



PaymentService

NET Genium Framework



netgenium.com

Content

1	Basic information	3
2	Supported banks	3
3	Required files	3
4	Technology box	4
5	Configure Azure Active Directory	4
6	Import email messages	5
7	Configuration file	6
8	Installation of the service	8
9	Backing up downloaded email messages	8
10	Sending notification messages	9
11	Notice	9

1 Basic information

PaymentService is a service that runs in the background of the Windows operating system and automatically creates bank payments, marks invoices and other obligations as paid based on communication with the bank. It is installed as an add-on to the NET Genium ERP application and requires a technology mailbox to which email messages about incoming and outgoing payments are sent by the bank. It automatically creates bank payments from e-mail messages delivered to the technology box and assigns them to specific invoices or other obligations.

- The basic condition is that the bank must support sending e-mail messages about incoming and outgoing payments.
- Payments are automatically matched with the relevant documents in NET Genium either based on the variable symbol and amount, or based on the account number and amount.
- Paired documents are automatically marked as partially or fully paid.
- The service works for both incoming and outgoing payments.

When a payment is recorded, a record of this payment is automatically created in the "Bank" application in the "Bank payments" table. The relevant document to which the payment relates is also automatically updated, and information about this payment is also sent to the e-mail defined in the service settings.

2 Supported banks

- eBank
- Raiffaisenbank
- Commercial Bank
- F.I.O
- Česká spořitelna – only outgoing payments (payment of received invoices and other obligations) and in addition, sending an email must be set for each outgoing payment
- UniCredit bank

3 Required files

For the PaymentService to work, it is necessary to have the following files in its directory:

- PaymentService.exe
- ERP.dll
- ngef.dll
- NETGeniumConnection.dll

+ configuration XML file for each technology box.

4 Technology box

A necessary condition for the PaymentService is the existence of a technological mailbox to which e-mail messages about incoming and outgoing payments are sent by the bank. The usual name is, for example, "ps@firma.cz" or "ps" in front of the culprit and with the domain of the given company.

The actual sending of e-mail messages about incoming and outgoing payments must be done manually in the application supplied by the banking service provider, typically in Internet banking.

5 Configure Azure Active Directory

PaymentService requires additional manual configuration of Azure Active Directory in case it downloads emails from a technology mailbox operated using Microsoft cloud services (<https://docs.microsoft.com/en-us/exchange/client-developer/legacy-protocols/how-to-authenticate-an-imap-pop-smtp-application-by-using-oauth>).

Configuring Azure Active Directory consists of the following steps:

- NET Genium configuration on "https://portal.azure.com" portal:
 - App Registration "**NET Genium**" on website https://portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/RegisteredApps
 - portal.azure.com / Azure Active Directory / App registrations
 - **NET Genium** as the application name
 - Finding **Application (client) ID**, which is automatically created when the application is registered
 - Application (client) ID
 - Finding **Directory ID (tenant)**, which is automatically created when the application is registered
 - Directory (tenant) ID
 - Creation **client secret code**
 - portal.azure.com / Azure Active Directory / App registrations / NET Genium / Certificates & secrets / New client secret
 - Setting API permissions for **Office 365 Exchange Online**
 - portal.azure.com / Azure Active Directory / App registrations / NET Genium / API permissions / Add a permission / APIs my organization uses / Office 365 Exchange Online
 - API Name "**POP.AccessAsApp**", type "**Application**"
 - API Name "**IMAP.AccessAsApp**", type "**Application**"

« Aktualizovat | Máte zpětnou vazbu?

- 🏠 Přehled
- 🚀 Rychlý start
- 🔧 Asistent integrace
- Správa
- 📄 Branding a vlastnosti
- 🔄 Ověřování
- 🔑 Certifikáty a tajné kódy
- 📄 Konfigurace tokenů
- 🔑 Oprávnění rozhraní API
- 🔑 Zpřístupnit rozhraní API

Nakonfigurovaná oprávnění

Když aplikacím udělí uživatelé nebo správci v rámci procesu souhlasu oprávnění, mají autorizaci volat rozhraní API. Seznam nakonfigurovaných oprávnění by všechna oprávnění, která aplikace potřebuje. [Další informace o oprávněních a souhlasu](#)

+ Přidat oprávnění ✓ Udělit souhlas správce pro NetGenium s.r.o.

