

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

----

versija

Rīgā 2013

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

Dokumenta identifikācija

|  |  |
| --- | --- |
| Dokumenta ID: | -----V- |
| Dokumenta nosaukums: | .  .  .  . |
| Dokumenta kods: | ---- |
| Versija: | Versija , Laidiens (saīsināti V ) |

Saskaņojumi

|  |  |  |  |
| --- | --- | --- | --- |
| Organizācija | Vārds, uzvārds, amats | Datums | Paraksts |
|  | Atbildīgā persona no Pasūtītāja puses |  |  |
|  | , projekta vadītājs par tehniskiem jautājumiem no Izpildītāja puses |  |  |
|  | E.Stāmurs, programmētājs |  |  |
|  | M.Pētersons, projekta vadītājs par administratīviem jautājumiem no Izpildītāja puses |  |  |
|  | E.Blumberga, projekta kvalitātes kontroles vadītāja |  |  |

Izmaiņu vēsture

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Versija | Datums | Apraksts | Organizācija | Autors |
| 1.00 | 17.07.2013. | Izveidota dokumenta sākotnējā versija | SIA „ABC software” | E.Stāmurs |
| 1.01 | 10.08.2013. | Atjaunota KDV formas konfigurācija | SIA „ABC software” | E.Stāmurs |
| 1.02 | 23.10.2013. | Atjaunots ceļvedis pēc izmaiņām KDV formas konfigurācijā. | SIA „ABC software” | E.Stāmurs |

**Satura rādītājs**

[Attēlu saraksts 6](#_Toc371584629)

[1. Ievads 7](#_Toc371584630)

[1.1. Dokumenta nolūks 7](#_Toc371584631)

[1.2. Darbības sfēra 7](#_Toc371584632)

[1.3. Termini un pieņemtie apzīmējumi 7](#_Toc371584633)

[1.4. Saistība ar citiem dokumentiem 7](#_Toc371584634)

[1.5. Dokumenta pārskats 8](#_Toc371584635)

[2. Risinājuma arhitektūra 9](#_Toc371584636)

[3. KDV formas piemērs 11](#_Toc371584637)

[3.1. Lvp.Kdv.Examples.SimpleForm.Web 11](#_Toc371584638)

[4. Izstrādes vides sagatavošana 12](#_Toc371584639)

[4.1. Izstrādes vides prasības 12](#_Toc371584640)

[4.2. Pakotnes saturs 12](#_Toc371584641)

[4.2.1. Lvp.Kdv.Template saturs 12](#_Toc371584642)

[4.2.2. Lvp.Kdv.Environment.zip pakotnes saturs 12](#_Toc371584643)

[4.2.2.1. Lvp.Eservice.Certificates apakš pakotnes saturs 12](#_Toc371584644)

[4.2.2.2. Lvp.Kdv.Examples apakš pakotnes saturs 13](#_Toc371584645)

[4.3. Portāla emulatora uzstādīšana 13](#_Toc371584646)

[4.4. Maršrutēšanas servisa uzstādīšana 13](#_Toc371584647)

[4.4.1. IIS apakš-aplikācijas izveide 13](#_Toc371584648)

[4.4.2. Sertifikātu importēšana 13](#_Toc371584649)

[4.4.3. Žurnalēšanas mapju uzstādīšana 16](#_Toc371584650)

[4.5. KDV formas piemēra uzstādīšana 17](#_Toc371584651)

[4.5.1. IIS apakš-aplikācijas izveide 17](#_Toc371584652)

[4.5.2. Sertifikātu importēšana 18](#_Toc371584653)

[4.5.3. Žurnalēšanas mapju uzstādīšana 18](#_Toc371584654)

[4.6. Jaunas KDV formas izveidošana 18](#_Toc371584655)

[4.6.1. Priekšnosacījumi 18](#_Toc371584656)

[4.6.2. Kdv formas projekta izveide 18](#_Toc371584657)

[4.6.3. KDV formas apakš-lietojuma izveide IIS 20](#_Toc371584658)

[4.6.4. KDV formas reģistrēšana emulatorā 21](#_Toc371584659)

[4.6.5. KDV formas sertifikāta reģistrēšana 21](#_Toc371584660)

[4.6.6. KDV formas īpašas piekļuves apgabala (realm) reģistrēšana 21](#_Toc371584661)

[4.6.7. Sagatavošanās lietotāja profila izmantošanai 21](#_Toc371584662)

[4.6.8. Sagatavošanās formas izmantošanai bez emulatora. 21](#_Toc371584663)

[4.6.9. KDV formas izstrāde izmantojot failu sistēmu 22](#_Toc371584664)

[5. KDV formas projekts 23](#_Toc371584665)

[5.1. Projekta struktūra 23](#_Toc371584666)

[5.2. IIS darbināta projekta atkļūdošana (debug) 24](#_Toc371584667)

[5.3. KDV formas realizēšana 25](#_Toc371584668)

[5.3.1. Klase Task<T> 25](#_Toc371584669)

[5.3.2. Klase ITaskPage<T, K> 25](#_Toc371584670)

[5.3.3. Ievades pareizības pārbaude 26](#_Toc371584671)

[5.4. Darba uzdevumu galamērķi 27](#_Toc371584672)

[6. Bibliotēkas KDV formu izstrādei 28](#_Toc371584673)

[6.1. .NET bibliotēka Lvp.Kdv.FunctionsLib 28](#_Toc371584674)

[6.2. .NET bibliotēka Lvp.Kdv.Controls 29](#_Toc371584675)

[6.3. JavaScript bibliotēkas 29](#_Toc371584676)

[7. KDV formas konfigurācija 30](#_Toc371584677)

[7.1. Lvp/general sekcija 30](#_Toc371584678)

[7.2. Kdv sekcija 30](#_Toc371584679)

[7.3. EserviceConfiguration galapunkta sekcija 31](#_Toc371584680)

[8. Portāla emulatora lietotāja ceļvedis 33](#_Toc371584681)

# Attēlu saraksts

[1.attēls. KDV formas ielāde 9](#_Toc371584611)

[2.attēls. KDV formas lietotnes integrācijas shēma 10](#_Toc371584612)

[3.attēls. Sertifikāta imports 14](#_Toc371584613)

[4.attēls. Sertifikāta importa vednis (2. solis) 14](#_Toc371584614)

[5.attēls. Sertifikāta importa vednis (3. solis) 15](#_Toc371584615)

[6.attēls. Sertifikāta importa vednis (4. solis) 15](#_Toc371584616)

[7.attēls. Sertifikāta privāto atslēgu tiesību pārvaldības atvēršana 16](#_Toc371584617)

[8.attēls. Sertifikāta privāto atslēgu tiesību pārvaldība 16](#_Toc371584618)

[9.attēls. KDV formas šablona nosaukuma aizstāšana 19](#_Toc371584619)

[10.attēls. KDV formas šablona nosaukuma aizstāšana projekta iestatījumos 19](#_Toc371584620)

[11.attēls. KDV formas GUID maiņa 20](#_Toc371584621)

[12.attēls. GUID gen rīka izmantošana 20](#_Toc371584622)

[13.attēls. Portāla kontroles panelis 22](#_Toc371584623)

[14.attēls. KDV formas konfigurācijas fragments LVP emulatorā. 22](#_Toc371584624)

[15.attēls. KDV formas projekta struktūra 23](#_Toc371584625)

[16.attēls. Pieslēgšanās dialoga atvēršana 24](#_Toc371584626)

[17.attēls. Pieslēgšanās IIS procesam 24](#_Toc371584627)

[18.attēls. Bibliotēkas KDV formu izstrādei 28](#_Toc371584628)

# Ievads

Viena no Latvija.lv portāla (turpmāk LVP) sastāvdaļām ir Klienta darba vieta (KDV), kas nodrošina darba vietu autentificētiem lietotājiem, kuri izpilda e-pakalpojumus. Klienta darba vieta nodrošina e-pakalpojumu nepabeigto darba uzdevumu atspoguļošanu formās (turpmāk KDV forma).

## Dokumenta nolūks

Dokumenta nolūks ir iepazīstināt izstrādātājus ar KDV formu veidošanas principiem, integrāciju ar LVP, piedāvātajām bibliotēkām, kā arī nodrošināt ar izstrādes uzsākšanai nepieciešamo informāciju.

Dokumentā iekļauts arī izstrādes vides apraksts un tās uzstādīšanas instrukcija.

## Darbības sfēra

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

## Termini un pieņemtie apzīmējumi

Apzīmējumu un terminu vārdnīca pieejama dokumentā [1].

Šī dokumenta ietvaros termins CDN tiek lietots, lai aprakstītu tīmekļa lietojumprogrammu, kuras vienīgās uzdevums ir nodrošināt resursus citām lietojumprogrammām (attēli, JS un CSS datnes).

## 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. „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. E-pakalpojuma izveidošanas ceļvedis Latvija.lv portālam. Programmētāja rokasgrāmata. (VRAA-6\_15\_11\_58-VISS\_2010-LVP\_WZD\_2-PR).

