DITEC
Data Information Technology & Expert Consulting

Integračná príručka

D.Bridge JS, v1.x

Copyright

Popisné charakteristiky dokumentu

Projekt	GOV_ZEP		A3019_002
Dokument	Integračná príručka		
Podnázov	D.Bridge JS, v1.x		
Ref. číslo	GOV_ZEP.270	Verzia	2

Vypracoval	Róbert Vittek, Martin Gašparík	Podpis	Dátum 26. 10. 2023
Preveril	Martin Gašparík	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.270	Verzia	2

Záznamy o zmenách

Autor	Popis zmien	Dátum	Verzia

Pripomienkovanie a kontrola

Autor	Stanovisko	Dátum	Verzia

Rozdeľovník

	Priezvisko Meno	Firma, Funkcia
Originál		
Kópia		
Kópia		
Kópia		

Obsah

1. Úvod

Tento dokument je určený pre integrátorov aplikácií pre kvalifikovaný elektronický podpis (KEP), ktoré budú integrované v rámci portálových riešení informačných systémov pomocou komponentu D.Bridge JS, v1.x:

- D.Signer/XAdES – aplikácia pre vytváranie zaručeného elektronického podpisu nad rôznymi typmi dátových objektov,
- D.Sig XAdES Extender – aplikácia pre vytváranie nadstavbových dátových štruktúr obsahujúcich zaručený elektronický podpis od zložených podpisov až po štruktúry elektronických podaní,
- ASiC Factory – aplikácia pre vytváranie a spracovanie ASiC kontajnera, ktorý slúži pre spojenie štruktúr elektronických podpisov a podpísaných dátových objektov,
- D.Viewer – aplikácia pre prezeranie dátových štruktúr slúžiacich na elektronickú výmenu dát, najmä elektronických podaní a elektronických úradných dokumentov, podpísaných zaručeným elektronickým podpisom,
- D.GINA – aplikácia pre autentifikáciu používateľov pomocou elektronického podpisu.

Dokument popisuje spôsob integrácie uvedených aplikácií do web stránok portálu informačného systému prostredníctvom javascript knižnice D.Bridge JS, v1.x, ktorá pre obe verzie aplikácií (.NET aj Java) definuje unifikované aplikačné rozhranie.

Jednotlivé časti dokumentácie je možné použiť pri tvorbe dokumentácie týchto informačných systémov po dohode s vlastníkmi autorských práv komponentu D.Bridge JS.

Knižnice D.Bridge JS sú súčasťou produktu D.Integration Suite.

2. Zoznam použitých skratiek

API – Application Programming Interface

ASiC – Associated Signature Container

JNLP – Java Network Launching Protocol

KEP – Kvalifikovaný elektronický podpis

ÚPVS – Ústredný portál verejnej správy

URL – Unified Resource Locator

XAdES – XML Advanced Electronic Signatures

3. Referencie

[1] Integračné príručky pre D.Signer/XAdES .NET, v4.0, DITEC, a.s., 2016
[2] Používateľská príručka D.Signer/XAdES .NET, v4.0, DITEC, a.s., 2016
[3] Integračné príručky pre D.Signer/XAdES Java, v2.0, DITEC, a.s., 2016
[4] Používateľská príručka D.Signer/XAdES Java, v2.0, DITEC, a.s., 2016
[5] Integračné príručky pre D.Sig XAdES Extender .NET, v4.0, DITEC, a.s., 2016
[6] Integračná príručka D.Sig XAdES Extender Java, v2.0, DITEC, a.s., 2016
[7] Integračné príručky pre ASiC Factory .NET, v1.1, DITEC, a.s., 2016
[8] Integračná príručka ASiC Factory Java, v1.0, DITEC, a.s., 2016
[9] Integračné príručky pre D.Viewer .NET, v4.0, DITEC, a.s., 2016
[10] Používateľská príručka D.Viewer .NET, v4.0, DITEC, a.s., 2016
[11] Integračná príručka D.Viewer Java, v2.0, DITEC, a.s., 2016
[12] Používateľská príručka D.Viewer Java, v2.0, DITEC, a.s., 2016
[13] Integračná príručka D.GINA Java, v2.0, DITEC, a.s., 2016
[14] Používateľská príručka D.GINA Java, v2.0, DITEC, a.s., 2016
[15] Používateľská príručka D.Launcher, v1.x, DITEC, .a.s., 2018
[16] Používateľská príručka D.Launcher, v2.x, DITEC, a.s., 2022

