



---

# dox42 Server

## V3.5.1-4.6

---

Dokumentation

---

---

# Zusammenfassung

In die diesem Dokument sind die Serviceschnittstelle sowie die Konfigurationseinstellung des **dox42Servers** beschrieben.

## Inhalt

Zusammenfassung.....	2
Inhalt.....	2
Dokumentinfo.....	3
Service-Schnittstelle .....	4
SOAP Schnittstelle.....	4
REST Schnittstelle .....	5
Service-Header.....	6
Azure Active Directory Header.....	6
HTTP Metadata Header .....	7
OutputActions.....	8
SaveAction.....	9
SharePointAction.....	10
Spezielle SharePoint und Managed Metadata Felder setzen.....	12
EmailAction .....	13
ReturnAction.....	18
PrintAction.....	19
dox42LIVEAction.....	19
AzureBlobStorageAction.....	20
Konfigurationsparameter.....	22
Beispiel .....	24
CORS Konfiguration.....	25
Azure Active Directory Zugriff.....	26
Custom Error Messages Beispiel .....	26
Hyphenation Dictionaries Beispiel .....	26
Maximale Request-Größe und Timeout.....	26
Lizenz Aktivierung.....	27
Installation.....	28
Upgrade.....	30
Support.....	30

## Dokumentinfo

Version: dox42 Server V3.5.1-4.6  
Autor: DI Christian Bauer, Lisa Pulsinger  
Datum: 15.02.2024

# Service-Schnittstelle

Der dox42 Server ist über eine SOAP- und eine REST-WebService Schnittstelle erreichbar.

## Operation: GenerateDocument/GenerateSlides

Die Operation lässt sich über **folgende Parameter** steuern:

- › **DocTemplate** (String)  
Pfad zum Dokumenttemplate  
(z.B. C:\doctemplates\solarantrag.docx, ~\templates\mobilfunkrechnung.docx, C:\doctemplates\salesreport.pptx)
- › **InputParams** (Key-Value String/String)  
Liste der Inputparameter und Werte:
  - › **ParamName:** Name des Parameters (nicht case-sensitive)
  - › **Value:** Wert des Parameters
- › **PostGenActions**  
Liste, der nach der Dokumentgenerierung abzuarbeitenden OutputActions:
  - › **ActionName:** Name der Action (z.B. SaveAction), nicht case-sensitive
  - › **Params:** Parameter der Action (wie InputParams)

## Operation: GenerateSpreadSheet

Die Operation lässt sich über **folgende Parameter** steuern:

- › **DataMap** (String)  
Pfad zum DataMap  
(z.B. C:\doctemplates\salesreport.dm, ~\templates\mobilfunkrechnung.dm)
- › **InputParams** (Key-Value String/String)  
Liste der Inputparameter und Werte:
  - › **ParamName:** Name des Parameters (schreibungsunabhängig)
  - › **Value:** Wert des Parameters
- › **PostGenActions**  
Liste, der nach der Dokumentgenerierung abzuarbeitenden OutputActions:
  - › **ActionName:** Name der Action (z.B. SaveAction), nicht case-sensitive
  - › **Params:** Parameter der Action (wie InputParams)

Die Parameter der einzelnen OutputActions sind in diesem Dokument beschrieben.

## SOAP Schnittstelle

URL: <http://www.YourDomain.com/dox42service.asmx>

WSDL: <http://www.YourDomain.com/dox42service.asmx?WSDL>

## REST Schnittstelle

URL: <http://www.YourDomain.com/dox42RestService.ashx>

REST und SOAP Schnittstelle haben identische Parameter und OutputActions. Bei der REST Schnittstelle werden alle Parameter als URL-Parameter übergeben. Ab Version 3.6.3 können alle Parameter auch als POST Parameter übergeben werden. Dabei gilt die folgende Syntax:

Parametername	Erfordernis	Auswirkung
<b>Operation</b>	Obligatorisch  GenerateDocument   GenerateSpreadSheet   GenerateSlides	Auswahl der aufzurufenden Operation
<b>DocTemplate</b>	Obligatorisch bei Operation=GenerateDocument oder GenerateSlides	Pfad zum Dokumenttemplate
<b>DataMap</b>	Obligatorisch bei Operation= GenerateSpreadSheet	Pfad zum DataMap
<b>InputParam.</b> <b>&lt;ParameterName&gt;</b>	Für jeden Inputparameter	Übergabe der Werte für Inputparameter, z.B.:  InputParam.KundenID=42
<b>&lt;OutputAction&gt;.</b> <b>&lt;ParameterName&gt;</b>	Für jeden Parameter jeder OutputAction	Übergabe der Werte für OutputAction-Parameter, z.B.: ReturnAction.Format=pdf
<b>ReturnAction.disp</b>	Optional  Inline   attachment	Steuert das Browser-Handling des zurückgelieferten Dokuments
<b>ReturnAction.Mode</b>	Optional  File (Standard)   Text	Bei Text wird der Inhalt des Dokuments direkt als Text im Browser ausgegeben. Das ermöglicht u.a. das dynamische Generieren von HTML Seiten.
<b>ReturnAction.fileName</b>	Optional, nur bei ReturnAction.disp = attachment sinnvoll	Name des zurückgelieferten Dokumentes
<b>RedirectURL</b>	Optional, nur ohne ReturnAction sinnvoll	Nach Generierung wird ein Redirect durchgeführt

<b>ReturnSavedFileInfo</b>	Optional (seit 4.5)	Bei true wird in einem REST Call die Info über gespeicherte Files als JSON zurückgegeben.
----------------------------	---------------------	---

**Achtung:** URL-Encoding für Sonderzeichen! ([http://www.w3schools.com/tags/ref\\_urlencode.asp](http://www.w3schools.com/tags/ref_urlencode.asp)):  
„Kunde.Name=Müller“ => „Kunde.Name=M%C3%BCller“

Mit der Syntax <%Datenquelle.Datenfeld%> übergeben Sie Ihren Aufrufen dynamische Werte.

#### Beispiel für einen REST-Aufruf:

[http://www.YourDomain.com/dox42RestService.ashx?  
Operation=GenerateDocument&  
DocTemplate=%7e%5cTemplates%5cSales\\_Report%5cSales\\_Report.doc &  
InputParam.SelectEmployee=5&InputParam.SelectYear=2023&  
ReturnAction.format=pdf&ReturnAction.fileName=SalesReport <%Employees.Name%>.pdf&Return  
Action.disp=attachment](http://www.YourDomain.com/dox42RestService.ashx?Operation=GenerateDocument&DocTemplate=%7e%5cTemplates%5cSales_Report%5cSales_Report.doc&InputParam.SelectEmployee=5&InputParam.SelectYear=2023&ReturnAction.format=pdf&ReturnAction.fileName=SalesReport <%Employees.Name%>.pdf&ReturnAction.disp=attachment)

Sie können auf unserer Website den [dox42 Server Designer runterladen](#), um Ihre REST und SOAP Aufrufe einfach zu konfigurieren.

#### HTTP Request-Body

Sollen umfangreiche Daten an das dox42 Service übertragen werden, so können diese als POST Parameter oder im Request-Body (ab Version 4.0) übertragen werden. Um den Request-Body auszulesen legen Sie in Ihrer DataMap einen InputParameter mit dem Namen „RequestBody“ an. An diesen InputParameter werden die Daten aus dem HTTP Request Body übergeben und können z.B. mit einer XML/JSON Datenquelle ausgelesen werden.

## Service-Header

Das dox42 SOAP Service lässt sich über einen Service-Header absichern. Dieser enthält Benutzername und Passwort. In der web.config kann die Verwendung des Service-Headers konfiguriert werden.

