Valsts reģionālās attīstības aģentūra



|  |  |
| --- | --- |
|  |  |

Valsts informācijas sistēmu savietotāja, Latvijas valsts portāla www.latvija.lv un elektronisko pakalpojumu izstrāde un uzturēšana

3.daļa "VISS un Portāla jaunu un esošo moduļu papildinājumu izstrāde, ieviešana, garantijas apkalpošana un uzturēšana saskaņā ar tehnisko specifikāciju"

IDDV formu izveidošanas ceļvedis VISS portālam

Programmētāja rokasgrāmata

VRAA----

10.01.2014. versija

Rīgā 2013

|  |  |  |
| --- | --- | --- |
|  | **IEGULDĪJUMS TAVĀ NĀKOTNĒ** |  |

Dokumenta identifikācija

|  |  |
| --- | --- |
| Dokumenta ID: | VRAA-----V- |
| Dokumenta nosaukums: | Valsts informācijas sistēmu savietotāja, Latvijas valsts portāla www.latvija.lv un elektronisko pakalpojumu izstrāde un uzturēšana.  3.daļa "VISS un Portāla jaunu un esošo moduļu papildinājumu izstrāde, ieviešana, garantijas apkalpošana un uzturēšana saskaņā ar tehnisko specifikāciju".  IDDV formu izveidošanas ceļvedis VISS portālam.  Programmētāja rokasgrāmata. |
| Dokumenta kods: | VRAA---- |
| Versija: | Versija , Laidiens (saīsināti V ) |

Saskaņojumi

|  |  |  |  |
| --- | --- | --- | --- |
| Organizācija | Vārds, uzvārds, amats | Datums | Paraksts |
| Valsts reģionālās attīstības aģentūra | J.Zandbergs, projekta vadītājs no Pasūtītāja puses |  |  |
| SIA "ABC software" | J.Korņijenko, projekta vadītājs par tehniskiem jautājumiem no Izpildītāja puses | 10.01.2014. |  |
| SIA "ABC software" | Z.Murāns, programmētājs | 10.01.2014. |  |
| SIA "ABC software" | M.Pētersons, projekta vadītājs par administratīviem jautājumiem no Izpildītāja puses | 10.01.2014. |  |
| SIA "ABC software" | E.Blumberga, projekta kvalitātes kontroles vadītāja | 10.01.2014. |  |

Izmaiņu vēsture

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Versija | Datums | Apraksts | Organizācija | Autors |
| 1.00 | 17.05.2013. | Izveidota dokumenta sākotnējā versija | SIA „ABC software” | Z.Murāns |
| 1.01 | 05.09.2013 | Veikti labojumi atbilstoši nodevuma izvērtējumam VRAA-6\_15\_11\_58-VISS\_2010-NI-142-PR v.1.00 | SIA „ABC software” | E.Stāmurs |
| 1.02 | 21.10.2013 | Veikti labojumi atbilstoši nodevuma izvērtējumam VRAA-6\_15\_11\_58-VISS\_2010-NI-142-PR v.1.01 | SIA „ABC software” | E.Stāmurs |
| 1.03 | 10.01.2014 | Atjaunota rokasgrāmata pēc izmaiņām lietojumos. | SIA „ABC software” | E.Stāmurs |

**Satura rādītājs**

[1. Ievads 7](#_Toc383097907)

[1.1. Dokumenta nolūks 7](#_Toc383097908)

[1.2. Darbības sfēra 7](#_Toc383097909)

[1.3. Termini un pieņemtie apzīmējumi 7](#_Toc383097910)

[1.4. Saistība ar citiem dokumentiem 7](#_Toc383097911)

[1.5. Dokumenta pārskats 8](#_Toc383097912)

[2. Risinājuma arhitektūra 9](#_Toc383097913)

[2.1. IDDV emulators 10](#_Toc383097914)

[3. Izstrādes vides sagatavošana 11](#_Toc383097915)

[3.1. Izstrādes vides prasības 11](#_Toc383097916)

[3.2. Pakotnes saturs 11](#_Toc383097917)

[3.3. Portāla emulatora uzstādīšana 11](#_Toc383097918)

[3.3.1. IDDV servisa maršrutēšanas pakalpes uzstādīšana 11](#_Toc383097919)

[3.3.1.1. Vispārējie norādījumi un vides prasības 11](#_Toc383097920)

[3.3.1.2. Sertifikātu uzstādīšana 12](#_Toc383097921)

[3.3.1.3. Pakalpes uzstādīšana 12](#_Toc383097922)

[3.3.2. Sertifikātu pārvaldības rīka atvēršana 17](#_Toc383097923)

[3.3.3. Saknes sertifikātu imports 19](#_Toc383097924)

[3.3.4. IDDV formu IIS lietojuma (application) izveide 21](#_Toc383097925)

[3.3.5. Portāla emulatora apakš-aplikācijas izveidošana 26](#_Toc383097926)

[3.3.6. IDDV formas apakš-lietojuma izveide IIS 28](#_Toc383097927)

[3.3.7. Tiesību piešķiršana apakš-lietojumu mapēm 28](#_Toc383097928)

[3.3.8. IDDV emulatora un formu sertifikāta reģistrēšana 29](#_Toc383097929)

[3.3.9. IDDV emulatora un formu īpašas piekļuves apgabala (realm) reģistrēšana 29](#_Toc383097930)

[3.4. CDN uzstādīšana 29](#_Toc383097931)

[4. IDDV formas realizēšana 31](#_Toc383097932)

[4.1. Klase Task<T> 31](#_Toc383097933)

[4.2. Klase ITaskPage<T, K> 31](#_Toc383097934)

[4.2.1. Administratīvie lauki 33](#_Toc383097935)

[4.2.2. Darba uzdevuma pielikumi 33](#_Toc383097936)

[4.2.3. Ievades pareizības pārbaude 34](#_Toc383097937)

[4.3. Pielikumu validācijas scenārijs 34](#_Toc383097938)

[4.4. Pielikuma izveides scenārijs 36](#_Toc383097939)

[4.5. Sesijas izmantošana 36](#_Toc383097940)

[4.6. Darba uzdevumu galamērķi 36](#_Toc383097941)

[4.7. Portāla emulatora konfigurēšana 37](#_Toc383097942)

[4.7.1. Portāla konfigurēšana servisu režīmā 39](#_Toc383097943)

[4.7.2. Portāla konfigurēšana failu sistēmas emulācijas režīmā 41](#_Toc383097944)

[4.8. Iddv formas konfigurēšana 42](#_Toc383097945)

[4.8.1. Formas konfigurācijas iegūšana 44](#_Toc383097946)

[4.8.2. Autentifikācijas konfigurēšana 45](#_Toc383097947)

[4.9. Vārdtelpas 47](#_Toc383097948)

**Attēlu saraksts**

[1.attēls. E-pakalpojuma ielāde 9](#_Toc383097867)

[2.attēls. IDDV risinājuma konceptuāla arhitektūra 9](#_Toc383097868)

[3.attēls. Net.Pipe Listener Adapter servisa stāvokļa pārbaude 12](#_Toc383097869)

[4.attēls. Uzstādīšanas palīga pirmais solis 12](#_Toc383097870)

[5.attēls. Licences nosacījumu apstiprināšana 13](#_Toc383097871)

[6.attēls. IIS iestatījumu norādīšana 13](#_Toc383097872)

[7.attēls. Pakalpes vispārējo iestatījumu norādīšana 14](#_Toc383097873)

[8.attēls. Pakalpes e-pakalpojumu konfigurācijas servisa norādīšana 14](#_Toc383097874)

[9.attēls. Pakalpes PFAS un notifikāciju servisu norādīšana 15](#_Toc383097875)

[10.attēls. Pakalpes NASA servisa norādīšana 15](#_Toc383097876)

[11.attēls. Instalācijas apstiprināšana 16](#_Toc383097877)

[12.attēls. Uzstādīšanas process 16](#_Toc383097878)

[13.attēls. Sekmīgas uzstādīšanas pēdējais solis 17](#_Toc383097879)

[14.attēls. MMC palaišana 17](#_Toc383097880)

[15.attēls. Sertifikātu spraudņa pievienošana (1) 18](#_Toc383097881)

[16.attēls. Sertifikātu spraudņa pievienošana (2) 18](#_Toc383097882)

[17.attēls. Sertifikātu spraudņa pievienošana (3) 18](#_Toc383097883)

[18.attēls. Sertifikātu spraudņa pievienošana (4) 19](#_Toc383097884)

[19.attēls. Saknes sertifikāta imports 19](#_Toc383097885)

[20.attēls. Saknes sertifikāta importa vednis (2. solis) 20](#_Toc383097886)

[21.attēls. Saknes sertifikāta importa vednis (3. solis) 20](#_Toc383097887)

[22.attēls. Hosts datnes labošana 21](#_Toc383097888)

[23.attēls. IIS Manager palaišana 21](#_Toc383097889)

[24.attēls. Sertifikātu pārvaldības atvēršana 22](#_Toc383097890)

[25.attēls. Sertifikātu importa uzsākšana 22](#_Toc383097891)

[26.attēls. Sertifikāta imports 23](#_Toc383097892)

[27.attēls. Sertifikāts veiksmīgi importēts 23](#_Toc383097893)

[28.attēls. Jauna lietojuma pievienošanas uzsākšana 24](#_Toc383097894)

[29.attēls. Lietojuma parametri 24](#_Toc383097895)

[30.attēls. Lietojuma pūla atvēršana 25](#_Toc383097896)

[31.attēls. Lietojuma pūla parametri 25](#_Toc383097897)

[32.attēls. Pūla lietotājs 26](#_Toc383097898)

[33.attēls. Apakš-aplikācijas izveides uzsākšana 27](#_Toc383097899)

[34.attēls. Apakš-aplikācijas parametri 27](#_Toc383097900)

[35.attēls. Formas ielāde servisa režīmā. 40](#_Toc383097901)

[36.attēls. Formas ielāde failu sistēmas režīmā bez autentifikācijas. 41](#_Toc383097902)

[37.attēls. Formas konfigurācijas iegūšana 45](#_Toc383097903)

[38.attēls. Autentifikācijas iespējošanas funkcionalitāte 46](#_Toc383097904)

[39.attēls. Direktorija SecureContent 47](#_Toc383097905)

[40.attēls. Metodes Page\_Load piemērs 47](#_Toc383097906)

# Ievads

Iestādes lietotāju (darbinieku) darba vietas lietotne (IDDV) ir VISS (Valsts informācijas sistēmu savietotāja) portāla sastāvdaļa, kas nodrošina darba vietu (*desktop*) valsts un pašvaldību iestāžu darbiniekiem, kuri ir iesaistīti e-pakalpojumu izpildes procesā.

Klienta darba vieta (KDV) ir Latvija.lv (LVP) portāla sastāvdaļa, kas nodrošina darba vietu iedzīvotajiem, kuri izpilda e-pakalpojumus.

## Dokumenta nolūks

Dokumenta nolūks ir iepazīstināt izstrādātājus ar IDDV formu realizācijas principiem. Dokumentā iekļauts arī formu izstrādes vides konfigurēšanas apraksts.

## Darbības sfēra

Šis dokuments paredzēts asinhrono e-pakalpojumu izstrādātājiem, kuriem ir nepieciešams realizēt lietotāju interaktīvo iejaukšanos e-pakalpojuma izpildes procesā.

## Termini un pieņemtie apzīmējumi

Apzīmējumu un terminu vārdnīca pieejama dokumentā [1].

