

Integrační příručka

DataValidator, v1.0

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.221	Verzia 4

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	DataValidator, v1.0	
Ref. číslo	GOV_ZEP.221	Verzia 4

Vypracoval	Róbert Vittek	Podpis	Dátum 17. 2. 2025
Preveril		Podpis	Dátum
Schválil		Podpis	Dátum

Formulár	Dokument		
Ref. číslo	Fo 11	Dátum poslednej aktualizácie	Dátum 18. 5. 2005

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.221	Verzia 4

Záznamy o zmenách

Autor	Popis zmien	Dátum	Verzia

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.221	Verzia 4

Obsah

1.	Úvod	6
2.	Referencie	7
3.	Popis aplikácie	9
4.	Architektúra a postavenie komponentu v rámci prevádzkového prostredia.....	10
5.	Špecifikácia funkčnosti	12
5.1.	Scenáre použitia komponentu DataValidator	12
5.1.1.	Validácia na základe informácií evidovaných v nadradenej aplikácii	14
5.1.2.	Validácia na základe konfigurácie a skupiny formátov dátových objektov	15
5.2.	Validácia obsahu pre podporované formáty dátových objektov.....	16
5.3.	Získanie relevantných informácií zo štruktúry XMLDataContainer.....	20
6.	Špecifikácia API.....	21
6.1.	Integračné API.....	21
6.1.1.	Popis funkcií a premenných triedy DataValidator	24
6.1.1.1.	Konštruktor triedy DataValidator.....	24
6.1.1.2.	ValidateXMLDataContainer	24
6.1.1.3.	ValidatePDF	25
6.1.1.4.	ValidatePNG.....	26
6.1.1.5.	ValidateTXT	26
6.1.1.6.	ValidateDataObject	27
6.1.1.7.	ValidateAllDataObjects.....	28
6.1.1.8.	GetDataObjectsValidationRules	29
6.1.1.9.	GetErrorMessage	29
6.1.1.10.	Konštruktor triedy XMLDataContiner.....	29
6.1.1.11.	Initialize	29
6.1.1.12.	Initialize2	29

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.221	Verzia 4

6.1.1.13.	InitializeFromByteArray	30
6.1.1.14.	InitializeFromXmlDocument	30
6.1.1.15.	GetErrorMessage.....	30
6.1.1.16.	XdcXMLData	30
6.1.1.17.	XdcIdentifier	30
6.1.1.18.	XdcVersion.....	30
6.1.1.19.	XdcUsedXSD	31
6.1.1.20.	XdcUsedXSLT	31
6.1.1.21.	XsdReferenceURI	31
6.1.1.22.	XsdReferenceTransformAlg.....	31
6.1.1.23.	XsdReferenceDigestMethod	31
6.1.1.24.	XsdReferenceDigestValue	31
6.1.1.25.	XslReferenceURI	31
6.1.1.26.	XslReferenceTransformAlg	31
6.1.1.27.	XslReferenceDigestMethod	31
6.1.1.28.	XslReferenceDigestValue	32
6.1.1.29.	XslMediaDestinationTypeDescription.....	32
6.1.1.30.	XslXSLTLanguage	32
6.1.1.31.	XslTargetEnvironment.....	32
6.1.1.32.	XdcIncludeRefs.....	32
6.1.1.33.	XdcNamespaceURI.....	32
7.	Konfigurácia komponentu	33
8.	Návratové kódy aplikácie	34

1. Úvod

Tento dokument popisuje integračné rozhranie komponentu DataValidator, ktorý slúži na validáciu dátových objektov podpísaných zaručeným elektronickým podpisom (resp. pečatou), ktoré boli vytvorené v súlade s profilmi XAdESbp [5] a XAdES_ZEPbp [18], resp. CAdESbp [7] a CAdES_ZEP v2.0 [26].

Tento dokument je prílohou používateľských príručiek:

- SCVA aplikácie D.Verifier-SVR/XAdES [19] pre overovanie zaručeného elektronického podpisu, resp. pečate vytvorenej v súlade s profilmi XAdESbp a XAdES_ZEPbp,
- SCVA aplikácie D.Verifier-SVR/CAdES [20] pre overovanie zaručeného elektronického podpisu, resp. pečate vytvorenej v súlade s profilmi CAdESbp a CAdES_ZEP v2.0.

Dokument je určený pre zhotoviteľov, prípadne prevádzkovateľov systémov, v rámci ktorých bude komponent DataValidator integrovaný.

Jednotlivé časti dokumentácie komponentu DataValidator je možné použiť pri tvorbe dokumentácie týchto systémov po dohode s vlastníkmi autorských práv komponentu DataValidator.

2. Referencie

- [1] CWA 14170:2004 E – Security requirements for signature creation applications
- [2] CWA 14171:2004 D/E/F – General guidelines for electronic signature verification
- [3] W3C/IETF Recommendation: "XML-Signature Syntax and Processing" v2002-02-12 (XMLDSIG)
- [4] ETSI TS 101 903 – XML Advanced Electronic Signatures (XAdES) v1.4.2
- [5] ETSI TS 103 171 – Electronic Signatures and Infrastructures (ESI) XAdES Baseline Profile, v2.1.1
- [6] ETSI TS 101 733 – Electronic Signatures and Infrastructures (ESI); CMS Advanced Electronic Signatures (CAAdES), v2.2.1
- [7] ETSI TS 103 173 – Electronic Signatures and Infrastructures (ESI); CAAdES Baseline Profile, v2.2.1
- [8] ETSI TS 102 918 – Electronic Signatures and Infrastructures (ESI); Associated Signature Containers (ASiC), v1.3.1
- [9] ETSI TS 103 174 – Electronic Signatures and Infrastructures (ESI) ASiC Baseline Profile, v2.2.1
- [10] Rozhodnutie komisie 2011/130/EU, ktorým sa ustanovujú minimálne požiadavky na cezhraničné spracovanie dokumentov elektronicky podpísaných príslušnými orgánmi v zmysle smernice Európskeho parlamentu a Rady 2006/123/ES o službách na vnútornom trhu
- [11] Rozhodnutie komisie 2014/148/EÚ zo 17. marca 2014, ktorým sa mení rozhodnutie 2011/130/EÚ, ktorým sa ustanovujú minimálne požiadavky na cezhraničné spracovanie dokumentov elektronicky podpísaných príslušnými orgánmi v zmysle smernice Európskeho parlamentu a Rady 2006/123/ES o službách na vnútornom trhu
- [12] Nariadenie Európskeho Parlamentu a Rady EÚ č. 910/2014 z 23. júla 2014 o elektronickej identifikácii a dôveryhodných službách pre elektronické transakcie na vnútornom trhu a o zrušení smernice 1999/93/ES
- [13] Zákon č. 215/2002 Z.z. o elektronickej podpise a o zmene a doplnení niektorých zákonov v znení neskorších predpisov
- [14] Vyhláška NBÚ č. 135/2009 Z.z. o vyhotovení a overovaní elektronického podpisu a časovej pečiatky
- [15] Vyhláška NBÚ č. 136/2009 Z.z. o spôsobe a postupe používania elektronického podpisu v obchodnom styku a administratívnom styku

