# System requirements

## Application minimum requirements:

The server:

* Operating system - Windows Server 2012 R2 Standard,. NET Core, dotnet windows hosting;
* Services - IIS 7.0 or higher;
* Database Management System – Currently SQL server. (Later SQL server, PostgreSQL or mySQL)
* Protocols for communication between the client and the database - TCP / IP.

The client:

* Operating system - Any OS (Linux, Windows etc.);
* Browser - Chrome 71 (and newer).

## System hardware minimum requirements:

The server:

* Processor: 64-bit, 8-core, 2.2 GHz;
* Memory: Min 32 GB;
* HDD: ~ 1TB. The data storage system should support up to 50MB / s for writing and up to 100MB / c for reading.

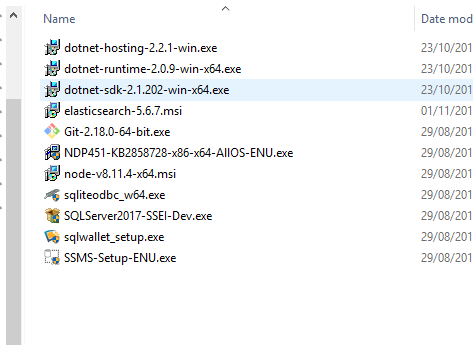
\* These servers can run on virtual machines VMWare, Hyper-V.

The workstation:

* Processor: 32/64-bit, 2-core, 2.2 GHz;
* Memory: Min 4GB;
* HDD: ~ 250 GB.

## Software requirements for system

The following software need to be installed (either before deployment start or during the process (see below)



(If the system is installed on Azure, Web deploy must also be installed). The software can be found in the zip file “software needed to run the system.zip”

*Note: after installing all requirements on some computers need to restart computer*

The installation files can be found here:

## Development environment (optional)

#### Additional software requirements for developer environment:

* dotnet-dev-win-x64.1.1.4 v (.NET SDK 1.1.4 version)
* Visual Studio editor(need to work with back-end development) or/and Visual Studio Code editor (need to work with front-end development)
* Git

.NET projects target netcoreapp1.1 or netstandard1.6

# Elastic search setup

## Installation

* Download Elastic search (elasticsearch-6.5.3)
* JRE - Java Runtime Environment (jre-8u191 or newer, stable version) for running system
* JDK - Java Development Kit (jdk-8u191 or newer, stable version) - if you are in development mode.

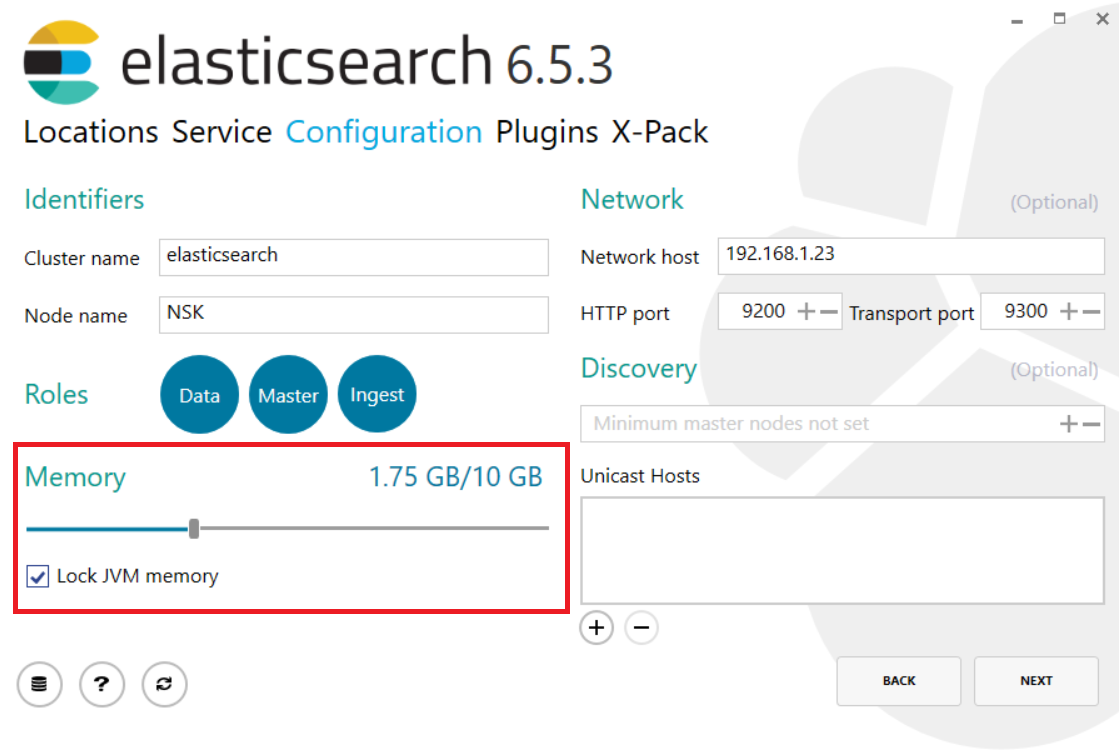
|  |
| --- |
| *IMPORTANT: Don`t forget to add jre, or jdk directory path string into Environment variables of Windows as JAVA\_HOME. If not, Elasticsearch service will be stopped automatically by Windows.*  *https://lh3.googleusercontent.com/ruSz0aVrHsFcAqTfz3s87S9wzJOQ7GnTNAzjHuxWk0JPfYGGXF4V-ZlnmmaquykLChDfRHU2LxcTeX_-epH69w5ZMzxbJjPcZr05_DgMevvCQzcUPNtXcGkUxprqJ4oqqYzUL-whttps://lh3.googleusercontent.com/fzo1EUpAVWttyrwMEWE9jScHypcTUqtyx32i9bHfVoXGroVx0KoxhpLJEGSKmFImdzr9r9S209DEimD4mNqAmliGpX-gjGO6-3X3HMoNf6ADTevmOsIvgJbBShsPT5F9haBi8jA* |

Begin install Elasticsearch you can accept default configurations at the beginning but then please configure as follows:

1. Check Lock JVM memory
2. Divide your RAM for Elasticsearch use - There Memory slider, pull it to right and divide needed memory (1GB is minimal requirement for Elasticsearch)
3. Network host - you can leave it empty, by default Elasticsearch looks to localhost (there we have defined URL of the machine) on port 9200.

*IMPORTANT 1: Elasticsearch and database should be located on one machine.*

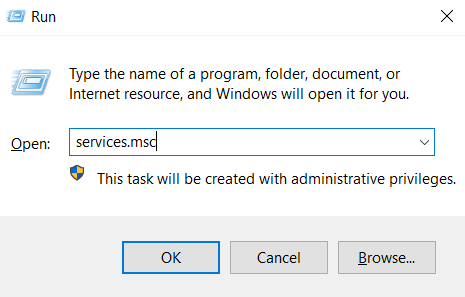
*IMPORTANT 2: Free disk space should be more than 15% of the total.*



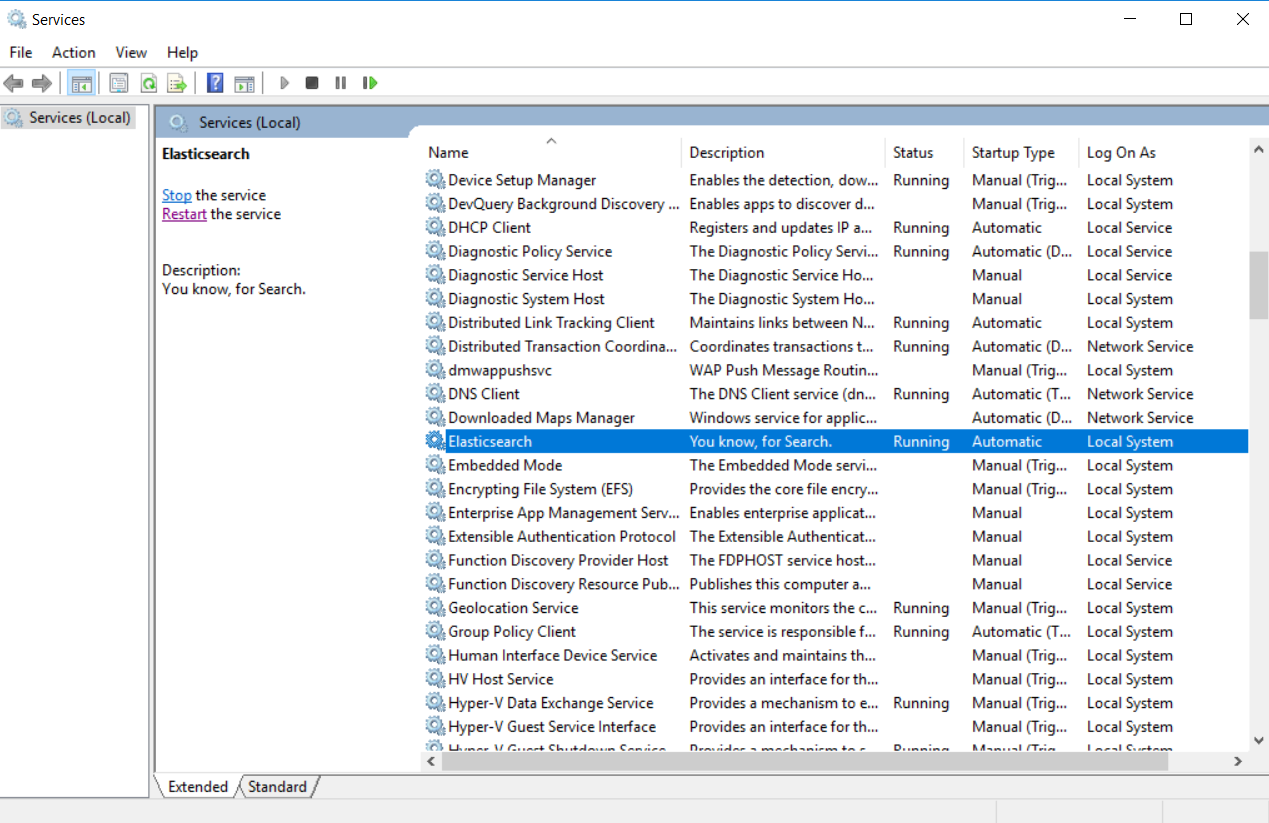
*Note: 1.75GB of memory is at least of requirements for Elasticsearch, divide more memory for Elasticsearch to work comfortably with search service.*

