

# **Integrační příručka**

## **D.Sig XAdES Extender Java, v2.0**

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

# Copyright

Všetky práva vyhradené

Tento dokument je vlastníctvom spoločnosti DITEC, a. s. Žiadna jeho časť sa nesmie akýmkoľvek spôsobom (elektronickým, mechanickým) poskytnúť tretej strane, rozmnožovať, kopírovať, vrátane spätného prevodu do elektronickej podoby, bez písomného povolenia spracovávateľa.

## Popisné charakteristiky dokumentu

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Podnázov	D.Sig XAdES Extender Java, v2.0	
Ref. číslo	GOV_ZEP.213	Verzia 8

Vypracoval	Mikuš Michal	Podpis	Dátum 10. 8. 2023
Preveril	Priezvisko Meno	Podpis	Dátum xx.xx.201x
Schválil	Priezvisko Meno	Podpis	Dátum xx.xx.201x

Formulár	Dokument		
Ref. číslo	Fo 11	Dátum poslednej aktualizácie	Dátum 03.01.2013

## Akceptované dňa : <Dátum akceptácie>

Za <Objednávateľa>:

Za <Dodávateľa>.::

\_\_\_\_\_  
<Meno zodpovednej osoby>

\_\_\_\_\_  
<Meno zodpovednej osoby >

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

### Záznamy o zmenách

Autor	Popis zmien	Dátum	Verzia
M.Mikuš	Súlad so špecifikáciou v.7. Zmena názvu parametrov class a oiClass v MC. XML a Json výstup GetObjectInfo v MC.	9.1.2017	4
M.Mikuš	Doplnenie popisu metódy konštruktora.	9.8.2023	9

### Pripomienkovanie a kontrola

Autor	Stanovisko	Dátum	Verzia

### Rozdeľovník

	Priezvisko Meno	Firma, Funkcia
Originál		
Kópia		
Kópia		
Kópia		

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

# Obsah

<b>1.</b>	<b>Úvod .....</b>	<b>6</b>
<b>2.</b>	<b>Použité zdroje .....</b>	<b>7</b>
<b>3.</b>	<b>Systémové požiadavky, distribúcia a inštalácia .....</b>	<b>8</b>
3.1.	Systémové požiadavky.....	8
3.2.	Distribúcia a inštalácia .....	9
<b>4.</b>	<b>Architektúra .....</b>	<b>11</b>
4.1.	Postavenie v rámci nadradenej aplikácie .....	11
4.2.	Vnútoraná architektúra.....	11
<b>5.</b>	<b>Špecifikácia API.....</b>	<b>12</b>
5.1.	Princípy integračného rozhrania Java API.....	13
5.2.	Princípy integračného rozhrania Java applet API .....	13
5.2.1.	Detekcia pripravenosti appletu .....	13
5.2.2.	Spracovanie dlhých reťazcov .....	13
5.2.3.	Notifikácia o ukončení Java applet API funkcie pomocou callback.	13
5.3.	<b>Extender .....</b>	<b>14</b>
5.3.1.	Java API XAdESExtender .....	14
5.3.2.	Applet API XAdESExtenderAppletWrapper .....	16
5.3.3.	Triedy použité ako výstupné hodnoty metód .....	18
5.4.	<b>MessageContainer .....</b>	<b>23</b>
5.4.1.	Java API MessageContainer .....	23
5.4.2.	Applet API MessageContainerAppletWrapper .....	24
5.4.3.	Triedy použité ako výstupné hodnoty metód .....	26
5.5.	<b>Popis metód triedy Extender .....</b>	<b>27</b>
5.5.1.	Popis spoločných metód.....	27
5.5.1.1.	konštruktor.....	27
5.5.1.2.	metóda getErrorMessage .....	27
5.5.1.3.	metóda getDataSignatures.....	27
5.5.1.4.	metóda getDocumentUnauthorized .....	28
5.5.1.5.	metóda getRegistration .....	28
5.5.1.6.	metóda getRegistrationBase64 .....	28
5.5.1.7.	metóda getDocumentCount.....	28
5.5.1.8.	metóda moveToDocument .....	28
5.5.1.9.	metóda getDocumentType .....	29
5.5.1.10.	metóda initialize .....	29

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

5.5.1.11.	metóda openFile .....	29
5.5.2.	Metódy zloženého elektronického podpisu .....	30
5.5.2.1.	metóda createNewDataSignatures.....	30
5.5.2.2.	metóda addDataEnvelopeToExistingDataSignatures.....	30
5.5.2.3.	metóda insertDataSignatures .....	31
5.5.2.4.	metóda getDataSignaturesInfo.....	31
5.5.2.5.	metóda getDataSignaturesInfo.....	32
5.5.2.6.	metóda verifyDataSignatures .....	32
5.5.3.	Dokument bez autorizácie .....	33
5.5.3.1.	createNewDocumentUnauthorized.....	33
5.5.3.2.	metóda getDocumentUnauthorizedInfo .....	33
5.5.3.3.	metóda getDocumentUnauthorizedInfo .....	34
5.5.4.	Podanie .....	34
5.5.4.1.	metóda createNewRegistration .....	34
5.5.4.2.	metóda getRegistrationInfo .....	34
5.5.4.3.	metóda getRegistrationInfo .....	35
<b>5.6.</b>	<b>Popis metód triedy MessageContainer .....</b>	<b>35</b>
5.6.1.	konštruktor.....	35
5.6.2.	metóda initialize .....	36
5.6.3.	metóda getErrorMessage .....	36
5.6.4.	metóda addXMLObject.....	36
5.6.5.	metóda addBase64Object .....	37
5.6.6.	metóda getMessageContainer.....	37
5.6.7.	metóda initialize .....	37
5.6.8.	metóda isInitialized .....	37
5.6.9.	metóda getMessageContainerInfo.....	38
5.6.10.	metóda getMessageContainerInfo.....	38
5.6.11.	metóda getObjectCount.....	38
5.6.12.	metóda getObjectInfo .....	39
5.6.13.	metóda getObjectInfo .....	39
5.6.14.	metóda getObjectData.....	39
5.6.15.	metóda getVersion .....	40
<b>6.</b>	<b>Návratové kódy aplikácie .....</b>	<b>41</b>

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

# 1. Úvod

Tento dokument je určený pre používateľov softvérového komponentu D.Sig XAdES Extender JAVA a vývojárov aplikácií pre vytváranie a spracovanie zaručených elektronických podpisov formátu XAdES\_ZEP.

Komponent D.Sig XAdES Extender JAVA je určený na

- vytváranie štruktúry zloženého elektronického podpisu [6], neautorizovaného dokumentu [7] a podania [8],
- spracovanie podania a získavanie informácií o tejto štruktúre a o obsiahnutých podpisoch.

Komponent je určený na integráciu do komplexnejších systémov ako pomocná knižnica a neposkytuje užívateľské rozhranie, takže jej popis je zredukovaný na zoznam funkcií, ich vstupno-výstupné charakteristiky (popísané v časti 5) a systémové požiadavky (zhrnuté v časti 3).

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

## 2. Použité zdroje

- [1] Formát jednoduchého elektronického podpisu XAdES\_ZEP, verzia 1.0.  
[http://www.ditec.sk/ep/signature\\_formats/xades\\_zep/v1.0/](http://www.ditec.sk/ep/signature_formats/xades_zep/v1.0/)
- [2] Formát jednoduchého elektronického podpisu XAdES\_ZEP, verzia 1.1.  
[http://www.ditec.sk/ep/signature\\_formats/xades\\_zep/v1.1/](http://www.ditec.sk/ep/signature_formats/xades_zep/v1.1/)
- [3] Formát jednoduchého elektronického podpisu XAdES\_ZEP, verzia 2.0.  
[http://www.ditec.sk/ep/signature\\_formats/xades\\_zep/v2.0/](http://www.ditec.sk/ep/signature_formats/xades_zep/v2.0/)
- [4] Formát zloženého elektronického podpisu XAdES\_ZEP, verzia 1.0.  
[http://www.ditec.sk/ep/signature\\_formats/xades\\_zep\\_data\\_signatures/v1.0/](http://www.ditec.sk/ep/signature_formats/xades_zep_data_signatures/v1.0/)
- [5] Formát zloženého elektronického podpisu XAdES\_ZEP, verzia 1.1.  
[http://www.ditec.sk/ep/signature\\_formats/xades\\_zep\\_data\\_signatures/v1.1/](http://www.ditec.sk/ep/signature_formats/xades_zep_data_signatures/v1.1/)
- [6] Formát zloženého elektronického podpisu XAdES\_ZEP, verzia 2.0.  
[http://www.ditec.sk/ep/signature\\_formats/xades\\_zep\\_data\\_signatures/v2.0/](http://www.ditec.sk/ep/signature_formats/xades_zep_data_signatures/v2.0/)
- [7] XML štruktúra dokumentu bez autorizácie  
<http://www.ditec.sk/ekr/unauthorized/v1.0>
- [8] XML štruktúra podania <http://www.ditec.sk/ekr/registration/v1.0>
- [9] Špecifikácia softvérového komponentu XAdES Extender.  
GOV\_ZEP.161.5.160405.Špecifikácia D.Sig.XAdES.Extender.docx