Název rozhraní API / oprávnění	Typ	Popis	Vyžaduje se souhlas ...	Stav
<ul style="list-style-type: none"> > Microsoft Graph (9) < Office 365 Exchange Online (2) 				
IMAP.AccessAsApp	Aplikace	IMAP.AccessAsApp	Ano	✓ Uděleno pro
POP.AccessAsApp	Aplikace	POP.AccessAsApp	Ano	✓ Uděleno pro

- Creating a configuration file "**MicrosoftOAuth.json**" in the "Config" directory of the given NET Genium in which the bank payments are to be created
 - Set file content to: `{"web":{"client_id":"Application ID (client)","tenant_id":"Directory ID (tenant)","auth_uri":"https://login.microsoftonline.com/organizations/oauth2/v2.0/authorize","token_uri":"https://login.microsoftonline.com/organizations/oauth2/v2.0/token","client_secret":"client secret code"}}`
- Connecting NET Genium to the technology box (<https://www.youtube.com/watch?v=bMYA-146dmM>) in PowerShell using the following script:
 - Set-ExecutionPolicy -Scope Process -ExecutionPolicy Unrestricted
 - Install-Module -Name ExchangeOnlineManagement
 - Install-Module Microsoft.Graph
 - Install-Module -Name AzureAD
 - Import-Module AzureAD
 - Import-Module ExchangeOnlineManagement
 - Connect-AzureAD -Tenant **Directory ID (tenant)**
 - Connect-ExchangeOnline-Organization **Directory ID (tenant)**
 - \$myapp = Get-AzureADServicePrincipal -SearchString "NET Genium"
 - New-ServicePrincipal -AppId \$myapp.AppId -ServiceId \$myapp.ObjectId -DisplayName "Service Principal for POP3/IMAP Services"
 - Add-MailboxPermission -Identity "ps@firma.cz" -User \$myapp.ObjectId -AccessRights FullAccess

6 Import email messages

PaymentService can import existing "eml" files, which are email messages saved in their original format. For these purposes, it is necessary to create an import directory and set its absolute path on the disk in the configuration file under the "emlDir" attribute. PaymentService reads the files in this directory, processes them in the standard way like other e-mail messages downloaded from the mail server, and then deletes the processed files from the import directory.

7 Configuration file

It is necessary to create a configuration XML file for each technology box and place it in the same directory as "PaymentService.exe". Example configuration file:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>

    <add key="rootPath" value="C:\inetpub\wwwroot\netgenium" />

    <add key="stopFrom" value="00:00" />
    <add key="stopTo" value="06:00" />

    <add key="smtpServer" value="localhost" />

    <add key="errorFrom" value="paymentservice@firma.cz" />
    <add key="errorTo" value="support@netgenium.com" />

    <add key="pop3Url" value="pop3.firma.cz" />
    <add key="pop3Name" value="ps@firma.cz" />
    <add key="pop3Password" value="pop3topsecret" />
    <add key="pop3OAuth" value="true" />

    <add key="infoFrom" value="banka@firma.cz" />
    <add key="infoTo" value="finance@firma.cz" />
    <add key="infoCc" value="" />

