

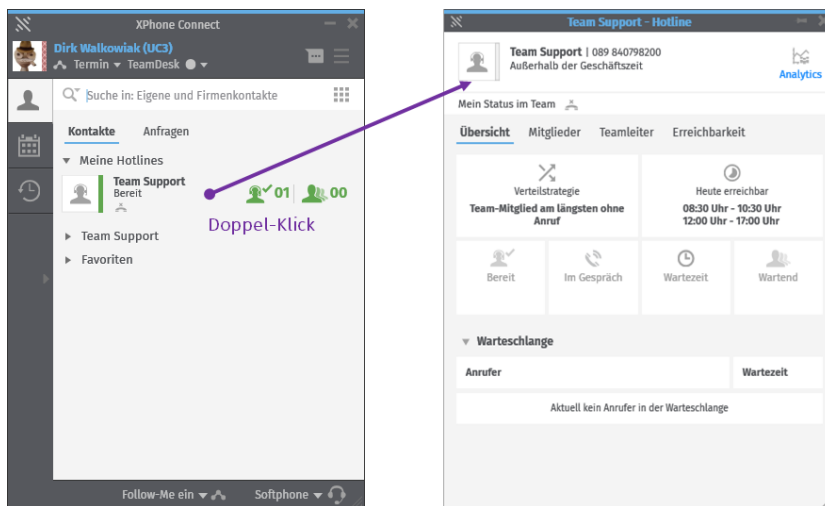
TeamDesk Dashboard V3

Stand: 10.01.2022

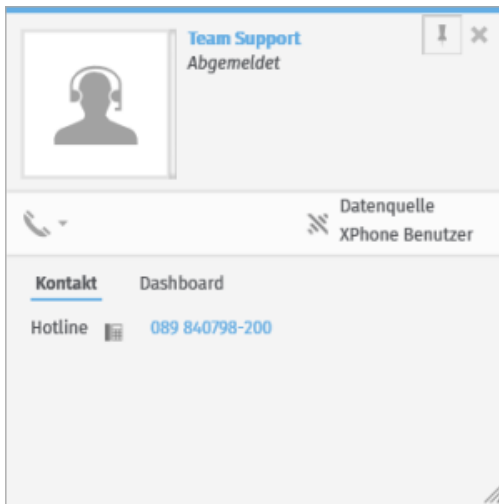
- Überblick
- Installation
 - Web Services
 - .NET Core IIS Hosting Runtime
 - Installation mit dem XpRestApiInstaller
 - Manuelle Installation
 - IIS Application Pool "XPhoneConnectApi"
 - Presence API
 - Rest API
 - Powershell Scripts
 - Überprüfen der Installation
 - Presence Api
 - Powershell Scripts
 - REST Api
- TeamDesk Dashboard
 - Import
 - Bedingungen
- Lizenzierung

Überblick

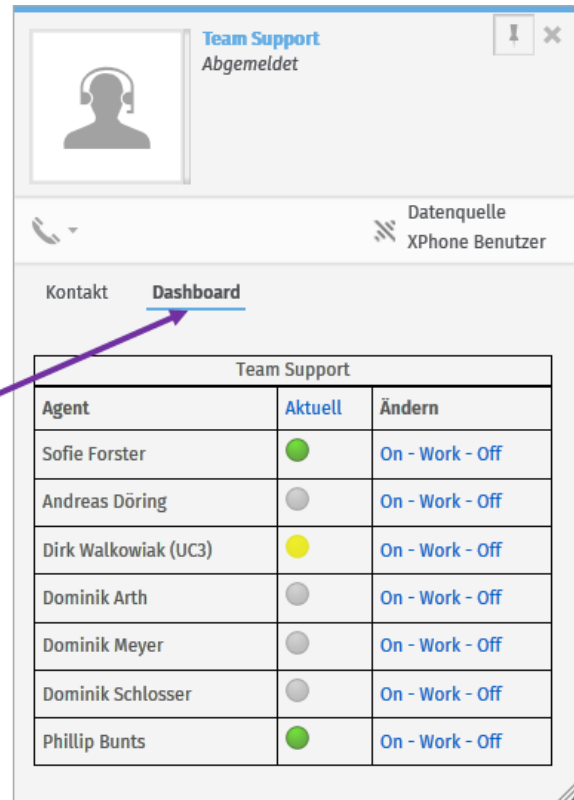
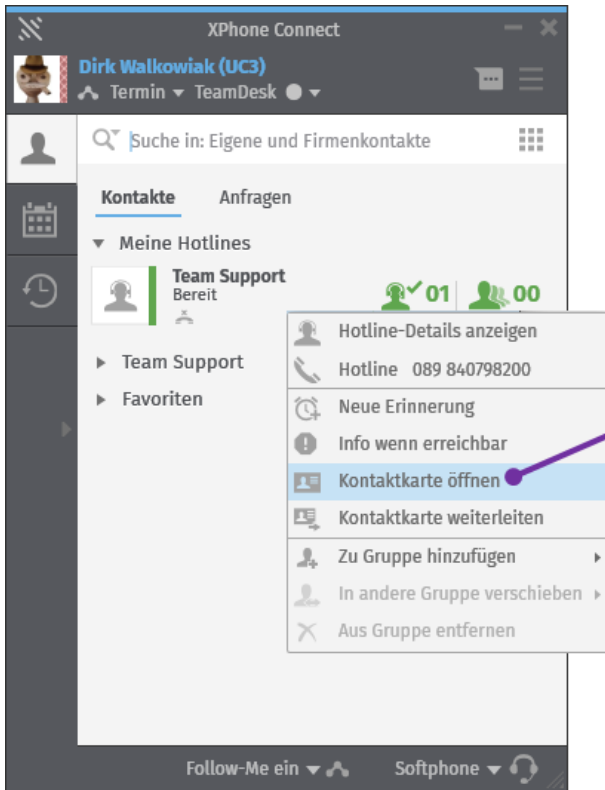
Hotlines (bzw. TeamDesks) bieten per Doppel-Klick eine Menge Informationen zum aktuellen Zustand an. Man hat jedoch kaum Steuerungs-Möglichkeiten.