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

## 3. Systémové požiadavky, distribúcia a inštalácia

### 3.1. Systémové požiadavky

Systémové požiadavky aplikácie D.Sig XAdES Extender Java sú nasledujúce:

- operačný systém – MS Windows 7 / 8 / 10 / 11, Mac OS X: verzia 10.12 – 10.15, 11, 12, procesor (architektúra CPU): x86\_64, arm (M1), prekladač Rosetta 2 – v prípade procesora arm (M1), GNU/Linux: Mint verzia 13, 17.x, 18, 19.x, 20.0, 20.1, 20.2, 20.3; Debian verzia 8, Mint Debian Edition 4, 5; Ubuntu verzia 12.04 LTS, 14.04 LTS, 16.04 LTS, 18.04 LTS, 20.04 LTS, 21.04, 21.10, 22.04; Fedora: verzia 23, 24, 25, 33, 34, 35, 36; Manjaro 21.0, 21.2.2,
- ak je aplikácia D.Sig XAdES Extender Java spúšťaná z web portálu pomocou aplikácie D.Launcher v1.x:
  - ⇒ Oracle Java 8 (<https://www.java.com/en/download/manual.jsp>), pozn. kombinácia OpenJDK a IcedTea nie je podporovaná,
  - ⇒ Java plugin do webového prehliadača, Java Web Start a Java FX verzia 2.1 a vyššia (súčasť inštalácie Oracle Java),
- web prehliadač podporujúci spúšťanie Java appletov<sup>[1]</sup> – MS Internet Explorer v7.0 alebo vyššia (len 32 bit), Mozilla Firefox, v45 – v51, resp. v59 ESR(len 32 bit, s podporou NP API), Safari 14, 15,
- prístup na internet (prípadne správne nastavenia pre proxy).

Ak je aplikácia D.Sig XAdES Extender Java spúšťaná z web portálu pomocou aplikácie D.Launcher v1.x, tak požiadavky na web prehliadač zahŕňajú aj prehliadače:

- MS Internet Explorer verzia 10/11 (aj 64 bit), Mozilla Firefox, v45 a vyššia aj 64-bit, Google Chrome v51 alebo vyššia (prípadne Chromium), Opera v38 alebo vyššia, MS Edge v25 alebo vyššia.

V tomto prípade je Java plugin vyžadovaný pre MS Internet Explorer 7/8/9, voliteľný pre MS Internet Explorer 10/11; môže byť nutné ho v prehliadači MS Internet Explorer povoliť pomocou voľby Tools/Manage add-ons. Systémové požiadavky pre aplikáciu D.Launcher sú špecifikované v rámci používateľskej príručky pre aplikáciu D.Launcher.

Ak je aplikácia D.Sig XAdES Extender Java spúšťaná z web portálu pomocou aplikácie D.Launcher v2.x a rozšírenia D.Bridge 2:

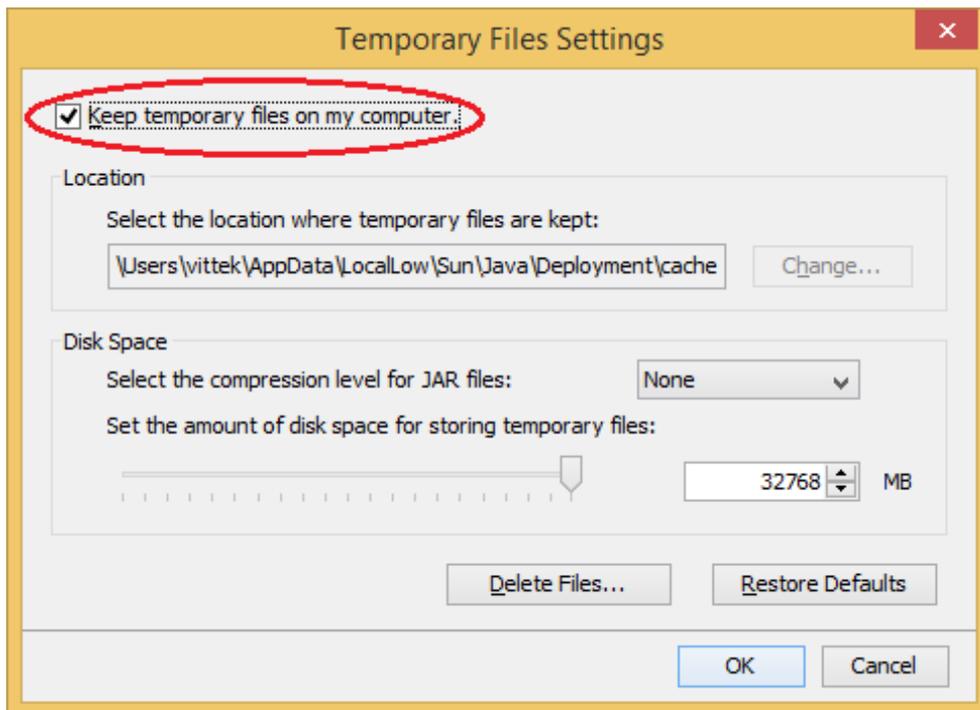
---

<sup>[1]</sup> Ak je aplikácia D.Sig XAdES Extender Java spúšťaná ako Java applet.

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

- webový prehliadač – MS Internet Explorer 11 (len 32bit verzia), Mozilla Firefox 78, 89, 91, 101, Google Chrome 91, 100, 101, Chromium 91, 100, 101, Opera 76, 78, Microsoft Edge 91, 96, 97,
- vo webovom prehliadači nainštalované a povolené rozšírenie D.Bridge 2, pre MS Internet Explorer sa vyžaduje vypnutý chránený režim.

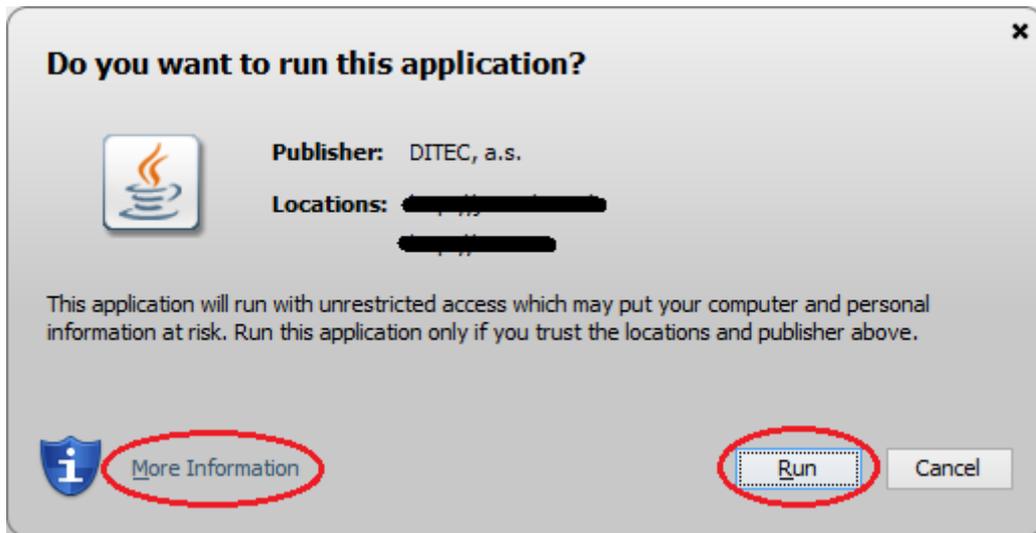
Aplikácia D.Sig XAdES Extender Java vyžaduje, aby bolo v nastaveniach Java povolené ukladanie dočasných súborov. Toto nastavenie je prístupné z Java Control Panel.



## 3.2. Distribúcia a inštalácia

Aplikácia D.Sig XAdES Extender Java môže byť integrovaná ako applet v rámci web aplikácie alebo ako komponent v rámci klientskej Java aplikácie bežiacей v JRE. Ak je distribúcia a inštalácia aplikácie D.Sig XAdES Extender Java na PC používateľa zabezpečená pomocou technológie webstart, tak integritu súborov aplikácie overuje technológia webstart pri spustení aplikácie. Jednotlivé JAR knižnice sú podpísané certifikátom výrobcu aplikácie (spoločnosť Ditec, a.s.) a je na ne vyžadovaná časová pečiatka. Používateľ si môže skontrolovať podrobnosti a platnosť certifikátu výrobcu kliknutím na link "More information" (prekl. Viac informácií) a potvrdiť spustenie aplikácie kliknutím na tlačidlo "Run" (prekl. Spustiť).