## Azure Active Directory Header

Das dox42 Service versteht einen Authorization Header, der ein Azure Active Directory Token enthält (ab Version 4.0).

Javascript Beispiel:

```
xhttp.setRequestHeader("Authorization", 'Bearer ' + token);
```

## HTTP Metadata Header und Base64 Metadata Header

Seit Version 4.6. können Sie bei dox42 Aufrufen einen String mit Datenfeldern im dedizierten HTTP Metadata Header "dox42MetaData" mitsenden. Der String könnte zum Beispiel so aussehen, der Header wird mit den ausgefüllten Datenfeldern zurückgeliefert:

```
{"sender": "<%User.mail%>"
```

```
"receiver": "<%Customer.mail%>"
```

```
"subject": "Here is your document <%quote.number%>"}
```

Seit Version 4.6.4. können Sie auch base64 codierte Metadata Header mitschicken. Benützen Sie dafür „dox42MetaDataMemberBase64“.

# OutputActions

Im Produkt dox42 Server sind folgende OutputActions enthalten:

- › **SaveAction:** Generiertes Dokument in einem Dateiordner speichern
- › **SharepointAction:** Generiertes Dokument in einer SharePoint-Dokumentenbibliothek speichern
- › **EmailAction:** Generiertes Dokument per E-Mail versenden
- › **ReturnAction:** Generiertes Dokument über die Webservice-Schnittstelle zurückliefern
- › **PrintAction:** Generiertes Dokument drucken
- › **dox42LiveAction:** Wird verwendet für Aufrufe mit dem Produkt dox42 Live
- › **AzureBlobStorageAction:** Generiertes Dokument wird in Azure Blob Storage gespeichert.
- › **DocuSignAction:** Generiertes Dokument wird mit DocuSign elektronisch signiert. Wird in Dokumentation dox42 DocuSign beschrieben
- › **DropBoxAction:** Wird in Dokumentation dox42 DropBox Action beschrieben
- › **SQLAction:** Generiertes Dokument wird in SQL Datenbank gespeichert. Wird in Dokumentation SQL\_OutputAction beschrieben.
- › **CustomOutputAction:** Sie können auch selbst eine dox42 OutputAction implementieren und verwenden. Custom Output Actions müssen in der web.config registriert werden (analog Custom Data Sources) und folgendes Interface implementieren:  
`dox42CustomOutputActionKit.ICustomOutputAction`

Die oben erwähnten ergänzenden Dokumentationen finden Sie unter

[https://www.dox42.com/de/Resources?filter\[\]=dokumentation&](https://www.dox42.com/de/Resources?filter[]=dokumentation&)

Im Folgenden sind die Parameter der OutputActions beschrieben. Die Parameternamen werden nicht case sensitive behandelt. In allen Parameterwerten können Datenfelder verwendet werden (siehe Tabelle für Ausnahmen).

Beispiel: FileName = „Antrag\_<%Buerger.Name%>.pdf“

Seit Version 4.6. steht Ihnen der Parameter „Condition“ für Ihre Output Action zur Verfügung. Bei „false“ wird die gewählte Output Action nicht ausgeführt.



## SaveAction

Parametername	Erfordernis	Auswirkung
<b>FileName</b>	obligatorisch	Speichernamen, unter dem das Dokument abgelegt wird. z.B. C:\Dokumente\SalesReport.pdf  Über die Dateierweiterung wird der Dateityp bestimmt (.docx, .xlsx, .pptx, .pdf, html,...). Siehe weiter unten.
<b>SpecialFormat</b> (nur dox42ForDocuments)	optional	<b>Bei PDF kann das PDF-Format bestimmt werden, Werte:</b> PDFA1b und PDFForm. Bei PDFForm bleiben Formularfelder im PDF erhalten.  <b>Bei HTML gibt es folgende Optionen:</b> HTMLCode Der Text des dox42-Templates wird als HTML-Code interpretiert.  HTMLZIP Eine ZIP Datei mit der HTML Datei und den enthaltenen Images wird erzeugt. (Ab Version 3.6.3).  <b>Bei Textdateien (*.txt) kann eine alternative Extension angegeben werden.</b> Damit können Metadaten-files für Archivsysteme erzeugt werden, Beispiel: FileName = Order_<%Cust.OderNo%>.txt SpecialFormat = jpl Ergebnis: Order_42.jpl (seit Version 3.5)
<b>metadatafilepath</b>	optional	Erstellt ein zusätzliches Metadatenfile. Beispiel: C:/dox42Server/generateddocuments/metadata.xml
<b>metadatafilecontent</b>	optional	Definiert den Inhalt für das Metadatenfile.
<b>Sign</b>	optional	Wert: true Bei PDF kann eine digitale Signatur aufgebracht werden (nur dox42ForDocuments). Das Zertifikat und das dazugehörige Passwort müssen in der web.config definiert sein.
<b>Condition</b>	optional (seit V. 4.6)	Bei „false“ wird die Output Action nicht ausgeführt.

### Mögliche Ausgabeformate

Word: doc, docx, docm, dot, dotx, pdf, html, txt, jpg, png, tiff, bmp, rtf, odt  
 Excel: xls, xlsx, xlsxm, xlt, xltm, xlsb, txt, csv, html, pdf, xml, tiff, ods  
 PowerPoint: pptx, pps, pdf, tiff

## SharePointAction

Parametername	Erfordernis	Auswirkung
<b>Site</b>	obligatorisch	Die Sharepoint-Site z.B. <a href="http://mysite.com">http://mysite.com</a>
<b>Library</b>	obligatorisch	Die Sharepoint-Dokumentenbibliothek bzw. der Name des Rootfolders
<b>FileName</b>	obligatorisch	Speichernamen, unter dem das Dokument abgelegt wird. z.B. SalesReport.pdf  Über die Dateierweiterung wird der Dateityp bestimmt (.docx, .xlsx, .pptx, .pdf, html,...).
<b>Folder</b>	optional	Dokument in diesen Ordner speichern.
<b>Rootfolder</b>	Optional (seit 4.4.1.7)	Der interne Name der SharePoint- Dokumentenbibliothek. Anzuwenden, wenn sich interner Name und Bibliotheksname unterscheiden.
<b>SpecialFormat</b> (nur dox42ForDocuments)	optional	Bei PDF kann das PDF-Format bestimmt werden, mögliche Werte: PDFA1b und PDFForm. Bei PDFForm bleiben Formularfelder im PDF erhalten.  Bei HTML kann die Option: HTMLCode gesetzt werden, dann wird der Text des generierten Dokuments als HTML-Code interpretiert.
<b>Sign</b>	optional	Wert: true  Bei PDF kann eine digitale Signatur aufgebracht werden (nur dox42ForDocuments) Das Zertifikat und das dazugehörige Passwort müssen in der web.config definiert sein.
<b>Field1.Name</b> <b>Field1.Value</b>	optional	Fields ermöglichen das Setzen von Spalten in der SharePoint-Dokumentenbibliothek. Sie können mehrere Felder setzen: Field1, Field2,... Benützen Sie als Feldname den SharePoint internal field name.  Tipp: Sie können den internal field name herausfinden, indem Sie die Liste mit der dox42