Continue the installation, accepting the default configuration. We use Elasticsearch only for searching so at the end accept the Basic License of Elasticsearch.

Check if Elasticsearch service is running on your machine in services like this: Press Win+R buttons you can see Run task window, enter “services.msc”

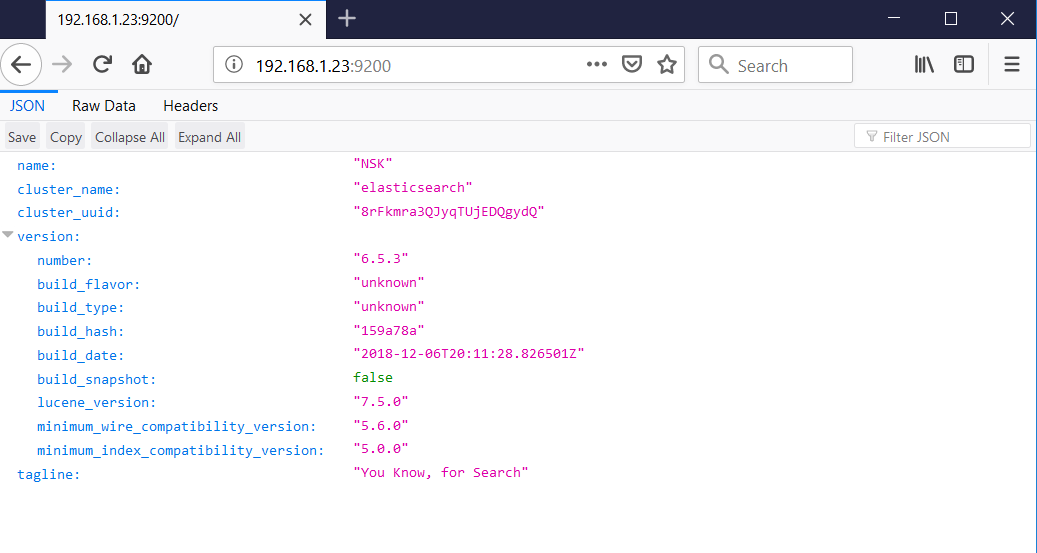


The Services window will open, find Elasticsearch and ensure its service status - it should be in Running and Startup Type should be Automatic.



*Note: if while the site is running, a search of statistical units is not working and there are code 500 errors from the site, you need to check if Elasticsearch service is in Running mode (It gets taken out by Windows if you didn't add a path to jre or jdk in Environment variables).*

Elasticsearch service availability can be checked by entering the elastic host address in the browser <http://localhost:9200>



# SQL Wallet setup

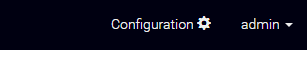
## Installation

1. Download the latest version of program from the official site *sqlwallet.com* (at time of writing this manual, the is version 1.3.1). We install according to the proposed step by step instructions. We need a security password.
2. Download SQLite ODBC Driver ()

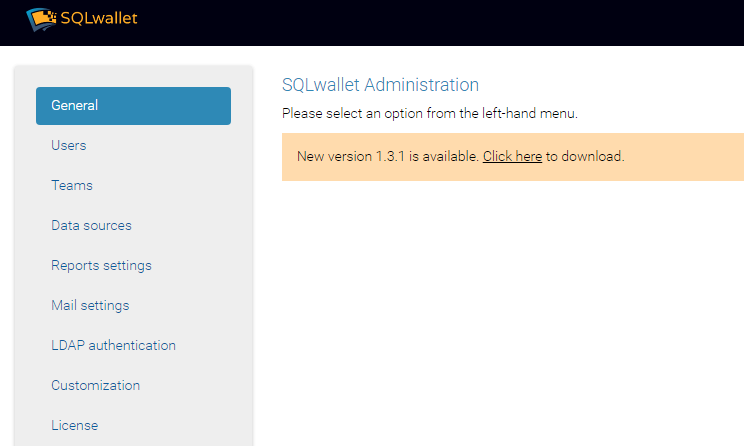
**NOTE:** *Check “ServerHostName” in appsettings.json file at C:\Program Files (x86)\SQLwallet\SQLwallet directory. It should have “localhost” value or IP address of the machine in the network. To change this file please use Administrator permission. After changing you need to stop SQLWallet service at services.msc and then start, not just restart.*

### Creating a connection string

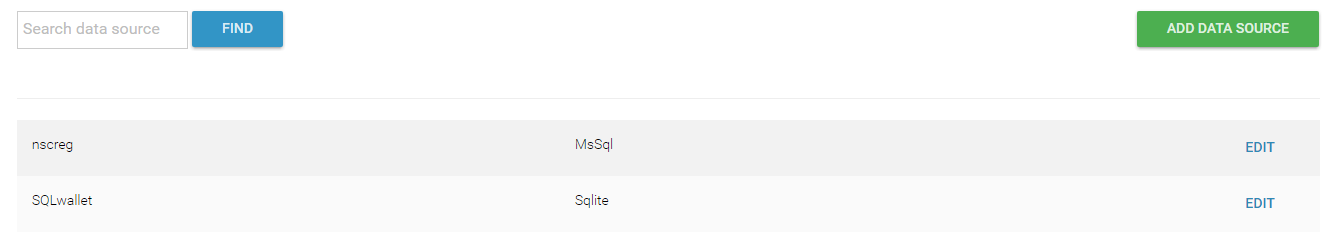
After installation go to site and click on link in the upper right corner of Configuration:



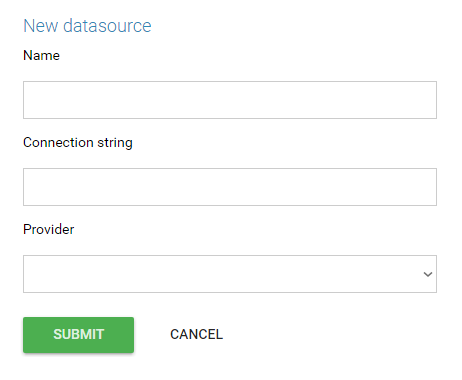
We get  into the SQLwallet configuration menu, all possible settings will be available to the left:



Basically we will be interested in 2 items of this menu, these are Users and DataSources. The first step is to create a connection string to our database, so go to DataSources and the following window will open:



Here we already have a connection string for the NSC, but now we will look at how to create a new one by clicking on ADD DATA SOURCE in the upper right corner, the following window will open:



Enter the name, connection string and select the provider.

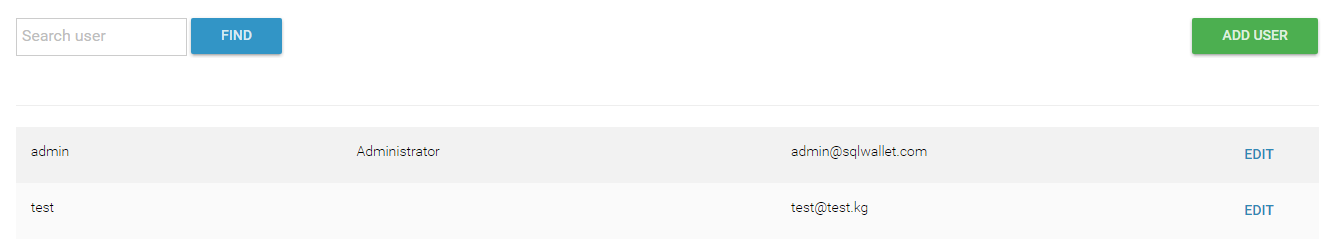
Server=servername;Database=nscreg;User Id=sa;Password=myPassword

*IMPORTANT: every provider has different connection strings! Check configuration strings examples in Database management system configurations chapter.*

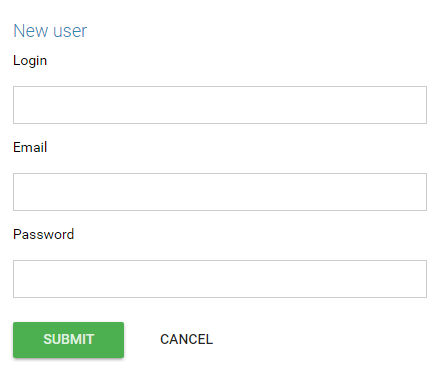
When everything is done press SUBMIT and that's it, we have the connect string.