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9



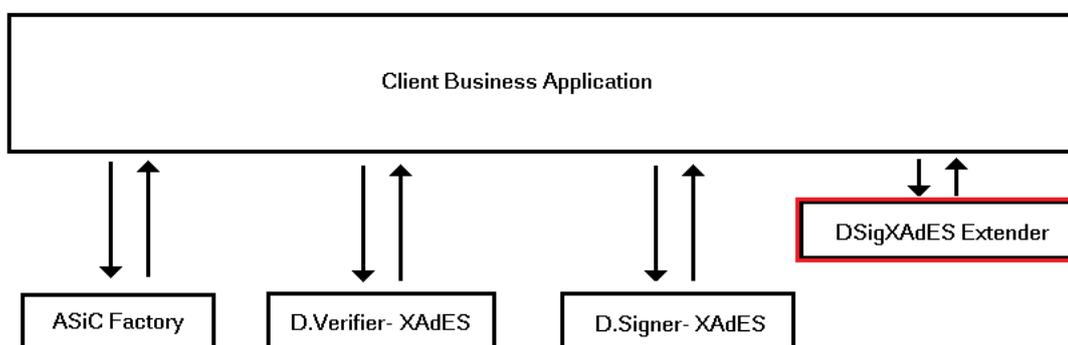
Alternatívnou možnosťou je distribúcia aplikácie D.Sig XAdES Extender Java spolu s klientskou aplikáciou, v rámci ktorej je integrovaná, z dôveryhodného zdroja napr. na CD médiu v rámci inštalačných súborov klientskej aplikácie. V tomto prípade je integrita súborov aplikácie D.Sig XAdES Extender Java zabezpečená samotným spôsobom distribúcie.

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

## 4. Architektúra

### 4.1. Postavenie v rámci nadradenej aplikácie

Tento komponent poskytuje volajúcej aplikácii rozhranie na vytváranie a spracovanie dátových obálok formátu xzep:DataSignatures a štruktúr podania podľa požiadaviek definovaných v špecifikácii [9]. Volajúca aplikácia ku tomu poskytuje všetky potrebné údaje, takže nie je potrebná interakcia so žiadnymi ďalšími modulmi.



Obr. 1: Postavenie komponentu D.Sig XAdES Extender v rámci širšieho systému na vytváranie a spracovanie elektronicke podpísaných dokumentov. Ostatné komponenty (ASiC Factory, D.Signer-XAdES, ...) sú uvedené ako príklad a nie sú potrebné pre fungovanie D.Sig XAdES Extender.

### 4.2. Vnútoraná architektúra

Vzhľadom na oddelené množiny funkčných požiadaviek postačuje, aby komponent bol tvorený jednou triedou pre základnú funkcionalitu (Extender) a jednou triedou pre prácu so štruktúrou MessageContainer.

Metódy týchto tried poskytujú možnosti pre všetky možné scenáre použitia a sú podrobne popísané v nasledujúcej časti.

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

## 5. Špecifikácia API

Funkcionalita aplikácie D.Sig XAdES Extender Java je rozdelená do samostatných modulov, ktoré je možné pri nasadení aplikácie kombinovať podľa požiadaviek zákazníka. Aplikáciu D.Sig XAdES Extender Java bude tvoriť sada JAR knižníc, ktoré budú poskytovať pre klientske aplikácie nasledujúce integračné rozhrania

- Java API – umožňuje volanie služieb komponentu z aplikácií pracujúcich v JRE
- Java Applet API – umožňuje volanie služieb komponentu priamo z prostredia webového prehliadača. Postupuje sa vytvorením appletu `sk.ditec.zep.dsigner.xades.extender.applet.XAdESExtenderApplet`, ktorý obsahuje metódy:
  - ⇒ `XAdESExtenderAppletWrapper createXAdESExtender()`,
  - ⇒ `MessageContainerAppletWrapper createMessageContainer()`
  - ⇒ `ASiCFactoryAppletWrapper createASiCFactory()`.

Inštancie týchto troch tried appletu už poskytujú príslušnú funkcionality popísanú nižšie. Detaily ohľadom využívania Java Appletu sú uvedené v časti 5.2 nižšie.

Ak je pre integráciu aplikácie D.Sig XAdES Extender Java použité Java API, tak pre správne fungovanie logovania je potrebné, aby integrátor poskytol implementáciu SLF4J (<https://www.slf4j.org/>).

### Príklad registrácie callback funkcie appletu `dSigXadesExtenderApplet` v javascripte:

```
var attributes = {id:dSigXadesExtenderApplet,
  code:'sk.ditec.zep.dsigner.xades.extender.applet.XAdESExtenderApplet',
  width:1,
  height:1};
var parameters = {jnlp_href:'dsigextender.jnlp',
  onLoadCallbackName:'dsigextenderOnLoad'};
var version = '1.6';
deployJava.runApplet(attributes, parameters, version);

//.....
function dsigextenderOnLoad(source) {
  alert(source);
}
```

Knižnica D.Sig XAdES Extender neposkytuje GUI a teda oznamovanie výstupov z knižnice je plne na strane volajúcej aplikácie.

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

## 5.1. Princípy integračného rozhrania Java API

## 5.2. Princípy integračného rozhrania Java applet API

### 5.2.1. Detekcia pripravenosti appletu

Integračné rozhranie Java applet API umožňuje volanie služieb komponentu D.Sig XAdES Extender Java priamo z prostredia webového prehliadača. Ak je potrebné v rámci web stránky detegovať pripravenosť appletu, tak je potrebné do objektu window priradiť callback funkciu s jedným parametrom a pri nasadzovaní appletu treba pridať parameter onLoadCallbackName s názvom vytvorenej funkcie:

```
<APPLET ....>
  <PARAM name="onLoadCallbackName" value="<meno funkcie">
</APPLET>
```

Po úspešnej inicializácii applet zavolá túto funkciu s hodnotou svojej inštancie.

### 5.2.2. Spracovanie dlhých reťazcov

Z dôvodu spracovania dlhých reťazcov, applet pri svojej inicializácii vloží do objektu window prototyp objektu ditec.WrappedString, ktorý slúži na obalovanie dlhých (rádovo 1MB) JavaScript reťazcov. V prípade že Applet API predpisuje dátový typ Object, tak skutočný typ môže byť buď JavaScript String (do 1MB) alebo ditec.WrappedString (ľubovoľná dĺžka).

Pre konverziu JavaScript Stringu na ditec.WrappedString je potrebné vytvoriť jeho inštanciu pomocou:

```
var s = new ditec.WrappedString(<hodnota JavaScript retazca>);
```

Pre získanie hodnoty ditec.WrappedString je potrebné zavolať jeho metódu: s.str().

### 5.2.3. Notifikácia o ukončení Java applet API funkcie pomocou callback

Rozhranie jednotlivých objektov je koncipované ako asynchrónne. Ak funkcia vracia návratovú hodnotu, tak funkcia Java applet API je navrhnutá podľa schémy:

```
public boolean funkcia
(
  <parametre zodpovedajúcej funkcie v Java API>
  ,   final JSObject callback
);
```

Posledný parameter musí byť JavaScript objekt reprezentujúci callback funkciu, ktorá je definovaná podľa nasledujúceho vzoru:

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

```

var callback_object = {
    onComplete : function(value, instance) {
        //value [ľubovolny typ]: navratova hodnota volanej metody
        //instance [applet]: instancia apletu, ktory zavolala
callback
    },

    onException : function(msg, stackTrace, instance) {
        //msg [String]: chybova hlaska, ktora vznikla pri volani
metody
        //stackTrace [String]: detail chyby
        //instance [applet]: instancia apletu, ktory zavolala
callback
    }
}

```

Callback funkcia musí byť objekt, ktorý obsahuje metódy `onComplete` a `onException`. V prípade, že daná operácia skončila korektne (teda aj v prípade, ak jej návratový kód reprezentuje chybu), je zavolaná metóda `onComplete`, pričom výsledok operácie je poskytnutý ako parameter `value` tejto metódy. V prípade neošetrenej výnimky je zavolaná metóda `onException` s technickými detailmi o vzniknutej chybe.

Dátový typ `value` zodpovedá JavaScript reprezentácii príslušného typu návratovej hodnoty v Jave. Java dátový typ `java.lang.String` sa vždy mapuje na JavaScript objekt `ditec.WrappedString`, z ktorého je možné získať jeho hodnotu zavolaním metódy `s.str()`.

Pri volaní funkcií Java applet API rozhrania je možné volať vždy len jednu metódu. Kým nie je ukončené jej vykonávanie spätným zavolaním funkcií `onComplete` alebo `onException` callback objektu, nie je možné volať iné funkcie Java applet API rozhrania. V prípade, že toto nastane, funkcia vráti okamžite `false` bez vykonania svojej činnosti, ale už neoznami svoje ukončenie prostredníctvom príslušných funkcií callback objektu.