## Saistība ar citiem dokumentiem

Dokuments ir izstrādāts, balstoties uz šādiem dokumentiem:

1. „Valsts informācijas sistēmu savietotāja, Latvijas valsts portāla www.latvija.lv un elektronisko pakalpojumu izstrāde un uzturēšana”. Iepirkuma priekšmeta 3.daļa - VISS un portāla jaunu un esošo moduļu papildinājumu izstrāde, ieviešana, garantijas apkalpošana un uzturēšana saskaņā ar tehnisko specifikāciju. Terminu un saīsinājumu indekss. (VRAA-6\_15\_11\_58-VISS\_2010-TSI).
2. Par Valsts informācijas sistēmu savietotāja, Latvijas valsts portāla www.latvija.lv un elektronisko pakalpojumu izstrāde un uzturēšana. 3.daļa "VISS un Portāla jaunu un esošo moduļu papildinājumu izstrāde, ieviešana, garantijas apkalpošana un uzturēšana saskaņā ar tehnisko specifikāciju". VISS izstrāde. Vadlīnijas. (VRAA-6\_15\_11\_58-VISS\_2010-VISS-VDL).
3. Microsoft .NET Framework, <http://www.microsoft.com/net/Download.aspx>;
4. Microsoft IIS, <http://www.iis.net/>;
5. Valsts informācijas sistēmu savietotāja, Latvijas valsts portāla www.latvija.lv un elektronisko pakalpojumu izstrāde un uzturēšana. 3.daļa "VISS un Portāla jaunu un esošo moduļu papildinājumu izstrāde, ieviešana, garantijas apkalpošana un uzturēšana saskaņā ar tehnisko specifikāciju". Iestādes lietotāju darba vieta. Administratoru rokasgrāmata. (VRAA-6\_15\_11\_58-VISS\_2010-IDDV-AR).

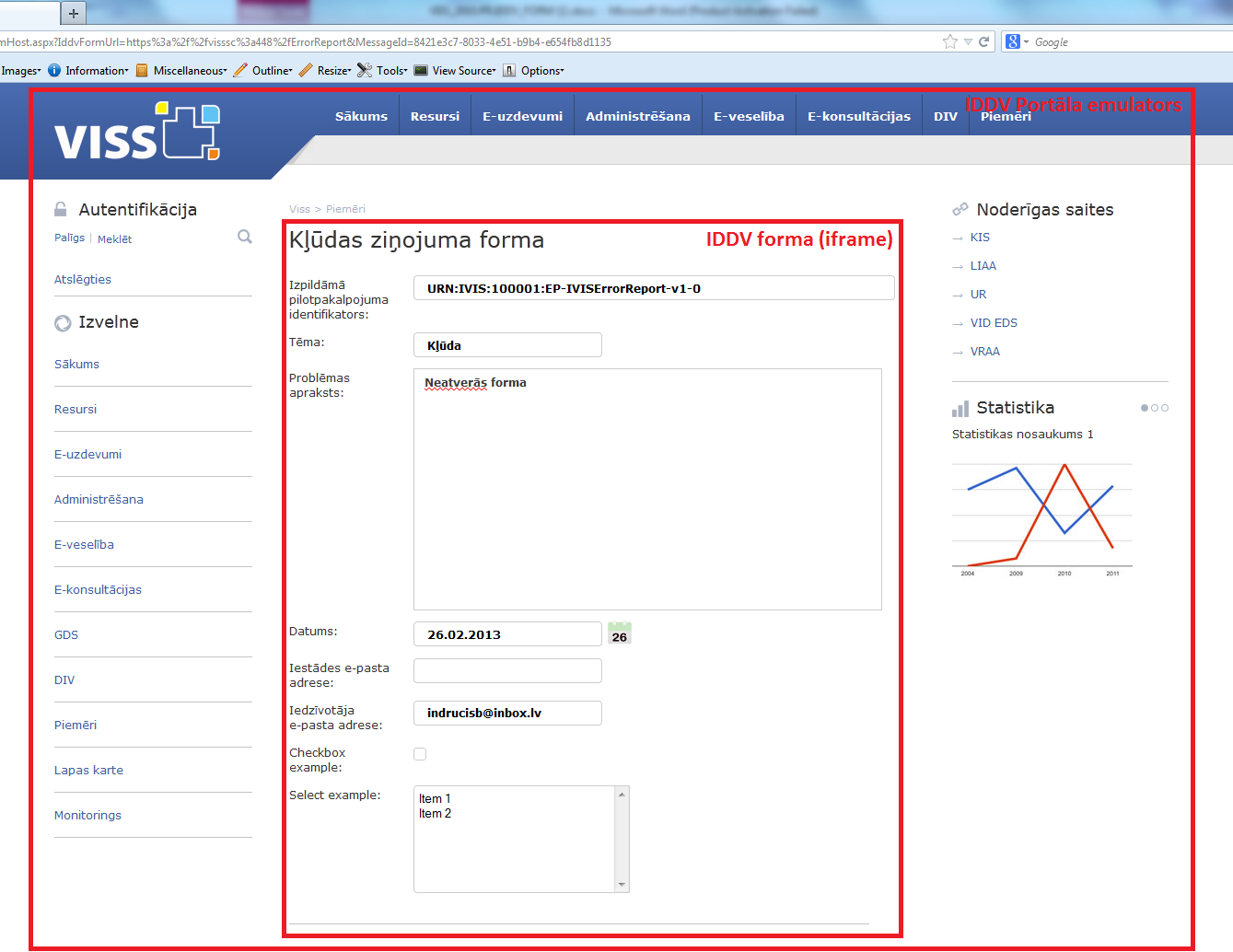
## Dokumenta pārskats

Dokumentu veido šādi nodalījumi:

* „Ievads” – iekļauta informācija par dokumenta vispārējo struktūru, darbības sfēru, nolūku, dokumentā izmantotajiem terminiem un apzīmējumiem, kā arī par saistību ar citiem dokumentiem.
* „Risinājuma arhitektūra” – vispārīgs apraksts par IDDV integrāciju ar VISS. Sākumpunkts izstrādātājiem, lai saprastu kopainu.
* „Izstrādes vides sagatavošana” – instrukcija e-pakalpojuma izstrādes vides sagatavošanai.
* „IDDV formas realizēšana” – IDDV formas realizēšanas apraksts.

# Risinājuma arhitektūra

IDDV formas tiek darbinātas kā atsevišķas ASP.NET tīmekļa aplikācijas. Integrācija ar portālu notiek, ielādējot formu lietotnes *iframe* elementā.



1.attēls. E-pakalpojuma ielāde

IDDV formu izstrādātājiem tiek piedāvāts VISS IDDV lietotnes emulators (skat. 3.2.nodaļu), kas formai nodrošina tādu pat saskarni, kāda tiks izmantota saziņai ar īsto VISS portālu.

IDDV lietotnes konceptuāla arhitektūra ir redzama 2.attēlā.



2.attēls. IDDV risinājuma konceptuāla arhitektūra

IDDV risinājumu veido šādi moduļi (skat. 2.attēlā):

* Datu bāze (DB);
* IDDV integrācijas serviss, kas nodrošina darba uzdevumu nodošanu no VISS Pieprasījumu servisa uz IDDV;
* IDDV datu izgūšanas serviss jeb Ziņojumu serviss;
* IDDV lietotne. Autentifikācijai tiek izmantots SSO. Ja lietotājs ir autentificējies VISS portāla, atverot IDDV formu, notiek automātiska autentifikācija arī IDDV formas lietotnē.
* IDDV lietotnes emulators – IDDV formu izstrādes vide.

Portāla standarta resursi (attēli, JS un CSS datnes) IDDV formai tiek piegādāti, izmantojot atsevišķu tīmekļa aplikāciju (CDN). Izstrādes videi šī aplikācija ir iebūvēta emulatorā.

## IDDV emulators

IDDV formu izstrādes videi tiek izmantots IDDV emulators, kas emulē daļu no portāla infrastruktūras:

* Konteineru, kurā tiek ielādēta IDDV forma;
* Portāla standarta resursus (CSS, JS, attēli).

Pavisam emulators darbojas divos režīmos:

* Kā pilnīgi neatkarīga lietotne, kur autentifikācijas un autorizācijas dati un darba uzdevumi tiek nodrošināti, izmantojot lokālo failu sistēmu;
* Ka pilnvērtīgs IDDV lietotnes emulators, izmantojot reālo autentifikāciju un autorizāciju, kā arī IDDV ziņojumu servisu. Šī režīma komponentes, kuras netiek emulētas un kuru darbībai nepieciešama atbilstoša e-pakalpojumu vides uzturētāja (turpmāk EPU) testa vides konfigurācija:
* Lietotāja autentifikācijas mehānisms (SSO);
* Pieprasījumu serviss (PS).

Savas IDDV formas izstrādātājs var darbināt, izveidojot IIS aplikāciju un piereģistrējot emulatorā (skat. 4.6.nodaļu). Darbināšana *VS Development Server* nav iespējama, jo IDDV formas nepieciešams izsaukt caur TLS (*https*) protokolu, lai nodrošinātu SSO drošību. *VS Development Server* neatbalsta TLS.

# Izstrādes vides sagatavošana

## Izstrādes vides prasības

Lai uzstādītu pakotnē iekļauto programmatūru, datorā pirms tam ir jābūt:

* Vismaz: Microsoft Windows Vista Service Pack 2;
* Instalētam Microsoft .NET Framework 4.0 [3][3];
* Uzstādīts vismaz IIS 7.0 [4];
* Instalētam Microsoft Visual Studio 2010;

## Pakotnes saturs

* **Iddv.Forms.Solution** – Portāla emulatora un IDDV formas iekļaujošais risinājums.
* **visssc.pfx** – IDDV SSL/TLS sertifikāts (https nodrošināšanai).
* **abc-W2008EVAL-CA\_Root\_CA.cer** – IDDV SSL/TLS sertifikāta CA sertifikāts.
* **VISS Root CA.cer** – VISS CA sertifikāts.
* **CDN** – starpaplikāciju web-resursi.
* **IddvFormsTransportEmulator** – failu sistēmas emulatora mapju struktūras paraugs.
* **Abc.Viss.Iddv.RouterService** – IDDV servisu maršrutēšanas pakalpe
* **Abc.Esigner.RouterService** – E-parakstīšanas servisu maršrutēšanas pakalpe. Uzstādīšanas instrukciju skatīt dokumentā [5].

***Iddv.Forms.Solution*** saturs:

* **Iddv.Forms.PortalEmulator** – IDDV formu portāla emulators (izstrādes videi).
* **Iddv.Forms.Example.ErrorReport.Web** – IDDV forma (piemērs).

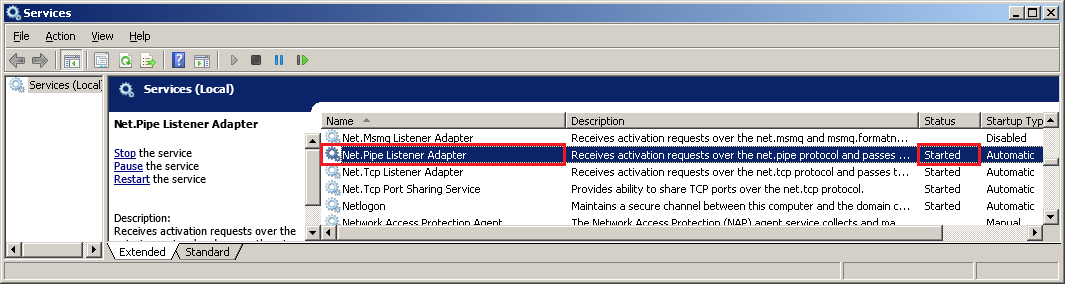
## Portāla emulatora uzstādīšana

### IDDV servisa maršrutēšanas pakalpes uzstādīšana

#### Vispārējie norādījumi un vides prasības

Maršrutēšanas pakalpe (serviss) jāizvieto uz tā paša servera, kurā tiks uzstādīts pats IDDV lietojums, jo komunikācijai tiek izmantots net.pipe protokols.