4. Architektúra

D.Bridge JS, v1.x je knižnica, ktorá uľahčuje integráciu komponentov a aplikácii určených na prácu so zaručeným elektronickým podpisom do webovej stránky. Hlavnou úlohou je detegovať prostredie používateľa (operačný systém, prehliadač, nainštalované komponenty a pod.) a na základne týchto informácii inštancovať správnu verziu komponentu. Ďalšou úlohou je zabezpečiť jednotné API, aby nebolo potrebné riešiť ktorá verzia komponentu bola použitá.

Knižnica je distribuovaná ako sada súborov, ktoré je potrebné vložiť do webovej stránky, ideálne z centrálneho úložiska, v tomto poradí:

- config.js – obsahuje predvolené hodnoty pre niektoré parametre metód (hlavne metóda deploy), ktoré sú špecifické pre konkrétne úložisko. Obsahuje najmä URL jednotlivých JNLP súborov,
- dCommon.min.js – obsahuje spoločné časti kódu,
- dSigXades.min.js a dSigXadesBp.min.js – obsahujú kód potrebný na integráciu aplikácie D.Signer/XAdES,
- dSigXadesExtender.min.js – obsahuje kód potrebný na integráciu aplikácie D.Sig XAdES Extender,
- dViewer.min.js – obsahuje kód potrebný na integráciu aplikácie D.Viewer,
- dGina.min.js – obsahuje kód potrebný na integráciu aplikácie D.GINA.

Predtým než sa začne daný komponent používať, je potrebné vykonať nasadenie zavolaním metódy deploy. Táto metódy vykonáva nasledujúce úkony:

- detekcia prostredia, v ktorom knižnica beží a vyradenie komunikačných rozhraní ktoré v danom prostredí nie je možné používať. Napr. ak knižnica nie je spustená na OS Windows, odstráni sa podpora pre .NET verziu produktov.
- postupné skúšanie nasadenia podporovaných komunikačných rozhraní, dokiaľ sa nasadenie nepodarí, alebo sa vyčerpajú všetky možnosti.

4.1. Komunikačné rozhrania

Knižnica momentálne podporuje tri spôsoby komunikácie s externou aplikáciou. Ktorý z týchto spôsobov sa použije, závisí od toho v akom prostredí knižnica beží.

Pre aktuálnu verziu komponentu D.Bridge JS, v1.x je implementácia nasledovná:

1) ak je na danom počítači nainštalovaná aplikácia D.Launcher, v2.x a zároveň sú splnené systémové požiadavky pre jej prevádzku, a vo webovom prehliadači nainštalované rozšírenie D.Bridge 2, tak na spustenie príslušného komponentu pre KEP sa použije aplikácie D.Launcher, v2.x a na komunikáciu prostredníctvom technológie Native Messaging sa použije rozšírenie D.Bridge 2,
2) ak knižnica beží v prehliadači Microsoft Internet Explorer a knižnici sa podarí detegovať nainštalovaný Java plugin, na komunikáciu sa použije Java applet verzia požadovaného komponentu. V prípade zlyhania detekcie sa použije .NET verzia komponentu (s využitím technológie ActiveX).
3) v ostatných prehliadačoch komunikácia prebieha pomocou aplikácie D.Launcher, v1.x, ktorú musí mať používateľ vopred nainštalovanú. Tento spôsob sa použije aj v prípade že, je použitý prehliadač Internet Explorer 10 / 11 a kroky z bodu 1) zlyhali. Automatické spustenie aplikácie D.Launcher, v1.x je zabezpečené zaregistrovaním vlastnej URL schémy „ditec-dlauncher“. Následná komunikácia prehliadača s touto aplikáciu a požadovaným komponentom prebieha pomocou technológie WebSocket. V prípade, že je dostupná aplikácia Java Web Start, použije sa Java verzia komponentu, v opačnom prípade .NET verzia (iba Windows).

