

mynevaTOgo Setup and Operation

[Microsoft + IIS]

Dokumentenversion 1.0 vom 6.5.2022

Inhalt

Inhalt	1
Architektur	2
Serverkomponente, Backend, myneva.carecenter.identity-server	2
Serverkomponente, Backend, myneva.carecenter.service	2
Benutzerschnittstelle, Frontend, myneva.carecenter.mobile App.....	2
Installation des Identity Servers.....	2
Installation und Inbetriebnahme der Serverkomponente.....	4



Architektur

Die myneva.carecenter.mobile Lösung besteht aus drei Teilen:

Serverkomponente, Backend, myneva.carecenter.identity-server

Diese Komponente dient zur sicheren Benutzeranmeldung. Der myneva.carecenter.identity-server kann CareCenter-Benutzer anmelden, indem er im Hintergrund auf die Benutzer-Tabelle der CareCenter-Datenbank zugreift, und Benutzernamen und Kennworthashes vergleicht.

Serverkomponente, Backend, myneva.carecenter.service

Diese Komponente ersetzt den bei der Vorgängerversion eingesetzten API-Gateway. Es handelt sich um eine REST API, welche den Zugriff auf die dahinterliegende, direkt angebundene CareCenter-Datenbank lesend und schreibend abstrahiert, und die Kommunikation mit der eigentlichen App ermöglicht. Dabei wird zur Datenübertragung in beide Richtungen der HL7 Fhir Standard in Version 4 verwendet.

Benutzerschnittstelle, Frontend, myneva.carecenter.mobile App

Diese Komponente folgt der bisher als CCMobile vertriebenen App nach. Es handelt sich um eine in Flutter realisierte Mobile App, die für das Android-Betriebssystem verfügbar ist, und in den Formaten APK und AAB bereitgestellt werden kann. Bevorzugt wird ein Hosting in einem Enterprise Mobile Management (EMM) Store des Kunden, wo der Kunde selbst die seiner Infrastruktur entsprechende Konfiguration einstellen kann.

Das Frontend wird wie bisher mit dem Backend verbunden, indem man in der Konfiguration der App die Url des Backend hinterlegt. Die kann am Log-In Screen im Bereich "Technische Informationen" durch tippen auf das Bleistift Symbol im Abschnitt der Server-Adresse gemacht werden. Es darf dabei kein Benutzer aktiv angemeldet sein (App wird z.B. aus Standby geöffnet)!