Lai strādātu net.pipe protokols, Windows servisam Net.Pipe Listener Adapter ir jābūt iestartētam.



3.attēls. Net.Pipe Listener Adapter servisa stāvokļa pārbaude

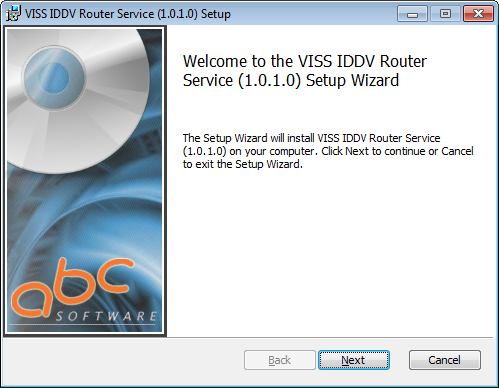
#### Sertifikātu uzstādīšana

Maršrutēšanas servisa autentifikācijai STS un IDDV tīmekļu pakalpes izsaukšanai nepieciešams uzstādīt sertifikātu:

* Jāatver sertifikātu pārvaldība (skatīt sadaļu 3.3.2):
* *Microsoft Management Console -> File -> Add/Remove Snap In -> Certificates -> Add -> Computer account -> Local computer;*
* Jāimportē sertifikāts Console Root\Certificates (Local Computer)\Personal\Certificates mapē (analogi sadaļai 3.3.3):
* *Action -> All tasks -> Import*
* Jāpārliecinās, ka 3.3.1.3 sekcijā norādītajam maršrutēšanas pakalpes pūla lietotājam ir tiesības lasīt sertifikāta privāto atslēgu:
* *Action -> All tasks -> Manage Private keys*

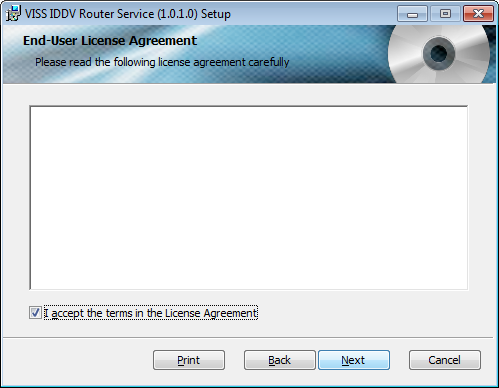
#### Pakalpes uzstādīšana

* Jāpalaiž uzstādīšanas pakotne Abc.Viss.Iddv.RouterService.Setup.msi:



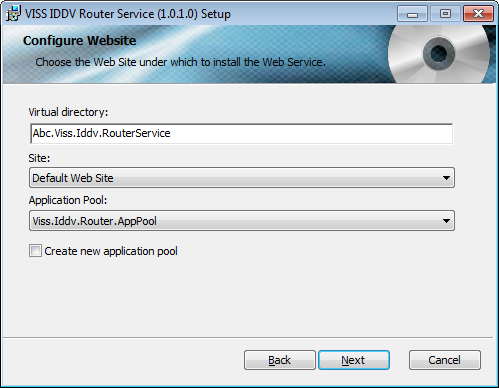
4.attēls. Uzstādīšanas palīga pirmais solis

* Jāapstiprina licences nosacījumi:



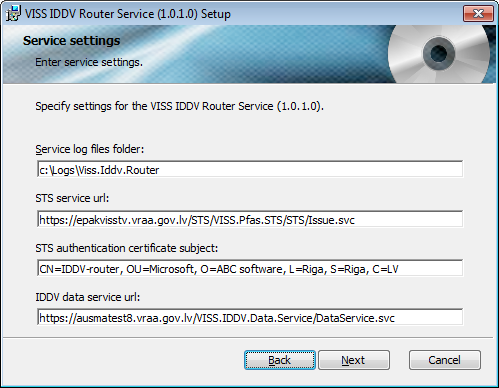
5.attēls. Licences nosacījumu apstiprināšana

* Jānorāda IIS iestatījumi. Izvēloties esošu aplikācijas pūlu, jāņem vērā, ka tā *.Net Framework* versijai jābūt 4.0. Izveidojot jaunu aplikācijas pūlu kā pūla veidotājs tiks uzstādīts *Network Service*.



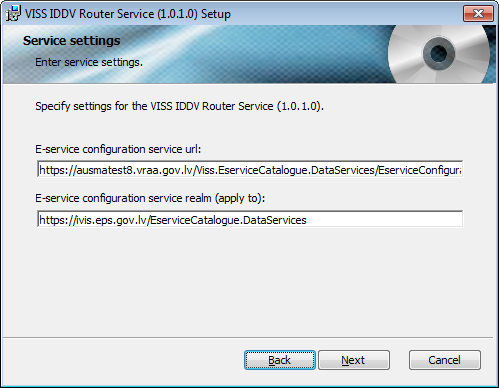
6.attēls. IIS iestatījumu norādīšana

* Pakalpes vispārējo iestatījumu un IDDV datu servisa norādīšana:



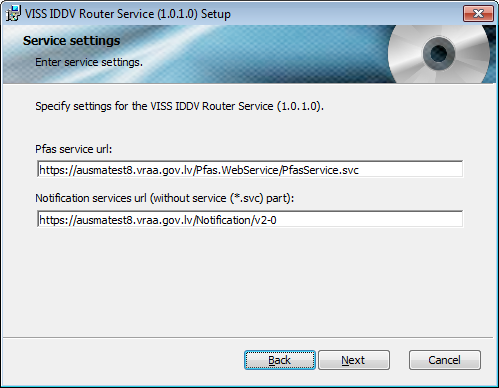
7.attēls. Pakalpes vispārējo iestatījumu norādīšana

* Pakalpes e-pakalpojumu kataloga konfigurācijas servisa norādīšana:



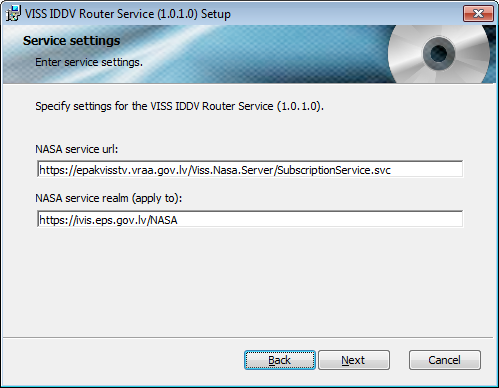
8.attēls. Pakalpes e-pakalpojumu konfigurācijas servisa norādīšana

* Pakalpes PFAS servisa un notifikāciju servisa norādīšana:



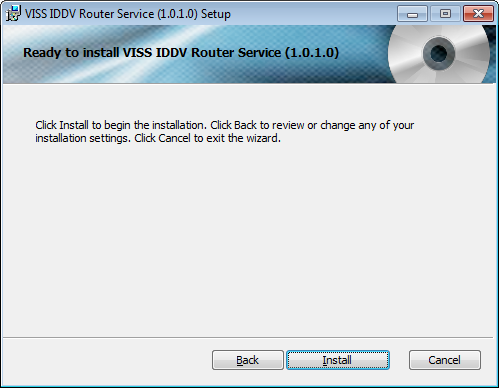
9.attēls. Pakalpes PFAS un notifikāciju servisu norādīšana

* Pakalpes NASA servisa norādīšana:



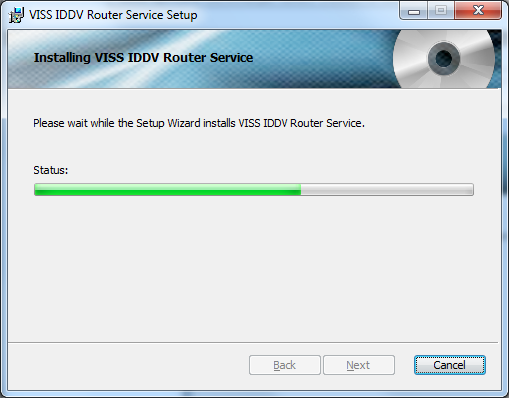
10.attēls. Pakalpes NASA servisa norādīšana

* Jāapstiprina instalācija:



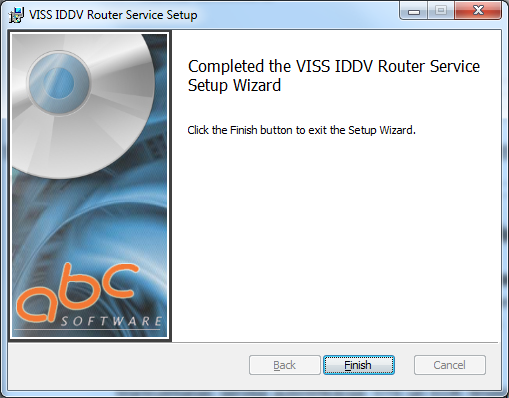
11.attēls. Instalācijas apstiprināšana

* Jāsagaida, kamēr uzstādīšana tiek pabeigta:



12.attēls. Uzstādīšanas process

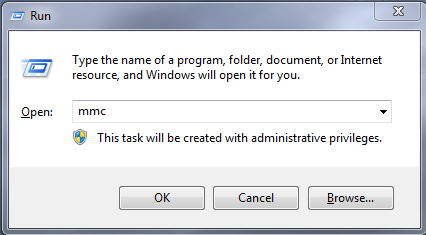
* Jābeidz uzstādīšana:



13.attēls. Sekmīgas uzstādīšanas pēdējais solis

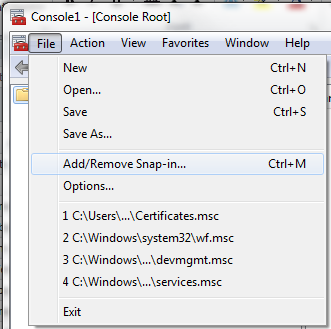
### Sertifikātu pārvaldības rīka atvēršana

1. *Start -> Run -> mmc*



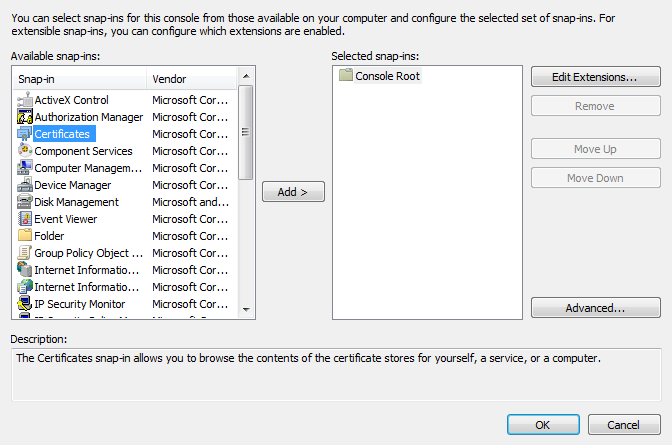
14.attēls. MMC palaišana

1. *File -> Add/Remove Snap-in*



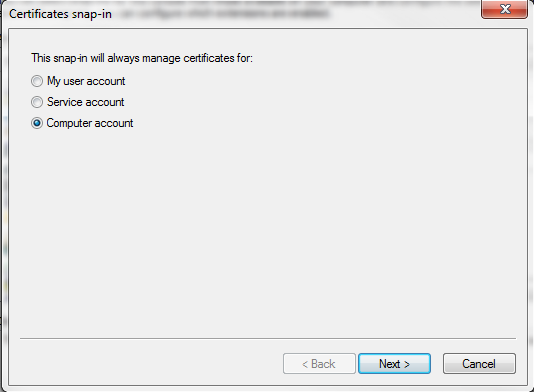
15.attēls. Sertifikātu spraudņa pievienošana (1)