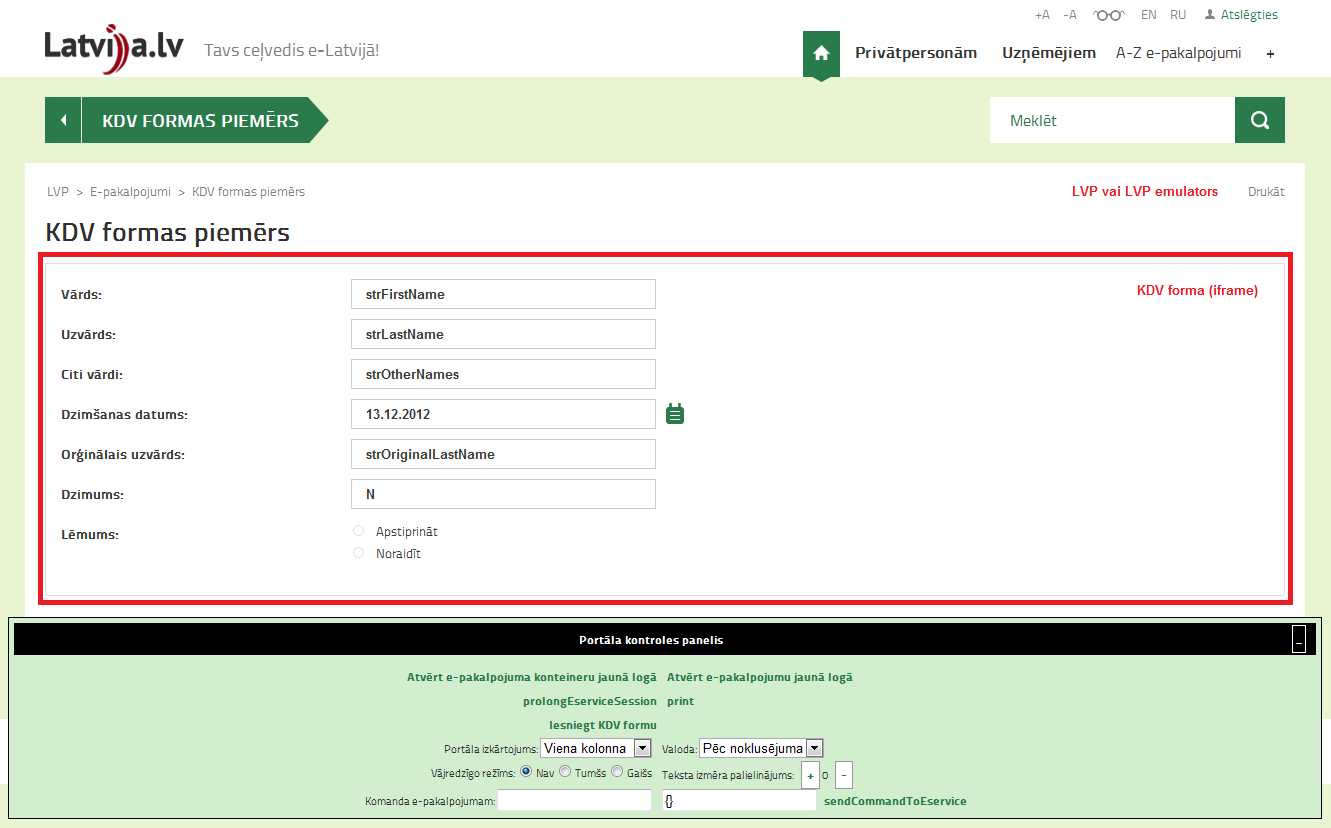
## Dokumenta pārskats

Dokumentu veido šādiem 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 e-pakalpojumu integrāciju ar LVP. Sākumpunkts izstrādātājiem, lai saprastu kopainu.
* „KDV formas piemērs” – KDV formas piemēra apraksts.
* „Izstrādes vides sagatavošana” – instrukcija KDV formas izstrādes vides sagatavošanai.
* „KDV formas projekts” – KDV formas projekta šablona apraksts.
* „Bibliotēkas KDV formu izstrādei” – izstrādātājiem piedāvāto komponenšu apraksts.
* „KDV formas konfigurācija” – KDV formas konfigurācijas sekciju apraksts.
* „Portāla emulatora lietotāja ceļvedis” – ceļvedis emulatoram, kas izstrādes vidē imitē LVP darbību.

# Risinājuma arhitektūra

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



.attēls. KDV formas ielāde

KDV formu izstrādātājiem tiek piedāvāts LVP emulators, kas KDV formai nodrošina tādu pat interfeisu, kāds tiks izmantots saziņai ar īsto LVP.

KDV formas lietotnes infrastruktūra ir redzama 2.attēlā.

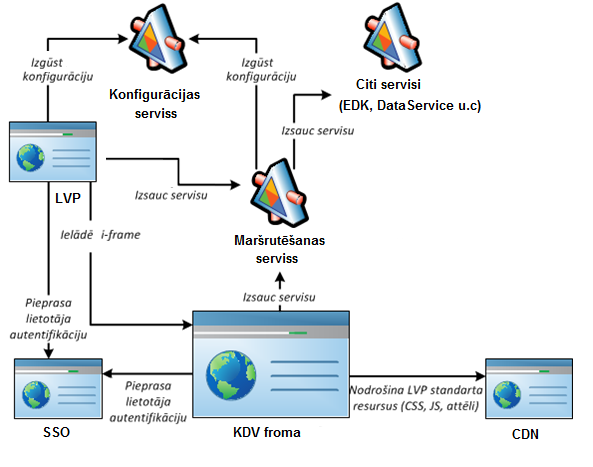
Autentifikācijai tiek izmantots SSO. Ja lietotājs ir ielogojies LVP, atverot KDV formu, notiek automātiska autentifikācija arī KDV formas aplikācijā.

Savukārt, ja lietotājs nav ielogojies LVP, pie KDV formas atvēršanas, notiek novirzīšana (*redirect*) uz SSO autentifikācijas lapu.

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

Saziņu ar servisiem KDV forma veic caur maršrutēšanas servisu. Maršrutēšanas serviss saņem ziņojumu un atkarībā no izsauktā maršrutēšanas servisa galapunkta (*endpoint*) nosūta to tālāk uz atbilstošo servisu. Nepieciešamības gadījumā tiek veikta lietotāja drošības talona apmaiņa pret servisa izsaukšanai nepieciešamo talonu. Tā kā vienu maršrutēšanas servisu var izmantot vairākas formas, tiek samazināta servisu galapunktu konfigurācijas dublēšanās (galapunktu konfigurācija glabājas maršrutēšanas servisā, nevis katras KDV formas konfigurācijā), kā arī tiek uzlabota lietotāja drošības talonu kešdarbe (*caching*).

Saziņa ar integrācijas servisiem notiek caur pieprasījumu servisu (turpmāk PS). Pieprasījumu serviss saņem ziņojumu un pārsūta uz norādītajam galamērķim atbilstošo integrācijas servisu. PS pārbauda ziņojuma struktūras atbilstību pret XSD shēmu, kā arī veic žurnalēšanu.



2.attēls. KDV formas lietotnes integrācijas shēma

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

* Konteineru, kurā tiek ielādēta KDV forma.
* Portāla standarta resursus (CSS, JS, attēli).
* Konfigurācijas servisu.

Lietas, kuras netiek emulētas un kuru darbībai nepieciešama atbilstoša e-pakalpojumu vides uzturētāja (turpmāk EPU) testa vides konfigurācija:

* Lietotāja autentifikācijas mehānisms (SSO).
* Citi servisi (EDK, lietotāja profils, e-parakstīšana utt.).

Savas KDV formas izstrādātājs var darbināt, izveidojot IIS aplikāciju un piereģistrējot tās emulatorā. Darbināšana *VS Development Server* nav iespējama, jo e-pakalpojumus nepieciešams izsaukt caur TLS (https) protokolu, lai nodrošinātu SSO drošību. *VS Development Server* neatbalsta TLS.

Veicot autentificētus izsaukumus caur maršrutēšanas servisu, tam ir nepieciešams sertifikāts, lai autentificētos STS.

Uzsākot KDV formas izstrādi ar pasūtītāju, ir jāvienojas par formas iestatījumiem un atbalstāmo LVP funkcionalitāti:

* **Mērogojams interfeiss**: Vai KDV forma atbalstīs izmēra maiņu atbilstoši pārlūka loga platumam (tiek piedāvāti pamata stili šādai funkcionalitātei, tomēr visticamāk būs jāveic arī formas specifiski pielāgojumi).
* **Portāla izkārtojums**: Kāds būs KDV formas (iframe) platums (iespējamas četras vērtības) un kādi LVP komponenti tiks rādīti blakus.
* **Vājredzīgo režīms**: Vai KDV forma nodrošinās vājredzīgo režīmiem atbilstošu attēlojumu (tiek piedāvāti pamata stili šādai funkcionalitātei, tomēr visticamāk būs jāveic arī formas specifiski pielāgojumi).
* **Teksta palielināšana/samazināšana**: Vai KD forma ļaus lietotājam palielināt un samazināt tā teksta izmēru (šim nolūkam tiek piedāvāta JS bibliotēka – skat. 6.3.sadaļu).

