



DS PDF Signer

Lietotāja rokasgrāmata

v.1.1

13.02.2013

Datu Sistēmas, SIA

Zemītāņu iela 2B-211, Rīga, LV-1012, Latvija

Tālrunis: 371-67273912 Fax: 371-67273913 Mailto: info@dsistemas.lv

Web: www.dsistemas.lv

Versijas numurs	Datums	Autors(i)	Izmaiņu apraksts
1.0	10.02.2013	Normunds Kinne	Sākotnējā versija
1.1	19.02.2013	Gatis Žeiris	Papildinātā versija

Saturs

1. <i>PDF</i> parakstīšanas sīklietotne	4
2. <i>PDF</i> parakstīšanas sīklietotne izsaukšana no <i>web</i> aplikācijas	5
3. <i>PDF</i> parakstīšanas sīklietotnes izsaukšanas piemērs no <i>PHP</i> <i>web</i> lapas.....	11
4. <i>PDF</i> parakstīšanas sīklietotnes izsaukšana no komandrindas	13
5. <i>PDF</i> parakstīšanas sīklietotnes izsaukšanas piemēri no <i>komandrindas</i>	14

1. *PDF* parakstīšanas sīklietotne

Parakstīšanas sīklietotne jeb *JAVA* applets ir programmatūras vienums, kas nodrošina *PDF* dokumentu parakstīšanu. Parakstīšanas sīklietotne nodrošina 2 veidu izsaukumus:

- 1) sīklietotne var tikt darbināta *web* pārlūka ietvaros;
- 2) sīklietotne var tikt izsaukta no komandrindas.

Lai būtu iespējams darbināt parakstīšanas sīklietotni *web* pārlūka ietvaros, uz darbstacijas nepieciešams uzstādīt:

- 1) Oracle *JAVA* un *JAVA web plug-in* (v1.7 vai jaunāku versiju);
- 2) *web* pārlūkam ir jāatbalsta *JAVA applet* jeb sīklietotņu ielāde;
- 3) *web* pārlūkam ir jāatbalsta *JavaScript* funkciju izpilde.

Lai būtu iespējams darbināt parakstīšanas sīklietotni no komandrindas, uz darbstacijas nepieciešams uzstādīt:

- 1) Oracle *JAVA* (v1.7 vai jaunāku versiju);

2. PDF parakstīšanas sīklītotne izsaukšana no web aplikācijas

Parakstāmā PDF dokumenta satura izgūšana:

Lai sīklītotne spētu parakstīt PDF dokumentu, tai ir jāizgūst parakstāmā PDF dokumenta saturs binārā formā.

Ir iespējami 2 veidi kā izgūt PDF dokumenta saturu:

- 1) izveidot web lapu, kas apstrādās HTTP GET pieprasījumu un atgriezīs parakstāmā PDF dokumenta bināros datus;
- 2) ļaut sīklītotnei izgūt datus no lokālas HDD mapes, kuras pilns ceļš ir tiek norādīts `documentsXml` parametrā `url`;

Parakstītā PDF dokumenta satura saglabāšana:

Lai sīklītotne spētu saglabāt parakstīto PDF dokumentu, tai ir jāpārņem parakstītais PDF dokuments uz serveri.

Ir iespējami 2 veidi kā saglabāt parakstītā PDF dokumentu saturu:

- 1) izveidot web lapu, kas apstrādās HTTP POST pieprasījumu un saglabās parakstītā PDF dokumenta bināros datus.
- 2) ļaut sīklītotnei saglabāt datus lokālā HDD mapē, kuras pilns ceļš ir tiek norādīts `targetUrl` parametrā `url`;

Ja tiek izmantota web saites pieeja, PDF dokumenta saglabāšanas web pieprasījumā sīklītotne pados šādus parametrus:

Parametra nosaukums	Apraksts
<code>documentid</code>	PDF dokumenta identifikators kā 32 bitu <i>integer</i> skaitlis, kas tika padots sīklītotnes inicializācijas parametra <code>documentsXml XML</code> 'ā. Piemērs: 1.
<code>documentlog</code>	Audita jeb <i>log</i> ieraksti teksta formā, kas apraksta PDF dokumenta parakstīšanas procesu.
<code>document</code>	Parakstītā PDF dokumenta binārais saturs vai arī vērtība <code>null</code> , ja parakstīšana nav noritējusi veiksmīgi.

