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"

Notifikācijas serviss

Integrācijas instrukcija un paraugi

VRAA-6\_15\_11\_58-VISS\_2010-NTF-II

04.03.2013. versija 1.00

Rīgā 2013

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

 Dokumenta identifikācija

|  |  |
| --- | --- |
| Dokumenta ID: | VRAA-6\_15\_11\_58-VISS\_2010-NTF-II-V1.00-04.03.2013. |
| 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".Notifikācijas serviss.Integrācijas instrukcija un paraugi. |
| Dokumenta kods: | VRAA-6\_15\_11\_58-VISS\_2010-NTF-II |
| Versija: | Versija 1.00, Laidiens 04.03.2013. (saīsināti V1.00 04.03.2013.) |

Saskaņojumi

|  |  |  |  |
| --- | --- | --- | --- |
| Organizācija | Vārds, uzvārds, amats  | Datums | Paraksts |
| Valsts reģionālās attīstības aģentūra | Atbildīgā persona no Pasūtītāja puses |  |  |
| SIA "ABC software" | J.Korņijenko, projekta vadītājs par tehniskiem jautājumiem no Izpildītāja puses | 04.03.2013. |  |
| SIA "ABC software" | M.Pētersons, projekta vadītājs par administratīviem jautājumiem no Izpildītāja puses | 04.03.2013. |  |
| SIA "ABC software" | A.Zeļikovičs, vecākais programmētājs  | 04.03.2013. |  |
| SIA "ABC software" | E.Blumberga, projekta kvalitātes kontroles vadītāja | 04.03.2013. |  |

Izmaiņu vēsture

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Versija | Datums | Apraksts | Organizācija | Autors |
| 1.00 | 04.03.2013. | Izveidota dokumenta sākotnējā versija  | SIA „ABC software” | A.Zeļikovičs |
|  |  |  |  |  |

**Satura rādītājs**

[Attēlu saraksts 5](#_Toc367288664)

[1. Ievads 6](#_Toc367288665)

[1.1. Dokumenta nolūks 6](#_Toc367288666)

[1.2. Darbības sfēra 6](#_Toc367288667)

[1.3. Termini un pieņemtie apzīmējumi 6](#_Toc367288668)

[1.4. Saistītie dokumenti 6](#_Toc367288669)

[1.5. Dokumenta pārskats 7](#_Toc367288670)

[2. Vispārējais apraksts 8](#_Toc367288671)

[2.1. Konfigurācija 8](#_Toc367288672)

[2.1.1. Datubāzes uzstādīšana 8](#_Toc367288673)

[2.1.2. Piemēra konfigurācija 9](#_Toc367288674)

[2.1.2.1. AppSettings konfigurācija 9](#_Toc367288675)

[2.1.2.2. Savienojuma rinda 9](#_Toc367288676)

[2.2. Piemērs sūtīšana grupai 10](#_Toc367288677)

[2.2.1. Piemēra palaišana 11](#_Toc367288678)

[2.3. Piemērs sūtīšana personai 11](#_Toc367288679)

[2.3.1. Piemēra palaišana 12](#_Toc367288680)

[2.4. Piemērs sūtīšana no anonīma lietotāja 12](#_Toc367288681)

[2.4.1. Piemēra palaišana 13](#_Toc367288682)

# Attēlu saraksts

[1.attēls. Saņēmēju XML shēma 8](#_Toc367288683)

# Ievads

## Dokumenta nolūks

Aprakstīt notifikācijas servisa biznesa loģikas izmantošanas piemērus. Dokuments ir paredzēts VISS sistēmu izstrādātājiem, kuriem nav paredzēta pilnīga VISS bibliotēkas *Diagnostic* notifikācijas iespēja (skat. [4]).