Im Kontext-Menü (rechte Maustaste anstatt Doppel-Klick) offenbart sich das TeamDesk - rein software-technisch gesehen - als "XPhone Benutzer": es wird eine Kontaktkarte angeboten! Diese ist allerdings eher langweilig: mehr als die Hotline-Nummer wird hier nicht angezeigt.



Aber - wo eine Kontaktkarte ist, kann ein Dashboard nicht weit sein! Und genau so verhält es sich auch: wir haben das TeamDesk-Dashboard entdeckt!



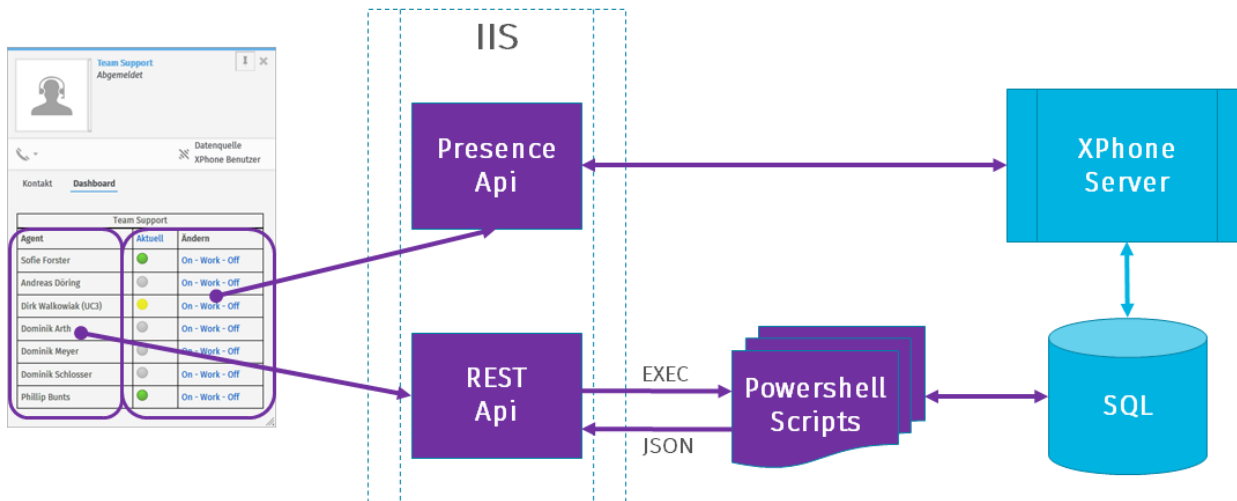
Dashboards sind dafür gemacht, zusätzliche Funktionalität in den Client zu bekommen. Das Kontext-Menü "Kontaktkarte öffnen" bietet einem Dashboard-Programmierer - also einem Web-Entwickler - alle Freiheiten. In diesem Artikel wird beschrieben, wie man mit Hilfe von passenden REST Schnittstellen ein Dashboard baut, das dem Teamleiter neue Steuerungsmöglichkeiten für seine Hotline bietet.

Das hier vorgestellte Dashboard

- zeigt alle zugewiesenen Agenten mit ihrem aktuellen TeamDesk-Status an und
- erlaubt dem Teamleiter zudem, den Agenten-Status aktiv zu ändern!

Die Technologie dahinter besteht aus zwei REST Api's, die auf dem IIS des XPhone Connect Servers installiert werden.

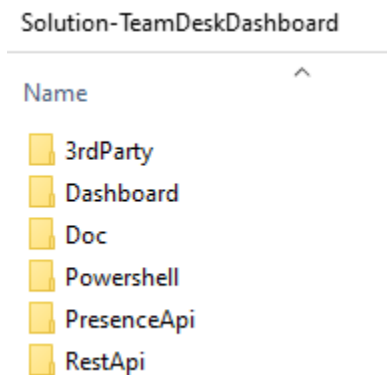
- Das "Presence Api" greift direkt auf die Präsenzschnittstelle des XPhone Servers zu. Damit werden die Agenten-Zustände gesteuert.
- Das "REST Api" liefert passgenau die Informationen, die man für den Aufbau des Dashboards benötigt. Für maximale Flexibilität werden dafür Powershell Scripts mit direktem SQL Server Zugriff eingesetzt.



Installation

Nachdem Sie dieses Kapitel Schritt für Schritt durchgearbeitet haben, können Sie das mitgelieferte TeamDesk Dashboard verwenden.

Das Installations-Medium ist eine ZIP Datei namens "Solution-TeamDeskDashboard.zip" mit dieser Struktur:



Web Services

Melden Sie sich mit Admin-Rechten auf dem XPhone Connect Server Rechner an.

Nach unseren bisherigen Erfahrungen kann die Installation bei laufendem Betrieb des XPhone Connect Servers durchgeführt werden. Rechnen Sie aber am besten auch damit, dass der XPhone Connect Server Dienst im Laufe der Installation beendet und neu gestartet werden muss.