		SharePoint Datenquelle öffnen und die Checkbox „use internal field names“ setzen.
<b>ContentType</b>	optional	Name des zu setzenden Content-Typs
<b>DocumentDataField</b>	optional	Wird dieses Feld angegeben, so wird nicht das generierte Dokument in die Sharepoint-Dokumentenbibliothek gespeichert, sondern das Dokument aus dem angegeben dox42 Datenfeld geladen (Z.B.: Data.MyDokument) und in die Sharepoint- Dokumentenbibliothek gespeichert. Das DocumentDataField kann einen Pfad, eine URL oder ein binäres Dokument enthalten.
<b>CheckInMessage</b>	optional	Wird dieses Feld angegeben, checkt der Server das Dokument nach dem Speichern mit der angegebenen Nachricht ein. Gibt es bereits ein gleichnamiges Dokument in der Library, so wird dieses vorher automatisch ausgecheckt.
<b>Username</b>	optional	Wenn nicht angegeben, wird der Service Account des dox42 Servers als SharePoint User verwendet.
<b>Password</b>	optional	
<b>SharePoint_online</b>	optional	true/false Muss bei SharePoint online (Office 365) gesetzt werden - alternativ zu <b>Azure_AD</b> .
<b>SharePoint_FBA</b>	optional	true/false Muss bei Sites mit aktivierter Forms based Authentication gesetzt werden.
<b>Azure_AD</b>	optional	true/false Im Request muss ein Authorization Header mit einem Azure Active Directory Token enthalten sein, welches zur Authentisierung verwendet wird - alternative zu <b>SharePoint_online</b> . (Ab Version 4.0).
<b>retry_count</b>	optional	Eine natürliche Zahl $\geq 0$ Definiert, dass die SharePointAction das Speichern im Fehlerfall (z.B.: Timeout) mehrfach versucht und wie oft. Default = 0 (no retry). (Ab Version 4.1.3).

<b>Condition</b>	optional (seit V. 4.6)	Bei „false“ wird die Output Action nicht ausgeführt.
------------------	------------------------	--

## Spezielle SharePoint und Managed Metadata Felder setzen

Bei Auswahlfeldern (Choice) setzen Sie einfach den Wert.

Bei Auswahlfeldern (Choice), die mehrere Werte enthalten können (Multi Value), setzen Sie diese durch „;#“ getrennt, also z.B.: „rot;#gelb;#blau“

Bei Lookup/Person/Group-Felder setzen Sie die ID.

Bei Lookup/Person/Group-Felder, die mehrere Werte enthalten können (Multi Value), müssen Sie ID und Wert angeben: „ID;#Wert“ , also z.B.: „3;#Hawaii;#9;#Vienna“. Tatsächlich berücksichtigt wird dabei aber nur die ID.

## EmailAction

Parametername	Erfordernis	Auswirkung
<b>Mode</b>	optional  <b>mögliche Werte:</b>  TxtEMail HtmlEMail AttachmentTxtBody Attachment (Standardwert)	Bestimmt ob das generierte Dokument als Text oder HTML für den E-Mail-Text verwendet, oder als Attachment mitgesendet wird.  TxtEMail und HtmlEMail sind nur bei GenerateDocument sinnvoll zu verwenden.
<b>FileName</b>	Obligatorisch bei Mode = Attachment, sonst ignoriert	Name des Attachments  Über die Dateierweiterung wird der Dateityp bestimmt (.docx, .xlsx, .pptx, .pdf, html,...).
<b>SpecialFormat</b> (nur dox42ForDocuments)	optional	Bei PDF kann das PDF-Format bestimmt werden, Werte: PDFA1b und PDFForm. Bei PDFForm bleiben Formularfelder im PDF erhalten.  Bei HTML kann die Option: HTMLCode gesetzt werden, dann wird der Text des generierten Dokuments als HTML-Code interpretiert.
<b>Sign</b>	optional	Wert: true  Bei PDF kann eine digitale Signatur aufgebracht werden (nur dox42ForDocuments). Das Zertifikat und das dazugehörige Passwort müssen in der web.config definiert sein.
<b>Receiver</b>	obligatorisch	Empfänger E-Mail-Adressen  bei mehreren Adressen durch ein Komma oder Strichpunkt getrennt
<b>ReceiverCC</b>	optional	Carbon Copy Empfänger
<b>ReceiverBCC</b>	optional	Blind Carbon Copy Empfänger
<b>Sender</b>	optional	Absender E-Mail-Adresse

Parametername	Erfordernis	Auswirkung
		Wird diese nicht angegeben, so wird die in der web.config konfigurierte Adresse verwendet.
<b>MailServerUsername</b>	optional	Kann zusätzlich zu Sender angegeben werden. Dann wird für die Authentifizierung am Mailserver dieser Username verwendet. Seit Version 3.6.2.4
<b>SenderPassWord</b>	entfällt, wenn Sender oder MailServerUsername nicht angegeben, sonst obligatorisch	Passwort des Absenders für sein E-Mail-Konto oder MailServerUsername auf den, in der web.config konfigurierten, E-Mail-Server.
<b>Header</b>	obligatorisch	E-Mail-Betreff
<b>Body</b>	optional	E-Mail-Text als reiner Text (nur bei mode = Attachment)
<b>HtmlBody</b>	optional	E-Mail-Text als HTML (nur bei mode = Attachment)
<b>BodyTemplate</b>	optional	Ein dox42 Word Template mit dem der Email-Body generiert wird. Dieses Template muss die gleiche Datamap als das Attachment benutzen. (nur bei mode = Attachment) Seit V 3.5.1 Bei mode = AttachmentTxtBody wird der E-Mail Body im Txt Format generiert. (Seit V 4.4.1.5)
<b>StaticAttachment1.Path</b>	optional (alternativ zu Base64) z.B. C:\Bilder\MeinBild.jpg	StaticAttachments ermöglichen das Mitsenden von statischen Beilagen. Sie können mehrere Beilagen mitsenden: StaticAttachment2... StaticAttachment3... Es kann der Pfad zu einer Datei oder die Beilage Base64-codiert übergeben werden.
<b>StaticAttachment1.Base64</b>	pg  Base64 codierte	
<b>StaticAttachment1.Name</b>	Datei optional (alternativ zu Path)  obligatorisch bei Base64, sonst optional z.B. MeinBild.jpg	

Parametername	Erfordernis	Auswirkung
<b>MailServer</b>	optional	Damit lässt sich der in der Web.Config eingestellte Mailserver übersteuern.
<b>MailServerPort</b>	optional Ab Version 3.4.3.10	Damit lässt sich der in der Web.Config eingestellte Mailserver Port übersteuern. Verwenden Sie zum Beispiel die Standard-Ports 25 oder 587.
<b>SSL</b>	optional Ab Version 3.5	true oder false
<b>SaveMSGFile</b>	optional Ab Version 3.5.0.19	Dateipfad unter dem das gesendete Mail als MSG-Datei gespeichert werden soll. Z.B.: C:\Windows\Temp\Order_<%Customer.ID%>.msg
<b>SendMode</b>	optional Ab Version 3.6	Mögliche Werte: send, save, return <b>send:</b> Standardwert, Mail wird sofort versendet.  <b>save:</b> Mail wird nicht versendet, sondern als Draft unter dem in SaveMSGFile angegebenen Pfad gespeichert. Parameter SaveMSGFile muss angegeben sein.  <b>return:</b> Mail wird nicht versendet, sondern als Draft in der Return-Message zurückgeliefert. Ab Version 3.6.3.
<b>MailMethod</b>	optional ab Version 4.3.3	Damit wird die Methode für den E-Mail-Versand gewählt. Mögliche Werte: exchangeonline, aspose, mailkit, onmailkiterrortryaspose, onasposeerrortrymailkit  <b>Exchangeonline: Muss verwendet werden, wenn E-Mails via Microsoft Exchange Online versendet werden</b>  <b>onasposeerrortrymailkit:</b> wenn die Mail Methode Aspose einen Error wirft, werden Mails via Mailkit versendet.  Wenn Sie den Parameter „MailMethod“ verwenden, empfehlen wir auch den MailServerPort zu konfigurieren.