- [16] Zákon č. 305/2013 o elektronickej podobe výkonu pôsobnosti orgánov verejnej moci a o zmene a doplnení niektorých zákonov (zákon o e-Governmente) v znení neskorších predpisov
- [17] Výnos MF SR č. 55/2014 Z.z. o štandardoch pre informačné systémy verejnej správy v znení neskorších predpisov
- [18] Profil XAdES_ZEPbp v1.0 – formát podpisu na báze XAdES_BP, DITEC, a.s., 2016
- [19] Používateľská príručka SCVA D.Verifier-SVR-XAdES, v3.0, DITEC, a.s., 2016
- [20] Používateľská príručka SCVA D.Verifier-SVR-CAdES, v2.0, DITEC, a.s., 2016
- [21] Integrovaná príručka ASiC Factory, v1.1, DITEC, a.s., 2016
- [22] Document management – Electronic document file format for long-term preservation – Use of PDF 1.4 (PDF/A-1), ISO 19005-1:2005(E)
- [23] Document management – Electronic document file format for long-term preservation – Use of PDF 1.4 (PDF/A-1), TECHNICAL CORRIGENDUM 1, ISO 19005-1:2005/Cor.1:2007(E)
- [24] Extensible Markup Language (XML) 1.0 (Fifth Edition) – <http://www.w3.org/TR/2008/REC-xml-20081126/>
- [25] Information technology – Computer graphics and image processing – Portable Network Graphics (PNG): Functional specification, ISO/IEC 15948:2004
- [26] Profil CAdES_ZEP v2.0 – Formát ZEP na báze CAdES, Ditec, 2015

3. Popis aplikácie

Komponent DataValidator slúži na validáciu dátových objektov podpísaných zaručeným elektronickým podpisom (resp. pečatou), ktoré boli vytvorené v súlade s profilmi XAdESbp [5] a XAdES_ZEPbp [18], resp. CAdESbp [7] a CAdES_ZEP v2.0 [26].

Legislatívne požiadavky na formát a obsah podpísaných dátových objektov špecifikuje výnos MF SR o štandardoch [17] v rámci § 57a Prijímanie a čítanie podpísaných elektronických dokumentov. Z uvedených požiadaviek na podporované formáty dátových objektov sú pre profily XAdES_ZEPbp [18] a CAdES_ZEP v2.0 [26], ktoré majú detached topológiu, relevantné nasledujúce formáty:

- textové súbory Portable Document Format vo verzii A-1a (PDF/A-1a),
- textové súbory (.txt) v kódovaní UTF-8,
- grafické súbory vo formáte Portable Network Graphics (.png),
- elektronické dokumenty vo formáte jazyka pre prenos dátových prvkov XML v1.0 v štruktúre podľa prílohy č. 11 (XMLDataContainer).

Profily XAdESbp [5] a CAdESbp [26] nešpecifikujú žiadne obmedzenia na formáty podpisovaných dátových objektov alebo ich vlastnosti.

Komponent DataValidator tvorí doplnok aplikácií pre overenie ZEP/ZEPe D.Verifier-SVR/XAdES a D.Verifier-SVR/CAdES, ktoré budú prevádzkované v automatizovanom režime, a teda v primeranej miere sa na ne vzťahujú aj bezpečnostné požiadavky, špecifikované v rámci dokumentov [1] a [2] na funkcionality prezentácie podpísaných dátových objektov a kontrolu ich obsahu.

Z funkčného hľadiska komponent DataValidator podporuje nasledujúce scenáre validácie podpísaných dátových objektov:

- validácia na základe informácií evidovaných pre daný typ podpísaného dátového objektu v nadradenej aplikácii; typ podpísaného dátového objektu je určený elementom xades:ObjectIdentifier, ktorý sa musí nachádzať pod elementom xades:DataObjectFormat v rámci štruktúry podpisu XAdES_ZEPbp,
- validácia na základe konfigurácie komponentu DataValidator a podľa príslušnej skupiny formátov dátových objektov, ktorú špecifikuje nadradená aplikácia na základe informácií z vyhodnotenia príslušného ZEP/ZEPe vo formátoch XAdES_ZEPbp, XAdESbp, CAdES_ZEP v2.0 a CAdESbp.

Systémové požiadavky na komponent DataValidator sú nasledujúce:

- operačný systém MS Windows 2008 Server, 2012 Server,
- MS .Net framework, verzia 2.0 – 3.5.

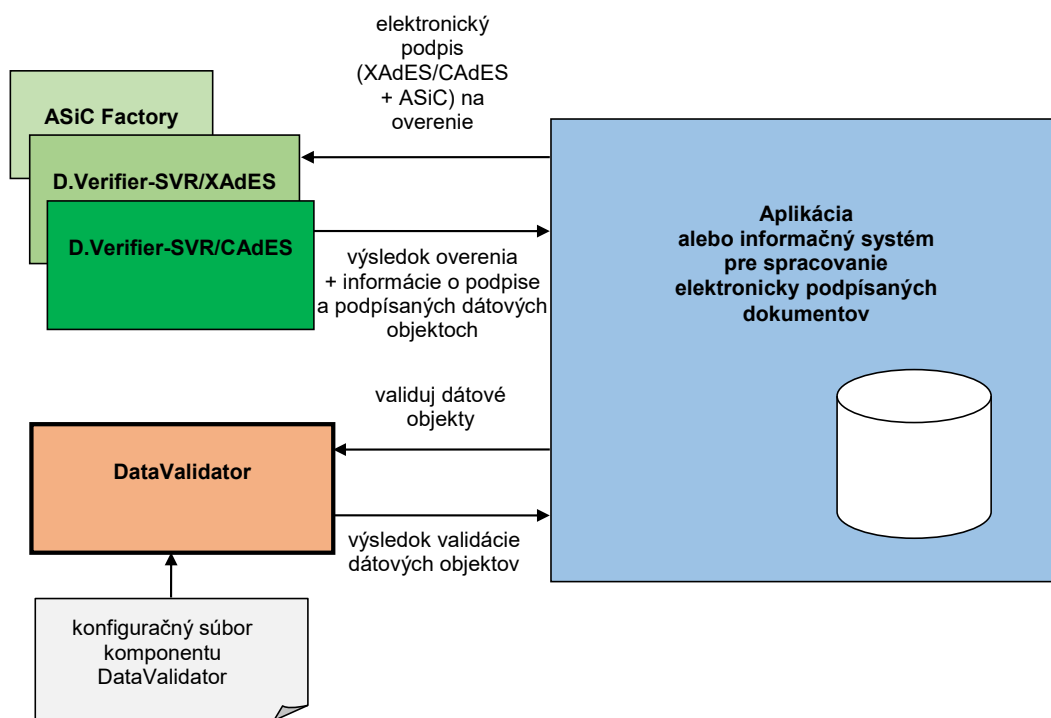
4. Architektúra a postavenie komponentu v rámci prevádzkového prostredia

Komponent DataValidator poskytuje funkcionality validácie dátových objektov podpísaných zaručeným elektronickým podpisom (ZEP), resp. pečatou (ZEPe), ktoré sú vytvorené v súlade s profilmi XAdES_ZEPbp/XAdESbp a CAdES_ZEP v2.0/CAdESbp, pre aplikácie a informačné systémy (IS), ktoré potrebujú spracovávať dátové štruktúry využívajúce autorizačné mechanizmy elektronického podpisu pre rôzne, napr. právne úkony.

Aplikácia alebo IS po overení daného elektronického podpisu prostredníctvom aplikácie D.Verifier-SVR/XAdES [19] alebo D.Verifier-SVR/CAdES bude môcť volaním API komponentu DataValidator overiť, či jednotlivé podpísané dátové objekty spĺňajú požiadavky na formát a obsah, ktoré sú dané legislatívnymi požiadavkami [17], bezpečnostnými požiadavkami špecifikovanými v rámci dokumentov [1] a [2], prípadne ďalšími požiadavkami danými napr. legislatívou príslušnej cieľovej agendy. Na základe výsledku validácie bude môcť volajúca aplikácia riadiť ďalšie spracovanie dátových objektov autorizovaných ZEP/ZEPe.

