **extension:** extension for the binary, only for output in the getDocument(.) function

Output:

```
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
}
```

- **$resultText**: int status code WSStatus.
- **$resultTextText**: status message, typically an error description.
- **$id_docu**: document ID.
- **$version_dver**: From 10.2.2.0.3 upwards, additional information on created or updated version – version order number.
- **$id _dver:** From 10.2.2.0.3 upwards, additional information on the created or updated version – unique version number.

Call example – request:

```
<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>shouldBeOverwritten by value of 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>Some ref.no.</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>John Submitter</value_name>
          </meta>
        </metaValueList>
        <fileList>
          <fileContent>base64 data</fileContent>
          <fileName>1.pdf</fileName>
        </fileList>
      </req>
    </ns1:setDocument>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Call example – response:

```
<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 ServiceDG, searchDocument

This method is used to look up a document based on search conditions specified in the **$criteria** parameter. The metadata to be output must be specified.

Input:

```
class SearchDocumentReq
{ /* @var string */
  public $folder;
  /* @var MetaOperand[] */
  public $citeria;
  /* @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 Optional. Version 10.2.3 and up */
  public $correlationId;
}
```

- **$folder**: sid (string id) of a folder.
- **$criteria**: list of search conditions.
- **$pageNo**: page number from 1, if not specified, value 1 will be used.
- **$pageSize**: page size, if not specified, value 200 will be used.
- **$metaList**: list of output columns.
- **$orderByMeta**: column to order by; if not specified, no ordering will be used.
- **$orderAsc**: if unspecified or **true**, ascending order will be used; otherwise descending.
- **$return_value_code_name_only:** If true, then the MetaDataValue output will include an empty value attribute to save data transfer. The data will be in value_code and, in the case of a list, also in value_name; otherwise value will only be returned in value. The default value is not specified, which means false.
- **$return_xml_datetime:** If true, metadata of the datatime type is returned in ISO8601 xml format.
- **$correlationId**: Optional. Allows grouping of identification of several calls in FormFlow logs.

Input:

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

- **$type**: logical operation type: ‘AND’, ‘OR’.
- **$criteria**: subexpression, implemented by () in sql. The subexpression can be a string of subconditions. If **$criteria** is nonempty, no more items may be specified.
- **$metaSpec**: marks the column used for searching.
- **$operator**: =, >, >=, <, <=, <>, !=, startsWith, endsWith, isEmpty, isNotEmpty.
- **$metaValue**: literal – metadata value.
- **$metaCode: the** parameter has been introduced in FF version 10.2.2 to allow specifying a list item code. Has priority over **$metaValue**. For non-list items, **$metaCode** may contain the usual metadata value.

Notes:

- **isEmpty** and **isNotEmpty** are unary operators starting with FF version 10.2.2, which means that **metaValue** is left **null** and a metadatum with the value ‘’ or **NULL** is looked up.
- For operators startsWith and **endsWith**, characters _ and % may not be used in **$metaValue** or **$metaCode** as special characters for the SQL expression **LIKE**. These characters are escaped.
- If querying specific metadata in the archive module folders (version 10.2.4 and up), the condition must specify all prerequisite metadata - see the setDocument method, table in paragraph “The archive module”.

Output:

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

- **$return**: found document

Output:

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

- **$document**: found page of documents
- **$resultText**: int status code WSStatus.
- **$resultText**Text: status message, typically an error description.

Call example – request:

The example below implements a condition whose SQL form is as follows. It is a more complex example using a subexpression.

```
            (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>
        <citeria>
          <type>AND</type>
          <metaSpec>
            <name_meta>NAME_DOCU</name_meta>
          </metaSpec>
          <operator>endsWith</operator>
          <metaValue>_~_5c00f37d69e323.47435298.pdf</metaValue>
        </citeria>
        <citeria>
          <type>AND</type>
          <criteria>
            <type>OR</type>
            <metaSpec>
              <name_meta>ID_CLAS_DOCU</name_meta>
            </metaSpec>
            <operator>=</operator>
            <metaValue>Should be overwritten with classCode element</metaValue>
            <metaCode>OIA</metaCode>
          </criteria>
          <criteria>
            <type>OR</type>
            <metaSpec>
              <name_meta>ID_CLAS_DOCU</name_meta>
            </metaSpec>
            <operator>=</operator>
            <metaValue>Should be overwritten with classCode element</metaValue>
            <metaCode>OXA</metaCode>
          </criteria>
        </citeria>
        <citeria>
          <type>AND</type>
          <metaSpec>
            <name_meta>CJ_DOCU</name_meta>
          </metaSpec>
          <operator>isNotEmpty</operator>
        </citeria>
        <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>
```

Call example – response:

```
<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>Client</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>internal</value_code>
              <value_name>Internal</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 ServiceDG, createDocumentVersion

A service that creates a new document version.

Input:

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

- **$id_docu**: int document ID.

Output:

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

Call example – request:

```
<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>
```

Call example – response:

```
<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 ServiceDG, lockDocumentByUser

A method to lock with a user's lock.

Input:

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

- **$id_docu**: id of the document to be locked/unlocked.
- **$doLock**: if true, lock is requested; otherwise unlock.

Output:

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

}
```

If successful, **$resultText** = 0, otherwise non-zero along with an error message.

Call example – request:

```
<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>
```

Call example – response:

```
<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 ServiceDG, createDocumentRelation

A method to create document relation.

Input:

```
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;
}
```

The **DocumentRelation** entity contains the primary document ID, related document ID and relation name.

Output:

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

Call example – request:

```
<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>default_relation</relation_name>
        </documentRelation>
      </req>
    </ns1:createDocumentRelation>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Call example – response:

```
<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 ServiceDG, createDocumentRelationCollection

A method for bulk creation of primary document relation. If any relation creation is refused due to an error, all relations created up to that point are canceled and the whole request must be repeated.

Input:

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

- **$primary_id_docu** – int ID for the document to be primary in the relation.
- **$related_id_docu** – int ID for the document to be related in the relation.
- **$relation_name** – relation type ID (CODE_TREL).