## Creating Users

Next, we need users, so in menu on the left side click on Users and the following window will open:



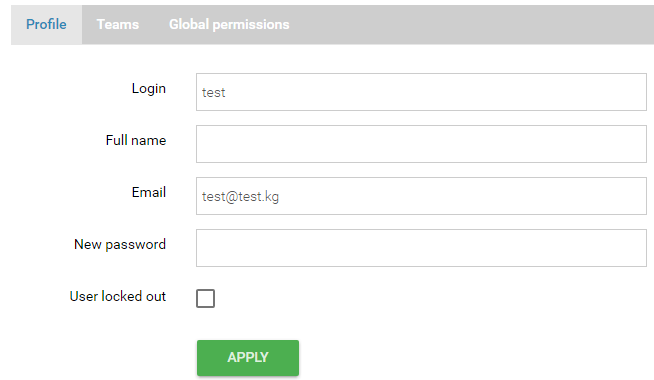
Here we already have one test user, but now we'll see how to create a new one, for this we click on **ADD USER** and the following window will open:



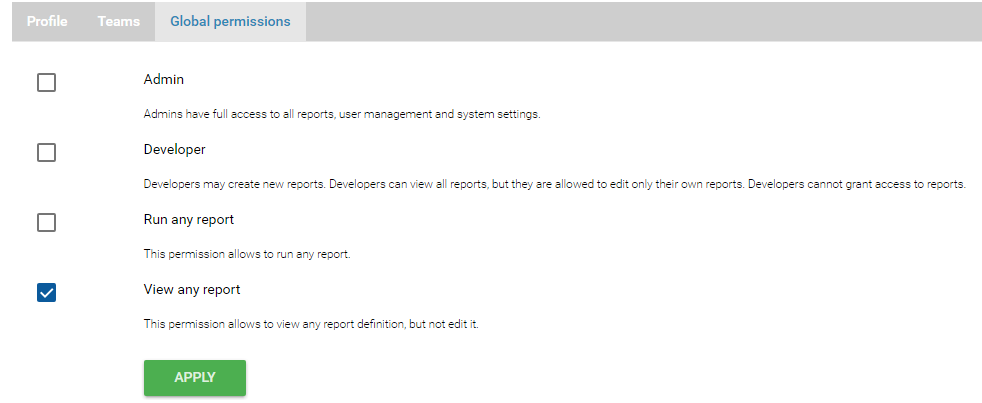
Enter requisites and  click **SUBMIT**.

**Setting the rights**

After the user is created, we need to configure his rights, so in the list of users right next to the user we are interested in, click on EDIT, we will see the following window:

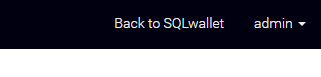


We are interested in the last **Global permissions tab**, click on it and the following window will open:



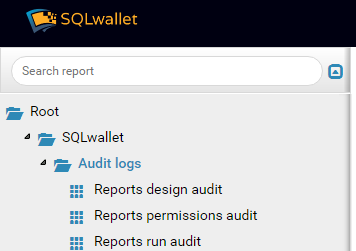
Of all the proposed "roles" only 2 are suitable- **View any Report** and **Admin**. There will be only **2** users in system: one with the role of viewing reports, the other with their creation and indicating the rights to this report. Users of **NSC** site at the moment have 3 roles: **Admin**-has the full right to edit and view the entire site, **NSC Employee** can read everything and can set up what can be edited and **External User** does not have the right to edit, but the view option is editable. The access to reports will be denied for **External User**. The **NSC Employee** user group will have access to the user in SQLwallet with rights **View any Report** - i.e. all users of **NSC** system will go to the reporting system under one user **SQLwallet**, which will have the right  only **to View any Report**. The **Admin** group of **NSC** site will be correlated with user in **SQLwallet** system with **Admin** rights. Therefore, only **admins** of **NSC** site will be able to create reports and give out the rights to reports, now we will see  how to do it.

To assign rights to the report, first it should be created, so we leave the configuration menu, for this we click on“Back to SQLwallet” link



## Creating a report and configuring report rights

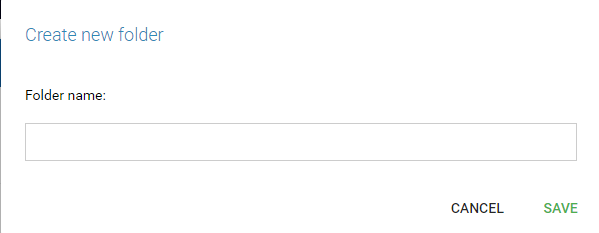
To the left there will be a report tree, initially there is only one **SQLwallet** folder, which stores pre-installed reports for monitoring the launch of reports and for changing user rights:



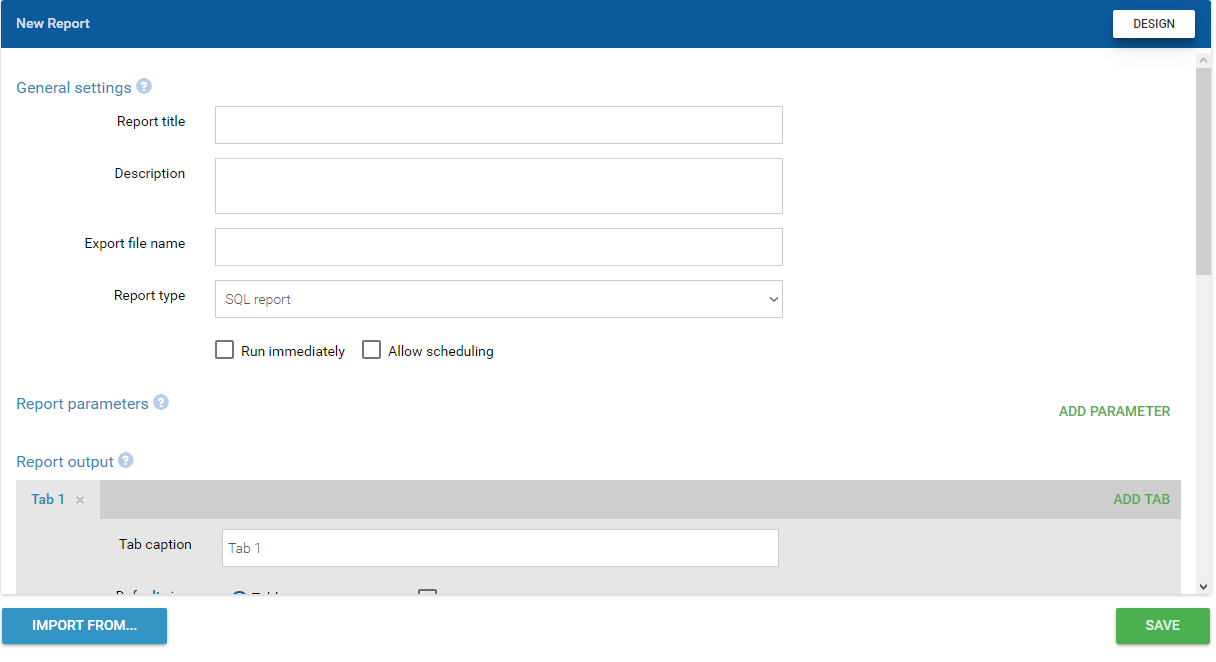
For conveniently building our reports, we need to create a new folder, I have it called **test**, to create it, we need to click on **NEW FOLDER** on the upper right:

https://lh3.googleusercontent.com/8UKKBT-dzy9XYZ6HbuGiJsq2NdXmarrBerYicv7oTe3aNFgEmJ0_fjdci2wTHKW7dRorFKsON_AzNkK_lKcBYVRp68mjLRW59fslI5kJfJrbJL7z6LhngEUJvAxk5xrGqyKhaTU

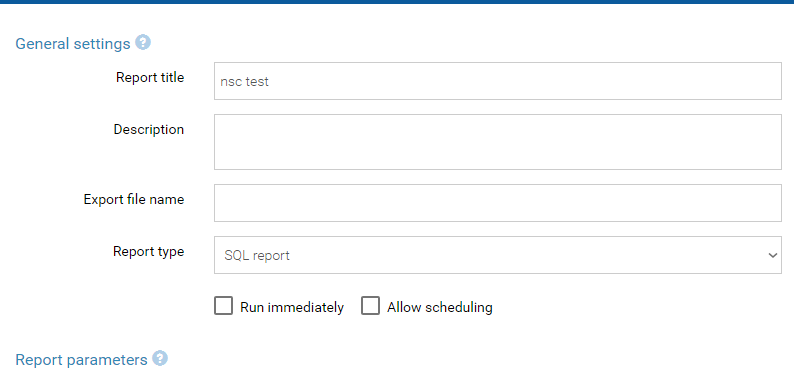
When you click on **NEW FOLDER**, a pop-up window pops up, in which we need to enter the folder name and press **SAVE**

