Valsts reģionālās attīstības aģentūra



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

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

----

versija

Rīgā 2013

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

Dokumenta identifikācija

|  |  |
| --- | --- |
| Dokumenta ID: | -----V- |
| Dokumenta nosaukums: | Valsts informācijas sistēmu savietotāja, Latvijas valsts portāla www.latvija.lv un elektronisko pakalpojumu izstrāde un uzturēšana.  .  .  . |
| Dokumenta kods: | ---- |
| Versija: | Versija , Laidiens (saīsināti V ) |

Saskaņojumi

|  |  |  |  |
| --- | --- | --- | --- |
| Organizācija | Vārds, uzvārds, amats | Datums | Paraksts |
|  | Atbildīgā persona no Pasūtītāja puses |  |  |
| SIA "ABC software" | J.Korņijenko, projekta vadītājs par tehniskiem jautājumiem no Izpildītāja puses |  |  |
| SIA "ABC software" | A.Zeļikovičs, projektētājs |  |  |
| SIA "ABC software" | E.Blumberga, projekta kvalitātes kontroles vadītāja |  |  |

Izmaiņu vēsture

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Versija | Datums | Apraksts | Organizācija | Autors |
| 1.00 | 13.02.2014 | Izveidota dokumenta sākotnējā versija | SIA „ABC software” | A.Zeļikovičs |
| 1.01 | 21.08.2014 | Pievienots aprasksts pār savienošanu ar sertifikātu | SIA „ABC software” | A.Zeļikovičs |

**Satura rādītājs**

[Attēlu saraksts 6](#_Toc396406839)

[1. Ievads 7](#_Toc396406840)

[1.1. Dokumenta nolūks 7](#_Toc396406841)

[1.2. Darbības sfēra 7](#_Toc396406842)

[1.3. Termini un pieņemtie apzīmējumi 7](#_Toc396406843)

[1.4. Saistītie dokumenti 7](#_Toc396406844)

[1.5. Dokumenta pārskats 8](#_Toc396406845)

[2. Integrācijas apraksts 9](#_Toc396406846)

[2.1. Konfigurācijas datnes apraksts 9](#_Toc396406847)

[2.2. Mapes izveidošanas vai pārbaudes piemērs ((1)CreateFolder) 10](#_Toc396406848)

[2.2.1. Metodes īpašības 10](#_Toc396406849)

[2.2.2. Piemērs 11](#_Toc396406850)

[2.3. Datnes izveidošana ((2)AddFile) 11](#_Toc396406851)

[2.3.1. Metodes īpašības 11](#_Toc396406852)

[2.3.2. Piemērs 11](#_Toc396406853)

[2.4. Datnes izveidošana un dzēšana ((3)AddFileRemove) 11](#_Toc396406854)

[2.4.1. Metodes īpašības 11](#_Toc396406855)

[2.4.2. Piemērs 12](#_Toc396406856)

[2.5. Mapes dzēšana ((4)RemoveFolder) 12](#_Toc396406857)

[2.5.1. Metodes īpašības 12](#_Toc396406858)

[2.5.2. Piemērs 12](#_Toc396406859)

[2.6. Dublēt datni citā mapē ((5)CopyFileToFolder) 12](#_Toc396406860)

[2.6.1. Metodes īpašības 13](#_Toc396406861)

[2.6.2. Piemērs 13](#_Toc396406862)

[2.7. Pārbaudīt mapi ((6)CheckFileAndFolderExist) 13](#_Toc396406863)

[2.7.1. Metodes īpašības 13](#_Toc396406864)

[2.7.2. Piemērs 14](#_Toc396406865)

[2.8. Pārbaudīt mapi ((7)GetDirFiles) 14](#_Toc396406866)

[2.8.1. Metodes īpašības 14](#_Toc396406867)

[2.8.2. Piemērs 14](#_Toc396406868)

[2.9. Dabūt datni ((8)GetFileContent) 14](#_Toc396406869)

[2.9.1. Metodes īpašības 14](#_Toc396406870)

[2.9.2. Piemērs 15](#_Toc396406871)

[2.10. Atrast vajadzīgas datnes mapē ((9)SearchFile) 15](#_Toc396406872)

[2.10.1. Metodes īpašības 15](#_Toc396406873)

[2.10.2. Piemērs 16](#_Toc396406874)

[2.10.3. Meklēšanas XML izveidošanas rekomendācija 16](#_Toc396406875)