Z pohľadu architektúry komponent DataValidator tvorí doplnok aplikácií pre overenie ZEP/ZEPe D.Verifier-SVR/XAdES, resp. D.Verifier-SVR/CAdES a prostredníctvom API poskytuje volajúcej aplikácii služby pre validáciu podpísaných dátových objektov. Pre získanie informácií o type ASiC kontajnera [8][9] a v ňom uložených elektronických podpisov bude nadradená aplikácia využívať komponent ASiC Factory [21]. Pre získanie potrebných parametrov pre validáciu dátovej štruktúry XMLDataContainer pomocou komponentu DataValidator bude nadradená aplikácia využívať triedu XMLDataContainer.

Na nasledujúcom obrázku je znázornený pohľad na architektúru a postavenie komponentu DataValidator v rámci jeho prevádzkového prostredia.



5. Špecifikácia funkčnosti

5.1. Scenáre použitia komponentu DataValidator

Nadradená aplikácia alebo informačný systém pre spracovanie elektronicky podpísaných dokumentov:

- 1) vytvorí inštanciu triedy *AsicFactory* [21] a na vstupe metódy pre inicializáciu poskytne kontajner vo formáte ASiC,
- 2) zavolá metódu *CheckContainer* a skontroluje tak korektnosť vstupného kontajnera podľa špecifikácií [8][9],
- 3) zavolaním metódy *GetType* získa informáciu o type ASiC kontajnera,
- 4) ak typ ASiC kontajnera je "XAdES", tak:
 - A) vytvorí inštanciu triedy *DVerifierSvrXadesBP* aplikácie D.Verifier-SVR/XAdES [19] a zavolá funkciu *GetSignatureInformation*, aby tak získala informácie o jednotlivých elektronických podpisoch,
 - B) ďalej zavolá funkciu *VerifySignature* pre overenie zvoleného elektronického podpisu vo formáte XAdES_ZEPbp, resp. XAdESbp,
 - C) zavolá funkciu pre získanie informácií o podpísaných dátových objektoch *GetSignedDataObjectInfos* a získa tak štruktúru *List<DataObjectInfoBP>*,
 - D) získa obsah samotných podpísaných dátových objektov volaním funkcie *GetSignedDataObject* s parametrom *referenceId* pre jednotlivé dátové objekty,
 - E) vyhodnotí informácie z výstupov volaní funkcií *VerifySignature*, *GetSignatureInformation* a *GetSignedDataObjectInfos* a rozhodne o spôsobe validácie podpísaných dátových objektov.
- 5) ak typ ASiC konajnera je "ASiC-s CMS" alebo "ASiC-e CMS", tak:
 - A) zavolaním metódy *GetInfo* získa získa štruktúru *AsicInfo* s informáciami o type ASiC kontajnera a jeho obsahu, t.j. elektronických podpisoch a podpísaných dátových objektoch uložených v konajneri,
 - B) z návratovej štruktúry *AsicInfo* vyberie zo zoznamu *SignatureInfoList* elektronický podpis a pomocou metódy *GetCadesSignature* získa štruktúru *Signature*, ktorá obsahuje CMS štruktúru CAdES podpisu (resp. pečiatky), podpísané dátové objekty, prípadne štruktúru ASiC Manifest (v prípade ASiC-e),
 - C) vytvorí inštanciu triedy *DVerifierSvrCades* aplikácie D.Verifier-SVR/CAdES [20] a na základe identifikovanej pokročilej formy zavolá príslušnú funkciu pre overenie CMS štruktúry CAdES podpisu nad príslušnými dátami (jeden dátový objekt v prípade ASiC-s, resp. štruktúra ASiC Manifest v prípade ASiC-e),

- D) môže zavolať ďalšie potrebné metódy triedy *DVerifierSvrCades* pre získanie ďalších informácií o elektronickom podpise, napr. *GetSigningCertificate*, *GetSignaturePolicyID* atď.
- E) vyhodnotí informácie z výstupov volaní funkcií pre overenie štruktúry podpisu, funkcií pre získanie informácií o elektronickom podpise a informácie o podpísaných dátových objektoch a rozhodne o spôsobe validácie podpísaných dátových objektov.
- 6) Následne nadradená aplikácia na základe získaných informácií o elektronickom podpise (napr. podpisový certifikát, podpisová politika) a podpísaných dátových objektoch zvolí jeden z nasledujúcich scenárov validácie podpísaných dátových objektov:
- A) validácia na základe informácií evidovaných pre daný typ podpísaného dátového objektu v nadradenej aplikácii, pričom:
- ♦ typ podpísaného dátového objektu je určený elementom *xades:ObjectIdentifier*, ktorý sa nachádza pod elementom *xades:DataObjectFormat* v rámci štruktúry podpisu *XAdES_ZEPbp*; zároveň sa nachádza v rámci návratovej štruktúry *DataObjectInfoBP* funkcie *GetSignedDataObjectInfos*, položka *Identifier*,
 - ♦ pre validáciu dátovej štruktúry *XMLDataContainer* nadradená aplikácia získa ďalšie potrebné parametre volaním API triedy *XMLDataContainer*,
 - ♦ popis validácie dátových objektov na základe informácií evidovaných pre daný typ dátového objektu je uvedený v kapitole 5.1.1,
- B) validácia na základe konfigurácie komponentu *DataValidator* a podľa príslušnej skupiny formátov dátových objektov, ktorú špecifikuje nadradená aplikácia na základe informácií z vyhodnotenia príslušného ZEP/ZEPe vo formátoch *XAdES_ZEPbp/XAdESbp*, resp. *CAdES_ZEP v2.0/CAdESbp* teda:
- ♦ ak nie je možné nastaviť validáciu na základe typu podpísaného dátového objektu, tak nadradená aplikácia môže požadovať vykonanie validácie podľa nastavenia konfigurácie komponentu *DataValidator* voči špecifikovanej skupine formátov dátových objektov,
 - ♦ skupinu formátov dátových objektov určí napr. na základe informácií o elektronickom podpise, prípadne na základe ďalších informácií získaných z nadradenej dátovej štruktúry (zložený podpis, elektronické podanie) a pod.
 - ♦ popis validácie na základe konfigurácie komponentu *DataValidator* je uvedený v kapitole 5.1.2.

V nasledujúcich kapitolách sú popísané podrobnosti validácie obsahu dátových objektov v rámci jednotlivých scenárov.

5.1.1. Validácia na základe informácií evidovaných v nadradenej aplikácii

V rámci nadradenej aplikácie alebo informačného systému bude existovať evidencia typov dátových objektov, pričom pre každý typ dátového objektu budú evidované minimálne nasledujúce informácie:

- identifikátor formátu dátového objektu prostredníctvom MimeType,
- identifikátor typu dátového objektu prostredníctvom jednoznačného URI,
- popis typu dátového objektu (textový reťazec).

Ďalšie evidované údaje pre jednotlivé typy dátových objektov sú závislé na formáte dátového objektu. Pre typy dátových objektov podporovaných formátov podľa [17] by mali byť v rámci nadradenej aplikácie evidované:

- pre application/pdf – požadovaná úroveň súladu so špecifikáciou PDF/A-1 [22][23],
- pre application/vnd.gov.sk.xmldatacontainer+xml – XML schéma, typ výslednej prezentácie (MediaDestinationTypeDescription), prezentačná schéma a ďalšie metadáta pre tzv. elektronický formulár [17].

