



webPDF in SAP einbinden

Ein Basistutorial

webPDF ...the PDF powerhouse.



SoftVision Development GmbH
Kurfürstenstraße 15
36037 Fulda
Deutschland

Telefon: +49 661 25100-0
Telefax: +49 661 25100-25

E-Mail: info@softvision.de
WebSite: <http://www.softvision.de>

webPDF in SAP einbinden

© 2020 SoftVision Development GmbH, Fulda, Deutschland

Alle Rechte vorbehalten. Keine Teile dieses Werkes dürfen in irgendeiner Form oder mit irgendwelchen Mitteln - grafisch, elektronisch oder mechanisch, einschließlich Fotokopie, Aufzeichnung, Bandsicherung oder Informationsarchivierungs- und Bereitstellungssysteme - ohne die schriftliche Genehmigung des Herausgebers reproduziert werden.

Produkte, die in diesem Dokument bezeichnet werden, können entweder Warenzeichen und/oder eingetragene Warenzeichen der jeweiligen Eigentümer sein. Der Verleger und der Autor stellen keinen Anspruch auf diese Marken.

Obwohl bei der Erstellung dieses Dokuments alle Vorsichtsmaßnahmen getroffen wurden, übernehmen der Verleger und der Autor keine Verantwortung für Fehler oder Auslassungen oder für Schäden, die aus der Nutzung der in diesem Dokument enthaltenen Informationen bzw. durch die Nutzung von Programmen und Source-Code, die es begleiten kann. In keinem Fall sind der Verleger und der Autor für entgangenen Gewinn oder sonstige kommerzielle Schäden haftbar. Egal ob diese durch das Dokument, direkt oder indirekt, verursacht oder angeblich verursacht wurden.

Gedruckt: Juni 2020 in Fulda, Deutschland

Herausgeber

*SoftVision Development GmbH
Kurfürstenstraße 15
36037 Fulda*

Autoren

*Alexey Babitskiy
Bernd Engelhardt*

Inhaltsverzeichnis

Teil I Einleitung	1
1 Was ist webPDF?	1
Teil II Voraussetzungen	2
1 Installation webPDF	2
2 SAP Entwickler-System (als Demo-System)	4
Teil III SOAP-Beispiel	5
1 Einrichtung	5
Einbindung als Service-Consumer	5
Anlegen eines Packet im Object Navigator	6
Erstellung eines Enterprise Services	8
Konfiguration des Services über den SOA Manager	14
Aktivierung und einfacher Test des Service	20
2 Anbindung	22
Teil IV REST-Beispiel	29
1 Anbindung	29
Teil V Anhang	36
1 SOAP ABAP Code	36
2 REST ABAP Code	39
3 Links	48

1 Einleitung

In dieser Anleitung soll anhand einfacher Beispiele, die Einbindung der webPDF SOAP- und REST-Webservices in SAP demonstriert werden.

Die hier gezeigten Beispiele sollen den Weg zeigen, wie man die Webservice-Funktionen von webPDF in eine SAP-Umgebung integrieren kann. Die Beispiele stellen dabei keine vollwertige Lösung dar, die direkt in bestimmten SAP-Anwendungen genutzt werden können. Sie dienen lediglich als Grundlage für eigene Entwicklung und sollen das grundsätzliche Vorgehen beschreiben.

Die Beispiele wurden mit einem [SAP Demo-System](#) erstellt und müssen daher ggf. bei der Verwendung in entsprechenden Live-Systemen angepasst werden.

Bei Fragen zu webPDF oder diese Dokument, stehen wir Ihnen gerne unter support@softvision.de zur Verfügung.

1.1 Was ist webPDF?

webPDF ist eine serverseitige Lösung für Windows und Linux, die Funktionen für die Konvertierung von Dateiformaten nach PDF und für die Verarbeitung von bestehenden PDF-Dokumenten bereitstellt. Die Lösung basiert auf einem Browser-basierten Portal und einer SOAP- und REST Webservice-Schnittstelle.

Allgemeine Informationen zu webPDF finden Sie unter <https://www.webpdf.de>

Technische Informationen zur webPDF finden Sie unter
<https://portal.webpdf.de/webPDF/help/doc/de/welcome.htm>

2

Voraussetzungen

In diesem Kapitel wird Ihnen erläutert, welche Bedingungen Sie auf Ihrem System benötigen, um die später aufgeführten Beispiele umzusetzen.

2.1 Installation webPDF

Für die nachfolgenden SOAP- und REST-Beispiele wird eine vorhandene webPDF-Installation benötigt. Sie finden den Download für eine Testinstallation unter <https://download.softvision.de/?product=webpdf>.

Für eine Installation unter Linux finden sie unter <https://packages.softvision.de/> die entsprechenden Informationen.

Sie können webPDF auch als virtuelle Maschine (https://download.softvision.de/?product=webpdf_vm) oder als Docker Container (<https://github.com/softvision-dev/webpdf-docker>) ausführen.

Wichtig ist, das webPDF ausgeführt wird und von ihrem SAP Server aus erreichbar ist!

Dass webPDF läuft und erreichbar ist, können Sie über zwei Wege testen:

1. Aufruf des Portals
2. Aufruf eines Endpunkts

Bitte beachten Sie, das die dies vom Rechner aus gemacht werden sollte, von dem aus auch die SAP-Installation auf den webPDF-Server zugreift.

1. Aufruf des Portals

Rufen Sie mittels eines Browsers die Startseite des webPDF-Portals auf und führen Sie beispielhaft eine Konvertierung durch.

```
http://<Adresse>:8080/webPDF
```

Ersetzen Sie in o.g. URL den Wert "<Adresse>" durch die IP-Adresse des Rechners, auf dem Sie webPDF installiert haben.

Wenn Sie unter Linux sind, so können Sie per Command Line mit CURL den Zugriff auf den Server testen. Eine einfache Form, in der <Adresse> durch die IP-Adresse des Servers zu ersetzen ist, wäre

```
curl -Is http://<Adresse>:8080/webPDF/ | head -n 1
```

wo der Server mit einem HTTP Status-Code "200 OK" im Erfolgsfall antwortet oder einem entsprechenden Fehlercode, wenn der Zugriff nicht möglich ist.

2. Aufruf eines Endpunkts

Rufen Sie mittels eines Browsers die WSDL-Informationen des SOAP-Webservice "Converter" ab.

```
http://<Adresse>:8080/webPDF/soap/converter?wsdl
```

Bei einem Aufruf sollte Ihnen die WSDL-Struktur für den Converter angezeigt werden.

```

<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
              xmlns:wsp="http://www.w3.org/ns/ws-policy"
              xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
              xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
              xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
              xmlns:tns="http://schema.webpdf.de/1.0/soap/converter"
              xmlns:xsd="http://www.w3.org/2001/XMLSchema"
              xmlns="http://schemas.xmlsoap.org/wsdl/"
              targetNamespace="http://schema.webpdf.de/1.0/soap/converter"
              name="ConverterService">
  <wsp:Policy wsu:Id="ConverterPortBinding_MTOM_Policy">
    <ns1:OptimizedMimeSerialization
      xmlns:ns1="http://schemas.xmlsoap.org/ws/2004/09/policy/optimizedmimeserialization"
      wsp:Optional="true"/>
  </wsp:Policy>
  <types>
    <xsd:schema>
      <xsd:import
        namespace="http://schema.webpdf.de/1.0/soap/converter"
        schemaLocation="http://localhost:8080/webPDF/soa
p/converter?xsd=1"/>
    </xsd:schema>
    <xsd:schema>
      <xsd:import
        namespace="http://schema.webpdf.de/1.0/operation"
        schemaLocation="http://localhost:8080/webPDF/soa
p/converter?xsd=2"/>
    </xsd:schema>
  </types>
  ...
  ...
  <service name="ConverterService">
    <port name="ConverterPort" binding="tns:ConverterPortBinding">
      <soap12:address
        location="http://localhost:8080/webPDF/soap/converter"/>
    </port>
  </service>
</definitions>

```

Wenn Sie unter Linux sind, so können Sie per Command Line mit CURL den Zugriff auf den Server testen. Eine einfach Form, in der `<Adresse>` durch die IP-Adresse des Servers zu ersetzen ist, wäre

```
curl -Is http://<Adresse>:8080/webPDF/soap/converter?wsdl | head -n 1
```

wo der Server mit einem HTTP Status-Code "200 OK" im Erfolgsfall antwortet oder einem entsprechenden Fehlercode, wenn der Zugriff nicht möglich ist.

Sie können die Ausgaben der Prüfung auch gerne mit unserem Live-Server im Web vergleichen, wenn sie sicher sein wollen, das ihr lokaler Abruf die korrekten Daten liefert:

webPDF-Portal: <https://portal.webpdf.de/webPDF/>

webPDF SOAP 1.1 WSDL: <https://portal.webpdf.de/webPDF/soap/converter?wsdl>

Sollte webPDF bei Ihnen noch nicht laufen, finden Sie alle Informationen zur Einrichtung in der [Dokumentation](#).

2.2 SAP Entwickler-System (als Demo-System)

Für die nachfolgenden Beispiele können Sie entweder Ihr vorhandenes SAP-System oder die Entwickler-Version des "ABAP application server on ASE 16.0" verwenden.

Hinweis: Dieses Dokument verwendet für die nachfolgenden Beispiele ein SAP Entwicklersystem als Demo-System, das gemäß dem Blog-Artikel <https://blogs.sap.com/2019/07/01/as-abap-752-sp04-developer-edition-to-download/> eingerichtet wurde. Der Artikel beschreibt ausführlich die die Einrichtung einer solchen Demo-Umgebung.

Diese Installation stellt die folgenden Komponenten zur Verfügung:

- SAP AS ABAP 7.52 SP04
- SAP GUI for the Java 7.5 and SAP GUI for Windows 7.50
- SAP Sybase ASE 16.0 SP02

Mit diesen Komponenten können Sie die nachfolgenden Beispiele nachvollziehen.

Eine entsprechende Anleitung zur Installation eines SAP-Systems steht für VirtualBox und VMware zur Verfügung:

- [Installing SAP AS ABAP 752 SP04 on Linux – Virtual Box](#)
- [Installing SAP AS ABAP 7.52 SP04 on Linux – VMWare](#)

Für diese Dokumentation wurde ein Demo-System nach der Anleitung "[Installing SAP AS ABAP 752 SP04 on Linux – Virtual Box](#)" erstellt.

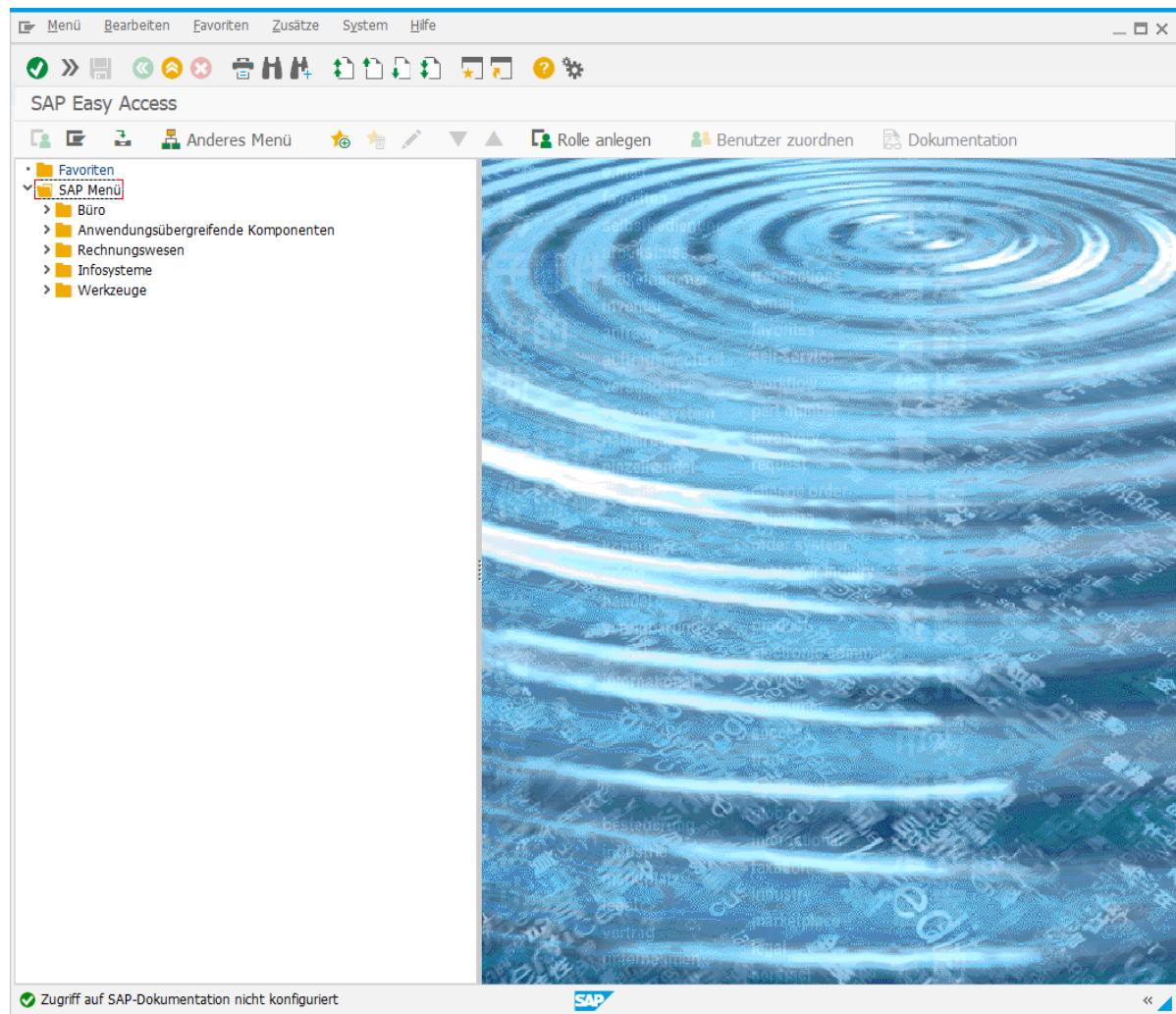
3 SOAP-Beispiel

Das nachfolgende Beispiel zeigt, wie Sie die SOAP 1.1 Webservices von webPDF aus SAP heraus verwenden können.

Das Beispiel zeigt dies an Hand des SOAP 1.1 Webservice "Converter".

Dieses Beispiel kann leicht auf die anderen Webservice Endpunkte übertragen werden.

Um dieses Beispiel zu verwenden, starten Sie bitte den SAP Business Client (des [Demo-Systems](#)) und melden sich am SAP-System an.

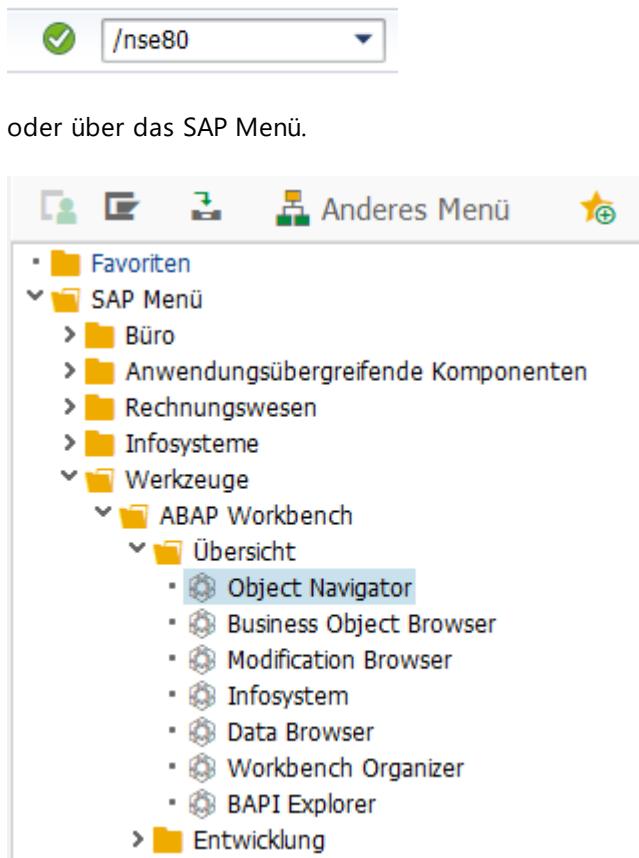


3.1 Einrichtung

3.1.1 Einbindung als Service-Consumer

Starten Sie den SAP Client und öffnen Sie danach den "Object Navigator".