[2.11. Savienošana ar sertifikātu 17](#_Toc396406876)

[2.11.1. Sertifikāta uzstādīšana 17](#_Toc396406877)

[2.11.1.1. Pārbaudīt sertifikāta thumbprint 19](#_Toc396406878)

[2.11.1.2. Konfigurācijas un sertifikāta sakritība 20](#_Toc396406879)

# Attēlu saraksts

[1.attēls. Piemēru projekti 9](#_Toc396406880)

[2.attēls. EDK izsaukšanas shēma 9](#_Toc396406881)

[3.attēls. EDK izsaukšanas shēmas konfigurācija 10](#_Toc396406882)

[4.attēls. xml izveidošanas piemērs 17](#_Toc396406883)

[5.attēls. Konsoles atvēršana 17](#_Toc396406884)

[6.attēls. Sertifikāta pievienošana 18](#_Toc396406885)

[7.attēls. Sertifikāta pievienošana 18](#_Toc396406886)

[8.attēls. Importēšanas izvēlne 18](#_Toc396406887)

[9.attēls. Importēšanas izvēlne 19](#_Toc396406888)

[10.attēls. Sertifikātu izvēlne 19](#_Toc396406889)

[11.attēls. Sertifikāta dati 19](#_Toc396406890)

# Ievads

## Dokumenta nolūks

Iestādes lietotāju (darbinieku) darba vietas aplikācija (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.

Dokumenta mērķis ir aprakstīt saskarnes, kas nodrošina moduļa integrāciju.

## Darbības sfēra

Šajā dokumentā ir uzskaitītas IDDV lietotnes integrācijas iespējas.

Šis dokuments ir paredzēts:

* projekta pasūtītāja (VRAA) pārstāvjiem, kuri ir atbildīgi par projekta nodevumu pieņemšanu un izvērtēšanu;
* projekta izpildītāja darbinieki, kuri ir atbildīgi par sistēmas projektēšanu, implementāciju un ieviešanu;
* trešo pušu izstrādātājiem, kuri izstrādās atsevišķas komponentes vai e-pakalpojumus, kuri tiks darbināti portālā.

## Termini un pieņemtie apzīmējumi

Apzīmējumu un terminu vārdnīca pieejama dokumentā [2].

## Saistītie dokumenti

Dokuments ir izstrādāts, balstoties uz šādiem dokumentiem:

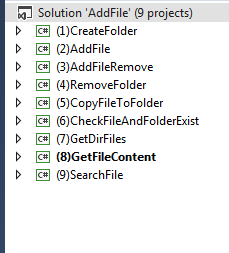
1. Valsts informācijas sistēmu savietotāja, Latvijas valsts portāla www.latvija.lv un elektronisko pakalpojumu izstrāde un uzturēšana. 3.daļa „VISS un Portāla jaunu un esošo moduļu papildinājumu izstrāde, ieviešana, garantijas apkalpošana un uzturēšana saskaņā ar tehnisko specifikāciju". Iestādes lietotāju darba vieta. Programmatūras prasību specifikācija.(VRAA-6\_15\_11\_58-VISS\_2010-IDDV-PPS)
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. Terminu un saīsinājumu indekss. (VRAA-6\_15\_11\_58-VISS\_2010-TSI).
3. 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". Elektronisko dokumentu krātuve: ārējās saskarnes. Programmatūras projektējuma apraksts. (VRAA-6\_15\_11\_58-VISS\_2010-EDK\_AS-PPA)

## Dokumenta pārskats

Dokuments sastāv no tādiem nodalījumiem:

* Dokumenta ievads – aprakstīts dokumenta nolūks, termini un pieņemtie apzīmējumi, kā arī norādīta saistība ar citiem dokumentiem un materiāliem;
* Vispārīgs apraksts – sniedz ieskatu par servisu izmantošanu;

# Integrācijas apraksts



.attēls. Piemēru projekti

## Konfigurācijas datnes apraksts

Lai izsaukt EDK servisus vajag autentificēties STSā. Autentifikācija tiek veidota ar ws2007FederationHttpBinding palīdzību (1).



2.attēls. EDK izsaukšanas shēma



.attēls. EDK izsaukšanas shēmas konfigurācija

## Mapes izveidošanas vai pārbaudes piemērs ((1)CreateFolder)

Lai veidot šo operāciju vajag pievienot Abc.Edk.Client.dll.