   Spôsob vytvorenia spojenia a zabezpečenia komunikácie medzi web stránkou prehliadača a komponentami pre zaručený elektronický podpis prostredníctvom aplikácie D.Launcher, v1.x je popísaný v rámci používateľskej príručky aplikácie D.Launcher, v1.x [15].

5. Systémové požiadavky

Pre správne fungovanie knižníc komponentu D.Bridge JS, v1.x musia byť splnené nasledujúce požiadavky:

- nainštalovaná aplikácia D.Launcher, v1.x alebo v2.x[1],
- systémové požiadavky špecifikované v používateľských, resp. integračných príručkách pre jednotlivé KEP komponenty (pozri kapitolu 3).

Podporované sú nasledujúce webové prehliadače:

- MS Internet Explorer v7.0 alebo vyššia (IE 7/8/9 len 32 bit, IE 10/11 32 aj 64 bit), Mozilla Firefox, v45 alebo vyššia (prípadne IcedWeasel), Safari 14, 15, Google Chrome v51 alebo vyššia (prípadne Chromium), Opera v38 alebo vyššia, MS Edge v25 alebo vyššia.

V rámci web stránok portálu odporúčame referencovať knižnice komponentu D.Bridge JS, v1.x, ktoré sú umiestnené na portáli ÚPVS (informácia o URL pre jednotlivé súbory je uvedená na stránkach pre integrátorov na portáli ÚPVS), alebo odporúčame umiestniť knižnice komponentu D.Bridge JS, v1.x v rámci centrálneho úložiska príslušného webového portálu.

[1] Aplikácia D.Launcher nie je potrebná len v prípade, že sa webová stránka zobrazuje cez webový prehliadač MS Internet Explorer, kedy je príslušný KEP komponent volaný prostredníctvom Java applet API alebo s využitím technológie ActiveX.

6. Integračné API

Rozhranie jednotlivých objektov je koncipované ako asynchrónne. Každá metóda (okrem pár výnimiek, pozri nižšie) musí obsahovať ako posledný parameter callback. Callback by mal byť objekt, ktorý obsahuje metódy onSuccess a onError. V prípade, že operácia skončila úspešne, je zavolaná metóda onSuccess, pričom výsledok operácie je poskytnutý ako parameter tejto metódy. V prípade chyby je zavolaná metóda onError. Dôvod chyby je uvedený ako prvý parameter. V prípade, ak volanie pomocnej funkcie ditec.utils.isDitecError(e), (kde e je dôvod z onError) vráti true, ide o ošetrenú výnimku. V tomto prípade je e inštancia javascript triedy Error, rozšírená o tieto vlastnosti (nie všetky musia byť vyplnene):

- code – návratový kód operácie (pozri kapitolu 7),
- message – chybová hláška; väčšinou získaná pomocou message/resp. getMessage() z použitej aplikácie,
- detail – doplňujúci údaj určený pre ladenie, napr. stacktrace.

Ukážka:

```
ditec.dSigXadesJs.getSignedXmlWithEnvelope({
    onSuccess : function(ret) {
        //v ret je vysledok volania
},

onError : function(e) {
    if (ditec.utils.isDitecError(e)) {
        // e instanceof Error == true
        // v e.code je návratovy kod volania
        // v e.message je chybova hlaska
        // v e.detail je doplnujuci udaj
    } else {
        // e je neznama chyba
    }
});
```

Všetky metódy okrem metód deploy, log a detectSupportedPlatforms, môžu byť volané len v prípade že bola najprv úspešne zavolaná metóda deploy. V čase od zavolania ľubovoľnej metódy objektu do jej ukončenia, t.j. dokiaľ nie je zavolaný callback, nie je dovolené vykonávať žiadne iné operácie na danom objekte.