Diesen können Sie über das Befehlsfeld mit dem Transaktionscode "se80" ("/nse80") machen

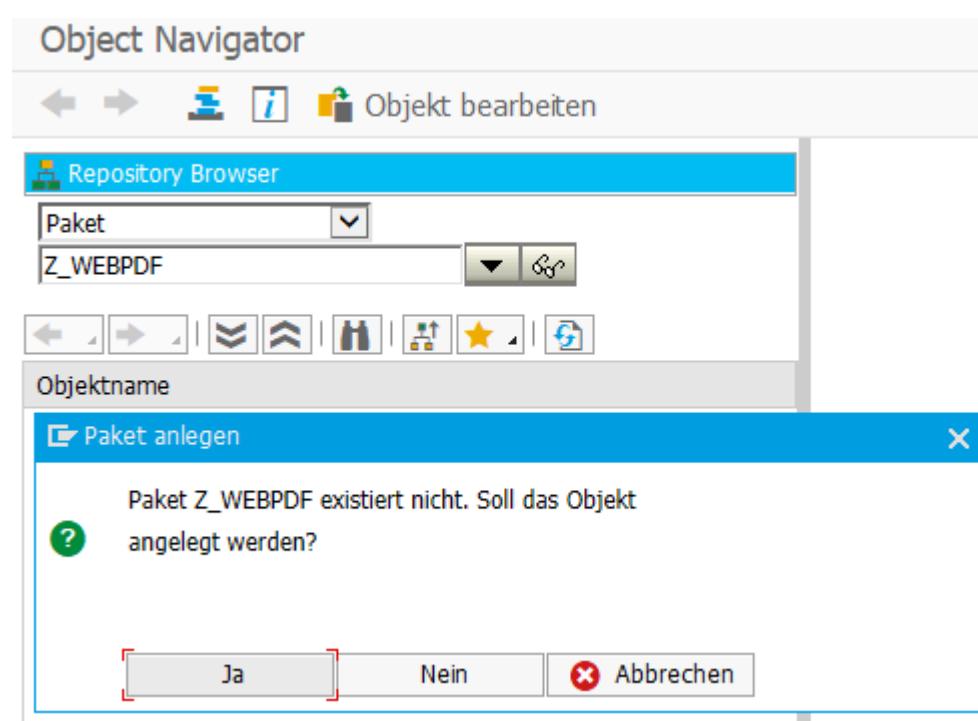


3.1.2 Anlegen eines Packet im Object Navigator

Im nächsten Schritt legen Sie ein Package für die notwendigen Komponenten an. Wählen Sie dazu "Paket" in der oberen Auswahl und geben darunter den Paketnamen ein, der verwendet werden soll.

Wir verwenden in diesem Beispiel den Namen "Z_WEBPDF".

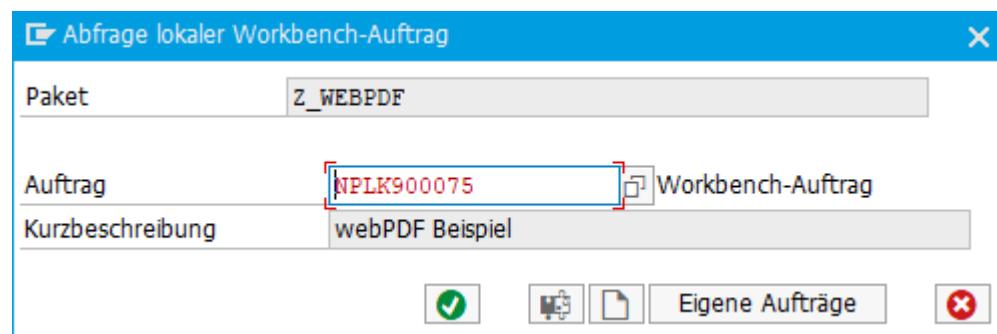
Nach der Bestätigung der Eingabe mit "Enter" kommt noch eine Sicherheitsfrage, die Sie mit "Ja" bestätigen, um das neue Paket zu erstellen.



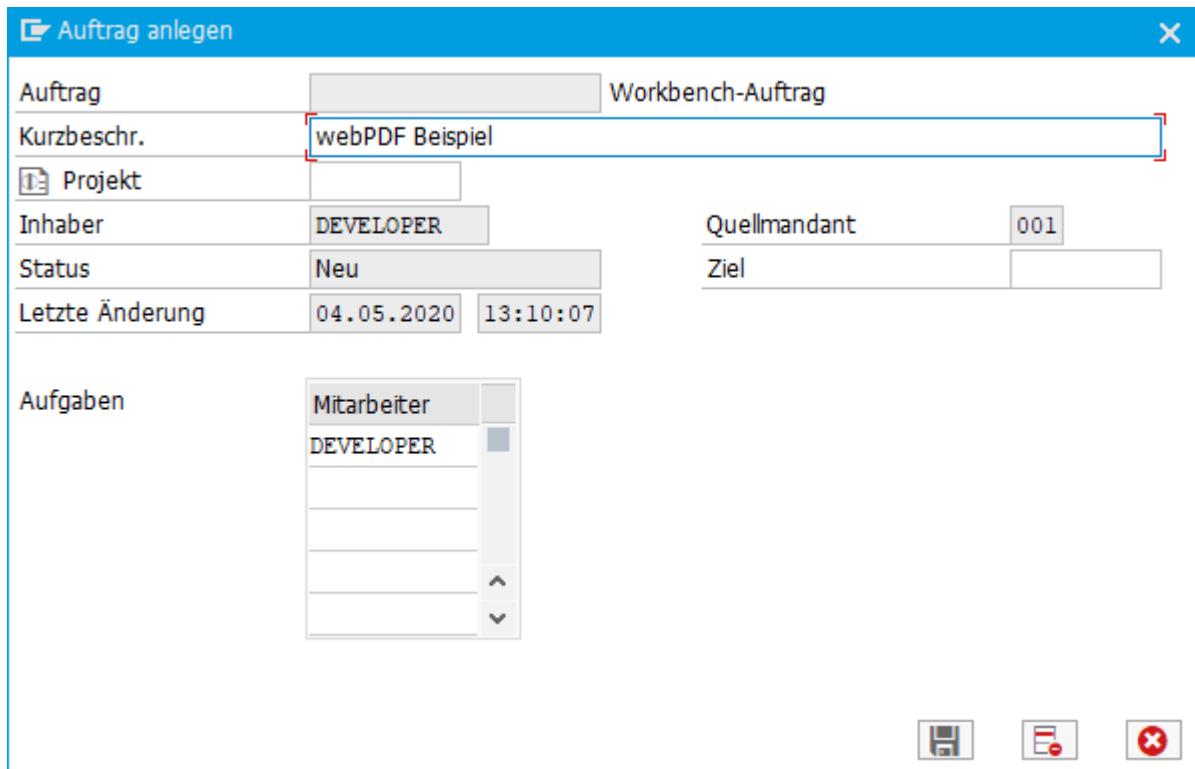
Im darauf folgendem Dialog können Sie noch eine Beschreibung für das Package angeben.



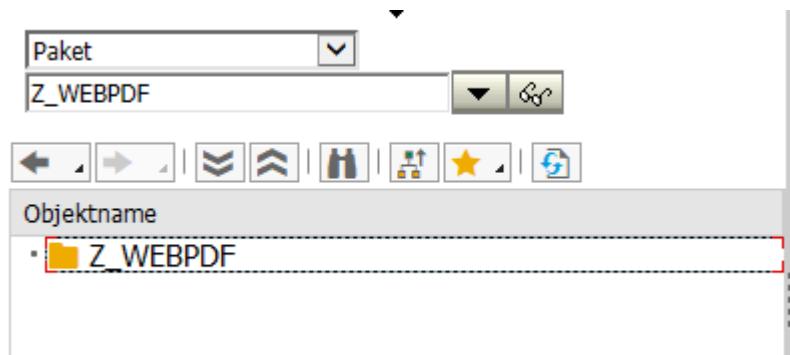
Im nächsten Dialog muss der "Workbench-Auftrag" ausgewählt werden.



Wenn kein Auftrag vorhanden ist, dann muss zuvor ein neuer Auftrag erstellt werden (mit dem Button neben Feld "A").



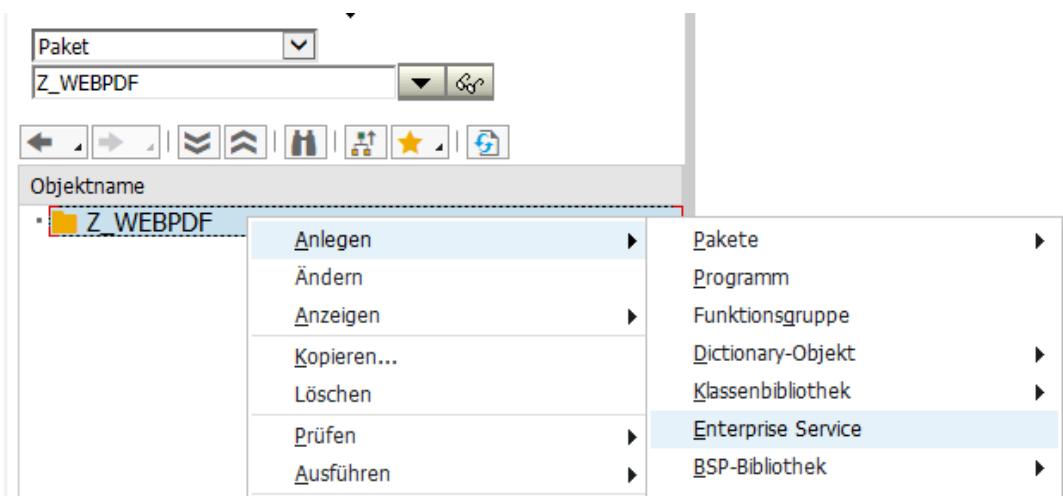
Wenn Sie einen entsprechenden Auftrag angelegt (bzw. ausgewählt) haben, wir das Paket angelegt.



3.1.3 Erstellung eines Enterprise Services

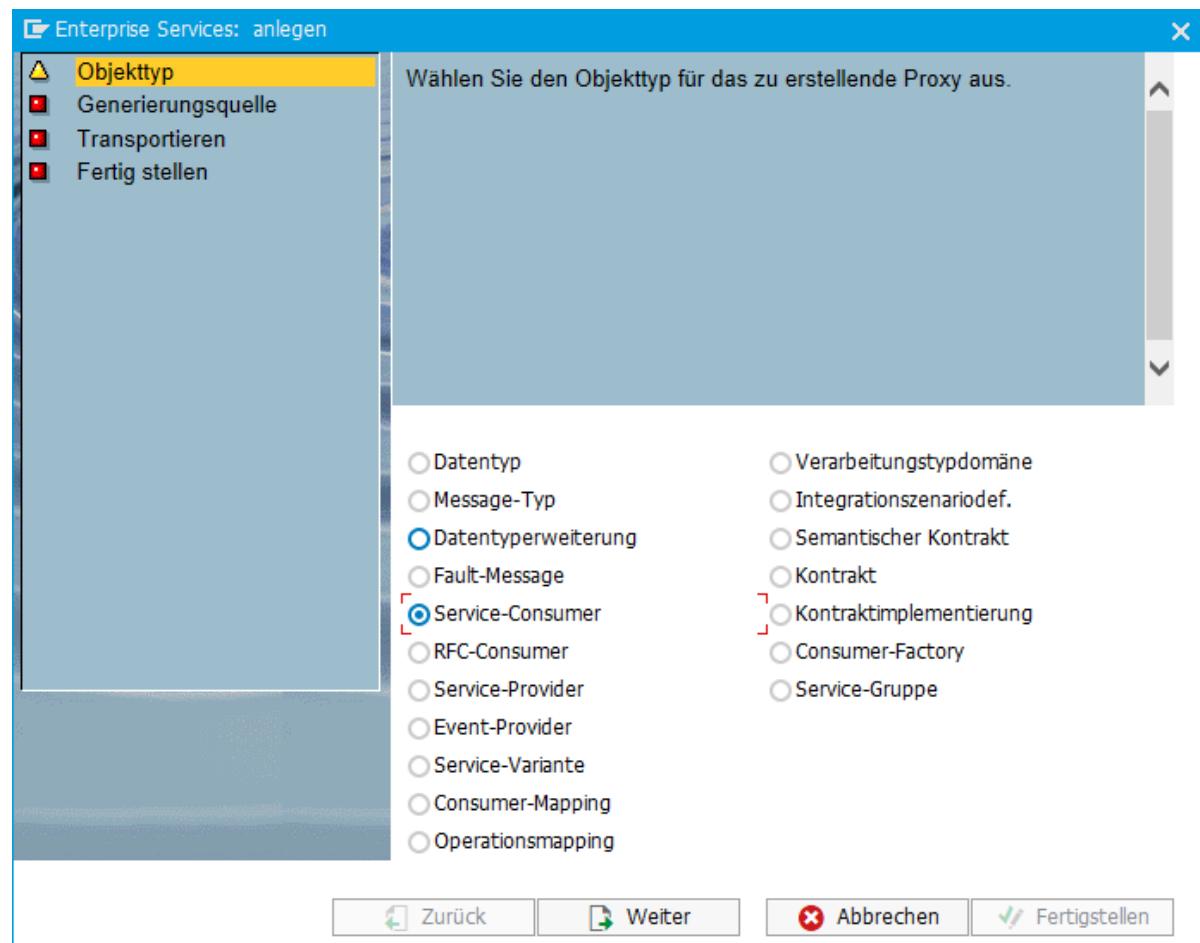
In dem angelegtem Paket erstellen Sie nun einen "Enterprise Service".

Dies wird mit einem Rechtsklick auf "Z_WEBPDF" und dem erscheinenden Kontextmenü unter "Anlegen->Enterprise Service" gemacht.