.NET Core IIS Hosting Runtime

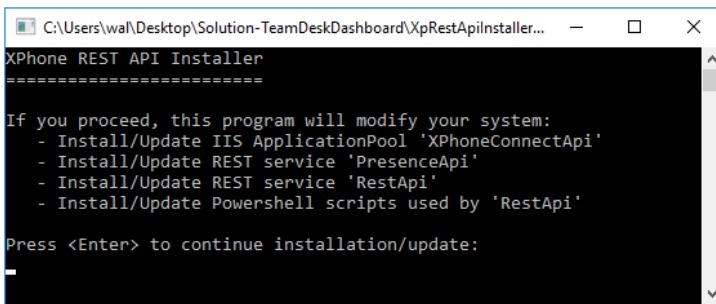
Installieren Sie das .NET Core IIS Hosting Runtime durch Ausführung des Setups ".\3rdParty\dotnet-hosting-3.1.16-win.exe".

Nach unseren bisherigen Erfahrungen kann das im laufenden Betrieb ohne Neustart des Rechners erfolgen. Das kann natürlich von System zu System anders sein.

Installation mit dem XpRestApiInstaller

Wir empfehlen diese Installations-Variante!

Sie finden das Programm "XpRestApiInstaller.Exe" nach dem Entpacken der ZIP-Datei im Hauptverzeichnis. Starten Sie das Programm per Doppel-Klick. Es fordert Admin-Rechte an und nach erfolgreicher Prüfung, ob sich auf dem System ein installierter XPhone Connect Server befindet, zeigt es dieses Konsolenfenster an:

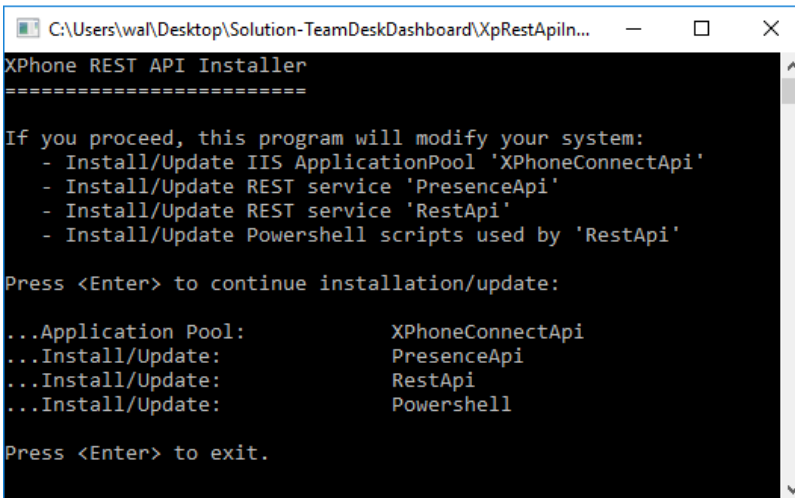


```
C:\Users\waf\Desktop\Solution-TeamDeskDashboard\XpRestApiInstaller...
XPhone REST API Installer
=====
If you proceed, this program will modify your system:
- Install/Update IIS ApplicationPool 'XPhoneConnectApi'
- Install/Update REST service 'PresenceApi'
- Install/Update REST service 'RestApi'
- Install/Update Powershell scripts used by 'RestApi'

Press <Enter> to continue installation/update:
_
```

Wenn Sie das Programm an dieser Stelle abbrechen, wurden noch keine Veränderungen am System vorgenommen!

Wenn Sie mit <Enter> die Installation fortsetzen, werden alle Komponenten installiert. Die Details dazu werden im Abschnitt "Manuelle Installation" beschrieben.



```
C:\Users\waf\Desktop\Solution-TeamDeskDashboard\XpRestApiIn...
XPhone REST API Installer
=====
If you proceed, this program will modify your system:
- Install/Update IIS ApplicationPool 'XPhoneConnectApi'
- Install/Update REST service 'PresenceApi'
- Install/Update REST service 'RestApi'
- Install/Update Powershell scripts used by 'RestApi'

Press <Enter> to continue installation/update:

...Application Pool:      XPhoneConnectApi
...Install/Update:      PresenceApi
...Install/Update:      RestApi
...Install/Update:      Powershell

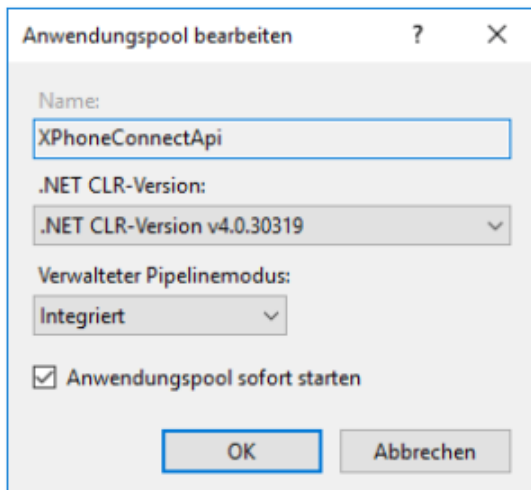
Press <Enter> to exit.
```

Manuelle Installation

Hier werden alle Installationsschritte im Detail beschrieben, so dass man die Installation auch manuell durchführen könnte. Außerdem dient das Kapitel dem besseren Verständnis, was der XpRestApiInstaller alles automatisch erledigt.

IIS Application Pool "XPhoneConnectApi"

Legen Sie im IIS Manager einen neuen Application Pool namens "XPhoneConnectApi" an.