6.1. Vlastné metódy

6.1.1. deploy

Úlohou metódy je inicializovať a spustiť príslušný komponent alebo aplikáciu. Ide o prvú metódu, ktorú je potrebné zavolať pred prácou s API príslušného komponentu.

Počas inicializácie a spúšťania daného komponentu pomocou metódy deploy odporúčame zobraziť informačné okno, kde bude používateľ informovaný o vykonávanej akcii, prípadne mu môžu byť zobrazené ďalšie informácie, ako napr. minimálne systémové požiadavky pre jeho operačné prostredie alebo odkaz na web stránku, kde môže tieto informácie získať. Na to je možné využiť nasledujúce konfiguračné premenné knižníc D.Bridge JS:

```
ditec.config.downloadPage = {
        url : "https://www.slovensko.sk/sk/na-stiahnutie",
        title : "slovensko.sk"
    };
```

vstupné parametre:

- options – objekt s konfiguračnými parametrami. Môže obsahovať parametre (všetky sú voliteľné):
  - ⇒ "platforms" – pole podporovaných spôsobov komunikácie (pozri popis metódy detectSupportedPlatfroms nižšie). Predvolená hodnota je null, tj. podpora všetkých dostupných spôsobov. Pri nasadzovaní sa pole postupne prechádza zľava doprava, dokiaľ sa nasadenie nepodarí, alebo dokiaľ sa nevyčerpajú všetky možnosti,

    Upozornenie! V prípade, že ste doteraz na webovom portáli používali staršiu verziu knižníc D.Bridge JS, v1.0 a pomocou tohto parametra ste zmenili predvolené nastavenie podporovaných spôsobov komunikácie, odporúčame doplniť na začiatok poľa podporovaných spôsobov komunikácie hodnotu "dLauncher2".
  - ⇒ "applet.jnlpUrl" – absolútna adresa k súboru JNLP applet verzie príslušnej aplikácie,
  - ⇒ "dlauncher.jnlpUrl" – absolútna adresa k súboru JNLP D.Launcher, v1.x verzie príslušnej aplikácie,
- callback – nevracia žiadnu hodnotu.

6.1.2. detectSupportedPlatforms

Deteguje, ktoré spôsoby komunikácie podporuje aktuálny webový prehliadač. Metóda nekontroluje, či sú splnené všetky kladené požiadavky pre daný spôsob komunikácie. Napr. ak je prehliadač MS Internet Explorer a beží na OS Windows, detekcia povolí .NET verziu bez ohľadu na to, či je nainštalovaný daný komponent alebo nie. Momentálne sú podporované nasledujúce spôsoby:

- “dLauncher2“ – spustenie KEP komponentu prostredníctvom aplikácie D.Launcher, v2.x,
- “java“ – Java verzia prostredníctvom java appletu,

- “dotNet“ – .NET verzia prostredníctvom ActiveX[2],
- “dLauncherJava“ – Java verzia prostredníctvom aplikácie D.Launcher, v1.x,
- “dLauncherDotNet“ – .NET verzia prostredníctvom aplikácie D.Launcher, v1.x[3].

vstupné parametre:

- platforms – pole hodnôt (pozri vyššie), ktorých podpora sa má kontrolovať. V prípade, že nie je uvedené, alebo je null, použije sa predvolená hodnota. Pre aktuálnu verziu komponentu D.Bridge JS, v1.x je to [“dLauncher2“, “java“, “dotNet“, “dLauncherJava“, “dLauncherDotNet“],
- callback – vracia kópiu pôvodného poľa, z ktorého sú odstránene spôsoby komunikácie, ktoré aktuálny prehliadač nepodporuje.