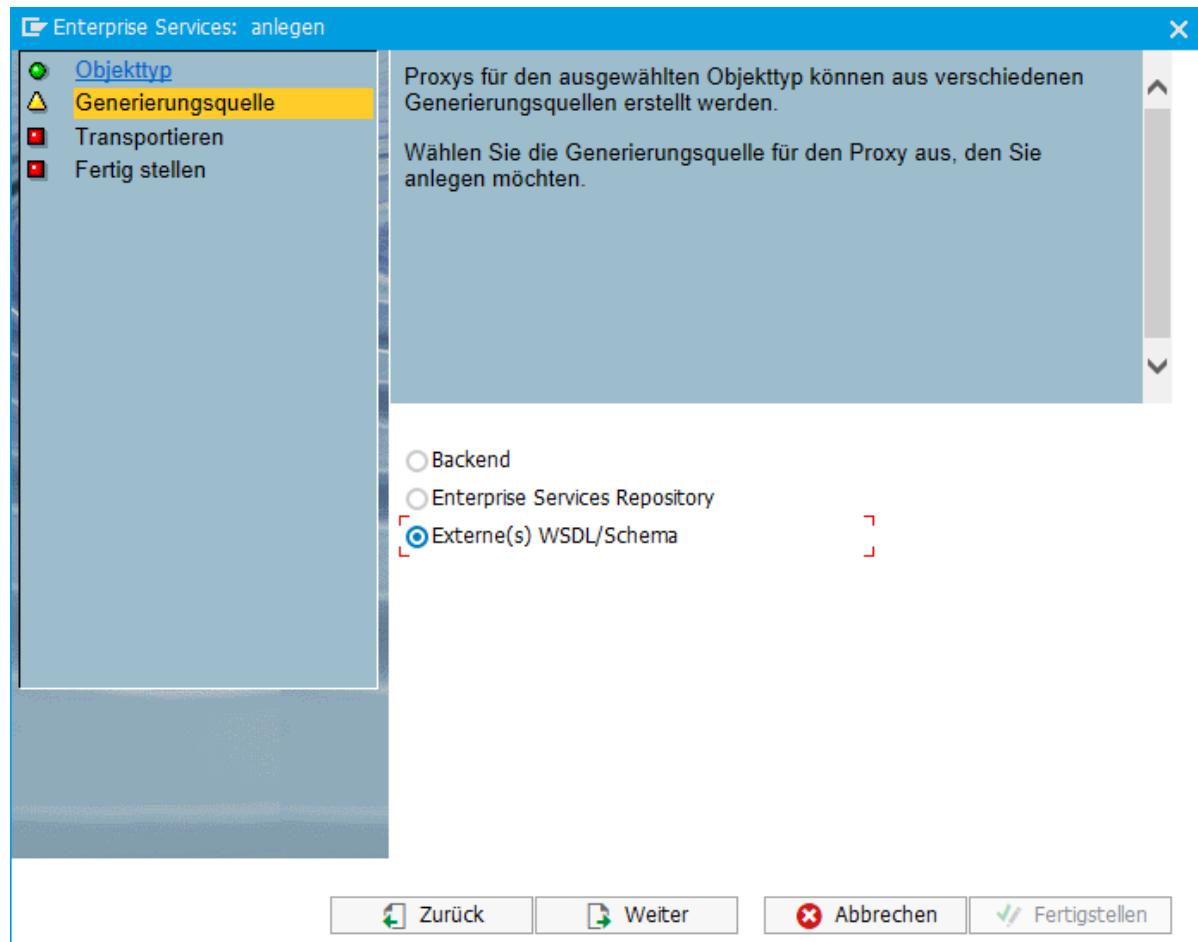


Es erscheint Dialog, in dem mehrere Punkte eingestellt werden müssen. Wählen Sie wie nachfolgende beschrieben die entsprechenden Optionen und klicken Sie dann jeweils auf "Weiter".

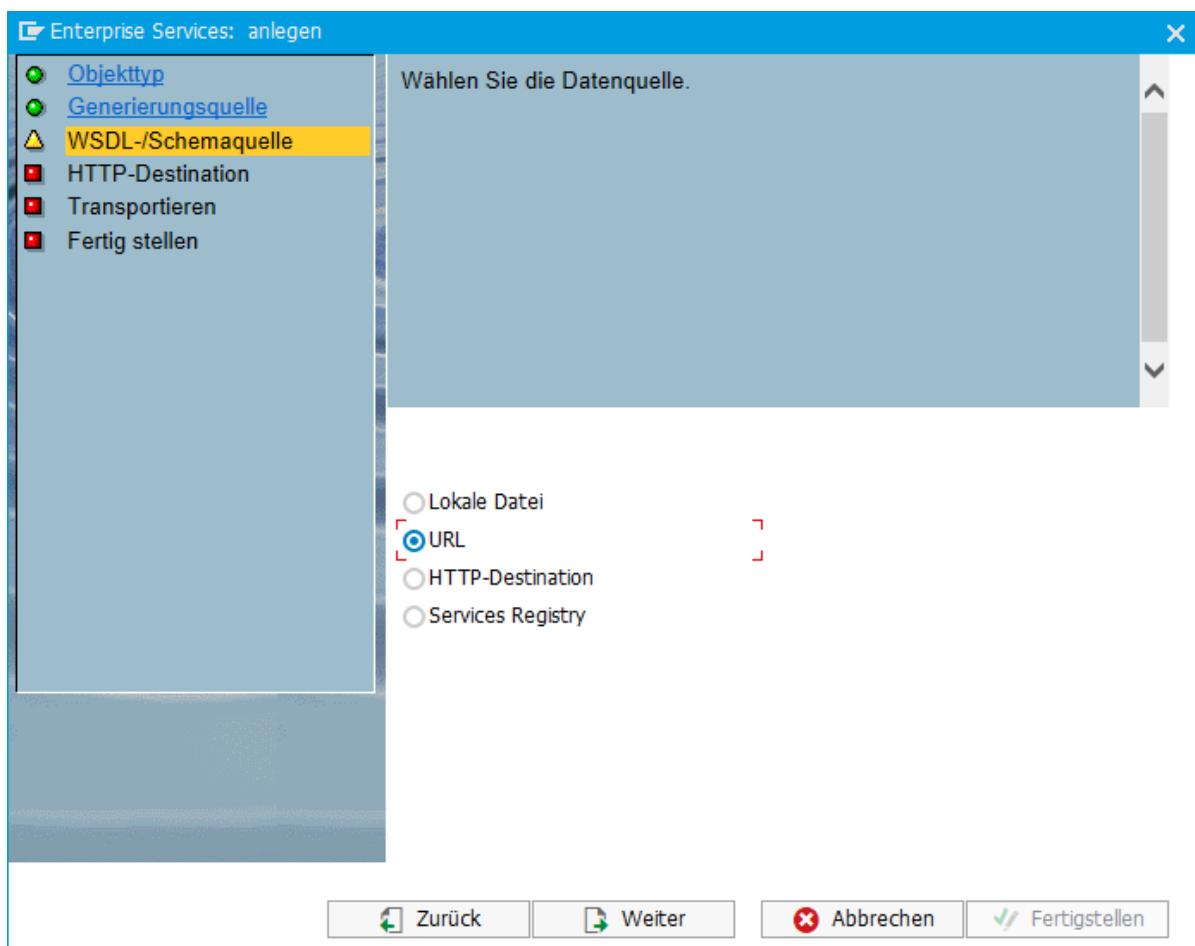
Als "Objekttyp" wählen Sie nun "Service-Consumer" aus.



Als "Generierungsquelle" wählen Sie "Externe(s) WSDL/Schema aus".



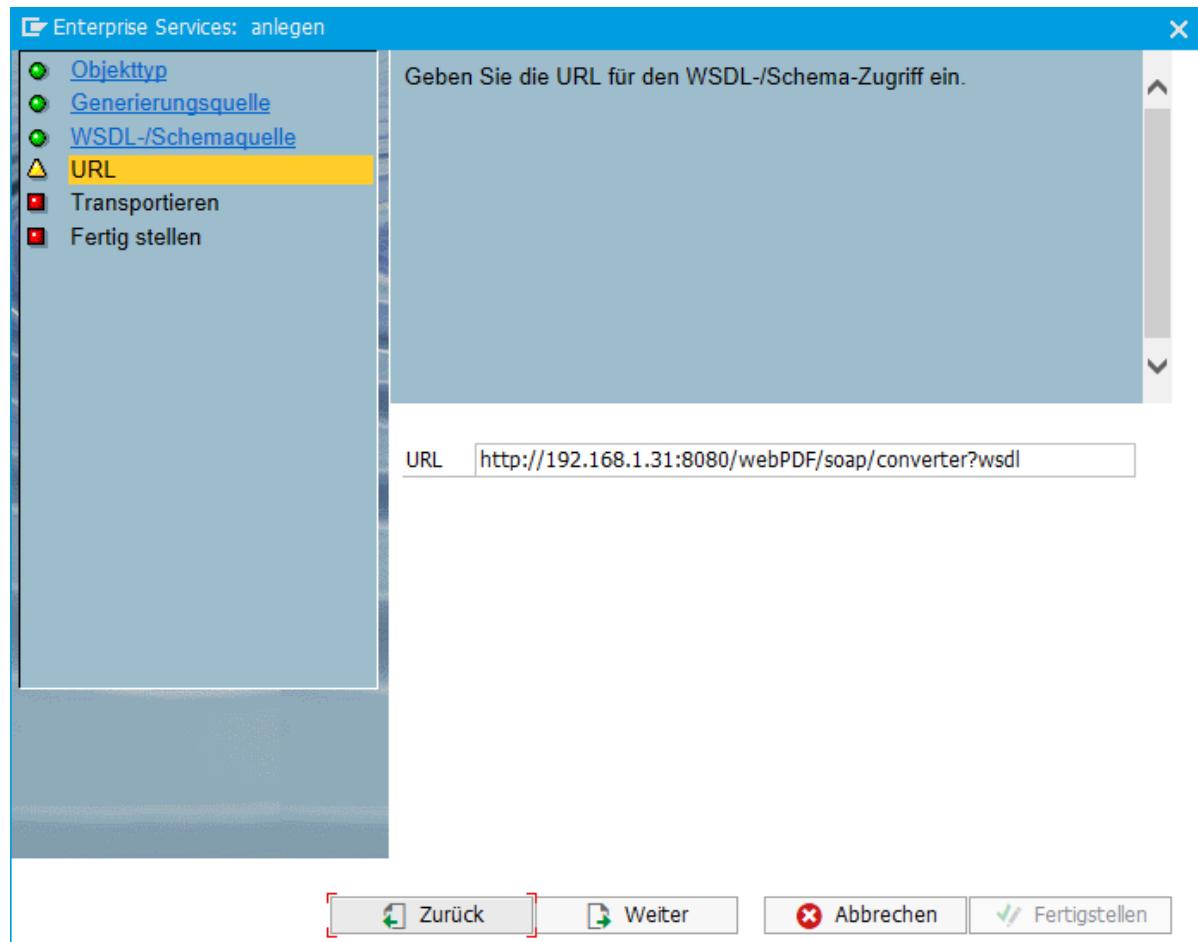
Als "Datenquelle" wählen Sie "URL" aus.



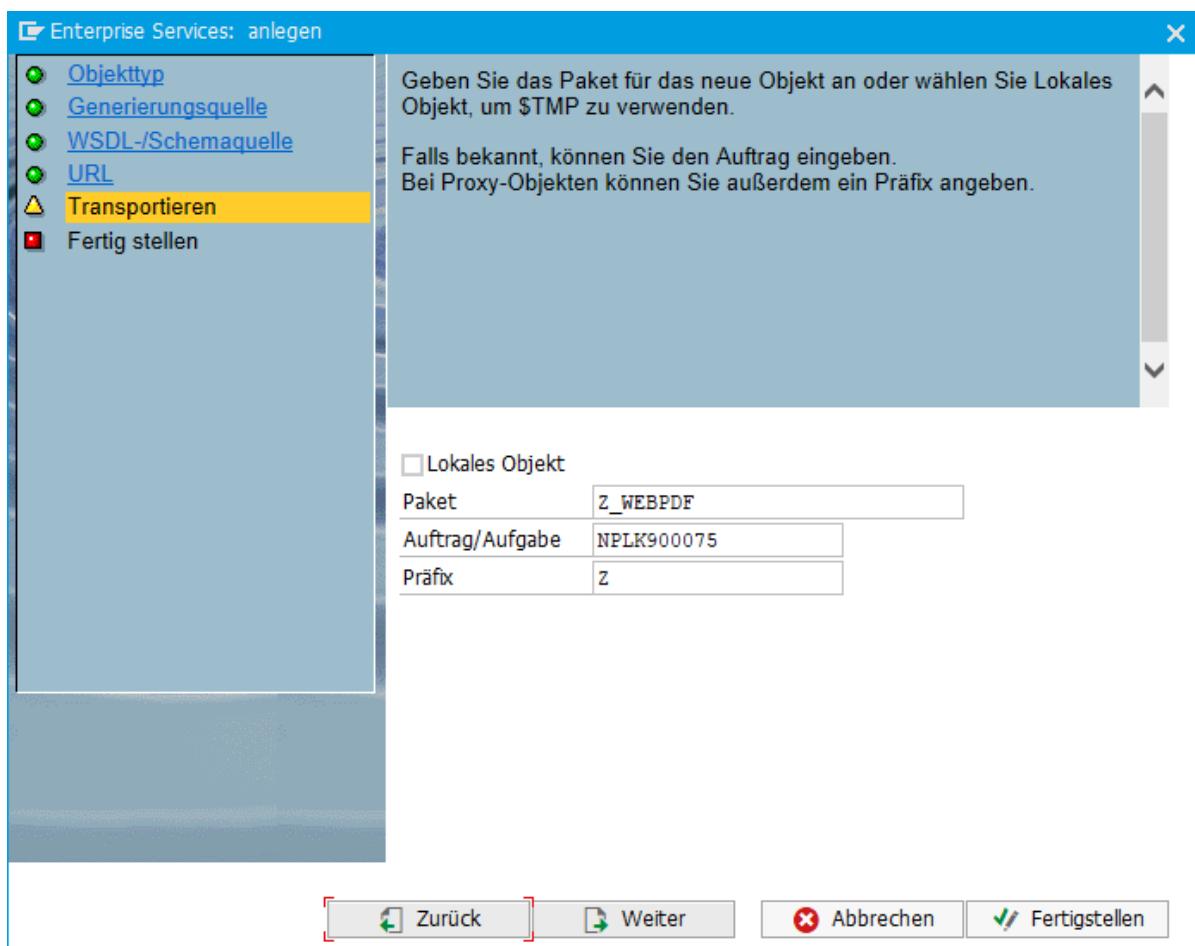
Als "URL" wählen die WSDL-Adresse des webPDF Converter Services aus (mit dieser haben Sie zuvor geprüft, ob webPDF erreichbar ist).

`http://<Adresse>:8080/webPDF/soap/converter?wsdl`

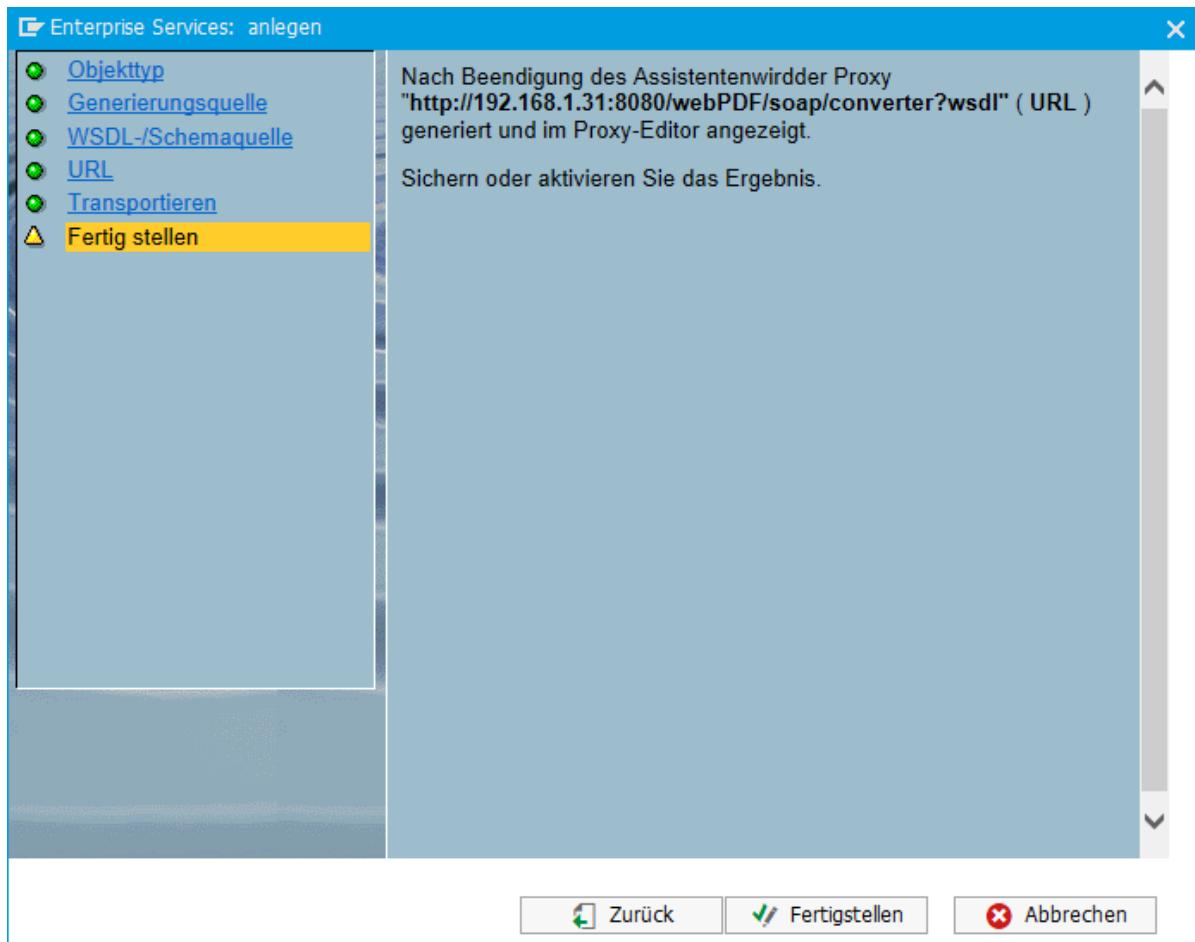
Für "<Adresse>" setzen Sie die IP-Adresse des Servers ein.