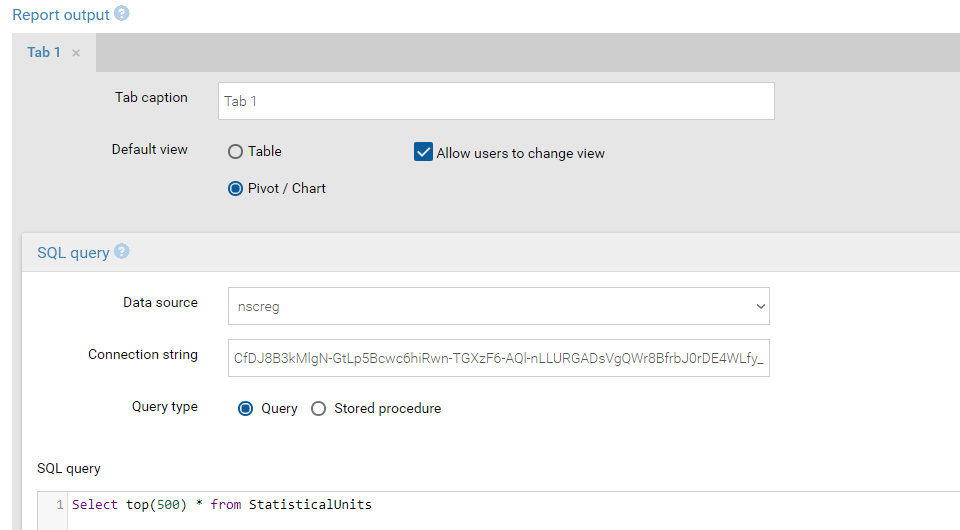


After we have created a new folder, we click on it to the left and select the item **NEW REPORT** at right, the following window will open:



This is the report designer, here we can make various settings that are intuitive, now we will create the simplest report, for this we should fill in the **Report title** field, select the connection string and write the simplest SQL query:

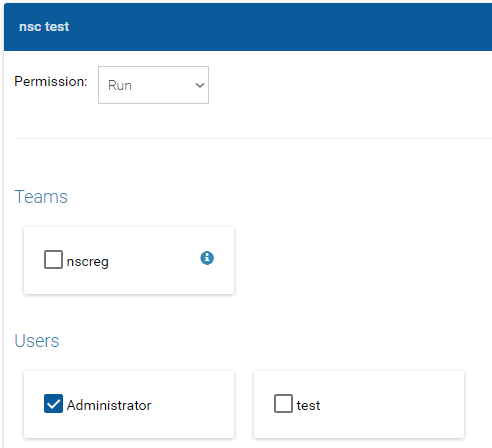




click the  SAVE. The report is ready, now we have menu of managing this report (upper right corner):

https://lh4.googleusercontent.com/Xln8GsT7e9J0pO5JaaE5CzJOThqcCuQN3EQtWWs8PRN2q9Xp8JmzjvUSwIG7RCF0r3yjzBcIM-XJBv4vLQj-P5KSLZoL8j0nQzkK4pISA-CEz36l5-qdlq7BihDWFKWJuTWCpD8

Now we are interested in PERMISSIONS item, click on it and see the following window:

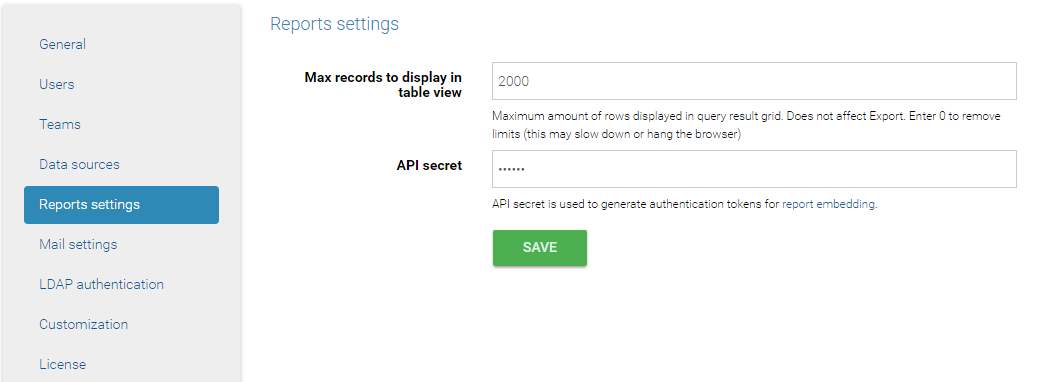


Here we see a list of access to the report (drop-down menu), commands and users. For the test user, we need to give access for Run, View, DataImport. To do this, simply select the required resolution from the drop-down list and tick the user's test. When all necessary permissions are selected, just click on SAVE (right bottom corner)

Further, when we log in as user test, we will have an access to test nsc report in the report

## Configuring the secret key in SqlWallet

For using the reporting engine from external systems (such as ours), we need to configure the secret key in SqlWallet system for access token obtaining. To do this, go to SqlWallet configuration and click on Reporting settings:



And in API secret field we enter our secret - it can be any word, but it should not be forgotten.

## Configuring the Linked Server

Since the **SqlWallet** system does not allow you to receive the report tree via the API. To get the report tree from **SqlWallet**, we need to create a link server to SqlWallet database, write a stored procedure that will go through link server in SqlWallet database behind the report tree and use it all.

The first thing you need to do is install the **ODBC** driver for **Sql Lite** Step-by-step instruction on how to install the driver is located at this link Creating a SQL Server Linked Server to SQLite to Import Data   (<https://www.mssqltips.com/sqlservertip/3087/creating-a-sql-server-linked-server-to-sqlite-to-import-data/>)

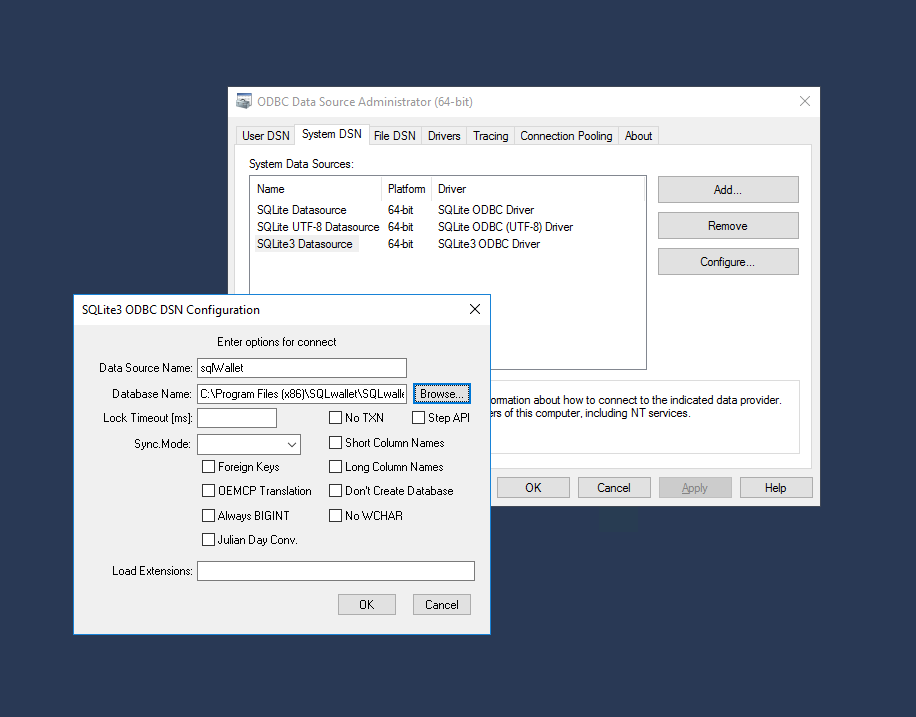
One example of how it can be done:

start “odbc data sources (64 bits)”