6.1.3. log

Interná metóda, ktorá slúži na logovanie ladiacich informácii. Dodávaná implementácia loguje pomocou metódy console.log(), ak ju prehliadač podporuje. V prípade, že integrujúca aplikácia používa svoj systém, prípadne logovanie nie je potrebné, je možné ju nahradiť vlastnou implementáciou. Táto metóda je synchrónna, neobsahuje callback.

vstupné parametre:

- msg – text, ktorý sa má zalogovať,
- o – doplňujúci objekt, napr. výnimka.

6.2. Integračné API pre D.Signer/XAdES

Komponent D.Bridge JS definuje pre aplikáciu D.Signer/XAdES:

Objekt:

```
ditec.dSigXadesJs
```

Konštanty:

```
DSigXadesJs.SHA1 = "http://www.w3.org/2000/09/xmldsig#sha1";
DSigXadesJs.SHA256 = "http://www.w3.org/2001/04/xmlenc#sha256";
DSigXadesJs.SHA384 = "http://www.w3.org/2001/04/xmldsig-more#sha384";
DSigXadesJs.SHA512 = "http://www.w3.org/2001/04/xmlenc#sha512";
```

[2] Integrácia a spúšťanie aplikácie D.GINA .NET pomocou knižníc D.Bridge JS a aplikácie D.Launcher nie je podporované.

[3] Integrácia a spúšťanie aplikácie D.GINA .NET pomocou knižníc D.Bridge JS a aplikácie D.Launcher nie je podporované.

```
DsigXadesJs.LANG_SK = "SK";
DsigXadesJs.LANG_EN = "EN";


DSigXadesJs.XML_VISUAL_TRANSFORM_TXT = "TXT";
DSigXadesJs.XML_VISUAL_TRANSFORM_HTML = "HTML";


DSigXadesJs.PDF_CONFORMANCE_LEVEL_1A = 0;
DSigXadesJs.PDF_CONFORMANCE_LEVEL_1B = 1;
DSigXadesJs.PDF_CONFORMANCE_LEVEL_NONE = 2;


DSigXadesJs.ERROR_SIGNING_CANCELLED = 1;
```

Metódy:

```
initialize(callback);
setWindowSize(width, height, callback);
setSigningTimeProcessing(displayGui, includeSigningTime, callback);
setLanguage(language, callback);
setCertificateFilter(filterID, callback);
setRevocationChecking(ocspCheck, crlCheck, ocspCertIdHashAlgorithm,
callback);
loadConfiguration(configsZipBase64, callback);
getVersion(callback);


sign(signatureId, digestAlgUri, signaturePolicyIdentifier, callback);


sign11
(   signatureId
,   digestAlgUri
,   signaturePolicyIdentifier
,   dataEnvelopeId
,   dataEnvelopeURI
,   dataEnvelopeDescr
,   callback
);


sign20
(   signatureId
,   digestAlgUri
,   signaturePolicyIdentifier
,   dataEnvelopeId
,   dataEnvelopeURI
,   dataEnvelopeDescr
,   callback
);
```

```
addXmlObject
(   objectId
,   objectDescription
,   sourceXml
,   sourceXsd
,   namespaceUri
,   xsdReference
,   sourceXsl
,   xslReference
,   callback
);

addXmlObject2
(   objectId
,   objectDescription
,   sourceXml
,   sourceXsd
,   namespaceUri
,   xsdReference
,   sourceXsl
,   xslReference
,   transformType
,   callback
);

addPdfObject
(   objectId
,   objectDescription
,   sourcePdfBase64
,   password
,   objectFormatIdentifier
,   reqLevel
,   convert
,   callback
);

addTxtObject
(   objectId
,   objectDescription
,   sourceTxt
,   objectFormatIdentifier
,   callback
);
```

```
addPngObject
(   objectId
,   objectDescription
,   sourcePngBase64
,   objectFormatIdentifier
,   callback
);


checkPDFACompliance
(   sourcePdfBase64
,   password
,   reqLevel
,   callback
);
convertToPDFA
(   sourcePdfBase64
,   password
,   reqLevel
,   callback
);


getSignatureTimeStampRequestBase64(reqPolicy, digestAlgUri, callback);
getSignatureTimeStampRequest2Base64
(   reqPolicy
,   digestAlgUri
,   nonce
,   certReq
,   extensions
,   callback
);
createXAdESZepT(tsResponseB64, tsCertB64, callback);


getSignedXmlWithEnvelope(callback);
getSignedXmlWithEnvelopeBase64(callback);
getSignedXmlWithEnvelopeGZipBase64(callback);
getSignedXmlWithEnvelopeAndTimeStamp(callback);
getSignedXmlWithEnvelopeAndTimeStampBase64(callback);
getSignedXmlWithEnvelopeAndTimeStampGZipBase64(callback);
```

```
getSignerIdentification(callback);
getSigningTime(callback);
getSigningCertificate(callback);
getConvertedPDFA(callback);
getSignatureTimeStampTokenBase64(callback);
getSignatureTimeStampCert(callback);
getSignatureTimeStampTime(callback);
getTSAIdentification(callback);
```