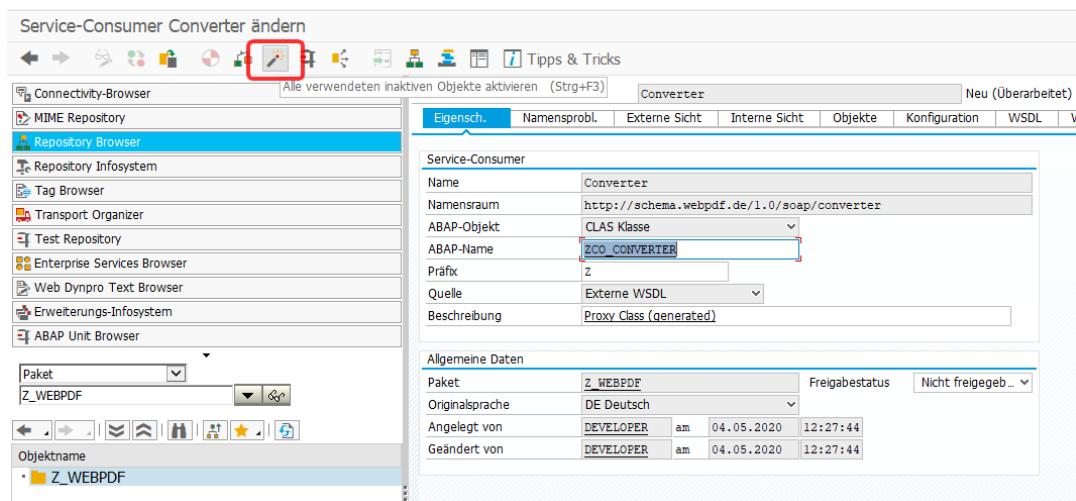
Wählen Sie bei "Transportieren" ihr Paket "Z_WEBPDF" und den passenden Auftrag aus.



Nun können Sie den Vorgang durch "Fertigstellen" oder "Weiter->Fertigstellen" abschließen.

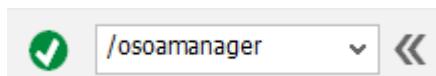


Am Ende aktivieren Sie das erstellte "Service Consumer"-Objekt noch.



3.1.4 Konfiguration des Services über den SOA Manager

Öffnen Sie den SOA Manager über das Befehlsfeld dem Transaktioncode "/osoamanager"



Wählen Sie aus dem im Browser geöffneten SOA Manager unter "Service-Administration" den Punkt "Web-Service-Konfiguration" aus.

Suchen Sie nach dem von Ihnen zuvor erstellten Consumer-Proxy Objekt, durch Eingabe des Suchbegriffs "CONVERTER" und einem Klick auf den Schalter "Suche".

Interner Name	Typ	Name	Namensraum
ZCO_CONVERTER	Consumer-Proxy	Converter	http://schema.webpdf.de/1.0/soap/c...

Klicken Sie auf den Eintrag "ZCO_CONVERTER" und wählen Sie danach in dem erscheinenden Dialog unter "Konfiguration", "Anlegen->WSDL-basierte Konfiguration" aus.

Web-Service-Konfiguration (NPL;001)

Details von Consumer-Proxy: ZCO_CONVERTER

Übersicht Konfigurationen Details

Logische Ports definieren

Anlegen	Log. Standardport festl.	Aktivieren	Deaktivieren	L	Zustand
WSDL-basierte Konfiguration					
Manuelle Konfiguration					
Process-Integration-Laufzeit					
Konfiguration für lokalen Shortcut					
Services-Registry-basierte Konfiguration					
Vorlagenbasierte Konfiguration					
WSDL-basierte Konfiguration mit Vorlage					

Fügen Sie in den folgenden Dialog einen logischen Port-Namen ein (z.B. "ZWEBPDF_LOGICAL_PORT") und aktivieren Sie die Checkbox "Logischer Port ist Standard".

1 Name des logischen Ports 2 WSDL-Informationen 3 Binding-Auswahl 4 Consumer-Sicherheit 5 HTTP-Einstellungen 6 SOAP-Protokoll

Zurück Weiter Fertig Abbrechen

Allgemeine Konfigurationseinstellungen

* Name des logischen Ports: Logischer Port ist Standard:

Beschreibung:

Klicken Sie auf "Weiter" und wählen Sie im nächsten Schritt bei den WSDL-Zugriffseinstellungen "Über HTTP-Zugriff" aus. Danach geben Sie die bekannte URL für das WSDL des Converter-Services an.

http://<Adresse>:8080/webPDF/soap/converter?wsdl

Für "<Adresse>" setzen Sie die IP-Adresse des Servers ein.

1 2 3 4

Name des logischen Ports WSDL-Informationen Binding-Auswahl Consumer-Sicherheit

Zurück Weiter Fertig Abbrechen

WSDL-Zugriffseinstellungen

WSDL-Basis: Über HTTP-Zugriff
 Über Datei
 WSDL aus Hochladen

WSDL-Lokation

URL für WSDL-Zugriff:
WSDL-Zugriffsbenutzer:
WSDL-Zugriffsbenutzerkennwort:
Proxy-Einstellungen ein-/ausblenden

Die folgenden Schritte "Binding-Auswahl" und "Consumer-Sicherheit" können Sie jeweils mit "Weiter" überspringen.

1 2 3 4 5 6

Name des logischen Ports WSDL-Informationen Binding-Auswahl Consumer-Sicherheit HTTP-Einstellungen SOAP-Protokoll

Zurück Weiter Fertig Abbrechen

Binding

Ausgewähltes Binding: ConverterPortBinding

Übersicht über mögliche vom WSDL-Dokument genannte Kombinationen

Aktionen	Index	Reihenfolge	Name des Inhabers	Eigenschaftsname	Eigenschaftswert
	34	1	ConverterPortBinding		

1 2 3 4

Name des logischen Ports WSDL-Informationen Binding-Auswahl Consumer-Sicherheit

Zurück Weiter Fertig Abbrechen

Konfiguration von Consumer-Einstellungen zusätzlich zu WSDL-Dokument-Info

Benutzer-ID und Kennwort

Benutzername:

Kennwort:

Eigenschaften von WSDL-Dokument

Authentifizierung

Authentifizierungsmethode:

Transportsicherheit

Signatur erwartet:

Verschlüsselung erwartet:

Message signieren:

Verschlüsselung hinzufügen:

Im Schritt "HTTP-Einstellungen" können Sie dann Ihre Angaben überprüfen, ohne weiter Eingaben zu machen.

1 Name des logischen Ports 2 WSDL-Informationen 3 Binding-Auswahl 4 Consumer-Sicherheit 5 HTTP-Einstellungen 6 SOAP-Protokoll 7 Identifiable Business Context

Zurück **Weiter** **Fertig** **Abbrechen**

URL-Zugriffspfad

Vollständige URL URL-Komponenten

* URL:

Anmeldesprache:

Proxy:

Name des Proxy-Rechners:

Portnummer des Proxy-Rechners:

Benutzername für Proxy-Zugriff:

Kennwort des Proxy-Benutzers:

Transport-Binding

Lokalen Aufruf durchführen:

* Transport-Binding-Typ:

Maxim. Wartezeit WS-Consumer:

Optimierter XML-Transfer:

HTTP-Message komprimieren:

Response komprimieren:

In Schritt "SOAP-Protokoll" müssen Sie lediglich darauf achten, dass "Message-ID-Protokoll" auf "ID-Übertragung unterdrücken" steht.

1 Name des logischen Ports 2 WSDL-Informationen 3 Binding-Auswahl 4 Consumer-Sicherheit 5 HTTP-Einstellungen 6 SOAP-Protokoll 7 Identifiable Business Context

Zurück **Weiter** **Fertig** **Abbrechen**

Message-ID (synchron)

Message-ID-Protokoll:

Statistik der Service-Aufrufe

Umfang des Datentransfers:

Transferprotokoll:

Handhabung Message-Attachments

Attachmentes verarbeiten:

Klicken Sie auf "Fertig" um die Konfiguration abzuschließen.

1 Name des logischen Ports 2 WSDL-Informationen 3 Binding-Auswahl 4 Consumer-Sicherheit 5 HTTP-Einstellungen 6 SOAP-Protokoll 7 Identifiable Business Context

Zurück **Weiter** **Fertig** **Abbrechen**

Identifiable Business Context

Sender-IBC-Identifikation:

Empfänger-IBC-Identifikation:

Senden der IBC-Identifikation unterdrücken:

Stellen Sie in der darauf folgenden Ansicht sicher, dass der Service konfiguriert und aktiviert ist.

Aktionen	Logischer Port	Zustand	Logischer Port ist Standard	Beschreibung	Erstellungstyp
	ZWEBPDF_LOGICAL_PORT	Aktiv	true	Port für webPDF	Manuell angelegt

Nun können Sie Ihren Browser schließen und zurück in den SAP Client wechseln.

3.1.5 Aktivierung und einfacher Test des Service

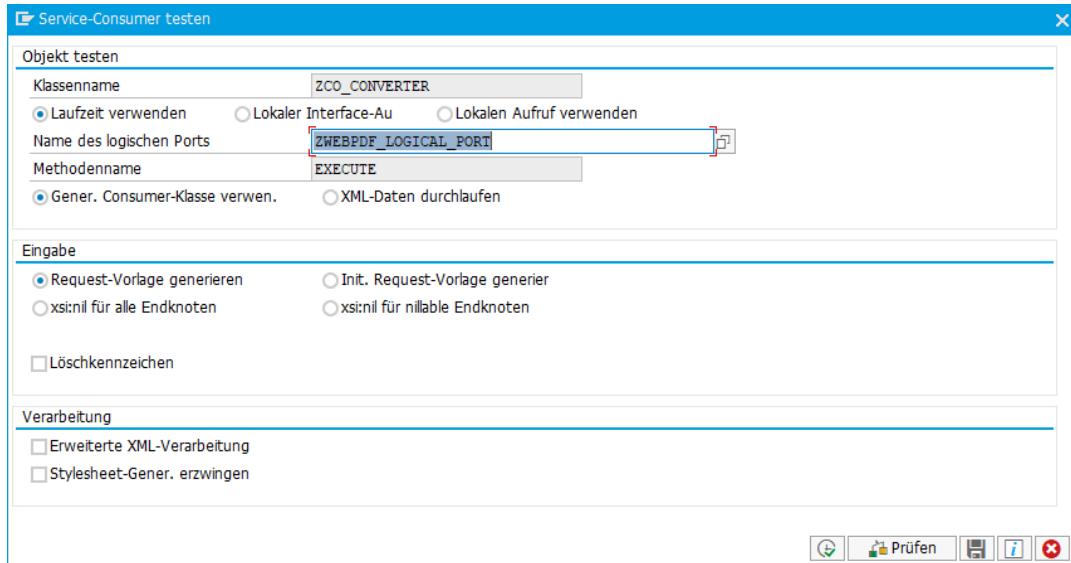
Wechseln Sie zurück in den "Object Navigator". Darin wählen Sie den angelegten Service-Consumer und öffnen ihn per Doppelklick.

In der erscheinenden Ansicht können Sie mit "F8" (oder Icon "Kompression") den Webservice testen.

Eigensch.	Externe Sicht	Interne Sicht	Objekte	Konfiguration	WSDL	Klassifizierung
Name: Converter Namensraum: http://schema.webpdf.de/1.0/soap/converter ABAP-Objekt: CLAS Klasse ABAP-Name: ZCO_CONVERTER Präfix: Z Quelle: Externe WSDL Beschreibung: Proxy Class (generated)						

Paket	Z_WEBPDF	Freigabestatus	Nicht freigegeb...
Originalsprache: DE Deutsch Angelegt von: DEVELOPER am 04.05.2020 12:27:44 Geändert von: DEVELOPER am 04.05.2020 12:31:28			

Im darauf folgenden Dialog wählen Sie den im SOA Manager angelegte Port unter "Name des logischen Ports" aus.



Wenn Sie nun "Ausführen" klicken wird Ihnen die XML des Requests angezeigt.

Service-Consumer testen: Anzeigen Request

Request

```
<?xml version="1.0"?>
- <n0:execute xmlns:n0="http://schema.webpdf.de/1.0/soap/converter">
  - <n1:operation xmlns:n1="http://schema.webpdf.de/1.0/operation">
    <n1:billing applicationName="String 3" userName="String 2" customerCode="String 1"/>
    <n1:password open="String 5" permission="String 4"/>
    - <n1:converter maxRecursion="11" fileExtension="String 10" accessPassword="String 9" embedFonts="true" dpi="8" reduceResolution="true" jpegQuality="7" compression="true" pages="String 6">
      <n1:page metrics="String 18" height="17" width="16" bottom="15" top="14" right="13" left="12"/>
      - <n1:pdfa>
        - <n1:convert autoTagging="true" successReport="String 22" errorReport="String 21" imageQuality="20" level="String 19">
          <n1:zugferd>
            <n1:xmlFile url="http://sap.com/anyURI" source="String 24" version="String 23">AQIDBAUGBwgJAA==</n1:xmlFile>
          </n1:zugferd>
        </n1:convert>
        <n1:analyze level="String 25"/>
      </n1:pdfa>
    - <n1:signature>
```

Hinweis: Sie an dieser Stelle den nachfolgende Funktionstest durchführen oder direkt zur [Anbindung](#) springen, um den Webservices mit einem Programm zu testen.

Sie können den Service jetzt testen indem Sie "Strg+F2" drücken und im Code-Editor die folgende XML-Struktur eingeben (per Copy&Paste).

```
<n0:execute xmlns:n0="http://schema.webpdf.de/1.0/soap/converter">
<n1:operation xmlns:n1="http://schema.webpdf.de/1.0/operation">
<n1:converter compression="true" embedFonts="true" fileExtension="txt">
</n1:converter>
</n1:operation>
<fileContent>SGVsbG8gU0FQIQ==</fileContent>
</n0:execute>
```

Unter "fileContent" ist eine kurze Textdatei BASE64 kodiert, die den Text "Hello SAP!" enthält.