Parametername	Erfordernis	Auswirkung
<b>AzureADAppID</b>	optional (seit V. 4.4.1.7)	Die ID der App Registrierung in Azure AD, mit der E-Mails über Exchange Online verschickt werden können. (kann auch in der Web.config/dox42 Online MAUI konfiguriert werden)
<b>AzureADTenantID</b>	optional (seit V. 4.4.1.7)	Die ID des Tenants in dem die obige App Registrierung in Azure AD, mit der E-Mails über Exchange Online verschickt werden können, registriert wurde (kann auch in der Web.config/dox42 Online MAUI konfiguriert werden)
<b>AzureADClientSecret</b>	optional (seit V. 4.4.1.7)	Das Client Secret der obigen AzureADAppId (kann auch in der Web.config/dox42 Online MAUI konfiguriert werden)
<b>Exchangeonlineurl</b>	optional (seit V. 4.4.1.7)	Beinhaltet den Standardwert <a href="https://outlook.office365.com/EWS/Exchange.asmx">https://outlook.office365.com/EWS/Exchange.asmx</a>
<b>Condition</b>	optional (seit V. 4.6)	Bei „false“ wird die Output Action nicht ausgeführt.

Um bei E-Mail Adresse einen Alias zu verwenden benutzen Sie folgende Syntax (seit Version 3.6.2.5):  
MyCompany Sales {[sales@mycompany.com](mailto:sales@mycompany.com)}

MyCompany Sales <[sales@mycompany.com](mailto:sales@mycompany.com)>, funktioniert auch, kann aber beim REST-Service zu Fehlern führen.

### Exchange Online

Um E-Mails über Exchange Online zu versenden, muss eine App Registrierung in Azure Active Directory hinterlegt sein. Weitere Informationen zur App Registrierung finden Sie in unserer SharePoint Dokumentation:

[https://www.dox42.com/de/Resources?filter\[\]=dokumentation.SharePoint&](https://www.dox42.com/de/Resources?filter[]=dokumentation.SharePoint&)

Außerdem müssen im Manifest der App Registrierung die notwendigen API-Berechtigungen hinzugefügt und danach von einem Admin freigegeben werden.

Unter „RequiredAccess“ im Manifest in Ihrer Azure AD App Registrierung folgendes hinzufügen:

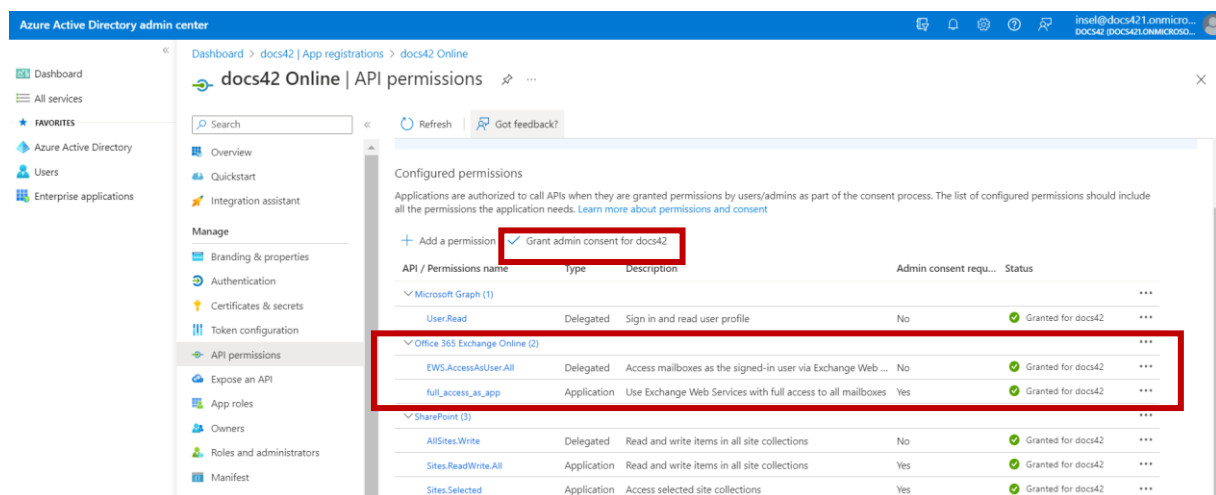


```

{
  "resourceAppId": "00000002-0000-0ff1-ce00-000000000000",
  "resourceAccess": [
    {
      "id": "3b5f3d61-589b-4a3c-a359-5dd4b5ee5bd5",
      "type": "Scope"
    },
    {
      "id": "dc890d15-9560-4a4c-9b7f-a736ec74ec40",
      "type": "Role"
    }
  ]
}

```

Danach schein folgendes in den API Permissions auf. Hier noch auf „Grant admin consent“ klicken.



Sobald Sie die notwendigen Einstellungen getroffen und auch ein Clientsecret angelegt haben, müssen Sie im Azure AD Abschnitt in Ihrer Web.config, bzw. zu den SharePoint Sites Ihrer dox42 Online MAUI, die Seite „<https://outlook.office365.com/>“ mit der App ID, Tenant ID und dem Client Secret hinzufügen.

## ReturnAction

Parametername	Erfordernis	Auswirkung
<b>Format</b>	obligatorisch	Dateityp (.docx, .xlsx, .pptx, .pdf, html, emf (since V.4.6.2)).
<b>Filename</b>	optional	Definiert den Dateinamen. Z. B. Contract.pdf. Für dynamische Dateinamen, folgende Syntax verwenden: Contract_<%employee.name%>.pdf
<b>Mode</b>	optional <b>mögliche Werte:</b>  File (Standard) Text	Definiert ob das generierte Dokument als Datei (Mode=File) in ServiceResult.GeneratedDocs oder als Text (Mode=Text) in ServiceResult.ResultMessage retourniert wird.
<b>SpecialFormat</b> (nur dox42ForDocuments)	optional	Bei PDF kann das PDF-Format bestimmt werden, Werte:  PDFA1b und PDFForm. Bei PDFForm bleiben Formularfelder im PDF erhalten.  Bei HTML kann die Option: HTMLCode gesetzt werden, dann wird der Text des generierten Dokuments als HTML-Code interpretiert.  EMFZIP (seit V.4.6.2.)
<b>Sign</b>	optional	Wert: true  Bei PDF kann eine digitale Signatur aufgebracht werden (nur dox42ForDocuments). Das Zertifikat und das dazugehörige Passwort müssen in der Web.config definiert sein.
<b>Condition</b>	optional (seit V. 4.6)	Bei „false“ wird die Output Action nicht ausgeführt.
<b>Numberofcopies</b>	Optional (seit V. 4.6.2.)	z. B. 1, 2 oder 3. Gibt an wie viele Kopien innerhalb eines Files erstellt werden sollen

## PrintAction

Parametername	Erfordernis	Auswirkung
<b>PrinterName</b>	optional	Der zu verwendende Drucker
<b>Collate</b>	optional	Sortieren? Werte: true/false
<b>Copies</b>	optional	Anzahl Kopien
<b>Duplex</b>	optional	Duplexdruck Werte: Vertical, Horizontal, Simplex
<b>FromPage</b>	optional	Drucken von Seite ...
<b>ToPage</b>	optional	...bis Seite
<b>Condition</b>	optional (seit V. 4.6)	Bei „false“ wird die Output Action nicht ausgeführt.

## dox42LIVEAction

Die dox42LIVEAction verwendet eine Output Konfiguration aus einer dox42LIVE.config Datei um Output Action auszuführen. Der Aufbau einer dox42LIVE.config ist in der dox42 LIVE Dokumentation beschrieben.

