

Einrichtung vom ServiceInside

Der ServiceInside verwendet die Technologien vom .NET 6.0 Framework und ist auf einem Windows Server mit IIS (Internet Information Services) in einer DMZ zu hosten. Der ServiceInside darf keinen Zugriff zum Intranet haben. Darüber hinaus darf der ServiceInside nicht vom Internet erreichbar sein. Nur der ServiceOutside darf mit dem ServiceInside über das HTTPS Protokoll kommunizieren. Es ist wichtig, dass der ServiceOutside zuerst eingerichtet wird und erst danach kann der ServiceInside vollständig eingerichtet werden. Auf dem Server vom ServiceInside soll eine Freigabe mit einem lokalen Benutzer eingerichtet werden. In dieser Freigabe, sollen die Dateien vom LIS/MIRTH/etc... zur Steuerung der Backendservices gelesen und geschrieben werden. (S. Dokumentation über die LIS-Anbindung). Bitte ersetzen Sie die in der Doku verwendeten IP-Adressen/Hostnames und Domains mit Ihren echten Daten. Der Connection String zur MySQL-Datenbank und das Kennwort vom SMTP-Server sollen in appsettings.json verschlüsselt gespeichert werden. Für die Verschlüsselung der Daten in der appsettings.json verwenden Sie bitte die PasswortEncryptor.exe.

Empfohlene Konfiguration des Servers

CPU	>= 8 CPU
RAM	>= 16 GB
SSD	>= 512 GB
OS	Windows Server >= 2019
IIS	>= 10

Inhalt der ServiceInside1_0.zip Datei

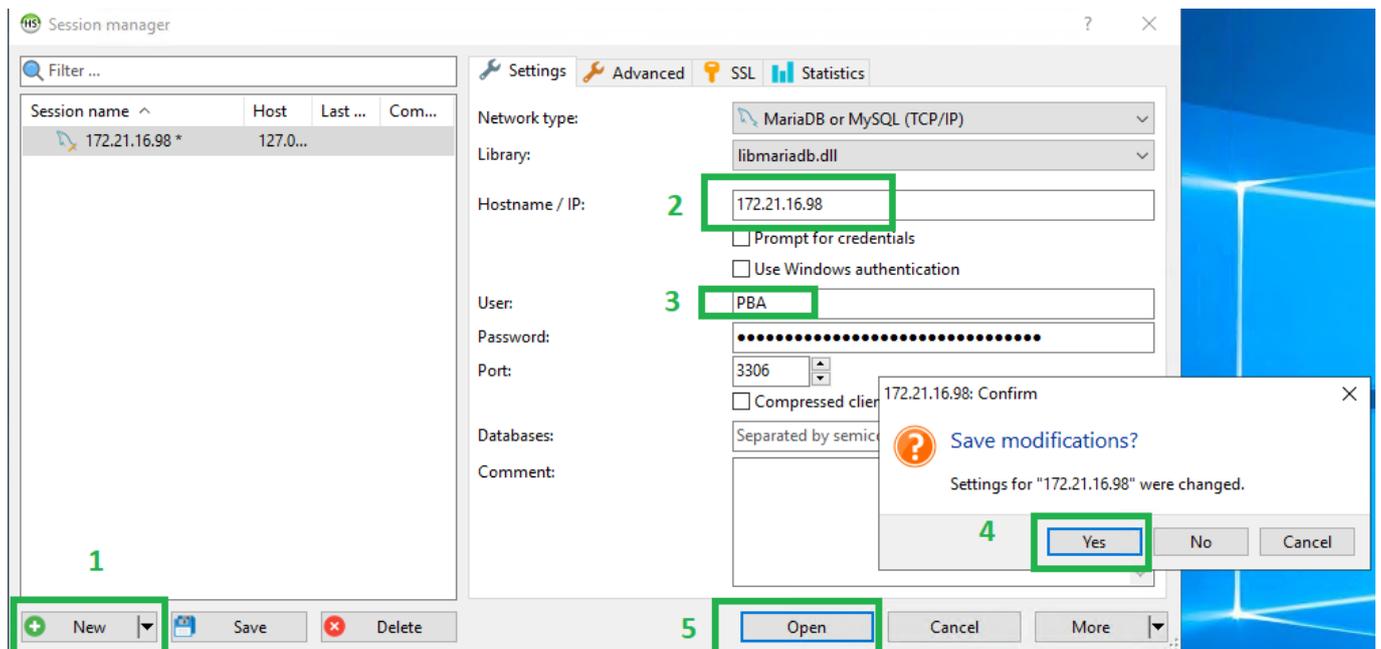
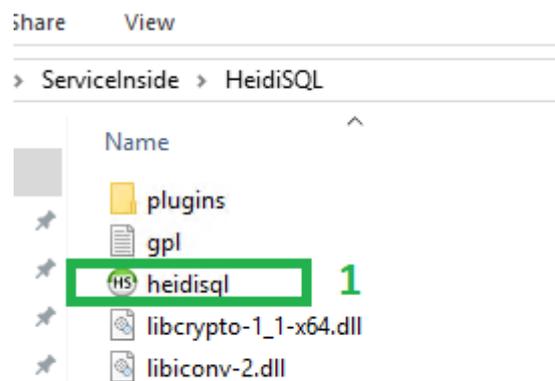
Die Zip Datei „**ServiceInside1_0.zip**“ enthält alle erforderlichen Dateien zum Installieren und Einrichten vom ServiceInside.