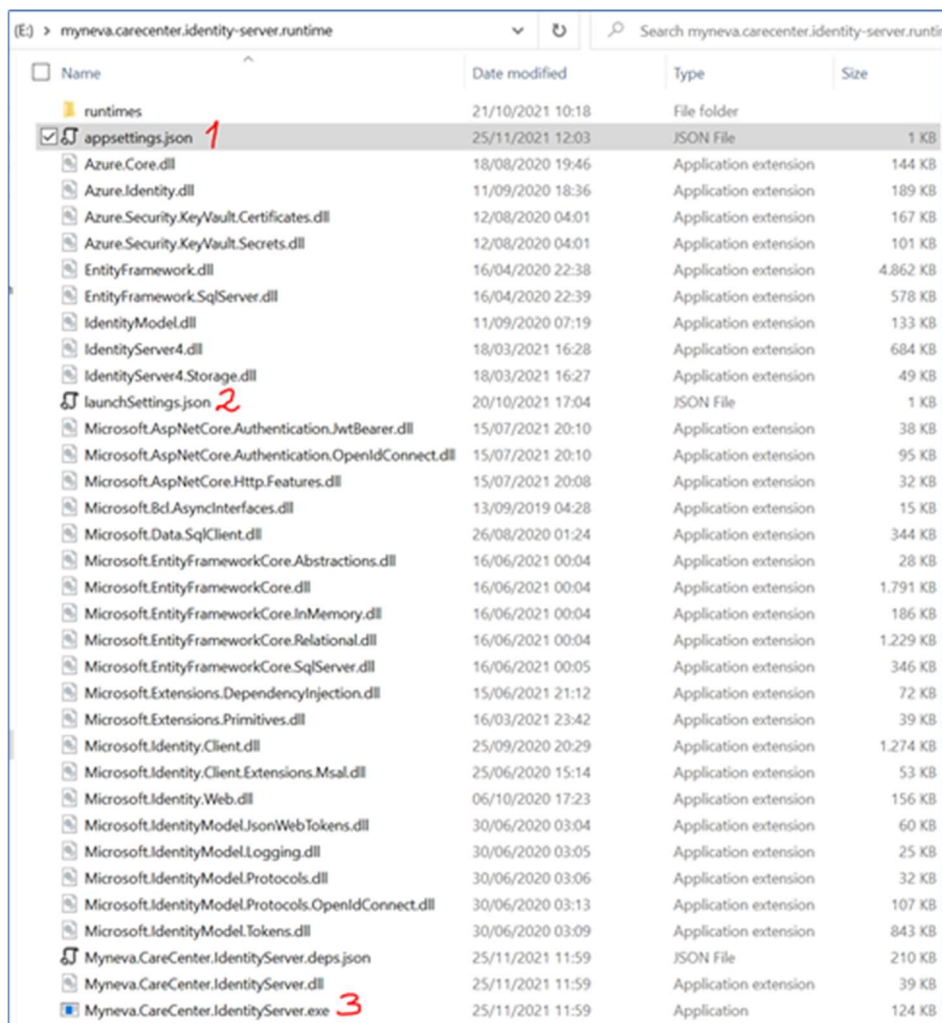
Installation des Identity Servers

Die zu installierende Version muss mit der betriebenen CareCenter-Datenbankversion (somit auch mit der Version der daran betriebenen CareCenter Desktop Anwendung) in den ersten zwei Teilen der Versionsangabe übereinstimmen: 4.3.12 Backend mit 4.3.5 Datenbank ist möglich, 4.5.0 Backend mit 4.2.7 Datenbank ist nicht möglich.

Anders als es Identity Server normalerweise tun, kommuniziert das Frontend (die App) hier nicht direkt, sondern nur über die eigentliche Serverkomponente mit den Anmeldediensten. Dies ermöglicht uns, den Identity Server gegen das Internet zu isolieren, was die mögliche

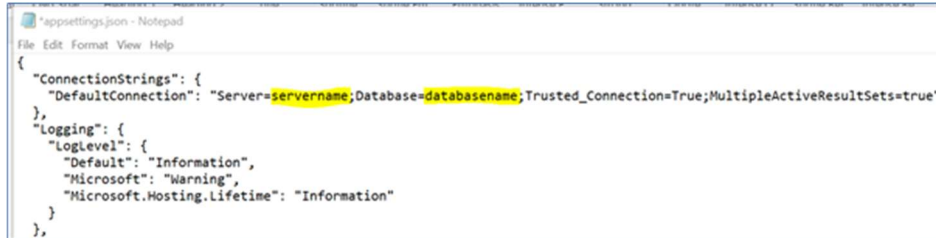
Angriffsfläche für Sicherheitslücken verringert. Konfigurieren Sie diesen Identity Server also jedenfalls so, dass er aus dem Internet nicht erreichbar ist – zum ordnungsgemäßen Betrieb muss er lediglich vom myneva.carecenter.service aus erreichbar sein.

1. Wählen Sie das zur Datenbankversion passende Installationspaket für die Distribution
2. Wenn Sie das Paket über das Internet auf die Kundenanlage transportiert haben, wählen Sie auf dem ZIP rechte Maustaste > Eigenschaften > Blockierung aufheben vor dem Entpacken.
3. Entpacken Sie das Installationspaket in ein leeres, lokales Verzeichnis auf einem lokalen Datenträger des Applikationsservers:



Name	Date modified	Type	Size
runtimes	21/10/2021 10:18	File folder	
appsettings.json 1	25/11/2021 12:03	JSON File	1 KB
Azure.Core.dll	18/08/2020 19:46	Application extension	144 KB
Azure.Identity.dll	11/09/2020 18:36	Application extension	189 KB
Azure.Security.KeyVault.Certificates.dll	12/08/2020 04:01	Application extension	167 KB
Azure.Security.KeyVault.Secrets.dll	12/08/2020 04:01	Application extension	101 KB
EntityFramework.dll	16/04/2020 22:38	Application extension	4.862 KB
EntityFramework.SqlServer.dll	16/04/2020 22:39	Application extension	578 KB
IdentityModel.dll	11/09/2020 07:19	Application extension	133 KB
IdentityServer4.dll	18/03/2021 16:28	Application extension	684 KB
IdentityServer4.Storage.dll	18/03/2021 16:27	Application extension	49 KB
launchSettings.json 2	20/10/2021 17:04	JSON File	1 KB
Microsoft.AspNetCore.Authentication.JwtBearer.dll	15/07/2021 20:10	Application extension	38 KB
Microsoft.AspNetCore.Authentication.OpenIdConnect.dll	15/07/2021 20:10	Application extension	95 KB
Microsoft.AspNetCore.Http.Features.dll	15/07/2021 20:08	Application extension	32 KB
Microsoft.Bcl.AsyncInterfaces.dll	13/09/2019 04:28	Application extension	15 KB
Microsoft.Data.SqlClient.dll	26/08/2020 01:24	Application extension	344 KB
Microsoft.EntityFrameworkCore.Abstractions.dll	16/06/2021 00:04	Application extension	28 KB
Microsoft.EntityFrameworkCore.dll	16/06/2021 00:04	Application extension	1.791 KB
Microsoft.EntityFrameworkCore.InMemory.dll	16/06/2021 00:04	Application extension	186 KB
Microsoft.EntityFrameworkCore.Relational.dll	16/06/2021 00:04	Application extension	1.229 KB
Microsoft.EntityFrameworkCore.SqlServer.dll	16/06/2021 00:05	Application extension	346 KB
Microsoft.Extensions.DependencyInjection.dll	15/06/2021 21:12	Application extension	72 KB
Microsoft.Extensions.Primitives.dll	16/03/2021 23:42	Application extension	39 KB
Microsoft.Identity.Client.dll	25/09/2020 20:29	Application extension	1.274 KB
Microsoft.Identity.Client.Extensions.Msal.dll	25/06/2020 15:14	Application extension	53 KB
Microsoft.Identity.Web.dll	06/10/2020 17:23	Application extension	156 KB
Microsoft.IdentityModel.JsonWebTokens.dll	30/06/2020 03:04	Application extension	60 KB
Microsoft.IdentityModel.Logging.dll	30/06/2020 03:05	Application extension	25 KB
Microsoft.IdentityModel.Protocols.dll	30/06/2020 03:06	Application extension	32 KB
Microsoft.IdentityModel.Protocols.OpenIdConnect.dll	30/06/2020 03:13	Application extension	107 KB
Microsoft.IdentityModel.Tokens.dll	30/06/2020 03:09	Application extension	843 KB
Myneva.CareCenter.IdentityServer.deps.json	25/11/2021 11:59	JSON File	210 KB
Myneva.CareCenter.IdentityServer.dll	25/11/2021 11:59	Application extension	39 KB
Myneva.CareCenter.IdentityServer.exe 3	25/11/2021 11:59	Application	124 KB

- Öffnen Sie die Datei (1) appconfig.json mit einem Texteditor (Notepad), und tragen Sie den Namen des SQL-Servers und den Namen der CareCenter-Datenbank ein:

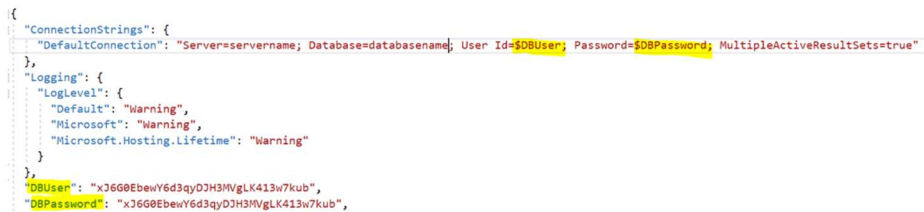


```

{
  "ConnectionStrings": {
    "DefaultConnection": "Server=servername;Database=databasename;Trusted_Connection=True;MultipleActiveResultSets=true"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  }
}

```

Die Verbindung zum SQL-Server kann entweder über den Windows-User (Trusted_Connection=True) aufgebaut werden, oder über einen SQL-User. In zweiten Falle können Benutzername/Passwort als Klartext im Connectionstring oder verschlüsselt als Parameter angegeben werden:



```

{
  "ConnectionStrings": {
    "DefaultConnection": "Server=servername; Database=databasename; User Id=$DBUser; Password=$DBPassword; MultipleActiveResultSets=true"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Warning",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Warning"
    }
  },
  "DBUser": "x36G0EbewY6d3qyDJH3MvGK413w7kub",
  "DBPassword": "x36G0EbewY6d3qyDJH3MvGK413w7kub",
}

```

\$DBUser sowie \$DBPassword im Connectionstring werden mit den entschlüsselten Werten der Parametern DBUser/DBPassword ersetzt

- Starten Sie den IIS Manager und fügen Sie unter Application Pools einen neuen Application Pool hinzu. Legen Sie eine neue Seite mit dem entsprechenden Verzeichnis und Application Pool an. Wenn https verwendet werden soll, fügen sie unter Bindings das entsprechende Zertifikat hinzu. Starten Sie die Seite.