Ja parakstītais PDF dokuments tiek saglabāts uz web servera ar speciāli šim mērķim paredzētu web lapu, tad parakstītā PDF dokumenta datnes nosaukums var tik formēts atbilstoši biznesa procesu prasībām.

Ja parakstītais PDF dokuments tiek saglabāts lokālajā HDD mapē, tad parakstītā PDF dokumenta datnes nosaukums tiek formēts kā *PDF dokumenta ID no documentsXml iezīmes vērtības + parakstāmā PDF dokumenta datnes paplašinājums*.

Sīklītotnes ievietošana web lapā:

Lai web aplikācijā būtu iespējams darbināt parakstīšanas sīklītotni, nepieciešams veikt sekojošas darbības:

- 1) web lietojumprogrammas ietvaros nepieciešams izveidot jaunu mapi *applet*, kurā nepieciešams iekopēt parakstīšanas sīklītotnes *.jar* datnes;
- 2) ja datu izgūšanu paredzēts veikt ar web saites palīdzību, web lietojumprogrammas ietvaros nepieciešams izveidot jaunu web lapu, kas:

- a. dinamiski ģenerēs **HTML**, kurā tiks iekļauts sīklietotnes inicializācijas **HTML** kods, inicializācijas parametri un inicializācijas parametru vērtības, tajā skaitā arī parametra **documentsXml XML** vērtību, kas apraksta ceļu/ceļus un parakstāmo **PDF** dokumentu/dokumentiem;
 - b. un kurā būs iekļauts notikumu apstrādes **JavaScript** funkcija;
- 3) ja datu saglabāšanu paredzēts veikt ar **web** saites palīdzību, **web** lietojumprogrammas ietvaros nepieciešams izveidot jaunu **web** lapu, kas nodrošinās parakstāmā **PDF** dokumenta bināro datu saglabāšanu;
 - 4) **web** lietojumprogrammas ietvaros nepieciešams izveidot jaunu **web** lapu, kas nodrošinās parakstītā **PDF** dokumenta bināro datu saglabāšanu pēc norādītā **PDF** dokumenta **id**.

Pēc parakstīšana **web** lapas ielādes un sīklietotnes inicializācijas, automātiski tiek uzsākts **PDF** dokumentu parakstīšanas process.

Sīklietotnes inicializācijas/komandrindas parametri:

Parametra nosaukums	Apraksts
debug	Parametrs, kas norāda vai pēc sīklietotnes darba beigām aizvērt JAVA konsoli. Iespējamās parametra vērtības ir yes , no . Noklusētā vērtība ir no .
authUsername	Parametrs lietotājvārdam Apache basic autentifikācijai, kas nodrošina piekļuvi parakstāmajām PDF datnēm. Noklusētā vērtība ir null .
authPassword	Parametrs parolei Apache basic autentifikācijai, kas nodrošina piekļuvi parakstāmajām PDF datnēm. Noklusētā vērtība ir null .
authHost	Parametrs hostam Apache basic autentifikācijai, kas nodrošina piekļuvi parakstāmajām PDF datnēm. Noklusētā vērtība ir null .
authRealm	Parametrs grupai basic autentifikācijai, kas nodrošina piekļuvi parakstāmajām PDF datnēm. Noklusētā vērtība ir null .
documentsXml	<p>Parakstāmo dokumentu aprakstošs XML, kas satur vienu vai vairākas PDF dokumentu aprakstošas sadaļas ar iezīmēm id jeb dokumenta identifikators, url jeb dokumenta izgūšanas HDD mapes adresi vai web saites adresi.</p> <p>Parametrs var saturēt:</p> <ol style="list-style-type: none"> 1) pilnu web saites adresi, kas veic parakstāmā PDF dokumenta satura izgūšanu; 2) pilnu mapes ceļu no kura tiks izgūts parakstītais PDF dokuments; <p>Parakstīšanas sīklietotne analizē norādītās vērtības un veic atbilstošu PDF datņu izgūšanu.</p> <p>Piemērs nr. 1 (web saites adrese):</p> <pre><documents> <document> <id>1</id> <url>http://www.dsistemas.lv/download_unsigned_pdf_documents.php?/pdf-document-1.pdf</url> </document> <document> <id>2</id> <url>http://www.dsistemas.lv/download_unsigned_pdf_documents.php?/pdf-document-2.pdf</url> </document> </documents></pre> <p>Piemērs nr. 2 (mapes adrese):</p> <pre><documents> <document> <id>1</id> <url>c:\unsigned pdf documents\pdf-document-1.pdf</url></pre>