(this can be started by pressing windows button and then just start typing the name of the program)

chose the “System DSN tab” and “add”

this will pop up this window:



enter data source

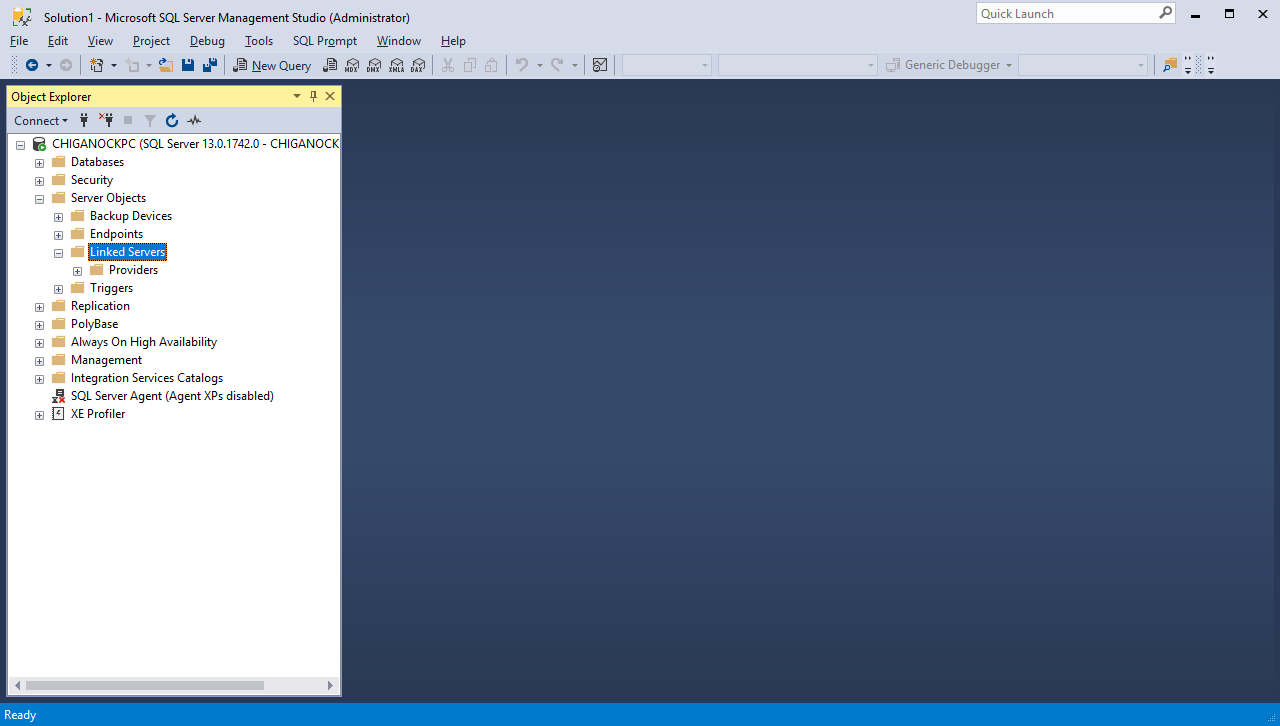
name: sqlWallet

database name: browse to the file sqlwallet.s3db

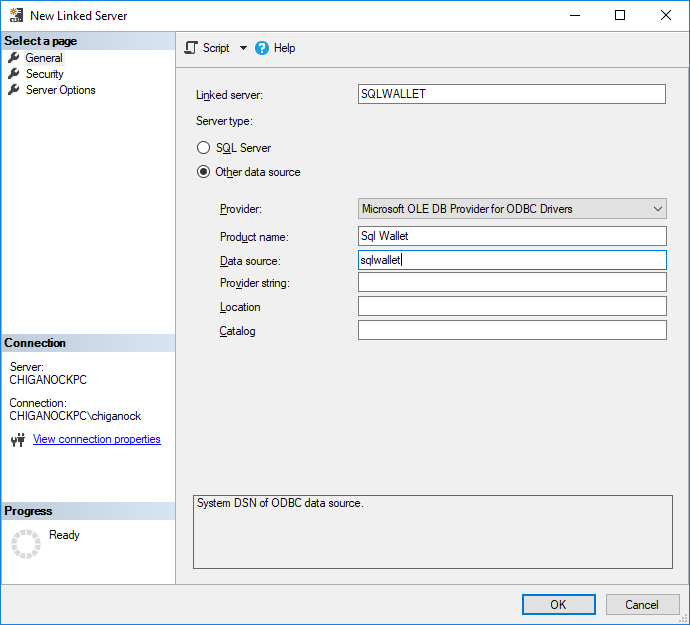
(typically found at C:\Program Files (x86)\SQLwallet\SQLwallet\App\_Data))

press ok until it all disappears.

By link above the linked server is created through stored procedures using, now we will figure out how to do this, I use the management studio. Go to **Microsoft SQL Server Management Studio**, connect to our database, then expand the **Server Object** folder and right click on **Linked Servers** folder:



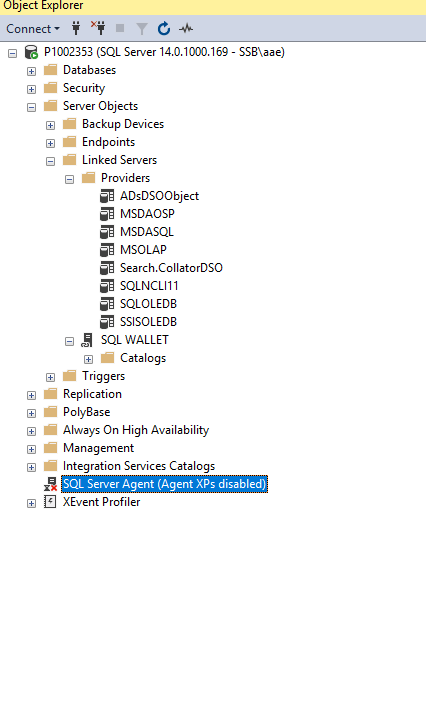
In resulting context menu select **New Linked Server ...** The following window will open:

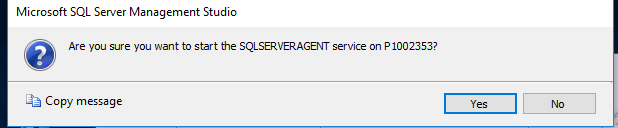


In Provider it is necessary to choose what is at picture above, in Product name we write the product name, and in the Data source field we must write the name of  connection we created when installed the ODBC driver for SqlLite.

*Note that the Linked server name has to be the same as in the reportsettings area in the appsettings.Production.json file*

After that, right click on SQL Server Agent (Agent XPs disabled) and press start:



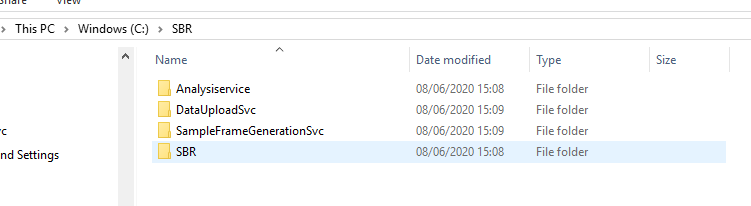


# Deployment of SBR System

There are 4 main programs and services that need to run:

* The web site itself
* DataUploadService – for uploading data from files
* AnalysisService – for analysis of data quality
* SampleFramesService . for creating sample frames

Each of these resides in separate folders, it is recommended to create a common folder for them, say something like this:



## The web site

### Unpack the SBR.zip to destination folder

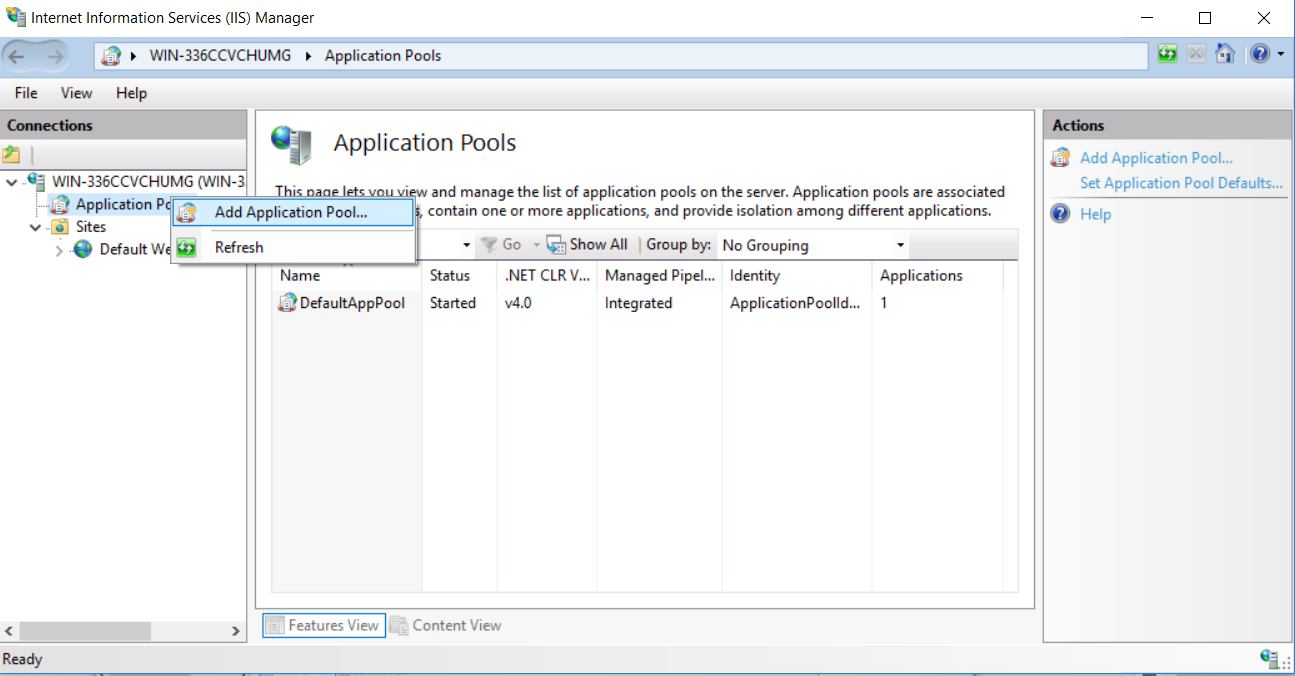
### Steps of deploy 1: Enable IIS Service

1. Navigate to Control Panel => Programs => Programs and Features => Turn Windows features on or off.
2. Open the group for “Internet Information Services” and “Web Management Tools”.
3. Check the box for “IIS Management Console”.
4. Check the box for “World Wide Web Services”.
5. Accept the default features for \_World Wide Web Services\_.