## 5.3. Extender

### 5.3.1. Java API XAdESExtender

**Package:**

`sk.ditec.zep.dsigner.xades.extender`

**Triedu:**

`XAdESExtender`

**Konštruktor:**

`public XAdESExtender()`

**Metódy:**

`public String getErrorMessage();`

`public String getDataSignatures();`

`public String getDocumentUnauthorized();`

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

```

public String getRegistration();
public String getRegistrationBase64();
public int getDocumentCount();
public int moveToDocument(int index);
public int getDocumentType();
public void initialize();
public String openFile(
    String title,
    String filter,
    int readBinary,
    int type
);
public int createNewDataSignatures(
    String inDataEnvelope,
    String inURI,
    String inID,
    String inDescription
);
public int createNewDataSignatures(
    String inDataEnvelope,
    String inURI,
    String inID,
    String inDescription,
    String dataSignaturesVersion
);
public int addDataEnvelopeToExistingDataSignatures(
    String inDataSignatures,
    String inDataEnvelope
);
public int insertDataSignatures(String inDataSignatures);
public DataEnvelopeInfo getDataEnvelopeInfo(String inDataEnvelope);
public String getDataEnvelopeInfo(
    String inDataEnvelope,
    int type
);
public DataSignaturesInfo getDataSignaturesInfo(String inDataSignatures);
public String getDataSignaturesInfo(
    String inDataSignatures,
    int type
);
public int verifyDataSignatures(String inDataSignatures);
public int createNewDocumentUnauthorized(
    String inURI,
    String inID,
    String inDescription,
    String objectData,
    String inObjectID,
    String inObjectMimeType,
    String inObjectEncoding,

```

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

```

        String inObjectIdentifier
    );
    public DocumentUnauthorizedInfo getDocumentUnauthorizedInfo(
        String inDocumentUnauthorized
    );
    public String getDocumentUnauthorizedInfo(
        String inDocumentUnauthorized,
        int type
    );
    public int createNewRegistration(
        String inURI,
        String inID,
        String inDescription,
        String inExternalIdentifier,
        String inBusinessIdentifier
    );
    public RegistrationInfo getRegistrationInfo(String inRegistration);
    public String getRegistrationInfo(
        String inRegistration,
        int type
    );
    public String getVersion();

```

### 5.3.2. Applet API XAdESExtenderAppletWrapper

Hlavný modul aplikácie D.Sig XAdES Extender Java publikuje pre webové aplikácie využívajúce Java applet API nasledujúce rozhranie:

#### Package:

```
sk.ditec.zep.dsigner.xades.extender.applet
```

#### Triedu:

```
XAdESExtenderAppletWrapper
```

#### Metódy:

```

public boolean getErrorMessage(final JSObject callback);
public boolean getDataSignatures(final JSObject callback);
public boolean getDocumentUnauthorized(final JSObject callback);
public boolean getRegistration(final JSObject callback);
public boolean getRegistrationBase64(final JSObject callback);
public int getDocumentCount();
public int moveToDocument(int index);
public int getDocumentType();
public void initialize();
public boolean openFile(
    final Object title,
    final Object filter,
    final int readBinary,
    final int type,
    final JSObject callback
);

```

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

```

public boolean createNewDataSignatures(
    final Object inDataEnvelope,
    final Object inURI,
    final Object inID,
    final Object inDescription,
    final JSObject callback
);
public boolean createNewDataSignatures(
    final Object inDataEnvelope,
    final Object inURI,
    final Object inID,
    final Object inDescription,
    final Object dataSignaturesVersion,
    final JSObject callback
);
public boolean addDataEnvelopeToExistingDataSignatures(
    final Object inDataSignatures,
    final Object inDataEnvelope,
    final JSObject callback
);
public boolean insertDataSignatures(
    final Object inDataSignatures,
    final JSObject callback
);
public boolean getDataEnvelopeInfo(
    final Object inDataEnvelope,
    final int type,
    final JSObject callback
);
public boolean getDataSignaturesInfo(
    final Object inDataSignatures,
    final int type,
    final JSObject callback
);
public boolean verifyDataSignatures(
    final Object inDataSignatures,
    final JSObject callback
);
public boolean createNewDocumentUnauthorized(
    final Object inURI,
    final Object inID,
    final Object inDescription,
    final Object objectData,
    final Object inObjectID,
    final Object inObjectMimeType,
    final Object inObjectEncoding,
    final Object inObjectIdentifier,

```

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

```

        final JSObject callback
    );
    public boolean getDocumentUnauthorizedInfo(
        final Object inDocumentUnauthorized,
        final int type,
        final JSObject callback
    );
    public boolean createNewRegistration(
        final Object inURI,
        final Object inID,
        final Object inDescription,
        final Object inExternalIdentifier,
        final Object inBusinessIdentifier,
        final JSObject callback
    );
    public boolean getRegistrationInfo(
        final Object inRegistration,
        final int type,
        final JSObject callback
    );
    public boolean getVersion(final JSObject callback);

```

### 5.3.3. Triedy použité ako výstupné hodnoty metód

```

public class DataEnvelopeInfo {
    private String id;
    private String uri;
    private String description;
    private SignatureInfo signatureInfo;

    public String getId()
    public void setId(String id)
    public String getUri()
    public void setUri(String uri)
    public String getDescription()
    public void setDescription(String description)
    public SignatureInfo getSignatureInfo()
    public void setSignatureInfo(SignatureInfo signatureInfo)
}

```

```

public class DataSignaturesInfo {

    private String id;
    private String uri;
    private String description;
    private String dataSignaturesVersion;
    private int signatureInfoListCount;
    private List<SignatureInfo> signatureInfoList;
}

```

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

```

private int dispatchObjectInfoListCount;
private List<DispatchObjectInfo> dispatchObjectInfoList;
private int objectIdListCount;
private List<String> objectIdList;
private int signatureIdListCount;
private List<String> signatureIdList;
private int dataEnvelopeListCount;
private List<String> dataEnvelopeList;

public String getId();
public void setId(String id);
public String getUri();
public void setUri(String uri);
public String getDescription();
public void setDescription(String description);
public int getSignatureInfoListCount();
public List<SignatureInfo> getSignatureInfoList();
public void setSignatureInfoList(List<SignatureInfo>
signatureInfoList);
public int getDispatchObjectInfoListCount();
public List<DispatchObjectInfo> getDispatchObjectInfoList();
public void setDispatchObjectInfoList(List<DispatchObjectInfo>
dispatchObjectInfoList);
public int getObjectIdListCount();
public List<String> getObjectIdList();
public void setObjectIdList(List<String> objectIdList);
public int getSignatureIdListCount();
public List<String> getSignatureIdList();
public void setSignatureIdList(List<String> signatureIdList);
public String getDataSignaturesVersion();
public void setDataSignaturesVersion(String dataSignaturesVersion);
public int getDataEnvelopeListCount();
public List<String> getDataEnvelopeList();
public void setDataEnvelopeList(List<String> dataEnvelopeList);

}

public class SignatureInfo {

private String signatureId;
private int signedObjectInfoListCount;
private List<SignedObjectInfo> signedObjectInfoList;
private String X509CertificateDataBase64;
private String signatureVersion;
private List<ProductInfo> productInfos;

```

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

```

    public String getSignatureId()
    public void setSignatureId(String signatureId)
    public int getSIGNEDObjectInfoListCount()
    public void setSignedObjectInfoListCount(int
signedObjectInfoListCount)
    public List<SignedObjectInfo> getSignedObjectInfoList()
    public void setSignedObjectInfoList(List<SignedObjectInfo>
signedObjectInfoList)
    public String getX509CertificateDataBase64()
    public void setX509CertificateDataBase64(String
x509CertificateDataBase64)
    public String getSignatureVersion()
    public void setSignatureVersion(String signatureVersion)
    public List<ProductInfo> getProductInfos()
    public void setProductInfos(List<ProductInfo> productInfos)
}

public class ProductInfo {
    private String productName;
    private String productVersion;

    public String getProductName()
    public void setProductName(String productName)
    public String getProductVersion()
    public void setProductVersion(String productVersion)
}

public class SignedObjectInfo {

    private String objectId;
    private String description;
    private String objectIdentifier;
    private String mimeType;
    private String data;
    private String encoding;
    private String verifDataObjectVersion;
    private Map<String, String> verifDataObjectParams;

    public String getObjectId()
    public void setObjectId(String objectId)
    public String getDescription()
    public void setDescription(String description)
    public String getObjectIdentifier()
    public void setObjectIdentifier(String objectIdentifier)
    public String getMimeType()
    public void setMimeType(String mimeType)

```

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

```

public String getData()
public void setData(String data)
public String getEncoding()
public void setEncoding(String encoding)
public String getVerifDataObjectVersion()
public void setVerifDataObjectVersion(String verifDataObjectVersion)
public Map<String, String> getVerifDataObjectParams()
public void setVerifDataObjectParams(Map<String, String>
verifDataObjectParams)
}

```

```

public class DispatchObjectInfo {

    private String objectId;
    private String mimeType;
    private String data;
    private String encoding;

    public String getObjectId()
    public void setObjectId(String objectId)
    public String getMimeType()
    public void setMimeType(String mimeType)
    public String getData()
    public void setData(String data)
    public String getEncoding()
    public void setEncoding(String encoding) }

```

```

public class DocumentUnauthorizedInfo {

    private String id;
    private String uri;
    private String description;
    private int unauthorizedObjectInfoListCount;
    private List<UnauthorizedObjectInfo> unauthorizedObjectInfoList;

    public String getId()
    public void setId(String id)
    public String getUri()
    public void setUri(String uri)
    public String getDescription()
    public void setDescription(String description)
    public int getUnauthorizedObjectInfoListCount()
    public List<UnauthorizedObjectInfo> getUnauthorizedObjectInfoList()
    public void
setUnauthorizedObjectInfoList(List<UnauthorizedObjectInfo>
unauthorizedObjectInfoList)
}