Lai labāk saprastu šo parametru nozīmi, ieteicams apskatīt 8.nodaļu.

# KDV formas piemērs

Lai atvieglotu KDV formas izstrādes uzsākšanu un ļautu vieglāk saprast darbības mehānismus, ir pieejams formas piemērs. Izstrādātāji var apskatīt to kodu un uzstādīt to lokālajā izstrādes vidē.

Tā kā KDV formu un e-pakalpojumu izstrādē tiek izmantots kopīgs emulators, emulatorā pēc noklusējuma reģistrēti vairāki e-pakalpojumu piemēri. Ja piemērus nav paredzēts izmantot, no emulatora var dzēst tiem atbilstošo konfigurāciju (skat. 8.nodaļu).

Lai izmantotu KDV formas piemēru, to nepieciešams reģistrēt LVP emulatorā.

## Lvp.Kdv.Examples.SimpleForm.Web

Šis piemērs nodrošina e-pakalpojuma specifiska darba uzdevuma atspoguļošanu, modificēšanu, ievaddatu validāciju un saglabāšanu. Forma atspoguļo *PersonBasicInfoStructure* struktūras darba uzdevumus.

# Izstrādes vides sagatavošana

## Izstrādes vides prasības

* Operētājsistēma: *Windows;*
* Uzinstalēta *.Net Framework* 4.0 komponente;
* Uzinstalēta IIS 7.0 komponente;
* Uzinstalēta *Visual Studio* 2010 vai *Visual Studio* 2012;

## Pakotnes saturs

Pakotne satur KDV formu izstrādes videi nepieciešamos lietojumus, formu piemērus un KDV formas šablonu (būtībā komponentes, ko apraksta šis dokuments). Pakotne jāuzstāda uz KDV formu izstrādātāju datoriem. Pakotņu atrašanās vieta (mape) ir brīvi izvēlēta un pēc uzstādīšanas pabeigšanas šīs pakotnes var dzēst.

Pakotnes datnes:

* **Lvp.Kdv.Template** – KDV formas sagatave (jaunas formas veidošanai).
* **Lvp.Kdv.Environment** – izstrādes videi nepieciešamās komponentes.

### Lvp.Kdv.Template saturs

* **Lvp.Kdv.FormTemplate.Web** – KDV formas sagataves projekts, kuru kopējot un pielāgojot paredzēts veidot jaunās KDV formas (skat. 4.6.2.nodaļu).
* **Reference Assemblies** – mape, kurā paredzēts glabāt bibliotēkas, kas nepieciešamas visām izstrādājamajām KDV formām un kuras projektā tiek iekļautas kā bibliotēkas (dll), nevis kā VS projekti.

### Lvp.Kdv.Environment.zip pakotnes saturs

Pakotnē atrodamas šādas mapes:

* **Lvp.Kdv.Certificates** – sertifikāti.
* **Lvp.Eservice.PortalEmulator** – LVP emulators (izstrādes videi).
* **Lvp.Kdv.RouterService** – Kdv formu maršrutēšanas serviss.
* **Lvp.Kdv.Examples** – Kdv formu piemēri.
* **KdvFormsTransportEmulator**– Failu sistēmas hierarhijas piemērs.

#### Lvp.Eservice.Certificates apakš pakotnes saturs

* **eservices.pfx** – e-pakalpojumu un KDV formu SSL/TLS sertifikāts (http**s** nodrošināšanai izstrādes vidē).
* **eservices\_Root\_CA.cer** – e-pakalpojumu SSL/TLS sertifikāta CA sertifikāts.
* **VISS\_Root\_CA.cer** – VISS CA sertifikāts.
* **Abc.Kdv.Router.cer** – maršrutēšanas servisa sertifikāts, lai autentificētos STS servisā.

#### Lvp.Kdv.Examples apakš pakotnes saturs

* **Lvp.Kdv.Examples.SimpleForm.Web** – KDV formas piemērs (skat. 3.1.nodaļu).
* **Reference Assemblies** – mape, kurā tiek glabātas bibliotēkas, kas nepieciešamas visām izstrādājamajām KDV formām un kuras projektā tiek iekļautas kā bibliotēkas (dll), nevis kā VS projekti.
* **BasicCorrectnessRules-custom.ruleset** – piemēru koda kvalitātes pārbaudes iestatījumi.
* **CustomDictionary.xml** – piemēru koda kvalitātes pārbaudes iestatījumi.

## Portāla emulatora uzstādīšana

Portāla emulatora uzstādīšanas instrukciju skatīt dokumenta [2] nodaļā *4.4.1. Portāla emulatora uzstādīšana*.

## Maršrutēšanas servisa uzstādīšana

Maršrutēšanas serviss atrodams *Lvp.Kdv.Environment.zip* pakotnes *Lvp.Kdv.RouterService* mapē. Ir paredzēts, ka vienu servisa eksemplāru izmantos vairākas (visas) izstrādājamās KDV formas.

Izstrādes vidē maršrutēšanas servisu var uzstādīt lokāli katram izstrādātājam vai arī starp vairākiem izstrādātājiem izmantot vienu eksemplāru uz atsevišķa servera.

### IIS apakš-aplikācijas izveide

Priekšnosacījumi: izpildītas dokumenta [2] sadaļu *4.3.2. Saknes sertifikātu imports* un *4.3.3. E-pakalpojumu IIS lietojuma (application) izveide* instrukcijas.

Jāpievieno jauna apakš-aplikācija **eservices** lietojumam (skat. dokumenta [2] sadaļu *4.3.4. Portāla emulatora apakš-aplikācijas izveidošana*) ar šādiem parametriem:

* **Alias**: Lvp.Kdv.RouterService;
* **Application pool**: noklusētā vērtība (eservices lietojuma pūls);
* **Physical path**: ceļš uz atarhivēto mapi.

Pūla lietotājam jāpiešķir tiesības apakš-aplikācijas mapē.

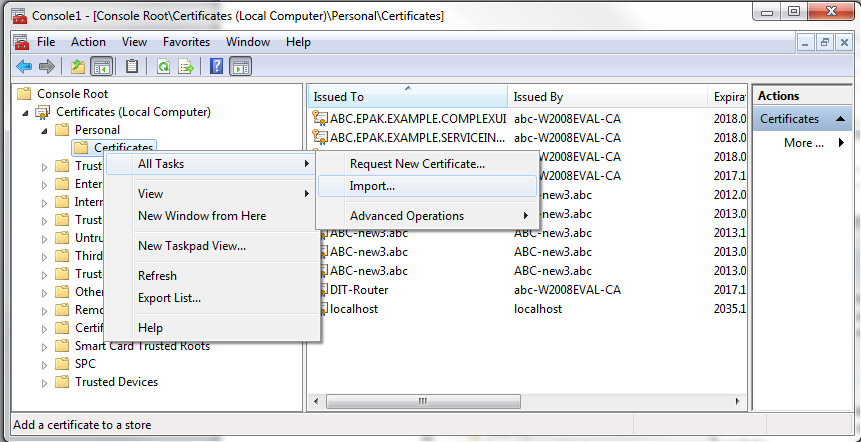
Lietojumam **eservices** pievienot bindings informāciju net.pipe protokolam norādot Bindings information vērtību „\*”.

Apakš-aplikācijai Lvp.Kdv.RouterService pievienot net.pipe protokolu atdalot to ar komatu: „http,net.pipe”.

### Sertifikātu importēšana

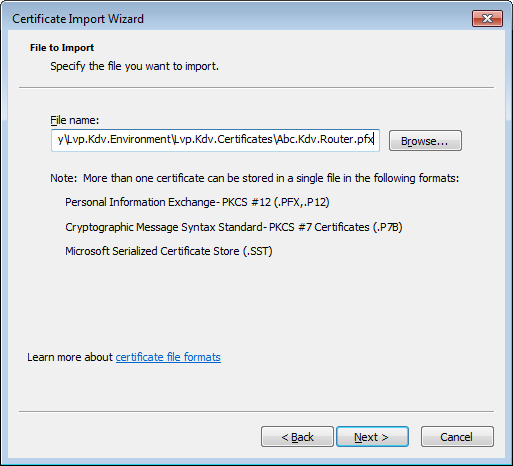
Lai no KDV formām varētu izsaukt drošos servisus (DataService), ir jāreģistrē *Abc.Kdv.Router* sertifikāts, kas atrodams sertifikātu arhīvā.

1. Jāatver sertifikātu pārvaldības rīks (skat. dokumenta [2] sadaļu *4.3.1.* *Sertifikātu pārvaldības rīka atvēršana*).
2. *All tasks -> Import* (no *Personal* mapes).