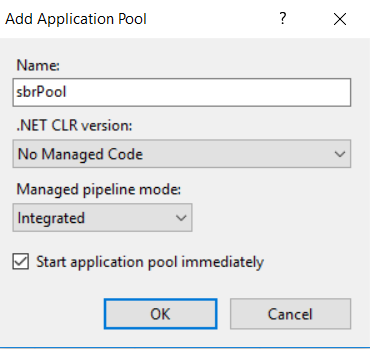
* more generalized instructions and error handling info is available in official documentation at [ASP.NET Core docs.microsoft.com]
  + (https://docs.microsoft.com/en-us/aspnet/core/publishing/iis)
* launching any executables, or even sign in, as \_Administrator account\_ is recommended during environment configuration

### Steps of deploy 2: Manual website configuration via IIS Manager

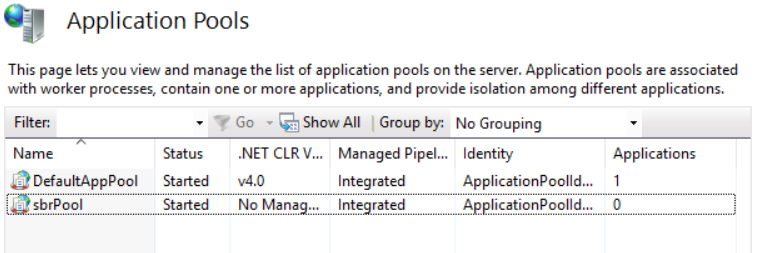
1. Place published project on your server machine(Example directory C:\SBR);
2. Open IIS Manager;
3. Create New Application Pool in left menu of IIS Manager;



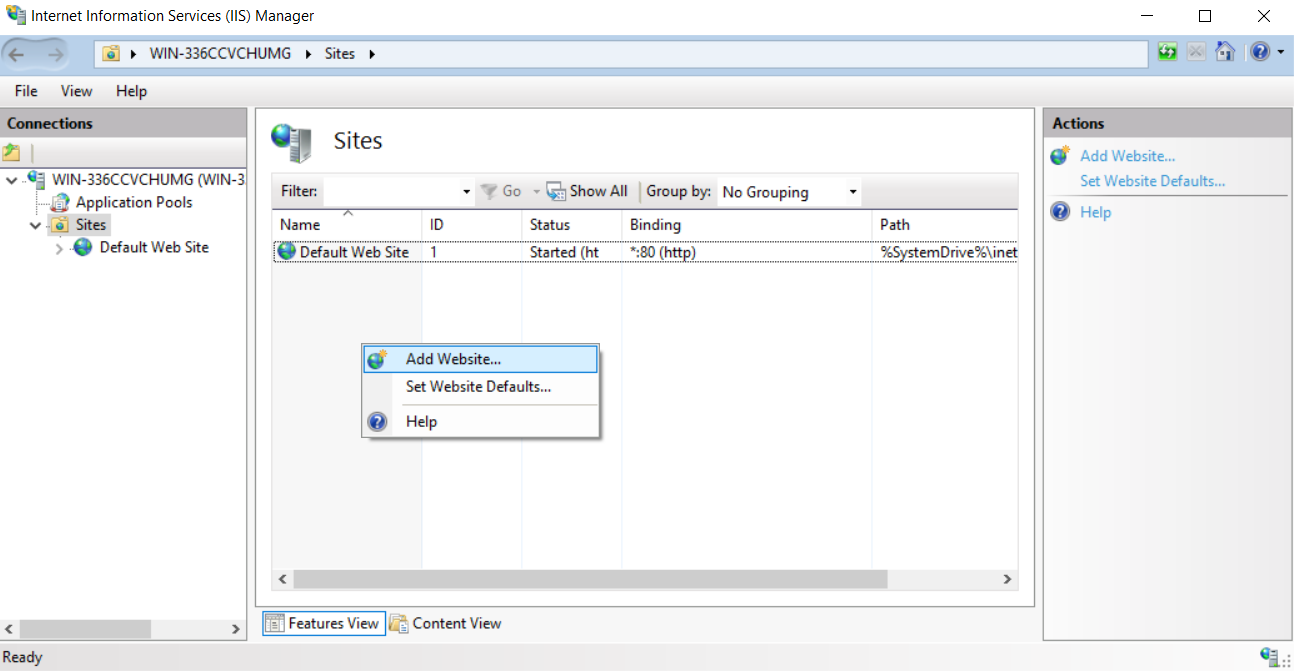
1. Set Application Pool name and in “.NET CLR Version” dropdown choose “No Managed Code” then click “OK” button;



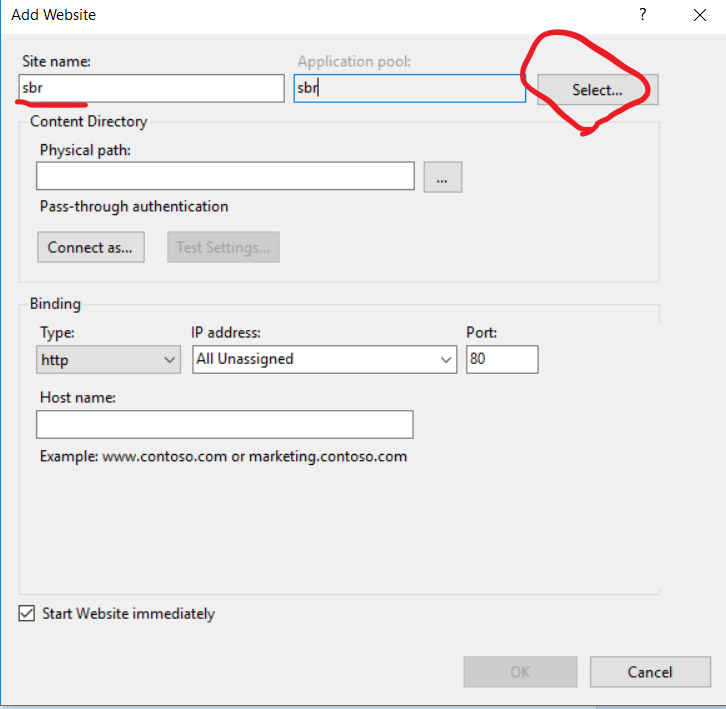
1. In Application Pools list you will see created pool;



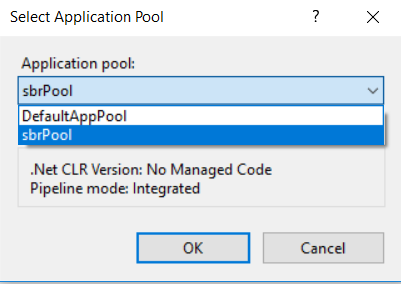
1. After creating Application Pool, need to add WebSite, right click to the Sites folder in left menu or click to the Sites folder you will get to the Sites folder right click and choose “Add Website”;



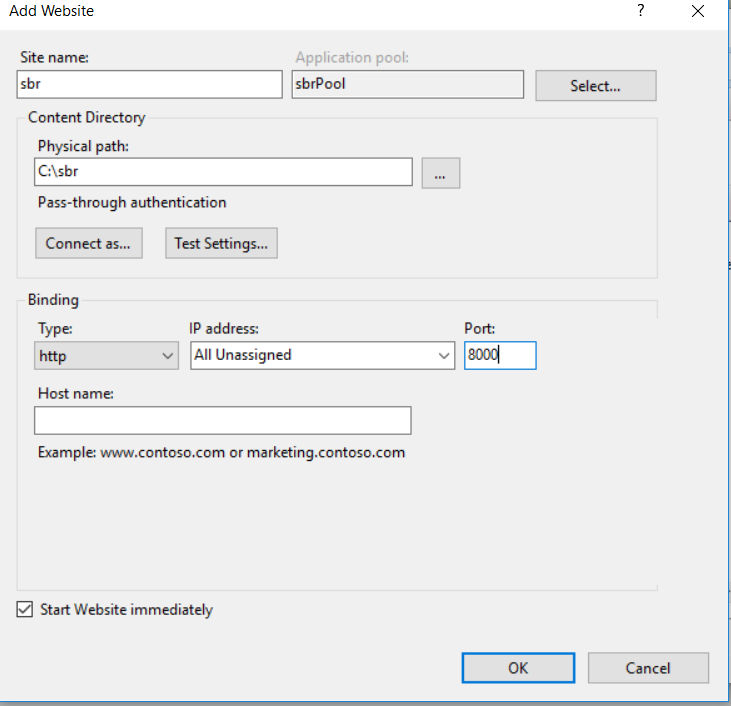
1. Enter needed Website name, and press Select button to select Application Pool;