Service-Consumer testen: Ändern Request

The screenshot shows the SAP Service-Consumer test interface. At the top, there's a toolbar with icons for saving, opening, and printing. Below the toolbar, tabs for 'Request', 'Ursprgl. Request', 'Ursprgl. Response', and 'Response' are visible, with 'Request' being the active tab. The main area contains a code editor with the following XML content:

```

1 <n0:execute xmlns:n0="http://schema.webpdf.de/1.0/soap/converter">
2 <n1:operation xmlns:n1="http://schema.webpdf.de/1.0/operation">
3 <n1:converter compression="true" embedFonts="true" fileExtension="txt">
4 </n1:converter>
5 </n1:operation>
6 <fileContent>SGVsbG8gU0FQIQ==</fileContent>
7 </n0:execute>

```

Wenn Sie danach "Ausführen" klicken bekommen Sie ein in XML eingebettetes und in BASE64 kodiertes Dokument als Antwort von webPDF zurückgeliefert.

Service-Consumer testen: Anzeigen Response

The screenshot shows the SAP Service-Consumer test interface with the 'Response' tab selected. The response content is a large block of XML text, which is partially visible here:

```

<?xml version="1.0"?>
<n0:executeResponse xmlns:prx="urn:sap:com:proxy:NPL:/1SAI/TXS3046E6ABFF292BD41E90:752" xmlns:n0="http://schema.webpdf.de/1.0/soap/converter">
<return>JVBERi0xLjQKJdPr6EKMSAwIG9iago8PC9DcmVhdG9yICdaHJvbW1bskKL1Byb2R1Y2VvICht
a2hl1BERiBthzUpCj9DcmVhdGvbkRhGUGKEQ6MjAyMDA1MDQzMj1Mj0rMDAnMDAnKQovTW9k
RGF02SAoDoyMDiWMDUwNDEyMjUyNCswMCcwMCcpj4KZWSkb2jqjMgMCByYmoKPDownY2EgMQov
Qk0gLSvcn1hbD4+CmVuZG9iago0IDAbg2JqCjw8L2Nh1c4NzA2C9CTSavIm9ybWFsPj4KZWSk
b2JqCjYgMCByYmokPDwwRmlsdGvYlC9GbGF0ZURlY29kZQovTGVuZ3RoIDE3O14+IHNoOmVhbQp4
nG1PSwrCQAzdzymyFpzm19YBcSFq18qAB1BbEcPy7wbm04VdnBDy8lZsyUSWVAzQFB0X5UYxpjql
Bm5jeAcWiWKmsEBIHGTekFpHTFMj3BdwcsnJB1bZzx/6p/SraCwP3SwQymIVSdwPAJPt2oQmqL
elB19M4ZlE71zn001akGys96ZUvesSKFIatbcd5GcRyHcwgh0InhaHoFxJCERRe2PL5CeXoWM
Z/MvkEc9OOpbmRzdHJYWW0KZW5kb2jqCjIgMCByYmoKPDownVHlwZSAvUGFnZQovUmVzb3VY2Vz
IDw8L1Byb2NTZXr2IfsvlIeRC1C9UZXh01C9JbWFnZU1g0lYYWdlQyAvsW1hZ2VlXQovRxh0R1N0
YXRIdw8L0c2IDMgMCBScI9HNCA0IDAgUj4+cI9Gb250IDw8L0Y1DugMCBSpj4+PgovTWVkaWFC

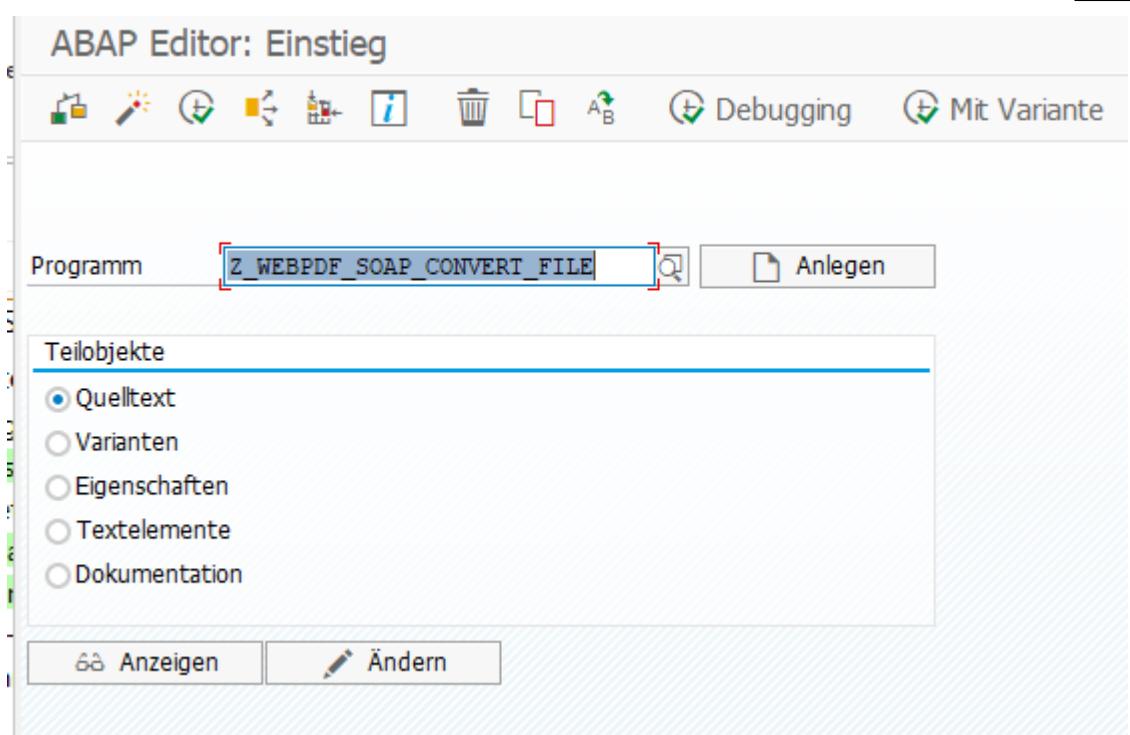
```

3.2 Anbindung

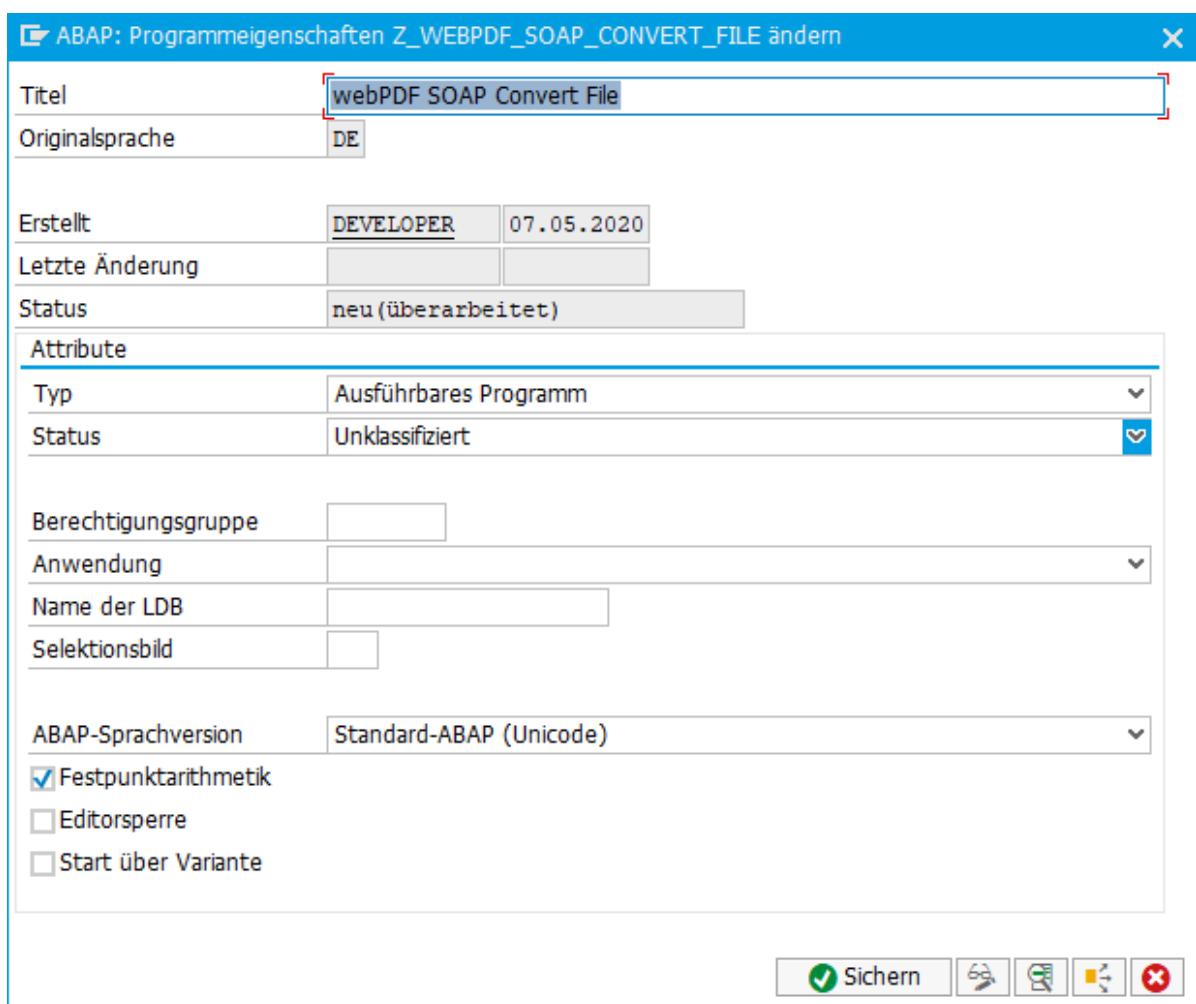
Starten Sie den SAP Client und starten mit "se38" den "ABAP Editor"



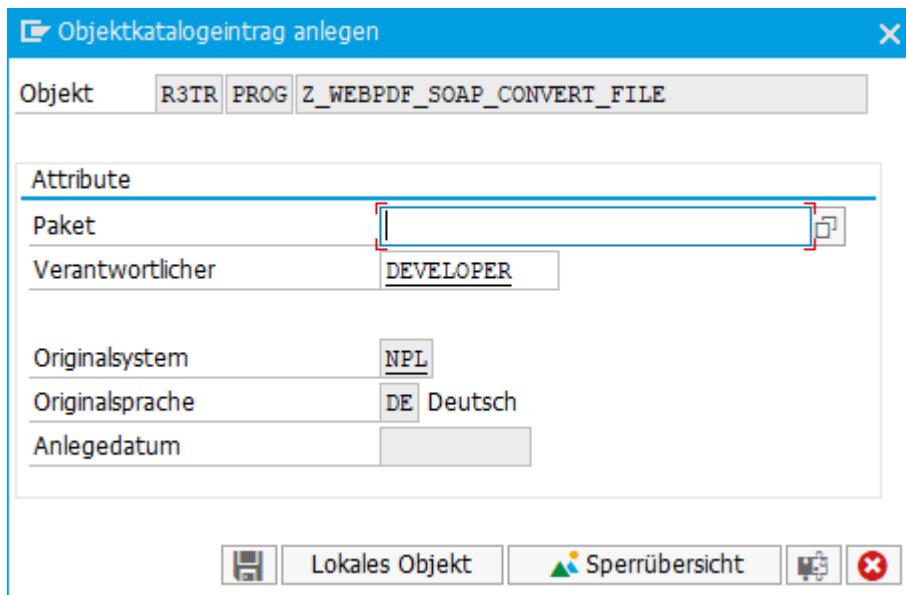
Zunächst müssen Sie ein neues Programm anlegen, um den definierten SOAP Proxy aufzurufen. Geben Sie dazu den Namen "Z_WEBPDF_SOAP_CONVERT_FILE" ein und klicken Sie auf "Anlegen".



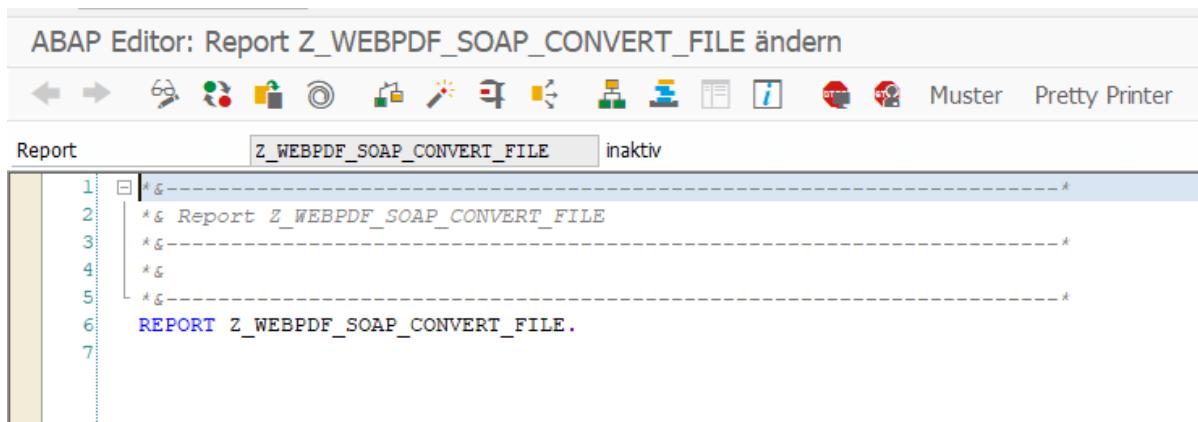
Geben Sie in dem nächsten Dialog "ABAP: Programmeigenschaften" einen Titel ein und wählen Sie unter Typ "Ausführbares Programm". Klicken Sie danach auf "Sichern", um den Dialog abzuschließen.



In dem Dialog "Objektkatalogeintrag anlegen" können Sie optional ein Paket wählen. In diesem Fall lassen wir den Dialog unverändert. Klicken Sie auf den Schalter "Lokales Objekt", um den Dialog abzuschließen.



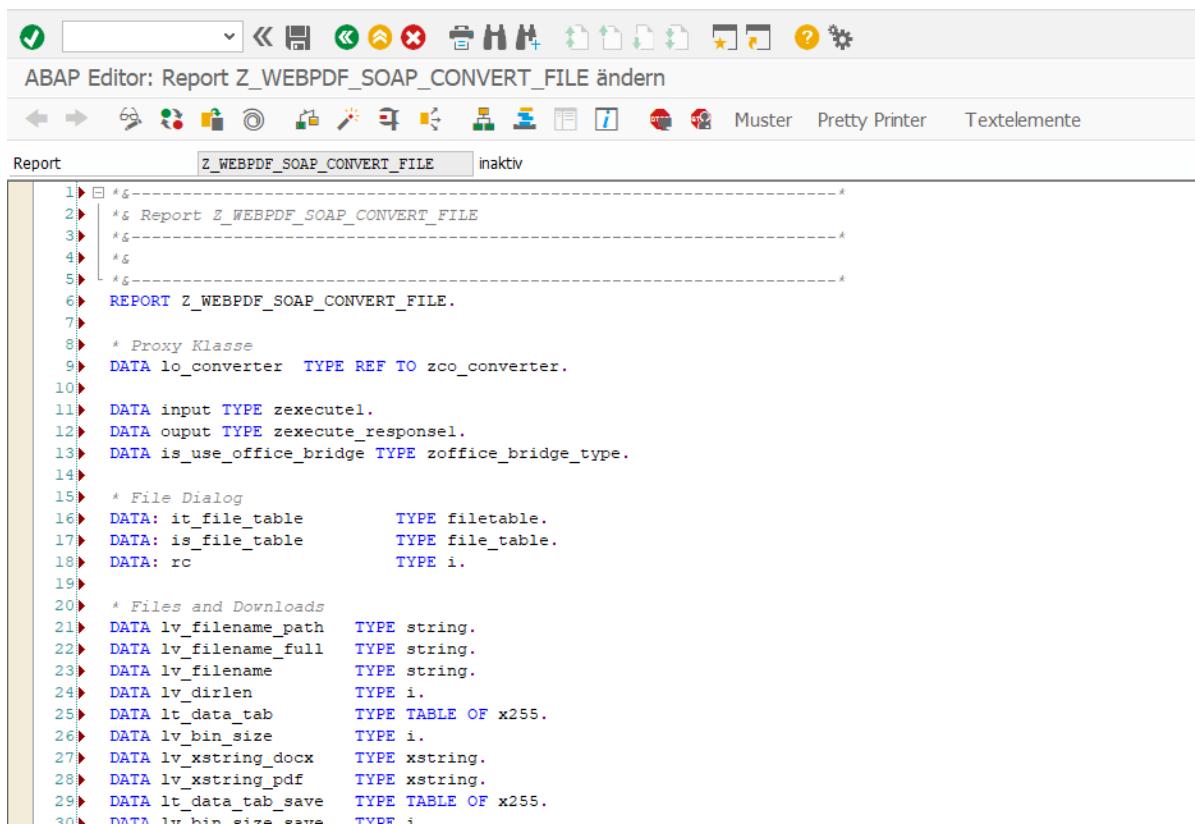
Es erscheint der "leere" Code-Editor mit einem Basis-Code-Fragment.