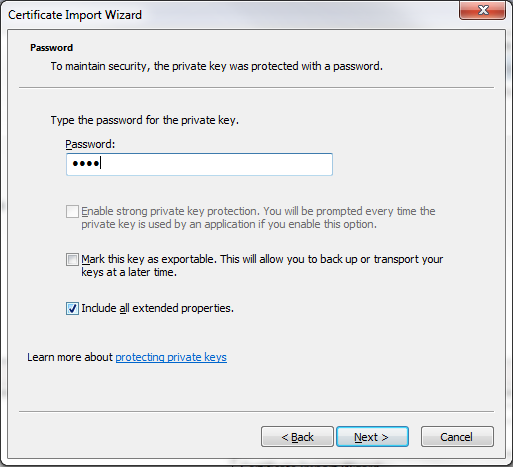
.attēls. Sertifikāta imports

1. Jāielādē sertifikāta fails *Abc.Kdv.Router.pfx*.



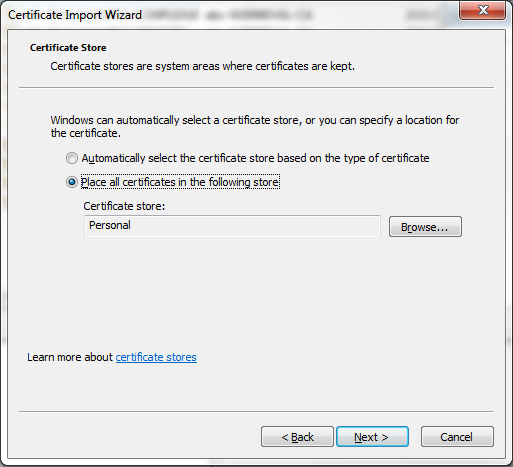
.attēls. Sertifikāta importa vednis (2. solis)

1. Jāievada sertifikāta parole (1234).



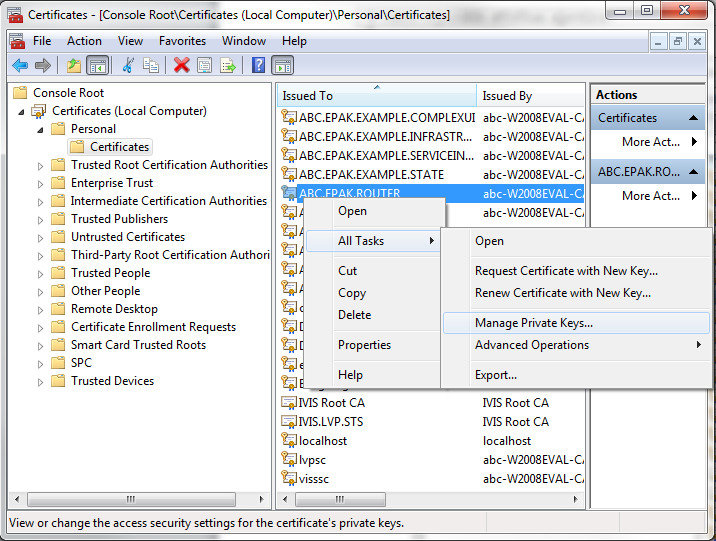
.attēls. Sertifikāta importa vednis (3. solis)

1. Jāpabeidz vednis.

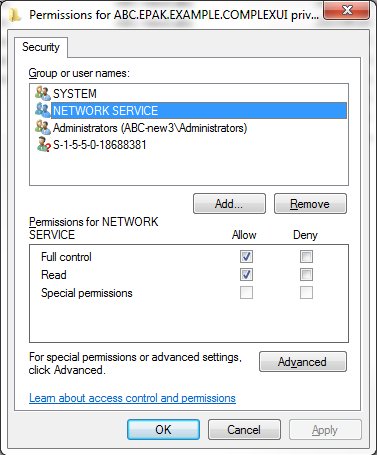


.attēls. Sertifikāta importa vednis (4. solis)

1. Jāpārliecinās, ka 4.3.sadaļā izveidotās tīmekļa aplikācijas pūla lietotājam ir tiesības lasīt sertifikāta privāto atslēgu (*All Tasks -> Manage Private Keys*).



.attēls. Sertifikāta privāto atslēgu tiesību pārvaldības atvēršana



.attēls. Sertifikāta privāto atslēgu tiesību pārvaldība

### Žurnalēšanas mapju uzstādīšana

Nepieciešams izveidot mapi žurnalēšanas datnēm, un iepriekšējos soļos izveidotā pūla lietotājam jāpiešķir tiesības rakstīt šajā mapē. Pretējā gadījumā žurnalēšanas informācija vienkārši netiks saglabāta.

Pēc noklusējuma mapes ceļš ir „C:\Logs\Lvp.Kdv.Router\”, bet to var mainīt konfigurācijas datnē (web.config):

<source name="Microsoft.IdentityModel" switchValue="Warning">

<listeners>

<add name="IdentityModelTraceListener"

initializeData="c:\Logs\Lvp.Kdv.Router\WIFTrace.svclog"

type="System.Diagnostics.XmlWriterTraceListener"

traceOutputOptions="Timestamp"/>

</listeners>

</source>

<source name="System.ServiceModel.MessageLogging"

switchValue="Warning, ActivityTracing">

<listeners>

<add name="ServiceModelMessageLoggingListener"

initializeData="c:\Logs\Lvp.Kdv.Router\web\_messages.svclog"

type="System.Diagnostics.XmlWriterTraceListener"

traceOutputOptions="Timestamp"/>

</listeners>

</source>

<source name="System.ServiceModel"

switchValue="Warning, ActivityTracing" propagateActivity="true">

<listeners>

<add name="ServiceModelTraceListener"

initializeData="c:\Logs\Lvp.Kdv.Router\web\_tracelog.svclog"

type="System.Diagnostics.XmlWriterTraceListener"

traceOutputOptions="Timestamp"/>

</listeners>

</source>

<source name="System.ServiceModel.Routing" switchValue="Warning">

<listeners>

<add name="ServiceModelTraceListener"

initializeData="c:\Logs\Lvp.Kdv.Router\web\_routerlog.svclog"

type="System.Diagnostics.XmlWriterTraceListener"

traceOutputOptions="Timestamp"/>

</listeners>

</source>

## KDV formas piemēra uzstādīšana

KDV formas piemēri atrodami *Lvp.Kdv.Environment.zip* pakotnes *Lvp.Kdv.Examples* mapē. Pakotne jāatarhivē un apakšnodaļās aprakstītās darbības jāveic katram piemēram (apakšmape ar nosaukumu, kas beidzas ar „.Web”).

### IIS apakš-aplikācijas izveide

Priekšnosacījumi: izpildītas 4.3 sadaļas instrukcijas.

Jāpievieno jauna apakš-aplikācija **eservices** lietojumam (skat. dokumenta [2] sadaļu *4.3.4. Portāla emulatora apakš-aplikācijas izveidošana*) ar šādiem parametriem:

* **Alias**: Piemēra nosaukums (piemēram, Example.SimpleForm);
* **Application pool**: noklusētā vērtība (eservices lietojuma pūls);
* **Physical path**: ceļš uz piemēra mapi.

Jāatceras arī pūla lietotājam piešķirt tiesības piemēru mapēs.

### Sertifikātu importēšana

Jāveic sertifikātu uzstādīšana un privātās atslēgas pieejas tiesību piešķiršana, kā tas ir aprakstīts. 4.4.2.sadaļā. Sertifikāti atrodami sertifikātu arhīvā (skat. 4.2.2.1.sadaļu) un to nosaukums atbilst KDV formas piemēra projekta nosaukumam (piemēram, *Abc.Kdv.Examples.SimpleForm.pfx*).

### Žurnalēšanas mapju uzstādīšana

Nepieciešams izveidot mapi žurnalēšanas datnēm un iepriekšējos soļos izveidotā pūla lietotājam jāpiešķir tiesības rakstīt šajā mapē. Pretējā gadījumā žurnalēšanas informācija vienkārši netiks saglabāta.