	<pre> </document> <document> <id>2</id> <url>c:\unsigned_pdf_documents\pdf-document-2.pdf</url> </document> </documents> </pre>
<i>directsmartcardaccess</i>	<p>Parametrs norāda vai paraksta sertifikāts tiek ņemts no viedkartes vai arī tiek atvērts sertifikātu izvēlnes logs. Iespējamās parametra vērtības ir <i>yes</i>, <i>no</i> (noklusētā vērtība ir <i>no</i>).</p> <p>Sertifikātu izvēlnes logs dotajā versijā strādā tikai un vienīgi <i>Win</i> platformas ietvaros (<i>Win XP, Win 7, Win 8</i>)!</p>
<i>targetUrl</i>	<p>Mapes ceļš vai <i>web</i> saite, kas tiek izmantota, lai saglabātu parakstītos <i>PDF</i> dokumentus.</p> <p>Parametrs var saturēt:</p> <ol style="list-style-type: none"> 1) pilnu <i>web</i> saites adresi, kas tiks izmantota, lai augšupielādētu un saglabātu parakstīto <i>PDF</i> dokumentu; 2) pilnu mapes ceļu, kurā tiks saglabāts parakstītais <i>PDF</i> dokuments; <p>Parakstīšanas sīklīetojne analizē norādītās vērtības un veic atbilstošu <i>PDF</i> datņu saglabāšanu.</p> <p>Piemērs nr. 1 (<i>web</i> saites adrese): http://www.dsistemas.lv/save_signed_pdf_document.php</p> <p>Piemērs nr. 2 (mapes adrese): c:\signed_pdf_documents</p>
<i>cache_archive</i>	<p>Parametrs, kas apraksta kādas bibliotēkas nepieciešams ielādēt no lokālās <i>JAVA</i> glabātuves (<i>cache</i>). Tas nodrošina, ka pie atkārtotas parakstīšanas <i>web</i> lapas ielādes sīklīetojnes netiek lādētas no <i>web</i> servera.</p> <p>Tekošajā produkta versijā jāiekļauj šāda vērtība: applet.jar,lib/bcmail-jdk15-143.jar,lib/bcprov-jdk15-143.jar,lib/bctsp-jdk15-143.jar,lib/edoc-1.4.4.jar,lib/edoc-config.jar,lib/hc.jar,lib/iaikPkcs11Wrapper.jar,lib/iText-5.0.2.jar,lib/log4j-1.2.14.jar,lib/resources.jar,lib/webservices-rt.jar</p>
<i>java_arguments</i>	<p>Parametrs, kas tiek izmantots specifisku <i>JAVA</i> parametrus inicializēšanai. Noklusētā vērtība ir <i>null</i>. Piemērs: <i>Xmx1024m</i>, kas palielina izmantojamās atmiņas daudzumu līdz vērtībai <i>1024Mb</i>.</p>
<i>timestamp</i>	<p>Parametrs, kas norāda, ka <i>PDF</i> dokumentam nepieciešams pievienot laika zīmogu. Iespējamās parametra vērtības ir <i>yes</i>, <i>no</i>. Noklusētā vērtība ir <i>null</i>.</p>
<i>timestampserver</i>	<p>Parametrs, kas norāda laika zīmoga servera <i>http/https web</i> adresi. Noklusētā vērtība ir <i>null</i>.</p>
<i>emestamp</i>	<p>Parametrs, kas norāda, ka laika zīmoga serverī nepieciešams autorizēties ar viedkartes autorizācijas sertifikātu. Iespējamās parametra vērtības ir <i>yes</i>, <i>no</i>. Noklusētā vērtība ir <i>no</i>.</p>
<i>timestampusername</i>	<p>Parametrs, kas norāda ar kādu lietotāja vārdu tiks veikta autorizācija laika zīmoga serverī. Noklusētā vērtība ir <i>null</i>.</p>
<i>timestamppassword</i>	<p>Parametrs, kas norāda ar kādu paroli tiks veikta autorizācija laika zīmoga serverī. Noklusētā vērtība ir <i>null</i>.</p>
<i>hidesignature</i>	<p>Parametrs, kas norāda vai rādīt/nerādīt <i>PDF</i> dokumentā paraksta redzamo daļu. Iespējamās parametra vērtības ir <i>yes</i>, <i>no</i>. Noklusētā vērtība ir <i>no</i>.</p>
<i>reason</i>	<p>Parakstīšanas iemesls. Noklusētā vērtība ir <i>null</i>. Piemērs: <i>Publikācijas parakstīšana</i>.</p>
<i>location</i>	<p>Parametrs ar kura palīdzību tiek norādīta parakstīšanas vieta. Noklusētā vērtība ir <i>null</i>. Piemērs: <i>Rīga</i>.</p>
<i>contact</i>	<p>Parametrs ar kura palīdzību tiek norādīta parakstītāja kontaktinformācija. Noklusētā vērtība ir <i>null</i>. Piemērs: <i>info@dsistemas.lv</i>.</p>
<i>crls</i>	<p>Parametrs, kas norāda, ka nepieciešams iekļaut pievienotajā parakstā <i>CRL</i></p>