Output:

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

- **$resultText** – int status, 0 is OK.
- **$resultTextTest** – string status message.

Call example – request:

```
<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>default_relation</relation_name>
      </req>
    </ns1:createDocumentRelationCollection>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Call example – response:

```
<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 ServiceDG, deleteDocumentRelation

A method to delete a document relation.

Input:

```
class DeleteDocumentRelationReq

{

  /* @var DocumentRelation */

  public $documentRelation;

}
```

Output:

```
class DeleteDocumentRelationResp

{

  /* @var int */

  public $result;

  /* @var string */

  public $resultText;

}
```

- result 0 means the status is OK.

Call example – request:

```
<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>default_relation</relation_name>
        </documentRelation>
      </req>
    </ns1:deleteDocumentRelation>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Call example – response:

```
<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 ServiceDG, searchDocumentRelation

A method for searching document relations.

Input:

```
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** to be present in the relation
- **relation_name** – relation type ID (CODE_TREL). May be empty, in that case all relations are returned regardless of type.

- **direction**:
  1: id_docu is to be primary in the relation,
  -1 id_docu is to be related in the relation,
  0 / empty: search regardless of direction,

Output:

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

- **result** 0 means the status is OK
- returns the **DocumentRelation[] array.**

Call example – request:

```
<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>
```

Call example – response:

```
<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>default_relation</relation_name>
        </documentRelation>
        <documentRelation>
          <primary_id_docu>175397</primary_id_docu>
          <related_id_docu>175395</related_id_docu>
          <relation_name>default_relation</relation_name>
        </documentRelation>
      </return>
    </ns1:searchDocumentRelationResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

2.12 ServiceDG, getDocumentVersions

A method to search for document versions, from 10.2.2.0.3 upwards

Input:

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

- **id_docu**, document ID
- **fileRequired**, specifies whether the output should contain the binary of the version.
- **versionNumber** – optional, limit the set of returned versions to one, based on the version order number

Output:

```
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 type*/
  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 means the status is OK
- returns the **DocumentVersion[] array.**

Call example – request:

```
<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>
```

Call example – response:

```
<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>Create11VersionsAndListThem.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>Q3JlYXRlMTFWZXJzaW9uc0FuZExpc3RUaGVtLjIwMTktMDgtMTNUMTM6MjQ6NDIrMDI
6MDAudmVyc2lvbjE=</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>

<fileContent>Q3JlYXRlMTFWZXJzaW9uc0FuZExpc3RUaGVtLjIwMTktMDgtMTNUMTM6MjQ6NDIrMDI
6MDAudmVyc2lvbjI=</fileContent>
        </documentVersions>
      </return>
    </ns1:getDocumentVersionsResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

2.13 DzhpService, insertDocument

A method to insert a DMS document in a record

Input:

```
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**, array of document IDs
- **id_ufrm_spis**, record ID
- **spisova_znacka** – file number
- **source,** source system ID

Output:

```
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 means the status is OK
- returns the **InsertDocumentInfoItem [] array.**

3 Electronic records management system (ERMS)

The web services offered by the *Electronic records management system* provide a limited way to use the features of the ERMS without the need to operate the ERMS directly using *FormFlow*.

The ERMS web service has three endpoints that are equal:

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

3.1 ERMS web service authentication

The user under whom the service is run may be, in the case of the so-called open system, specified in the profile (see below). When calling the WS, mandatory elements **UserName** and **Password** are passed on as part of the call; in the case of FFS installation that supports domains, the optional **Domain** is included as well (see wsdl, tns:AuthHeader type). However, most installations use the basic authentication. Therefore, starting with version 10.2.4.0.5, it is no longer necessary to specify UserName and Password in the call.

Input:

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

- **$UserName**: user name.
- **$Password**: password.
- **$Domain**: domain.

When calling the WS, the FFS checks whether the user exists, is allowed to log in to the FFS and is authorized to use the form specified in the profile. If these conditions are met, the WS continues to run. Otherwise, an error is returned (see response statuses).

3.2 Installation

WS are part of the standard ERMS distributions and do not need special installation. However, they need to be configured first. WS configuration is done after FFS configuration and after setting up the whole ERMS.

WS configuration consists of creating communication profiles. They are written in the **/dzhp/ws_profile.php**, whose structure can be copied from the distribution (sample) file, **/dzhp/ws_profile.php.distrib**.

The individual profiles are created as items of the configuration array, the profile number (key in the **$config_ws_profile** array) is included in service call. You can set up one or more profiles for the individual WS operations. That way you can make small changes in some WS configuration parameters without having to change the calling systems. They also do not need to always transfer the whole configuration information but give the profile number instead.

3.3 Profiles

Each WS call contains a profile number for the profile to be used (`$config_ws_profile[1]`) as a mandatory parameter. A profile is a set configuration on the application server side. The configuration includes:

- `id_serv_subj` – ID for the entity who will own all the forms created and in whose records management the forms are placed (`'id_serv_subj'=>4,`).
- `MSG_IN_id_fmve` – form ID for MSG_IN form template (`'MSG_IN_id_fmve'=>7,`).
- `MSG_OUT_id_fmve` – form ID for MSG_OUT template (`'MSG_OUT_id_fmve'=>8,`).
- `SPIS_id_fmve` – form ID for SPIS – record template (`'SPIS_id_fmve'=>4,`).
- `SPIS_cj_id` – numbering series for creating record numbers (`'SPIS_cj_id'=>4,`).
- `MSG_IN_can_use_default' => 0` – prohibits the use of default values when creating MSG_IN.
- `MSG_OUT_can_use_default' => 0` – prohibits the use of default values when creating MSG_OUT.
- `SPIS_can_use_default' => 0` – prohibits the use of default values when creating SPIS.
- `login_user` – optional parameter. When specified, authentication is ignored and the whole service is run under this user. It is recommended to use authentication and not specify this parameter. For FFS 6.0 with domain support and higher versions, the value from DB column `login_user` (`'login_user'=>user,`) is used.