1. Das Verzeichnis „**Web**“ enthält die Dateien für den Webserver, der in IIS gehostet wird.
2. Das Verzeichnis „**HeidiSQL**“ enthält eine Anwendung „**heidisql.exe**“ mit der man die Datenbankverbindung zum MySQL Server testen kann.

3. Das Verzeichnis „**Postman**“ enthält eine Anwendung mit der man die Installation und Einrichtung vom ServiceInside testen kann.
4. Das Verzeichnis „**NET 6**“ enthält die Installationsdatei vom .NET Framework 6.0.8

Installation und Einrichtung vom Webserver

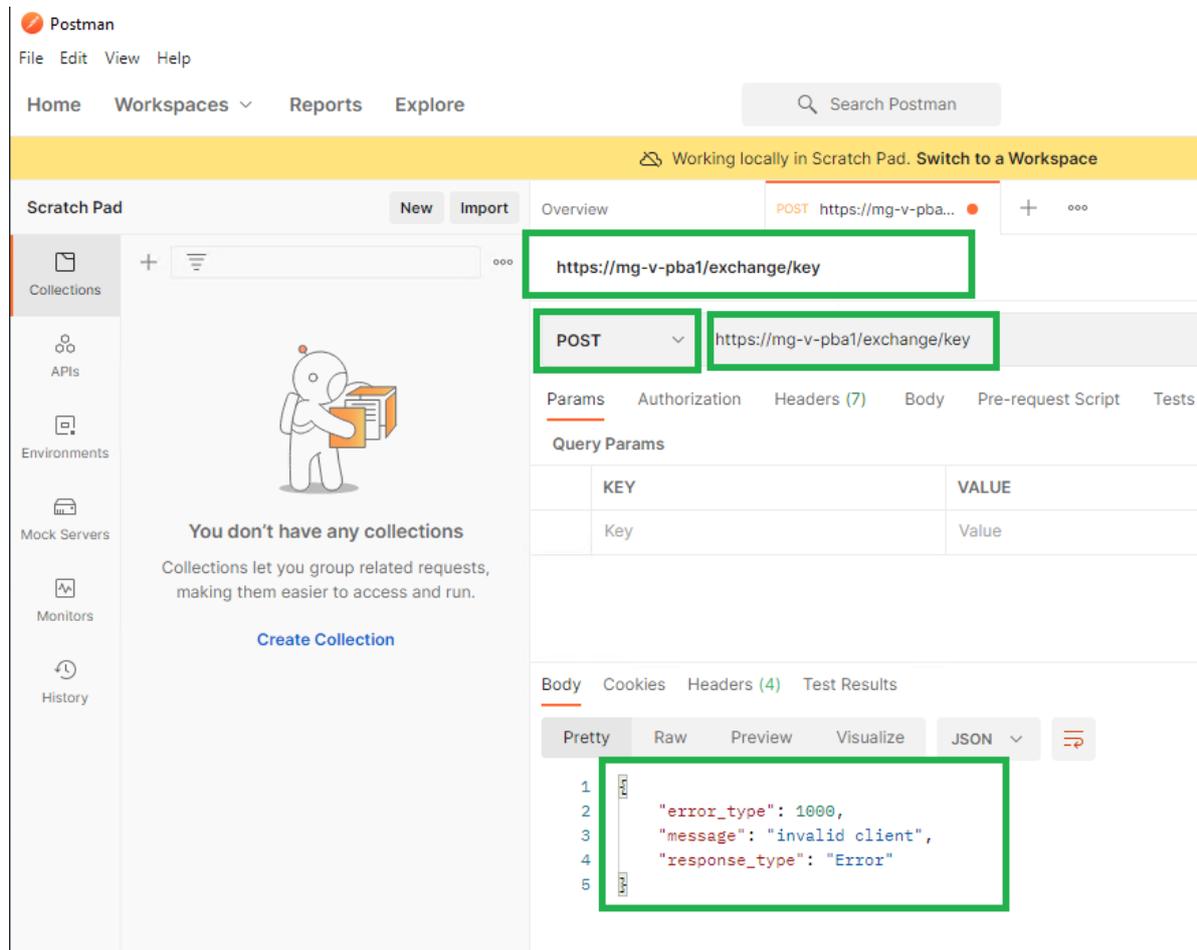
Bevor man mit der Installation und Einrichtung vom Webserver anfängt soll zuerst geprüft werden, ob der ServiceInside die MySQL-Datenbank vom ServiceOutside über die IP Adresse und Port erreichen kann. Zum Testen der Datenbankverbindung führen Sie bitte die Anwendung „**heidisql.exe**“ im Verzeichnis „**HeidiSQL**“ aus und verbinden Sie sich mit dem MySQL Server. (Der ServiceOutside muss zuerst eingerichtet werden)