Nadradená aplikácia získala volaním funkcie *GetSignedDataObjectInfos* aplikácie D.Verifier-SVR/XAdES informácie o podpísaných dátových objektoch – štruktúru *List<DataObjectInfoBP>*. Následne pre daný konkrétny dátový objekt získa na základe položky *DataObjectInfoBP.Identifier* evidované informácie pre daný typ dátového objektu. Pre získanie ďalších potrebných parametrov pre validáciu dátovej štruktúry XMLDataContainer využije komponent triedu XMLDataContainer.

Pri validácii dátového objektu na základe evidovaných informácií pre daný typ dátového objektu nadradená aplikácia:

- 7) vytvorí inštanciu triedy *DataValidator*,
- 8) vytvorí objekt triedy *DataObjectInfo* a naplní hodnoty jednotlivých atribútov na základe hodnôt atribútov príslušnej štruktúry *DataObjectInfoBP*,
- 9) následne podľa položky *DataObjectInfoBP.MimeType* zavolá príslušnú funkciu komponentu *DataValidator* pre validáciu obsahu dátových objektov daného formátu a špecifikuje potrebné parametre na základe evidovaných informácií:

- ⇒ *ValidateXMLDataContainer*,
- ⇒ *ValidatePDF*,
- ⇒ *ValidatePNG*,
- ⇒ *ValidateTXT*.

5.1.2. Validácia na základe konfigurácie a skupiny formátov dátových objektov

V rámci konfigurácie komponentu DataValidator budú definované skupiny formátov dátových objektov. V rámci jednej skupiny formátov dátových objektov je možné definovať povolené formáty dátových objektov (z množiny podporovaných formátov dátových objektov, vid' 5.2) a príslušnú úroveň ich validácie. Skupina formátov dátových objektov je identifikovaná svojim názvom.

Takto je možné definovať rôzne skupiny povolených formátov dátových objektov a rôzne úrovne ich validácie pre rôzne procesy spracovania elektronicky podpísaných dokumentov alebo pre rôzne nadradené aplikácie, či informačné systémy.

Napríklad pre účely vytvárania a overovania ZEP/ZEPe v súlade so slovenskou legislatívou bude definovaná skupina SkObjectTypes, ktorá bude obsahovať zoznam podporovaných formátov dátových objektov v súlade s výnosom MF SR o štandardoch [17], pričom pre jednotlivé formáty dátových objektov bude v konfigurácii určená najvyššia požadovaná úroveň ich validácie.

Pri validácii na základe konfigurácie a skupiny formátov dátových objektov nadradená aplikácia špecifikuje v parametroch volania názov skupiny formátov dátových objektov, voči ktorej sa má validácia vykonať, a špecifikuje úroveň validácie. Výsledná úroveň pre konkrétny formát dátových objektov bude daná minimom:

- úrovne určenej pomocou vstupného parametra metódy pre validáciu,
- úrovne určenej pre daný formát dátového objektu konfiguráciou komponentu v rámci príslušnej skupiny formátov dátových objektov.

Komponent DataValidator zvaliduje jednotlivé dátové objekty vzhľadom na špecifikovanú skupinu formátov dátových objektov a vypočítanú úroveň validácie.

Pri validácii na základe konfigurácie a skupiny formátov dátových objektov budú podporované nasledujúce úrovne validácie dátových objektov:

- úroveň 0 – nevykoná sa žiadna validácia dátového objektu,
- úroveň 1 – vykoná sa validácia na úrovni povolených formátov dátových objektov, tzn. skontroluje sa, či MIMEType daného dátového objektu je uvedený v rámci príslušnej skupiny formátov dátových objektov; ak sa MIMEType dátového objektu nachádza v špecifikovanej skupine, tak validácia prebehne úspešne,

- úroveň 2 – okrem validácie na úrovni povolených formátov podľa predchádzajúceho bodu sa vykoná aj validácia obsahu jednotlivých dátových objektov, tzn. vykonajú sa validácie špecifikované v súlade s bezpečnostnými požiadavkami na príslušný typ obsahu dát (ContentType) podľa dokumentov [1] a [2] a validácie požadované legislatívou [17]; validácie pre jednotlivé podporované formáty dátových objektov sú podrobne špecifikované v nasledujúcej kapitole.

Pri validácii dátových objektov na základe konfigurácie a skupiny formátov dátových objektov nadradená aplikácia:

- 7) vytvorí inštanciu triedy *DataValidator*,
- 8) na základe informácií zo štruktúry podpisu zvolí skupinu formátov dátových objektov, voči ktorej sa má validácia vykonať, a špecifikuje úroveň validácie,
- 9) podľa potreby môže zavolať funkciu pre validáciu všetkých dátových objektov referencovaných v danej štruktúre podpisu – *ValidateAllDataObjects* alebo môže volať funkciu *ValidateDataObject* pre validáciu na základe konfigurácie pre dátové objekty jednotlivo.

5.2. Validácia obsahu pre podporované formáty dátových objektov

Komponent *DataValidator* v súlade s výnosom o štandardoch [17] zabezpečuje validáciu obsahu pre nasledujúce podporované formáty dátových objektov:

- textové súbory Portable Document Format vo verzii A-1a (PDF/A-1a); MimeType = application/pdf,
- textové súbory (.txt) v kódovaní UTF-8; MimeType = text/plain; charset=UTF-8,
- grafické súbory vo formáte Portable Network Graphics (.png); MimeType = image/png,
- elektronické dokumenty vo formáte jazyka pre prenos dátových prvkov XML v1.0 v štruktúre podľa prílohy č. 11 (XMLDataContainer); MimeType = application/vnd.gov.sk.xmldatacontainer+xml; charset=UTF-8.

Pri validácii dátových objektov na základe informácií evidovaných v nadradenej aplikácii a pri validácii na základe konfigurácie a skupiny formátov dátových objektov na úrovni 2 budú pre jednotlivé podporované formáty dátových objektov vykonané nasledujúce kontroly:

- application/pdf:
 - ⇒ kontrola, či vstupné dáta predstavujú naozaj PDF dokument,
 - ⇒ kontrola verzie PDF dokumentu na hodnotu 1.4,

- ⇒ kontrola na súlad so špecifikáciou PDF/A-1 [22][23] na základe hodnoty vstupného parametra alebo hodnoty z konfigurácie z príslušnej skupiny formátov dátových objektov (ak je v konfigurácii uvedená); pozn. ak je špecifikovaná hodnota *e_AsDeclaredBySigner*, tak sa vykoná validácia na tej úrovni, ktorá je deklarovaná v rámci PDF dokumentu,
- text/plain:
 - ⇒ kontrola kódovania UTF-8,
 - ⇒ kontrola na nepovolené riadiace znaky pre entitu char (viď <http://www.w3.org/TR/2008/REC-xml-20081126/#charsets> [24]),
- image/png – kontrola, či špecifikované vstupné dáta sú naozaj PNG obrázok, viď [25],
- application/vnd.gov.sk.xmldatacontainer+xml:
 - ⇒ kontrola kódovania UTF-8,
 - ⇒ validácia voči príslušnej XML schéme pre XMLDataContainer, viď príloha č. 11 výnosu o štandardoch [17],
 - ⇒ kontrola hodnôt príslušných elementov a atribútov štruktúry XMLDataContainer voči hodnotám vstupných parametrov (pozn. ak je daný vstupný parameter prázdny, tak sa hodnota príslušného elementu nekontroluje):
 - ♦ atribút XMLData/Identifier – kontrola voči vstupnému parametru *xdIdentifier*,
 - ♦ atribút XMLData/Version – kontrola voči vstupnému parametru *xdcVersion*,
 - ♦ v prípade, že je vyplnený element UsedXSDEmbedded¹, vykoná sa validácia XML dát uložených pod elementom XMLData voči príslušnej schéme z elementu UsedXSDEmbedded,
 - ♦ v prípade, že je vyplnený element UsedPresentationSchemaEmbedded², aplikuje sa daná XSLT transformácia z elementu UsedPresentationSchemaEmbedded na XML dáta z elementu XMLData a podľa hodnoty atribútu UsedPresentationSchemaEmbedded /MediaDestinationTypeDescription sa vykoná:

¹ Tento krok sa vykoná aj pri validácii na základe konfigurácie a skupiny formátov dátových objektov na úrovni 2.

