

Seafile auf Windows 7/8 mit https installieren

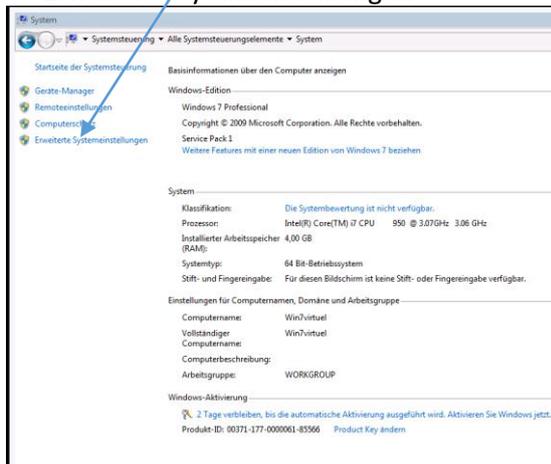
Zum Installieren und konfigurieren von Seafile werden folgende Programme benötigt:

- Python in der Version **2.7.4** (keine neuere Version verwenden!)
- Seafile Server Windows 4.0.6 oder aktueller
- Microsoft Web Plattform Installer 5.0 oder aktueller
- Seafile Client for Windows 4.1.4 um mit einem Windows Rechner synchronisieren zu können. Entsprechende Clients für MAC, Android und iOS sind ebenfalls erhältlich
- 7-ZIP (oder vergleichbares ZIP-Programm das .tar und .gz unterstützt)
- Notepad++ oder einen anderen Editor

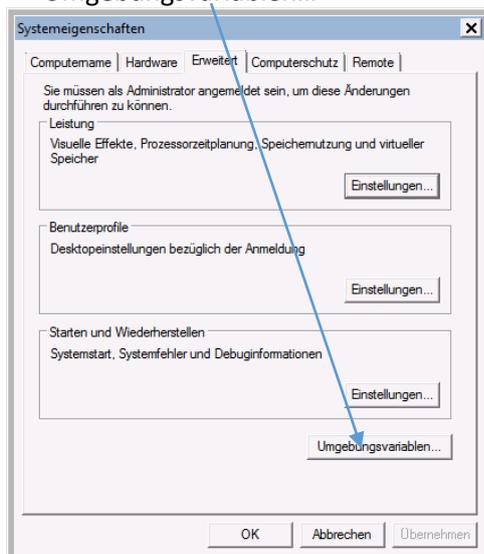
Zunächst werden 7-ZIP (oder ein vergleichbares ZIP-Programm das .tar und .gz unterstützt) und Notepad++ oder einen anderer Editor, zur Not auch Notepad von Microsoft installiert.

Nun wird **Python 2.7.4** installiert. Dazu werden einfach sämtliche Standardeinstellungen übernommen.

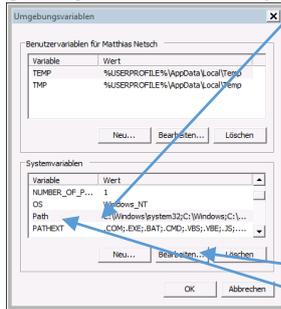
Wichtig: Anschließend muss eine Pfadvariable in den Systemvariablen von Windows angepasst werden. Dazu geht man folgendermaßen vor: Über Start => Systemsteuerung => System => Erweiterte Sytemeinstellungen



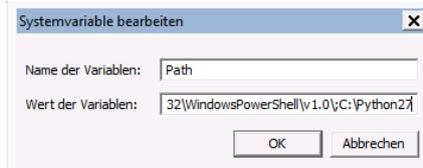
=> Umgebungsvariablen...



gelingt man zu den Systemvariablen



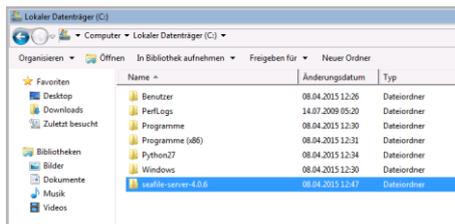
Hier die Variable „Path“ durch anklicken auswählen und „Bearbeiten“ gehen. Nun am Ende der Variable den Pfad zu Python (Standardmäßig „C:\Python27“) anhängen. Dazu zunächst ein Semikolon und anschließend den Pfad (z.B. C:\Python27) eingeben.



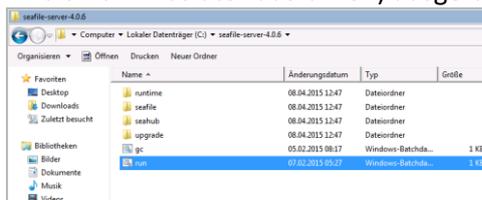
Und alles mit ok bestätigen.