1. *Certificates -> Add*



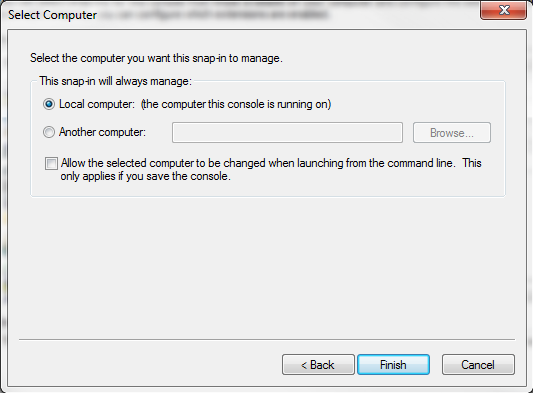
16.attēls. Sertifikātu spraudņa pievienošana (2)

1. *Computer Account*



17.attēls. Sertifikātu spraudņa pievienošana (3)

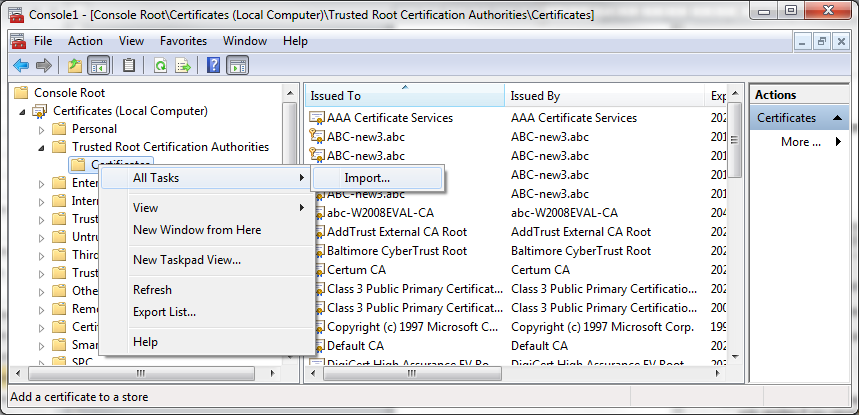
1. *Local computer*



18.attēls. Sertifikātu spraudņa pievienošana (4)

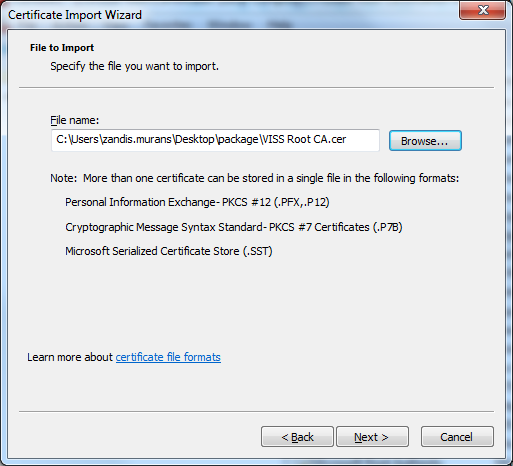
### Saknes sertifikātu imports

1. *All tasks -> Import (no Trusted Root Certification Authorities mapes)*

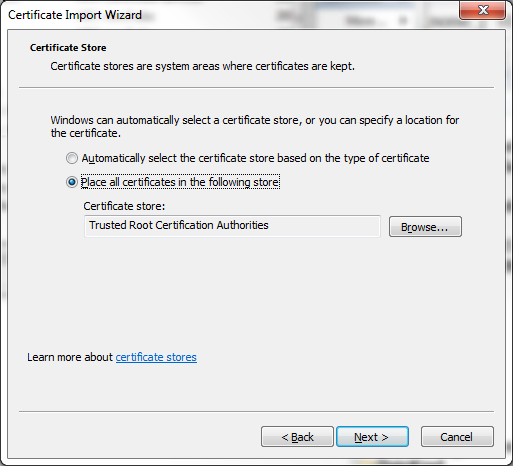


19.attēls. Saknes sertifikāta imports

1. Jāielādē saknes sertifikāta fails *VISS\_Root\_CA.cer*.



20.attēls. Saknes sertifikāta importa vednis (2. solis)



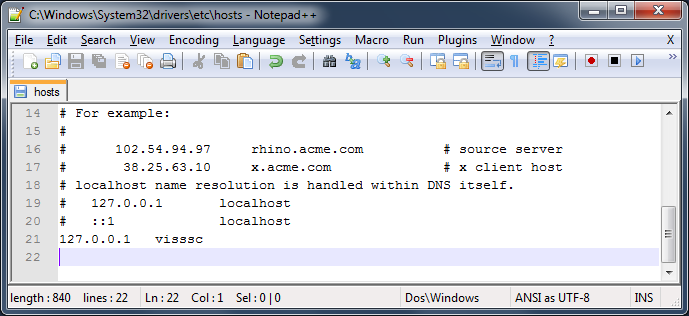
21.attēls. Saknes sertifikāta importa vednis (3. solis)

1. Jāielādē saknes sertifikāta fails *abc-W2008EVAL-CA\_Root\_CA.cer* (skat. Iepriekšējo punktu).

### IDDV formu IIS lietojuma (application) izveide

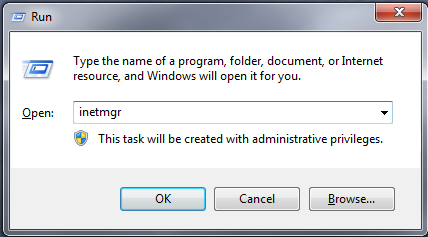
1. *Hosts* datnē jāpievieno rindiņa:

127.0.0.1 visssc



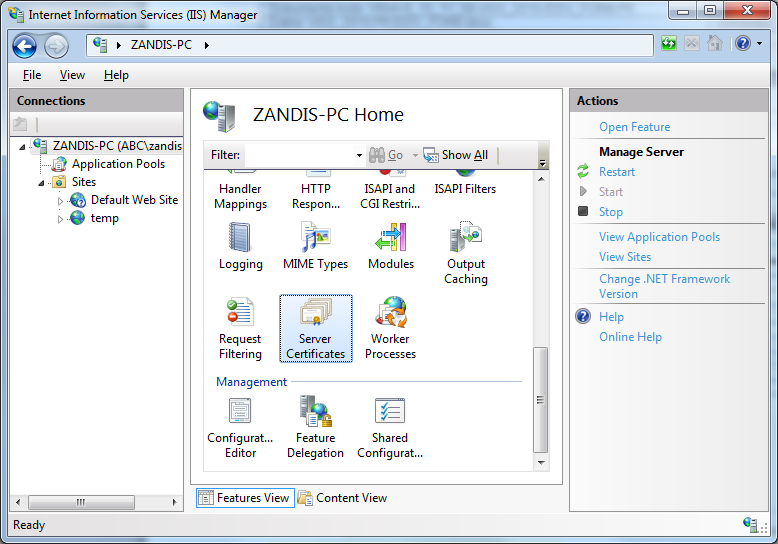
22.attēls. Hosts datnes labošana

1. IIS jāimportē sertifikāts, lai varētu lietojumā nodrošināt https:
   1. Jāatver *IIS Manager (Start -> Run -> inetmgr).*



23.attēls. IIS Manager palaišana

* 1. Jāatver IIS sertifikātu pārvaldība *(Server Certificates).*



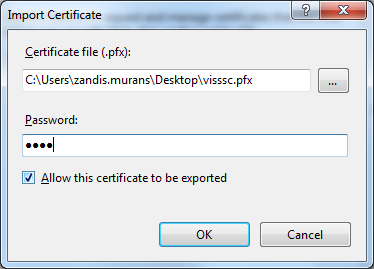
24.attēls. Sertifikātu pārvaldības atvēršana

Jāuzsāk sertifikāta imports *(Import).*



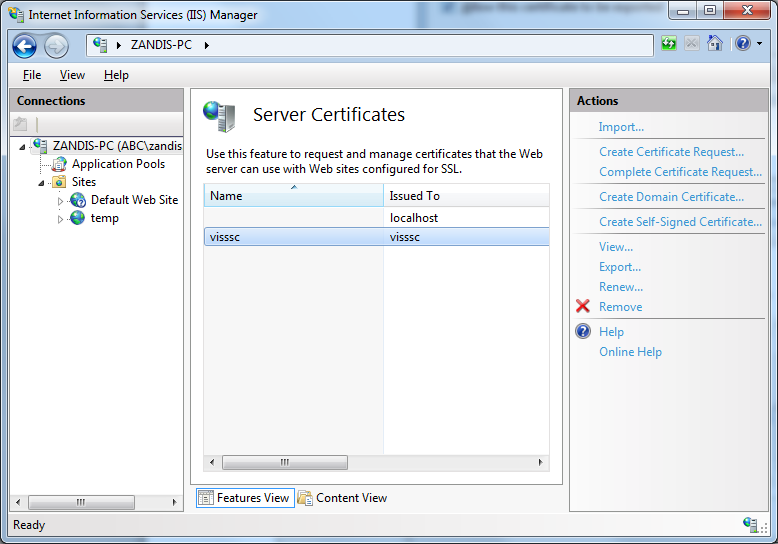
25.attēls. Sertifikātu importa uzsākšana

* 1. Jāielādē *visssc.pfx* datneun jānorāda parole *1234.*



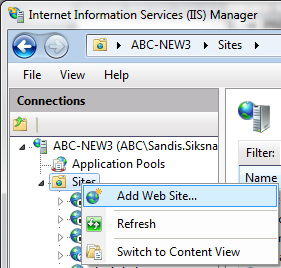
26.attēls. Sertifikāta imports

* 1. Jāpārliecinās, ka sertifikāts ir veiksmīgi importēts.



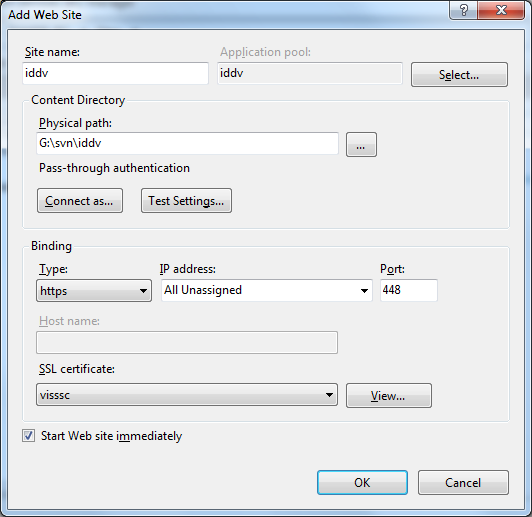
27.attēls. Sertifikāts veiksmīgi importēts

1. IIS jāizveido jauna aplikācija:
   1. *Sites -> Add Web Site.*



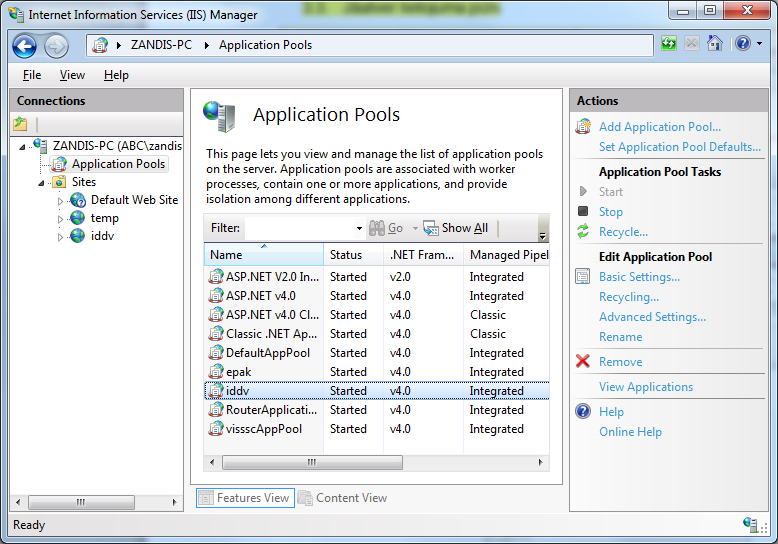
28.attēls. Jauna lietojuma pievienošanas uzsākšana

* 1. Jāievada parametri:
* **Site name**: iddv (var ievadīt brīvi izvēlētu vērtību).
* **Physical path**: ceļš - brīvi izvēlētu mapi (lietojumā nav paredzēts neko darbināt – IDDV forma un emulators tiks uzstādīti kā šī lietojuma apakš-aplikācijas).
* **Type**: https.
* **Port**: 448.
* **SSL certificate**: visssc.