	<p>datnes. Iespējamās parametra vērtības ir <i>yes</i>, <i>no</i>. Noklusētā vērtība ir <i>no</i>.</p> <p>Piezīmes: Ja parametra vērtība ir <i>no</i> vai vērtība nav norādīta, <i>PDF</i> dokuments tiek verificēts tiešsaistē. <i>no</i> ir ieteicamā vērtība, jo tā ļauj samazināt parakstītā <i>PDF</i> dokumenta izmēru.</p>
<i>certifyforformfilling</i>	<p>Parametrs ar kura palīdzību tiek norādīts, ka <i>PDF</i> dokumentu nepieciešams sertificēt uz izmaiņām. Iespējamās parametra vērtības ir <i>yes</i>, <i>no</i>. Noklusētā vērtība ir <i>no</i>.</p>
<i>signatureimage</i>	<p>Parametrs ar kura palīdzību tiek norādīts ceļš uz grafisko attēlu <i>*.png</i> vai <i>*.jpg</i> formātā, kas identificē pievienojamo parakstu. Grafiskais attēls tiks attēlots <i>PDF</i> dokumentā. Piemērs: <i>c:\paraksta_attels.jpg</i>.</p> <p>Piezīmes: dotais parametrs darbojas tikai komandrindas izsaukuma gadījumā.</p>
<i>signaturefieldpositionkeyword</i>	<p>Teksta frāze, kuras koordinātas izmantot paraksta lauka novietošanai. Piemēram, ja šajā parametrā tiek norādīts teksts <i>'paraksts šeit:'</i>, parakstīšanas procesā <i>PDF</i> dokumentā tiks meklēta šī atslēgas frāzē. Ja frāze tiks atrasta, tad paraksta redzamā daļa tiks novietota ar šīs frāzes koordinātēm.</p>
<i>signaturefieldwidth</i>	<p>Parametrs ar kura palīdzību tiek norādīts paraksta lauka redzamās daļas platums milimetros. Doto parametru nepieciešams lietot kopā ar <i>signaturefieldpositionkeyword</i>. Piemērs: <i>100</i>. Noklusētā vērtība ir <i>null</i>.</p>
<i>signaturefieldheight</i>	<p>Parametrs ar kura palīdzību tiek norādīts paraksta lauka redzamās daļas augstums milimetros. Doto parametru nepieciešams lietot kopā ar <i>signaturefieldpositionkeyword</i>. Piemērs: <i>40</i>. Noklusētā vērtība ir <i>null</i>.</p>
<i>signaturefieldoffsetx</i>	<p>Parametrs ar kura palīdzību tiek norādīts paraksta redzamās daļas nobīde milimetros pa x asi. Doto parametru nepieciešams lietot kopā ar <i>signaturefieldpositionkeyword</i>. Piemērs: <i>120</i>. Noklusētā vērtība ir <i>null</i>.</p>
<i>signaturefieldoffsety</i>	<p>Parametrs ar kura palīdzību tiek norādīts paraksta redzamās daļas nobīde milimetros pa y asi. Doto parametru nepieciešams lietot kopā ar <i>signaturefieldpositionkeyword</i>. Piemērs: <i>120</i>. Noklusētā vērtība ir <i>null</i>.</p>

Notikumu (events) apstrāde:

Parakstīšanas procesa gaitā, sīklītotne ar *JavaScript* notikumu funkcijas *progress* palīdzību informē pamat lapu par tekošo parakstīšanas statusu.