Als nächstes wird der Seafile Server mit dem ZIP-Programm entpackt. Und den Inhalt der Entpackten Datei noch ein zweites Mal.

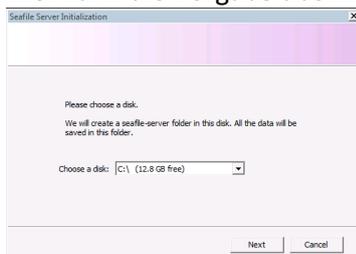
Dieses zweite entpackte Verzeichnis kopiere ich nun direkt nach C: (oder eine andere gewünschte Stelle.



Nun wird das Verzeichnis geöffnet und die Batchdatei „run“ wird als Administrator (rechte Maustaste => als Administrator ausführen) ausgeführt.



Es wird anschließend nach dem gewünschten Laufwerk für das Seafiler-Server Verzeichnis gefragt. Hier kann die Vorgabe übernommen oder wenn gewünscht ein anderes Laufwerk gewählt werden.

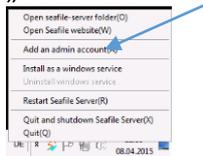


Anschließend wird über den Knopf „Next“ der Seafiler-Server installiert. Die erfolgreiche Installation erkennt man an einem Symbol unten rechts in der Taskleiste.



Hat das Symbol einen kleinen Haken war die Installation erfolgreich.

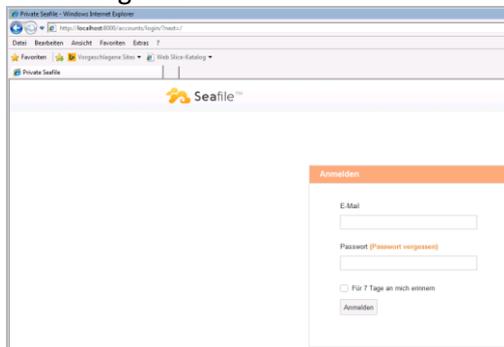
Als erstes muss nun ein Admin Account angelegt werden. Dazu rechter Mausklick auf das Symbol und „Add an Admin Account“ auswählen.



Nun den Admin-Account anlegen und mit ok bestätigen.

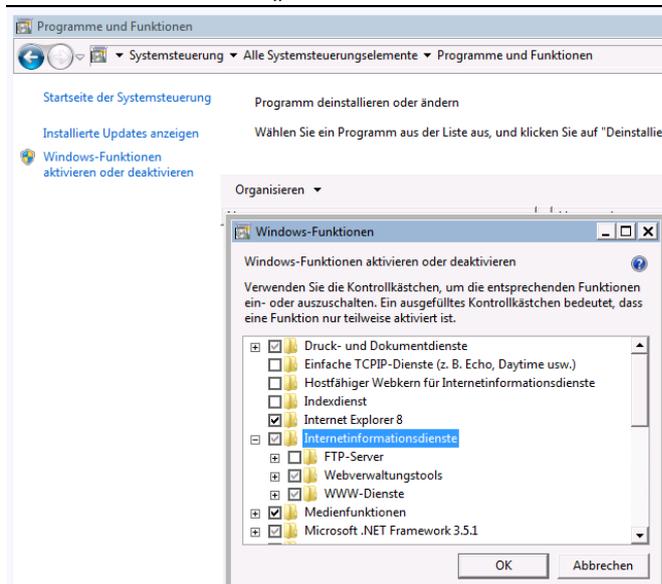


Jetzt kann der Server getestet werden. Hierzu einen Browser öffnen und <http://localhost:8000> in die Adresszeile eingeben. Das Ergebnis sollte so aussehen und eine Anmeldung mit dem Admin Account sollte möglich sein.

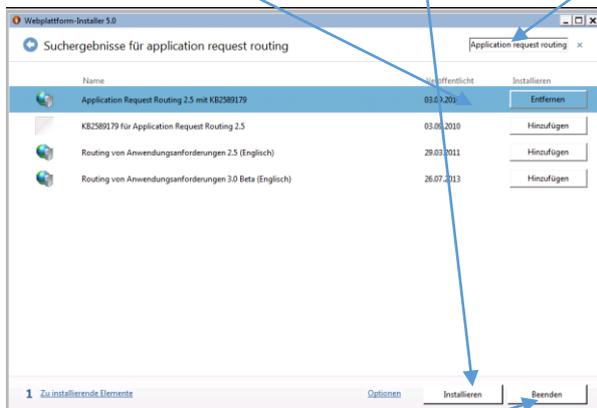


Damit ist die Grundeinrichtung von Seafiler beendet. Nachdem der Server bisher nur über http und den Port 8000 erreichbar ist, was mir persönlich zu unsicher ist erfolgt nun die Umstellung auf https (hier am Beispiel der internen Adresse 192.168.234.19, dies muss entsprechend angepasst werden!) über den Port 443 (oder einen anderen beliebig wählbaren Port). Hierzu wird ein sog. https-Proxy benötigt, den ich im nächsten Schritt einrichte.