Objekt:

```
ditec.dSigXadesBpJs
```

Konštanty:

```
DSigXadesBpJs.SHA1 = "http://www.w3.org/2000/09/xmldsig#sha1";
DSigXadesBpJs.SHA256 = "http://www.w3.org/2001/04/xmlenc#sha256";
DSigXadesBpJs.SHA384 = "http://www.w3.org/2001/04/xmldsig-more#sha384";
DSigXadesBpJs.SHA512 = "http://www.w3.org/2001/04/xmlenc#sha512";


DsigXadesBpJs.LANG_SK = "SK";
DsigXadesBpJs.LANG_EN = "EN";


DSigXadesBpJs.XML_MEDIA_DESTINATION_TYPE_DESC_TXT = "TXT";
DSigXadesBpJs.XML_MEDIA_DESTINATION_TYPE_DESC_HTML = "HTML";


DSigXadesBpJs.XML_XDC_NAMESPACE_URI_V1_0
    ="http://data.gov.sk/def/container/xmldatacontainer+xml/1.0";
DSigXadesBpJs.XML_XDC_NAMESPACE_URI_V1_1
    ="http://data.gov.sk/def/container/xmldatacontainer+xml/1.1";


DSigXadesBpJs.PDF_CONFORMANCE_LEVEL_1A = 0;
DSigXadesBpJs.PDF_CONFORMANCE_LEVEL_1B = 1;
DSigXadesBpJs.PDF_CONFORMANCE_LEVEL_NONE = 2;


DSigXadesBpJs.ERROR_SIGNING_CANCELLED = 1;
```

Metódy:

```
initialize(callback);
setWindowSize(width, height, callback);
setLanguage(language, callback);
setCertificateFilter(filterID, callback);
setRevocationChecking(ocspCheck, crlCheck, ocspCertIdHashAlgorithm,
callback);
loadConfiguration(configsZipBase64, callback);
getVersion(callback);
```

```
sign(signatureId, digestAlgUri, signaturePolicyIdentifier, callback);


addXmlObject
(   objectId
,   objectDescription
,   objectFormatIdentifier
,   xdcXMLData
,   xdcIdentifier
,   xdcVersion
,   xdcUsedXSD
,   xsdReferenceURI
,   xdcUsedXSLT
,   xslReferenceURI
,   xslMediaDestinationTypeDescription
,   xslXSLTLanguage
,   xslTargetEnvironment
,   xdcIncludeRefs
,   xdcNamespaceURI
,   callback
);

addXmlObject2
(   objectId
,   objectDescription
,   objectFormatIdentifier
,   xdcXDCB64
,   xdcUsedXSD
,   xdcUsedXSLT
,   callback
);


addPdfObject
(   objectId
,   objectDescription
,   sourcePdfBase64
,   password
,   objectFormatIdentifier
,   reqLevel
,   convert
,   callback
);


addTxtObject
(   objectId
,   objectDescription
,   sourceTxt
,   objectFormatIdentifier
,   callback
);
```

```
addPngObject
(   objectId
,   objectDescription
,   sourcePngBase64
,   objectFormatIdentifier
,   callback
);


checkPDFACompliance
(   sourcePdfBase64
,   password
,   reqLevel
,   callback
);
convertToPDFA
(   sourcePdfBase64
,   password
,   reqLevel
,   callback
);


getSignatureTimeStampRequestBase64(reqPolicy, digestAlgUri, callback);
getSignatureTimeStampRequest2Base64
(   reqPolicy
,   digestAlgUri
,   nonce
,   certReq
,   extensions
,   callback
);
createXAdESZepBpT(tsResponseB64, tsCertB64, callback);


getSignatureWithASiCEnvelopeBase64(callback);
getSignatureAndTimeStampWithASiCEnvelopeBase64(callback);


getSignerIdentification(callback);
getSigningTime(callback);
getSigningCertificate(callback);
getConvertedPDFA(callback);
getSignatureTimeStampTokenBase64(callback);
getSignatureTimeStampCert(callback);
getSignatureTimeStampTime(callback);
getTSAIdentification(callback);
```