## Darbības sfēra

Notifikāciju servisam tiek izvirzīti šādi pamatuzdevumi:

* Nosūtīt ziņojumu – metode, kura nosūta paziņojumu, izmantojot norādīto paziņojumu sūtīšanas kanālu. Paziņojumu veidošana notiek, izmantojot saņemto XML un atbilstošo XSL transformāciju. Papildus esošai funkcionalitātei ir paredzēts pievienot iespēju nosūtīt paziņojumu noteiktā datumā un laikā;
* Izgūt informāciju par lietotāja vai procesa sūtītajiem paziņojumiem – metode, kura atgriež sūtītāja nosūtītos paziņojumus noteiktā laika periodā;
* Izgūt informāciju par lietotāja saņemtajiem paziņojumiem – metode, kura atgriež paziņojumus, ko lietotājs ir saņēmis norādītajā laika periodā.

## 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". Notifikāciju serviss. Programmatūras prasību specifikācija. (VRAA-6\_15\_11\_58-VISS\_2010-NTF-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. 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". Notifikāciju serviss. Programmatūras projektējuma apraksts. VRAA-6\_15\_11\_58-VISS\_2010-NTF-PPA.
4. 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 sistēmas žurnāls. Koplietojuma bibliotēku apraksts. (VRAA-6\_15\_11\_58-VISS\_2010-VISS\_ZUR-KBA).

## Dokumenta pārskats

Dokuments sastāv no šādiem nodalījumiem:

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

# **Vispārējais apraksts**

Piemēros ir aprakstīti sūtīšanas piemēri trīs saņēmēju grupām (skat. 1.attēlā):

* Lietotājam’;
* Grupai;
* Sūtīšana no anonīma lietotāja.



1.attēls. Saņēmēju XML shēma

## Konfigurācija

Notifikācijas servisa piemēru darbināšana ietver šādus obligātus soļus:

* Datubāzes uzstādīšana;
* .NET piemēra konfigurācija.

### Datubāzes uzstādīšana

Lai uzstādītu Notifikāciju datubāzi, nepieciešams atvērt „SQL Server Management Studio” programmatūru, pieslēdzoties pie SQL servera, kur tiek plānota datubāzes uzstādīšana, un tajā ielādēt instalācijas skriptu „VISS\_Notifications\_DB\_Setup.sql”. Nepieciešams pārslēgt skripta izpildes režīmu uz „SQLCMD”, režīmu – izvēlnē „Query” atzīmēt „SQLCMD mode”.

SQL skripta sākumdaļā nepieciešams uzstādīt šādus parametrus:

:setvar DefaultDataPath "F:\MSSQLDATA\MSSQL10\_50.MSSQLSERVER\MSSQL\DATA\"

:setvar DefaultLogPath "F:\MSSQLDATA\MSSQL10\_50.MSSQLSERVER\MSSQL\DATA\"

Parametru vērtības aprakstītas 1.tabulā:

1.tabula

DPPS datubāzes uzstādīšanas skripta parametri

|  |  |
| --- | --- |
| Parametrs | Vērtības apraksts |
| DefaultDataPath | Ceļš, kur jāglabā datubāzes datus saturošo datni. |
| DefaultLogPath | Ceļš, kur jāglabā datubāzes transakcijas žurnāla datni. |

Pēc parametru uzstādīšanas nepieciešams palaist šo skriptu. Rezultātā tiks izveidota Notifikāciju datubāze.

Nepieciešams izveidot datubāzes lietotāju Notifikāciju datubāzei un to piesaistīt DB lomām „db\_datareader” un „db\_datawriter”, kā arī piešķirt tiesības „Execute” datubāzes līmenī.