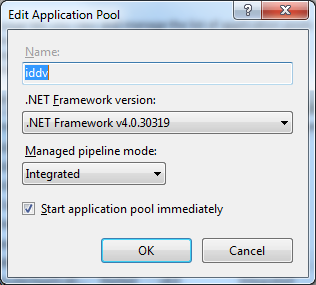
29.attēls. Lietojuma parametri

* 1. Jāatver lietojuma pūls.



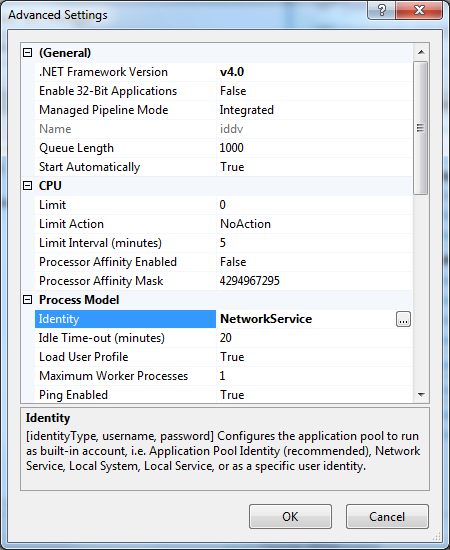
30.attēls. Lietojuma pūla atvēršana

* 1. Jāpārliecinās, ka pūlam ir šādi parametri:
* **.NET Framework version**: v4.0.
* **Managed pipeline mode**: Integrated.



31.attēls. Lietojuma pūla parametri

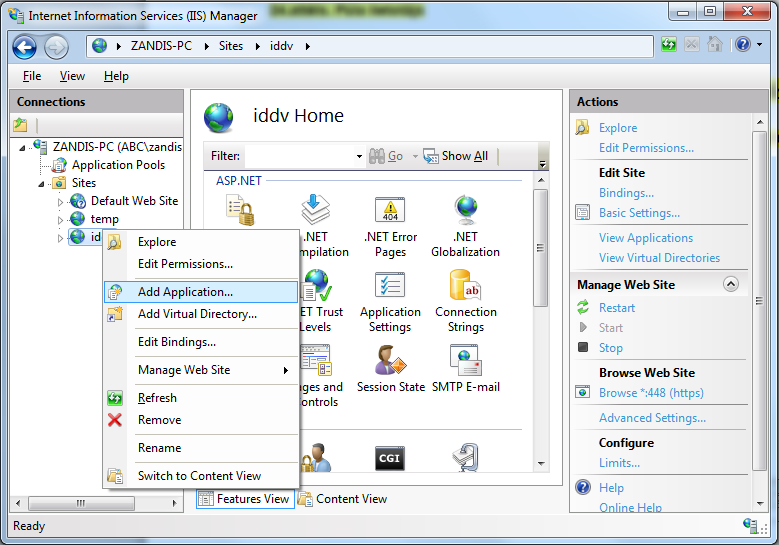
* 1. Jānorāda pūla lietotājs. Piemērā izmantosim *NetworkService*.



32.attēls. Pūla lietotājs

### Portāla emulatora apakš-aplikācijas izveidošana

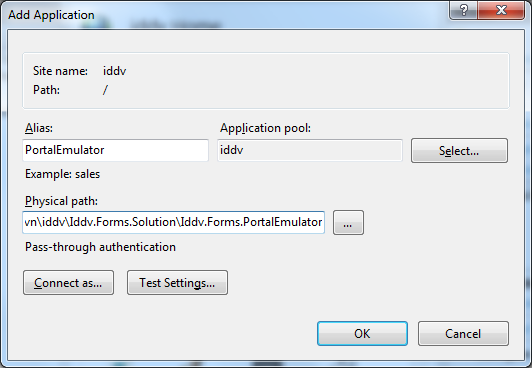
1. Jāatarhivē *Iddv.Forms.Solution.zip* mapes saturs.
2. Jāuzsāk jaunas apakš-aplikācijas izveide (*Add Application*) **iddv** lietojumam.



33.attēls. Apakš-aplikācijas izveides uzsākšana

1. Jāpievieno apakš-aplikācija ar parametriem:

* **Alias**: PortalEmulator;
* **Application pool**: noklusētā vērtība (iddv lietojuma pūls);
* **Physical path**: ceļš uz 1.solī atarhivētās mapes apakšmapi „Iddv.Forms.PortalEmulator”.



34.attēls. Apakš-aplikācijas parametri

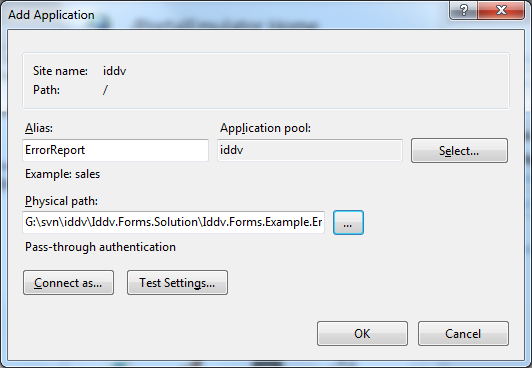
1. Jāpārliecinās, ka iddv pūla lietotājam ir tiesības lasīt un rakstīt apakš-lietojuma mapē.

### IDDV formas apakš-lietojuma izveide IIS

Priekšnosacījumi: izpildītas 3.3.3. un 3.3.4.sadaļu instrukcijas.

Jāpievieno jauna apakšaplikācija ***iddv*** lietojumam (skat. 3.3.5. sadaļu) ar šādiem parametriem:

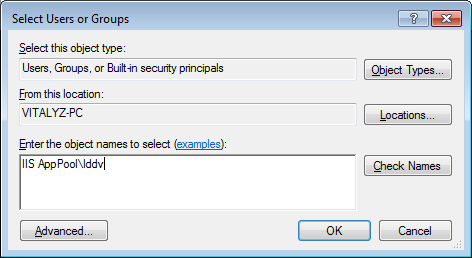
* **Alias**: Apakš-adrese, caur kuru e-pakalpojums būs pieejams, piemēram, *ErrorReport;*
* **Application pool**: noklusētā vērtība (*iddv* lietojuma pūls);
* **Physical path**: ceļš uz IDDV formas projektu.



### Tiesību piešķiršana apakš-lietojumu mapēm

Priekšnosacījumi: izpildītas 3.3.5 un 3.3.6 sadaļu instrukcijas.

Lai varētu piekļūt izveidotajiem apakš-lietojumiem un nepieciešamības gadījumā rakstīt žurnalēšanas datus, nepieciešams apakš-lietojuma pūlam piešķirt lasīšanas/rakstīšanas tiesības apakš-lietojuma mapei.



### IDDV emulatora un formu sertifikāta reģistrēšana

Katram IDDV emulatoram un formai ir nepieciešams ģenerēt sertifikātu, ar kuru tiek parakstīts pieprasījums pēc drošības talona. Šī sertifikāta publiskajai atslēgai jābūt reģistrētai arī VISS (jāsazinās ar EPU).

Jāveic sertifikātu uzstādīšana un privātās atslēgas pieejas tiesību piešķiršana, kā tas ir aprakstīts 3.3.3.sadaļā.

Visbeidzot uzstādītais sertifikāts jāpiefiksē projekta konfigurācijas datnē (*web.config*):

<system.serviceModel>

<behaviors>

<endpointBehaviors>

…

<behavior name="certificate">

<clientCredentials>

<clientCertificate

findValue="CN=ABC.EPAK.EP123, L=Riga, S=Riga, C=LV"

storeLocation="LocalMachine"

storeName="My"

x509FindType="FindBySubjectDistinguishedName" />

</clientCredentials>

</behavior>

<endpointBehaviors>

<behaviors>

<system.serviceModel>

### IDDV emulatora un formu īpašas piekļuves apgabala (realm) reģistrēšana

Lai darbotos SSO, VISS vides STS konfigurācijā nepieciešams IDDV emulatoram un formai reģistrēt īpašas piekļuves apgabalu (*realm*).

EPU reģistrēto apgabalu nepieciešams piefiksēt IDDV emulatora un formas konfigurācijas failā (*web.config*):

<microsoft.identityModel>

<service saveBootstrapTokens="true">

…

<audienceUris>

<add value="http://ivis.abcsoftware.lv/ep123"/>

</audienceUris>

…

<federatedAuthentication>

<wsFederation

issuer="https://ivistv.abcsoftware.lv/IVIS.Pfas.STS/Default.aspx"

realm="http://ivis.abcsoftware.lv/ep123"/>

<cookieHandler requireSsl="false"/>

</federatedAuthentication>

…

</service>

</microsoft.identityModel>

## CDN uzstādīšana

Portāla standarta resursi (attēli, JS un CSS datnes) IDDV formai tiek piegādāti, izmantojot atsevišķu tīmekļa lietojumu (CDN).

Izstrādes videi šis lietojums ir iebūvēts emulatorā.

CDN paredzēts IDDV koplietojamo (atkārtoti izmantojamo) resursu glabāšanai. Šāds risinājums samazina koda dublēšanu, ļauj veikt resursu optimizāciju (*minification*), kā arī uzlabo tīmekļa pārlūka kešdarbes iespējas. IDDV formu specifiskos resursus paredzēts glabāt pašā formā.

Izstrādes vidē var izmantot gan emulatorā iebūvēto, gan atsevišķā aplikācijā izdalīto CDN.

Konfigurācijas piemērs emulatorā iebūvētā CDN izmantošanai.

<iddv>

…

cdnUrl="https://visssc:448/PortalEmulator/Cdn/v1.0"

…

</iddv>

Konfigurācijas piemērs CDN apakš-lietojuma izmantošanai.

<iddv>

…

cdnUrl="https://visssc:448/Cdn/v1.0"

…

</iddv>

Jaunākajās emulatora versijās vēlams versijas numuru nenorādīt tieši, bet tā vietā ievietot marķieri „*{{version}}*”. Šādā gadījumā versija tiks ņemta no e-pakalpojumu katalogā reģistrētās *CdnVersion* lauka vērtības vai no emulatora konfigurācijas failā reģistrētās formas konfigurācijas (t.i. CDN versiju konfigurāciju nevajadzēs dublēt).