² Tento krok sa vykoná aj pri validácii na základe konfigurácie a skupiny formátov dátových objektov na úrovni 2.

- pre HTML, XHTML – kontrola na nepovolené HTML tagy: applet, script, iframe, link, object,
 - pre TXT – kontrola na nepovolené riadiace znaky pre entitu char (vid' <http://www.w3.org/TR/2008/REC-xml-20081126/#charsets> [24]),
 - v prípade vzniku chyby pri načítaní alebo aplikovaní transformácie, bude vrátený chybový kód a príslušná chybová hláška.
- ♦ v prípade, že je vyplnený element `UsedPresentationSchemaEmbedded`, vykoná sa tiež kontrola nasledujúcich atribútov:
- `UsedPresentationSchemaEmbedded/MediaDestinationTypeDescription` – kontrola voči vstupnému parametru *xslMediaDestinationTypeDescription*,
 - `UsedPresentationSchemaEmbedded/Language` – kontrola voči vstupnému parametru *xslXSLTLanguage*,
 - `UsedPresentationSchemaEmbedded/TargetEnvironment` – kontrola voči vstupnému parametru *xslTargetEnvironment*.
- ♦ V prípade, že je vyplnený element `UsedXSDReference`:
- vykoná sa validácia XML dát uložených pod elementom `XMLData` voči príslušnej XML schéme zo vstupného parametra *xdcUsedXSD*,
 - zároveň sa skontroluje hodnota elementu `UsedXSDReference` voči príslušnému vstupnému parametru *xsdReferenceURI*,
 - na XML schému z príslušného vstupného parametra *xdcUsedXSD* sa aplikuje kanonikalizačný algoritmus špecifikovaný v atribúte `UsedXSDReference/TransformAlgorithm`, pomocou hašovacej funkcie špecifikovanej v atribúte `UsedXSDReference/DigestMethod` sa vypočíta digitálny odtlačok a jeho hodnota sa skontroluje voči hodnote atribútu `UsedXSDReference/DigestValue`,
- ♦ v prípade, že je vyplnený element `UsedPresentationSchemaReference`:
- aplikuje sa daná XSLT transformácia z príslušného vstupného parametra *xdcUsedXSLT* na XML dáta z elementu `XMLData` a podľa hodnoty atribútu `UsedPresentationSchemaReference/MediaDestinationTypeDescription` sa vykoná:

- pre HTML, XHTML – kontrola na nepovolené HTML tagy: applet, script, iframe, link, object,
- pre TXT – kontrola na nepovolené riadiace znaky pre entitu char (viď <http://www.w3.org/TR/2008/REC-xml-20081126/#charsets> [24]),
- v prípade vzniku chyby pri načítaní alebo aplikovaní transformácie, bude vrátený chybový kód a príslušná chybová hláška.
- zároveň sa skontroluje hodnota elementu UsedPresentationSchemaReference voči príslušnému vstupnému parametru *xmlReferenceURI*,
- na XSLT schému z príslušného vstupného parametra *xdcUsedXSLT* sa aplikuje kanonikalizačný algoritmus špecifikovaný v atribúte UsedPresentationSchemaReference/TransformAlgorithm, pomocou hašovacej funkcie špecifikovanej v atribúte UsedPresentationSchemaReference/DigestMethod sa vypočíta digitálny odtlačok a jeho hodnota sa skontroluje voči hodnote atribútu UsedPresentationSchemaReference/DigestValue,
- ♦ v prípade, že je vyplnený element UsedPresentationSchemaReference, vykoná sa tiež kontrola nasledujúcich atribútov:
 - UsedPresentationSchemaReference/MediaDestinationTypeDescription – kontrola voči vstupnému parametru *xmlMediaDestinationTypeDescription*,
 - UsedPresentationSchemaReference/Language – kontrola voči vstupnému parametru *xmlXSLTLanguage*,
 - UsedPresentationSchemaEmbedded/TargetEnvironment – kontrola voči vstupnému parametru *xmlTargetEnvironment*.

V prípade validácie dátových objektov na základe informácií evidovaných v nadradenej aplikácii sa pre každý podpísaný dátový objekt navyše skontrolujú voči príslušným vstupným parametrom nasledujúce informácie z príslušnej dátovej štruktúry *DataObjectInfo*:

- *MimeType* – kontrola hodnoty atribútu štruktúry *DataObjectInfo*, teda či bola zavolaná správna funkcia pre validáciu dátových objektov na základe informácií evidovaných v nadradenej aplikácii,
- *ObjectIdentifier* – kontrola voči vstupnému parametru *objectIdentifier*,
- *ObjectDescription* – kontrola voči vstupnému parametru *objectDescription*.

Pozn. ak je daný vstupný parameter prázdny, tak sa hodnota príslušného atribútu štruktúry *DataObjectInfo* nekontroluje.

5.3. Získanie relevantných informácií zo štruktúry XMLDataContainer

Aby bolo možné vykonať validáciu obsahu dátového objektu vo formáte XMLDataContainer na základe informácií evidovaných v nadradenej aplikácii, je potrebné získať zo štruktúry XMLDataContainer relevantné identifikátory z elementov obsahujúcich referencie schém, prípadne samotný obsah vložených (embedded) schém z príslušných elementov štruktúry XMLDataContainer.

Nadradená aplikácia môže potom na základe týchto identifikátorov získať zo svojej evidencie typov dátových objektov ďalšie vstupné parametre pre validáciu obsahu dátového objektu v štruktúre XMLDataContainer.

Pre získanie hodnôt jednotlivých elementov a atribútov dátovej štruktúry XMLDataContainer môže nadradená aplikácia využívať triedu XMLDataContainer.

6. Špecifikácia API

Komponent DataValidator tvorí DLL knižnica, ktorá pre klientske aplikácie poskytuje .Net API integračné rozhranie.

DataValidator definuje v rámci integračného API:

- triedu pre informácie o dátovom objekte DataObjectInfo,
- triedu pre dátový objekt DataObject,
- triedu pre parsovanie štruktúry XMLDataContainer.

V nasledujúcich častiach je popísané rozhranie a detailný popis jednotlivých metód.