Anwendungspool bearbeiten

Name: XPhoneConnectApi

.NET CLR-Version: .NET CLR-Version v4.0.30319

Verwalteter Pipelinemodus: Integriert

☒ Anwendungspool sofort starten

OK Abbrechen

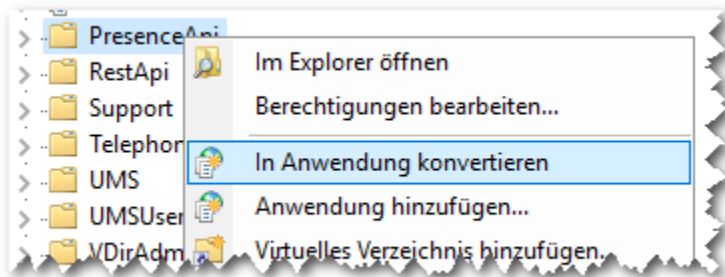
Der Application Pool wird später dafür verwendet, dem REST Api Webservice die notwendigen Berechtigungen auf den XPhone Connect SQL Server zu geben.

Presence API

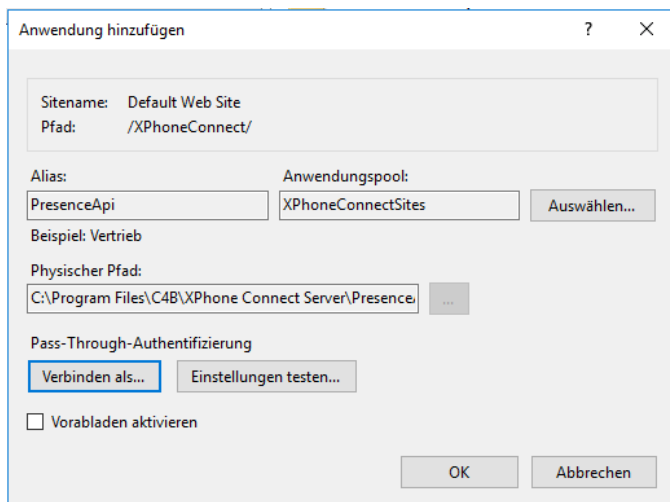
Kopieren Sie den gesamten Ordner "PresenceApi" als gleichnamigen Unterordner ins XPhone Server Programmverzeichnis.

Starten Sie jetzt den IIS-Manager, z.B. über den XPhone Connect Server Manager (Menü "Tools > IIS Manager").

Navigieren Sie zu ".\Sites\Default Web Site\XPhoneConnect\PresenceApi" und konvertieren Sie das Verzeichnis in eine Anwendung:



Sie können alle angebotenen Einstellungen unverändert mit "OK" bestätigen.

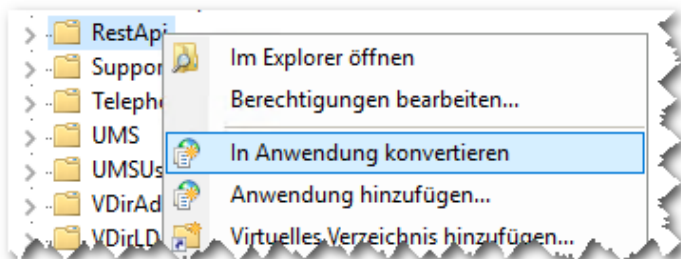


Rest API

Kopieren Sie den Ordner "RestApi" als gleichnamigen Unterordner ins XPhone Server Programmverzeichnis.

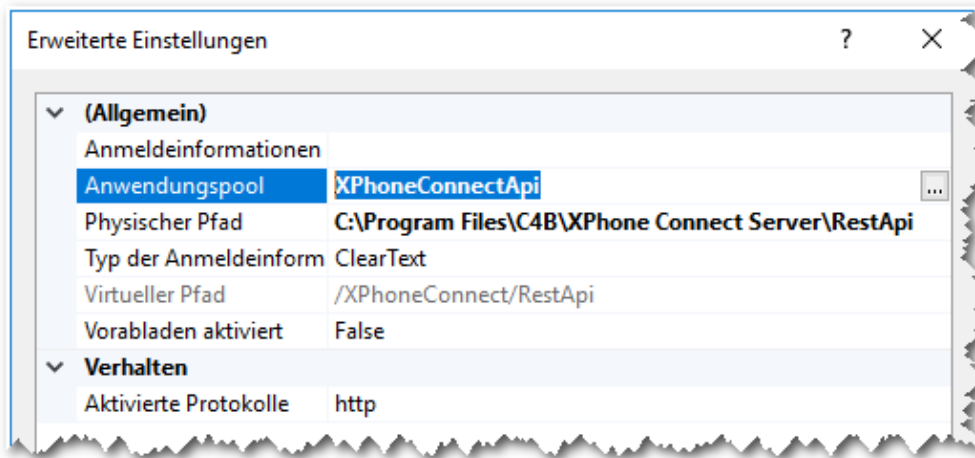
Starten Sie jetzt den IIS-Manager, z.B. über den XPhone Connect Server Manager (Menü "Tools > IIS Manager").

Navigieren Sie zu ".\Sites\Default Web Site\XPhoneConnect\RestApi" und konvertieren Sie das Verzeichnis ebenfalls in eine Anwendung:



Sie können wiederum alle angebotenen Einstellungen unverändert mit "OK" bestätigen.