Pirms veidot operāciju vajag inicializēt klasi ar repozitorija identifikatoru.

### Metodes īpašības

Ar metodes palīdzību var izveidot mapes. Ja ceļā (parametra), kuru padodat metodē, mapes neeksistē EDK, tad metode pati izveidos šo mapi un atgriezis izveidotas vai eksistējošas galu mapes EDK identifikatoru.

### Piemērs

string authorityId = System.Configuration.ConfigurationManager.AppSettings.Get("TokenAuth");

string transactionId = "URN:IVIS:100001:EP-Authority-v1-3-TR-340";

DateTime today = DateTime.Now;

string path = string.Format(CultureInfo.InvariantCulture, "/Ath/{0}/{1}", authorityId, transactionId); ;

string repositoryURN = System.Configuration.ConfigurationManager.AppSettings.Get("EdkRepositoryURN");

EdkClients.InitForNoImpersonalization(repositoryURN);

var ret = EdkClients.Helper.GetOrCreateFolderPath(path);

## Datnes izveidošana ((2)AddFile)

Lai veidot šo operāciju vajag pievienot Abc.Edk.Client.dll.

Pirms veidot operāciju vajag inicializēt klasi ar repozitorija identifikatoru.

### Metodes īpašības

Datnes izveidošanas metode piemēra tiek izveidota mape vai dabūts mapes identifikators.

Izmantojot metodes CreateFile tiek padots baitu masīvs, mapes identifikators un datnes nosaukums.

### Piemērs

string authorityId = System.Configuration.ConfigurationManager.AppSettings.Get("TokenAuth");

string transactionId = "URN:IVIS:100001:EP-Authority-v1-3-TR-340";

DateTime today = DateTime.Now;

string path = string.Format(CultureInfo.InvariantCulture, "/Ath/{0}/EPAK/{1}/In/{2}/{3}/", authorityId, transactionId, today.ToString("yyyyMM"), today.ToString("ddHH")); ;

string repositoryURN = System.Configuration.ConfigurationManager.AppSettings.Get("EdkRepositoryURN");

EdkClients.InitForNoImpersonalization(repositoryURN);

var file = EdkUtils.FileToByteArray("LargeData.xml");

var fileName = EdkClients.Helper.CreateFile(path, file, "LargeData.xml");

var fileUrn = edkUtils.CreateFile(file, ret, "data.xml");

## Datnes izveidošana un dzēšana ((3)AddFileRemove)

Lai veidot šo operāciju vajag pievienot Abc.Edk.Client.dll.

Pirms veidot operāciju vajag inicializēt klasi ar repozitorija identifikatoru.

### Metodes īpašības

Lai nodzēst datni vajag padot datnes identifikatoru

### Piemērs

string authorityId = System.Configuration.ConfigurationManager.AppSettings.Get("TokenAuth");

string transactionId = "URN:IVIS:100001:EP-Authority-v1-3-TR-340";

DateTime today = DateTime.Now;

string path = string.Format(CultureInfo.InvariantCulture, "/Ath/{0}/EPAK/{1}", authorityId, transactionId);

string repositoryURN = System.Configuration.ConfigurationManager.AppSettings.Get("EdkRepositoryURN");

EdkClients.InitForNoImpersonalization(repositoryURN);

var ret = EdkClients.Helper.CreateFile(path, "Data.xml");

if (ret != null)

{

EdkClients.Helper.DeleteObject(ret.Id);

}

## Mapes dzēšana ((4)RemoveFolder)

Lai veidot šo operāciju vajag pievienot Abc.Edk.Client.dll.

Pirms veidot operāciju vajag inicializēt klasi ar repozitorija identifikatoru.

### Metodes īpašības

Ar metodi DeleteFolderWithFiles būs dzēstas visas datnes no mapes, un pēc tam dzēsta pati mape.

### Piemērs

string authorityId = System.Configuration.ConfigurationManager.AppSettings.Get("TokenAuth");

string transactionId = "URN:IVIS:100001:EP-Authority-v1-3-TR-340";

DateTime today = DateTime.Now;

string path = string.Format(CultureInfo.InvariantCulture, "/Ath/{0}/EPAK/{1}", authorityId, transactionId);

string repositoryURN = System.Configuration.ConfigurationManager.AppSettings.Get("EdkRepositoryURN");

EdkClients.InitForNoImpersonalization(repositoryURN);

var ret = EdkClients.Helper.CreateFile(path, "Data.xml");

if (ret != null)

{

EdkClients.Helper.DeleteFolderWithFiles(ret.FolderId, true);

}

## Dublēt datni citā mapē ((5)CopyFileToFolder)

Lai veidot šo operāciju vajag pievienot Abc.Edk.Client.dll.