6.2.1. Integračné API XMLDataContainer

Komponent D.Bridge JS definuje pre XMLDataContainer:

Objekt:

```
ditec.dSigXadesBpJs.getXMLDataContainer();
```

Metódy:

```
initialize(xdcDoc, callback);
initialize2(xdcDocB64, callback);


getXdcIdentifier(callback);
getXdcVersion(callback);
getXdcXMLData(callback);
getXdcUsedXSD(callback);
getXsdReferenceURI(callback);
getXsdReferenceTransformAlg(callback);
getXsdReferenceDigestMethod(callback);
getXsdReferenceDigestValue(callback);
getXdcUsedXSLT(callback);
getXslReferenceURI(callback);
getXslReferenceTransformAlg(callback);
getXslReferenceDigestMethod(callback);
getXslReferenceDigestValue(callback);
getXslMediaDestinationTypeDescription(callback);
getXslTargetEnvironment(callback);
getXslXSLTLanguage(callback);
getXdcIncludeRefs(callback);
getXdcNamespaceURI(callback);
```

Popis jednotlivých metód je uvedený v rámci integračných príručiek aplikácií D.Signer/XAdES .NET [1] a D.Signer/XAdES Java [3].

6.3. Integračné API pre D.Sig XAdES Extender

Komponent D.Bridge JS definuje pre aplikáciu D.Sig XAdES Extender:

Objekt:

```
ditec.dSigXadesExtenderJs.getExtender();
```

Metódy:

```
createNewDataSignatures
(   inDataEnvelope
,   inURI
,   inID
,   inDescription
,   dataSignaturesVersion
,   callback
);
createNewDataSignatures
(   inDataEnvelope
,   inURI
,   inID
,   inDescription
,   callback
);
addDataEnvelopeToExistingDataSignatures
(   inDataSignatures
,   inDataEnvelope
,   callback
);
getDataEnvelopeInfo(inDataEnvelope, type, callback);
getDataSignaturesInfo(inDataSignatures, type, callback);
verifyDataSignatures(inDataSignatures, callback);
getDocumentCount(callback);
moveToDocument(index, callback);
getDocumentType(callback);
createNewDocumentUnauthorized
(   inURI
,   inID
,   inDescription
,   objectData
,   inObjectID
,   inObjectMimeType
,   inObjectEncoding
,   inObjectIdentifier
,   callback
);
getDocumentUnauthorizedInfo(inDocumentUnauthorized, type, callback);
createNewRegistration
(   inURI
,   inID
,   inDescription
,   inExternalIdentifier
,   inBusinessIdentifier
,   callback
);
getRegistrationInfo(inRegistration, type, callback);
initialize(callback);
openFile(title, filter, readBinary, type, callback);
```

```
insertDataSignatures(inDataSignatures, callback);
getVersion(callback);

getDataSignatures(callback);
getDocumentUnauthorized(callback);
getRegistration(callback);
getRegistrationBase64(callback);
```