Pēc noklusējuma mapes ceļš ir „C:\Logs\Lvp.Kdv.Forms\”, bet to var mainīt konfigurācijas datnē (web.config):

<add name="MessageListener"

type="IVIS.Diagnostics.LogFlatFileTraceListener, IVIS.Diagnostics"

initializeData="**c:\Logs\Lvp.Kdv.Forms\**Example.SimpleForm.txt" />

<add name="XMLTraceListener

initializeData="**c:\Logs\Lvp.Kdv.Forms\**Example.SimpleForm.Messages.svclog"

type="System.Diagnostics.XmlWriterTraceListener"

traceOutputOptions="Timestamp"/>

## Jaunas KDV formas izveidošana

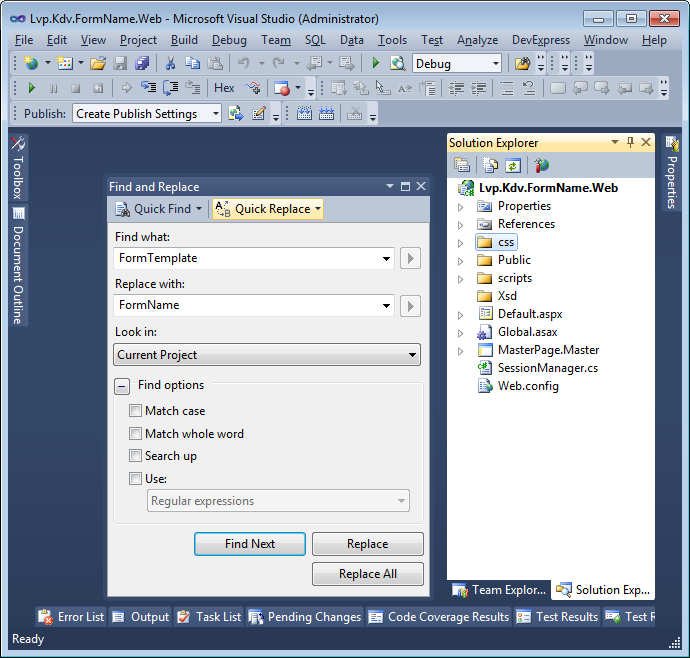
### Priekšnosacījumi

Lai e-pakalpojuma darba uzdevumu formu varētu veiksmīgi integrēt ar ārpus izstrādātāja datora esošo infrastruktūru (SSO, STS utt.), nepieciešams veikt šādas darbības:

* Reģistrēt jauno KDV formu e-pakalpojumu reģistrā un iegūt KDV formas URN.
* Reģistrēt KDV formas sertifikātu Valsts informācijas sistēmu savienotāja (turpmāk VISS) – iesniegums EPU. Pašu sertifikātu var vai nu pieprasīt no EPU vai arī uzģenerēt (pieņemot, ka izstrādātājiem ir pieejams savs sertifikātu izsniedzējs (CA)). Produkcijas videi sertifikātu vienmēr ģenerēs EPU. Papildu informācijai skatīt 4.6.5.sadaļu.
* Reģistrēt KDV formas *realm* STS (iesniegums EPU). Papildu informācijai skatīt 4.6.6.sadaļu.

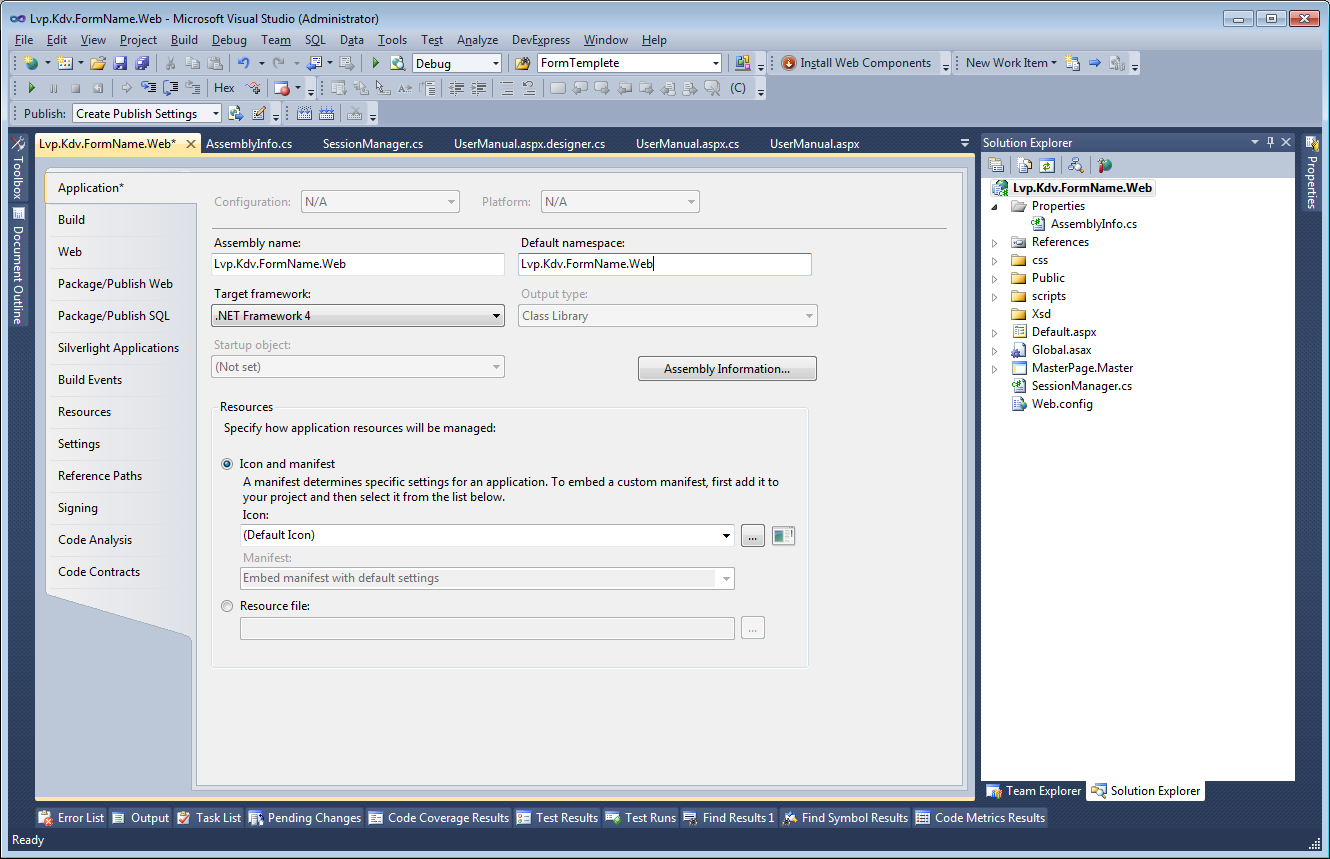
### Kdv formas projekta izveide

1. Jāatarhivē *Lvp.Kdv.FormTemplate.Web.zip*.
2. Jāpārsauc projekta fails (*Lvp.Kdv.FormTemplate.Web.csproj*) un mape, kurā tas atrodas, aizstājot *FormTemplate* ar KDV formas nosaukumu, ko izsniedzis EPU.
3. Ar *Visual Studio* jāatver iepriekš minētais projekts.
4. Projektā jāveic globāla teksta aizstāšana (*replace*), *FormTemplate* vietā liekot KDV formas nosaukumu.



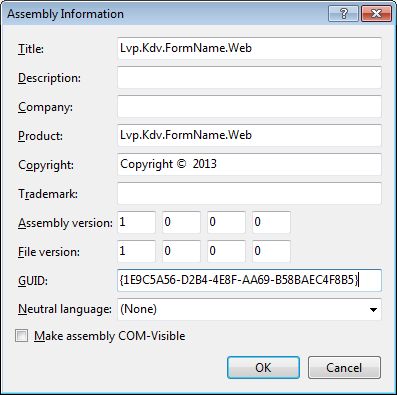
.attēls. KDV formas šablona nosaukuma aizstāšana

1. Jāveic izmaiņas projekta uzstādījumos, ievietojot KDV formas nosaukumu *Assembly Name* un *Default Namespace* laukos.



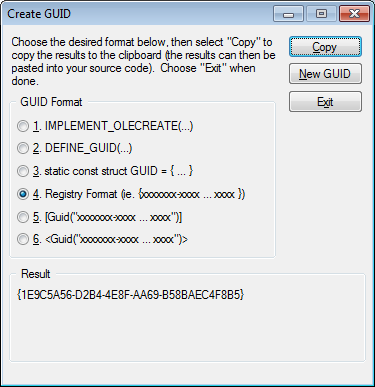
.attēls. KDV formas šablona nosaukuma aizstāšana projekta iestatījumos

1. Jāuzģenerē un jānomaina projekta identifikators (*GUID*).



.attēls. KDV formas GUID maiņa

Jauno identifikatoru var ģenerēt ar VS *GUID gen* rīku.



.attēls. GUID gen rīka izmantošana

### KDV formas apakš-lietojuma izveide IIS

Priekšnosacījumi: izpildītas dokumenta [2] sadaļu *4.3.2. Saknes sertifikātu imports* un *4.3.3. E-pakalpojumu IIS lietojuma (application) izveide* instrukcijas.

Jāpievieno jauna apakš-aplikācija **eservices** lietojumam (skat. dokumenta [2] sadaļu *4.3.4. Portāla emulatora apakš-aplikācijas izveidošana*) ar šādiem parametriem:

* **Alias**: apakš-adrese, caur kuru KDV forma būs pieejama, piemēram KDV formas nosaukums;
* **Application pool**: noklusētā vērtība (eservices lietojuma pūls);
* **Physical path**: ceļš uz KDV formas projektu.

### KDV formas reģistrēšana emulatorā

Jauno KDV formu nepieciešams reģistrēt emulatorā (skat. dokumenta [2] nodaļu *8.1.*).