Im ersten Schritt müssen Windows-Dienste nachinstalliert werden. Das geschieht über Systemsteuerung => Programme und Funktionen => Windows Funktionen aktivieren und Deaktivieren. Hier bei „Internetinformationsdienste“ einen Haken setzen und mit ok bestätigen.

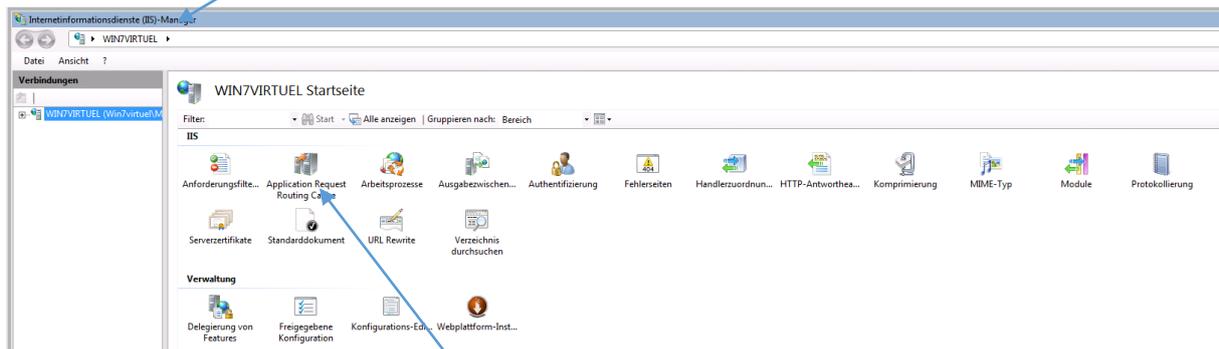
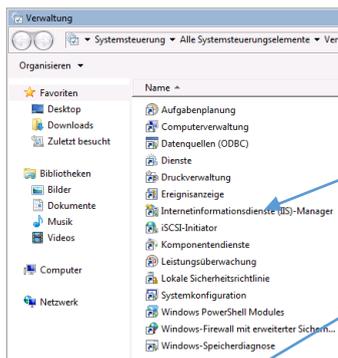


Wenn diese Installation abgeschlossen ist muss noch der „Microsoft Web Platform Installer 5.0“ installiert werden. Hier suchen wir nun nach „Application request routing“, wählen das entsprechende Paket aus und installieren es.



Danach kann das Programm beendet werden.

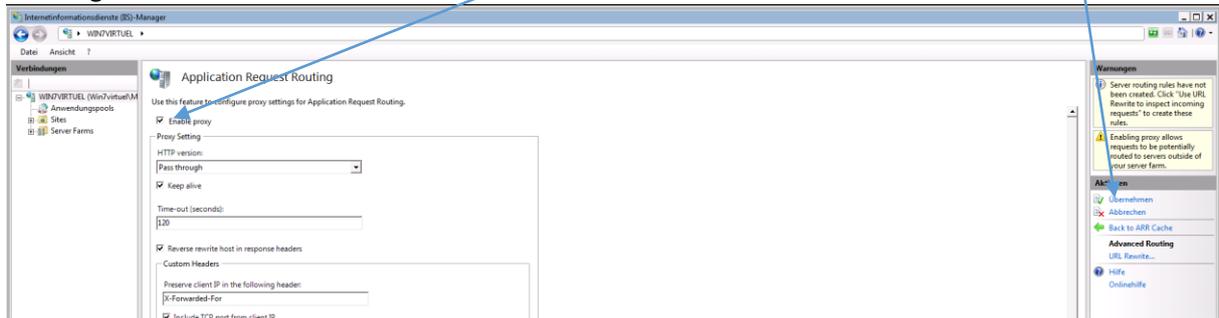
Nun wird über Systemsteuerung => Verwaltung der „Internetinformationsdienste (IIS)-Manager“ geöffnet