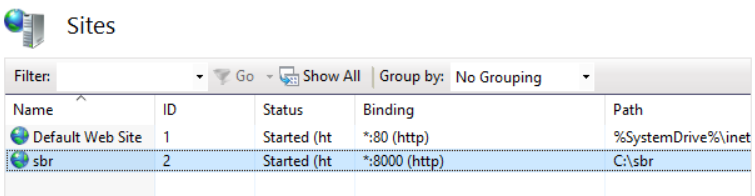
1. In appeared menu select Application Pool that you created before;



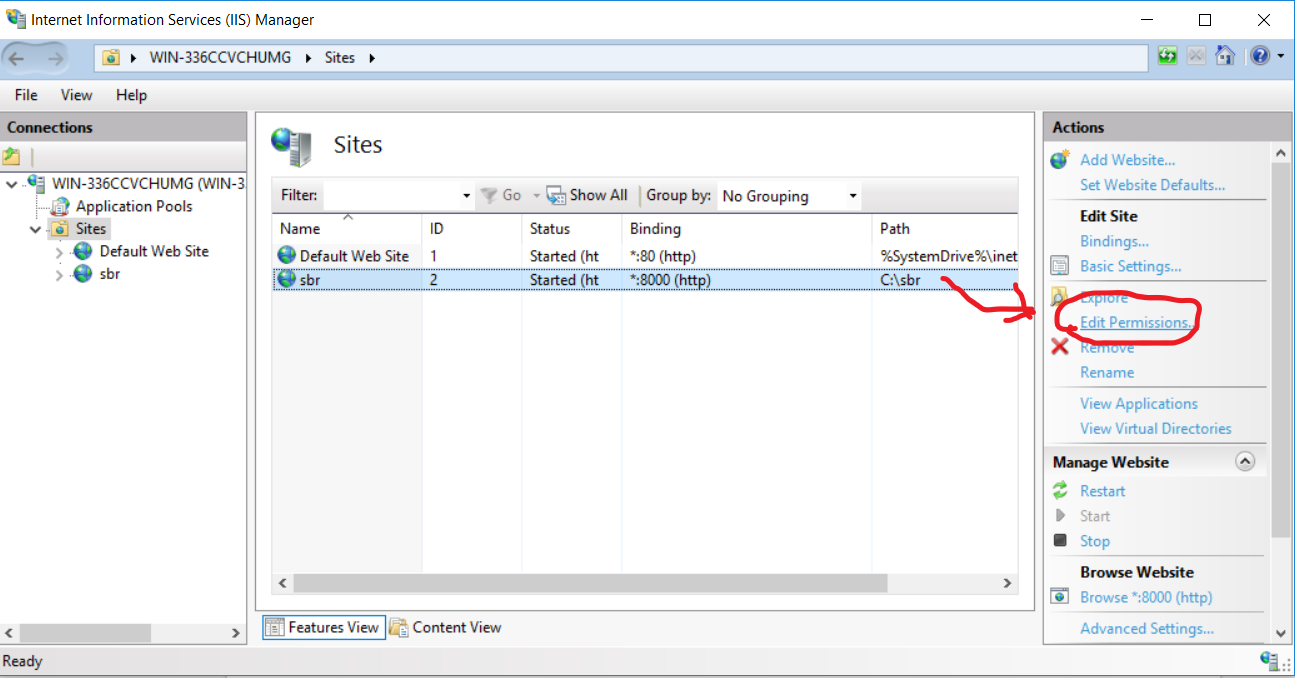
1. Set Physical Path to the project placed on your machine, assign needed Port and click OK;



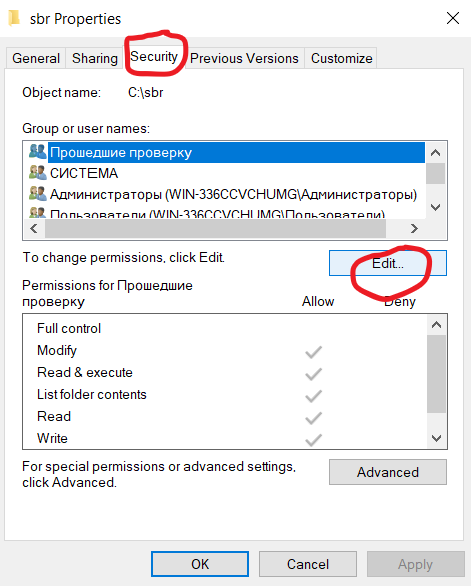
1. After adding website you can see it in Sites list below;



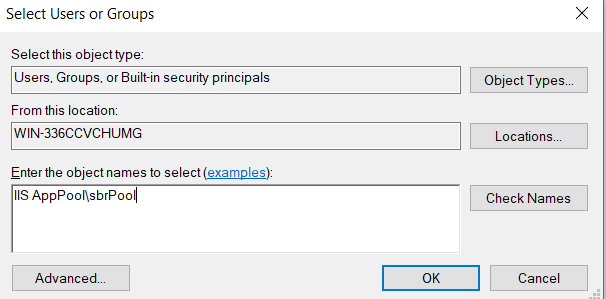
1. To ensure that created App Pool has appropriate permissions (read and write) on the Website host directory Click to the created website in right menu click “Edit Permissions”;



1. In appeared window of folder Properties go to Security tab and in Group and Names area click Edit button;

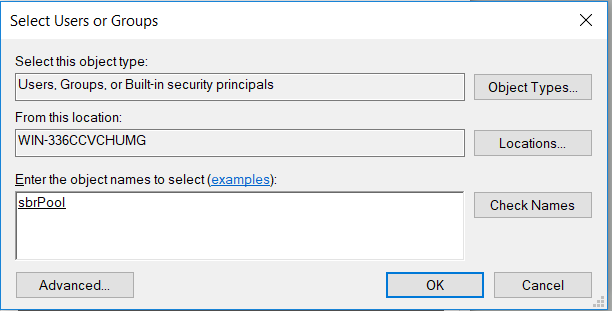


1. Click Add button to add AppPool in permissions. Enter IIS AppPool\\*your pool name\* there and click Check Names to see that AppPool is exists;



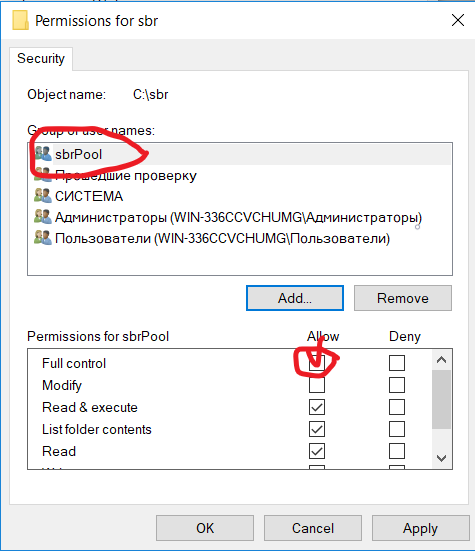
(If there is a problem here, make sure that “From this location” (in the figure above) is correct. It needs to be the computer, not the whole network)