Im Browser können Sie mit [endpunkt]/.well-known/openid-configuration überprüfen ob der Identity Server läuft, bzw eine Verbindung zur Datenbank aufbauen kann.

Installation und Inbetriebnahme der Serverkomponente

Auch hier muss die zu installierende Version mit der betriebenen CareCenter-Datenbankversion (somit auch mit der Version der daran betriebenen CareCenter Desktop Anwendung) in den ersten zwei Teilen der Versionsangabe übereinstimmen.

- Wählen Sie das zur Datenbankversion passende Installationspaket für die Distribution
- Wenn Sie das Paket über das Internet auf die Kundenanlage transportiert haben, wählen Sie auf dem ZIP rechte Maustaste > Eigenschaften > Blockierung aufheben vor dem Entpacken.
- Entpacken Sie das Installationspaket in ein leeres, lokales Verzeichnis auf einem lokalen Datenträger des Applikationsservers.
- Öffnen Sie die Datei appsettings.json mit einem Texteditor (Notepad):



```

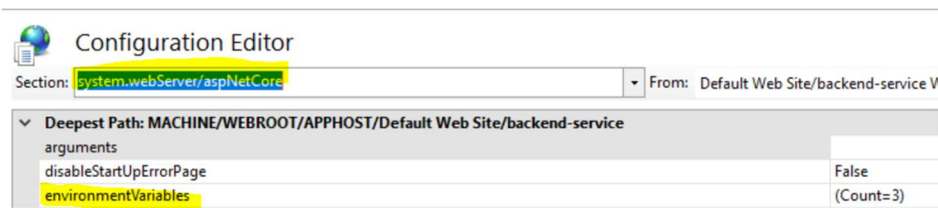
https://json.schemastore.org/appsettings.json
1 {
2   "DatabaseProvider": "SqlServer",
3   "ConnectionStrings": {
4     "DefaultConnection": "Server=servername;Database=datenbankname;Trusted_Connection=True;
5   },
6   "Logging": {
7     "LogLevel": {
8       "Default": "Warning",
9       "Microsoft": "Warning",
10      "Microsoft.Hosting.Lifetime": "Warning"
11    }
12  },
13  "AllowedHosts": "*",
14  "IdentityApiEndpoint": "https://localhost:6001",

```

Stellen Sie hier den Namen des SQL-Servers, den Namen der Datenbank und den in Kapitel 2 definierten Endpunkt, auf dem der Identity Server läuft, ein. Speichern Sie die Änderungen (siehe auch Punkt 4. Identity Server für weitere Konfigurationsmöglichkeiten).

5. Starten Sie den IIS Manager und fügen Sie unter Application Pools einen neuen Application Pool hinzu. Legen Sie eine neue Seite mit dem entsprechenden Verzeichnis und Application Pool an. Wenn https verwendet werden soll, fügen sie unter Bindings das entsprechende Zertifikat hinzu. Starten Sie die Seite.

Wenn Sie den Identity Server ohne Verwendung von TLS (https) betreiben wollen, gibt es die Möglichkeit die Environment Variable `IDENTITY_REQUIRE_HTTPS_METADATA = 0` zu setzen. Wählen Sie dazu im Configuration Editor die Sektion: `system.webServer/aspNetCore` und fügen sie die Variable unter `environmentVariables` hinzu.



Im Browser können Sie mit [endpunkt] überprüfen ob die Serverkomponente läuft, bzw eine Verbindung zur Datenbank aufbauen kann.

Damit ist das Service bereit zum Verbinden mit der mobilen App.

Die von uns als Hersteller der Software offiziell empfohlene Variante ist der Betrieb beider Serverkomponenten (Identity Server und Service) als Docker Container.