Nepieciešams izveidot SQL sistēmas regulāro darbību (JOB), kas nostrādā katru dienu ik pēc 10 (desmit) minūtēm, un izpilda zemāk aprakstīto „Transact-SQL skript” tipa soli uz „VISS\_Notifications” datubāzes:

* Novecojušo notifikāciju atcelšanas solis:

DECLARE @return\_value int

EXEC @return\_value = [dbo].[NTF\_Notification\_Cancel\_Overdue]

GO

### Piemēra konfigurācija

Notifikācijas servisa piemēru konfigurācijas datnes fragments:

<appSettings>

        <add key="RC.adress" value="https://ivis.eps.gov.lv/RC.WebApp"/>

        <add key="MailForm" value="test@abcsoftware.lv"/>

        <add key="MailServer" value="mail.abc"/>

    </appSettings>

    <connectionStrings>

        <add name="NotificationDB" connectionString="metadata=res://\*/DB.NotificationDB.csdl|res://\*/DB.NotificationDB.ssdl|res://\*/DB.NotificationDB.msl;provider=System.Data.SqlClient;provider connection string=&quot;Data Source=SQL-01;Initial Catalog=VISS\_Notifications;Integrated Security=True;MultipleActiveResultSets=True;Application Name=EntityFramework&quot;" providerName="System.Data.EntityClient"/>

    </connectionStrings>

#### AppSettings konfigurācija

Notifikācijas servisa konfigurācija ietver šādus parametrus:

* RC.adress– resursu kataloga adrese;
* MailForm – e-pasta adrese, no kuras tiek sūtītas vēstules;
* MailServer – servera adrese, no kuras tiek sūtītas vēstules.

#### Savienojuma rinda

Sekcijā <connectionStrings> atribūtam „NotificationDB” jānorāda Notifikācijas datubāzes pieslēguma rinda. Jānomaina tikai „Data Source” un „Initial Catalog” vērtības.

## Piemērs sūtīšana grupai

 public static void TestGroupNotification(string groupName, AuthorityType auth)

        {

            List<ReceiverOption> receiverList = new List<ReceiverOption>();

            ReceiverOption receiver = new ReceiverOption(groupName, auth, SendingType.email);

            receiverList.Add(receiver);

            SenderReceiver senderReceiver = new SenderReceiver("PK:111111111", receiverList);

            var result = notification.SendNotification("title", senderReceiver);

            Console.WriteLine(result);

            var selectResult = notification.SelectNotification(result, false, false);

            if (selectResult != null && selectResult.ReceiverList != null && selectResult.ReceiverList.ReceiverData != null && selectResult.ReceiverList.ReceiverData.Count > 0)

            {

                foreach (var item in selectResult.ReceiverList.ReceiverData)

                {

                    if (item != null && item.Receiver != null && item.Receiver.Item != null)

                    {

                        GroupAddressStructure group = (GroupAddressStructure)item.Receiver.Item;

                        if (group.Item is GroupCodeInfoStrucure)

                        {

                            GroupCodeInfoStrucure groupData = (GroupCodeInfoStrucure)group.Item;

                            if (groupData.AuthorityID != null)

                            {

                                Console.WriteLine(string.Format("Gropup auhtority: {0}", groupData.AuthorityID));

                                Console.WriteLine(string.Format("Gropup GroupCode: {0}", groupData.GroupCode));

                            }

                            else

                            {

                                Console.WriteLine(string.Format("Gropup auhtority: {0}", groupData.AuthorityID));

                            }

                        }

                        else

                        {

                            string authorityR = (string)group.Item;

                            Console.WriteLine(string.Format("Group only auhtority: {0}", authorityR));

                        }

                    }

                }

            }

            notification.CancelNotification(result);

            Console.WriteLine(string.Format("notification canceled: {0}", result));

        }

### Piemēra palaišana

Ir iespēja sūtīt notifikāciju grupai, uz grupu ar iestādes identifikatoru vai tikai uz iestādi.

notification = new SendNotificationLogic(ServiceSettings.GetNotificationConnectionString, ServiceSettings.GetResourceCatalogAdress);

List<ReceiverOption> receiverList = new List<ReceiverOption>();

string groupName = "gropuo";

string authority = "100000";

AuthorityType auth = new AuthorityType(authority);

TestGroupNotification(groupName, auth);

Console.WriteLine("=====");

TestGroupNotification(null, auth);