Um eine dox42LIVEAction in einem dox42 SOAP oder REST Service zu verwenden speichern Sie die dox42LIVE.config Datei in dasselbe Verzeichnis als dox42RestService.ashx und dox42Service.asmx , normalerweise: C:\dox42Server.

Ab Version 4.1.3.

Parametername	Erfordernis	Auswirkung
<b>LiveActionName</b>	obligatorisch	Der LiveActionName wie in der dox42LIVE.config definiert.
<b>Condition</b>	optional (seit V. 4.6)	Bei „false“ wird die Output Action nicht ausgeführt.

## AzureBlobStorageAction

Mit der Azure BlobStorageAction können Sie Ihre generierten Dokumente in einem Azure Blob Speicher Container speichern. Diese Output Action ist ab Version 4.3.1 verfügbar. Dafür müssen Sie im Azure Portal einige Einstellungen treffen:

In Ihrer dox42 Azure Active Directory App Registrierung müssen Sie bei den API-Berechtigungen delegierte Berechtigungen für Azure Storage user impersonation hinterlegen und als Admin freigeben. Weitere Infos zu dox42 Azure AD App Registrierungen finden Sie in unseren dox42 SharePoint und D365 CE Dokumentationen:

[https://www.dox42.com/Resources?filter\[\]=dokumentation.SharePoint&filter\[\]=dokumentation.CE-CRM&](https://www.dox42.com/Resources?filter[]=dokumentation.SharePoint&filter[]=dokumentation.CE-CRM&)

Außerdem müssen Sie in den Access Control (IAM) Einstellungen Ihres Azure Storage Accounts Ihre dox42 Azure Active Directory App Registrierung als Owner und Contributor hinzufügen:

dox42blobtest | Access Control (IAM) ...

Storage account

Search

+ Add Download role assignments Edit columns Refresh Remove Got feedback?

Check access **Role assignments** Roles Deny assignments Classic administrators

Number of role assignments for this subscription 6 4000

Search by name or email Type: All Role: All Scope: All scopes Group by: Role

6 items (1 Users, 4 Service Principals, 1 Managed Identities)

Name	Type	Role	Scope	Condition
Contributor				
<input type="checkbox"/> 52a732f6981343e3865900da	App	Contributor	Subscription (Inherited)	None
<input checked="" type="checkbox"/> Storage Blob Data Contributor				
<input type="checkbox"/> AzureBlobStorageTest	App	Storage Blob Data Contributor	This resource	Add
<input type="checkbox"/> dox42 Azuro Server	App	Storage Blob Data Contributor	This resource	Add
Storage Blob Data Owner				
<input checked="" type="checkbox"/> Storage Blob Data Owner				
<input type="checkbox"/> AzureBlobStorageTest	App	Storage Blob Data Owner	This resource	Add
<input type="checkbox"/> dox42 Azuro Server	App	Storage Blob Data Owner	This resource	Add

Fügen Sie danach noch <https://storage.azure.com> sowie Ihre AppID, Ihre Tenant ID und das Client Secret zum Azure AD Abschnitt in Ihrer web.config hinzu. Wenn Sie dox42 Online verwenden, hinterlegen Sie dies in den SharePoint Online Sites Ihrer dox42 Online MAUI. In Ihren dox42 Aufrufen verwenden Sie folgende Parameter:

Parametername	Erfordernis	Auswirkung
<b>AzureBlobStorageAccountname</b>	obligatorisch	Der Name des Azure Speicherkontos.
<b>AzureBlobStorageContainername</b>	obligatorisch	Der Name des Azure Speicher Containers.
<b>UploadToAzurePath</b>	obligatorisch	Der Filename (kann auch Verzeichnisse beinhalten) and die Format-Extension: Example: VerzeichnisA/Document.docx
<b>Condition</b>	optional (seit V. 4.6)	Bei „false“ wird die Output Action nicht ausgeführt.

# Konfigurationsparameter

Folgende Parameter können für den **dox42** Server in den AppSettings der web.config konfiguriert werden:

- › **LicenseFilePath:** Gibt den Pfad des Speicherortes an, wo der Server das License-File ablegen soll. Ist kein License-File vorhanden, nimmt der **dox42** Server Kontakt mit dem **dox42** License-Server auf, um den LicenseKey zu verifizieren. Gelingt dies, so schreibt er das License-File auf den konfigurierten Pfad.
- › **LicenseKey:** LicenseKey des **dox42** Servers
- › **MailServer:** E-Mail-Server, der zum Versenden von E-Mails genutzt wird (optional)
- › **MailServerPort:** E-Mail-Server Port (optional)
- › **dox42ServerMail:** Standard Absenderadresse des **dox42** Services (optional)
- › **MailServerPassWord:** Passwort zum **dox42ServerMail** (optional)
- › **SendErrorsToMail:** Standard E-Mail-Adresse für Fehlermeldungen (optional)
- › **Certificate:** zum Signieren verwendetes Zertifikat (optional)
- › **CertificatePassWord:** Passwort des Zertifikates (optional)
- › **CheckServiceHeader (true/false):** Gibt an, ob der Service-Header ausgewertet werden soll. Bei "false" muss kein Service-Header mitgesendet werden.
- › **UserPermissionODBCConnectionString:** ODBC ConnectionString auf eine Datenbank, welche User-Permissions enthält  
(Pflicht, wenn CheckServiceHeader = true)
- › **CheckUserPermissionSQL:** Select Statement, welches auf der UserPermissionODBCConnectionString-Datenbank ausgeführt wird. Das Statement muss 1 oder "true" für erfolgreiche Autorisierung des Users zurückliefern. Im Statement werden die Tags %username% und %password% gegen die im Header mitgelieferten Werte ausgetauscht.  
(Pflicht, wenn CheckServiceHeader = true)
- › **ThrowExceptionOnError (true/false):**  
true: Bei einem Fehler retourniert das Service eine Exception.  
false (oder nicht vorhanden): Fehlermeldungen werden in der ResultMessage übergeben.  
Dieser Konfigurationsparameter steht ab Build 3.3.0.8 zur Verfügung
- › **CustomErrorMessageFile:**  
Pfad eines XML Files mit den Custom Error Messages (optional). Mittels Custom Error Messages ist es möglich Fehlermeldungen, die einen definierten Text beinhalten, durch andere Fehlermeldungen zu ersetzen. Ein Beispiel für ein solches File finden Sie im Anhang.
- › **HyphenationDictionariesFile:**  
Pfad eines XML Files mit Hyphenation Dictionaries (optional). Mittels Hyphenation Dictionaries ist es möglich die automatische Silbentrennung zu steuern. Ein Beispiel für ein solches File finden Sie im Anhang. (Ab Version 3.4.3.10)
- › **CheckTrustedTemplateLocations (true/false):**  
**true:** Bei jedem Aufruf wird geprüft, ob der Pfad/Link des Templates bzw. der Datamap mit einem Pfad/Link in der Sektion „trustedTemplateLocations“ beginnt. Ist das nicht der Fall, wird der Call abgebrochen.  
**false** (oder nicht vorhanden): Keine Prüfung.  
**ACHTUNG: Das Ausschalten dieser Option ermöglicht jeden, der Ihr dox42 Service**

**aufrufen darf, beliebige Templates/DataMaps auch von unbekanntem Internet URLs auf Ihrem Server zu generieren! Wir empfehlen daher DRINGEND diese Option immer auf true einzustellen.**