### KDV formas sertifikāta reģistrēšana

Ne katrai nepubliskajai KDV formai nepieciešams ģenerēt sertifikātu. Gadījumā ja šī forma izsauc drošos servisus tikai caur maršrutēšanas servisu, tad autentifikācija tiks veikta, izmantojot maršrutēšanas servisa sertifikātu, un KDV formai to nav nepieciešams ģenerēt.

Ja sertifikāts tiek ģenerēts, šī sertifikāta publiskajai atslēgai jābūt reģistrētai arī VISS (jāsazinās ar EPU). Jāveic sertifikātu uzstādīšana un privātās atslēgas pieejas tiesību piešķiršana, kā tas ir aprakstīts 4.4.2. sadaļā.

### KDV formas īpašas piekļuves apgabala (realm) reģistrēšana

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

EPU reģistrēto apgabalu nepieciešams piefiksēt KDV fromas konfigurācijas failā (*web.config*):

<microsoft.identityModel>

<service saveBootstrapTokens="true">

…

<audienceUris>

<add value="https://ivis.eps.gov.lv/Kdv/FormName/1.0"/>

</audienceUris>

…

<federatedAuthentication>

<wsFederation

issuer="https://epakvisstv.vraa.gov.lv/STS/VISS.LVP.STS/Default.aspx"

realm=" https://ivis.eps.gov.lv/Kdv/FormName/1.0"/>

</federatedAuthentication>

…

</service>

</microsoft.identityModel>

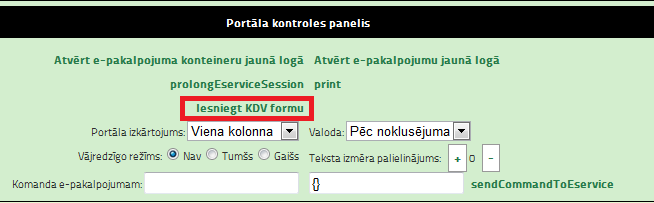
### Sagatavošanās lietotāja profila izmantošanai

Šīs nodaļas izpilde nepieciešama tikai, ja KDV formā nepieciešams izmantot lietotāja vai lietotāja pārstāvētā uzņēmuma profila informāciju (skat. 6.1Error! Reference source not found. sadaļu).

KDV formas projektam nepieciešams piesaistīt papildu bibliotēkas (Add Reference) *Abc.UserProfile.Base.dll, Abc.UserProfile.Client.dll*, kuras atrodamas Reference Assemblies mapē.

### Sagatavošanās formas izmantošanai bez emulatora.

Formas šablons satur izpildīšanas un atcelšanas pogas, pārejot uz produkcijas vidi, tās nepieciešams paslēpt, jo to notikumi tiks izsaukti izmantojot JS no ārējās aplikācijas. Izstrādājot formu caur emulatora vecākām versijām šīs pogas nepieciešams parādīt, lai varētu testēt formas funkcionalitāti. Jaunākajās versijās formas izpildi ir iespējams veikt no „portāla kontroles paneļa”, līdz ar to pogas vienmēr paliek paslēptas.

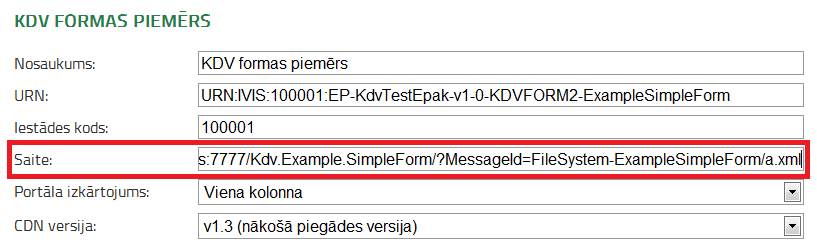


.attēls. Portāla kontroles panelis

### KDV formas izstrāde izmantojot failu sistēmu

Lai izstrādes vidē palaistu KDV formu izmantojot LVP emulatoru un lokālo failu sistēmu nepieciešams veikt sekojošas darbības:

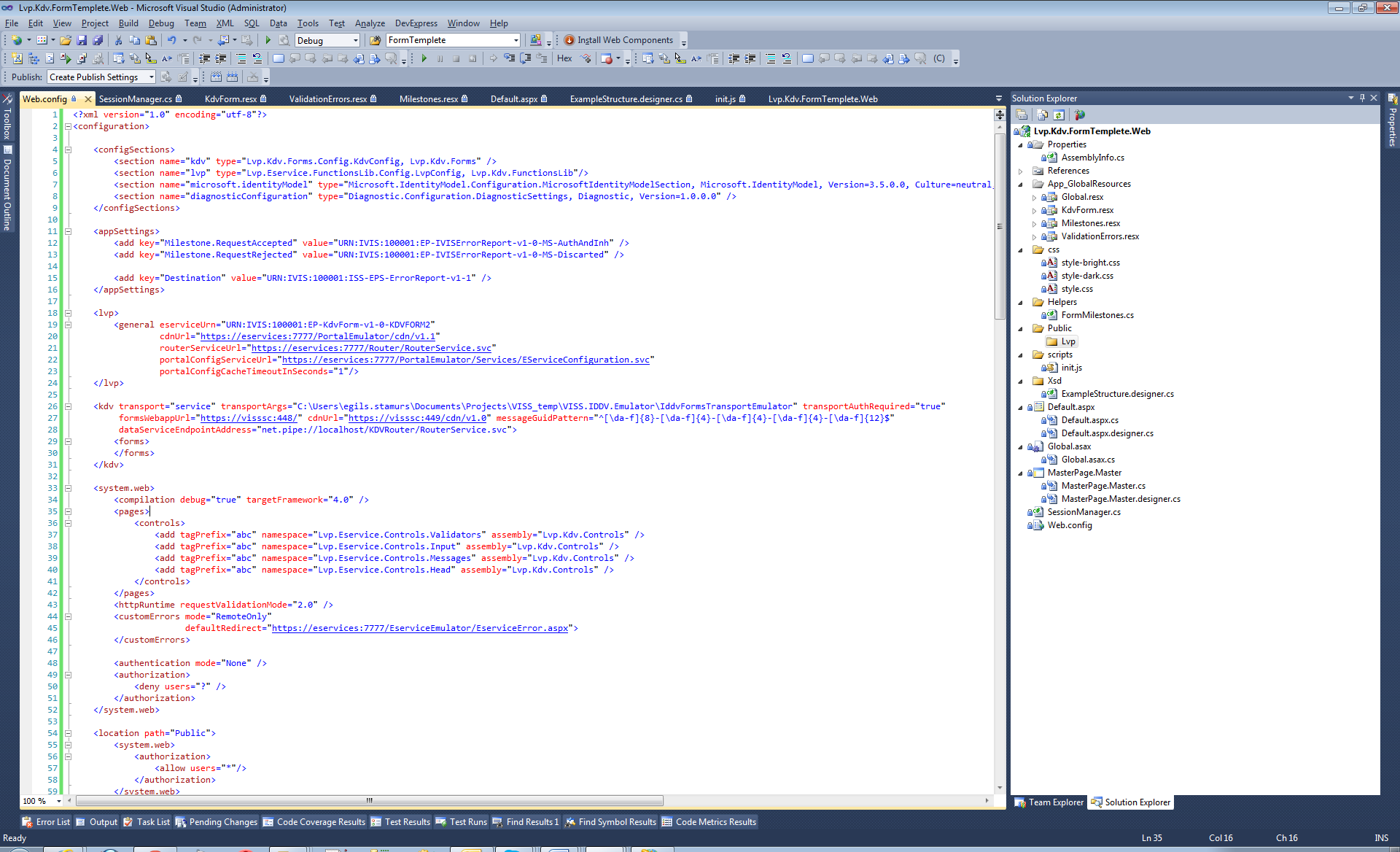
1. Izveidot datņu hierarhiju – formas ievaddatu xml (7.2);
2. Veikt formas konfigurēšanu:
   1. modificēt kdv sekciju (7.2);
   2. konfigurēt formu darbam ar emulatoru (7.3);
   3. konfigurācijas parametram lvp/general/eserviceUrn jāsakrīt ar emulatorā reģistrēto URN.
3. Reģistrēt formu LVP emulatorā (papildus info. skatīt [2] dokumentā);
   1. Failu sistēmas režīmā, jāizmaina formas saites parametra *MessageId* vērtību, tajā norādot ceļu līdz ievaddatu xml iepriekš izveidotajā datņu hierarhijā. Izmantojot servisu šis parametrs satur reāla darba uzdevuma identifikatoru (*guid*).



.attēls. KDV formas konfigurācijas fragments LVP emulatorā.