Example:

```
$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 General data structures

Because operations run by the WS are not trivial and require large amounts of data to complete successfully, the data is structured in logical units. This results in a set of general data structures used repeatedly for various calls.

3.4.1 The key – value structure: KeyValuePair

General data structures used especially in plugins (see plugin), a trivial data structure **Key-value** is used. It always contains a text representation of the key with a text representation of its assigned value.

Input:

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

- **$Key**: A key that has a value assigned.
- **$Value**: The assigned value.

3.4.2 Data structure for the plugin: PluginDataStruct

Every WS call contains one element of **PluginDataStruct** type, in which 0 to N elements of **KeyValuePair** type can be inserted and used to pass on more data outside of common WS call to a plugin.

Input:

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

- **$Item**: 0 to N of key-value structures.

An example of filling PluginData for 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;Contract 111';
      $out_param[0]['PluginData'][2]['Key']='KLIENT_2';
      $out_param[0]['PluginData'][2]['Value']='2222278992;Contract 222';
      $out_param[0]['PluginData'][3]['Key']='KLIENT_3';
      $out_param[0]['PluginData'][3]['Value']='3333378992;Contract 333';
```

If there is more than one client, the **Key** must start with the string **KLIENT** which can then be followed by anything to make the keys unique. **`VALUE`** for one client has the following structure: **`RC ; CONTRACT_NO`**

`KEY IdMsg` is only used once and **`VALUE`** contains **`ID DZ`** to which it responds.

To submit the MSG_OUT automatically to the output queue, set the element **`WS SendType = 1;`**

3.4.3 Record structure: FileStruct

The structure describes the metadata of a single record, that is, the filing and retention data and subject information.

Input:

```
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
}
```

- **$id_spis**: record ID in the internal database (ID_UFRM).
- **$spis_name**: Record name.
- **$spis_znak**: File plan code.
- **$skart_znak**: Retention code.
- **$skart_lhuta**: Retention period.
- **$spisova_znacka**: Filing number.
- **$id_firma**: Subject ID in the internal database.
- **$adr_name**: Compound subject name.
- **$adr_address**: Compound subject address.

3.4.4 Record structure: Files

Simple structure describing 0 to N records. Contains **FileStruct** type items.

Input:

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

- **$File**: 0 to N structures describing records.

3.4.5 List of item structures: ItemListStruct

The structure describes the structure of a general ESS item, that is, a document or a file.

Input:

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

- **$ListItem**: 0 to N structures of key-value type.

3.4.6 Item structure: ItemStruct

The structure describes a general ESS item, that is, a document or a file.

Input:

```
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
}
```

- **$type**: 0 = record, 1 = document.
- **$name**: record name / document annotation.
- **$id_doc**: internal identifier of entry (ID_UFRM).
- **$cj**: assigned reference number.
- **$jid**: assigned UIN.
- **$autor_id**: form user – author ID.
- **$autor_name**: legible author name (Name Surname).
- **$date_created**: Date and time of creating the record/document.
- **$date_filled**: date and time of last change.
- **$spis_id**: internal ID (ID_UFRM) of containing record (if the record or document is placed in a record).

- **$spis_name**: name of containing record (if the record or document is placed in a record).
- **$spis_jid**: UIN of containing record (if the record or document is placed in a record).
- **$archived**: 1 = record / document is archived, 0 = record / document is not archived.

3.4.7 Item structure: CjStruct

Reference number structure.

Input:

```
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 Address structure: AddressStruct

The structure describes a general address that can be used in a FSS web service, mostly to identify an originator of an incoming message or recipient of an outgoing message. Whether the individual elements are mandatory depends on the reception/sending method of the particular message; if the element is mandatory, number 1, 2 or 3 (or their combination) is given, meaning that it is mandatory for sending by paper mail (1), sending electronically by e-mail (2) or using the DMIS system (3).

Input:

```
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
}
```

- **$Type**: 1=DM, 2=traditional post, 3=e-mail.
- **$Type**Adr: 1=representative before company, 2=representative after company.
- **$IdFirma**: Takes precedence over others, the address book is not updated.
- **$IdDS**: data mailbox ID.
- **$Email**: e-mail address of recipient.
- **$Salutation**: salutation.
- **$Name**: recipient name.
- **$SubName**: recipient name expanded.
- **$Representative**: name of company representative.
- **$Street**: street (without numbers).
- **$RedNumber**: land registry number.
- **$BlueNumber**: street number.
- $**ZipCode**: ZIP code.
- **$City**: city name.
- **$State**: country code.
- **$OrgUnitName**: organizational unit name.
- **$OrgUnitNumber**: organizational unit number.
- **$DeliveryMethod**: ID or name of delivery method.

3.4.9 Single attachment structure: DocumentStruct

A structure describing a document. The document may be a link to SFLS or data inserted directly in base64 format.

Input:

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

- **$Description**: document description.
- **$Data**: either APX link or data. Always in base64 format.
- **$Filename**: file name.
- **$MimeType**: Mime type according to FFS usage, e.g. “pdf”.

  $InternalId: internal link to document (SFLF) in the records management WS, this element may contain XG_SFLF.ID_SFLF or XG_SFLF.ID. If this element is specified, it takes precedence over the “Data” element in the same structure. The ESS WS implements the support for the following ASiC forms: MSG_IN, MSG_OUT and SPIS.

3.4.10 More attachments structure: Document

A structure describing a document. The document may be a link to SFLS or data inserted directly in base64 format.

Input:

```
class Document{
  public $DocumentList; // DocumentList, mandatory
  public $NumPhysicalSheets; // int
  public $NumAttachementSheets; // int
  public $Category; // int
}
```

- **$Document**List: Document attachments (components).
- **$NumPhysicalSheets**: Number of physical sheets of the document.
- **$NumAttachementSheets**: Number of physical sheets of attachments.
- **$Category**: Document category number.

3.5 List of reference number items: CjList

Input:

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

- **$OneCj**: a single reference number.

3.6 List of attachments: DocumentList

A structure describing a document. The document may be a link to SFLS or data inserted directly in base64 format.

Input:

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

- **$Document**: List of attachments.

3.7 Inserting a document in a record – ESSAddDocumentToSpis

This method is used to insert a document in a record.

Input:

```
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
}
```

- **$profileID**: specifies the profile used by this WS (ws_profile.php).
- **$IdSubject**: entity ID entered using WS request, otherwise it is taken from ws_profile (id_serv_subj).
- **$IdUserTo**: ID of user who inserts the document in the record.
- **$UserEmail**: e-mail of user who inserts the document in the record.

- **$FileId**: filing number of the record where to place the document.
- **$IdUfrmSpis**: id_ufrm of the record where to place the document.
- **$Jid**: assigned UIN.
- **$Cj**: ref.no. of the document.
- **IdDoc: ID_UFRM** of the document to be inserted.

Output:

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

- **$resultText**Code: Response status number, see the list of response statuses.
- **$ErrMsg**: description if there is an error.
- **$IdDoc**: ID_UFRM of the document placed in the record.
- **$IdUfrmSpis**: ID_UFRM of the record where the document was placed.

3.7.1 Incoming document ESSCreateMsgIn

The structure describes the request to create an incoming document.

Input:

```
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 $SenderIdent; // 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
}
```

- **$profileID**: entity ID entered using WS request, otherwise it is taken from ws_profile.php.
- **$IdSubject**: old element – entity ID entered using WS request, otherwise it is taken from ws_profile (id_serv_subj).
- **$IdSubjekt**: old element – entity ID entered using WS request, otherwise it is taken from ws_profile (id_serv_subj).
- **$IdFmve**: ID_FMVE of MSG_IN form template, otherwise it is taken from ws_profile (MSG_IN_id_fmve).
- **$IdUserTo**: user ID of form author.
- **$UserEmail**: e-mail address of form author.
- **$DocumentName**: name of MsgIn document.
- **$MailBody**: e-mail message body.

  = 1: MSG_IN goes to the archive, if all ESS requirements are met. Type of incoming message is determined based on Sender.Type. WS supports the following values: 2 = traditional mail and 3 = e-mail message.

- **$FinishingMethod**: processing method ID or name.
- **$ClassificationCode**: file plan code.
- **$ShreddingMark**: retention code.
- **$ShreddingPeriod**: retention period.
- **$ShreddingMode**: retention policy name.
- **$PhysicalArchive**: physical location of the document.

- **$ExpeditionNote**: note.
- **$SenderRefNumber**: sender's reference number SenderRefNumber.
- **$SenderIdent**: sender's filing numberSenderIdent.
- **$RecipientRefNumber**: recipient's reference number RecipientRefNumber.
- **$RecipientIdent**: recipient's filing number RecipientIdent.
- **$SenderCj**: old element – sender's reference number SenderRefNumber.
- **$SenderSpisZnak**: old element – sender's filing number SenderIdent.
- **$RecipientCj**: old element – recipient's reference number RecipientRefNumber.
- **$RecipientSpisZnak**: old element – recipient's filing number RecipientIdent.
- **$IdCategory**: category ID DZHP_MSG_TYP.ID_TYP.
- **$Category**: category name DZHP_MSG_TYP.MSG_TYP.
- **$companyType**: category name DZHP_MSG_TYP.MSG_TYP.
- **$DeliveryDate**: delivery date.
- **$DeliveryTime**: delivery time.
- **$DeliveryMethod**: ID or name of delivery method.
- **$SerialNumber**: posting number.
- **$PostOfOrigin**: post office of posting.
- **$NumPhysicalSheets**: Number of physical sheets of the document.
- **$NumAttachementSheets**: Number of physical sheets of attachments.
- **$IdFirma**: sender ID = DZHP_FIRMA.ID_FIRMA takes precedence over the Sender structure, therefore address book details cannot be edited.
- **$FileId**: filing number of the record where to place the document.
- **$Sender**: sender address.
- **$DocumentList**: document attachments (components).
- **$Documents**: original set of attachments – do not use anymore.
- **$PostOfOrigin**: post office of posting.
- **$PluginData**: PluginDataStruct.

Output:

```
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
}
```

- **$resultTextCode**: Response status number, see the list of response statuses.
- **$ErrMsg**: description if there is an error.
- **$Jid**: Assigned UIN.
- **$Cj**: Assigned reference number.
- **$RefNumber**: assigned reference number.
- **$DocIdentifier**: ID_UFRM of the created MSG_IN.
- **$IdUfrm**: ID_UFRM of the created MSG_IN.
- **$IdUfrmSpis**: ID_UFRM of the record where the newly created document was placed.
- **$InfoMsg**: info, if failed to insert the document in a record.
- **$Temp**: to display outWsParam.
- **$PluginData**: PluginDataStruct.

3.7.2 Outgoing document – ESSCreateMsgOut

The structure describes a request to create an outgoing document.

Input:

```
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 $SenderIdent; // 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}
```

- **$profileID**: Specifies the profile used by this WS.
- **$IdSubject**: entity ID entered using WS request, otherwise it is taken from ws_profile (id_serv_subj).
- **$IdSubjekt**: old element – subject ID entered using WS request, otherwise it is taken from ws_profile (**id_serv_subj**).
- **$IdFmve**: ID_FMVE of MSG_IN form template, otherwise it is taken from ws_profile (MSG_IN_id_fmve).
- $IdUserTo: user ID of form author.
- $UserEmail: e-mail address of form author.
- **$DocumentName**: MsgOut document name (annotation).
- $MailBody: E-mail message body.
- **$Type**: element in **AddressStruct DZHP_DENIK_ADR.TYP_MSG 1=DM, 2=traditional, 3=e-mail.**
- $SendType = 1: MSG_OUT goes to the outgoing queue to be sent, if all ESS requirements are met.
- $**ClassificationCode**: file plan code.
- **$ShreddingMark**: retention code.
- **$ShreddingPeriod**: retention period.
- **$ShreddingMode**: retention policy name.
- **$PhysicalArchive**: physical location of the document.
- **$SenderRefNumber**: sender's reference number SenderRefNumber.
- **$SenderIdent**: sender's filing numberSenderIdent.
- **$RecipientRefNumber**: recipient's reference number RecipientRefNumber.
- **$RecipientIdent**: recipient's filing number RecipientIdent.
- **$SenderCj**: old element – sender's reference number SenderRefNumber.
- **$SenderSpisZnak**: old element – sender's filing number SenderIdent.
- **$RecipientCj**: old element – recipient's reference number RecipientRefNumber.
- $**RecipientSpisZnak**: old element – recipient's filing number RecipientIdent.
- **$IdCategory**: category ID DZHP_MSG_TYP.ID_TYP.
- **$Category**: category name DZHP_MSG_TYP.MSG_TYP.
- **$companyType**: category name DZHP_MSG_TYP.MSG_TYP.
- **$NumPhysicalSheets**: Number of physical sheets of the document.

- **$NumAttachementSheets**: Number of physical sheets of attachments.
- **$FileId**: filing number of the record where to place the MSG_OUT document.
- **$Recipient**: single recipient.
- **$Recipients**: more than one recipient.
- **$DocumentList**: document attachments (components).
- **$Documents**: original set of attachments – do not use anymore!
- **$PluginData**: PluginDataStruct.

Output:

```
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
}
```

- **$resultTextCode**: Response status number, see the list of response statuses.
- **$ErrMsg**: description if there is an error.
- **$Jid**: assigned UIN.
- **$Cj**: assigned reference number.
- **$RefNumber**: assigned reference number.
- **$DocIdentifier**: ID_UFRM of the created MSG_IN.
- **$IdUfrm**: ID_UFRM of the created MSG_IN.
- **$IdUfrmSpis**: ID_UFRM of the record where the newly created document was placed.
- **$InfoMsg**: info, if failed to insert the document in a record.
- **$Temp**: to display outWsParam.
- $**PluginData**: PluginDataStruct.

3.7.3 Create a record – ESSCreateSpis

Input:

```
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
}
```

- **$profileID**: specifies the profile used by this WS.
- **$IdSubject**: entity ID entered using WS request, otherwise it is taken from ws_profile (id_serv_subj).
- **$FileType**: record type; not in use yet.
- **$IdUserTo**: ID of user to whom the record will be assigned.
- **$UserEmail**: e-mail of user to whom the record will be assigned.
- **$FileName**: record name.
- **$FileDescription**: record description.
- **$CjId**: CJ_ID of numbering series for the filing number.
- **$CjName**: name of numbering series for the filing number.
- **$FinishingMethod**: processing method ID or name.
- **$ClassificationCode**: file plan code.
- **$ShreddingMark**: retention code.
- **$ShreddingPeriod**: retention period.
- **$IdFirma**: DZHP_FIRMA.ID_FIRMA of the related subject. Takes precedence over the address of the related subject.
- $**Subject**: address of the related subject.
- **$DocumentList**: document attachments (components).
- **$PluginData**: PluginDataStruct.

Output:

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

- **$resultText**Code: Response status number, see the list of response statuses.
- **$ErrMsg**: description if there is an error.
- $**RefNumber**: assigned reference number.
- $**DocIdentifier**: ID_UFRM of the created MSG_IN.
- **$IdUfrm**: ID_UFRM of the created MSG_IN.
- **$Temp**: to display outWsParam.
- **$PluginData**: PluginDataStruct.

3.7.4 Get reference number – ESSGetCj

Input:

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

- **$profileID**: specifies the profile used by this WS.
- **$IdSubject**: entity ID entered using WS request, otherwise it is taken from ws_profile (**id_serv_subj**).
- **$idCert**: subject ID.
- **$PluginData**: PluginDataStruct.

Output:

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

- **$resultTextCode**: Response status number, see the list of response statuses.
- **$ErrMsg**: description if there is an error.
- **$Cjs**: list of reference numbers.
- $**PluginData**: PluginDataStruct.

3.7.5 Get list – ESSGetList

Input:

```
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 $SenderIdent; // string
  public $RecipientRefNumber; // string
  public $RecipientIdent; // string
  public $Archived; // int
  public $PluginData; // PluginDataStruct
}
```

- **$profileID**: specifies the profile used by this WS.
- **$IdSubject**: entity ID entered using WS request, otherwise it is taken from ws_profile (`id_serv_subj`).
- **$IdUser**: user ID.
- **$UserEmail**: e-mail of user to whom the record will be assigned.
- **$RequestFor**: `0=records, 1=documents.`
- **$Start**: from 1.
- **$Limit**: number of entries in the response.
- **$Jid**: assigned UIN.
- **$Name**: name.
- **$CreatedDateFrom**: date of creation from.
- $**CreatedDateTo**: date of creation to.
- **$ModificationDateFrom**: date of modification from .
- **$ModificationDateTo**: date of modification to.
- **$DeliveryDateFrom**: date of delivery from.
- **$DeliveryDateTo**: date of delivery to.
- **$PodaciCislo**: posting number.
- **$PodaciPosta**: post office of posting.
- **$DeliveryMethod**: delivery method ID.
- **$IdFirma**: `DZHP_FIRMA.ID_FIRMA` Takes precedence over others; the address book is not updated.
- **$ClassificationCode**: file plan code.
- **$ShreddingMark**: retention code.
- **$ShreddingPeriod**: retention period.
- **$SenderRefNumber**: sender's reference number SenderRefNumber.
- **$SenderIdent**: sender's filing number`SenderIdent`.
- $**RecipientRefNumber**: recipient's reference number RecipientRefNumber`.`
- **$RecipientIdent**: recipient's filing number `RecipientIdent`.
- **$Archived**: 0=in process, 1=archived, none=in process + archived`.`
- **$PluginData**: PluginDataStruct.

Output:

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

- **$resultText**Code: Response status number, see the list of response statuses.
- **$ErrMsg**: description if there is an error.
- **$Items**: ItemListStruct.
- **$PluginData**: PluginDataStruct.

3.7.6 ESSSetCategory

Input:

```
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
}
```

- **$profileID**: entity ID entered using WS request, otherwise it is taken from ws_profile.php.
- **$IdSubject**: old element – subject ID entered using WS request, otherwise it is taken from ws_profile (id_serv_subj).
- **$IdUserTo**: user ID of the user who changes the category.
- **$UserEmail**: E-mail address of the user who changes the category.
- **$IdCategory**: category ID.
- **$Category**: category name.
- **$IdDoc**: ID_UFRM of the document.
- **$Jid**: assigned UIN.
- **$Cj**: assigned reference number.

Output:

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

- **$resultText**Code: Response status number, see the list of response statuses.
- **$ErrMsg**: description if there is an error.

3.7.7 Getting a specific ESS document – ESSGetDocument

Input:

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

- **$profileID**: WS configuration ID (config_ws_profile), which can contain an entity ID and user ID
- **$IdSubject**: entity ID entered using WS request, otherwise it is taken from config_ws_profile (id_serv_subj).
- **$IdUserTo**: ID of the user who is downloading the document. The user must be authorized to get the document.
- **$UserEmail**: E-mail of the user who is downloading the document. The user must be authorized to get the document.
- **$IdDoc**: ID_UFRM of the document.
- **$Jid**: Document UIN.
- **$Cj**: Document reference number.
  IdDoc or Jid or Cj must be specified

Output:

```
class ESSGetDocumentResponse{
  public $Result;  // Output, 0 = O.K., anything else is an error
  public $ErrMsg;  // Error description
  public $InfoMsg; // Information
  public $id_cert; // entity ID
  public $id_user; // ID_USER of the user who is processing or has processed the
document
  public $id_ufrm; // ID_UFRM of the document found
  public $Type; // Type of the document found (1=incoming, 2=outgoing)
  public $DocumentName; // Name of the document found
  public $Cj; // Reference number of the document found
  public $Jid; // UIN of the document found
  public $CreateTimestamp; // Creation date and time for the document found
  public $ModificationTimestamp; // Date and time for the last modification of t
he document
  public $Category; // Category of the document found
  public $FinishingMethod; Finishing method for the document found
  public $ExecutionMethod; // Trigger event description for the document found
  public $SpisZnak; // File plan code for the document found
  public $SkartZnak; // Retention code for the document found
  public $SkartLhuta; // Retention period for the document found
  public $NumPhysicalSheets; // Number of sheets for the document found
  public $NumAttachementSheets; // Number of attachment sheets for the document
found
  public $PhysicalArchive; // Physical location of the document found
  public $SenderRefNumber; // Sender's reference number
  public $SenderIdent; // Sender's filing number
  public $RecipientRefNumber; // Recipient's reference number
  public $RecipientIdent; // Recipient's filing number
  public $ToHands; // * To the attention of
  public $PersonalDelivery; // To the addressee only
  public $AllowSubstDelivery; // Fiction of delivery enabled
  public $DocumentData; // Whole ZFO forms
  public $SenderRecipientList; // Document senders/recipients - structure
  public $InSpis; // Records in which the found document is placed - structure
  public $CommentList; // List of comments to the document found - structure
  public $DocumentList; // Document attachments - structure
}

Structure $SenderRecipientList - Document senders/recipients
class SenderRecipientList {
  public $SenderRecipient;  // Single sender/recipient
}
class SenderRecipient {
  public $DeliveryTimestamp; // Delivery date and time
  public $SerialNumber;  // Posting number
  public $PostOfOrigin; // Post office of posting
  public $TypeMsg; // Message type (1=data message, 2=paper mail, 3=e-mail)
  public $DeliveryMethod; // Delivery method
  public $IdDS; // Sender/recipient data mailbox ID
  public $Email; // Sender/recipient e-mail address
  public $IdFirma; // Sender/recipient address book ID
  public $IdKontakt;  // Sender/recipient address book contact ID
  public $AddrName; // Sender/recipient name
  public $AddrAddress; // Sender/recipient address
}

Structure InSpis - Records in which the found document is placed
class InSpis {
  public $OneSpis;  // Single record
}
Structure $OneSpis - single record
class OneSpis {
  public $id_ufrm_spis;  // Record DZHP_SPIS.ID_UFRM
  public $spis_name;  // Record name
  public $spis_znak; // File plan code
  public $skart_znak;  // Retention code
  public $spisova_znacka; // File number
  public $id_firma;  // Related entity ID in the address book
  public $adr_name;  // Related entity name
  public $adr_address; // Related entity address
}

Structure $CommentList - List of comments for the document found
class CommentList {
  public $CommentItem; // Single comment - structure
}
Structure $CommentItem - Single comment
class CommentItem {
  public $Comment; //Single comment
  public $DocumentList; // List of attachments of a single comment - structure
}

Structure $DocumentList - List of files inserted in documents or comments
```

```
class DocumentList {
  public $Document; // Single attached file - structure
}
Structure $Document - Single file
class Document {
  public $Description;  // File description
  public $Filename;  // File name
  public $MimeType; // File mime type
  public $Data;  // Binary data of the file
  public $InternalId; // ID or ID_SFLF from the XG_SFLF table
  public $Ltd; // LTD - specifies whether to long-term preserve the document -
values of 0,1,NULL
}
```

- **$resultText**Code: Response status number, see the list of response statuses.

3.7.8 Checking the status of a sent data message – ESSCheckSend

Input:

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

- **$DocIdentifier or IdUfrm**: ID_UFRM of the sent MSG_OUT.

Output:

```
class ESSCheckSendResponse{
  public $ResultCode; // < 0 an error, described in $ErrMsg
                         1 = the document is in progress
                         2 = the document is sent to all recipients
                         3 = the document is delivered to all recipients
                         4 = the document is received by all recipients
  public $ErrMsg; // Error description
  public $InfoMsg; // Information
}
```

4 Handling LTV documents

4.1 Service602LTV service, computeHashDocument

This method is used to get a hash calculation for a single document based on its **binary contents**.

Input:

```
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. */
}
```

- **$input**: binary content in base64.
- **$externalId**: optionally, the binary file will be downloaded and saved using sflf external_id.
- **$url**: relative endpoint in mode S or full url in mode C.
- **$hashType**: default value "SHA-256" else "SHA-512" or "MD5".
- **$system**: system identification, requires a value filled from the TIF header: RQ_APP_ID.
- **$requestId**: system identification, requires a value filled from TIF header: OPERATION_ID.
- **$businessRequestId**: businessRequestId request identification (not used for TIF).

Output:

```
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 */
}
```

- **$resultCode**: status code, 0 is OK. The code list is given in \FS\XG\WS\Service602LTV\SvcStatus.php.
- **$errorCode**: numerical error or warning code.
- **$errorDescription**: error or warning description.
- **$requestId**: system identification, requires a value filled from the TIF header: OPERATION_ID.
- **$businessRequestId**: marks the correlation id, used by the caller.

4.2 Service602LTV service, LtvAnalyze

This method returns in its response the file converted to PDF. You can specify in the request whether the PDF is to be digitally signed and whether to visualize the signature or not.

Input:

```
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: only all valid
qualified signatures are accepted. https://gitlab.602.cz/sofa/sofa/-/issues/735
*/
  public $lang; /* @var string|null cs-CZ, en-US, ...language identification -
planned, not in use yet. */
  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 analys
is for the purpose of evidence of possible added time stamp. UUID variant.*/
}
```

- **$request**: technical parameters– filled in from the TIF header.
- **$input**: binary content in base64.
- **$externalId**: optionally, the binary file will be downloaded and saved using sflf external_id.
- **$fileType**: PDF or XML.
- **$bAllowUnknownCertificateAuthority**: allows an unknown certification authority (CRL lists are not checked, only the signature and document integrity).
- **$revocationChecking**: whether to check CRL, lists the last signature in the signature tree only (EndCertificateOnly) and leaves out the root certificate (ExcludeRoot) without CRL check (None). By default, all options are checked [ExcludeRoot;EndCertificateOnly;None].
- **$bReturnValidationMaterial**: whether the output returns CRLs, the default is not returning them.
- **$visibleSignChecking**: if set to true, the PKI checks whether all visible characters are placed in the allowed area defined in vsPosX, vsPosY, vsSizeX, vsSizeY, vsPage.
- **$vsPosX**: X position coordinate.
- **$vsPosY**: Y position coordinate.
- **$vsSizeX**: X width.
- **$vsSizeY**: Y width.
- **$vsPage**: page number where to place the image.
- **$validationMode:** if set to 1, globalStatus will be only valid if all signatures are valid. If there is no end time stamp, and the document is presumed valid, the process adds a time stamp.
- **$lang**: language of returned messages. Planned feature, not in use yet.
- **$idLtdConf**: which tenant (configuration) will be used for analysis to register possible added time stamp.
- **$guidLtdConf**: which tenant (configuration) will be used for analysis to register possible added time stamp (UUID variant).

Output:

```
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 new binary, created for $validation
Mode = 1. - Pdf documents without a time stamp at the end must be stamped first
with GL735 */
  public $globalStatusMessage; /* @var string|null Short txt description of cert
ificate validation */
}
```

- **$result**: 0=OK, 4=warning, 8=error, C=critical [values 0;4;8;C].
- **$docType**: document type.
- **$expiration**: document expiry date, relates to all signatures and time stamps.
- **$docDataHash**: document hash.
- **$docDataHashAlg**: algorithm hash.
- **$currSignHashAlg**: hash of last signature or time stamp algorithm.
- **$globalStatus:** global validity status for the document. WARNING means a signature currently valid but not validated after the given period or the time stamp is missing. [results: OK; WARNING; ERROR; NO_SIGNATURE].
- **$sigInfos:** signature information array.
- **$docMimeType**: replacing docType with a full mime type string.
- **$newExternalId**: see input validationMode=1.
- **$globalStatusMessage**: short txt certificate validation result summary. Filled, if validationMode = 1.

4.3 Service602LTV service, LtvDeleteTenant

Non-public ws.

Input:

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

Output:

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

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

4.4 Service602LTV service, ltvGetDocumentInfo

This method is used to get the details of a specific document placed in long-term preservation.

Input:

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

- **$tenant**:
  - **$idLtdConf**: which tenant (configuration) will be used for analysis to register possible added time stamp (UUID variant).
  - **$guidLtdConf**: which tenant (configuration) will be used for analysis to register possible added time stamp (UUID variant).
- **$externalId**: optionally, the binary file will be downloaded/saved using sflf external_id.

Output:

```
class LtvGetDocumentInfoResponse
{
  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 */
}
```

- **$errorCode**: numerical error or warning code.
- **$errorDescription**: error or warning description.
- **$externalFileId**: file identifier in an external system.
- **$externalBlobId**: information on file in an external system saved in Azure Blob Storage.
- **$state**: overall document status.
- **$expirationDate:** expiry date.

4.5 Service602LTV service, LtvInsert

Service that provides document registration in long-term preservation.

Input:

```
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.*/
}
```

- **$request**: technical parameters– filled in from the TIF header.
- **$input**: binary content in base64.
- **$externalId**: optionally, the binary file will be downloaded/saved using sflf external_id.
- **$fileType**: PDF or XML.
- **$idLtdConf**: which tenant (configuration) will be used for analysis to register possible added time stamp.
- **$fileName**: name of input file.
- **$guidLtdConf**: which tenant (configuration) will be used for analysis to register possible added time stamp (UUID variant).
- **$ruleException**: allows preservation of documents even if some of their signatures are invalid.
- **$baMode**: N/A or 0 means classic insertion to DDA.
  1 – only validates the time stamp once or registers, to prevent expiry. Suitable for documents in progress.

  2 - fully archived in the charged archive.

Output:

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

- **$result**: technical parameters mapped from the TIF header.
- **$resultCode**: status code, 0=OK, 4=warning, 8=error, C=critical [0;4;8;C].

4.6 Service602LTV service, LtvRegisterTenant

Non-public ws. This method is used to register a document in the SecuSign database.

Input:

```
class LtvRegisterTenant
{
  public $tenantName; /* @var string $tenantName Company name
  public $tenantIdent; /* @var string $tenantIdent UUID identifier 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 - */
}
```

Output:

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

4.7 Service602LTV service, LtvUnregister

This method is used to unregister a long-term preserved document from the SecuSign database.

Input:

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

- **$tenant**:
    - **$idLtdConf**: which tenant (configuration) will be used for analysis to register possible added time stamp (UUID variant).
    - **$guidLtdConf**: which tenant (configuration) will be used for analysis to register possible added time stamp (UUID variant).
- **$externalId**: optionally, the binary file will be downloaded/saved using sflf external_id.

Output:

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

- **$errorCode**: numerical error or warning code.
- **$errorDescription**: error or warning description.

4.8 Service602LTV service, SignDocument

This method is used to apply a qualified certificate to the input document.

Input:

```
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 */
}
```

- **$request**: technical parameters – filled in from the TIF header.
- **$input**: binary content in base64.
- **$externalId**: optionally, the binary file will be downloaded and saved using sflf external_id.
- **$id_docu:** document identifier.
- **$url:** relative endpoint in mode S or fullurl in mode C.
- **$uri**: alternative file source.
- **$fileType**: PDF or XML.

- **$addTimestamp**: False / True, set if you need to apply a time stamp along with the signature.
- **$signatureType**: may contain the following values: "DEFAULT" [DEFAULT; ENVELOPED; DETACHED; ENCAPSULATE].
- **$signatureID**: ID of certificate to be used for signing.
- **$userData**: user data to be used in signature detail.
- **$userDataVisible**: user data to be used in signature visualization.
- **$visibleSign**: True=Visual signature, False or empty value = standard signature.
- **$signNumber**: signature number (expecting 1 to 5). When filled in, the following parameters from the configuration will be used: signReason, signlocation, signComment, signImage, signPosX, signPosY, signSizeX, signSizeY.
- **$signReason**: reason for signing.
- **$signLocation**: signed at …..
- **$signComment**: signature comment.
- **$signImage**: image used for visualized signature.
- **$signPosX**: X position from the upper left corner for the visualized signature area.
- **$signPosY**: Y position from the upper left corner for the visualized signature area.
- **$signSizeX**: visualized signature object X size in millimeters.
- **$signSizeY**: visualized signature object Y size in millimeters.
- **$signPage**: page number where to place the signature; MAXINT value = last page.
- **$customValues**: custom values.

Output:

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

- **$result**: 0=OK, 4=warning, 8=error, C=critical [values 0;4;8;C].
  - **$errorCode**: numerical error or warning code.
  - **$errorDescription**: error or warning description.

4.9 Service602LTV service, TimestampPdfDocument

This method is used to seal the input document with a qualified seal.

Input:

```
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. */
}
```

- **$request**: technical parameters– filled in from the TIF header.
- **$input**: binary content in base64.
- **$externalId**: optionally, the binary file will be downloaded and saved using sflf external_id.
- **$id_docu:** document identifier.
- **$uri**: the URI parameter is passed on to a plugin that downloads the file from an external source and returns the file.

Output:

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

- **$result**: 0=OK, 4=warning, 8=error, C=critical [values 0;4;8;C].
- **$input:** binary content in base64.
- **$uri**: alternative time stamp URI from an external source.

5 WS return statuses

5.1 Successfully processed

Error code	Error text	Meaning
0	" "	The WS call completed without errors

5.2 Range of 100–199 – configuration and call errors

Error code	Error text	Meaning
100		Unspecified configuration error
101		The profile specified in the WS call is not configured on the server, unknown profile
102		The profile definition is incomplete, does not contain the form ID_FMVE
103		The profile definition is incomplete, does not contain subject identification
104		The profile definition is incomplete, does not contain SFLF / DMS definition
105		The WS is not authorized
106		SFLF error

5.3 Range of 200–299 - runtime errors

They are usually caused by incorrect service call.

Error code	Error text	Meaning
200		Unspecified runtime error
201		Failed to download the form template to be filled out
202		The subject specified has not been found and cannot be used
203		There is no default file plan code set for the subject
204		There is no default processing method set for the subject
205		Authentication error, invalid user
206		Failed to assign ref.no./UIN
207		The filing number is not unique
208		Failed to assign filing number
209		The service call is incomplete, required data is missing, most commonly counterparty identification or address
210		Failed to upload files to SFLF
211		The service call does not contain any attachments
212		The service call does not contain the name (annotation) of attachments
213		The metadata for the attached files is incomplete
214		Failed to insert a document in a record
215		Failed to find the record where to insert
216		Record not found
217		Document not found
218		Failed to determine document type
219		The “from” or “to” limit does not meet requirements. Typically from > to, from not specified, etc.

220		You are requesting too many results (the limit is 1000 for a single WS call)
221		Service runtime error
222		The category ID specified is invalid, the category does not exist or is disabled
223		The category name specified is invalid, the category does not exist or is disabled
224		The user is not authorized to use this object
225		User not found
226		Form template not found
227		The user is not authorized to use the form template
228		The reference number has not been found
229		The user is not authorized to use this reference number ID
230		Error when creating a reference number
231		The user is not authorized to use this document
232		Error when creating a UIN
233		Invalid data
234		The document is already closed
235		The XML data is empty

5.4 Range of 300–399 – database errors

These statuses help with debugging and initial settings. They should not appear with correct use.

Error code	Error text	Meaning
300, 399		Unspecified database error
301		Incorrect manipulation with XG_UFRM table
302		Incorrect manipulation with XG_UFSS table
303		Incorrect manipulation with XG_UFDF table
304		Error when creating a record in DZHP_SPIS table
305		Error when updating a record in DZHP_SPIS table
306		Error when creating a journal entry
307		Error when creating an address book entry
308		Error when updating a record in the journal
309		Error when updating a record in the address book
310		Error when looking up id ufrm
311		Error when loading the list of document categories
399		Database error