Konfigurācijas piemērs izmantojot versijas marķieri.

<iddv>

…

cdnUrl="https://visssc:448/Cdn/{{version}}"

…

</iddv>

# IDDV formas realizēšana

Lai realizētu IDDV formu, nepieciešams paplašināt šādas klases un implementēt mainīgās daļas, kas apkopotas 1.tabulā.

1.tabula

Klašu saraksts

| Nosaukums | Apraksts |
| --- | --- |
| *Task* | Darba uzdevuma aprakstoša klase. Klase paredzēta darbam ar darba uzdevuma aprakstošo XML – uzglabāšana, ielāde/saglabāšana utml. |
| *ITaskPage* | Darba uzdevuma reprezentatīvās daļas aprakstošā klase. Klase paredzēta darbam ar ievades formu – validācija, kontroļu aizpilde, lietotāja ievades nolasīšana utml. |

## Klase Task<T>

*Generic* tipu atšifrējums dots 2.tabulā, bet implementējamo metožu saraksts 3.tabulā.

2.tabula

Generic tipu atšifrējums

| Nosaukums | Apraksts |
| --- | --- |
| T | Darba uzdevuma reprezentējošās XML klases tips |

3.tabula

Implementējamās metodes

| Nosaukums | Obligāts | Apraksts |
| --- | --- | --- |
| *void LoadXmlInstance(string messageGuid)* | Nē | Metodi implementē, lai pēc norādītā darba uzdevuma GUID ielādētu darba uzdevuma XML un saglabātu to klases mainīgajā “*XmlInstance*” |
| *void SaveXmlInstance()* | Nē | Metodi implementē, lai saglabātu ielādēto XML (pretēji *LoadXmlInstance())* |

## Klase ITaskPage<T, K>

Lai izveidotu IDDV formu, jārealizē jauna ASPX lapa, kura manto no bāzes klases *TaskPage<T, K>.* *Generic* tipu atšifrējums dots 4.tabulā, bet implementējamo metožu saraksts 5.tabulā.

4.tabula

Generic tipu atšifrējums

| Nosaukums | Apraksts |
| --- | --- |
| T | Darba uzdevumu reprezentējošās XML klases tips |
| K | Formai atbilstošā galamērķu (*Milestone*) tips (*Enum*) |

5.tabula

Implementējamās metodes

| Nosaukums | Obligāts | Apraksts |
| --- | --- | --- |
| *void LoadForm()* | Jā | Metodi implementē, lai aizpildītu kontroles ar vērtībām, kas izlasītas no XML |
| *void CreateXmlStructure()* | Jā | Metodi implementē, lai izveidotu pareizu (normalizētu) XML instances struktūru. |
| *void SaveForm()* | Jā | Metodi implementē, lai saglabātu kontrolēs ievadītās vērtības atpakaļ XML (pretēji *LoadForm())* |
| *void EnableDisableForm()* | Jā | Metodi implementē, lai iespējotu/atspējotu kontroles |
| *string GetMilestoneLabel(K milestone)* | Nē | Metodi implementē, lai atgrieztu norādītā formas galamērķa virsrakstu. |
| *void PerformPreSaveActions()* | Nē | Metodi implementē, lai veiktu jebkādas pirmssaglabāšanas operācijas. Metode tiek izpildīta tikai tajos gadījumos, kad formas ievades pareizības pārbaude ir noritējusi veiksmīgi. |
| *System.Web.UI.WebControls.HiddenField GetAdministrativeField1()* | Jā | Metodi implementē, lai norādītu uz formas slēpto lauku, kas nepieciešams administratīvajām vajadzībām. |
| *System.Web.UI.WebControls.HiddenField GetAdministrativeField2()* | Jā | Metodi implementē, lai norādītu uz formas slēpto lauku, kas nepieciešams administratīvajām vajadzībām. |

Realizējot šo klasi, ir jānodrošina, lai tā saturētu darba uzdevumu aprakstošās klases objektu (*Task* implementācija). Šis objekts saglabājams klases mainīgajā “*Task*“. Lai atvieglotu darbu ar objekta saglabāšanu sesijā, klasē *TasksHelper* ir izveidotas palīgmetodes — *LoadTaskFromSession()* un *PrepareTask().*

Sesijas objekta ielādes/saglabāšanas piemērs:

// load form from session

if (!TasksHelper.LoadTaskFromSession<IVISErrorReportStructure, ErrorReportMilestone>(this.Session, this))

{

// if form was not found in session, try to create new one

if (TasksHelper.PrepareTask<IVISErrorReportStructure, ErrorReportMilestone>(new Task<IVISErrorReportStructure> (this.User), this.Session, this, this.Request.QueryString["MessageId"]))

{

// enable/disable controls in form

this.EnableDisableForm();

}

else

{

// unable to prepare task form

throw new Exception("Undetermined exception");

// TODO: Determine error

}

}

### Administratīvie lauki

| Nosaukums | Apraksts |
| --- | --- |
| *AdministrativeField1* | Lauks glabā darba uzdevuma ID. Darba uzdevuma ID tiek izmantots sesijas objekta atslēgā. |
| *AdministrativeField2* | Lauks glabā informāciju par darba uzdevuma pielikumiem serializētā formātā. Sīkāk par darba uzdevuma pielikumiem un serializēšanas piemēru skatīties sadaļā 4.2.1 |

### Darba uzdevuma pielikumi

Darba uzdevuma pielikumi tiek glabāti Elektronisko dokumentu krātuvē (EDK).

Lai darba uzdevumam piesaistītu pielikumus, saglabājot darba uzdevumu, šis lauks ir jāaizpilda ar pielikumu aprakstu serializētā formātā.

Darba uzdevuma pielikumu serializēšanas piemērs:

// create attachments structure

var attachment1 = new Iddv.Forms.Attachment

{

Name = "URN:IVIS:100266:DOC-2266874-V0.0",

Description = "Darba uzdevuma pielikuma piemērs 1"

};

var attachment2 = new Iddv.Forms.Attachment

{

Name = "URN:IVIS:100266:DOC-2266875-V0.0",

Description = "Darba uzdevuma pielikuma piemērs 2"

};

var attachments = new Iddv.Forms.Attachments

{

Items = new System.Collections.Generic.List<Iddv.Forms.Attachment>(2)

{

attachment1,

attachment2

}

};

// create serializer

var serializer = new System.Runtime.Serialization.Json.DataContractJsonSerializer(typeof(Iddv.Forms.Attachments));

using (var stream = new System.IO.MemoryStream())

{

// serialize object

serializer.WriteObject(stream, attachments);

// get serialized data

stream.Position = 0;

using (var streamReader = new System.IO.StreamReader(stream))

{

this.GetAdministrativeField2().Value = streamReader.ReadToEnd();

}

}

### Ievades pareizības pārbaude

Formas ievades pareizības pārbaudei jānotiek gan klienta pusē — gan servera pusē. Klienta puses pareizības pārbaude realizējama ar ASP.NET iebūvētajiem pareizības pārbaudes rīkiem (skatīt: <http://msdn.microsoft.com/en-us/library/bwd43d0x(v=vs.100).aspx)>.

Lai atvieglotu kļūdu paziņojumu parādīšanu, ITaskPage<T, K> šim nolūkam tiek piedāvātas palīgmetodes:

##### Metode „AddErrorMessage”

Metodes apraksts:

void AddErrorMessage(string error)

Izvada uz ekrāna kļūdas paziņojumu.

Metodes parametru apraksts:

|  |  |
| --- | --- |
| Nosaukums | Apraksts |
| error | kļūdas paziņojuma teksts |

##### Metode „AddErrorMessage”

Metodes apraksts:

void AddErrorMessage(Exception e, string error = null, bool hideExceptionDetails = false)

Izvada uz ekrāna kļūdas paziņojumu.

Metodes parametru apraksts:

|  |  |
| --- | --- |
| Nosaukums | Apraksts |
| e | kļūda |
| error | kļūdas paziņojuma teksts |
| hideExceptionDetails | pazīme tam, lai netiktu kļūdas paziņojumā iekļautas kļūdas detaļas |

Piemērs:

protected override void PerformPreSaveActions()

{

try

{

(1 / (int)Math.Sin(0)).ToString();

}

catch (Exception e)

{

base.AddErrorMessage(e, "Saglabājot darba uzdevumu, radās kļūda", false);

}

}

## Pielikumu validācijas scenārijs

Lai veiktu lietotāja pievienota vai darba uzdevumā saņemta pielikuma validāciju nepieciešams veikt sekojošas darbības:

1. Iegūt pielikumu sarakstu izsaucot portāla JS funkciju *GetAttachments*. Funkcija atgriez pielikumu Urn un aprakstu sarakstu;

iddv.portalProxy.sendCommandToPortal("GetAttachments");

1. Pēc pielikuma Urn izgūt dokumenta no EDK izsaucot *ITaskPage* metodi *GetEdkDocument;*
2. Saņemtā dokumenta parakstu validē izsaucot *ITaskPage* metodi *ValidateDocument*;
3. Atbilstoši apstrādā saņemtos validācijas rezultātus.

Pielikumu validācijas piemērs aplūkojams Iddv formā *Iddv.Forms.Example.ErrorReport.Web*.

Piemērs *ITaskPage* metožu izsaukšanai:

string item = this.uxAttachemntList.Text;

if (!string.IsNullOrWhiteSpace(item)) {

var content = GetEdkDocument(item);

if (content == null) {

throw new Exception("Content doesn't exist!");

}

using (var stream = new System.IO.MemoryStream(content.stream)) {

AddEdocInMsg document = new AddEdocInMsg(

Path.GetExtension(content.filename).Trim('.'),

content.filename,

content.stream.Length,

stream);

var validationResult = ValidateDocument(document);

if (validationResult != null && validationResult.Count > 0) {

// ToDo Implementēt nepieciešamās validācijas XML pārbaudes, kļūdas gadījumā izvadīt atbilstošu paziņojumu.

}

else {

AddErrorMessage("Pielikuma validācija nesekmīga!");

}

}

}

Lai varētu veikt pielikumu validāciju jābūt norādītai maršrutēšanai apkalpei darbam ar Edk (edkServiceEndpointAddress parametrs), kā arī nepieciešams konfigurēt *esign* (e-parakstītāja) sekciju formas konfigurācijas datnē *web.config*:

<configSections>

<section name="esign" type="Abc.Viss.Esigner.EsigningServices.Config.EsignConfig,

Abc.Viss.Esigner" />

</configSections>

<esign>

<general signingRouterServiceUrl="[https://visssc:448/Abc.Esigner.RouterService /RouterService.svc](https://visssc:448/Abc.Esigner.RouterService%20/RouterService.svc)" />

<esigner javaScriptLibraryUrl="[https://signetest.vraa.gov.lv/WebStatic/Scripts /Zzdats.Div.DocumentViewer.min.js](https://signetest.vraa.gov.lv/WebStatic/Scripts%20/Zzdats.Div.DocumentViewer.min.js)"

statusServerUrl=<https://signetest.vraa.gov.lv/WebApi/>

staticDataBaseUrl="<https://signetest.vraa.gov.lv/WebStatic/>" />

</esign>

E-parakstīšanas maršrutēšanas pakalpes uzstādīšanas instrukciju skatīt dokumentā [5]

## Pielikuma izveides scenārijs

Standarta gadījumā pielikumu pievienošanu veic lietotājs ārpus Iddv formas līdz ar to formu izstrādē nav nepieciešams veikt nekādu papildus apstrādi.