```

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

```

public class UnauthorizedObjectInfo {

    private String id;
    private String mimeType;
    private String encoding;
    private String identifier;
    private String data;

    public String getId()
    public void setId(String id)
    public String getMimeType()
    public void setMimeType(String mimeType)
    public String getEncoding()
    public void setEncoding(String encoding)
    public String getIdentifier()
    public void setIdentifier(String identifier)
    public String getData()
    public void setData(String data)
}

public class RegistrationInfo {

    private String id;
    private String uri;
    private String description;
    private String externalIdentifier;
    private String businessIdentifier;
    private int documentInfoListCount;
    private List<DocumentInfo> documentInfoList;

    public String getId()
    public void setId(String id)
    public String getUri()
    public void setUri(String uri)
    public String getDescription()
    public void setDescription(String description)
    public String getExternalIdentifier()
    public void setExternalIdentifier(String externalIdentifier)
    public String getBusinessIdentifier()
    public void setBusinessIdentifier(String businessIdentifier)
    public int getDocumentInfoListCount()
    public List<DocumentInfo> getDocumentInfoList()
    public void setDocumentInfoList(List<DocumentInfo>
documentInfoList)
}

```

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

```

public class DocumentInfo {

    private int documentType;
    private String data;

    public int getDocumentType()
    public void setDocumentType(int documentType)
    public String getData()
    public void setData(String data)
}

public class OpenFileInfo {

    private String fileExtension;
    private String fileName;
    private String fileNameWithPath;
    private String fileContent;
    private String fileContentBase64;

    public OpenFileInfo()
    public OpenFileInfo(File file, String data, boolean base64)
    public String getFileExtension()
    public String getFileName()
    public String getFileNameWithPath()
    public String getFileContent()
    public String getFileContentBase64()
}

```

## 5.4. MessageContainer

### 5.4.1. Java API MessageContainer

**Package:**

sk.ditec.zep.dsigner.xades.extender

**Triedu:**

MessageContainer

**Konštruktor:**

**public** MessageContainer()

**Metódy:**

**public void** initialize(  
     String messageId,  
     String senderId,  
     String recipientId,  
     String messageType,  
     String messageSubject,

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

```

        String senderBusinessReference,
        String recipientBusinessReference)
;
public void initialize(String mc);
public int addXMLObject(
    String id,
    String name,
    String description,
    String classAttr,
    Boolean isSigned,
    String mimeType,
    String objectData)
;
public int addBase64Object(
    String id,
    String name,
    String description,
    String classAttr,
    Boolean isSigned,
    String mimeType,
    String objectDataBase64
);
public String getMessageContainer();
public MessageContainerInfo getMessageContainerInfo();
public String getMessageContainerInfo(int type);
public int getObjectCount();
public ObjectInfo getObjectInfo(int i);
public String getObjectInfo(
    int i,
    int type
);
public String getObjectData(int i);
public String getErrorMessage();
public boolean isInitialized();
public String getVersion();

```

## 5.4.2. Applet API MessageContainerAppletWrapper

### Package:

sk.ditec.zep.dsigner.xades.extender.applet

### Triedu:

MessageContainerAppletWrapper

### Metódy:

```

public boolean initialize(
    final Object messageId,
    final Object senderId,
    final Object recipientId,
    final Object messageType,

```

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

```

        final Object messageSubject,
        final Object senderBusinessReference,
        final Object recipientBusinessReference,
        final JSObject callback
    );
    public boolean initialize(
        final Object mc,
        final JSObject callback
    );

    public boolean addXMLObject(
        final Object id,
        final Object name,
        final Object description,
        final Object classAttr,
        final Boolean isSigned,
        final Object mimeType,
        final Object objectData,
        final JSObject callback
    );
    public boolean addBase64Object(
        final Object id,
        final Object name,
        final Object description,
        final Object classAttr,
        final Boolean isSigned,
        final Object mimeType,
        final Object objectDataBase64,
        final JSObject callback
    );

    public boolean getMessageContainer(final JSObject callback);
    public boolean getMessageContainerInfo(
        final int type,
        final JSObject callback
    );
    public boolean getObjectCount(final JSObject callback);
    public boolean getObjectInfo(
        final int i,
        final int type,
        final JSObject callback
    );
    public boolean getObjectData(
        final int i,
        final JSObject callback
    );
    public boolean getErrorMessage(final JSObject callback);
    public boolean isInitialized(final JSObject callback);

```

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

`public boolean getVersion(final JSONObject callback);`

### 5.4.3. Triedy použité ako výstupné hodnoty metód

```
public class MessageContainerInfo {
    private String messageId;
    private String senderId;
    private String recipientId;
    private String messageType;
    private String messageSubject;
    private String senderBusinessReference;
    private String recipientBusinessReference;

    public String getMessageId()
    public void setMessageId(String messageId)
    public String getSenderId()
    public void setSenderId(String senderId)
    public String getRecipientId()
    public void setRecipientId(String recipientId)
    public String getMessageType()
    public void setMessageType(String messageType)
    public String getMessageSubject()
    public void setMessageSubject(String messageSubject)
    public String getSenderBusinessReference()
    public void setSenderBusinessReference(String
senderBusinessReference)
    public String getRecipientBusinessReference()
    public void setRecipientBusinessReference(String
recipientBusinessReference)
}

public class ObjectInfo{
    private String id;
    private String name;
    public String description;
    public String classAttr;
    public Boolean isSigned;
    public String mimeType;
    public String encoding;

    public String getId()
    public void setId(String id)
    public String getName()
    public void setName(String name)
    public String getDescription()
    public void setDescription(String description)
    public String getClassType()
    public void setClassType(String classType)
```

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

```

public Boolean getIsSigned()
public void setIsSigned(Boolean isSigned)
public String getMimeType()
public void setMimeType(String mimeType)
public String getEncoding()
public void setEncoding(String encoding)
}

```

## 5.5. Popis metód triedy Extender

Nasleduje stručný popis metód triedy XAdESExtender, resp. XAdESExtenderAppletWrapper. Metódy sú rozdelené do štyroch množín podľa účelu na: spoločné (pomocné) metódy, metódy zloženého podpisu, metódy dokumentu bez autorizácie a metódy podania.

Všetky metódy Java Appletu majú oproti Java API ako vstupný parameter objekt:

- **final** JSObject callback.

Tento parameter je použitý na získanie výstupu danej metódy, detaily sú popísané v časti 5.2.

### 5.5.1. Popis spoločných metód

#### 5.5.1.1. konštruktor

**vstupné parametre:** žiadne

**výstupné parametre:** žiadne

**popis:**

Vytvorí sa prázdna štruktúra triedy Extender. Očakáva sa jeho naplnenie metódou `Initialize()`.

#### 5.5.1.2. metóda `getErrorMessage`

**vstupné parametre:** žiadne

**výstupné parametre:** žiadne

**návratová hodnota:** textový reťazec

**popis:**

Metóda vráti popis chyby, ktorá nastala počas volania poslednej metódy. Chyby sa týkajú len metód, v ktorých popise je uvedené hlásenie chýb.

#### 5.5.1.3. metóda `getDataSignatures`

**vstupné parametre:** žiadne

**výstupné parametre:** žiadne

**návratová hodnota:** textový reťazec

**popis:**

Metóda vráti štruktúru vytvoreného zloženého elektronického podpisu v súlade s [4] alebo [5] a [6].

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

#### 5.5.1.4. metóda **getDocumentUnauthorized**

**vstupné parametre:** žiadne

**výstupné parametre:** žiadne

**návratová hodnota:** textový reťazec

**popis:**

Metóda vráti štruktúru dokumentu bez autorizácie podľa [7].

#### 5.5.1.5. metóda **getRegistration**

**vstupné parametre:** žiadne

**výstupné parametre:** žiadne

**návratová hodnota:** textový reťazec

**popis:**

Metóda vráti štruktúru podania podľa [8].

#### 5.5.1.6. metóda **getRegistrationBase64**

**vstupné parametre:** žiadne

**výstupné parametre:** žiadne

**návratová hodnota:** textový reťazec

**popis:**

Metóda vráti štruktúru podania podľa [8] v base64 kódovaní.

#### 5.5.1.7. metóda **getDocumentCount**

**vstupné parametre:** žiadne

**výstupné parametre:** žiadne

**návratová hodnota:** celé číslo

**popis:**

Metóda vráti počet inštancií dokumentov vytvorených pomocou metód `CreateNewDataSignatures` alebo `CreateNewDocumentUnauthorized` (čiže súčet inštancií zložených podpisov `DataSignatures` a nepodpísaných dokumentov `DocumentUnauthorized`).

#### 5.5.1.8. metóda **moveToDocument**

**vstupné parametre:** celé číslo

**výstupné parametre:** žiadne

**návratová hodnota:** celé číslo

**popis:**

Metóda nastaví dokument so vstupným indexom ako aktuálny (nad ktorým sa budú vykonávať nasledujúce operácie). Ak je vstupný index mimo rozsahu (menší ako nula, väčší alebo rovný ako počet dokumentov) nevykoná nič a vráti chybu. Používa sa číslovanie od nuly, t.j. prvý dokument má poradové číslo 0.

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

### 5.5.1.9. metóda `getDocumentType`

**vstupné parametre:** žiadne

**výstupné parametre:** žiadne

**návratová hodnota:** celé číslo

**popis:**

Metóda vráti typ aktuálne nastaveného dokumentu. Návratová hodnota:

- 0 zložený elektronický podpis,
- 1 znamená dokument bez autorizácie,
- záporná chyba.

Iné hodnoty sa vrátiť nesmú.