Notikumu *JavaScript* funkcija satur šādus parametrus:

Parametra nosaukums	Apraksts
<i>documentid</i>	<p><i>PDF</i> dokumenta identifikators, kas tika padots sīklītotnes inicializācijas parametra <i>documentsXml XML</i>'ā. Piemērs: <i>1</i>. Ja vērtība ir <i>-1</i>, tad tas nozīmē, ka tiek aprakstīts vispārējs stāvoklis, kas attiecās uz kopējo parakstīšanas procesu nevis uz kādu konkrētu parakstāmo <i>PDF</i> dokumentu.</p>
<i>state</i>	<p>Stāvoklis, kurā pašlaik atrodas parakstīšanas process.</p> <p>Vispārējie parakstīšanas procesa stāvokļi:</p> <ol style="list-style-type: none"> <i>RESULT_DESCRIPTOR_XML_NOT_SPECIFIED</i> - inicializējot sīklītotni, netika norādīts dokumentu aprakstošais <i>XML</i>; <i>RESULT_RESULT_UPLOAD_URL_NOT_SPECIFIED</i> - inicializējot sīklītotni, netika norādīta saite, kur vērsties, augšupielādējot parakstīto dokumentu un audīta ierakstus; <i>RESULT_UNABLE_TO_PARSE_XML</i> – <i>PDF</i> dokumentu aprakstošajam <i>XML</i> ir bojāta struktūra; <i>RESULT_UNABLE_TO_READ_XML</i> – <i>PDF</i> dokumenta aprakstošajam <i>XML</i> nepareiza uzbūve; <i>MESSAGE_ALL_DONE</i> – <i>PDF</i> dokumentu apstrāde pabeigta; <p><i>PDF</i> dokumenta parakstīšanas procesa stāvokļi:</p> <ol style="list-style-type: none"> <i>STEP_DOWNLOADING_DOCUMENT</i> – notiek <i>PDF</i> dokumenta lejupielāde uz klienta darbstaciju;

	<ol style="list-style-type: none"> 2) <code>STEP_SIGNING_DOCUMENT</code> – notiek <i>PDF</i> dokumenta parakstīšana; 3) <code>STEP_UPLOADING_SIGNED_DOCUMENT</code> – notiek parakstītā <i>PDF</i> dokumenta augšupielāde uz serveri; 4) <code>STEP_FAILURE_DOWNLOADING_DOCUMENT</code> – notikusi kļūda, lejupielādējot <i>PDF</i> dokumentu uz klienta darbstaciju; 5) <code>STEP_FAILURE_SIGNING_DOCUMENT</code> – notikusi kļūda, parakstot <i>PDF</i> dokumentu; 6) <code>STEP_FAILURE_UPLOADING_SIGNED_DOCUMENT</code> – notikusi kļūda, augšupielādējot parakstīto <i>PDF</i> dokumentu; 7) <code>RESULT_COMPLETED</code> – parakstīšanas process noritējis veiksmīgi; 8) <code>RESULT_NOT_COMPLETED</code> – parakstīšanas process noritējis neveiksmīgi; 9) <code>RESULT_PDF_NOT_SPECIFIED</code> – nav norādīts ceļš uz parakstāmo <i>PDF</i> dokumentu; 10) <code>RESULT_PIN_BLOCKED</code> – <i>PIN</i> koda ievade ir bloķēta (vairāk kā 3 reizes <i>PIN</i> kods ievadīts nepareizi); 11) <code>RESULT_PIN_INPUT_ABORTED</code> – <i>PIN</i> koda ievades forma aizvērta ar <i>Atcelt</i> pogas palīdzību; 12) <code>RESULT_CARD_NOT_INSERTED</code> – parakstīšanas viedkarte nav ievietota karšu lasītājā; 13) <code>RESULT_CARD_IS_INSERTED</code> – parakstīšanas viedkarte tika ievietota karšu lasītājā; 14) <code>RESULT_INTERNAL_ERROR</code> – parakstīšanas iekšējā kļūda; 15) <code>RESULT_INVALID_HTTP_RESPONSE</code> – saņemtā <i>HTTP</i> atbilde uz lejupielādes/saglabāšanas <i>web</i> pieprasījumu ir nekorekta; 16) <code>RESULT_NO_HTTP_RESPONSE</code> – nav saņemta <i>HTTP</i> atbilde uz lejupielādes/saglabāšanas <i>web</i> pieprasījumu; 17) <code>RESULT_OCSP_INVALID_RESPONSE</code> – <i>OCSP</i> atbilde satur kļūdu; 18) <code>RESULT_INVALID_FILE_PATH</code> – norādītais ceļš uz <i>PDF</i> dokumentu ir nekorekts vai arī <i>PDF</i> dokuments neeksistē; 19) <code>RESULT_INVALID_TARGET_URL</code> – <i>targetUrl</i> ir norādīts nekorekts ceļš; 20) <code>RESULT_TARGET_DOESNT_EXIST</code> – <i>targetUrl</i> ir norādīts neeksistējošs ceļš; <p><code>RESULT_PDF_NOT_VALID</code> – norādītais dokuments nav <i>PDF</i> dokuments vai arī <i>PDF</i> dokuments ir bojāts.</p>
--	---