Im Editor geben Sie den Code für den Webservice-Aufruf ein.

Sie finden diesen auf der Seite "[Anhang: SOAP ABAP Code](#)" oder als Anlage in diesem Dokument mit dem Link "[Source Code](#)".

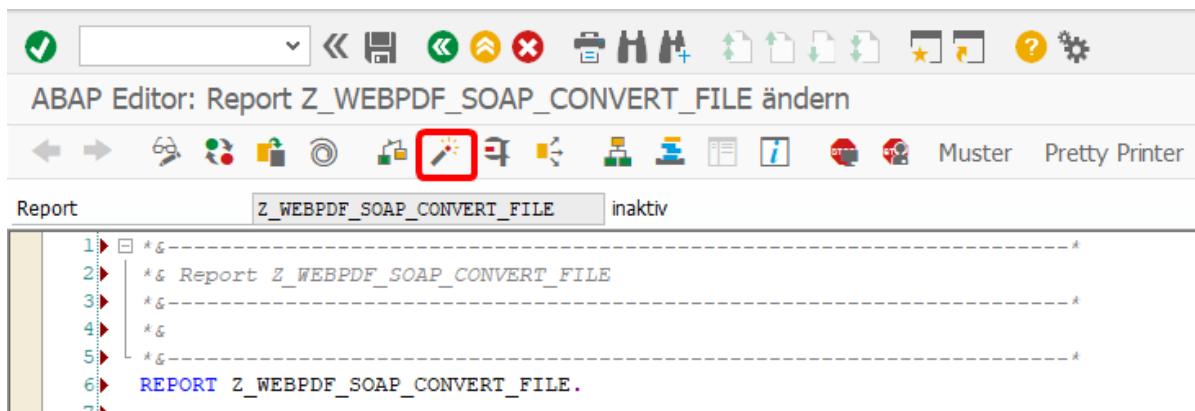
Kopieren Sie den Inhalt aus dem Dokument oder der Datei "Z_WEBPDF_SOAP_CONVERT_FILE.TXT" als Text in die Zwischenablage und fügen sie den Inhalt dann in dem Editor ein.



The screenshot shows the ABAP Editor interface with the title "ABAP Editor: Report Z_WEBPDF_SOAP_CONVERT_FILE ändern". The report name "Z_WEBPDF_SOAP_CONVERT_FILE" is selected in the dropdown. The code area contains the following ABAP code:

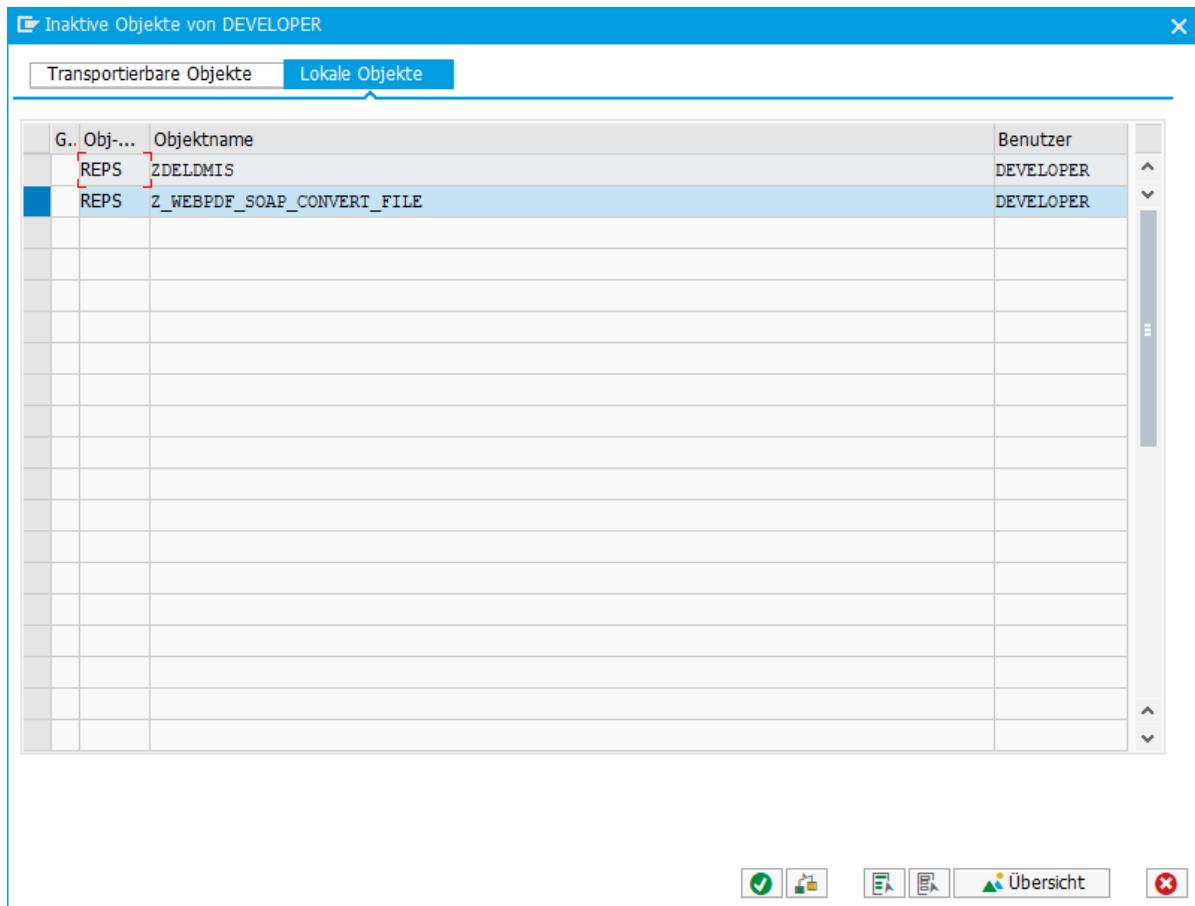
```
1 REPORT Z_WEBPDF_SOAP_CONVERT_FILE.  
2  
3  
4  
5  
6  
7  
8 * Proxy Klasse  
9 DATA lo_converter TYPE REF TO zco_converter.  
10  
11 DATA input TYPE zexecutel.  
12 DATA output TYPE zexecute_responsel.  
13 DATA is_use_office_bridge TYPE zoffice_bridge_type.  
14  
15 * File Dialog  
16 DATA: it_file_table TYPE filetable.  
17 DATA: is_file_table TYPE file_table.  
18 DATA: rc TYPE i.  
19  
20 * Files and Downloads  
21 DATA lv_filename_path TYPE string.  
22 DATA lv_filename_full TYPE string.  
23 DATA lv_filename TYPE string.  
24 DATA lv_dirlen TYPE i.  
25 DATA lt_data_tab TYPE TABLE OF x255.  
26 DATA lv_bin_size TYPE i.  
27 DATA lv_xstring_docx TYPE xstring.  
28 DATA lv_xstring_pdf TYPE xstring.  
29 DATA lt_data_tab_save TYPE TABLE OF x255.  
30 DATA lv_bin_size_save TYPE i.
```

Wenn der Code eingefügt wurde, müssen Sie diesen mit einem Klick auf den entsprechenden Button aktivieren:

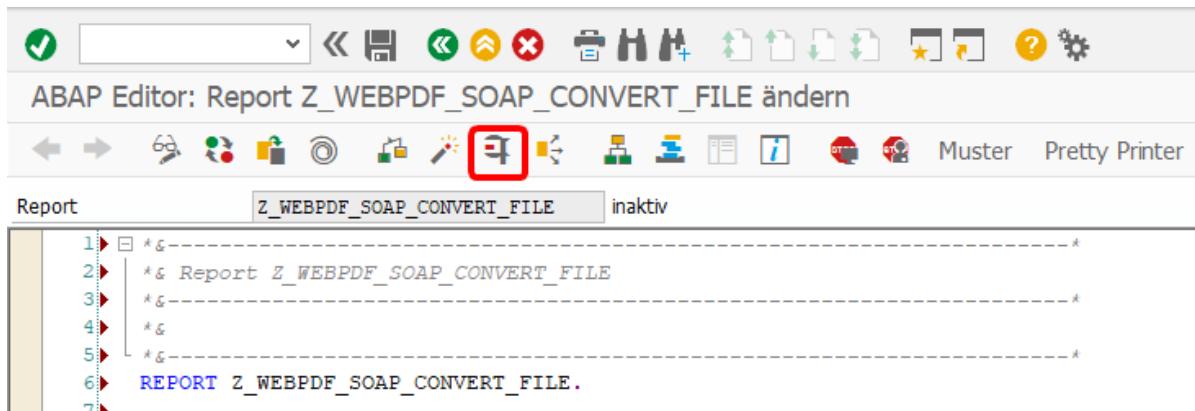


The screenshot shows the ABAP Editor interface with the title "ABAP Editor: Report Z_WEBPDF_SOAP_CONVERT_FILE ändern". The report name "Z_WEBPDF_SOAP_CONVERT_FILE" is selected in the dropdown. The code area contains the same ABAP code as the previous screenshot. The "Activate" button (a pencil icon) in the toolbar is highlighted with a red box.

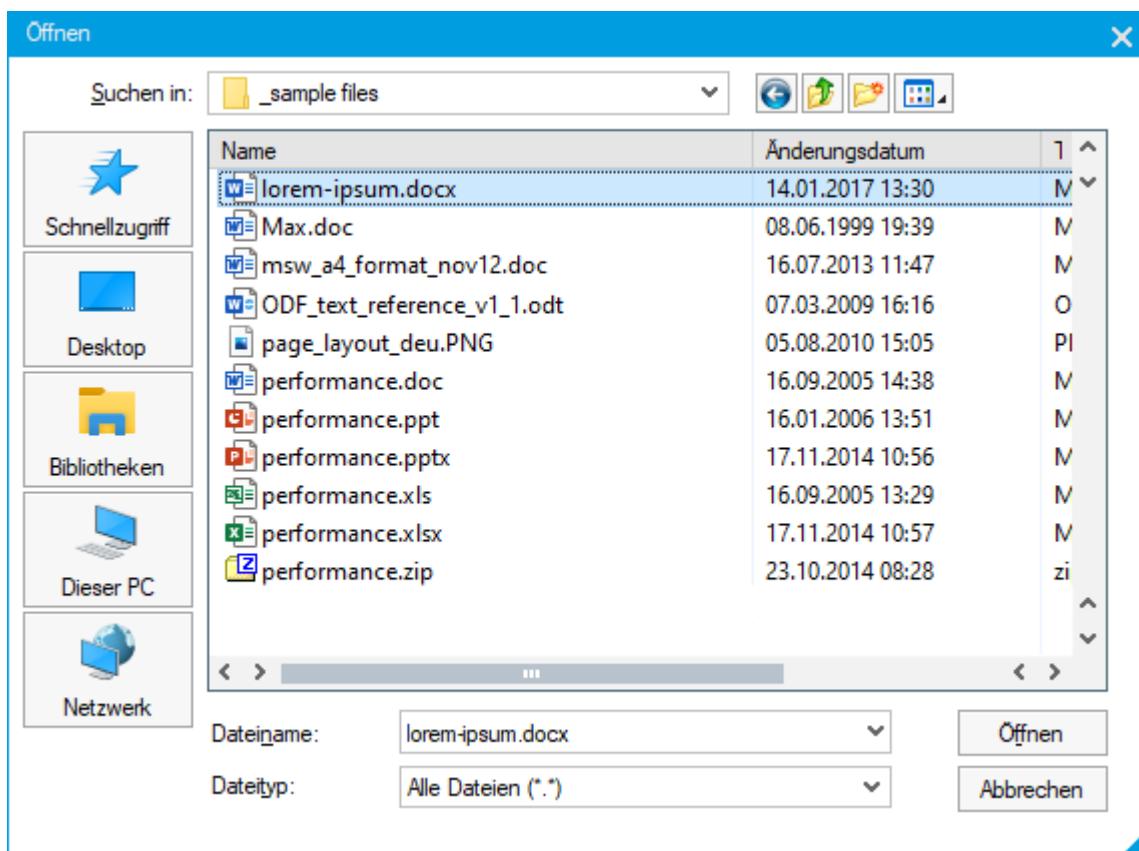
In dem nachfolgenden Dialog müssen Sie die Einstellungen bestätigen.