Als nächstes prüfen Sie bitte, ob der ServiceOutside über das HTTPS Protokoll vom ServiceInside erreichbar ist. Dafür führen Sie bitte die Anwendung „**Postman.exe**“ im Verzeichnis „**Postman**“ aus und führen Sie anschließend in „**Postman**“ die HTTPS POST-Abfrage mit der URL „Hostname vom ServiceOutside“/exchange/key aus:

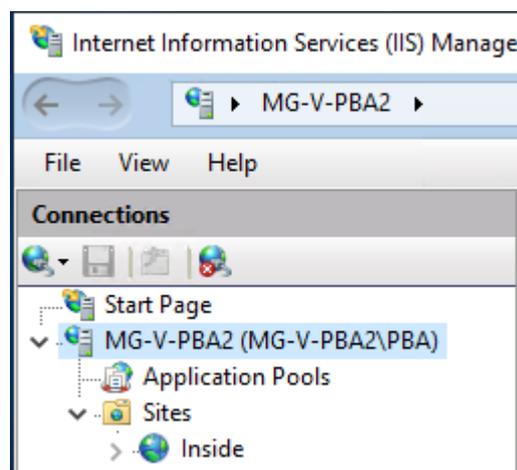
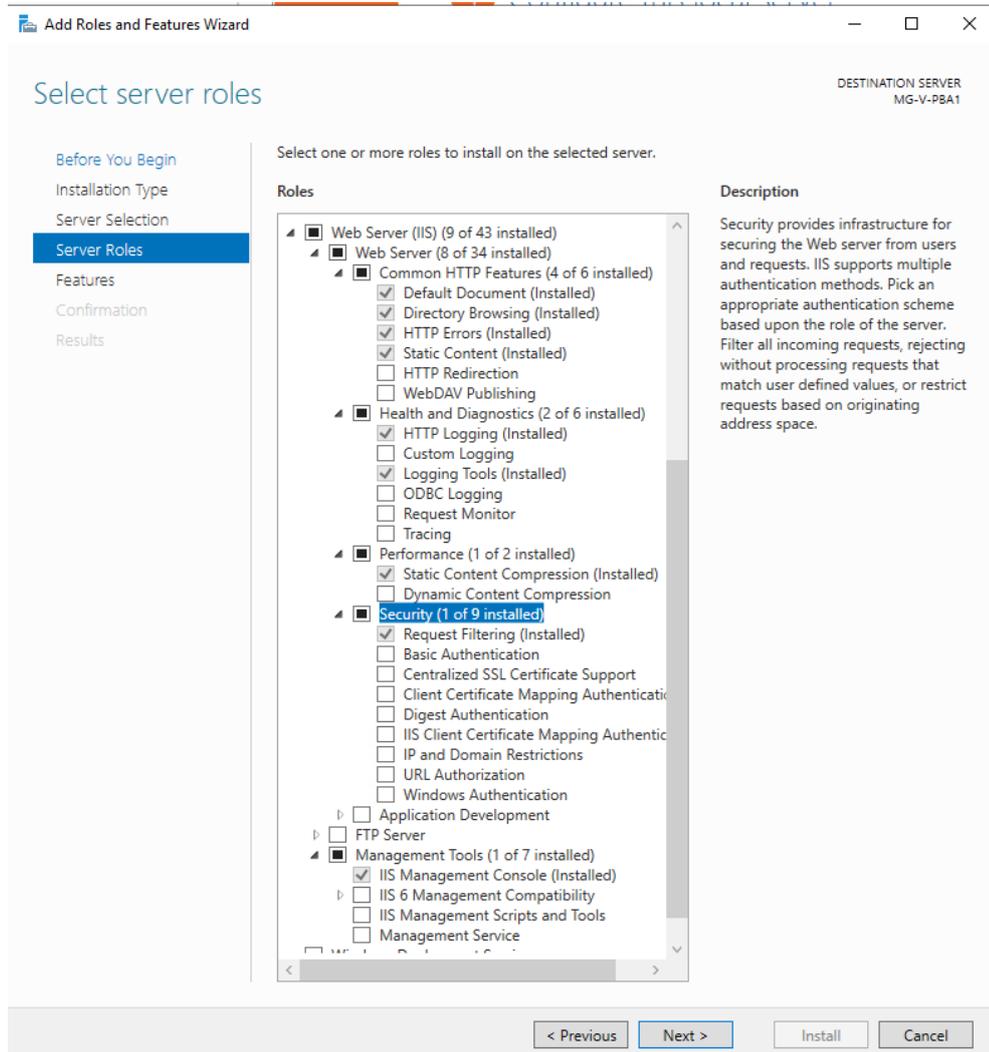
z.B: Hostname vom ServiceOutside: mg-v-pba1 (Nur intern erreichbar)

Auszuführende URL: mg-v-pba1/exchange/key

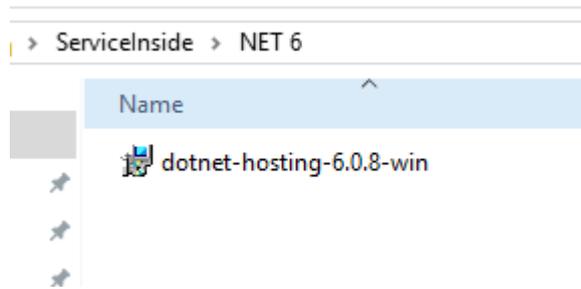


Meldet sich der ServiceOutside mit der Antwort `error_type: 1000 „invalid client“`, heißt es, dass die Verbindung zum ServiceOutside erfolgreich war und es kann nun mit der Installation und Einrichtung vom ServiceInside begonnen werden.