### 5.5.1.10. metóda `initialize`

**vstupné parametre:** žiadne

**výstupné parametre:** žiadne

**návratová hodnota:** žiadna

**popis:**

Metóda nastaví triedu do iniciálneho stavu:

1. nastaví počet dokumentov na nula,
2. nastaví počet zložených podpisov na nula,
3. všetky interné zoznamy nastaví na prázdne.

### 5.5.1.11. metóda `openFile`

**vstupné parametre:**

- textový reťazec `title` – popis v hlavičke dialógu,
- textový reťazec `filter` – umožňuje použiť filter pri výbere súboru,
- celé číslo `readBinary` – či sa načítava binárne (1=áno, inak nie),
- celé číslo `type` – určuje typ výstupu (0=XML, inak JSON) štruktúry `OpenFileInfo`
- textový reťazec `callbackName` – určuje meno funkcie, ktorá sa má zavolať, keď užívateľ vyberie v dialógu súbor (tento parameter sa využíva iba pre applet).

**výstupné parametre:** žiadne

**návratová hodnota:** štruktúra `OpenFileInfo` (typu `string`)

Štruktúra `OpenFileInfo` obsahuje:

- textový reťazec `fileExtension` – koncovka súboru,
- textový reťazec `fileName` – meno súboru,
- textový reťazec `fileNameWithPath` – meno súboru vrátane cesty,
- textový reťazec `fileContent` – obsah súboru, v prípade binárnych súborov sa nenaplní,

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

- textový reťazec `fileContentBase64` – obsah súboru v base64 kódovaní, v prípade textových súborov sa nenapĺňa.

#### **popis:**

Metóda zobrazí dialóg pre načítanie súboru podľa vstupných parametrov. Vstupný filter môže byť napr. v tvaroch „XML File|\*.xml|All|\*.\*“, alebo “Image Files (\*.bmp, \*.jpg)|\*.bmp;\*.jpg“.

Ak je nastavený `readBinary` na jedna, tak sa načítava binárne a výstupný reťazec `fileContent*` bude prázdny. V prípade, že sa jedná o metódu triedy `XAdESExtenderAppletWrapper`, sa po zvolení názvu súboru zavolá daná funkcia `callbackName`.

## **5.5.2. Metódy zloženého elektronického podpisu**

### **5.5.2.1. metóda `createNewDataSignatures`**

#### **vstupné parametre:**

- textový reťazec `inDataEnvelope` – štruktúra jednoduchého podpisu podľa [1], [2] alebo [3],
- textový reťazec `inURI`,
- textový reťazec `inID`,
- textový reťazec `inDescription`,
- textový reťazec `dataSignaturesVersion` – (nepovinný parameter) označenie formátu vytváraného zloženého el.podpisu (povolené hodnoty sú 1.0, 1.1 a 2.0).

**výstupné parametre:** žiadne

**návratová hodnota:** celé číslo

#### **popis:**

Parameter `dataSignaturesVersion` je nepovinný (metóda je preťažená). V prípade absencie je jeho hodnota nastavená na hodnotu „1.1“.

Metóda vytvorí na základe vstupných parametrov štruktúru zloženého elektronického podpisu podľa [4] alebo [5] a [6]. Vytvorená štruktúra bude zaradená na koniec zoznamu dokumentov a bude predstavovať aktuálny dokument, nad ktorým sa budú vykonávať nasledujúce operácie.

V prípade úspechu je návratová hodnota 0, v opačnom prípade číslo rôzne od nuly. Výsledný elektronický podpis bude uložený v `DataSignatures` vlastnosti..

### **5.5.2.2. metóda `addDataEnvelopeToExistingDataSignatures`**

#### **vstupné parametre:**

- textový reťazec `inDataSignatures` – štruktúra zloženého elektronického podpisu verzie 1.0 [4], 1.1 [5] alebo 2.0 [6],
- textový reťazec `inDataEnvelope` – štruktúra jednoduchého zloženého podpisu verzie 1.0 [1], 1.1 [2], alebo 2.0 [3].

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

**výstupné parametre:** žiadne

**návratová hodnota:** celé číslo

**popis:**

Metóda vloží vstupnú štruktúru `inDataEnvelope` jednoduchého podpisu do zloženej štruktúry `inDataSignatures` a výslednú zloženú štruktúru vloží na aktuálnu pozíciu v zozname zložených elektronických podpisov triedy.

V prípade úspechu je návratová hodnota 0, v opačnom prípade číslo rôzne od nuly. Výsledný elektronický podpis bude uložený v `DataSignatures` vlastnosti.

### 5.5.2.3. metóda `insertDataSignatures`

**vstupné parametre:** textový reťazec `inDataSignatures`

**výstupné parametre:** žiadne

**návratová hodnota:** celé číslo

**popis:**

Metóda vloží vstupný zložený podpis do kolekcie zložených podpisov.

V prípade úspechu je návratová hodnota 0, v opačnom prípade číslo rôzne od nuly. Výsledný elektronický podpis bude uložený v `DataSignatures` vlastnosti.

### 5.5.2.4. metóda `getDataSignaturesInfo`

**vstupné parametre:** textový reťazec `inDataSignatures`

**výstupné parametre:** žiadne

**návratová hodnota:** štruktúra `DataSignaturesInfo`

**popis:**

Metóda vráti informácie o štruktúre aktuálneho zloženého podpisu. Štruktúra `DataSignaturesInfo` obsahuje:

- `string Id` – atribút `Id` z `DataSignatures`,
- `string Uri` – atribút `URI` z `DataSignatures`,
- `string Description` – atribút `Description` z `DataSignatures`,
- `int SignatureInfoListCount` – počet podpisov,
- `List<SignatureInfo> SignatureInfoList` – zoznam podpisov,
- `int DispatchObjectInfoListCount` – počet `dispatch` objektov,
- `List<DispatchObjectInfo> DispatchObjectInfoList` – zoznam `dispatch` objektov,
- `int ObjectIdListCount` – počet objektov (mimo `dispatchnotes` objektov),
- `List<string> ObjectIdList` – zoznam `Id` atribútov objektov (mimo `dispatchnotes` objektov),
- `int SignatureIdListCount` – počet podpisov,
- `List<string> SignatureIdList` – zoznam `Id` atribútov objektov.

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

Štruktúra `SignatureInfo` obsahuje:

- `string SignatureId` – Id atribút konkrétneho podpisu,
- `int SignedObjectInfoListCount` – počet podpísaných objektov,
- `List<SignedObjectInfo> SignedObjectInfoList` – zoznam podpísaných objektov,
- `string X509CertificateDataBase64` – podpisový certifikát v base64 (vyhľadáva sa postupom špecifikovaným v metóde `VerifyDataSignatures`).

Štruktúra `SignedObjectInfo` obsahuje:

- `string ObjectId` – Id objektu,
- `string Description` – popis objektu,
- `string ObjectIdentifier` – identifikátor objektu,
- `string MimeType` – mimetype objektu,
- `string Data` – dáta objektu,
- `string Encoding` – kódovanie dát v objekte.

Štruktúra `DispatchObjectInfo` obsahuje informácie o nepodpísaných objektoch:

- `string ObjectId` – Id objektu,
- `string MimeType` – mimetype objektu,
- `string Data` – dáta objektu,
- `string Encoding` – kódovanie dát v objekte.

### 5.5.2.5. metóda `getDataSignaturesInfo`

**vstupné parametre:**

- textový reťazec `inDataSignatures`,
- celé číslo `type` – určuje požadovaný typ výstupu.

**výstupné parametre:** žiadne

**návratová hodnota:** textový reťazec

**popis:**

Preťažená metóda vráti informácie o štruktúre aktuálneho zloženého podpisu vo formáte XML (vstup `type = 0`), alebo JSON (vstup `type = 1`).

### 5.5.2.6. metóda `verifyDataSignatures`

**vstupné parametre:** textový reťazec `inDataSignatures`

**výstupné parametre:** žiadne

**návratová hodnota:** celé číslo

**popis:**

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

Metóda vykoná overenie integrity zloženého elektronického podpisu tak, že overí každý jednoduchý podpis nasledovným spôsobom:

- i) vezme sa SignedInfo, nájdu sa všetky referencie v ňom a skontroluje sa, či každá referencia ukazuje tam kde má a či každá referencia je správna - teda či haš sedí s tým, čo je tam uvedené,
- ii) overí sa podpis elementu SignedInfo voči SignatureValue pomocu nájdeného podpisového certifikátu. Podpisový certifikát sa určí tak, že sa vezmú všetky certifikáty z KeyInfo (v podelemente X509Data/Certificate). A z nich sa vráti ten, ktorý je uvedený v xades:SignedSignatureProperties/SigningCertificate (podľa hašu). Ak sa podpisový certifikát nenájde, metóda vráti chybu.

V prípade úspechu vráti číslo 0, v opačnom prípade vráti číslo rôzne od nuly.

### 5.5.3. Dokument bez autorizácie

#### 5.5.3.1. createNewDocumentUnauthorized

**vstupné parametre:**

- textové reťazce `inURI`, `inID`, `inDescription` – parametre identifikujúce štruktúru dokumentu bez autorizácie,
- textový reťazec `objectData` – dáta objektu bez autorizácie,
- textové reťazce `inObjectID`, `inObjectMimeType`, `inObjectEncoding`, `inObjectIdentifier` – parametre identifikujúce dáta dokumentu bez autorizácie.