Dieser Konfigurationsparameter steht ab Build 3.3.4 zur Verfügung

- › **TempFolder:**  
Pfad auf ein Verzeichnis, in dem der dox42 Server temporäre Dateien ablegen soll. Wird der TempFolder nicht angegeben, so wird das Standard Windows Temp-Verzeichnis verwendet.  
Dieser Konfigurationsparameter steht ab Version 3.5 zur Verfügung
- › **RESTInterfaceReturnsFullErrorInfo (true/false):**  
Bei true wird beim REST Call umfangreichere Fehlerinformation ausgegeben.  
Dieser Konfigurationsparameter steht ab Version 3.6.1 zur Verfügung
- › **RESTInterfaceReturnsHTTPError500 (true/false):**  
Bei true wird beim REST Call bei einem Fehler der http Statuscode 500 gesetzt.  
Dieser Konfigurationsparameter steht ab Version 4.0.1 zur Verfügung
- › **SOAPServiceReturnsFullErrorInfo(true/false):**  
Bei true wird beim SOAP Call bei einem Fehler umfangreichere Fehlerinformation ausgegeben. Dieser Konfigurationsparameter steht ab Version 4.1.1.0 zur Verfügung
- › **SaveActionEnabled(true/false):**  
Bei true können Sie die dox42 Save Action verwenden, um generierte Dokumente am Server zu speichern. Dieser Konfigurationsparameter steht ab Version 4.3.2.3 zur Verfügung
- › **RedirectURLEnabled (true/false):**  
Bei true können Sie Ihren REST Calls eine Redirect URL mitsenden, zu der nach Generierung weitergeleitet wird: &RedirectURL=https://www.yourURL.com, bei false wird nicht zu einer URL weitergeleitet. Dieser Konfigurationsparameter steht ab Version 4.3.2.3 zur Verfügung
- › **MailMethod:** (ab Version 4.3.3)  
ExchangeOnline: Muss verwendet werden, wenn E-Mails über Exchange Online verschickt werden (siehe weitere Infos im Abschnitt E-Mail Action)  
onaspouseerrortrymailkit: E-Mails werden via Aspose Methode versendet, ist dies nicht erfolgreich, werden Mails über MailKit Methode versendet.  
aspose: E-Mails werden via Aspose versendet.  
onmailkiterrortryaspose: E-Mails werden via MailKit Methode versendet, ist dies nicht erfolgreich, werden Mails über Aspose Methode versendet.  
mailkit: E-Mails werden via MailKit versendet.
- › **LoadLocalFilesFromTrustedTemplateLocationOnly (true/false):** (Verfügbar seit Version 4.4.1)  
Bei true können Dateien in Datenfeldern und dynamischen Feldern nur aus definierten Trusted Template Locations geladen werden.
- › **AADLibrary (MSAL/ADAL):** (Verfügbar seit Version 4.6.1)  
Entscheidet welche Microsoft Authentication Library verwendet wird. MSAL ist der hinterlegte Standardwert.

## Beispiel

```
<appSettings>
  <add key="LicenseFilePath" value="c:\dox42\dox42WebService\License" />
  <add key="LicenseKey" value="..." />
  <add key="MailServer" value="smtp.1und1.de"/>
  <add key="dox42ServerMail" value="testserver@dox42.com"/>
  <add key="MailServerPassWord" value="..." />
  <add key="SendErrorsToMail" value="monitoring@dox42.com"/>
  <add key="MailServerPort" value="587"/>
  <add key="MailMethod" value="onaspouseerrortrymailkit"/>
  <add key="Certificate" value="c:\dox42\dox42WebService\Certs\Certificate.pfx"/>
  <add key="CertificatePassWord" value="..." />
  <add key="CheckServiceHeader" value="true"/>
  <add key="UserPermissionODBCConnectionString"
    value="Driver={SQL Server Native Client 10.0};Server=...;Database=...;Uid=...;Pwd=...;" />
  <add key="CheckUserPermissionSQL"
    value="SELECT COUNT(*) FROM Users where Username =
    '%username%' and Password = '%password%'" />
  <add key="ThrowExceptionOnError" value="false"/>
  <add key="CustomErrorMessageFile" value="C:\dox42Server\CustomErrors.xml"/>
  <add key="CheckTrustedTemplateLocations" value="true"/>
  <add key="TempFolder" value="C:\dox42Server\WorkingDir"/>
  <add key="RESTInterfaceReturnsFullErrorInfo" value="false"/>
  <add key="RESTInterfaceReturnsHTTPError500" value="false" />
  <add key="SOAPServiceReturnsFullErrorInfo" value="false" />
  <add key="SaveActionEnabled" value="true" />
  <add key="RedirectURLEnabled" value="true" />
  <add key="LoadLocalFilesFromTrustedTemplateLocationOnly" value="false" />
</appSettings>
```

Custom Data Sources werden in einer eigenen Sektion namens „customDataSources“ registriert.

```
<customDataSources>
  <add key="CSVDataEngine.CSVDataSourceParser"
    value="CSV Datenquelle;c:\dox42\dox42WebService\bin\CSVDataEngine.dll;" />
</customDataSources>
```

Custom Output Actions Sources werden in einer eigenen Sektion namens „customOutputActions“ registriert.

```
<customOutputActions>
  <add key="MyCustomOutputActions.MyOutputAction"
    value=" MyOutputAction; c:\dox42\dox42WebService\bin\MyCustomOutputActions.dll;" />
</customOutputActions>
```

Trusted Template Locations werden in einer Sektion namens „trustedTemplateLocations“ erfasst.

```
<trustedTemplateLocations>
  <add key="local" value="C:\dox42Server\templates"/>
  <add key="sharepoint" value="http://www.mysharepoint.mycompany.com/dox42Templates"/>
</trustedTemplateLocations>
```



In der NLog.config-Datei können Log Einstellungen vorgenommen werden. Die Konfiguration des verwendeten Logging-Frameworks NLog ist unter [www.nlog-project.org](http://www.nlog-project.org) dokumentiert.

## CORS Konfiguration

Wird der dox42 Server z.B. per Javascript XMLHttpRequest aus einer Web-Applikation unter einer anderen URL als die Applikation selbst aufgerufen, kann es zu einem Cross-Origin Resource Sharing (CORS) Fehler kommen.

### Beispiel:

Aufrufende Web-Applikation: <https://yourcompany.sharepoint.com/yoursite>

dox42 Server: <https://dox42server.yourcompany.com>

Um dies zu ermöglichen können Sie den Zugriff in der web.config ermöglichen, Beispiel:

```
<system.webServer>
  <httpProtocol>
    <customHeaders>
      <clear />
      <add name="Access-Control-Allow-Origin" value="*" />
      <add name="Access-Control-Allow-Headers"
          value="Content-Type, Accept, Authorization" />
      <add name="Access-Control-Allow-Methods" value="GET,POST,OPTIONS" />
      <add name="Access-Control-Expose-Headers"
          value="Content-Type, Content-Disposition, Accept" />
    </customHeaders>
  </httpProtocol>
</system.webServer>
```

### Siehe auch:

<https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>

<http://www.w3.org/TR/cors/>

## Azure Active Directory Zugriff

Diese Einstellung ist nötig damit der dox42 Server auf Templates, welche in SharePoint Online verwaltet werden, zugreifen kann. (Ab Version 4.0.)