Weisen Sie dieser Anwendung den vorhin angelegten Application Pool "XphoneConnectApi" zu:



Powershell Scripts

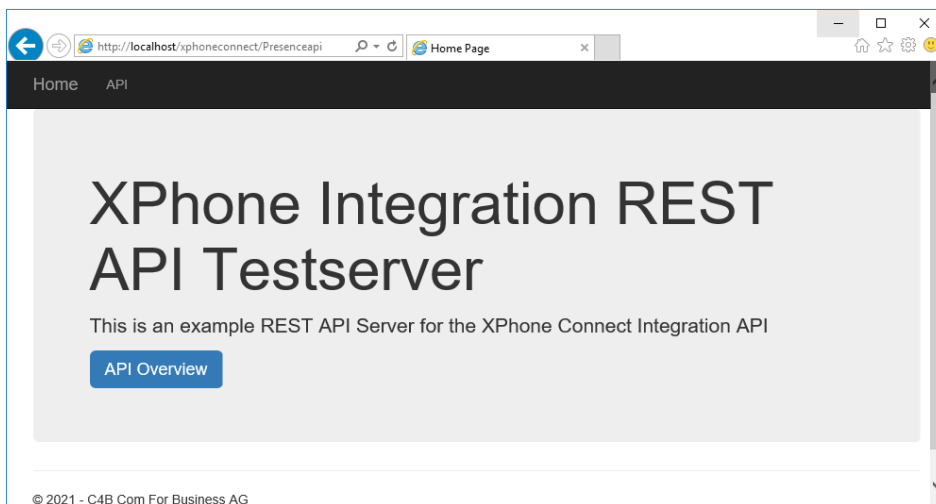
Kopieren Sie den Ordner "Powershell" als neuen Unterordner nach "C:\ProgramData\C4B\Powershell".

Überprüfen der Installation

Presence Api

Prüfen Sie die korrekte Installation durch Aufruf dieser URL im Web-Browser auf dem XPhone Connect Server: "<http://localhost/XPhoneConnect/PresenceApi/>"

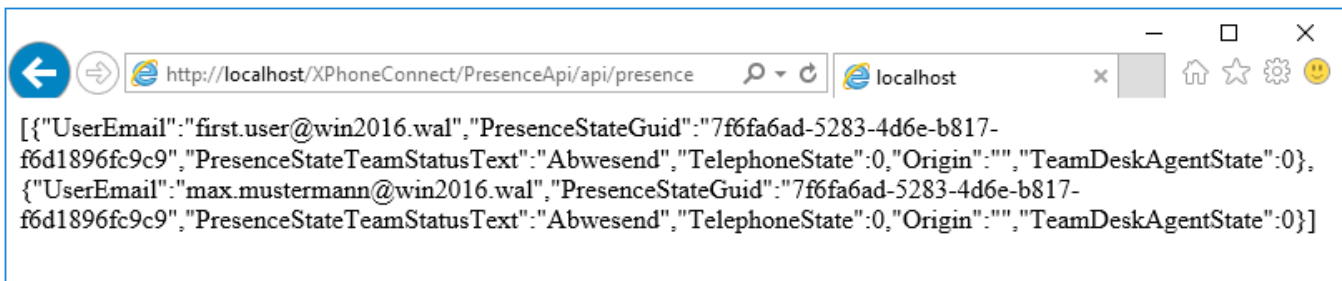
Wenn Sie diese Seite sehen, war die Installation im IIS erfolgreich:



Prüfen Sie zusätzlich anhand dieser URL "<http://localhost/XPhoneConnect/PresenceApi/api/presence>", ob Sie auch tatsächlich auf die Präsenzinformation der XPhone Benutzer zugreifen können.

Hinweis: Beachten Sie, dass vom Presence Api nur Benutzer mit einer eindeutigen E-Mail-Adresse angezeigt werden können!

Im Erfolgsfall sehen Sie eine Ausgabe vergleichbar mit dieser:



Es könnte sein, dass der Internet Explorer auf Ihrem System keine .JSON Dateien anzeigen kann (will). Stattdessen bietet er sie zum Download an. Auch das kann als erfolgreicher Aufruf gewertet werden!

Tipp: Mit folgendem Registry-Key kann man den IE dazu bringen, die .JSON Dateien im Browser-Fenster anzuzeigen. Speichern Sie den Inhalt des Fensters in einer Textdatei mit der Endung ".REG" und führen Sie die Datei per Doppel-Klick aus:

```
Windows Registry Editor Version 5.00;
; Tell IE 7,8,9,10,11 to open JSON documents in the browser on Windows XP and later.
; 25336920-03F9-11cf-8FD0-00AA00686F13 is the CLSID for the "Browse in place" .
;
[HKEY_CLASSES_ROOT\MIME\Database\Content Type\application/json]
"CLSID"="{25336920-03F9-11cf-8FD0-00AA00686F13}"
"Encoding"=hex:08,00,00,00
```

Powershell Scripts

Die Powershell Scripts benötigen lesenden Zugriff auf den XPhone Connect SQL Server.

Überprüfen Sie im Script "Get-TeamDeskAssignedAgents.ps1" die SQL Connection Parameter.