Console.WriteLine("=====");

Console.ReadLine();

## Piemērs sūtīšana personai

public static void TestUserNotificationPerson(SendingType sendingType)

        {

            List<ReceiverOption> receiverList = new List<ReceiverOption>();

            VissNameIdentifier personCode = new VissNameIdentifier("PK:15057511226");

            ReceiverOption receiver = new ReceiverOption(personCode, sendingType);

            receiverList.Add(receiver);

            SenderReceiver senderReceiver = new SenderReceiver("PK:111111111", receiverList);

            var result = notification.SendNotification("title", senderReceiver);

             Console.WriteLine(result);

             var selectResult = notification.SelectNotification(result, false, false);

            if (selectResult != null && selectResult.ReceiverList != null && selectResult.ReceiverList.ReceiverData != null && selectResult.ReceiverList.ReceiverData.Count > 0)

            {

                foreach (var item in selectResult.ReceiverList.ReceiverData)

                {

                    if (item != null && item.Receiver != null && item.Receiver.Item != null)

                    {

                        UserAddressStructure user = (UserAddressStructure)item.Receiver.Item;

                        if (user.VissIdentifier != null)

                        {

                            VissNameIdentifier ident = new VissNameIdentifier(user.VissIdentifier);

                            Console.WriteLine(string.Format("receiver personCode: {0}", ident[VissNameIdentifier.PersonCodePart]));

                        }

                    }

                }

            }

            notification.CancelNotification(result);

            Console.WriteLine(string.Format("notification canceled: {0}", result));

        }

### Piemēra palaišana

   TestUserNotificationPerson(SendingType.email);

            Console.WriteLine("=====");

            Console.ReadLine();

## Piemērs sūtīšana no anonīma lietotāja

 public static void TestUnknownSenderSuccessWithBody()

        {

            Assembly \_assembly = Assembly.GetExecutingAssembly();

            StreamReader \_textStreamReader = new StreamReader(\_assembly.GetManifestResourceStream("SendConsole.data.xml"));

            XmlDocument xDoc = new XmlDocument();

            xDoc.Load(\_textStreamReader);

            XPathNavigator xPathNav = xDoc.CreateNavigator();

            List<ReceiverOption> receiverList = new List<ReceiverOption>();

            string correctMail = "alex@abcsoftware.lv";

            string bodyTransformationUrn = "URN:IVIS:100001:xslt-100000-FullNameDitTransform-0.0";

            string messageTypeUrn = "URN:IVIS:100001:XSD-Person-FullName-v1-0-TYPE-PersonFullName";

            MailAddress mailAdress = new MailAddress(correctMail);

            SenderReceiver senderReceiver = new SenderReceiver(mailAdress);

            var result = notification.SendNotification("Vasja", new URNType(messageTypeUrn), new URNType(bodyTransformationUrn), xPathNav, senderReceiver, null, null);

            Console.WriteLine(result);

            var selectResult = notification.SelectNotification(result, false, false);

            if (selectResult != null && selectResult.ReceiverList != null && selectResult.ReceiverList.ReceiverData != null && selectResult.ReceiverList.ReceiverData.Count > 0)

            {

                foreach (var item in selectResult.ReceiverList.ReceiverData)

                {

                    if (item != null && item.Receiver != null && item.Receiver.Item != null)

                    {

                        GuestAddressStructure guest = (GuestAddressStructure)item.Receiver.Item;

                        if (guest != null && guest.ItemElementName == ItemChoiceType.Email && !string.IsNullOrEmpty(guest.Item))

                        {

                            Console.WriteLine(string.Format("receiver email: {0}", guest.Item));

                        }

                    }

                }

            }

            notification.CancelNotification(result);

            Console.WriteLine(string.Format("notification canceled: {0}", result));

        }

### Piemēra palaišana

            TestUnknownSenderSuccessWithBody();

            Console.WriteLine("=====");

            Console.ReadLine();