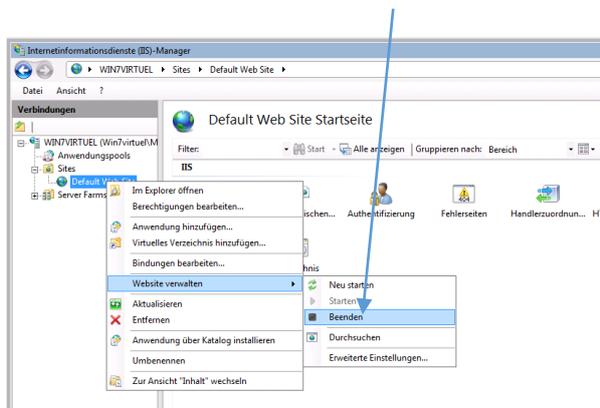
Und hier als erstes das „Application Request Routing“ aktiviert. Dazu machen wir einen Doppelklick auf „Application Request Routing“ und klicken dann auf der rechten Seite auf „Server Proxy Settings...“



Nun muss im mittleren Fenster nur „Enable Proxy“ angeklickt und das Ganze mit „Übernehmen“ bestätigt werden.

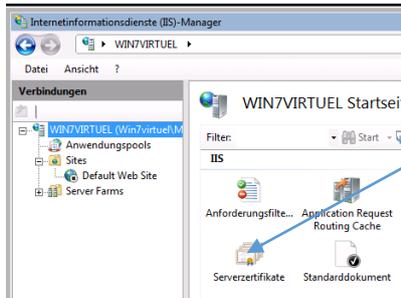


Jetzt muss eine Umleitungsseite eingerichtet werden (nicht die Default Web Site). Die „Default Web Site“ wird jetzt deaktiviert bzw. beendet (falls sie nicht für anderer Web-Dienste verwendet wird).

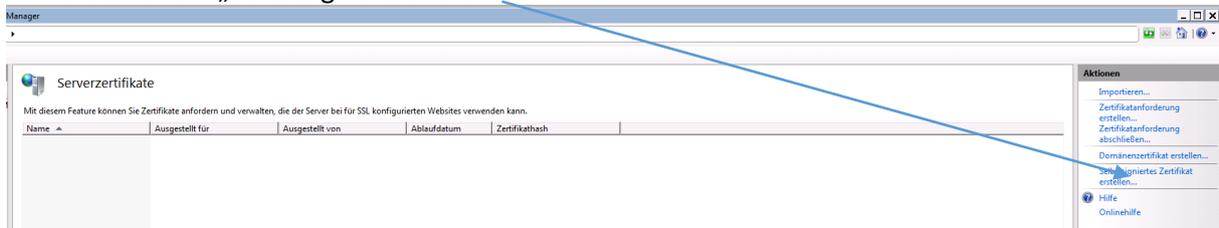


Um https einrichten zu können muss zunächst ein Serverzertifikat eingerichtet werden. Der Einfachheit halber nehmen wir ein selbst signiertes.

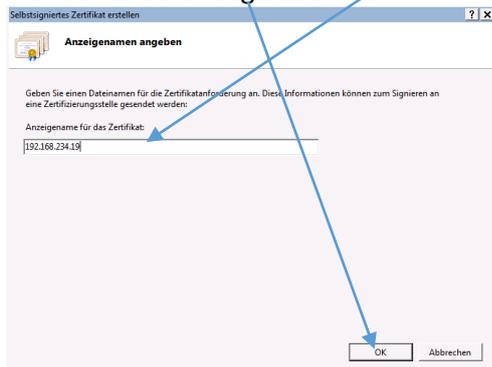
Dazu klicken wir im Mittleren Teil des Fensters auf „Serverzertifikate“ doppelt



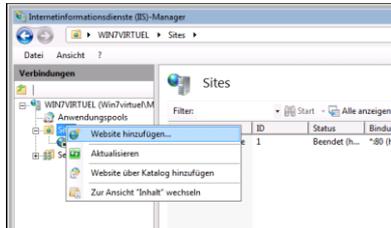
Und erstellen ein „Selbstsigniertes Zertifikat“



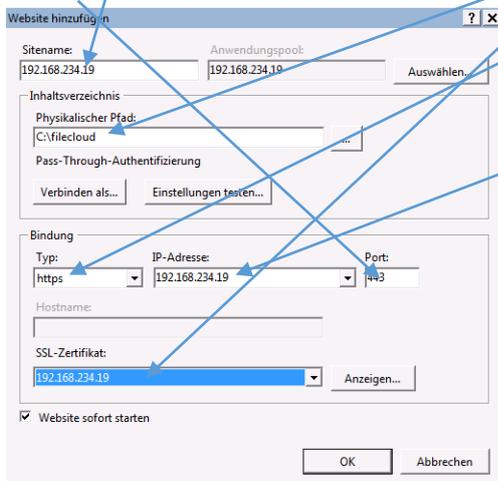
Als Anzeigename wählen wir die IP-Adresse des Servers aus, da bisher keine öffentliche IP-Adresse besteht und bestätigen mit ok.