6.1. Integračné API

Pre .Net aplikácie komponent DataValidator publikuje:

Triedu:

`Ditec.Zep.DataValidator.DataValidator`

Metódy a premenné:

```
public DataValidator();
```

```
class DataObjectInfo
{
    string MimeType
    ,   string ObjectIdentifier
    ,   string ObjectDescription
    ,   string Encoding
};
```

```
public int ValidateXMLDataContainer
(   DataObjectInfo doi
,   string objectDescription
,   string objectIdentifier
,   string xdcDoc
,   string xdcIdentifier
,   string xdcVersion
,   string xdcUsedXSD
,   string xsdReferenceURI
,   string xdcUsedXSLT
,   string xslReferenceURI
,   string xslMediaDestinationTypeDescription
,   string xslXSLTLanguage
,   string xslTargetEnvironment
);
```

```
public int ValidatePDF
(   DataObjectInfo doi
,   string objectDescription
,   string objectIdentifier
,   byte[] sourcePDF
,   RequiredConformanceLevel reqLevel
,   string password
);
```

```
public int ValidatePNG
(   DataObjectInfo doi
,   string objectDescription
,   string objectIdentifier
,   byte[] sourcePNG
);
```

```
public int ValidateTXT
(   DataObjectInfo doi
,   string objectDescription
,   string objectIdentifier
,   byte[] sourceTXT
);
```

```
public int ValidateDataObject
(   DataObject obj
,   int dataObjectsValidationLevel
,   string dataObjectsValidationSet
);

public bool ValidateAllDataObjects
(   List<DataObject> objs
,   int dataObjectsValidationLevel
,   string dataObjectsValidationSet
);

public static string GetDataObjectsValidationRules
(   int dataObjectsValidationLevel
,   string dataObjectsValidationSet
);

public string GetErrorMessage();

enum RequiredConformanceLevel
{   e_Level1A
,   e_Level1B
,   e_None
,   e_AsDeclaredBySigner
};

class DataObject
{   string MimeType
,   string DataB64
};
```

Pre získanie hodnôt jednotlivých elementov a atribútov dátovej štruktúry XMLDataContainer môže nadradená aplikácia využívať triedu XMLDataContainer. Pre .Net aplikácie teda komponent DataValidator publikuje:

Triedu:

[Ditec.Zep.DataValidator.XmlDataContainer](#)

Metódy a premenné:

```
int Initialize(string xdcDoc);
int Initialize2(string xdcDocB64);
int InitializeFromByteArray(byte[] xdcData);
int InitializeFromXmlDocument(XmlDocument xdcDoc);

string GetErrorMessage();

string XdcXMLData { get; }
string XdcIdentifier { get; }
string XdcVersion { get; }
string XdcUsedXSD { get; }
string XdcUsedXSLT { get; }
string XsdReferenceURI { get; }
string XsdReferenceTransformAlg { get; }
string XsdReferenceDigestMethod { get; }
string XsdReferenceDigestValue { get; }
string XslReferenceURI { get; }
string XslReferenceTransformAlg { get; }
string XslReferenceDigestMethod { get; }
string XslReferenceDigestValue { get; }
string XslMediaDestinationTypeDescription { get; }
string XslXSLTLanguage { get; }
string XslTargetEnvironment { get; }
bool XdcIncludeRefs { get; }
string XdcNamespaceURI { get; }
```

6.1.1. Popis funkcií a premenných triedy DataValidator

6.1.1.1. Konštruktor triedy DataValidator

Vytvorenie inštancie triedy DataValidator.

6.1.1.2. ValidateXMLDataContainer

Vstupné parametre:

DataObjectInfo doi – informácie o dátovom objekte získané zo štruktúry podpisu XAdES_ZEPbp,

string objectDescription – popis podpísaných XML dát elektronického formulára z evidencie typov dátových objektov,

string objectIdentifier – jednoznačný identifikátor podpísaných XML dát elektronického formulára z evidencie typov dátových objektov,

string xdcDoc – XML štruktúra XMLDataContainer,

string xdcIdentifier – identifikátor elektronického formulára,

string xdcVersion – jednoznačný identifikátor verzie elektronického formulára,
string xdcUsedXSD – XML schéma elektronického formulára,
string xsdReferenceURI – jednoznačný identifikátor XML schémy elektronického formulára,
string xdcUsedXSLT – XSLT prezentačná schéma elektronického formulára,
string xslReferenceURI – jednoznačný identifikátor XSLT prezentačnej schémy elektronického formulára,
string xslMediaDestinationTypeDescription – identifikácia výstupu XSLT transformácie; práve jedna z možností "TXT", "HTML" a "XHTML",
string xslXSLTLanguage – jazyk použitej prezentačnej schémy,
string xslTargetEnvironment – odlišenie nástroja alebo prostredia, pre ktoré je prezentačná schéma určená.

Návratová hodnota:

V prípade úspešnej validácie 0, inak chybový kód.

Popis:

Funkcia skontroluje hodnoty atribútov príslušnej dátovej štruktúry *DataObjectInfo* a vykoná ďalšie validácie vstupnej dátovej štruktúry *XMLDataContainer* (parameter *XDC*) špecifikované pre *MimeType* = *application/vnd.gov.sk.xmldatacontainer+xml* v súlade s kapitolou 5.2.

6.1.1.3. ValidatePDF

Vstupné parametre:

DataObjectInfo doi – informácie o dátovom objekte získané zo štruktúry podpisu XAdES_ZEPbp,

string objectDescription – popis príslušného typu dátového objektu vo formáte PDF z evidencie typov dátových objektov,

string objectIdentifier – jednoznačný identifikátor príslušného typu dátového objektu vo formáte PDF z evidencie typov dátových objektov,

byte[] sourcePDF – vstupný PDF dokument,

RequiredConformanceLevel reqLevel – požadovaná úroveň súladu so špecifikáciou PDF/A-1,

string password – heslo pre prístup k PDF dokumentu, ak je chránený heslom.

Návratová hodnota:

V prípade úspešnej validácie 0, inak chybový kód.

Popis:

Funkcia skontroluje hodnoty atribútov príslušnej dátovej štruktúry *DataObjectInfo* a vykoná ďalšie validácie vstupného PDF dokumentu špecifikované pre MimeType = application/pdf v súlade s kapitolou 5.2.

6.1.1.4. ValidatePNG**Vstupné parametre:**

DataObjectInfo doi – informácie o dátovom objekte získané zo štruktúry podpisu XAdES_ZEPbp,

string objectDescription – popis príslušného typu dátového objektu vo formáte PNG z evidencie typov dátových objektov,

string objectIdentifier – jednoznačný identifikátor príslušného typu dátového objektu vo formáte PNG z evidencie typov dátových objektov,

byte[] sourcePNG – vstupný PNG obrázok,

Návratová hodnota:

V prípade úspešnej validácie 0, inak chybový kód.

Popis:

Funkcia skontroluje hodnoty atribútov príslušnej dátovej štruktúry *DataObjectInfo* a vykoná ďalšie validácie vstupného PNG obrázku špecifikované pre MimeType = image/png v súlade s kapitolou 5.2.

6.1.1.5. ValidateTXT**Vstupné parametre:**

DataObjectInfo doi – informácie o dátovom objekte získané zo štruktúry podpisu XAdES_ZEPbp,

string objectDescription – popis príslušného typu dátového objektu vo formáte TXT z evidencie typov dátových objektov,

string objectIdentifier – jednoznačný identifikátor príslušného typu dátového objektu vo formáte TXT z evidencie typov dátových objektov,

byte[] sourceTXT – vstupný TXT dokument,

Návratová hodnota:

V prípade úspešnej validácie 0, inak chybový kód.

Popis:

Funkcia skontroluje hodnoty atribútov príslušnej dátovej štruktúry *DataObjectInfo* a vykoná ďalšie validácie vstupného TXT dokumentu špecifikované pre MimeType = text/plain v súlade s kapitolou 5.2.

6.1.1.6. ValidateDataObject

Vstupné parametre:

DataObject obj – štruktúra obsahujúca atribút *MimeType* a base64 kódované dáta objektu,

int dataObjectsValidationLevel – úroveň validácie dátových objektov (0 = bez validácie; 1 = validácia na úrovni povolených MIME typov; 2 = validácia na úrovni obsahu dátových objektov),

string dataObjectsValidationSet – názov skupiny formátov dátových objektov z konfigurácie.