1. Aktivieren Sie nun den Webserver (IIS) in „Roles and Features“ vom System und rufen Sie anschließend den IIS Manager auf. Falls in IIS eine Default Site bereits vorhanden ist, löschen Sie diese bitte zuerst, bevor Sie mit der Konfiguration vom ServiceInside fortsetzen.



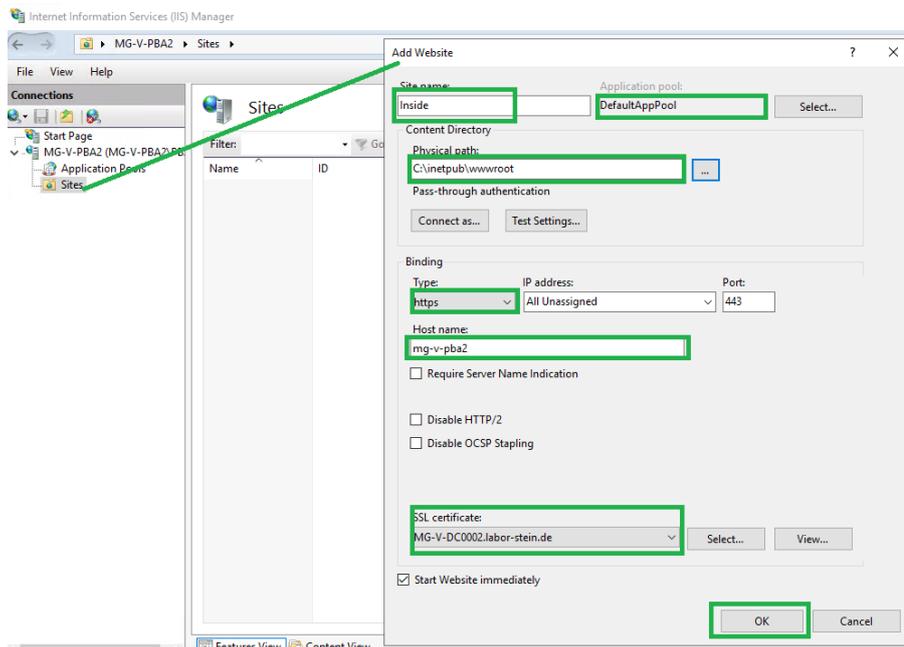
2. Als nächstes führen Sie bitte die Installationsdatei „**dotnet-hosting-6.0.8-win.exe**“ im Verzeichnis „**NET 6**“ aus und installieren Sie das .NET Framework 6.



3. Kopieren Sie die Dateien aus dem Verzeichnis „**Web**“ nach **C:\inetpub\wwwroot**.

4. Legen Sie nun eine neue Seite im IIS Manager an

- a: Geben Sie als Name der Seite „**Inside**“ ein.
- b: Wählen Sie als „**Application Pool**“ „**DefaultAppPool**“ aus.
- c: Wählen Sie als physikalischen Pfad das Verzeichnis „**C:\inetpub\wwwroot**“ aus.
- d: Wählen Sie als Standard Binding das https Protokoll (443) aus und tragen Sie als Hostname den Namen des Servers ein.



Wichtig: Es muss ein gültiges Zertifikat verwendet werden.

5. Für die Verschlüsselung der Befunde und die Steuerung des LIS, verwendet den ServiceInside einen Hintergrundprozess, welcher auf Dauer aktiv bleiben muss. Um die Unterbrechung dieses Prozesses zu vermeiden, passen Sie bitte den „**Start**