Enter Name

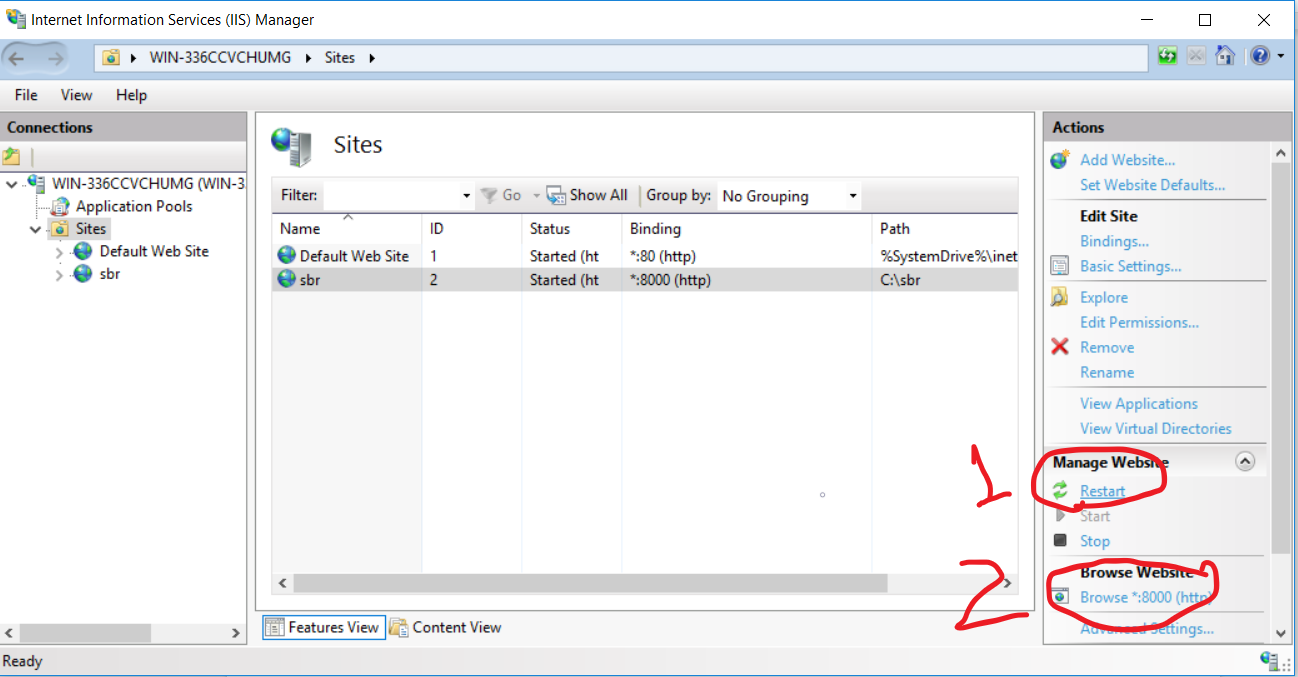


After Check Names button click

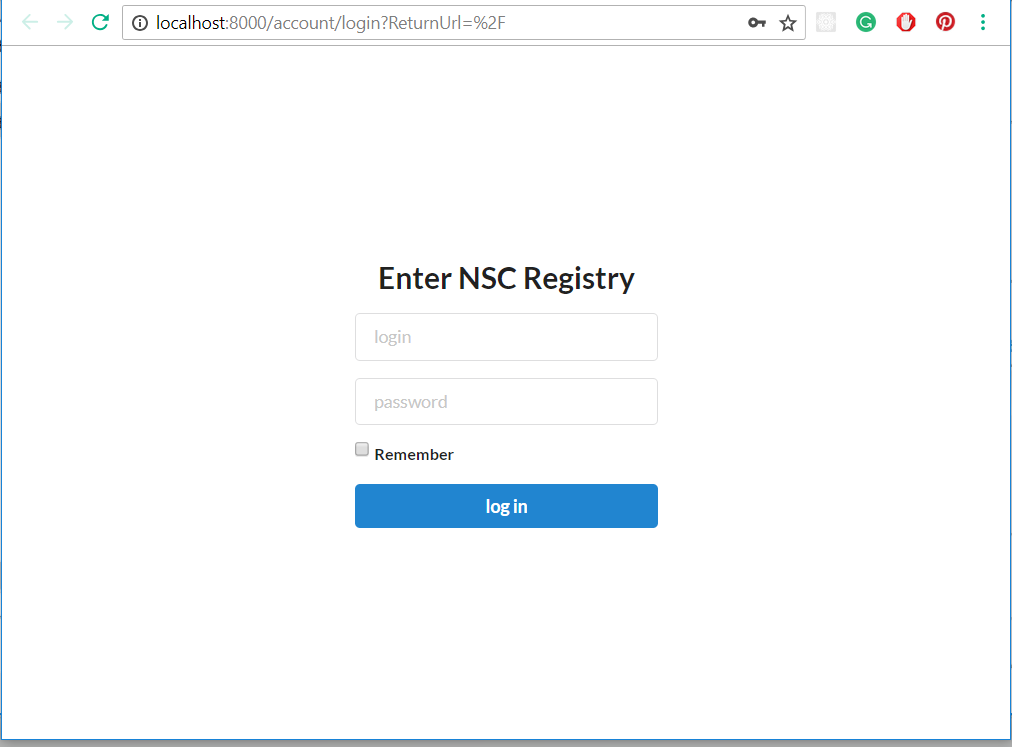
1. Now you can see in Group or user names list added name, allow all permissions and click Apply;



1. Edit the appsettings.Production.json file found in the root directory of the application so that the ConnectionSettings match the settings of your database; (More about appsettings file later)
2. Now click to Restart in right side menu and Browse to your site;



1. Check configuration file for the connection settings to the database in the config file appsettings.Production.json and restart the server;
2. Browse button will open site on your port;



Make sure you log in to the system, as this will generate the database to be populated with the business register data.

## Services Installation

Initially we had to start the services manually. Now in order to make the services to be started automatically, the services should be installed from PowelShell or CommandPrompt.

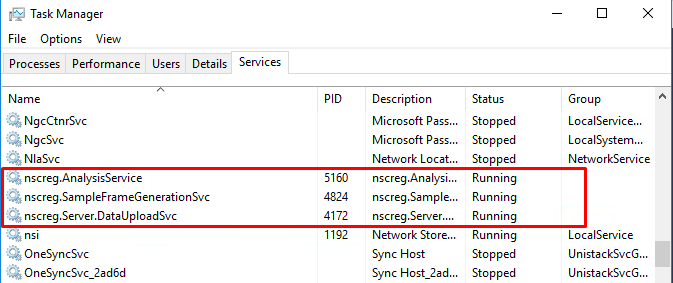
Unpack all of the services files to the destination folder

Steps to install the service

* + - 1. open PowelShell or CommandPrompt
      2. for each service need to run the command

dotnet "C:\*path to service folder*\service\_name.dll" action:install

* + - 1. try to restart the system or republish any of the services
      2. open Task Manager => go to Services tab => all services should be running



There is also SQL Server Agent service (required for SQL Wallet) that should be also started automatically. To configure this you need to set startup type in the properties of the service

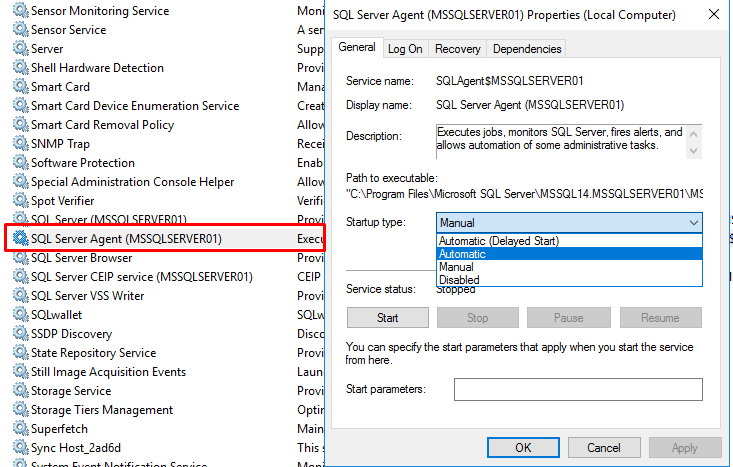
Steps to configure:

open Windows Services

select SQL Server Agent

right click Properties

select Startup type = Automatic



## ElasticSearch integration to SBR system

Elasticsearch is integrated to statistical units search functionality.

After successful installation of Elasticsearch you need to configure your system. Check appsettings JSON file: appsettings.Production.json file if you are in the published folder, or appsettings.Development.json if you are in developer mode.



Need to add ElasticServiceAddress and ElasticStatUnitSearchIndexName:

|  |
| --- |
| "ElasticServiceAddress": "http://localhost:9200",  "ElasticStatUnitSearchIndexName": "statunitsearchview" |

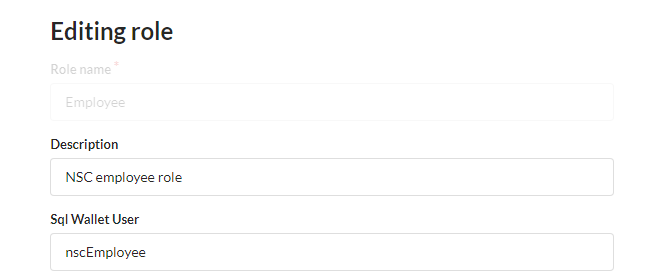
Where ElasticServiceAddress - HTTP address of elastic service that you configured at installation; and ElasticStatUnitSearchIndexName - the name of the index that will be created by Elasticsearch service and use it.

*Note: When you first-time run system and Elasticsearch, index of Elasticsearch is created by an initial search of statistical units; this may take time. For example for 5GB of data index creation can take around 20 mins.*

*Note: Elasticsearch has a response limit - it can analyse all data but can only return up to 10 thousand rows. So if you search statistical units enter more filters.*

## Integration of SQL wallet with SBR system

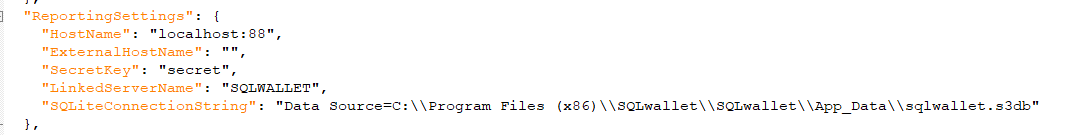
After all settings in SqlWallet are completed, it is necessary to integrate it with our system. To do this, in our system, in  role management there was added one more field Sql Wallet User in which we enter user login of SqlWallet system.



*Pic: SBR site - Administrative tools, Editing role*

For Administrator group in our system, we enter login of administrator of SqlWallet system, and for Employee group in our system, we enter there the user login of  SqlWallet system, which has rights only for viewing and executing reports. For External User group in our system, the Sql Wallet User field is not displayed.

Double check that the appsettings.Production.json has the right settings. Something like this:



# Appsettings files

More to come here… 😊

# Populating the register with data

More to come here too.. But for now: There is a demo database available on github (demoDB.bak) which can be used for now.