Návratová hodnota:

V prípade úspešnej validácie 0, inak chybový kód.

Popis:

Funkcia overí, či parameter *dataObjectsValidationLevel* obsahuje platnú hodnotu. Ak parameter obsahuje hodnotu 0 (bez validácie), tak funkcia vráti 0. Ak obsahuje hodnotu väčšiu ako 2 a menšiu ako 0, tak funkcia vráti chybový kód.

Následne funkcia načíta konfiguráciu komponentu DataValidator a overí, či sa skupina formátov dátových objektov, definovaná v parametri *dataObjectsValidationSet*, nachádza v konfiguračnom súbore. Ak sa daná skupina v konfiguračnom súbore nenachádza, funkcia vráti chybový kód.

Inak funkcia pre referencovaný dátový objekt vypočíta minimálnu úroveň požadovanej validácie na základe konfigurácie a vstupného parametra *dataObjectsValidationLevel* a vykoná nasledovnú kontrolu:

Ak je výsledná hodnota požadovanej validácie

- 0 – nevykoná nič,
- 1 – overí, či sa atribút *MimeType* dátovej štruktúry *DataObject* príslušného dátového objektu nachádza medzi povolenými MimeTypes danej skupiny formátov dátových objektov *dataObjectsValidationSet*,
- 2 – vykoná sa validácia obsahu dátového objektu v súlade s pravidlami popísanými v kapitole 5.2 (pozn. pre *MimeType* = application/vnd.gov.sk.xmldatacontainer+xml sa nevykoná kontrola hodnôt príslušných elementov a atribútov štruktúry XMLDataContainer voči hodnotám vstupných parametrov).

Ak sa referencovaný dátový objekt overí korektne, tak funkcia vráti 0.

6.1.1.7. ValidateAllDataObjects

Vstupné parametre:

List<DataObject> objs – zoznam štruktúr obsahujúcich atribút *MimeType* a base64 kódované dáta objektov,

int dataObjectsValidationLevel – úroveň validácie dátových objektov (0 = bez validácie; 1 = validácia na úrovni povolených MIME typov; 2 = validácia na úrovni obsahu dátových objektov),

string dataObjectsValidationSet – názov skupiny formátov dátových objektov z konfigurácie.

Návratová hodnota:

V prípade úspešnej validácie true, inak false.

Popis:

Funkcia overí, či parameter *dataObjectsValidationLevel* obsahuje platnú hodnotu. Ak parameter obsahuje hodnotu 0 (bez validácie), tak funkcia vráti true. Ak obsahuje hodnotu väčšiu ako 2 a menšiu ako 0, tak funkcia vráti false a nastaví chybovú hlášku.

Následne funkcia načíta konfiguráciu komponentu DataValidator a overí, či sa skupina formátov dátových objektov, definovaná v parametri *dataObjectsValidationSet*, nachádza v konfiguračnom súbore. Ak sa daná skupina v konfiguračnom súbore nenachádza, funkcia vráti false a nastaví chybovú hlášku.

Inak funkcia pre referencovaný dátový objekt vypočíta minimálnu úroveň požadovanej validácie na základe konfigurácie a vstupného parametra *dataObjectsValidationLevel* a vykoná nasledovnú kontrolu:

Ak je výsledná hodnota požadovanej validácie

- 0 – nevykoná nič,
- 1 – overí, či sa atribút *MimeType* dátovej štruktúry *DataObject* príslušného dátového objektu nachádza medzi povolenými MimeTypes danej skupiny formátov dátových objektov *dataObjectsValidationSet*; zistené chyby sa vrátia v rámci textu chybovej hlášky,
- 2 – vykoná sa validácia obsahu všetkých dátových objektov v súlade s pravidlami popísanými v kapitole 5.2; zistené chyby sa vrátia v rámci textu chybovej hlášky (pozn. pre *MimeType* = application/vnd.gov.sk.xmldatacontainer+xml sa nevykoná kontrola hodnôt príslušných elementov a atribútov štruktúry XMLDataContainer voči hodnotám vstupných parametrov).

Ak sa referencované dátové objekty overia korektne, tak funkcia vráti true, inak false.

6.1.1.8. **GetDataObjectsValidationRules**

vstupné parametre:

int *dataObjectsValidationLevel* – úroveň validácie dátových objektov (0 = bez validácie; 1 = validácia na úrovni povolených MIME typov; 2 = validácia na úrovni obsahu dátových objektov),

string *dataObjectsValidationSet* – názov skupiny formátov dátových objektov pre validáciu.

návratová hodnota:

textový reťazec

popis:

Metóda overí, či parameter *dataObjectsValidationLevel* obsahuje platnú hodnotu. Ak obsahuje hodnotu väčšiu ako 2 a menšiu ako 0, tak metóda vráti NULL. Následne metóda načíta konfiguráciu komponentu DataValidator a overí, či sa skupina formátov dátových objektov, definovaná v parametri *dataObjectsValidationSet*, nachádza v konfiguračnom súbore. Ak sa daná skupina v konfiguračnom súbore nenachádza, metóda vráti NULL. Inak metóda vytvorí a vráti XML dokument, v ktorom sa nachádza zoznam povolených MIME typov a k nim priradených úrovní validácie, ktoré sa použijú (alebo boli použité) pri validácii.

6.1.1.9. **GetErrorMessage**

V prípade výskytu chyby v rámci procesu validácie dátového objektu (resp. objektov) bude obsahovať príslušnú chybovú správu.

6.1.1.10. **Konštruktor triedy XMLDataContainer**

Vytvorenie inštancie triedy XMLDataContainer.

6.1.1.11. **Initialize**

Inicializácia triedy XmlDataContainer.

Parametre:

string xdcDoc – XML dokument v štruktúre XMLDataContainer ako textový reťazec.

Návratová hodnota:

V prípade úspechu 0; inak chybový kód.

6.1.1.12. **Initialize2**

Inicializácia triedy XmlDataContainer.

Parametre:

string xdcDocB64 – XML dokument v štruktúre XMLDataContainer ako bytestream v UTF-8 kódovaný v base64.

Návratová hodnota:

V prípade úspechu 0; inak chybový kód.

6.1.1.13. InitializeFromArray

Inicializácia triedy XmlDataContainer.

Parametre:

byte[] xdcData – XML dokument v štruktúre XMLDataContainer ako bytestream v UTF-8.

Návratová hodnota:

V prípade úspechu 0; inak chybový kód.

6.1.1.14. InitializeFromXmlDocument

Inicializácia triedy XmlDataContainer.

Parametre:

XmlDocument xdcDoc – rozparsovaný XML dokument v štruktúre XMLDataContainer ako objekt .NET triedy XmlDocument.

Návratová hodnota:

V prípade úspechu 0; inak chybový kód.

6.1.1.15. GetErrorMessage

V prípade neúspechu inicializácie triedy XMLDataContainer vráti príslušnú chybovú správu.

6.1.1.16. XdcXMLData

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu elementu XMLData (dáta elektronického formulára).

6.1.1.17. XdcIdentifier

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu atribútu XMLData/Identifier (jednoznačný identifikátor elektronického formulára).

6.1.1.18. XdcVersion

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu atribútu XMLData/Version (verzia elektronického formulára).

6.1.1.19. XdcUsedXSD

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu elementu UsedXSDEmbedded (XML schéma elektronického formulára).

6.1.1.20. XdcUsedXSLT

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu elementu UsedPresentationSchemaEmbedded (prezentačná schéma elektronického formulára).