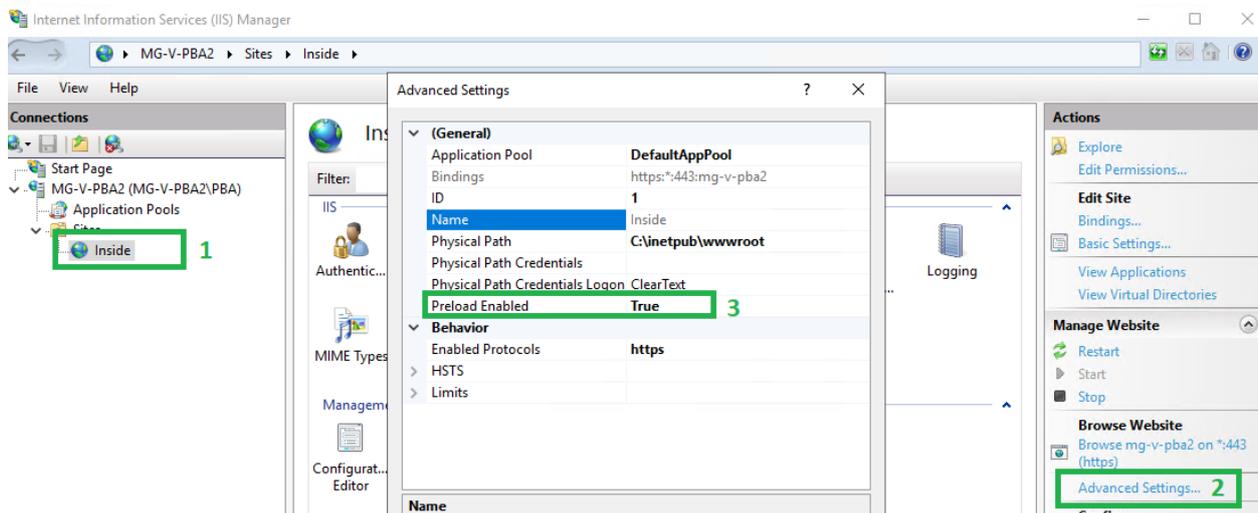
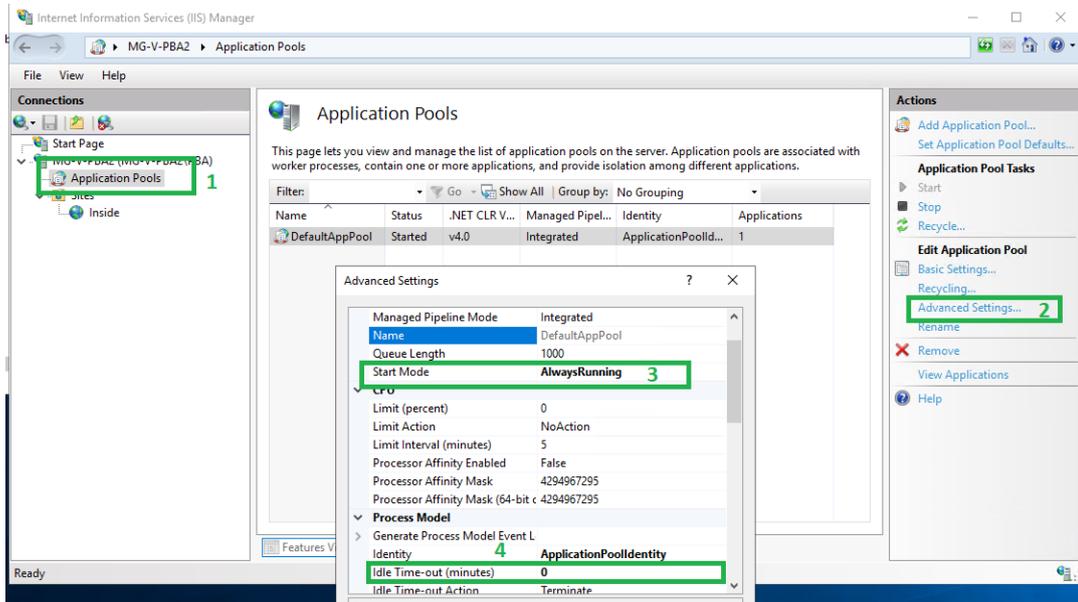
Mode“, die **„Idle Time-Out“**, **„Preload Enabled“** und den **„autoStart“** in IIS-Einstellungen entsprechend an:

Start Mode: AlwaysRunning.