# KDV formas projekts

## Projekta struktūra

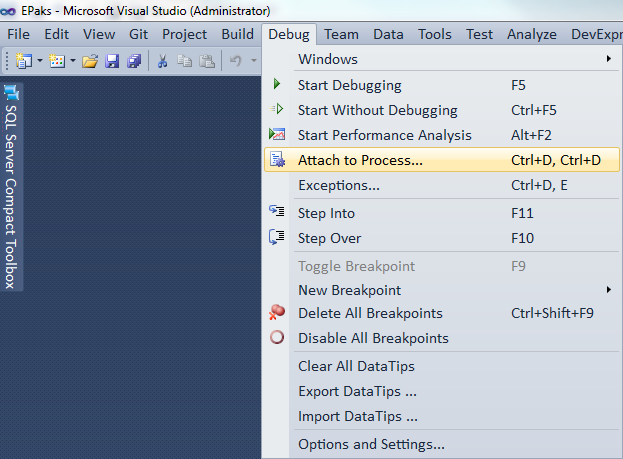


.attēls. KDV formas projekta struktūra

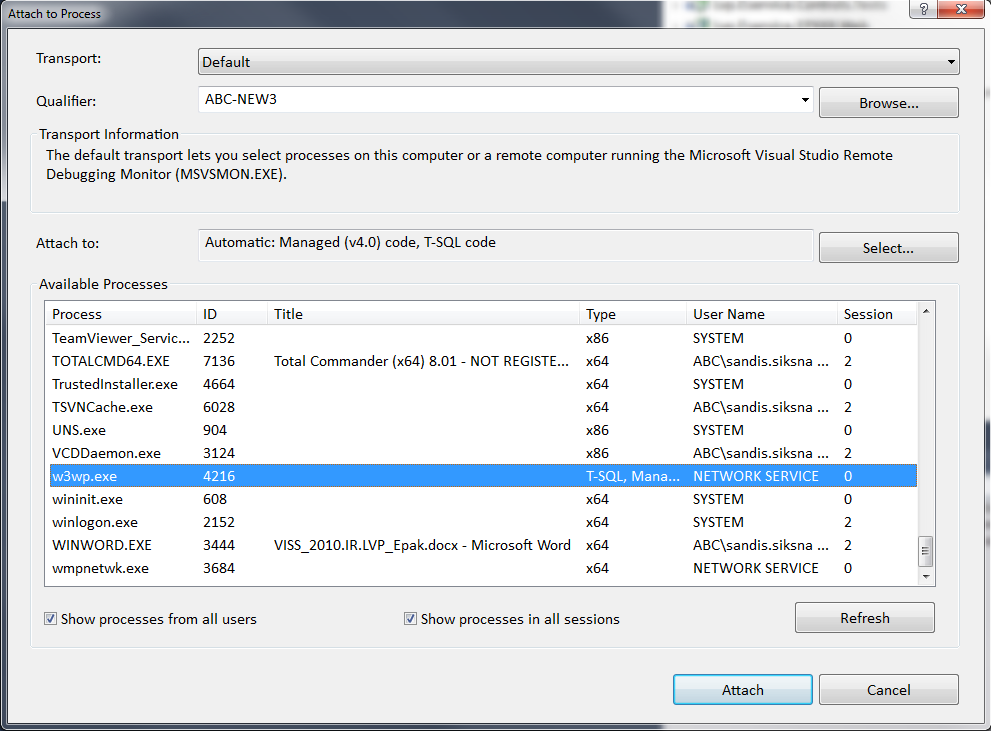
* **App\_GlobalResources**: vieta resursu datnēm, saskarnes lokalizēšanai.
* **css\style.css**: vieta CSS stiliem.
* **FormMilestones.cs**: KDV formas pieturas punkti.
* **Public**: mape, kurai piekļuve vienmēr būs atļauta arī neautentificētiem lietotājiem neatkarīgi no KDV formai nepieciešamā autentifikācijas līmeņa.
* **Public\Lvp**: mape, kurā tiek reģistrēti pieprasījumu apdarinātājai. Mapei jābūt tukšai.
* **scripts\init.js**: vieta KDV formas JS skriptiem. Šablonā skripts ir aizpildīts ar pamata darbībām LVP JS bibliotēku inicializācijai.
* **Xsd**: Mape, kurā ievietot darba uzdevumam atbilstošo XML struktūru.
* **Xsd\ExampleStructure.design.cs**: Darba uzdevuma struktūras piemērs.
* **Default.aspx**: KDV formas lietojuma galvenā lapa. Lapas nosaukums jāatstāj nemainīgs, lai tā tiktu atvērta, pieprasot KDV formas saknes URL. Lapas nosaukumu vai atrašanās vietu mainīt nedrīkst.
* **Global.asax**: Lietojuma globālo notikumu apstrādātājs.
* **MasterPage.Master**: KDV formas šablons, kuru izmanto lietojuma galvenā lapa.
* **SessionManager.cs**: Klase, kura jāizmanto piekļuvei pie ASP.NET sesijas.
* **Web.config**: Lietojuma konfigurācijas datne.

## IIS darbināta projekta atkļūdošana (debug)

IIS darbinātu tīmekļa aplikāciju var atkļūdot, pieslēdzoties *w3wp.exe* procesam:



.attēls. Pieslēgšanās dialoga atvēršana



.attēls. Pieslēgšanās IIS procesam

## KDV formas realizēšana

Lai realizētu KDV 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 KDV formu, jārealizē 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 pirms saglabāšanas 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

}

}

### Ievades pareizības pārbaude

Formas ievades pareizības pārbaudei jānotiek gan klienta pusē — gan servera pusē. Klienta puses pareizības pārbaude realizējama ar nodaļā 6.2. minētajām vadīklām.

Lai atvieglotu kļūdu paziņojumu parādīšanu, *ITaskPage* <T, K> šim nolūkam ir implementētas palīgmetodes:

| Nosaukums | Apraksts |
| --- | --- |
| *void AddErrorMessage(string error)* | Parāda lietotājam specifisku kļūdas paziņojumu |
| *void AddErrorMessage(Exception e, string error = null, bool hideExceptionDetails = false)* | Parāda lietotājam kļūdas informāciju |

## 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.RequestAccepted" value="URN:IVIS:100001:EP-IVISErrorReport-v1-0-MS-AuthAndInh" />

<add key="Milestone.RequestRejected" 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 FormMilestone {

/// <summary>

/// Request accepted

/// </summary>

[Description("Milestone.RequestAccepted")]

Accepted,

/// <summary>

/// Milestone accepted with notification

/// </summary>

[Description("Milestone.RequestAcceptedWithNotfication")]

AcceptedWithNotification,

/// <summary>

/// Request rejected

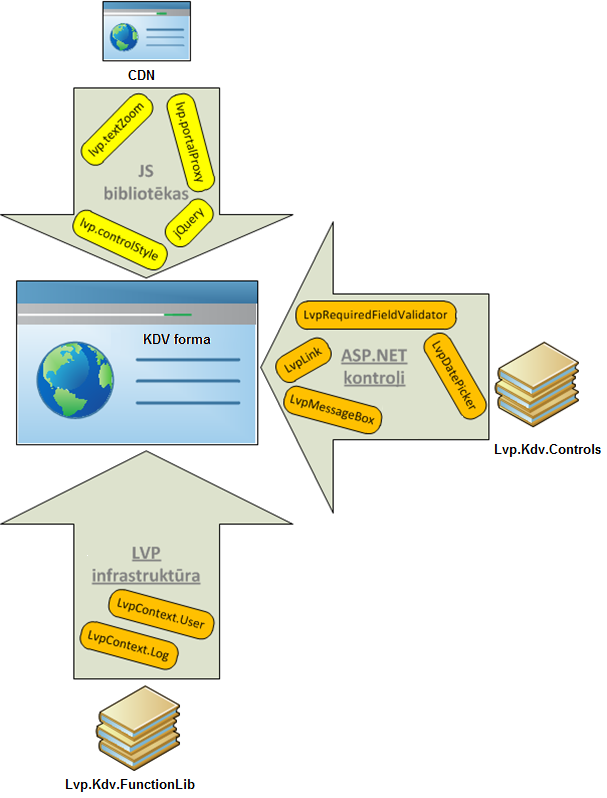
/// </summary>

[Description("Milestone.RequestRejected")]

Rejected

}

# Bibliotēkas KDV formu izstrādei



.attēls. Bibliotēkas KDV formu izstrādei

## .NET bibliotēka Lvp.Kdv.FunctionsLib

Pamata bibliotēkas funkcionalitāte ir pieeja caur *KdvContext* klases statiskajām metodēm un atvasināta no *Lvp.Eservice.FunctionLib* bibliotēkas. *KdvContext* klase tiek inicializēta aplikācijas darbības uzsākšanas sākumā (*Application\_Start*). Iekļautas sekojošas metodes:

1. *KdvContext.Config;*
2. *KdvContext.Log;*
3. *KdvContext.User;*
4. *KdvContext.UserProfile*;
5. Sesijas izmantošana.

Detalizētu informāciju par iekļautajām metodēm skatīt [2] dokumentā.

## .NET bibliotēka Lvp.Kdv.Controls

Bibliotēka nodrošina ASP.NET vadīklas un citus elementus lietotāja interfeisa standartizēšanai un izstrādes atvieglošanai. Bibliotēka atvasināta no *Lvp.Eservice.Controls* bibliotēkas.