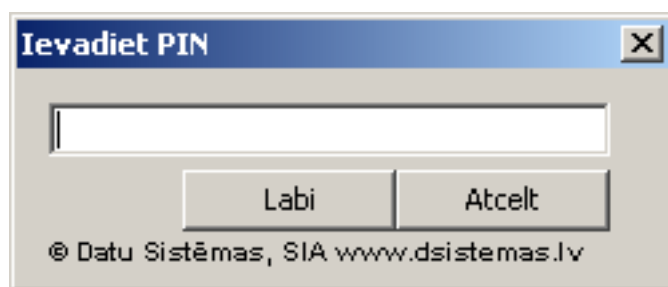
Notikumu *JavaScript* funkcijas bāzes piemērs:

```
function progress(id, state) {
  switch (state) {
    case 'STEP_DOWNLOADING_DOCUMENT':
      alert('Downloading...');
      break;
    case 'STEP_FAILURE_DOWNLOADING_DOCUMENT':
      alert('Error! Could not download PDF document!');
      break;
    case 'STEP_SIGNING_DOCUMENT':
      alert('Signing...');
      break;
    case 'STEP_FAILURE_UPLOADING_SIGNED_DOCUMENT':
      alert('Error! Could not upload signed PDF document!');
      break;
    case 'RESULT_COMPLETED':
      alert('All done!');
      break;
  }
}
```

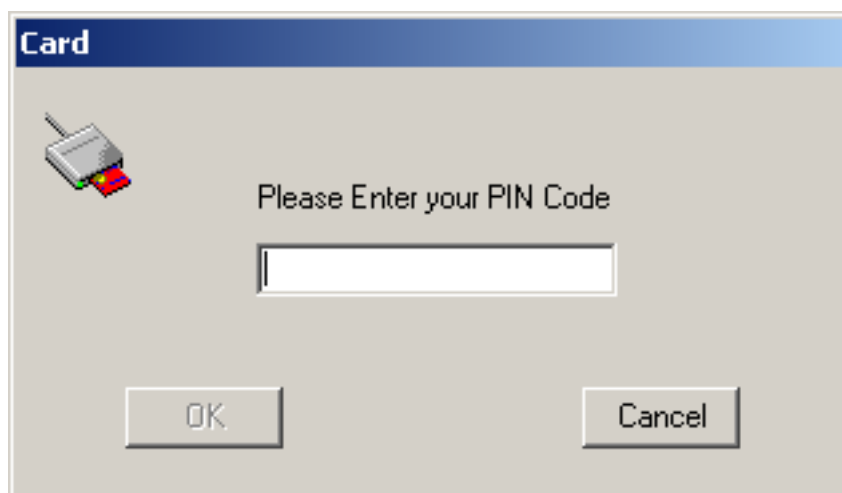
***PIN* koda ievade:**

Veicot *PDF* dokumentu parakstīšanu, lietotājam, tiks pieprasīts ievadīt *PIN* kodu ar ko tiek aizsargāts paraksta sertifikāts. Ja tiek parakstīti vairāk kā viens *PDF* dokuments, *PIN* kods paraksta sesijas ietvaros ir jāievada tikai vienu reizi.