Für einen lokal installierten SQL Server (Instanz: XPCONNECT) sieht das so aus:

Get-TeamDeskAssignedAgents.ps1

```
Param(
    # SQL Query Parameter
    [string]$TeamDeskName = "Demo Team",

    # SQL Connection Parameter
    [string]$sqlHost = "(local)\XPCONNECT"
    , [string]$DB = "XPDATA"
    #, [string]$User = "api_user"
    #, [string]$Password = "api_pwd"
)

#Script geht hier weiter
#...
```

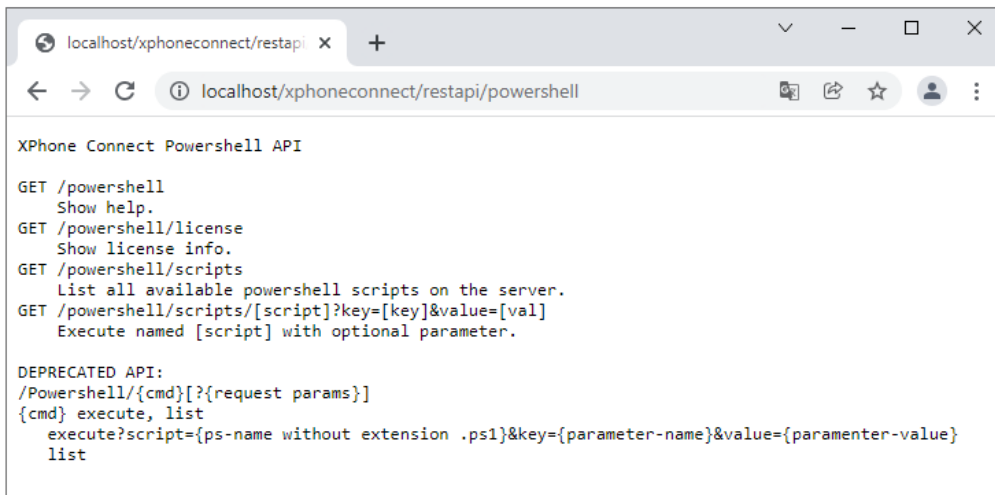
REST Api

Die Überprüfung erfolgt in 3 Schritten:

1. Wurde die REST Api erfolgreich im IIS installiert?
2. Kann die REST Api erfolgreich auf die Powershell-Skripts zugreifen?
3. Können die Powershell-Skripts erfolgreich auf die Datenbank zugreifen, wenn sie über das REST Api ausgeführt werden.

Test 1. Prüfen Sie die korrekte Installation der REST Api durch Aufruf dieser URL im Web-Browser auf dem XPhone Connect Server: "<http://localhost/xphoneconnect/RestApi/Powershell>"

Wenn Sie diese Seite mit der Beschreibung der Aufruf-Syntax sehen, war die Installation erfolgreich:



localhost/xphoneconnect/restapi x +

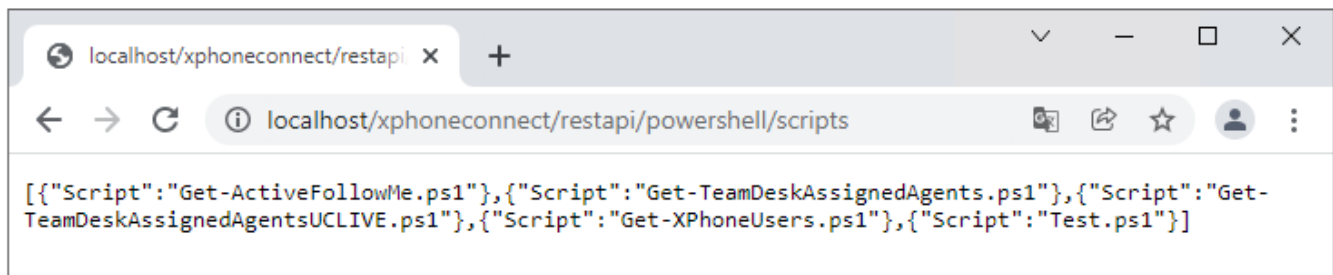
localhost/xphoneconnect/restapi/powershell

XPhone Connect Powershell API

```
GET /powershell
Show help.
GET /powershell/license
Show license info.
GET /powershell/scripts
List all available powershell scripts on the server.
GET /powershell/scripts/[script]?key=[key]&value=[val]
Execute named [script] with optional parameter.

DEPRECATED API:
/Powershell/{cmd}[?{request params}]
{cmd} execute, list
execute?script={ps-name without extension .ps1}&key={parameter-name}&value={parameter-value}
list
```

Test 2. Rufen Sie diese URL "<http://localhost/xphoneconnect/RestApi/Powershell/scripts>" im Browser auf. Sie erhalten als Ergebnis die Liste der auf dem System installierten Powershell-Skripts:



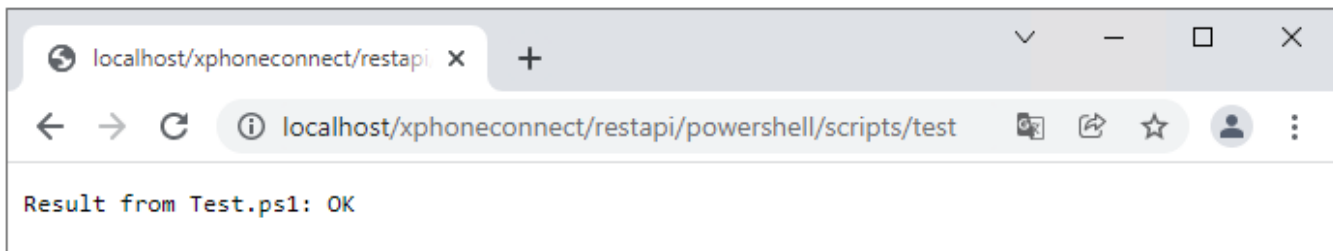
localhost/xphoneconnect/restapi x +

localhost/xphoneconnect/restapi/powershell/scripts

```
[{"Script": "Get-ActiveFollowMe.ps1"}, {"Script": "Get-TeamDeskAssignedAgents.ps1"}, {"Script": "Get-TeamDeskAssignedAgentsUCLIVE.ps1"}, {"Script": "Get-XPhoneUsers.ps1"}, {"Script": "Test.ps1"}]
```

Rufen Sie diese URL "<http://localhost/xphoneconnect/RestApi/Powershell/scripts/test>" im Browser auf. Das Script "test.ps1" liefert einen hart kodierten Text als Ergebnis zurück. Es wird noch kein Zugriff auf die Datenbank versucht.