### ConfigSection:

```
<configSections>
    . . .
    <section name ="azureAD" type ="dox42.Core.Utils.AzureADSectionHandler,dox42Core"/>
</configSections>
```

### Azure AD Section:

```
<azureAD>
  <add resource365="https://mytenant.sharepoint.com"
        appID="..."
        tenant="..."
        clientKey="*****" />
</azureAD>
```

## Custom Error Messages Beispiel

```
<?xml version="1.0" encoding="utf-8"?>
<CustomErrorMessagesDS xmlns="http://dox42.com/CustomErrorMessagesDS.xsd">
  <CustomErrorMessage>
    <Test2SearchInExc>Error uploading file http://mysharepoint</Test2SearchInExc>
    <NewMessage>
SharePoint-farm not responding, please contact your administrator</NewMessage>
  </CustomErrorMessage>
</CustomErrorMessagesDS>
```

Wenn Sie dieses File mit einem Text-Editor bearbeiten, achten Sie auf das Speichern im Encoding UTF-8.

## Hyphenation Dictionaries Beispiel

```
<?xml version="1.0" standalone="yes"?>
<HyphenationDictionary xmlns="http://www.dox42.com/HyphenationDictionary.xsd">
  <Dictionary>
    <Culture>de-CH</Culture>
    <DictionaryPath>C:\dox42Server\hyph_de_CH.dic</DictionaryPath>
  </Dictionary>
</HyphenationDictionary>
```

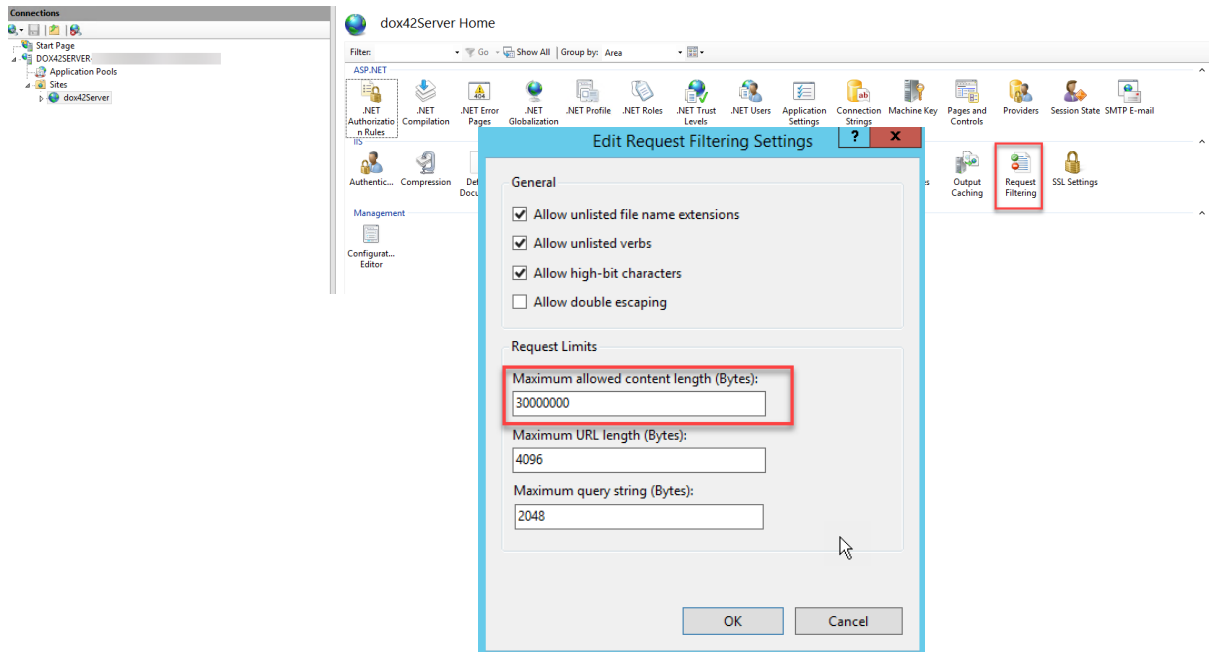
Um automatische Silbentrennung mit dem dox42 Word Add-In zu verwenden, stellen Sie ein entsprechendes File mit Namen HyphenationDictionary.config auf C:\Users\<<USER>>\AppData\Local\dox42V1

## Maximale Request-Größe und Timeout

Wenn Sie große Request an dox42 senden, große Dateien retournieren oder bei Massendruck kann es nötig sein die maximale Request-Größe oder auch den Timeout zu erhöhen. (Der Default Timeout in ASP.NET beträgt 90 Sekunden. Danach bricht der IIS den Call ab.) Diese kann ebenfalls in der web.config eingestellt werden:

```
<system.web>
  <httpRuntime maxRequestLength="1000000" executionTimeout="1000"/>
</system.web>
```

Wenn Sie in dox42 D365 FO Szenarien mit der https Methode POST (base 64) arbeiten und die Fehlermeldung „413 Request entity too large“ erhalten, können Sie den Parameter „Maximum allowed content length (bytes)“ im IIS ändern. Navigieren Sie dazu zum Icon „Request Filtering“, klicken Sie auf „Edit Feature Set“ und stellen Sie den Parameter von 30000000 auf 2147483647 um.



## Lizenz Aktivierung

Tragen Sie den Lizenzschlüssel in der Web.config unter „LicenseKey“ ein.

Beim ersten Aufruf kontaktiert der dox42 Server den dox42 License-Server um den Key zu aktivieren und schreibt eine Lizenzdatei (\*.lic) in das unter „LicenseFilePath“ angegebene Verzeichnis.

Sollte der Server nur über einen Proxy ins Internet zugreifen können, hilft vielleicht folgender Eintrag in der web.config.

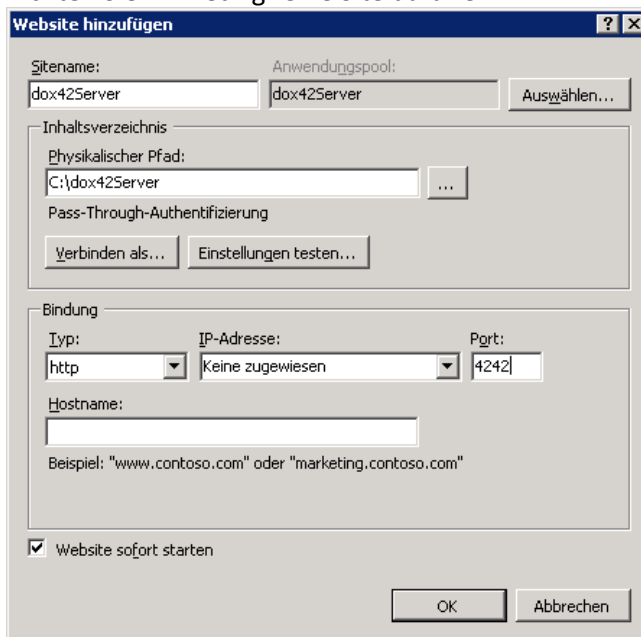


Sollten Sie keine Internetverbindung von Ihrem Server aus herstellen können, wenden Sie sich bitte an [support@dox42.com](mailto:support@dox42.com). Wir stellen Ihnen dann eine Lizenzdatei zur Verfügung.

# Installation

Im Folgenden wird die Installation auf einem Windows Server 2008 mit IIS 7.5 und installiertem ASP.NET 4.0 (oder höher) beschrieben. Sie können natürlich die jeweiligen neueren Versionen oder einen Windows Azure Server für die Installation verwenden.