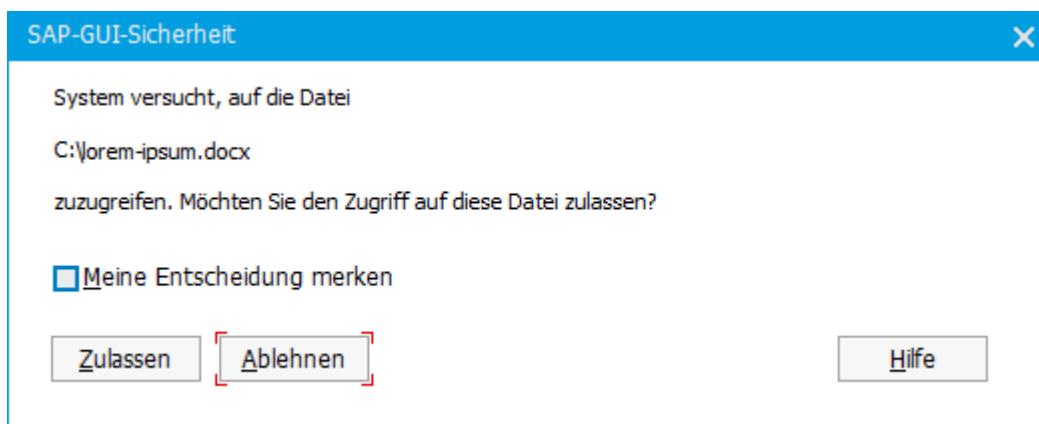
Danach steht das Programm zur Ausführung bereit. Packen Klicken Sie in der Werkzeugeiste auf das entsprechende Symbol, um das Programm auszuführen:



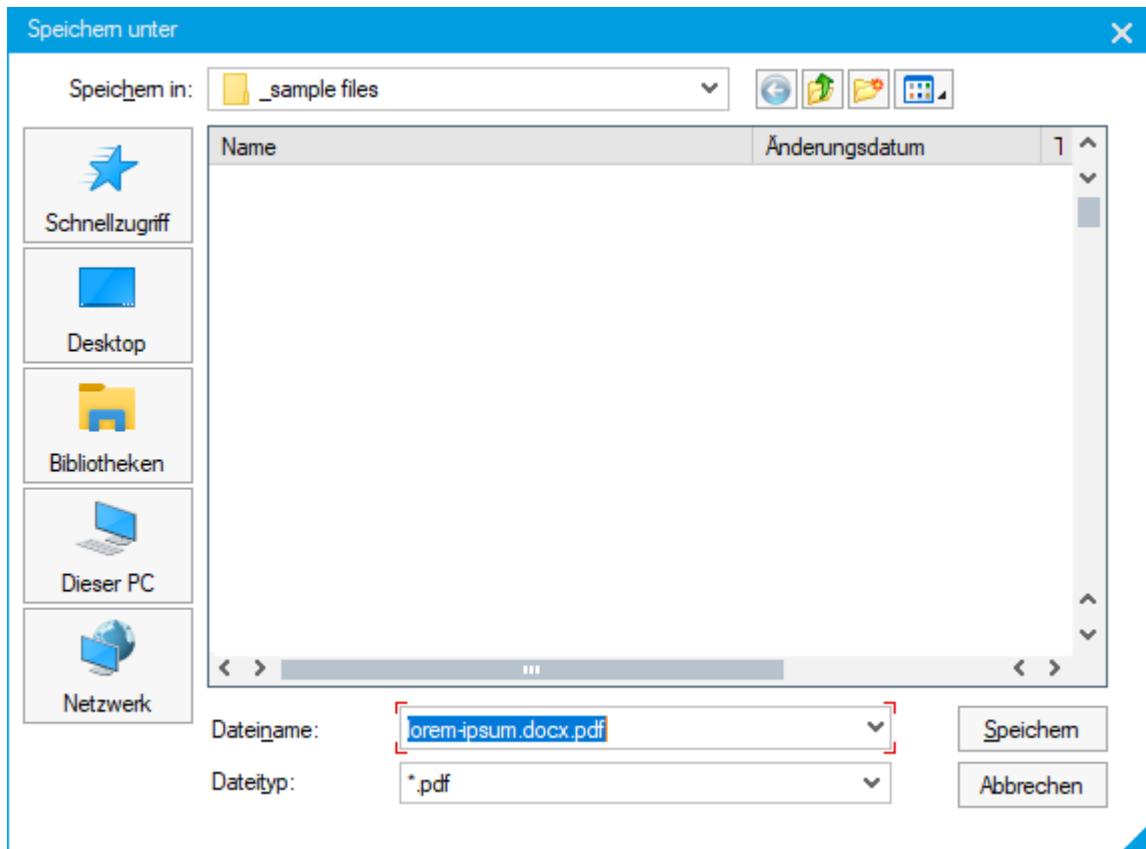
Wenn das Programm startet, werden Sie zur Auswahl einer entsprechenden Datei aufgefordert. Wählen Sie eine Datei die konvertiert werden soll und klicken Sie dann auf "Öffnen".



Bestätigen Sie noch den Zugriff auf die Datei, um den Aufruf des Webservice "Converter" von webPDF zu starten.



Nach der Ausführung des Webservice erscheint ein Dialog, in dem Sie den Namen für die Ausgabedatei angeben können.



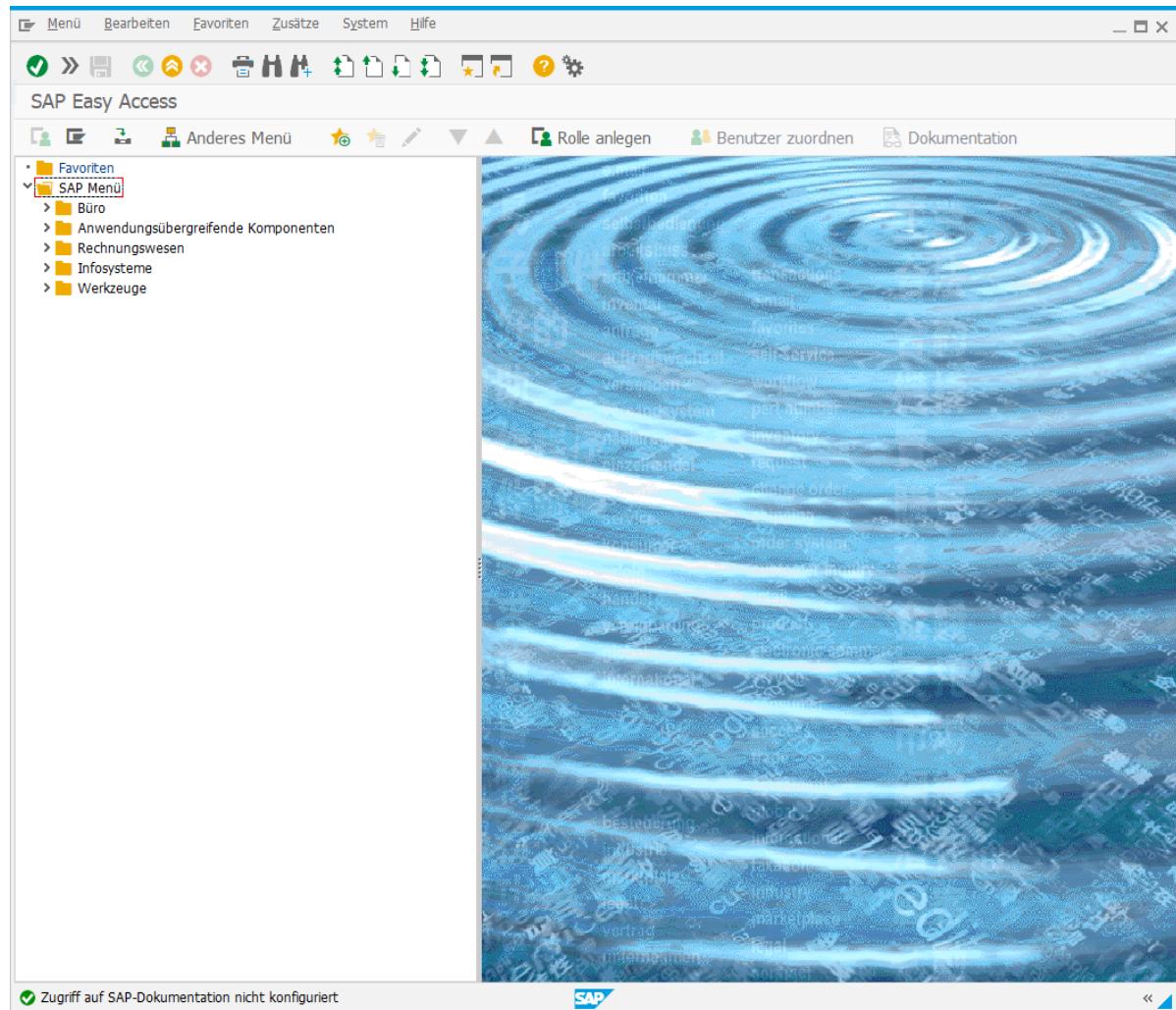
Danach wird das PDF-Dokument unter diesem Namen gespeichert und mit dem im System registrierten PDF-Anzeigeprogramm gestartet.

4 REST-Beispiel

Das nachfolgende Beispiel zeigt, wie Sie die REST-Webservices von webPDF aus SAP heraus verwenden können.

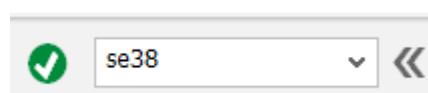
Das Beispiel zeigt dies an Hand des REST-Webservice "Converter". Dieses Beispiel kann leicht auf die anderen Webservice Endpunkte übertragen werden.

Um dieses Beispiel zu verwenden, starten Sie bitte den SAP Business Client (des [Demo-Systems](#)) und melden sich am SAP-System an.

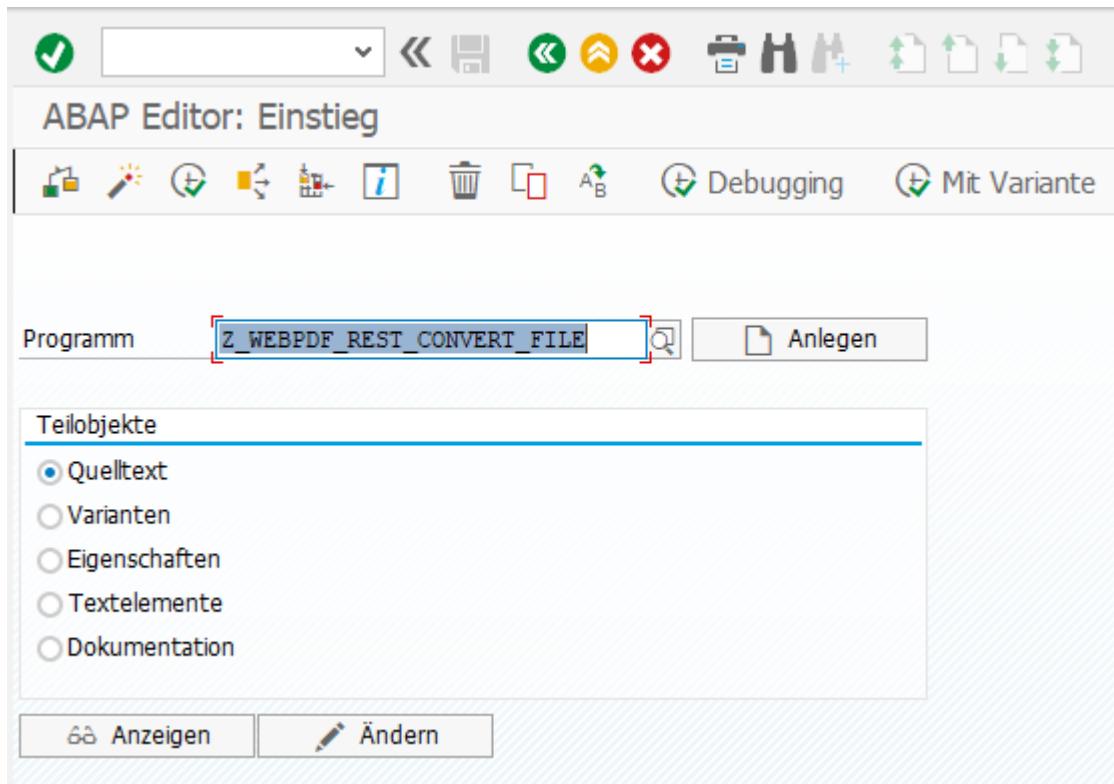


4.1 Anbindung

Starten Sie den SAP Client und starten mit "se38" den "ABAP Editor"



Zunächst müssen Sie ein neues Programm anlegen. Geben Sie dazu den Namen "Z_WEBPDF_REST_CONVERT_FILE" ein und klicken Sie auf "Anlegen".



In dem erscheinenden Dialog geben Sie einen Titel ein und wählen unter Typ "Ausführbares Programm" aus. Klicken Sie danach dann auf "Sichern"

ABAP: Programmeigenschaften Z_WEBPDF_REST_CONVERT_FILE ändern

Titel	webPDF REST Convert File	
Originalsprache	DE	
Erstellt	DEVELOPER	04.05.2020
Letzte Änderung		
Status	neu (überarbeitet)	
Attribute		
Typ	Ausführbares Programm	
Status	Unklassifiziert	
Berechtigungsgruppe		
Anwendung		
Name der LDB		
Selektionsbild		
ABAP-Sprachversion	Standard-ABAP (Unicode)	
<input checked="" type="checkbox"/> Festpunktarithmetik <input type="checkbox"/> Editorsperre <input type="checkbox"/> Start über Variante		
 Sichern    		

Im nächsten Schritt "Objektkatalog anlegen" können Sie die Einstellungen beibehalten und dann auf "Lokales Objekt" klicken, um die Einstellungen zu übernehmen.

Objektkatalogeintrag anlegen

Objekt	R3TR PROG Z_WEBPDF_REST_CONVERT_FILE
Attribute	
Paket	\$TMB
Verantwortlicher	DEVELOPER
Originalsystem	NPL
Originalsprache	DE Deutsch
Anlegedatum	
 Lokales Objekt  	

Es erscheint der "ABAP Editor", in dem Sie jetzt den Code für den Aufruf des "Converter Webservice" eingeben.

ABAP Editor: Report Z_WEBPDF_REST_CONVERT_FILE ändern

```

Report Z_WEBPDF_REST_CONVERT_FILE inaktiv
1*& Report Z_WEBPDF_REST_CONVERT_FILE
2*&
3*&
4*&
5*&
6REPORT Z_WEBPDF_REST_CONVERT_FILE.
7

```

Im Editor geben Sie den Code für den Webservice-Aufruf ein.

Sie finden diesen auf der Seite "[Anhang: REST ABAP Code](#)" oder als Anlage in diesem Dokument mit dem Link "[Source Code](#)".

Kopieren Sie den Inhalt aus dem Dokument oder der Datei "Z_WEBPDF_REST_CONVERT_FILE.TXT" als Text in die Zwischenablage und fügen sie den Inhalt dann in dem Editor ein.

Wichtig: Sie müssen in dem Code die IP-Adresse des Servers anpassen.

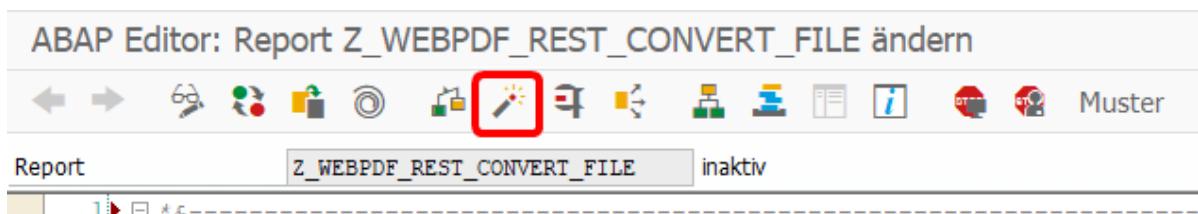
ABAP Editor: Report Z_WEBPDF_REST_CONVERT_FILE ändern

```

Report Z_WEBPDF_REST_CONVERT_FILE inaktiv
1*& Report Z_WEBPDF_REST_CONVERT_FILE
2*&
3*&
4*&
5*&
6REPORT Z_WEBPDF_REST_CONVERT_FILE.
7
8* Web service Options
9
10* Please check the webPDF server IP:
11DATA lv_webpdf_url      TYPE string VALUE 'http://192.168.1.31:8080'.
12* REST converter service: the parameters are passed to the JSON structures.
13DATA lv_ws_parameters    TYPE string VALUE '{"converter": {"embedFonts": false, "jpegQuality": 90, "pages": "", "page": {"met
14* User name and password:
15DATA lv_username          TYPE string VALUE 'user'.
16DATA lv_password          TYPE string VALUE 'user'.
17
18* REST classes
19DATA lo_http_client       TYPE REF TO if_http_client.
20DATA lo_rest_client        TYPE REF TO cl_rest_http_client.
21DATA lo_json               TYPE REF TO /ui2/cl_json.
???