Visām šīs bibliotēkas vadīklām nosaukums sākas ar „Lvp”. Pēc noklusējuma vadīklas ir reģistrētas e-pakalpojuma lietojumā ar birkas prefiksu (tagPrefix) „abc”.

Lielākā daļa vadīklu ir standarta ASP.NET vadīklu atvasinājumi, kas pielāgoti LVP dizainam un lietojumam. Pirms lietot kādu ASP.NET kontroli var pameklēt, vai šī bibliotēka nepiedāvā pielāgotu tā versiju (ar LVP prefiksu).

Piemēram, *RequiredFieldValidator* vietā ieteicams lietot *LvpRequiredFieldValidator*. Savukārt *LvpTextBox* vadīkla neeksistē, tādēļ jālieto *TextBox*.

Bibliotēka iekļauj sekojošas vadīklas:

1. *LvpDatePicker;*
2. *LvpDateTimePicker*
3. *LvpDatePickerValidator*
4. *Validatori*
5. *LvpDisplayState*
6. *LvpMessageBox*
7. *LvpTooltip*
8. *LvpLink*
9. *LvpResolutionLink*
10. *LvpScript*
11. *LvpWizardPageBase*
12. *LvpUserManualPageBase*
13. *LvpMasterPageBase*
14. *SessionKeepAlive*
15. *WrapperPage*

Detalizētu informāciju par iekļautajām vadīklām skatīt [2] dokumentā.

## JavaScript bibliotēkas

Detalizētu informāciju par iekļautajām JS bibliotēkām skatīt [2] dokumentā.

# KDV formas konfigurācija

## Lvp/general sekcija

Konfigurācijas sekcijā *lvp/general* tiek norādīti KDV formas pamata iestatījumi (skat 6.tabulu).

<lvp>

<general eserviceUrn="URN:IVIS:100001:EP-KdvTestEpak-v1-0-KDVFORM2-ExampleForm"

cdnUrl="https://eservices:7777/PortalEmulator/cdn/{{version}}"

routerServiceUrl="https://eservices:7777/Router/RouterService.svc"/>

<nasa subscribeToNotifications="false"/>

…

</lvp>

6. tabula

**lvp.general elementu uzskaitījums**

|  |  |
| --- | --- |
| **Iestatījums** | **Apraksts** |
| eserviceUrn | KDV formas identifikators jeb URN (izsniedz EPU). |
| cdnUrl | CDN saknes URL (tiek izmantots CSS un JS piegādāšanai e-pakalpojumam).  Jānorāda CDN tīmekļa lietojuma adrese ar KDV formai atbilstošo versijas mapi noslēgumā. Parasti adrese jāveido kopā ar virtuālās direktorijas nosaukumu)+ „*CDN*” + versijas numurs. Vēlams versijas numuru nenorādīt tieši, bet tā vietā ievietot marķieri „*{{version}}*” (piemēram, „*https://epakalpojumi.vraa.gov.lv/Lvp.Eservice.Cdn/CDN/{{version}}*”). Šādā gadījumā versija tiks ņemta no e-pakalpojumu katalogā reģistrētās *CdnVersion* lauka vērtības (t.i. CDN versiju konfigurāciju nevajadzēs dublēt divās vietās). |
| routerServiceUrl | E-pakalpojumu maršrutēšanas servisa URL, piemēram, [*https://epakalpojumi.vraa.gov.lv/Lvp.Eservice.RouterService/RouterService.svc*](https://epakalpojumi.vraa.gov.lv/Lvp.Eservice.RouterService/RouterService.svc)*.* |
| userProfileApplicationName | Identificē sistēmu, kuras lietotāju profila dati tiek pieprasīti. Parasti nevajag norādīt, jo noklusētā vērtība ir atbilstoša. |
| edkRepositoryId | Nosaka EDK izmantojamo repozitoriju. Paredzēts izmantot tikai vienu repozitoriju, tādēļ parasti vērtību nevajag norādīt, jo pēc noklusējuma tā ir atbilstoša. |
| portalConfigCacheTimeoutInSeconds | Nosaka, cik ilgi kešatmiņā tiek glabāti no e-pakalpojumu konfigurācijas servisa izgūtie iestatījumi. Pēc noklusējuma 30 minūtes. Norādot vērtību 0, rezultāti kešatmiņā glabāti netiek. E-pakalpojumu izstrādes laikā ir ieteicams šo vērtību uzstādīt uz 0, lai emulatorā mainot e-pakalpojuma iestatījumus izmaiņas tūlīt stātos spēkā e-pakalpojumā. Produkcijā vēlams šo iestatījumu nenorādīt (noklusētā vērtība). |
| subscribeToNotifications | Parametrs nosaka, vai veikt pierakstu uz NASA ziņojumiem. Izstrādājot KDV formas, NASA pieraksta veikšana nav nepieciešama, bet produkcijas vidē tā ir vitāli svarīga (ja nebūs pieraksta, KDV forma neuzzinās, ka e-pakalpojumu katalogā ir veiktas izmaiņas). Pēc noklusējuma atribūta vērtība ir true, tādēļ dzēšot atribūtu, NASA pierakstu veidošana tiks ieslēgta. |

## Kdv sekcija

Konfigurācijas sekcijā tiek norādīti sekojoši iestatījumi:

<kdv

transport="service"

transportArgs="C:\Users\egils.stamurs\Documents\Projects\LVP\Lvp.Kdv.Forms\Sources\KdvFormsTransportEmulator "

transportAuthRequired="true"

messageGuidPattern="^[\da-f]{8}-[\da-f]{4}-[\da-f]{4}-[\da-f]{4}-[\da-f]{12}$"

dataServiceEndpointAddress="net.pipe://localhost/Lvp.Kdv.RouterService/RouterService.svc">

</kdv>

7. tabula

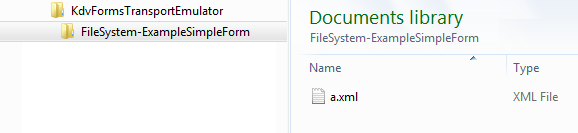
**kdv elementu uzskaitījums**

| **Iestatījums** | **Apraksts** |
| --- | --- |
| transport | XML saglabāšanas veids. Iespējamās vērtības:  „service” – bez emulācijas (tiek strādāts ar Web-servisiem);  „fs” (file system) – ar emulāciju (tiek strādāts ar XML datnēm lokālajā failu sistēmā). |
| *transportArgs* | Saglabāšanas veida argumenti. Atkarībā no saglabāšanas veida, saglabāšanas veida argumentos var norādīt:  Saglabāšanas veids „service” – neko;  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. |
| *transportAuthRequired* | Pazīme vai izvēlētajam saglabāšanas veidam būs nepieciešama lietotāja autentifikācija. |
| *messageGuidPattern* | Darba uzdevuma identifikatora šablons. |
| *dataServiceEndpointAddress* | Norāde uz KDV maršrutēšanas pakalpi. |

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:



## EserviceConfiguration galapunkta sekcija

Gadījumā, ja KDV formas konfigurācijas datnē ir galapunkta (*endpoint*) sekcija ar nosaukumu *EserviceConfigurationService*, tā tiek izmantota, lai pieslēgtos konfigurācijas servisam. Ja sekcijas ar minēto nosaukumu nav, konfigurācijas servisa izsaukšana notiek caur maršrutēšanas servisu.

Tādēļ gadījumos, kad KDV formu izstrādā vairāki cilvēki un tiek izmantots viens maršrutēšanas serviss, šai sekcijai ir jāpaliek lietojuma konfigurācijā, lai nodrošinātu, ka uzstādījumi tiek ņemti no katra izstrādātāja emulatora. Savukārt, ja maršrutēšanas serviss ir uzstādīts pie katra izstrādātāja vai arī KDV forma tiek darbināta produkcijas vidē, šo sekciju no e-pakalpojuma konfigurācijas var dzēst un paļauties uz maršrutēšanas servisa iestatījumiem.

<bindings>

<customBinding>

<binding name="EserviceConfigurationService.Binding">

<textMessageEncoding messageVersion="Soap12"/>

<httpsTransport/>

</binding>

</customBinding>

</bindings>

<client>

<endpoint name="EserviceConfigurationService"

address=[https://eservices:7777/PortalEmulator/Services/ EserviceConfigurationService.svc/soap12](https://eservices:7777/PortalEmulator/Services/%20EserviceConfigurationService.svc/soap12)

binding="customBinding"

bindingConfiguration="EserviceConfigurationService.Binding"

contract="Viss.EserviceCatalogue.DataServices.Base.Configuration.IEserviceConfigurationService" />

</client>

# Portāla emulatora lietotāja ceļvedis

Pēc noklusētās konfigurācijas emulators pieejams adresē [*https://eservices:7777/PortalEmulator*](https://eservices:7777/PortalEmulator).

Detalizētu informāciju par portāla emulatoru skatīt [2] dokumentā.