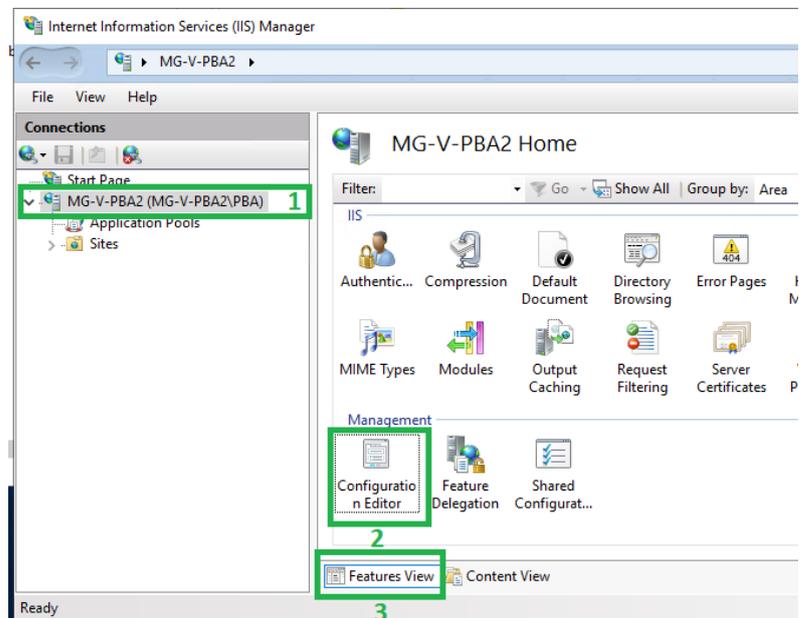
Idle Time-out: 0

Preload Enabled: True

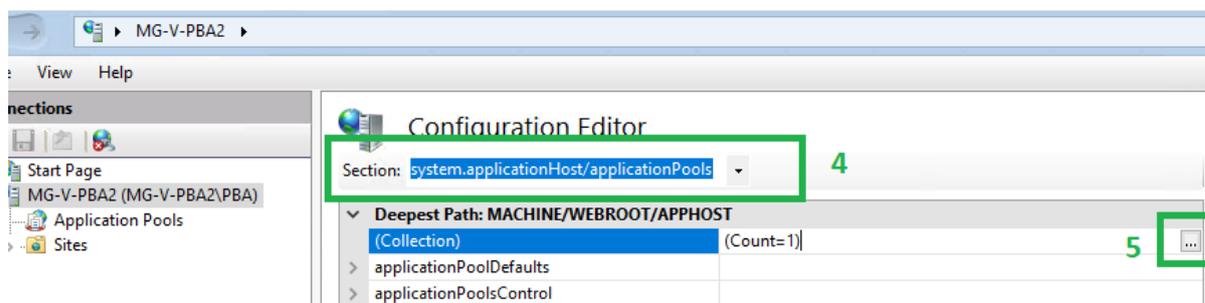
autoStart: True



Im **„Configuration Editor“** muss noch den **„autoStart“** aktiviert werden:



Internet Information Services (IIS) Manager



Collection Editor - system.applicationHost/applicationPools/

Items:										
name	queueLength	autoStart	enable32BitAppOnWin64	managedRuntimeVersion	managedRuntimeLoader	enableConfigurationOverride	managedPipelineMode	CLRConfigFile	passAnonymous	
DefaultAppPool	1000	True	False	v4.0	webengine4.dll	True	Integrated		True	
Properties:										
applicationPoolSid	S-1-5-82-3006700770-424185619-1745488364-794895919-4004696415									
autoStart	True									
CLRConfigFile										
cpu										
enable32BitAppOnWin64	False									
enableConfigurationOverride	True									
environmentVariables	(Count=0)									
failure										
managedPipelineMode	Integrated									
managedRuntimeLoader	webengine4.dll									
managedRuntimeVersion	v4.0									
name	DefaultAppPool									
passAnonymousToken	True									
processModel										
identityType	ApplicationPoolIdentity									
idleTimeout	00:00:00									
idleTimeoutAction	Terminate									
loadUserProfile	False									
logEventOnProcessModel	IdleTimeout									
logonType	LogonBatch									
manualGroupMembership	False									
maxProcesses	1									
password										
pingingEnabled	True									
pingInterval	00:00:30									
pingResponseTime	00:01:30									
setProfileEnvironment	True									

- Öffnen Sie die „**appsettings.json**“ Datei im Verzeichnis „**C:\inetpub\wwwroot**“, um den ServiceInside vollständig zu konfigurieren.

This PC > Local Disk (C:) > inetpub > wwwroot

Name	Date modified	Type	Size
appsettings.json	8/11/2022 4:46 PM	JSON File	
AspNetCoreRateLimit.dll	3/12/2022 12:41 PM	Application extens...	
aspnetcore2_inprocess.dll	2/24/2022 9:50 AM	Application extens...	
BinaryCache.Create.dll	2/10/2021 10:04 PM	Application extens...	

7. Tragen Sie in den Feldern „ServiceOutsideIP“ und „ServiceOutsideURL“ die IP und die URL vom ServiceOutside ein

```

j,
"CleanUpDays": 30,
"PrivateDirectory": "c:\\pba\\private",
"PublicDirectory": "c:\\pba\\public",
"MaxWorkerThreads": 10,
"ServiceOutsideIP": "172.21.16.98",
"ServiceOutsideURL": "https://mg-v-pba1",

```

8. Für die Datenbankverbindung tragen Sie bitte in der Konfigurationsdatei „appsettings.json“ unter „ConnectionString“ bei default „prod“ und im prod ersetzen Sie die Daten durch die echten Daten von Ihrem MySQL Server.

```

"ServiceInsideIP": "172.21.16.106",
"ServiceInsideURL": "https://172.21.16.106",
"ConnectionStrings": {
  "default": "prod",
  "prod": "Server=myServerAddress;Database=myDataBase;Uid=myUsername;Pwd=myPassword",
  "dev": "",
  "test": ""
}