6.1.1.21. XsdReferenceURI

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu elementu UsedXSDReference (URI XML schémy elektronického formulára).

6.1.1.22. XsdReferenceTransformAlg

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu atribútu UsedXSDReference/TransformAlgorithm (URI použitého kanonikalizačného algoritmu pred výpočtom digitálneho odtlačku XML schémy elektronického formulára).

6.1.1.23. XsdReferenceDigestMethod

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu atribútu UsedXSDReference/DigestMethod (URN OID použitého algoritmu digitálneho odtlačku pre výpočet hodnoty referencie XML schémy elektronického formulára).

6.1.1.24. XsdReferenceDigestValue

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu atribútu UsedXSDReference/DigestValue (hodnota digitálneho odtlačku referencie XML schémy elektronického formulára).

6.1.1.25. XslReferenceURI

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu elementu UsedPresentationSchemaReference (URI prezentačnej schémy elektronického formulára).

6.1.1.26. XslReferenceTransformAlg

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu atribútu UsedPresentationSchemaReference/TransformAlgorithm (URI použitého kanonikalizačného algoritmu pred výpočtom digitálneho odtlačku prezentačnej schémy elektronického formulára).

6.1.1.27. XslReferenceDigestMethod

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu atribútu UsedPresentationSchemaReference/DigestMethod (URN OID použitého

algoritmu digitálneho odtlačku pre výpočet hodnoty referencie prezentačnej schémy elektronického formulára).

6.1.1.28. XslReferenceDigestValue

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu atribútu UsedPresentationSchemaReference/DigestValue (hodnota digitálneho odtlačku referencie prezentačnej schémy elektronického formulára).

6.1.1.29. XslMediaDestinationTypeDescription

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu atribútu UsedPresentationSchemaEmbedded/MediaDestinationTypeDescription, resp. UsedPresentationSchemaReference/MediaDestinationTypeDescription (cieľové prostredie, pre ktoré je prezentačná schéma určená).

6.1.1.30. XslXSLTLanguage

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu atribútu UsedPresentationSchemaEmbedded/Language, resp. UsedPresentationSchemaReference/Language (jazyk použitej prezentačnej schémy).

6.1.1.31. XslTargetEnvironment

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu atribútu UsedPresentationSchemaEmbedded/TargetEnvironment, resp. UsedPresentationSchemaReference/TargetEnvironment (cieľové prostredie, pre ktoré je prezentačná schéma určená).

6.1.1.32. XdcIncludeRefs

V prípade úspechu inicializácie triedy XMLDataContainer vráti true, ak boli do štruktúry XMLDataContainer zahrnuté referencie schém prostredníctvom elementu UsedSchemasReferenced; false, ak boli do štruktúry XMLDataContainer zahrnuté hodnoty schém prostredníctvom elementu UsedSchemasEmbedded.

6.1.1.33. XdcNamespaceURI

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu atribútu xmlns elementu XMLDataContainer.

7. Konfigurácia komponentu

V rámci konfiguračného súboru komponentu DataValidator budú uložené údaje v súlade s nižšie definovanou štruktúrou, pričom znak * signalizuje, že príslušná položka sa môže vyskytovať ľubovoľný počet krát (0 a viac), ? signalizuje, že príslušná položka je nepovinná a hlbšia úroveň odrážok definuje obsah štrukturovanej položky:

- Ditec.Zep.DataValidator – hlavná sekcia
 - ⇒ ObjectsValidation – sekcia pre konfiguráciu validácie formátov dátových objektov a ich obsahu
 - ◆ ObjectTypeSets – množina validačných skupín formátov dátových objektov
 - ObjectTypeSet* – validačná skupina formátov dátových objektov
 - Name – názov skupiny
 - ObjectType* – formát dátového objektu
 - MimeType – MIME typ dátového objektu
 - Level – úroveň validácie daného dátového typu (povolené hodnoty: 0 - žiadna kontrola, 1 - kontrola MIME typu objektu, 2 - kontrola obsahu)
 - Parameters? – dodatočné parametre pre validáciu úrovne 2; napr. pre PDF dokumenty je možné špecifikovať požadovanú úroveň súladu so špecifikáciou PDF/A-1[22][23] – možnosti reqLevel = 'No_PDF/A-1_Conformance', 'PDF/A-1_LevelAConformance', 'PDF/A-1_LevelBConformance'.
 - ◆ XmlDataContainerSchema – XSD schéma pre XmlDataContainer
 - version – URI schémy s označením verzie.

8. Návrátové kódy aplikácie

V nasledujúcej tabuľke sú uvedené návratové kódy a príslušné návratové hlášky komponentu DataValidator.

Návratový kód	Popis
0	"Validácia bola vykonaná úspešne."
1	"Nesprávna validácia podľa MimeType."
2	"Chybný popis dátového objektu (object description)."
3	"Chybné OID dátového objektu (object identifier)."
4	"Chybná štruktúra dokumentu voči XSD."
5	"Xml dátový kontajner je neznámej verzie."
6	"Xml dátový kontajner sa nepodarilo načítať."
7	"Kontajner XML údajov (XMLDataContainer) obsahuje identifikátor elektronického formulára, ktorý je odlišný od identifikátora uvedeného v module elektronických formulárov."
8	"Xml dátový kontajner má chybnú verziu."
9	"HTML dokument obsahuje nepovolené html tagy."
10	"Xml dátový kontajner má nesprávny xsiMediaDestinationTypeDescription."
11	"Xml dátový kontajner má nesprávny xsiXSLTLanguage."
12	"Xml dátový kontajner má nesprávny xsiTargetEnvironment."
13	"Xml dátový kontajner má nesprávny xsdReferenceURI."
14	"Xml dátový kontajner má nesprávny xsiReferenceURI."
15	"Výpočet hash odtlačku nie je možné vykonať nad prázdny dokumentom."
16	"Neznámy transformačný algoritmus."
17	"Xml dátový kontajner má nesprávnu xsiReferenceDigestValue."
18	"Xml dátový kontajner má nesprávnu xsdReferenceDigestValue."

19	"TXT dokument nezodpovedá kódovaniu UTF-8."
20	"TXT dokument obsahuje nepovolené znaky."
21	"Dátový objekt nie je typu PNG obrázok."
22	"Dátový objekt nie je typu PDF dokument."
23	"Chybné heslo pre prístup k PDF dokumentu."
24	"Chyba pri vyhodnocovaní súladu so špecifikáciou PDF/A-1."
25	"PDF dokumentu nie je verzie 1.4."
26	"Parameter dataObjectsValidationLevel obsahuje nepovolenú hodnotu."
27	"Parameter dataObjectsValidationSet obsahuje neznámu hodnotu."
28	"Neznámy mimetype pre požadovaný dataObjectsValidationSet."
29	"Neznámy mimetype pre ValidateDataObject."
30	"Dátový objekt obsahuje referenciu na schému, ktorá nie je uvedená na vstupe."
31	"Vstupná schéma xdcUsedXSD nepatrí k vstupnému dátovému objektu."
32	"Neznámy hash algoritmus."
33	"Dátový objekt obsahuje referenciu na transformáciu, ktorá nie je uvedená na vstupe."
34	"Parameter xdcDoc neobsahuje xml dokument."
35	"Chyba pri načítaní alebo vykonaní XSL transformácie."
-1	"Neznáma chyba: <ex.Message>"

Ostatné funkcie API komponentu DataValidator vrátia v prípade chyby prázdny string, resp. hodnotu Null (v závislosti od typu návratovej hodnoty).