Objekt:

```
ditec.dSigXadesExtenderJs.getMessageContainer();
```

Metódy:

```
Initialize
(   messageId
,   senderId
,   recipientId
,   messageType
,   messageSubject
,   senderBusinessReference
,   recipientBusinessReference
,   callback
);
initialize2(mc, callback);


addXMLObject
(   id
,   name
,   description
,   oiClass
,   isSigned
,   mimeType
,   objectData
,   callback
);
addBase64Object
(   id
,   name
,   description
,   oiClass
,   isSigned
,   mimeType
,   objectDataBase64
,   callback
);
```

```
getMessageContainer(callback);
getMessageContainerInfo(type, callback);
getObjectCount(callback);
getObjectInfo(i, type, callback);
getObjectData(i, callback);
getVersion(callback);
```

Popis jednotlivých metód je uvedený v rámci integračných príručiek aplikácií D.Sig XAdES Extender .NET [5] a D.Sig XAdES Extender Java [6].

6.4. Integračné API pre ASiC Factory

Komponent D.Bridge JS definuje pre aplikáciu ASiC Factory:

Objekt:

```
ditec.dSigXadesExtenderJs.getAsicFactory();
```

Metódy:

```
initialize(appIdentification, callback);
initialize2(appIdentification, asicB64, callback);

checkContainer(callback);
joinContainers(asicAB64, asicBB64, callback);

getJoinedAsicB64(callback);
getType(callback);
getInfo(type, callback);
getVersion(callback);
```

Popis jednotlivých metód je uvedený v rámci integračných príručiek aplikácií ASiC Factory .NET [7] a ASiC Factory Java [8].

6.5. Integračné API pre D.Viewer

Komponent D.Bridge JS definuje pre aplikáciu D.Viewer:

Objekt:

```
ditec.dViewerJs
```

Metódy:

```
show(data, showCertificate, expandTree, callback);
show2(data, callback);
getVersion(callback);
setLanguage(language, callback);
```

Popis jednotlivých metód je uvedený v rámci integračných príručiek aplikácií D.Viewer .NET [9] a D.Viewer Java [11].

6.6. Integračné API pre D.GINA

Komponent D.Bridge JS definuje pre aplikáciu D.GINA:

Objekt:

```
ditec.dGinaJs
```

Konštanty:

```
DginaJs.LANG_SK = "SK";
DginaJs.LANG_EN = "EN";

DginaJs.ERROR_SIGNING_CANCELLED = 1;
```

Metódy:

```
initialize(callback);
setLanguage(language, callback);
setCertificateFilter(filterID, callback);
loadConfiguration(configsZipBase64, callback);
getVersion(callback);


authenticate(challengeId, authDataObject, callback);
selectClientCertificate(callback);


getSignedXmlWithEnvelope(callback);
getSignedXmlWithEnvelopeBase64(callback);
getSignedXmlWithEnvelopeGZipBase64(callback);


getSignerIdentification(callback);
getSigningTime(callback);
getSigningCertificate(callback);
getClientCertificateBase64(callback);
```

Popis jednotlivých metód je uvedený v rámci integračnej príručky aplikácie D.GINA Java [13]. Integrácia a spúšťanie aplikácie D.GINA .NET pomocou knižníc D.Bridge JS a aplikácie D.Launcher nie je podporované.

7. Návratové kódy

V nasledujúcej tabuľke sú uvedené návratové kódy komponentu D.Bridge JS.

Návratový kód	Popis
1	Zrušenie operácie.
-200	Všeobecná chyba.
-201	Nie je nainštalovaná niektorá požadovaná súčasť pre komponent D.Bridge JS.
-202	Zlyhalo spustenie niektorej požadovanej súčasti komponentu D.Bridge JS.

Návratové kódy jednotlivých aplikácií pre kvalifikovaný elektronický podpis (KEP), ktoré sú v rámci komponentu D.Bridge JS podporované sú uvedené v príslušných integračných príručkách pre jednotlivé aplikácie (pozri kapitolu 3).