Pirms veidot operāciju vajag inicializēt klasi ar repozitorija identifikatoru.

### Metodes īpašības

Var dublēt datni citā mapē vienkārši noradot datnes identifikatoru un mapes identifikatoru.

Izmantojot šo metodi datne būs abstrakti dublēta.

### Piemērs

string authorityId = System.Configuration.ConfigurationManager.AppSettings.Get("TokenAuth");

string transactionId = "URN:IVIS:100001:EP-Authority-v1-3-TR-340";

DateTime today = DateTime.Now;

string path = string.Format(CultureInfo.InvariantCulture, "/Ath/{0}/EPAK/{1}", authorityId, transactionId);

string repositoryURN = System.Configuration.ConfigurationManager.AppSettings.Get("EdkRepositoryURN");

EdkClients.InitForNoImpersonalization(repositoryURN);

var ret = EdkClients.Helper.GetOrCreateFolderPath(path);

var file = EdkClients.Helper.CreateFile(path, "Data.xml");

if (file != null)

{

authorityId = System.Configuration.ConfigurationManager.AppSettings.Get("TokenAuth");

transactionId = "URN:IVIS:100001:EP-Authority-v1-3-TR-341";

path = string.Format(CultureInfo.InvariantCulture, "/Ath/{0}/EPAK/{1}", authorityId, transactionId);

EdkClients.Helper.DublicateObjectToFolder(path, file.Id);

}

## Pārbaudīt mapi ((6)CheckFileAndFolderExist)

Lai veidot šo operāciju vajag pievienot Abc.Edk.Client.dll.

Pirms veidot operāciju vajag inicializēt klasi ar repozitorija identifikatoru.

### Metodes īpašības

Metode nepievienojot datni pārbauda vai tās eksistē

### Piemērs

string authorityId = System.Configuration.ConfigurationManager.AppSettings.Get("TokenAuth");

string transactionId = "URN:IVIS:100001:EP-Authority-v1-3-TR-340";

DateTime today = DateTime.Now;

string path = string.Format(CultureInfo.InvariantCulture, "/Ath/{0}/EPAK/{1}", authorityId, transactionId);

string repositoryURN = System.Configuration.ConfigurationManager.AppSettings.Get("EdkRepositoryURN");

EdkClients.InitForNoImpersonalization(repositoryURN);

var ret = EdkClients.Helper.GetOrCreateFolderPath(path);

var result = EdkClients.Helper.CheckFolderExists(path);

## Pārbaudīt mapi ((7)GetDirFiles)

Lai veidot šo operāciju vajag pievienot Abc.Edk.Client.dll.

Pirms veidot operāciju vajag inicializēt klasi ar repozitorija identifikatoru.

### Metodes īpašības

Ar metodi GetDirFiles būs dabūti visas datnes no mapes, kuru noradījāt parametrā.

### Piemērs

string authorityId = System.Configuration.ConfigurationManager.AppSettings.Get("TokenAuth");

string transactionId = "URN:IVIS:100001:EP-Authority-v1-3-TR-340";

DateTime today = DateTime.Now;

string path = string.Format(CultureInfo.InvariantCulture, "/Ath/{0}/EPAK/{1}", authorityId, transactionId);

string repositoryURN = System.Configuration.ConfigurationManager.AppSettings.Get("EdkRepositoryURN");

EdkClients.InitForNoImpersonalization(repositoryURN);

var ret = EdkClients.Helper.GetOrCreateFolderPath(path);

var files = EdkClients.Helper.GetObjects(ret);

## Dabūt datni ((8)GetFileContent)

Lai veidot šo operāciju vajag pievienot Abc.Edk.Client.dll.

Pirms veidot operāciju vajag inicializēt klasi ar repozitorija identifikatoru.