Ja paraksta sertifikāts tiek izvēlēts no viedkartes (sīklietotnes inicializācijas parametrs *directsmartcardaccess* satur vērtību *yes*), tiek izvadīts šāds *PIN* koda ievades logs:



Ja paraksta sertifikāts tiek izvēlēts no sertifikātu izvēlnes loga (sīklīdnes inicializācijas parametrs *directsmartcardaccess* satur vērtību *no*), tiek izvadīts šāds sistēmā iebūvēts *PIN* koda ievades logs:



3. PDF parakstīšanas sīklietotnes izsaukšanas piemērs no PHP web lapas

Veicamie šoļi, lai integrētu PDF parakstīšanas sīklietotni PHP web aplikācijā:

1. solis – uz web servera tiek izveidota mape `unsigned_pdf_documents`, kas saturēs PDF dokumentus, kurus nepieciešams parakstīt; veikt Apache web servera konfigurāciju, lai web aplikācijai būtu piekļuve šai mapei;
2. solis – uz web servera tiek izveidota mape `signed_pdf_documents`, kurā tiks saglabāti parakstītie PDF dokumenti; veikt Apache web servera konfigurāciju, lai web aplikācijai būtu piekļuve šai mapei;
3. solis – uz web servera web lietojumprogrammas ietvaros tiek izveidota mape `applet`, kurā tiek iekopētas sīklietotnes `*.jar` datnes;
4. solis – uz web servera tiek izveidota parakstīšanas PHP web lapa, kurā tiek iekļauts sīklietotnes HTML inicializācijas kods un notikumu JavaScript funkcija; paraksta web lapā jāimplementē funkcionalitāte, kas aizpilda inicializācijas parametrus ar korektām vērtībām;
5. solis – uz web servera tiek izveidota PHP web lapa `download_unsigned_pdf.php`, kas veiks parakstīto PDF dokumentu izgūšanu no `unsigned_pdf_documents` mapes.
6. solis – uz web servera tiek izveidota PHP web lapa `save_signed_pdf.php`, kas veiks parakstīto PDF dokumentu saglabāšanu `signed_pdf_documents` mapē.

PHP `save_signed_pdf.php` web lapas paraugs:

```
<?php
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $uploadsDir = 'signed_pdf_documents/';
    $documentId = $_POST['documentid'];
    if ($documentId == null)
        return;

    if (isset($_POST['documentlog'])) {
        $file = null;
        if ($file = fopen($uploadsDir.$documentId.'.log', 'a+')) {
            fwrite($file, $_POST['documentlog']);
            fclose($file);
        }
    }
    if (isset($_FILES["document"]) && trim($_FILES["document"]['name']) != '') {
        $target_path = $uploadsDir . $documentId.'_signed.pdf';

        if(move_uploaded_file($_FILES['document']['tmp_name'], $target_path)) {
            echo 'ok';
        }
    }
}
?>
```

PHP paraugs ar iekļautu parakstīšanas sīklietotnes HTML koda paraugu:

```
<applet name="applet" id="applet" code="org.datasystems.AppletMain" height="0" width="0">
<param name="documentsXml" value="<?=$xml?>"/>
<param name="targetUrl" value="http://web_server_address/save_signed_pdf.php"/>
<param name="scriptable" value="true" />
<param name="codebase" value="applet" />
<param name="cache_archive" value=" applet.jar,lib/bcmail-jdk15-143.jar,lib/bcprov-jdk15-143.jar,lib/bctsp-jdk15-143.jar,lib/edoc-1.4.4.jar,lib/edoc-config.jar,lib/hc.jar,lib/iaikPkcs11Wrapper.jar,lib/iText-2.1.7.jar,lib/log4j-1.2.14.jar,lib/resources.jar,lib/webservices-rt.jar" />
<param name="java_arguments" value="-Xmx1024m"/>
<?=( $hideSignatureFlag ? '<param name="hidesignature" value="yes"/>' : '' )?>
<?=(isset($_POST['reason']) ? '<param name="reason" value="'.$_POST['reason'].'"/>' : '' )?>
<?=(isset($_POST['location']) ? '<param name="location" value="'.$_POST['location'].'"/>' : '' )?>
<?=(isset($_POST['contact']) ? '<param name="contact" value="'.$_POST['contact'].'"/>' : '' )?>
</applet>
```