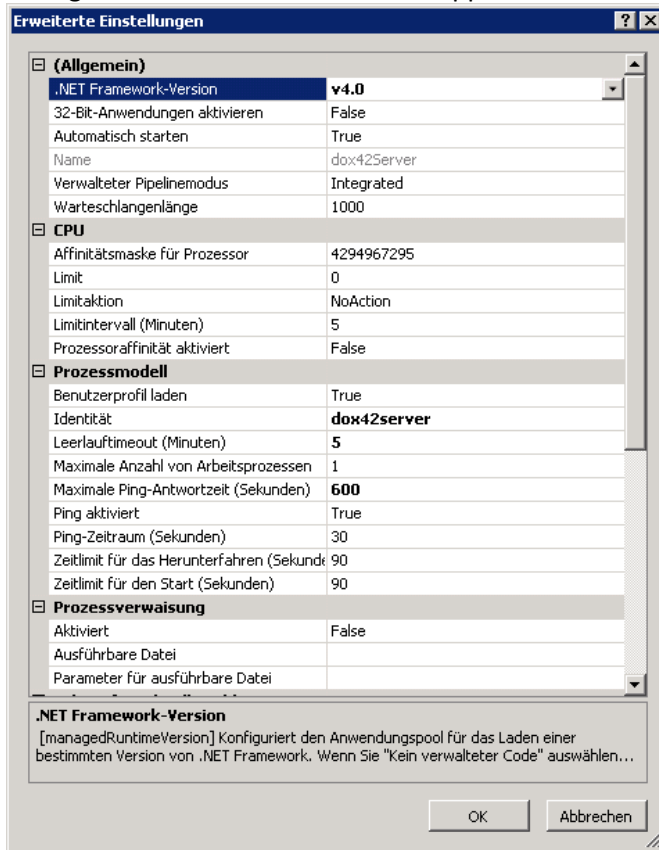
1. Kopieren Sie das Verzeichnis **dox42ServerPack\dox42ServerVXXXX\dox42Server** auf **C:\dox42Server** (Sie können auch ein anderes virtuelles Verzeichnis wählen.)
2. Richten Sie im inetmgr eine Site dafür ein:



Achten Sie dabei auf den Port! Sie können natürlich einen anderen Port vergeben. Sehr häufig ist aber der vorgelegte (Port 80) bereits durch andere Webanwendungen (z.B. Sharepoint) in Verwendung. Selbstverständlich unterstützt der dox42 Server https.

3. Kopieren Sie Ihre Aspose.Total.Lic File in das Bin Verzeichnis Ihres dox42 Servers (C:\dox42Server\Bin) – für Testversion nicht notwendig.
4. Installieren Sie das NET 4.5.2 Framework.
5. Aktivieren Sie ASP.NET mittels Server Manager (inklusive HTTP Activation).
6. Wenn Sie mit diesem dox42 Server auf SharePoint Online auf Office 365 zugreifen möchten, installieren Sie außerdem Microsoft SharePoint Server 2013 Client Components SDK. Download: <http://www.microsoft.com/en-us/download/details.aspx?id=35585>
7. Führen Sie iisreset aus.

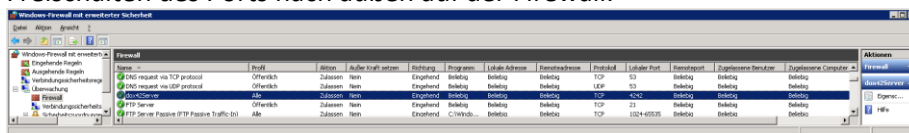
8. Konfigurieren Sie den dox42 Server-AppPool etwa wie folgt:



Hier wurde ein Service-User namens dox42Server als Identität verwendet. Statt dessen können Sie auch ApplicationPoolIdentity verwenden, oder mittels Windows-Authentication den aufrufenden User impersonieren.

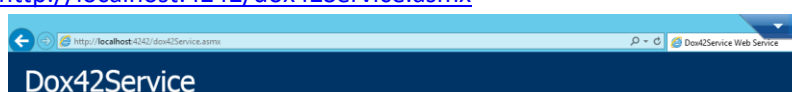
Um dox42 Beispielvorlagen mit MS Access DB Zugriff (benötigt 32 Bit) generieren zu können müssen Sie ev. 32-Bit-Anwendungen aktivieren.

9. Wechseln Sie wieder im inetmgr zur Site > Berechtigungen verwalten und geben dem dox42 Service User oder der ApplicationPoolIdentity (User „IIS AppPool\dox42Server“ mit Suchpfad auf lokalen Rechner) Lese- und Schreibrechte auf **C:\dox42Server** und auf das Temp-Verzeichnis.
10. Konfigurieren Sie die web.config (key, license-dir, Mail-Server) und NLog.config (log-dir).
11. Freischalten des Ports nach außen auf der Firewall.



12. Testen Sie den Aufruf des dox42 Webservices in Ihrem Browser:

"<http://localhost:4242/dox42Service.asmx>"



The following operations are supported. For a formal definition, please review the [Service Description](#).

- **GenerateDocument**  
Generates a Document based on the specified dox42 Word template. Version: 4.2.1.10
- **GenerateSlides**  
Generates Slides based on the specified dox42 PowerPoint template. Version: 4.2.1.10
- **GenerateSpreadSheet**  
Generates a spread sheet based on the specified dox42 data map. Version: 4.2.1.10

13. Führen Sie einen Test mit dem dox42 Server Designer durch. Installieren und öffnen Sie diesen. Sie finden die Applikation unter C:\dox42ServerPackVXXXX\ dox42ServerDesigner\_\*. Im Server Designer die HelloWorldTest\_ServerDesigner.Config Datei öffnen, welche Sie unter C:\dox42Server\templates\HelloWorldTest finden. Klicken Sie danach auf „Call Service“ um den Test abzuschließen.

Der dox42 Server Designer unterstützt Sie beim einfachen Aufbau Ihrer dox42 Webservice Aufrufe. Im Video Tutorial [#dox42Class of dox42 Server Designer](#) werden die Möglichkeiten vorgestellt.

14. Achten Sie auf adäquate Zugriffskontrolle auf Ihr dox42 Service um unberechtigten Zugriff zu verhindern.
15. Um AzureAD zu verwenden benötigen Sie ein HTTPS Zertifikat. Wenn Sie Hilfe in der Installation eines Zertifikats benötigen, sollte Ihnen dieses Video helfen:  
<https://youtu.be/HHU46l6G2oc>

## Upgrade

Um eine neue Version zu installieren gehen Sie bitte wie folgt vor:

1. Machen Sie ein Backup Ihres dox42 Server Verzeichnisses, so können Sie im Problemfall zurücksteigen.
2. Löschen Sie alle Dateien im Bin Verzeichnis AUSSER Aspose.Total.lic und aller dll-Dateien, die zu Ihren Custom Data Sources und Output Actions gehören. Custom Data Sources und Output Actions sind in der web.config eingetragen, z.B.: dox42SAP, dox42AX, dox42DynamicCRM, etc. Bitte sehen Sie in der jeweiligen Dokumentation nach um die Liste der benötigten Dateien zu ermitteln.
3. Kopieren Sie alle Dateien vom Bin Verzeichnis des dox42 Server Pack in das Bin Verzeichnis Ihres dox42 Servers.
4. Kopieren Sie alle Dateien AUSSER dox42LIVE.config vom dox42LIVE Verzeichnis des dox42 Server Pack in das dox42LIVE Verzeichnis Ihres dox42 Servers.
5. Überprüfen Sie ob Dox42Service.asmx und dox42RestService.ashx im dox42 Server Verzeichnis vorhanden sind.
6. Überprüfen Sie Ihre web.config auf neue Einstellungsmöglichkeiten.
7. Starten Sie den dox42 Server im IIS neu.
8. Test mit dox42 Server Designer
9. Das war's, viel Spaß 😊

## Support

Für Supportanfragen wenden Sie sich bitte an [support@dox42.com](mailto:support@dox42.com)

**Viel Erfolg mit Ihrem dox42 Server!**