Wenn Sie diese Seite sehen, war die Installation erfolgreich:



localhost/xphoneconnect/restapi x +

localhost/xphoneconnect/restapi/powershell/scripts/test

Result from Test.ps1: OK

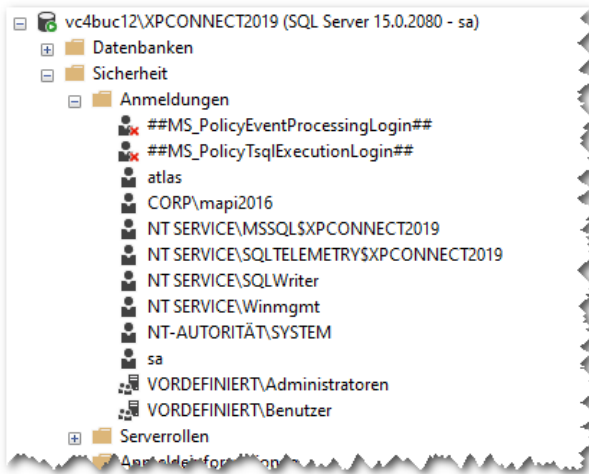
Test 3. Jetzt geht es darum, ob die Powershell Scripts auch auf die XPhone Datenbank zugreifen können.

Rufen Sie diese URL "<http://localhost/xphoneconnect/RestApi/Powershell/scripts/Get-XPhoneUsers>" im Browser auf. Das Script "Get-XPhoneUsers.ps1" ermittelt (bis zu) 10 XPhone Benutzer aus der der Datenbank und zeigt deren Namen und E-Mail-Adressen im JSON Format an.

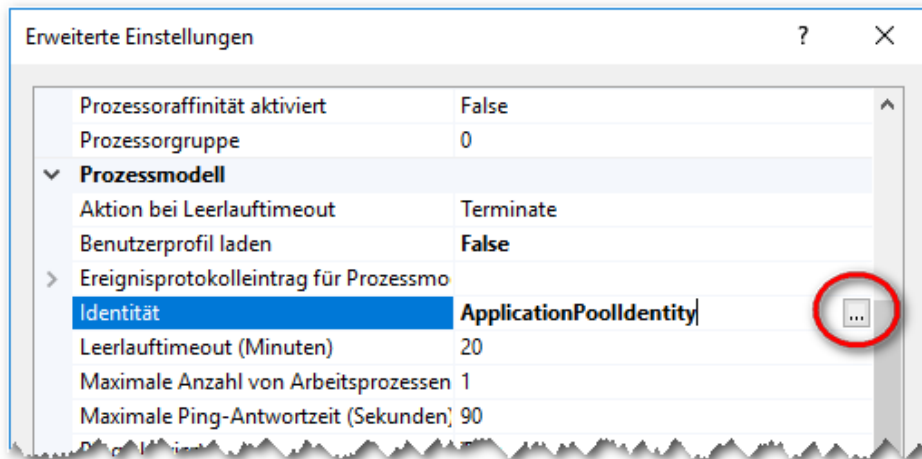
Beim ersten Versuch werden in der Regel keine Ergebnisse zurück geliefert! Stattdessen wird ein leeres JSON Array ("[]") ausgegeben. Das hat mit fehlenden Datenbank-Berechtigungen des Accounts zu tun, unter dem die REST Api im IIS läuft.

Die REST Api läuft mit den Berechtigungen, die für den IIS Application Pool "XPhoneConnectApi" vergeben wurden. Im Default ist das "ApplicationPoolIdentity" - und dieser hat keine Berechtigungen auf die Datenbank.

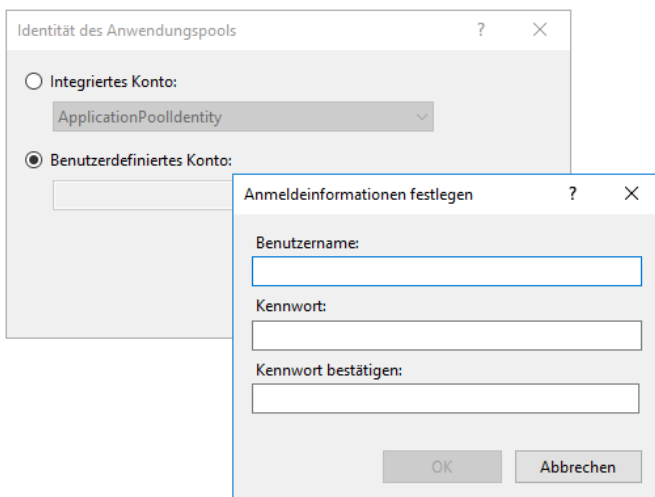
Ermitteln Sie einen berechtigten Datenbank-Benutzer z.B. mit Hilfe des SQL Server Managers:



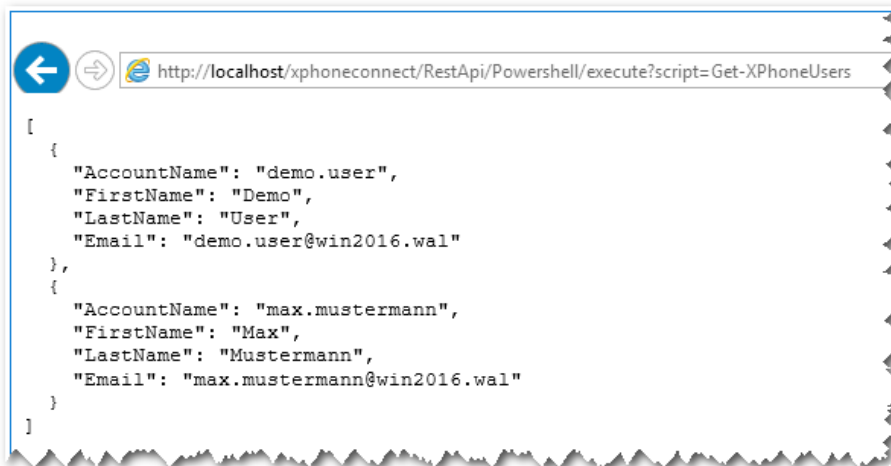
Navigieren Sie im IIS Manager zu den Application Pools und öffnen Sie die Erweiterten Einstellungen des Application Pools "XPhoneConenctApi":



Tragen Sie in der Maske zum Ändern des Identität einen Benutzer Account ein, der auf die Datenbank berechtigt ist:



Nachdem Sie alle Dialoge mit "Ok" bestätigt haben, rufen Sie erneut diese URL "<http://localhost/xphoneconnect/RestApi/Powershell/scripts/GetXPhoneUsers>" im Browser auf. Sie sollten jetzt eine Ausgabe in dieser Form erhalten:



TeamDesk Dashboard

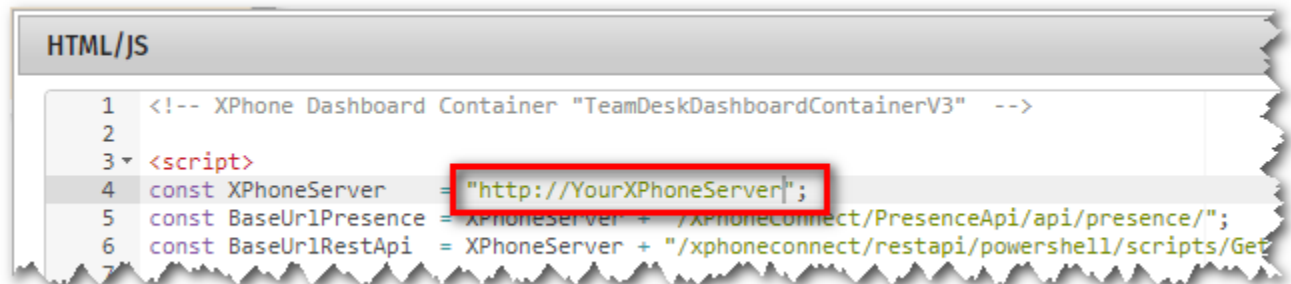
Import

Importieren Sie die Solution "TeamDeskDashboardSolution.zip" aus dem Ordner "Dashboard" in den XPhone Connect Dashboard Designer. Der Import bietet ein Dashboard "TeamDeskDashboard" und einen Container "TeamDeskDashboardContainer" an:



Öffnen Sie nach dem Import den HTML/JS Editor im Container "TeamDeskStatusRestApiPowershell".

Passen Sie die Konstante "XPhoneServer" für Ihre Installation an.



Bedingungen

Passen Sie die Bedingungen für das TeamDesk Dashboard an. Wählen Sie als Datenquelle "XPhone Benutzer":

Das Dashboard wird angezeigt, wenn die Kontaktdaten aus folgenden Datenquellen stammen:

☒ XPhone Benutzer



Legen Sie fest, für welche TeamDesks dieses Dashboard angezeigt werden soll. Die Bedingung enthält den Namen der TeamDesk Gruppe und wird über das Attribut "Contact.DISTINGUISHEDNAME" abgeprüft.

Nach dem Import sind exemplarisch bereits zwei Einträge vorhanden ("Demo Team 1", "Demo Team 2"). Dort können einfach die Namen der Teams an Ihre Umgebung angepasst werden. Weitere TeamDesks können nach dem gleichen Schema hinzugefügt werden.

☒ Zusätzliche Bedingungen

The screenshot shows a filter configuration interface. At the top, there are two buttons: "AND" and "OR", with "OR" selected. To the right of these buttons are icons for adding and removing conditions. Below the buttons, there are two filter conditions stacked vertically. Each condition consists of a dropdown menu showing "Contact.DISTINGUISHEDNAME", a text input field containing "Demo Team 1" (for the first condition) and "Demo Team 2" (for the second condition), and a dropdown menu showing "contains". Each condition also has a small "X" icon to its right for removal. The entire filter configuration is enclosed in a light gray border.

Lizenzierung

Das XPhone Connect REST API benötigt eine gültige Lizenz. Jede Entität ("Powershell", "AnyBell", "LogFile", etc.) wird dabei separat lizenziert.

Eine Lizenz besteht aus einer signierten XML Datei "license.xml", die in das Verzeichnis "C:\ProgramData\C4B\ApiLicense" kopiert werden muss. In dieser Datei sind alle erworbenen Lizenzen mit dem jeweiligen Ablaufdatum aufgeführt.

Achtung: Jede inhaltliche Änderung an der Datei führt dazu, dass die Signatur ungültig wird und somit die REST API Schnittstellen nicht mehr funktionieren!

Bitte wenden Sie sich an den Vertrieb oder den Support von C4B für weitere Informationen.