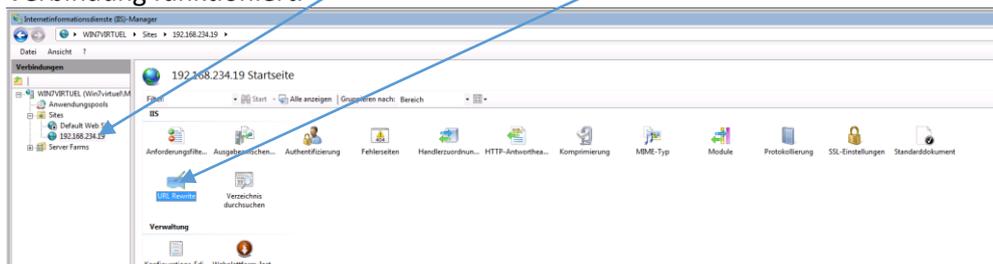
Im nächsten Schritt wird unter „Sites“ auf der linken Seite eine Webseite hinzugefügt werden.



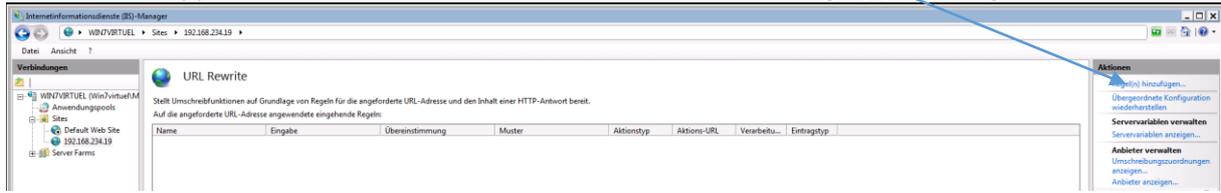
Als Sitename wähle ich wieder die IP-Adresse und Physikalischen Pfad lege ich einen Ordner Namens „filecloud“ an. Der Name und Ort ist frei wählbar. Als Bindungstyp wähle ich „https“, Bindungs-IP-Adresse ist die eigene IP-Adresse und als SSL-Zertifikat wähle ich das vorhin erstellte Zertifikat aus. Der Port kann auf 443 bleiben. Nun alles mit ok bestätigen.



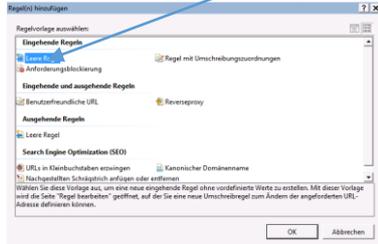
Jetzt müssen in der neuen Website noch Umleitungsregeln angelegt werden, damit die https-Verbindung funktioniert.



Dazu einen Doppelklick auf „URL Rewrite“ und anschließend auf „Regel(n) hinzufügen...“



Folgende zwei Leere-Regeln müssen hier angelegt werden:



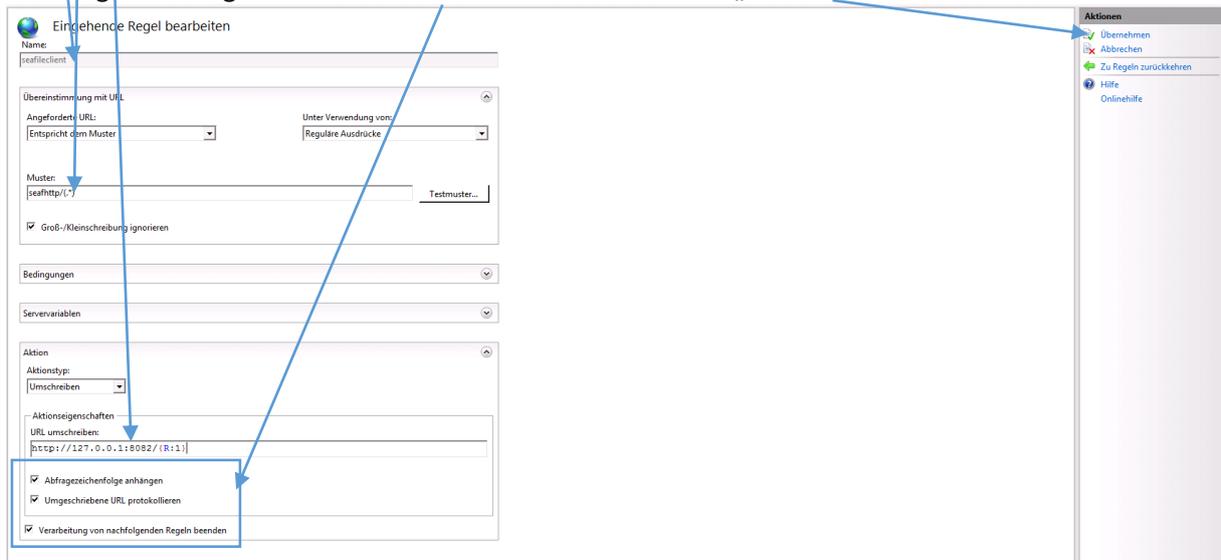
Seafielclient mit folgenden Werten:

Name: seafielclient

Muster: seafhttp/(.*)

URL umschreibung: http://127.0.0.1:8082/{R:1}

„Abfragezeichenfolge anhängen“, „Umschriebene URL protokollieren“ und „Verarbeitung von nachfolgenden Regeln beenden“ anhaken. Anschließend auf „Übernehmen“ klicken.



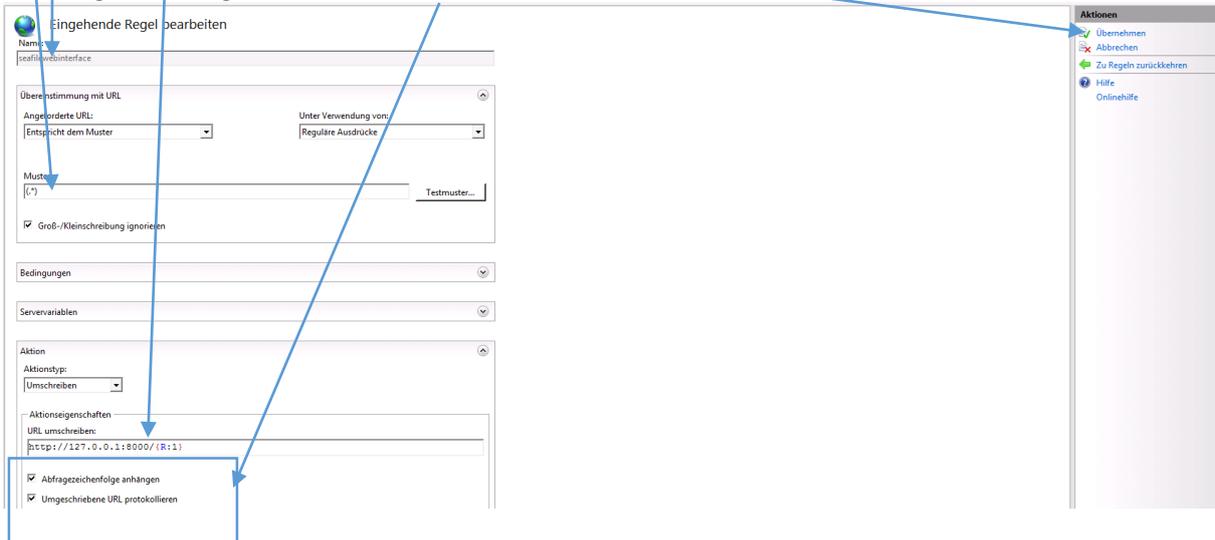
seafirewebinterface mit folgenden Werten:

Name: seafirewebinterface

Muster: (.*)

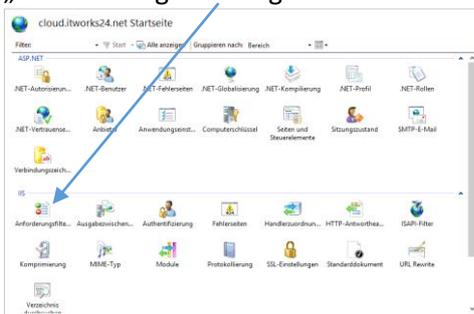
URL umschreibung: http://127.0.0.1:8000/{R:1}

„Abfragezeichenfolge anhängen“, „Umschreibung URL protokollieren“ und „Verarbeitung von nachfolgenden Regeln beenden“ anhängen. Anschließend auf „Übernehmen“ klicken.