Izņēmuma situācijās, kad nepieciešams izveidot pielikumu pašā formā, nepieciešams veikt sekojošas darbības:

1. Izveidot vai ielādēt dokumentu formā;
2. Izveidot Edk direktoriju dokumenta ielādei izsaucot *ITaskPage* metodi *GetOrCreateEdkFolder*;
3. Ielādēt dokumentu izveidotājā direktorijā izsaucot *ITaskPage* metodi *CreateEdkDocument;*
4. Pievienot izveidotā dokumenta Urn formas pielikumu sarakstam izsaucot *Task* metodi *ProcessAttachemtns*;

Pielikumu validācijas piemērs aplūkojams Iddv formā *Iddv.Forms.Example.ErrorReport.Web*.

Piemērs *ITaskPage* un *Task* metožu izsaukšanai:

try {

string testDocumentData = "Testing 1-2-3";

byte[] byteArray = Encoding.UTF8.GetBytes(testDocumentData);

MemoryStream stream = new MemoryStream(byteArray);

string path = string.Format(ConfigManager.GetIddvConfigSection().EdkDirectoryPattern, GetAuthority(this.User as IClaimsPrincipal), System.DateTime.Now, this.Task.MessageGuid);

string folderId = GetOrCreateEdkFolder(path);

string testFileName = string.Format("doc{0:yyyyMM}\_{0:ddHHmm}.txt", System.DateTime.Now);

string objectId = CreateEdkDocument(FillProperties(testFileName), folderId, FillStream(testFileName, byteArray));

this.Task.ProcessAttachments.Add(objectId, "Sistemas izveidota datne");

}

catch (Exception ex) {

// ToDo "502" -> "Fails jau eksistē!"

AddErrorMessage(ex, "Fails izveides kļūda!", false);

}

Lai varētu veikt pielikumu izveidi jābūt norādītai maršrutēšanai apkalpei darbam ar Edk ( parametri edkServiceEndpointAddress un edkDirectoryPattern) formas konfigurācijas datnē *web.config*:

## Sesijas izmantošana

Ja izstrādājamajā formā nepieciešams izmantot sesiju datu glabāšanai, nepieciešams veidot formas instancei unikālus sesijas mainīgo nosaukumus. Tas nepieciešams, lai nerastos situācijas kad atverot dažādus darba uzdevumus notiktu nekorekta sesijas mainīgo vērtību uzstādīšana vai nolasīšana.

To visērtāk ir realizēt sesijas mainīgo nosaukumā iekļaujot darba uzdevuma identifikatoru.

Session["myVariable\_" + this.Task.MessageGuid] = "value";

## Darba uzdevumu galamērķi

Lai izpildītu darba uzdevumu, lietojumam šis darba uzdevums ir jāpārsūta uz kādu no galamērķiem (*Milestone*). Lai organizētu darba uzdevuma pārsūtīšanu uz kādu no galamērķiem, galamērķu identifikatori tiek glabāti iekšējā lietojuma konfigurācijas failā.

Piemērs:

<appSettings>

<add key="Milestone.Authority" value="URN:IVIS:100001:EP-IVISErrorReport-v1-0-MS-AuthAndInh"/>

<add key="Milestone.AuthorityAndPerson" value="URN:IVIS:100001:EP-IVISErrorReport-v1-0-MS-SendToAuth"/>

<add key="Milestone.None" value="URN:IVIS:100001:EP-IVISErrorReport-v1-0-MS-Discarted"/>

</appSettings>

Lai sasaistītu galamērķi ar konfigurācijas failā esošajām atslēgām, definējot galamērķa enumerācijas elementu, anotācijā „*Description*” norādām konfigurācijas ieraksta atslēgu.

Piemērs:

public enum ErrorReportMilestone

{

/// <summary>

/// Milestone accepted

/// </summary>

[Description("Milestone.Authority")]

Accepted,

/// <summary>

/// Milestone accepted with notification

/// </summary>

[Description("Milestone.AuthorityAndPerson")]

AcceptedWithNotification,

/// <summary>

/// Milestone None

/// </summary>

[Description("Milestone.None")]

None

}

## Portāla emulatora konfigurēšana

Lai atvieglotu izstrādi, emulatoru un izstrādājamās formas var konfigurēt darbam failu sistēmas vai servisa režīmā. Lai pārslēgtos starp režīmiem jāveic emulatora web.config datnes modificēšana.

Produkcijas vidē failu sistēmas režīms nav pieejams.

Konfigurācijas atribūti, kas attiecās uz darbības režīmu ir sekojoši:

| Nosaukums | Apraksts |
| --- | --- |
| iddv/transport | XML saglabāšanas veids (darba režīms). Iespējamās vērtības:  „service” – bez emulācijas (tiek strādāts ar web-servisiem);  „fs” (file system) – ar emulāciju (tiek strādāts ar XML datnēm lokālajā failu sistēmā).  Neobligāts parametrs, noklusētā vērtība „service”. |
| iddv/transportArgs | Saglabāšanas veida argumenti. Atkarībā no saglabāšanas veida, saglabāšanas veida argumentos var norādīt:  Saglabāšanas veids „service” – parametrs netiek ņemts vērā – to var nenorādīt;  Saglabāšanas veids „fs” – ceļu uz direktoriju, kurā tiks meklētas XML datnes. Direktorijai jābūt iepriekš izveidotai un jāsatur xml datnes, kas organizētas pēc zemāk definētās struktūras. |
| iddv/transportAuthRequired | Pazīme vai izvēlētajam saglabāšanas veidam būs nepieciešama lietotāja autentifikācija. Iespējamās vērtības „true” vai „false”.  Neobligāts parametrs, noklusētā vērtība „true”. |

Citi emulatora konfigurācijas atribūti.

| Nosaukums | Apraksts |
| --- | --- |
| iddv/general/iddvUrn | Emulatorā šis parametrs netiek izmantots. |
| iddv/general/cdnUrl | Jānorāda CDN tīmekļa lietojuma adrese ar izstrādājamajai Iddv formai atbilstošo versijas mapi noslēgumā. Parasti adrese jāveido kā 3.4 nodaļā uzstādītā lietojuma adrese (kopā ar virtuālās direktorijas nosaukumu) + „*CDN*” + versijas numurs. Vēlams versijas numuru nenorādīt tieši, bet tā vietā ievietot marķieri „*{{version}}*” (piemēram, „*https://vraa.gov.lv/Iddv.Cdn/CDN/{{version}}*”). Šādā gadījumā versija tiks ņemta no e-pakalpojumu katalogā reģistrētās *CdnVersion* lauka vērtības vai no emulatora konfigurācijas failā reģistrētās formas konfigurācijas (t.i. CDN versiju konfigurāciju nevajadzēs dublēt). |
| iddv/general/routerServiceUrl | Jānorāda IDDV formu maršrutēšanas pakalpes adrese (3.3.1 nodaļā uzstādītā lietojuma adrese + „/RouterService.svc”). Adresei jāizmanto net.pipe protokols, piemēram, net.pipe://lokalhost/Viss.Iddv.RouterService/RouterService.svc. |

Lai sasaistītu darba uzdevumus ar atbilstošām IDDV formām ir jāveic darba uzdevumu un izveidoto IDDV formu kartēšana, pēc formas URN un IDDV formas IIS lietojuma nosaukuma. Kartēšana tiek veikta emulatora konfigurācijas datnē un par to atbild sekojoši parametri:

Formas reģistrēšanas konfigurācijas parametri emulatorā.

| Nosaukums | Apraksts |
| --- | --- |
| iddv/forms/form/destination | Jānorāda Iddv formas identifikators. Servisa režīmā tas saturēs formas URN, bet failu sistēmas režīmā direktorijas nosaukumu, kas satur ieejas xml datni. Parametram jāsakrīt ar formās konfigurācijas datnē norādīto URN. Lai atvieglotu konfigurēšanu paralēli var izveidot gan konfigurāciju priekš failu sistēmas, gan servisa režīma un modificēt tikai formas iddv/general/iddvUrn, lai noteiktu kuru no tām izmantot. |
| iddv/forms/form/url | Jānorāda Iddv formas IIS lietojuma adrese bez jebkādiem papildus parametriem, piemēram, „https://visssc:448/ErrorReport”. |
| iddv/forms/form/cdnVersion | Jānorāda Iddv formas izmantotā CDN versija, piemēram „v1.0” vai „v1.1”. |
| iddv/forms/form/title | Iddv formas virsraksts latviešu valodā. (citas valodas emulatorā nav iespējams uzstādīt). Neobligāts parametrs ar noklusēto vērtību „Noklusētais IDDV formas virsraksts”. |
| iddv/forms/form/ eserviceIsActiveVersion | Nosaka vai Iddv forma atrodas zem aktīvās e-pakalpojuma versijas. neobligāts parametrs ar noklusēto vērtību "true". |
| iddv/forms/form/eserviceStatus | Iddv formas e-pakalpojuma status. Neobligāts parametrs ar noklusēto vērtību „Published”. Iespējams uzstādīt sekojošas vērtības: „Published”, „Closed”, „Deleted”, „Development”. |

### Portāla konfigurēšana servisu režīmā

Servisu režīmā darba uzdevumi tiek iegūti no Iddv maršrutēšanas pakalpes.



35.attēls. Formas ielāde servisa režīmā.

Emulatora konfigurācijas piemērs darbam servisu režīmā.

<iddv

transport="service"

transportAuthRequired="true">

<general

iddvUrn="Emulatorā netiek izmantots!"

cdnUrl="https://visssc:448/PortalEmulator/Cdn/{{version}}"

routerServiceUrl="net.pipe://localhost/Abc.Viss.Iddv.RouterService /RouterService.svc"/>

<forms>

<form

destination="FileSystemTransportDestination-ErrorReport"

url="https://visssc:448/ErrorReport"

cdnVersion="v1.1"

title="Iddv kļūdas formas piemērs."

/>

<form

destination="URN:IVIS:100001:EP-IVISErrorReport-v1-0-IDDVFORM2-ErrorReport"

url="https://visssc:448/ErrorReport"

cdnVersion="v1.1"

title="Iddv kļūdas formas piemērs."

/>

</forms>

</iddv>

Pēc šādas konfigurācijas verot vaļā darba uzdevumu „X”, kuram īpašība „*destination*” būs „URN:IVIS:100001:EP-IVISErrorReport-v1-0-IDDVFORM2-ErrorReport”, iekšējā lietojuma (*iframe*) saite tiks veidota kā:  [https://visssc:448/ErrorReport/?MessageIs=X](%20https://visssc:448/ErrorReport/?MessageIs=X).

### Portāla konfigurēšana failu sistēmas emulācijas režīmā

Lai atvieglotu izstrādi emulatoru un formas aplikāciju iespējams konfigurēt strādāšanai failu sistēmas režīmā.

Failu sistēmas režīmā darba uzdevuma dati tiek nodoti kā XML datnes, kas tiek ņemtas no lokālās failu sistēmas.



36.attēls. Formas ielāde failu sistēmas režīmā bez autentifikācijas.

Konfigurācijas piemērs, strādāšanai failu sistēmas režīmā:

<iddv

transport="fs"

transportArgs="C:\IddvFormsTransportEmulator"

transportAuthRequired="true">

<general

iddvUrn="Emulatorā netiek izmantots!"

cdnUrl="https://visssc:448/PortalEmulator/Cdn/{{version}}"

routerServiceUrl="net.pipe://localhost/Abc.Viss.Iddv.RouterService /RouterService.svc"/>

<forms>

<form

destination="FileSystemTransportDestination-ErrorReport"

url="https://visssc:448/ErrorReport"

cdnVersion="v1.1"

title="Iddv kļūdas formas piemērs."