Notikumu JavaScript funkcijas paraugs:

```
function progress(id, state) {
```

```
switch (state) {
  case 'STEP_DOWNLOADING_DOCUMENT':
    alert('Downloading...');
    break;
  case 'STEP_FAILURE_DOWNLOADING_DOCUMENT':
    alert('Error! Could not download PDF document!');
    break;
  case 'STEP_SIGNING_DOCUMENT':
    alert('Signing...');
    break;case
  case 'STEP_FAILURE_UPLOADING_SIGNED_DOCUMENT':
    alert('Error! Could not upload signed PDF document!');
    break;
  case 'RESULT_COMPLETED':
    alert('All done!');
    break;
}
}
```

4. **PDF** parakstīšanas sīklietotnes izsaukšana no komandrindas

PDF parakstīšanas sīklietotnes izsaukšana no komandrindas:

PDF parakstīšanas sīklietotne izsaukuma no komandrindas vispārīgā forma ir:

```
java -jar dspdfsigner.jar documentsXmlFilePath <ceļš uz parakstāmo PDF dokumentu  
aprakstošo XML datni> targetUrl <web saite vai mapes ceļš, kur tiks novietots  
parakstītais PDF dokuments> <parametrs 1> <parametra 1 vērtība> <parametrs 2>  
<parametra 2 vērtība> ...
```

, kur parametri:

- 1) `documentsXmlFilePath` – ceļš uz **XML** datni, kas apraksta parakstāmo **PDF** dokumentu *id* un atrašanās ceļus; doto parametru nepieciešams obligāti norādīt;
- 2) `targetUrl` – ceļš uz mapi, kurā tiks saglabāti parakstītie **PDF** dokumenti, vai *web* saite, kas veiks parakstīto **PDF** dokumentu saglabāšanu; doto parametru nepieciešams obligāti norādīt;

Parametru vērtības, kuras ir iespējams norādīt komandrindas izsaukumā opcionāli, sakrīt ar *web* sīklietotnes inicializācijas parametru nosaukumiem (skatīt *web* sīklietotnes aprakstu).

Ja parakstītais **PDF** dokuments tiek saglabāts lokālajā **HDD** mapē, tad parakstītā **PDF** dokumenta datnes nosaukums tiek formēts kā **PDF** dokumenta *ID* no `documentsXml` iezīmes vērtības + *parakstāmā PDF dokumenta datnes paplašinājums*.

Ja parakstītais **PDF** dokuments tiek saglabāts ar *web* saites palīdzību, dokumenta apstrāde notiek identiski *web* sīklietotnes gadījumam, kad dokumenti tiek saglabāti ar *web* saites palīdzību (skatīt *web* sīklietotnes aprakstu).

Audita jeb log datne:

Ja parakstītais **PDF** dokuments tiek saglabāts lokālajā **HDD** mapē, papildus tiek izveidota arī audita datne, kuras nosaukums tiek formēts kā **PDF** dokumenta *ID* no `documentsXml` iezīmes vērtības + `.log` kā datnes paplašinājums.

5. *PDF* parakstīšanas sīklīetotnes izsaukšanas piemēri no *komandrindas*

PDF parakstīšanas izsaukuma piemērs nr. 1 - parakstītie *PDF* dokumenti tiek saglabāti mapē *c:\signed_pdf_documents*:

```
java -jar dspdfsigner.jar documentsXmlFilePath c:\pdf-documents-to-be-signed-xml-metadata.xml targetUrl c:\signed_pdf_documents
```

PDF parakstīšanas izsaukuma piemērs nr. 2 - parakstītie *PDF* dokumenti tiek saglabāti mapē *c:\signed_pdf_documents*, paraksta sertifikāts tiek nolasīts no viedkartes:

```
java -jar dspdfsigner.jar documentsXmlFilePath c:\pdf-documents-to-be-signed-xml-metadata.xml targetUrl c:\signed_pdf_documents directsmartcardaccess yes
```

PDF parakstīšanas izsaukuma piemērs nr. 3 - parakstītie *PDF* dokumenti tiek saglabāti ar *web* saites palīdzību:

```
java -jar dspdfsigner.jar documentsXmlFilePath c:\pdf-documents-to-be-signed-xml-metadata.xml targetUrl http://web_server_address/save_signed_pdf_document.php
```

, kur *pdf-documents-to-be-signed-xml-metadata.xml* datne satur šādu *XML*:

```
<documents>
  <document>
    <id>1</id>
    <url>c:\unsigned_pdf_documents\pdf-document-1.pdf</url>
  </document>
  <document>
    <id>2</id>
    <url>c:\unsigned_pdf_documents\pdf-document-2.pdf</url>
  </document>
</documents>
```