Da der IIS beim Upload von Haus aus ein Limit von 30 MB eingestellt hat muss das noch angepasst werden.

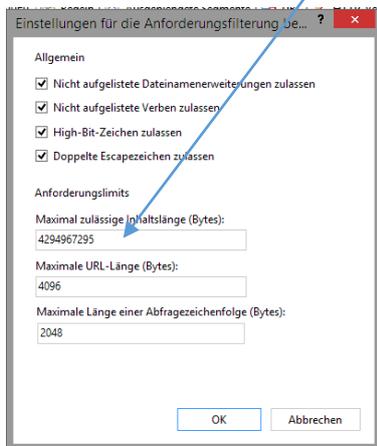
Das erfolgt indem man im IIS die entsprechende Web-Site auswählt und im mittleren Bereich dann „Anforderungsfilterung“ auswählt.



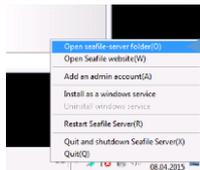
Die Filterung mit einem Doppelklick öffnen und rechts „Featureeinstellungen bearbeiten...“ auswählen.



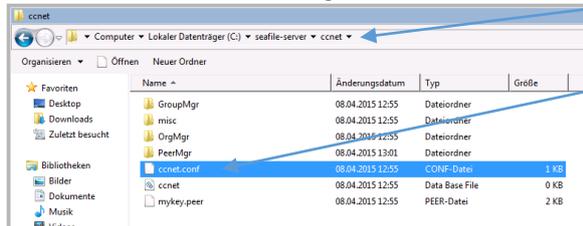
Nun unter „Maximal zulässige Inhaltslänge (Bytes):“ den gewünschten Wert eintragen (der maximal zulässige Wert beträgt 4294967295, was etwa 4 GB entspricht; der Standardwert beträgt 3000000 was etwa 30 MB entspricht)



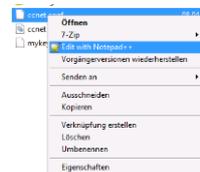
Anschließend müssen noch zwei Config-Dateien von Seafile angepasst werden. Dazu klickt man auf das Seafile-Symbol in der Taskleiste mit der rechten Maustaste und wählt „Open seafile-server folder“ aus.



Im sich öffnenden Ordner gibt es einen Unterordner „ccnet“ und hier die Datei „conf.conf“.



Diese über die rechte Maustaste mit Notepad++ öffnen



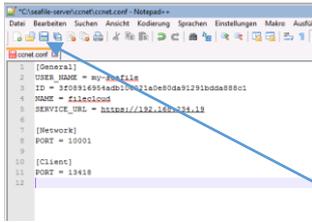
Und folgendes ändern:

NAME = filecloud (optional)

SERVICE_URL = <https://192.168.234.19> (ist die Adresse unter der man sich am Server anmeldet.

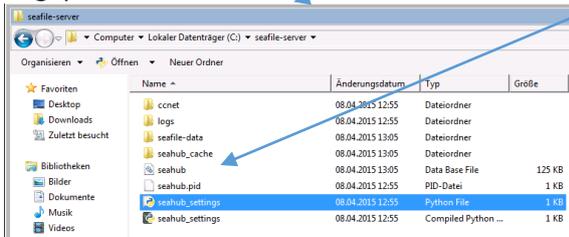
Wenn ein anderer Port als 443 verwendet werden soll muss er angehängt werden. Z.B. :450).

FILE_SERVER_ROOT = <https://meine.domain.de/seafhttp>

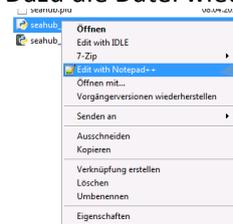


Anschließend die Änderungen speichern.

Als zweites muss im seafile-server Ordner noch die Datei „seahub_settings.py“ mit Notepad++ angepasst werden.

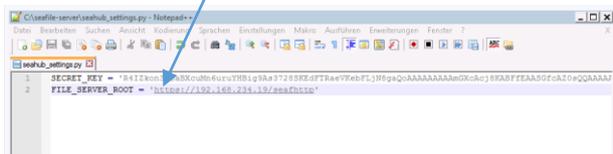


Dazu die Datei wieder über einen rechten Mausklick mit Notepad++ öffnen.



Und folgende Zeile unter der bestehenden einfügen:

FILE_SERVER_ROOT = 'https://192.168.234.19/seafhttp' (bei einer öffentlichen Adresse müsste diese wie auch an den anderen Stellen eingetragen werden. Das gilt auch für einen anderen Port als 443).



Anschließend die Datei in Notepad++ speichern.