/>

<form

destination="URN:IVIS:100001:EP-IVISErrorReport-v1-0-IDDVFORM2-ErrorReport"

url="https://visssc:448/ErrorReport"

cdnVersion="v1.1"

title="Iddv kļūdas formas piemērs."

/>

</forms>

</iddv>

Failu sistēmas emulācijas režīmā atribūta „*transportArgs”* saturošajā direktorijā datnēm ir jābūt izkārtotām šādā struktūrā:

[Saknes direktorija] -> [Galamērķa direktorija] -> [*a.xml*, *b.xml*, *c.xml*, …]

Piemērs:



Šī konfigurācijas sekcija ir jāsaglabā kā Portāla konfigurācijas failā — tā iekšājās aplikācijas konfigurācijas failā.

## Iddv formas konfigurēšana

Tā kā emulators spēj darboties divos režīmos, failu sistēmas un servisa, tad arī formai ir nepieciešams norādīt kuru no tiem izmantot.

Iddv formas konfigurācijas atribūti.

| Nosaukums | Apraksts |
| --- | --- |
| iddv/general/iddvUrn | Servisa režīmā jānorāda EPK piešķirtais formas URN, tas tiks izmantots lai no EPK servisa vai emulatora iegūtu formas konfigurāciju.  Failu sistēmas režīmā jānorāda iddv/transportArgs parametrā norādītās direktorijas apakšdirektorija kas satur ieejas xml. |
| iddv/general/cdnUrl | Jānorāda CDN tīmekļa lietojuma adrese ar izstrādājamajai Iddv formai atbilstošo versijas mapi noslēgumā. Parasti adrese jāveido kā 3.4 nodaļā uzstādītā lietojuma adrese (kopā ar virtuālās direktorijas nosaukumu) + „*CDN*” + versijas numurs. Vēlams versijas numuru nenorādīt tieši, bet tā vietā ievietot marķieri „*{{version}}*” (piemēram, „*https://vraa.gov.lv/Iddv.Cdn/CDN/{{version}}*”). Šādā gadījumā versija tiks ņemta no e-pakalpojumu katalogā reģistrētās *CdnVersion* lauka vērtības vai no emulatora konfigurācijas failā reģistrētās formas konfigurācijas (t.i. CDN versiju konfigurāciju nevajadzēs dublēt). |
| iddv/general/routerServiceUrl | Jānorāda IDDV formu maršrutēšanas pakalpes adrese (3.3.1 nodaļā uzstādītā lietojuma adrese + „/RouterService.svc”). Adresei jāizmanto net.pipe protokols, piemēram, net.pipe://lokalhost/Viss.Iddv.RouterService/RouterService.svc. |
| iddv/general/transportedSchemaUrl | Jānorāda shēmas adrese pēc kuras validēt izejošo darba uzdevuma xml. Neobligāts parametrs, kas jānorāda tikai izstrādes vidē un darbojas tikai izmantojot failu sistēmas režīmu. |
| iddv/general/portalConfigCacheTimeoutInSeconds | Nosaka, cik ilgi kešatmiņā tiek glabāti no e-pakalpojumu konfigurācijas servisa izgūtie iestatījumi. Pēc noklusējuma 1 gads. Norādot vērtību 0, rezultāti kešatmiņā glabāti netiek. Produkcijā vēlams šo iestatījumu nenorādīt (noklusētā vērtība), jo formai tiek sūtīti ziņojumi par konfigurācijas izmaiņām. Saņemot šādu ziņojumu, kešatmiņa tiek tīrīta. |
| iddv/general/speedLogEnabled | Nosaka, vai veikt servisu izsaukumu laika trasēšanu. Pēc noklusējuma True. |
| iddv/edk/edkServiceEndpointAddress | Jānorāda EDK maršrutēšanas pakalpes adrese + „/RouterService.svc”. Adresei jāizmanto net.pipe protokols, piemēram, net.pipe://lokalhost/Viss.Edk.RouterService/RouterService.svc.  Neobligāts parametrs, nepieciešams aizpildīt ja formā tiek veikta pielikumu ģenerēšana vai to validācija. |
| iddv/edk/edkDirectoryPattern | Jānorāda adreses šablons pēc kuras EDK tiks glabāti formā izveidotie pielikumi. Neobligāts parametrs, nepieciešams aizpildīt ja formā tiek veikta pielikumu ģenerēšana vai to validācija. |
| iddv/nasa/topic | NASA pieraksta temats (pēc noklusējuma „IddvConfigChange”). |
| iddv/nasa/filterXPathTemplate | NASA notikumu filtra šablons. |
| iddv/nasa/filterNamespacePrefix | filterXPathTemplate izmantotās vārdu telpas apzīmējums (prefikss). |
| iddv/nasa/filterNamespace | filterXPathTemplate izmantotā vārdu telpa, kas tiek apzīmēta ar filterNamespacePrefix. |
| iddv/nasa/subscriptionLengthInMinutes | NASA pieraksta ilgums. Pēc noklusējuma 20 dienas. |
| iddv/nasa/subscribeToNotifications | Nosaka, vai veikt pierakstīšanos uz NASA notikumiem. Izstrādes vidē konfigurācija tiek ņemta no emulatora, nevis e-pakalpojumu kataloga. Līdz ar to pierakstam uz notifikācijām nav nozīmes.  Savukārt produkcijas vidē pieraksts uz NASA notikumiem ir nepieciešams. Pēc noklusējuma true. |

Konfigurācijas piemērs, darbam ar failu sistēmu.

<iddv

transport="fs"

transportArgs="C:\Users\egils.stamurs\Documents\Projects\VISS\_temp \Viss.Iddv.Forms\IddvFormsTransportEmulator"

transportAuthRequired="true">

<general

iddvUrn="FileSystemTransportDestination-ErrorReport"

cdnUrl="https://visssc:448/PortalEmulator/Cdn/{{version}}"

routerServiceUrl="net.pipe://localhost/Abc.Viss.Iddv.RouterService /RouterService.svc"

transportedSchemaUrl="http://ivis.eps.gov.lv/XMLSchemas/100000 /ErrorReport/v1-0/IVISErrorReport.xsd"

/>

<edk

edkServiceEndpointAddress="net.pipe://localhost/Abc.Viss.Edk. RouterService/RouterService.svc"

edkDirectoryPattern="/Ath/{0}/VISS/IDDV/Out/{1:yyyyMM}\_{1:ddHH}/{2}"

/>

<nasa

subscribeToNotifications="false"

/>

</iddv>

Konfigurācijas piemērs, darbam servisu režīmā.

<iddv

transport="service"

transportAuthRequired="true">

<general

iddvUrn="URN:IVIS:100001:EP-IVISErrorReport-v1-0-IDDVFORM2-ErrorReport"

cdnUrl="https://visssc:448/PortalEmulator/Cdn/{{version}}"

routerServiceUrl="net.pipe://localhost/Abc.Viss.Iddv.RouterService /RouterService.svc"

transportedSchemaUrl="http://ivis.eps.gov.lv/XMLSchemas/100000 /ErrorReport/v1-0/IVISErrorReport.xsd"

/>

<edk

edkServiceEndpointAddress="net.pipe://localhost/Abc.Viss.Edk. RouterService/RouterService.svc"

edkDirectoryPattern="/Ath/{0}/VISS/IDDV/Out/{1:yyyyMM}\_{1:ddHH}/{2}"

/>

<nasa

subscribeToNotifications="false"

/>

</iddv>

### Formas konfigurācijas iegūšana

Neatkarīgi no izvēlētā formas darbības režīma (failu sistēmas vai servisa) formas konfigurāciju iespējams iegūt divos veidos:

* No portāla emulatora – nepieciešams formas konfigurācijas datnē web.config izveidot attēlā redzamo konfigurācijas sekciju „2” un norādīt adresi uz portāla emulatora konfigurācijas servisu „3”.
* No e-pakalpojumu kataloga servisa – konfigurācijas izgūšana notiek no EPK servisa, ja netiek norādīta attēlā redzamā konfigurācijas sekcija „2”.



37.attēls. Formas konfigurācijas iegūšana

Formai ielādējoties tā vienmēr pieprasa konfigurācijas datus pēc kāda no izvēlētajiem veidiem.

### Autentifikācijas konfigurēšana

Izstrādājamās formas iespējams veidot tā lai lietotāja autentifikāciju varētu iespējot vai atspējot mainot vienu konfigurācijas parametru.

Autentifikācijas iespējošanas vai atspējošanas parametrs.

<iddv

transportAuthRequired="true">

/>

...

</iddv>



38.attēls. Autentifikācijas iespējošanas funkcionalitāte

Lai šī funkcionalitāte darbotos korekti ir jāveic sekojošas darbības:

* Jānokonfigurē lietojuma autentifikācijas sekciju un atbilstoši arī drošības talonu servisu (ko var nedarīt izstrādes vidē ja autentifikācija nav nepieciešama).
* Formas konfigurācijas datnei jāsatur sekojoša sekcija, kas nosaka ka lietojumā ir direktorija „SecureContent” kuras saturam piekļūt var tikai autentificēti lietotāji.

<configuration>

...

<location path="SecureContent">

<system.web>

<authorization>

<deny users="?" />

</authorization>

</system.web>

</location>

...

</configuration>

* Formas konfigurācijas datnei jāsatur sekojoša sekcija, kas nosaka ka pārējām formas datnēm var piekļūt jebkurš lietotājs.

<configuration>

...

<system.web>

<authorization>

<allow users="\*" />

</authorization>

...

</system.web>

</configuration>

* Formas lietojumā jāimplementē sekojošs koda fragments, kas pārbaudīs autentifikācijas parametru un ja nepieciešams veiks pāreju uz drošu lietojuma apgabalu, kas izsauks autentifikācijas pieprasījumu.

protected override void OnPreLoad(EventArgs e) {

if (ConfigManager.GetIddvConfigSection().TransportAuthRequired

&& (Page.User == null || !Page.User.Identity.IsAuthenticated)) {

Response.Redirect("~/SecureContent?" + Request.QueryString);

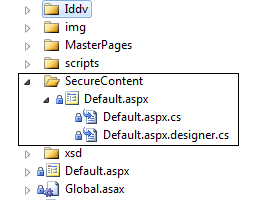
return;

}

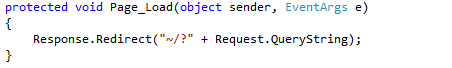
...

}

* Formas lietojumā jāizveido direktorija „SecureContent” un tajā jāizveido forma Default.aspx kuras Page\_Load metode satur pāreju uz saknes direktoriju saglabājot sākotnēji saņemtos parametrus.



.attēls. Direktorija SecureContent



.attēls. Metodes Page\_Load piemērs

Iepriekš minētie koda piemēri attiecas uz gadījumiem kad Iddv formas lietojums satur vienu ieejas formu Default.aspx, kas tiek atvērta pēc noklusējuma to specifiski nenorādot.

Formas kas izstrādātas ir iepriekšējām bibliotēku versijām pieļāva vairākas ieejas formas vienā Iddv formas lietojumā, lai šādā lietojumā implementētu šo funkcionalitāti nepieciešams modificēt koda piemērus tā, lai pēc autentifikācijas notiktu atgriešanās uz to atbilstošo ieejas formu.

## Vārdtelpas

Vārdtelpas apkopotas 6.tabulā.

6.tabula

Vārdtelpas

| Klase | Vārdu telpa |
| --- | --- |
| Task | Iddv.Forms |
| ITaskPage | Iddv.Forms |
| TasksHelper | Iddv.Forms.Helpers |
| Attachment | Iddv.Forms |
| Attachments | Iddv.Forms |