```

Verschlüsselter ConnectionString

```

"ConnectionStrings": {
  "default": "dev",
  "prod": "XHQS8gr8XSwdVnoDZ04Ha/wZe10/XVr5wKc1TjcKe/Z3aW0MbqKQGZVhRTHXZkv4mkkxv9z3PK7p0IMAM3J+ta5d6kid8"
}

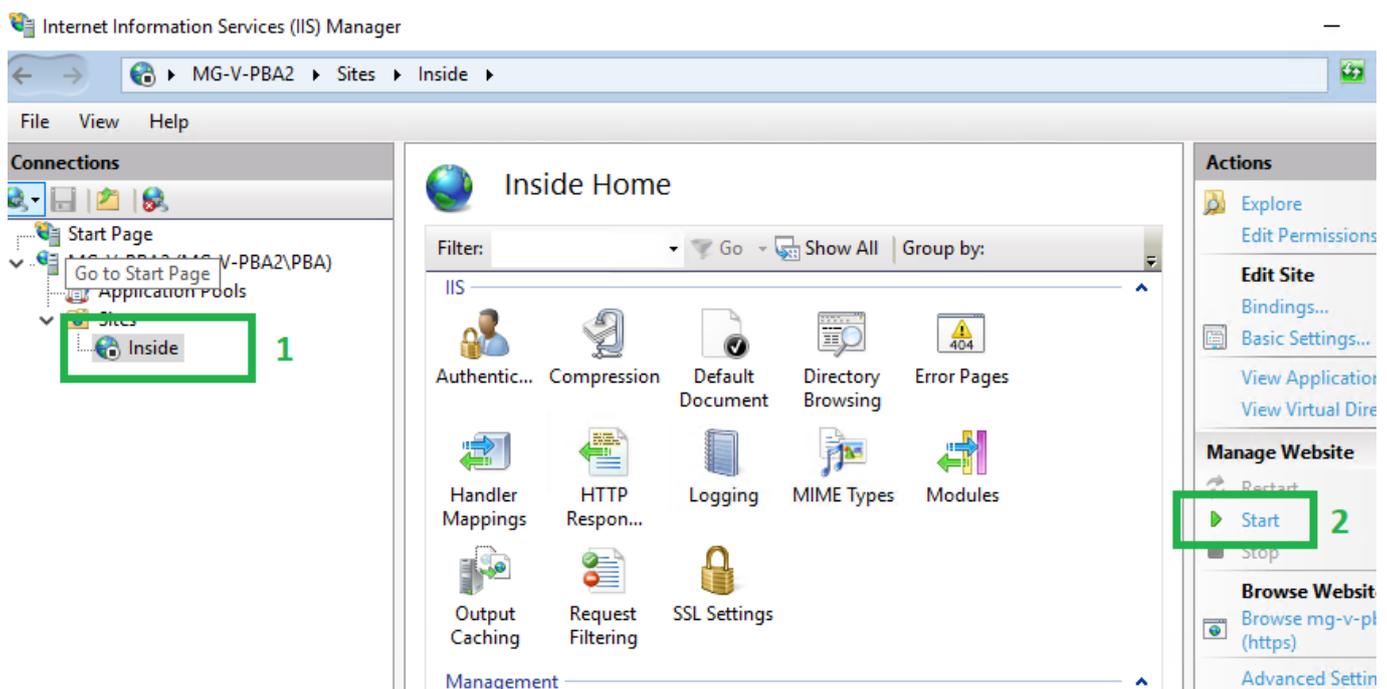
```

Bitte legen Sie in der Konfigurationsdatei „appsettings.json“ die Pfade für die Verzeichnisse „private“ und „public“ fest. Das Verzeichnis „private“ muss anschließend

für LIS/MIRTH/etc... freigegeben werden. In dieser Freigabe, sollen die Dateien vom LIS/MIRTH/etc... zur Steuerung der Backendservices gelesen und geschrieben werden. (S. Dokumentation über die LIS-Anbindung). Wichtig: Bitte nur das Verzeichnis „**private**“ freigeben. Das Verzeichnis „**public**“ wird vom ServiceInside für interne Zwecke verwendet.

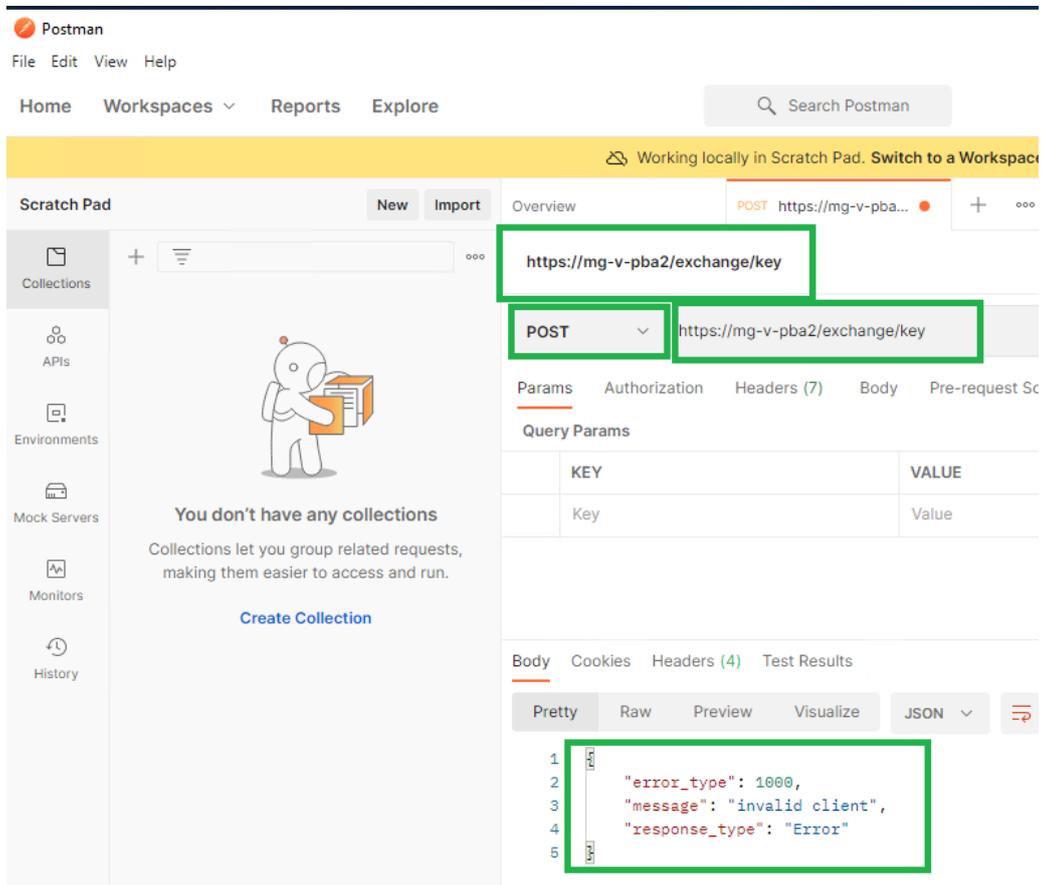
```
},  
"CleanUpDays": 30,  
"PrivateDirectory": "c:\\pba\\private",  
"PublicDirectory": "c:\\pba\\public",  
"MaxWorkerThreads": 10,  
"ServiceOutsideIP": "172.21.16.98",
```