    <add key="emlDir" value="D:\Services\PaymentService\eml" />

  </appSettings>
</configuration>
```

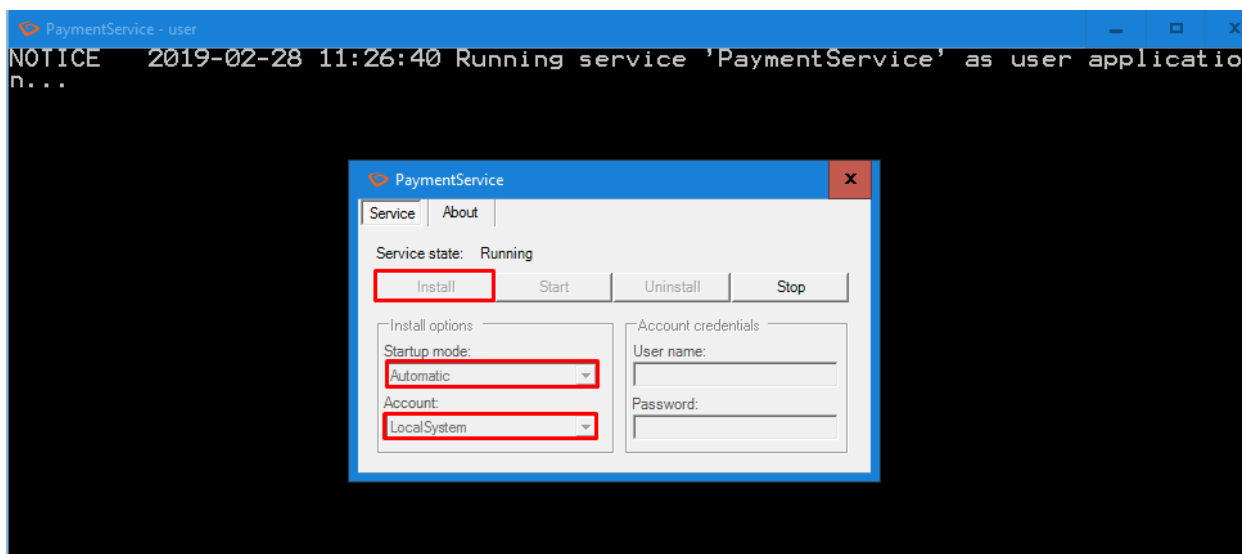
- **rootPath** – the absolute disk path to the NET Genium directory
- **stopFrom** – the start time of the technology pause in the format "HH:mm", during which the PaymentService will not connect to the technology box
- **stop to** – the end time of the technology pause in the format "HH:mm", during which the PaymentService will not connect to the technology box
- **smtpServer** – the address of the outgoing mail server
- **errorFrom** – e-mail address from which notifications about possible errors in the PaymentService are sent
- **errorTo** – e-mail address to which notifications about possible errors in the PaymentService are sent
- **pop3Url** – the address of the mail server of the incoming mail (for SSL security it is possible to use the syntax "server name: port number")
- **pop3Name** – login name to the technology mailbox
- **pop3Password** – the password to the technology mailbox
- **pop3OAuth** – flag whether to use OAuth2 enhanced security for login
- **infoFrom** – e-mail address from which the notification about the payment will be sent
- **infoTo** – e-mail address to which a notification about the payment will be sent
- **infoCc** – an optional e-mail address (copy) to which a notification about the payment will be sent

- **emlDir** – the absolute path on the disk to the directory from which the PaymentService reads and processes the “eml” files

All e-mail messages that arrive in the technology mailbox are deleted after processing.

8 Installation of the service

- i *PaymentService* is a service that runs in the background of the Windows operating system and automatically activates and downloads emails every 10 minutes. Therefore, it is not enough to simply run the "PaymentService.exe" program, but it is necessary to install the service using the "PaymentService.exe – user.lnk" file. The file "PaymentService.exe – user.lnk" is created automatically by running the "PaymentService.exe" program.
- h After running the "PaymentService.exe" program, check the dialog settings. Set Startup mode to "Automatic" and Account to "LocalSystem". You can complete the installation of the service by clicking the "Install" button.



- i You can also uninstall the program by running the "PaymentService.exe" program and clicking the "Uninstall" button.

9 Backing up downloaded email messages

Copies of all e-mail messages delivered to the technology mailbox are automatically saved in the "POP3Downloads\yyyy-MM-dd" subdirectory, which is located in the same directory as "PaymentService.exe". E-mail messages are stored under a random name (GUID.eml) and are placed in a subdirectory whose name identifies the day the e-mail message was downloaded.

p *Example:*

D:\Services\PaymentService\POP3Downloads\2013-05-28\154c463d-386a-4b61-bf0d-6b2cd91a1bcb.eml

10 Sending notification messages

An information e-mail is sent to the address specified in the configuration file each time a payment is identified and paired with the relevant document. Example of notification email text:

```
Outgoing invoice No. 130100024 issued to the company Smetáček a lopatka, spol. s.r.o. in the amount of CZK 10,890.00 was paid.
```

11 Notice

The PaymentService is dependent on the e-mail messages and their content sent by the respective bank. If, for example, the bank does not send an e-mail when paying less than CZK 100, the PaymentService will not capture this movement. In the same way, the service is not able to capture, for example, payments mediated by direct debit or standing order. Therefore, it is not recommended to use PaymentService as the only and full-fledged tool for assigning payments to individual documents. The possibility of importing a monthly account statement into NET Genium and subsequently automatically matching payments based on it is widely used, for example. This functionality can be used concurrently without restrictions with the PaymentService service.