### Metodes īpašības

Ar metodi GetFileContent būs dabūts datnes

### Piemērs

string authorityId = System.Configuration.ConfigurationManager.AppSettings.Get("TokenAuth");

string transactionId = "URN:IVIS:100001:EP-Authority-v1-3-TR-340";

DateTime today = DateTime.Now;

string path = string.Format(CultureInfo.InvariantCulture, "/Ath/{0}/EPAK/{1}", authorityId, transactionId);

string repositoryURN = System.Configuration.ConfigurationManager.AppSettings.Get("EdkRepositoryURN");

EdkClients.InitForNoImpersonalization(repositoryURN);

var ret = EdkClients.Helper.CreateFile(path, "Data.xml");

if (ret != null)

{

var files = EdkClients.Helper.GetObjects(ret.FolderId);

foreach (var item in files)

{

if (item != null && !item.IsFolder && item.File != null)

{

var stream = EdkClients.Helper.GetContentStream(item.File.Id);

}

}

}

## Atrast vajadzīgas datnes mapē ((9)SearchFile)

Lai veidot šo operāciju vajag pievienot Abc.Edk.Client.dll.

Pirms veidot operāciju vajag inicializēt klasi ar repozitorija identifikatoru.

### Metodes īpašības

Ar metodi SearchFile būs dabūti visas datnes no mapes, kuru noradījāt parametrā, ar filtrēšanu

### Piemērs

string authorityId = "100267";

string transactionId = "URN:IVIS:100001:EP-Authority-v1-3-TR-340";

DateTime today = DateTime.Now;

string path = string.Format(CultureInfo.InvariantCulture, "/Ath/{0}/EPAK/{1}", authorityId, transactionId);

string repositoryURN = "URN:IVIS:100266:EDK-0000001";

EdkClients.InitForNoImpersonalization(repositoryURN);

var ret = EdkClients.Helper.GetOrCreateFolderPath(path);

var data = EdkClients.Helper.GetObjects(ret, fileName: "data.xml", objectMaxSize: 63, objectMinSize: 61);

if (data != null)

{

}

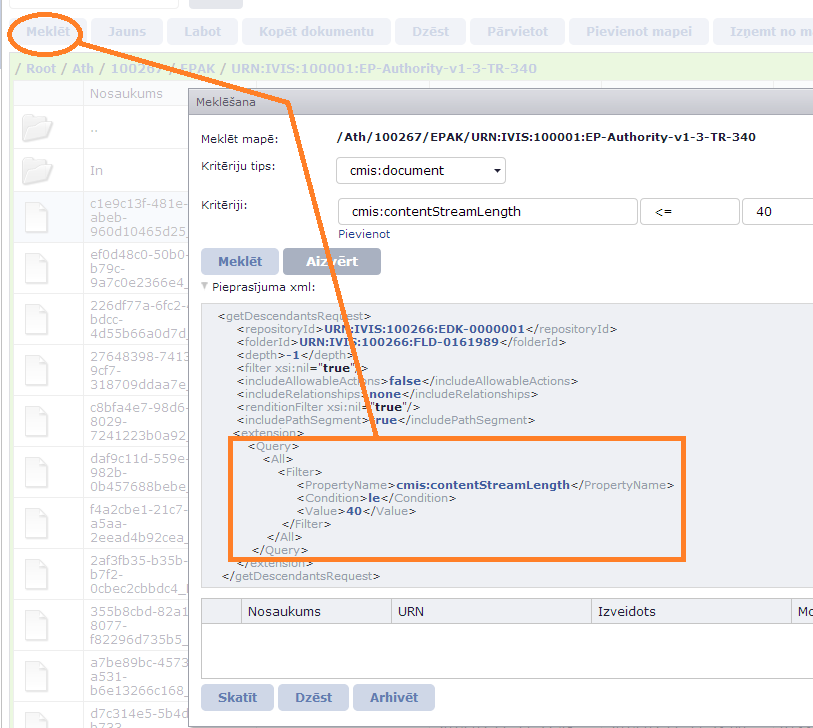
data = EdkClients.Helper.GetObjects(ret, "<Query> <Either> <Either> <All> <Filter><PropertyName>cmis:name</PropertyName><Condition>like</Condition><Value>Data</Value> </Filter> </All></Either> </Either></Query>");

if (data != null)

{

}

### Meklēšanas XML izveidošanas rekomendācija



.attēls. xml izveidošanas piemērs

## Savienošana ar sertifikātu

### Sertifikāta uzstādīšana

Izvēlamies sertifikātu osb-test.pfx (piemēram).