9. Starten Sie nun die neu angelegte Seite „Inside“ im IIS Manager.



10. Zum Testen vom ServiceInside führen Sie bitte die Anwendung „**Postman.exe**“ im Verzeichnis „**Postman**“ aus. Anschließend führen Sie bitte in der Postman eine HTTPS POST-Abfrage mit der URL Hostname/exchange/key aus.

z.B: <https://mg-v-pba2/exchange/key>



Sollte der Webserver eine Antwort mit dem error_type: 1000 „invalid client“ zurück liefern, so wurde der ServiceInside erfolgreich eingerichtet. Sollte der ServiceInside bei der Abfrage nicht korrekt reagieren, aktivieren Sie bitte in der Konfigurationsdatei „appsettings.json“ den **DEBUG** Log und legen Sie einen Pfad für die Protokollierung der Fehler fest. Der ServiceInside protokolliert nun alle Abfragen mit den Fehlern in der Datei.

```
"Log": {
  "Type": "DEBUG",
  "Path": "c:\\tmp\\inside",
  "LogLevel": "Debug"
}
```

Zur weiteren Recherche von Problemen kann auch in der „C:\\inetpub\\wwwroot\\web.config“ Datei die Protokollierung der Console-Ausgabe aktiviert werden.

Verschlüsseltes Smtppassword

```
"SmtpPort": 25,
"SmtpSSL": false,
"SmtpUsername": "SmtpUser",
"SmtpPassword": "yd/KLV70/pXzyHaveZM3b01CfcVklLF1kDVHXYU0GrQv6B+w="
```

```
web.config - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <location path="." inheritInChildApplications="false">
    <system.webServer>
      <handlers>
        <add name="aspNetCore" path="*" verb="*" modules="AspNetCoreModuleV2" resourceType="Unspecified" />
        <aspNetCore processPath=".\\ServiceInside.exe" stdoutLogEnabled="true" stdoutLogFile="c:\\tmp\\logs.txt" hostingModel="inprocess" />
      </handlers>
    </system.webServer>
  </location>
</configuration>
<!--ProjectGuid: 19971e8e-9475-495d-b0a7-16c69c0594ab-->
```

Nach der erfolgreichen Installation vom ServiceInside setzen Sie bitte den LOG Modus in der „**appsettings.json**“ auf ERROR, damit nur die Fehler protokolliert werden.

Einstellungen der erneuten Anfragen an LIS

Sollte das LIS bei der initialen Anfrage (.log – SUBSCRIBE) keinen Auftrag in seinen Stammdaten finden, kann der ServiceInside so konfiguriert werden, dass er nach bestimmter Zeit dem LIS erneut die Anfrage sendet.

```
"MaxTryNotFoundResults": 2, // Number of tries to check not found results
"CheckIntervalNotFoundResultsInMinutes": 1440, // Time interval in minutes to check not found results again
"ServiceOutsideIP": "::1", // IP of service outside to whitelist for communication (trusted header)
```

Der Parameter **MaxTryNotFoundResults** legt die Anzahl der Versuche von nicht gefundenen Befunden fest.

Der Parameter **CheckIntervalNotFoundResultsInMinutes** legt das Zeitintervall fest, in dem der ServiceInside die erneuten Anfrage an LIS bzgl. nicht gefundenen Befunde senden soll.

Ist der Parameter **MaxTryNotFoundResults** mit dem Wert 2 und der Parameter **CheckIntervalNotFoundResultsInMinutes** mit dem Wert 1440(24 Std.) parametrisiert, so sendet der ServiceInside insgesamt zwei Mal innerhalb 48 Std. (je nach 24 Std.) dem LIS eine erneute Anfrage (.log – SUBSCRIBE)