**výstupné parametre:** žiadne

**návratová hodnota:** celé číslo

**popis:**

Metóda vytvorí štruktúru dokumentu bez autorizácie v súlade s [7]. Táto štruktúra bude zaradená na koniec zoznamu dokumentov a bude predstavovať aktuálny dokument, nad ktorým sa budú vykonávať ďalšie operácie.

V prípade úspechu vráti číslo 0, v opačnom prípade vráti číslo rôzne od nuly.

#### 5.5.3.2. metóda getDocumentUnauthorizedInfo

**vstupné parametre:** textový reťazec `inDocumentUnauthorized`

**výstupné parametre:** žiadne

**návratová hodnota:** štruktúra `DocumentUnauthorizedInfo`

**popis:**

Metóda vráti informácie o vstupnom dokumente bez autorizácie. V prípade úspechu vráti nasledovnú štruktúru, v opačnom prípade vráti null.

Štruktúra `DocumentUnauthorizedInfo` obsahuje:

- string `Id` – atribút `Id` z `DocumentUnauthorized`,
- string `Uri` – atribút `Uri` z `DocumentUnauthorized`,

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

- `string Description` – atribút `Description` z `DocumentUnauthorized`,
- `int UnauthorizedObjectInfoListCount` – počet objektov,
- `List<UnauthorizedObjectInfo> UnauthorizedObjectInfoList` – zoznam objektov.

Štruktúra `UnauthorizedObjectInfo` obsahuje:

- `string Id` – atribút `Id` z konkrétneho objektu,
- `string MimeType` – atribútu `MimeType` z konkrétneho objektu,
- `string Encoding` – atribút `Encoding` z konkrétneho objektu,
- `string Identifier` – atribút `Identifier` z konkrétneho objektu,
- `string Data` – dáta konkrétneho objektu.

### 5.5.3.3. metóda `getDocumentUnauthorizedInfo`

**vstupné parametre:**

- textový reťazec `inDocumentUnauthorized`,
- celé číslo `type` – požadovaný typ výstupu

**výstupné parametre:** žiadne

**návratová hodnota:** textový reťazec

**popis:**

Preťažená metóda vráti informácie o vstupnej štruktúre vo formáte XML (ak `type=0`), alebo vo formáte JSON (ak `type=1`).

V prípade neúspechu je návratová hodnota prázdny reťazec.

## 5.5.4. Podanie

### 5.5.4.1. metóda `createNewRegistration`

**vstupné parametre:** textové reťazce `inURI`, `inID`, `inDescription`, `inExternalIdentifier`, `inBusinessIdentifier` – parametre identifikujúce štruktúru podania

**výstupné parametre:** žiadne

**návratová hodnota:** celé číslo

**popis:**

Metóda vytvorí štruktúru podania podľa [8] z dokumentov, ktoré boli vytvorené v rámci inštancie triedy (pričom zachováva ich poradie). V prípade úspechu je návratová hodnota nula, v opačnom prípade rôzna od nuly. Výsledné podanie bude uložené v `Registration` a `Registration64` vlastnosti.

### 5.5.4.2. metóda `getRegistrationInfo`

**vstupné parametre:** textový reťazec `inRegistration`

**výstupné parametre:** žiadne

**návratová hodnota:** `RegistrationInfo`

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

### popis:

Metóda vráti na výstup štruktúru obsahujúcu informácie o vstupnom podaní. Štruktúra obsahuje nasledovné:

- `string Id` – atribút z `RegistrationInfo`,
- `string Uri` – atribút z `RegistrationInfo`,
- `string Description` – atribút `Description` z `RegistrationInfo`,
- `string ExternalIdentifier` – atribút `Description` z `RegistrationInfo`,
- `string BusinessIdentifier` – atribút `Description` z `RegistrationInfo`,
- `int DocumentInfoListCount` – počet dokumentov,
- `List<DocumentInfo> DocumentInfoList` – zoznam dokumentov.

Štruktúra `DocumentInfo` obsahuje:

- `int DocumentType` – typ konkrétneho dokumentu:
  - ◆ 0 – zložený elektronický podpis,
  - ◆ 1 – dokument bez autorizácie,
  - ◆ -1 – neznámy typ dokumentu,
- `string Data` – dáta konkrétneho dokumentu.

#### 5.5.4.3. metóda `getRegistrationInfo`

**vstupné parametre:**

- textový reťazec `inRegistration`,
- celé číslo `type` – požadovaný typ výstupu

**výstupné parametre:** žiadne

**návratová hodnota:** textový reťazec

**popis:**

Preťažená metóda vráti informácie o štruktúre podania v XML (`type=0`) alebo JSON (`type=1`) štruktúre.

## 5.6. Popis metód triedy `MessageContainer`

Nasleduje popis funkcionality metód triedy `MessageContainer`, resp. `MessageContainerAppletWrapper`.

Všetky metódy Java Appletu (triedy `AppletWrapped`) majú oproti Java API ako vstupný parameter objekt:

- `final JSObject callback`.

Tento parameter je použitý na získanie výstupu danej metódy. Detaily sú uvedené v kapitole 5.2.

### 5.6.1. konštruktor

**vstupné parametre:** žiadne

**výstupné parametre:** žiadne

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

**popis:**

Vytvorí sa prázdna štruktúra MessageContainer. Očakáva sa jeho naplnenie metódou `Initialize()`.

### 5.6.2. metóda initialize

**vstupné parametre:**

- textový reťazec `messageId` - guid, RFC4122,
- textový reťazec `senderId` - URI,
- textový reťazec `recipientId` - URI,
- textový reťazec `messageType`,
- textový reťazec `messageSubject` - nepovinný parameter,
- textový reťazec `senderBusinessReference` - nepovinný parameter,
- textový reťazec `recipientBusinessReference` - nepovinný parameter.

**výstupné parametre:** žiadne

**návratová hodnota:** žiadna

**popis:**

Metóda vloží zadané parametre do štruktúry MessageContainer-a.

### 5.6.3. metóda getErrorMessage

**vstupné parametre:** žiadne

**výstupné parametre:** žiadne

**návratová hodnota:** textový reťazec

**popis:**

Metóda vráti obsah internej premennej ErrorMessage.

### 5.6.4. metóda addXMLObject

**vstupné parametre:**

- textový reťazec `id` - guid, RFC4122,
- textový reťazec `name`,
- textový reťazec `description`,
- textový reťazec `classAttr`,
- boolean `isSigned`,
- textový reťazec `mimeType`,
- textový reťazec `objectData`.

**výstupné parametre:** žiadne

**návratová hodnota:** celé číslo

**popis:**

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

Metóda skontroluje, či vstupné údaje tvoria platný XML dokument (well-formed) a ak nie, skončí s chybou. Ak je vstupný dokument platný, tak odstráni prípadnú XML deklaráciu a vloží ho do kolekcie objektov v existujúcej štruktúre MessageContainer-a. Atribút Encoding sa nastaví na hodnotu „XML“. Na výstup vráti chybový kód.

### 5.6.5. metóda addBase64Object

**vstupné parametre:**

- textový reťazec `id`,
- textový reťazec `name`,
- textový reťazec `description`,
- textový reťazec `classAttr`,
- boolean `isSigned` - true/false,
- textový reťazec `mimeType`,
- textový reťazec `objectDataBase64`.

**výstupné parametre:** žiadne

**návratová hodnota:** celé číslo

**popis:**

Metóda skontroluje, či je vstupný reťazec vo formáte base64 a ak áno, vloží zadané údaje do kolekcie objektov v existujúcej štruktúre MessageContainer-a. Atribút Encoding sa nastaví na hodnotu „base64“. Na výstup vráti chybový kód.

### 5.6.6. metóda getMessageContainer

**vstupné parametre:** žiadne

**výstupné parametre:** žiadne

**návratová hodnota:** textový reťazec

**popis:**

Metóda vráti na výstup štruktúru MessageContainer.

### 5.6.7. metóda initialize

**vstupné parametre:**

- textový reťazec `mc` - obsah existujúcej štruktúry MessageContainer.

**výstupné parametre:** žiadne

**návratová hodnota:** žiadna

**popis:**

Metóda načíta vstupnú štruktúru kontajnera MessageContainer. V prípade neúspechu sa chyba zaznačí do internej premennej.

### 5.6.8. metóda isInitialized

**vstupné parametre:** žiadne

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

**výstupné parametre:** žiadne

**návratová hodnota:** boolean

**popis:**

Metóda vráti `true`, ak bol kontajner správne inicializovaný, teda metóda `Initialize` skončila korektne. Inak vráti `false`.

### 5.6.9. metóda `getMessageContainerInfo`

**vstupné parametre:** žiadne

**výstupné parametre:** žiadne

**návratová hodnota:** štruktúra `MessageContainerInfo`

**popis:**

Metóda vráti údaje z načítaného kontajnera. Štruktúra `MessageContainerInfo` obsahuje:

- textový reťazec `MessageId`,
- textový reťazec `SenderId`,
- textový reťazec `RecipientId`,
- textový reťazec `MessageType`,
- textový reťazec `MessageSubject` – nepovinne,
- textový reťazec `SenderBusinessReference` – nepovinne,
- textový reťazec `RecipientBusinessReference` – nepovinne.