Sertifikāta uzstādīšana konsolē:

1. Atvērt konsoli;



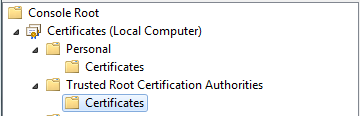
.attēls. Konsoles atvēršana

1. Pievienot sertifikātu; (parole: 123)



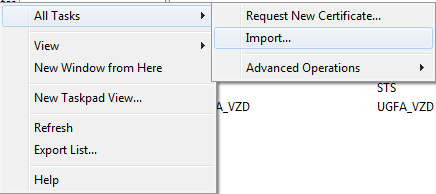
.attēls. Sertifikāta pievienošana

1. Atvērt *Trusted Root Sertificate Authorities* sertifikātus;



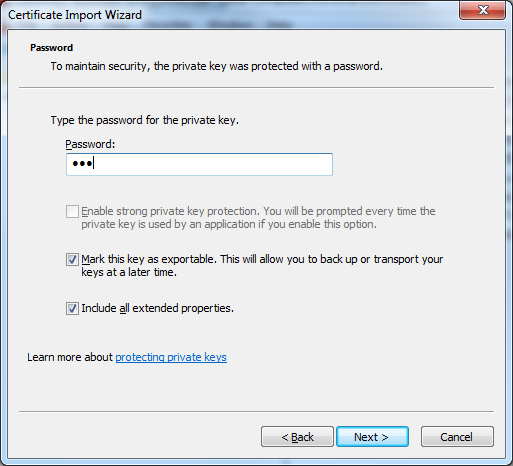
.attēls. Sertifikāta pievienošana

1. Importēt osb-test.pfx personālos sertifikātos:



.attēls. Importēšanas izvēlne

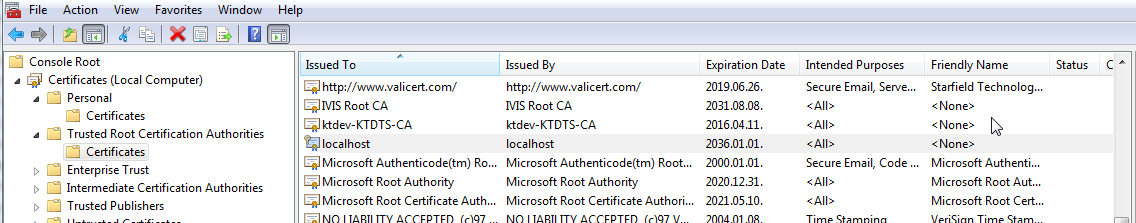
1. Izvelēties sertifikātu un atzīmēt, tā kā paradīts 9.attēlā ar paroli „123”:
2. „*Mark this key as*….” jābūt atzīmētai;
3. „*Include all expanded properties*” jābūt atzīmētai.



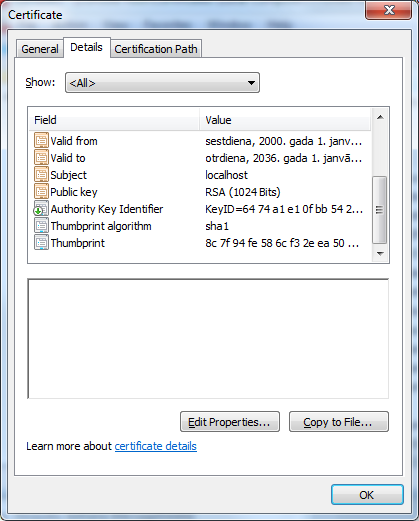
.attēls. Importēšanas izvēlne

#### Pārbaudīt sertifikāta thumbprint

1. Ar kreiso peles pogu noklikšķināt divas reizes uz sertifikāta;



.attēls. Sertifikātu izvēlne



.attēls. Sertifikāta dati

#### Konfigurācijas un sertifikāta sakritība

Konfigurācija un sertifikāta sakritība izskatīsies, kā zemāk redzams, turklāt tai jāsakrīt ar osb-test.pfx *thumbprint*.

<behaviors>

<endpointBehaviors>

<behavior name="certificateIPDataRouter">

<clientCredentials>

<clientCertificate findValue="7d a6 83 cf 8b 85 9a 23 8f 26 c3 19 1a 98 fb 0c e3 9f 60 12"

storeLocation="LocalMachine" x509FindType="FindByThumbprint" />

</clientCredentials>

</behavior>

</endpointBehaviors>

</behaviors>