Nachdem alles wieder geschlossen ist den Server über das Symbol in der Taskleiste neu starten



Und sehen ob der Zugriff über https im Browser funktioniert (<https://192.168.234.19>)

Die nun folgende Sicherheitszertifikatswarnung ist normal und kann mit „Laden der Website fortsetzen (nicht empfohlen)“ ignoriert werden.

In der Firewall muss noch der Port 443 TCP (oder der genutzte) freigegeben werden!

Um die Emailfunktion nutzen zu können muss in der Datei „seahub_settings.py“ noch folgendes angehängt werden (exemplarisch, funktioniert mit der Einstellung nur wenn der Provider keine SSL bzw. TLS-Verschlüsselung zwingend voraussetzt! Ansonsten muss „EMAIL_USE_TLS=True“ und bei „EMAIL_PORT=“ der entsprechende Port gesetzt werden). Die eigenen Angaben entsprechen den Daten die der Provider zur Verfügung gestellt hat.

```
EMAIL_USE_TLS = False
EMAIL_HOST = '127.0.0.1' # smtp server
EMAIL_HOST_USER = email@domain' # username and domain
EMAIL_HOST_PASSWORD = 'Passwort' # password
EMAIL_PORT = '25'
DEFAULT_FROM_EMAIL = EMAIL_HOST_USER
SERVER_EMAIL = EMAIL_HOST_USER
```

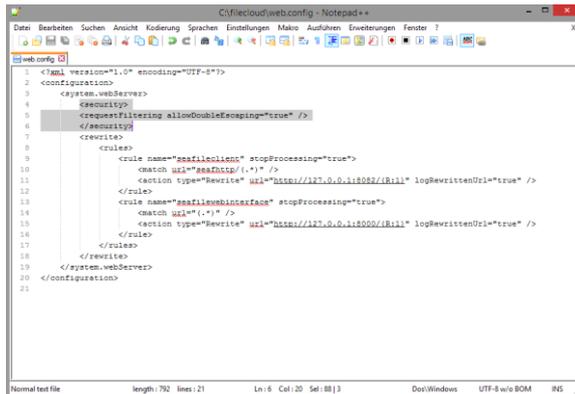
```
*D:\seafiler-server\seahub_settings.py - Notepad++
Datei Bearbeiten Suchen Ansicht Kodierung Sprachen Einstellungen Makro Ausführen Erweiterungen Fenster ?
seahub_settings.py
1 SECRET_KEY = 'o7tzM9HA4ngJmMhX25X7Cxr/7S7gdu9vZmjg5fWPFwUKEW+1sk+nzQoAAACgEkAA6L1VALj8KABFFeAAeL5VAJ0sQQAAAAF
2 FILE_SERVER_ROOT = '
3 EMAIL_USE_TLS = False
4 EMAIL_HOST = '127.0.0.1' # smtp server
5 EMAIL_HOST_USER = 'email@domain' # username and domain
6 EMAIL_HOST_PASSWORD = 'Passwort' # password
7 EMAIL_PORT = '25'
8 DEFAULT_FROM_EMAIL = EMAIL_HOST_USER
9 SERVER_EMAIL = EMAIL_HOST_USER
Python file length: 563 lines: 9 Ln: 9 Col: 31 Sel: 264 | 7 Dos\Windows UTF-8 w/o BOM INS
```

Wichtig!!!

Damit es beim Synchronisieren und Öffnen mit der Android-App keine Probleme mit Umlauten und Sonderzeichen (ä,ü,ö,ß,Ä,Ö,Ü) sowie Leerzeichen kommt muss in der „web.config“ (Pfad in dem Beispiel: C:\filecloud) von IIS noch folgendes eingetragen werden:

```
<security>
<requestFiltering allowDoubleEscaping="true" />
</security>
```

Das Ganze sieht dann z.B. so aus



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <configuration>
3   <system.webServer>
4     <security>
5       <requestFiltering allowDoubleEscaping="true" />
6     </security>
7     <rewrite>
8       <rules>
9         <rule name="msafileclient" stopProcessing="true">
10          <match url="msafileclient/*" />
11          <action type="Rewrite" url="https://127.0.0.1:5002/IR11" logRewrittenUrl="true" />
12        </rule>
13        <rule name="msafilewebinterface" stopProcessing="true">
14          <match url=".*" />
15          <action type="Rewrite" url="https://127.0.0.1:5002/IR11" logRewrittenUrl="true" />
16        </rule>
17      </rules>
18    </rewrite>
19  </system.webServer>
20 </configuration>
21
```

Nun müsste alles funktionieren!