### 5.6.10. metóda `getMessageContainerInfo`

**vstupné parametre:** celé číslo `type` – určuje typ výstupu

**výstupné parametre:** žiadne

**návratová hodnota:** textový reťazec

**popis:**

Preťažená metóda vráti údaje z načítaného kontajnera vo formáte XML, ak `type` bolo rovné nule, alebo vo formáte JSON, ak `type` bolo rovné jednej. V ostatných prípadoch (hodnoty parametra `type`) vráti prázdny string a chybu zapíše do internej premennej.

### 5.6.11. metóda `getObjectCount`

**vstupné parametre:** žiadne

**výstupné parametre:** žiadne

**návratová hodnota:** celé číslo

**popis:**

Metóda vráti počet objektov v načítanej štruktúre `MessageContainer`.

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

### 5.6.12. metóda getObjectInfo

**vstupné parametre:** celé číslo *i*

**výstupné parametre:** žiadne

**návratová hodnota:** štruktúra ObjectInfo

**popis:**

Metóda vráti údaje *i*-teho objektu z kontajnera. Štruktúra ObjectInfo obsahuje:

- textový reťazec *Id*,
- textový reťazec *Name*,
- textový reťazec *Description*,
- textový reťazec *classAttr*,
- boolean *IsSigned*,
- textový reťazec *MimeType*,
- textový reťazec *Encoding*.

Ak je číslo *i* mimo rozsahu, vráti null a do ErrorMessage zapíše chybu.

### 5.6.13. metóda getObjectInfo

**vstupné parametre:**

- celé číslo *i* – číslo objektu,
- celé číslo *type* – určuje typ výstupu.

**výstupné parametre:** žiadne

**návratová hodnota:** textový reťazec

**popis:**

Preťažená metóda vráti údaje z načítaného kontajnera vo formáte XML, ak *type* bolo rovné nule, alebo vo formáte JSON, ak *type* bolo rovné jednej. Pri zápise vlastnosti *ClassAttr* sa do výstupu (XML aj JSON) zapíše označenie *Class*. Štruktúra *MessageContainer* je totiž definovaná s atribútom *Class*, interne sa však toto označenie premenovalo na *ClassAttr*, pretože „class“ je vyhradené v jazyku Java.

Ak je číslo *i* mimo rozsahu, vráti null a do ErrorMessage zapíše chybu.

### 5.6.14. metóda getObjectData

**vstupné parametre:**

- celé číslo *i* – číslo objektu.

**výstupné parametre:** žiadne

**návratová hodnota:** textový reťazec

**popis:**

Metóda vráti údaje zo zvoleného objektu.

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

### **5.6.15. metóda getVersion**

**vstupné parametre:** žiadne

**výstupné parametre:** žiadne

**návratová hodnota:** textový reťazec

**popis:**

Metóda vráti na výstup názov a verziu komponentu D.Sig XAdES Extender.

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

## 6. Návrátové kódy aplikácie

Návrátové kódy aplikácie pozostávajú len z dvoch hodnôt. V prípade korektného priebehu vracajú metódy kód nula a v prípade chyby kód „-1“. Popis chyby, ktorý možno získať volaním `getErrorMessage`, je uvedený v nasledujúcich tabuľkách.

Chybové hlásenia metódy `CreateNewDataSignatures`:

<b>CreateNewDataSignatures</b>
"Zle špecifikovaná verzia DataSignatures - " + dataSignaturesVersion + ". Podporované hodnoty sú 1.0, 1.1 a 2.0."
"Nastala chyba pri parsovaní DataEnvelope."
"Neočakávaný koreňový element pre DataEnvelope: " + dataEnvelope.getDocumentElement().toString()
"Signature element v DataEnvelope neobsahuje atribút Id."
"DataEnvelope obsahuje viac ako jeden Signature element."
"DataEnvelope neobsahuje Signature element."
"Chyba pri vytváraní XML Objektu."
"Vstupné štruktúry obsahujú duplicitné atribúty Id."
"Chýbajúce Id v Object elemente."

Návrátové kódy funkcie `AddDataEnvelopeToExistingDataSignatures`:

<b>AddDataEnvelopeToExistingDataSignatures</b>
"Nastala chyba pri parsovaní DataEnvelope."
"Neočakávaný koreňový element pre DataEnvelope: " + dataEnvelope.getDocumentElement().toString()
"Nastala chyba pri parsovaní DataSignatures."
"Neočakávaný koreňový element pre DataSignatures: " + dataSignatures.getDocumentElement().toString()
"DataSignatures nema špecifikovaný namespace."
"DataEnvelope nema špecifikovaný namespace."
"Signature element v DataEnvelope neobsahuje atribút Id."
"V DataSignatures už existuje Signature s Id=" + signatureIdString
"Chýbajúce Id v Signature elemente."
"Chýbajúce Id v Object elemente."
"Chýbajúce Id v Object elemente v DispatchNodesData."

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

"Chyba pri získavaní odtlačku pre objekt s id=" + objectId + " z DataEnvelope."
"Chyba pri získavaní odtlačku pre objekt s id=" + objectId + " z DataSignatures."
"Chyba pri vkladaní DataEnvelope do DataSignatures - objekt s ID=" + objectId + " už v DataSignatures existuje."
"DataSignatures neobsahuje Signature element."
"DataEnvelope obsahuje viac ako jeden Signature element."
"DataEnvelope neobsahuje Signature element."
"Vstupné štruktúry obsahujú duplicitné atribúty Id."
"Chyba pri vytváraní XML Objektu."
"Chyba pri ukladaní DataSignatures. Zoznam dokumentov je prázdny."

Chybové hlásenia funkcie CreateNewDocumentUnauthorized:

<b>CreateNewDocumentUnauthorized</b>
"Vstupné štruktúry obsahujú duplicitné atribúty Id."
"Chyba pri vytváraní XML Objektu."

Chybové hlásenia funkcie MoveToDocument:

<b>MoveToDocument</b>
"Index je mimo povoleného rozsahu."

V nasledujúcej tabuľke sú uvedené návratové kódy funkcie CreateNewRegistration:

<b>CreateNewRegistration</b>
Nastala chyba pri parsovaní výsledného Registration.
Neočakávaný koreňový element pre Registration: + regDoc.getDocumentElement().toString()
Vstupné štruktúry obsahujú duplicitné atribúty Id.
Chyba pri vytváraní XML Objektu.

Chybové hlásenia funkcie VerifyDataSignatures:

<b>VerifyDataSignatures</b>
Nastala chyba pri parsovaní DataSignatures.
Neočakávaný koreňový element pre DataSignatures: + dataSignatures.getDocumentElement().toString()
Verzia podpisu pre niektorý zo Signature elementov nie je špecifikovaná.

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

Nepodporovaná verzia podpisu - + signatureVersion
Chýbajúce Id pre Reference element v Signature.
Viac referencií na KeyInfo v SignedInfo elemente.
SignedInfo element obsahuje referenciu s nešpecifikovaným Type atribútom.
Viac referencií na SignatureProperties v SignedInfo elemente.
Viac referencií na SignedProperties v SignedInfo elemente.
Referencia na KeyInfo v SignedInfo pre ZEP2.0 nesmie obsahovať atribút Type.
Viac referencií na KeyInfo v SignedInfo elemente.
ZEP1.0 a ZEP1.1 nepodporujú referencie typu Object okrem referencie na KeyInfo.
Referencie na Manifest nie sú pre ZEP 2.0 povolené.
Signature neobsahuje všetky povinné referencie podľa profilu XAdES_ZEP.
Element Signature neobsahuje KeyInfo.
Element Signature neobsahuje SignatureProperties.
"Element Signature neobsahuje SignedProperties."
"Chýba element Manifest s id=" + id + "."
"Nie je možné skontrolovať referenciu " + referenceld + " na neexistujúci element."
Chýbajúce Id v Object elemente.
Nepodporovaný algoritmus pre výpočet odtlačku - + DigestMethod + "."
Neznámy algoritmus pri výpočte odtlačku certifikátu.
Chyba pri získavaní certifikátu.
"Nepodporovaný algoritmus podpisu" + signatureMethod + "."
Nastala chyba pri overovaní hodnoty podpisu.
Vstupné štruktúry obsahujú duplicitné atribúty Id.
"Chyba v referencii " + referenceld + " - odtlačky sa nezhodujú."
"Chyba pri kontrole referencie " + referenceld + "."

Chybové hlásenia metód addXMLObject a addBase64Object:

<b>addXMLObject a addBase64Object</b>
"Message Container nebol inicializovaný. Nie je možné pridať objekt."
"Parameter id je povinný."
"Parameter classType je povinný."
"Parameter mimeType je povinný."

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.213	Verzia 9

"Objekt s id=" + id + " už existuje."
"Vstupný XML objekt nie je platný XML súbor."
"Vstupný reťazec nie je vo formáte base64."
"Vstupný objekt je prázdny."

Ostatné metódy vrátia (podľa popisu) v prípade chyby prázdny string, resp. hodnotu Null (v závislosti od typu návratovej hodnoty). Detail chyby je možné aj v tomto prípade získať volaním `getErrorMessage()`.