```

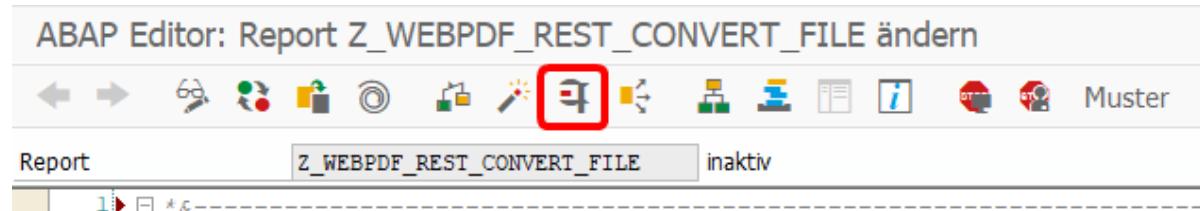
Aktivieren Sie das Programm im nächsten Schritt, in dem Sie in der Werkzeugeiste auf das entsprechende Symbol klicken:



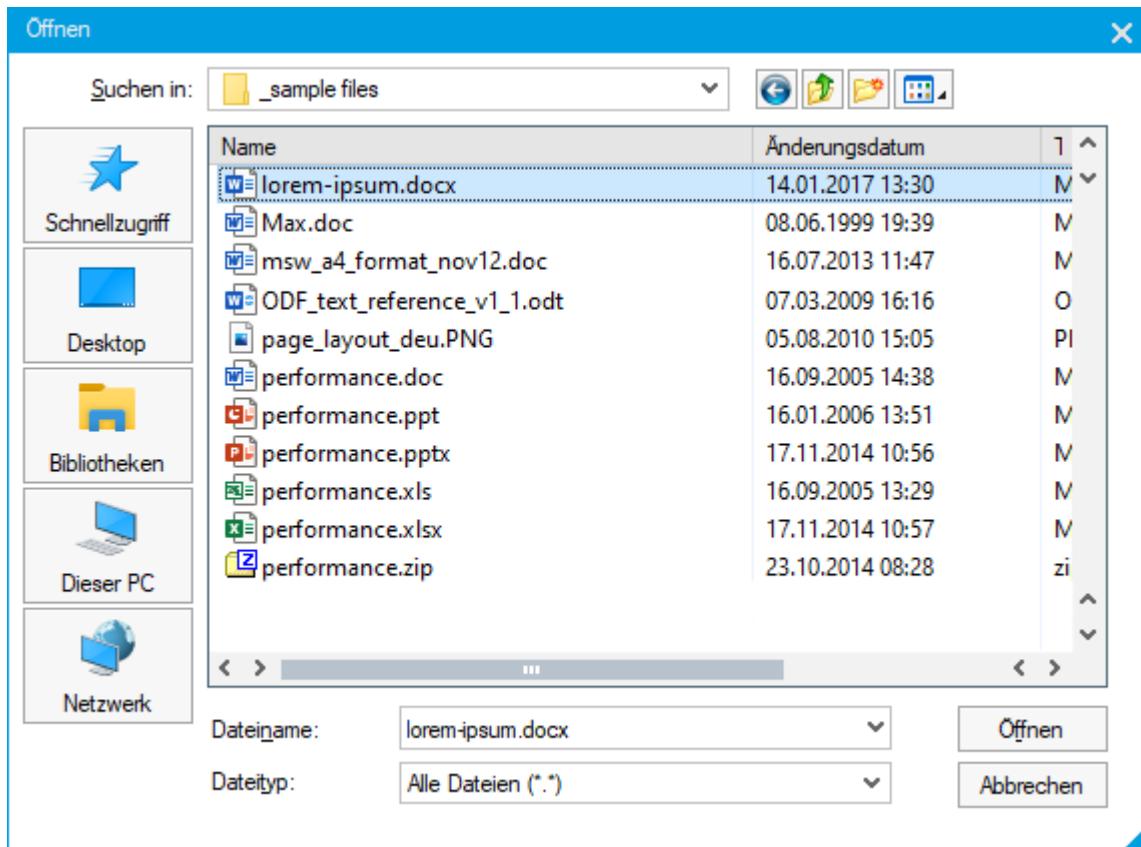
In dem nachfolgenden Dialog können Sie die Einstellungen unverändert übernehmen, um das Programm zu aktivieren.

Inaktive Objekte von DEVELOPER		
Transportierbare Objekte		Lokale Objekte
G.. Obj...	Objektname	Benutzer
REPS	ZDELDMDIS	DEVELOPER
REPS	Z_WEBPDF_REST_CONVERT_FILE	DEVELOPER

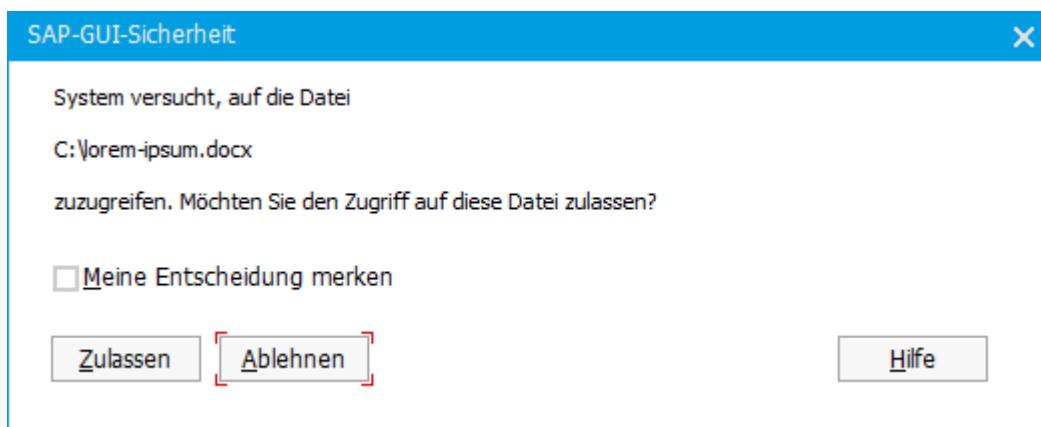
Damit ist das Anlegen des Programms abgeschlossen und Sie können den Programm über das entsprechende Symbol starten:



Wenn die Ausführung startet, fragt das Programm nach der Ausgangsdatei, die in ein PDF konvertiert werden soll. Wählen Sie eine entsprechende Datei von Ihrem lokalen Dateisystem.



Bestätigen Sie die Sicherheitsabfrage vom System für den Zugriff auf die Datei.



Datei wird über den Webservice "Converter" von webPDF konvertiert.

Es erscheint ein Dialog, in dem Sie den Namen für die Ausgabedatei angeben können. Danach wird das PDF-Dokument unter diesem Namen gespeichert und mit dem im System registrierten PDF-Anzeigeprogramm gestartet.

Als Antwort im SAP-Client erhalten Sie eine Auflistung aller durchgeföhrten REST-Calls mit der HTTP-Methode und dem "Result" (Ergebnis) des Aufrufs.

webPDF REST Convert File

```
webPDF REST Convert File

1. LOGIN
   GET:      http://192.168.1.31:8080/webPDF/rest/authentication/user/login
   Result (Token): D426CE6E131C41DF775DF2A60248A0A2
2. UPLOAD
   POST:     http://192.168.1.31:8080/webPDF/rest/documents
   Result (Redirect url): http://192.168.1.31:8080/webPDF/rest/documents/33a07dcdfb344f18cbb6aebcf409cf0/info
3. 303 Redirect
   GET:      http://192.168.1.31:8080/webPDF/rest/documents/33a07dcdfb344f18cbb6aebcf409cf0/info
   Result (DocID): 33a07dcdfb344f18cbb6aebcf409cf0
4. CONVERT
   POST:     http://192.168.1.31:8080/webPDF/rest/converter/33a07dcdfb344f18cbb6aebcf409cf0
   Result (HTTP Status): 303
5. DOWNLOAD
   GET:      http://192.168.1.31:8080/webPDF/rest/documents/33a07dcdfb344f18cbb6aebcf409cf0
   Result (Binary Data):    33.056 Bytes
6. LOGOUT
   GET:      http://192.168.1.31:8080/webPDF/rest/authentication/user/logout
   Result (HTTP Status): 200
```

Wenn alle Operation durchgelaufen sind, wird das konvertierte Dokument als PDF im gleichen Ordner abgelegt. An den originalen Dateinamen wird dabei die Erweiterung ".pdf" angehängt.

5 Anhang

5.1 SOAP ABAP Code

```

*&-----*
*& Report Z_WEBPDF_SOAP_CONVERT_FILE
*&-----*
*&
*&-----*
REPORT Z_WEBPDF_SOAP_CONVERT_FILE.

* Proxy Klasse
DATA lo_converter TYPE REF TO zco_converter.

DATA input TYPE zexecute1.
DATA ouput TYPE zexecute_responsel.
DATA is_use_office_bridge TYPE zoffice_bridge_type.

* File Dialog
DATA: it_file_table      TYPE filetable.
DATA: is_file_table      TYPE file_table.
DATA: rc                  TYPE i.

* Files and Downloads
DATA lv_filename_path     TYPE string.
DATA lv_filename_full    TYPE string.
DATA lv_filename          TYPE string.
DATA lv_dirlen            TYPE i.
DATA lt_data_tab          TYPE TABLE OF x255.
DATA lv_bin_size           TYPE i.
DATA lv_xstring_docx     TYPE xstring.
DATA lv_xstring_pdf       TYPE xstring.
DATA lt_data_tab_save    TYPE TABLE OF x255.
DATA lv_bin_size_save     TYPE i.

* Filesave Dialog
DATA: it_data_tab_save   TYPE TABLE OF x255.
DATA: bin_size_save       TYPE i.

TRY.

* --- 01. OPEN FILE DIALOG
-----
-----
cl_gui_frontend_services=>file_open_dialog(
*   exporting
*     WINDOW_TITLE           =      " Titel des Datei-Öffnen Dialogs
*     DEFAULT_EXTENSION       =      " Vorschlagserweiterung
*     DEFAULT_FILENAME        =      " Vorschlagsdateiname
*     FILE_FILTER             =      " Filterstring für Dateierweiterung
*     WITH_ENCODING           =      " file encoding
*     INITIAL_DIRECTORY       =      " Ausgangsverzeichnis
*     MULTISELECTION          =      " Mehrfachselektion möglich
   CHANGING
     " Tabelle, die selektierte Dateien enthält
     file_table              = it_file_table
     " Rückgabewert: Anzahl Dateien oder -1 falls Fehler auftritt
     rc                      = rc
   *   USER_ACTION             =      " Benutzeraktion( s. Kl.konstanten ACTION_OK,
ACTION_CANCEL)
   *   FILE_ENCODING            =
   EXCEPTIONS
     file_open_dialog_failed = 1

```

```

        cntl_error          = 2
        error_no_gui       = 3
        not_supported_by_gui = 4
        OTHERS             = 5
    ) .
    IF sy-subrc <> 0.
        MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
            WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
    ENDIF.

    * Only one file selected?
    IF rc <> 1.
        EXIT.
    ENDIF.

    * Get file name
    LOOP AT it_file_table INTO is_file_table.
        MOVE is_file_table-filename TO lv_filename_full.
    ENDLOOP.

    * load file from client
    CALL FUNCTION 'GUI_UPLOAD'
        EXPORTING
            filename      = lv_filename_full
            filetype      = 'BIN'
        IMPORTING
            filelength   = lv_bin_size
    *     HEADER        =
        TABLES
            data_tab     = lt_data_tab.

    * get xstring
    CALL FUNCTION 'SCMS_BINARY_TO_XSTRING'
        EXPORTING
            input_length = lv_bin_size
        IMPORTING
            buffer       = lv_xstring_docx
        TABLES
            binary_tab   = lt_data_tab.

    * --- 02. CONVERT
-----
-----


    * Create LO_CONVERTER
    CREATE OBJECT lo_converter
        EXPORTING
            logical_port_name = 'ZWEBPDF_LOGICAL_PORT'.

    * Set parameter
    input-parameters-operation-converter-pages = '*'.
    input-parameters-operation-converter-office_bridge = is_use_office_bridge.
    input-parameters-operation-converter-compression = abap_true.

    * Set content
    input-parameters-file_content = lv_xstring_docx.
    "INPUT-PARAMETERS-FILE_CONTENT = 'SGVsbG8gV2ViUERGIQ=='.

    * Convert operation
    lo_converter->execute(
        EXPORTING
            input           = input
        IMPORTING
            output          = ouput
    ) .

```

```

*      catch CX_AI_SYSTEMFAULT.          "
*      catch ZCX_WEBSERVICE_EXCEPTION.    "

* Result Stream
  lv_xstring_pdf = ouput->parameters->return.

* --- 03. DOWNLOAD RESULT FILE
-----
-----

* get filename and path
  lv_filename = lv_filename_full.
  WHILE lv_filename CA ':'.
    ADD 1 TO sy-fdpos.
    ADD sy-fdpos TO lv_dirlen.
    SHIFT lv_filename LEFT BY sy-fdpos PLACES.
  ENDWHILE.
  lv_filename_path = lv_filename_full+0(lv_dirlen).

* Save dialog
  lv_filename = lv_filename && '.pdf'.
  cl_gui_frontend_services->file_save_dialog(
    EXPORTING
*      WINDOW_TITLE           = " Fenstertitel
*      default_extension       = 'pdf' " Vorschlagserweiterung
*      default_file_name       = lv_filename " Vorschlagsdateiname
*      WITH_ENCODING           =
*      file_filter             = '*.pdf' " Filter Tabelle für Dateityp
*      initial_directory       = lv_filename_path " Ausgangsverzeichnis
*      PROMPT_ON_OVERWRITE     = 'X'
    CHANGING
      filename                = lv_filename " Dateiname für Sichern
      path                     = lv_filename_path " Pfad zu Datei
      fullpath                 = lv_filename_full " Pfad + Dateiname
      user_action              = rc " Benutzeraktion ( K.Konst. ACTION_OK,
ACTION_OVERWRITE usw.)
*      FILE_ENCODING           =
    EXCEPTIONS
      cntl_error               = 1
      error_no_gui             = 2
      not_supported_by_gui     = 3
      invalid_default_file_name = 4
      OTHERS                   = 5
    ).
  IF sy-subrc <> 0.
    MESSAGE ID sy-msgid TYPE sy-msgtyp NUMBER sy-msgno
      WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
  ENDIF.

* Check user action
  IF rc = cl_gui_frontend_services->action_cancel.
    EXIT.
  ENDIF.

* Get xstring
  CALL FUNCTION 'SCMS_XSTRING_TO_BINARY'
    EXPORTING
      buffer        = lv_xstring_pdf
    IMPORTING
      output_length = lv_bin_size_save
    TABLES
      binary_tab    = lt_data_tab_save
    EXCEPTIONS
      OTHERS        = 1.

```

```

IF sy-subrc <> 0.
MESSAGE ID sy-msgid TYPE sy-msgt y NUMBER sy-msgno
      WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.

* Download file
CALL FUNCTION 'GUI_DOWNLOAD'
  EXPORTING
    bin_filesize = lv_bin_size_save
    filename     = lv_filename_full
    filetype     = 'BIN'
  TABLES
    data_tab      = lt_data_tab_save
  EXCEPTIONS
    OTHERS        = 1.
IF sy-subrc <> 0.
MESSAGE ID sy-msgid TYPE sy-msgt y NUMBER sy-msgno
      WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.

* Show pdf file (ShellExecute)
cl_gui_frontend_services->execute(
  EXPORTING
    document          = lv_filename_full " Pfad+Dokumentname
*     application       =                      " Pfad + Anwendungsname
*     parameter         =                      " Parameter für Anwendung
*     default_directory =                      " Vorschlagsverzeichnis
*     maximized        =                      " Fenster maximiert anzeigen
*     minimized         =                      " Fenster minimiert anzeigen
*     synchronous       =                      " Wenn 'X': die Anwendung läuft
synchron
    operation          = 'OPEN'           " Reserviert: Verb für ShellExecute
  EXCEPTIONS
    cntl_error         = 1               " Controlfehler
    error_no_gui       = 2               " Kein GUI verfügbar
    bad_parameter      = 3               " Falsche Parameterkombination
    file_not_found     = 4               " Datei nicht gefunden
    path_not_found     = 5               " Pfad nicht gefunden
    file_extension_unknown = 6           " Anwendung für angegebene Erweiterung
wurde nicht gefunden
    error_execute_failed = 7             " Durchführung der Anwendung oder des
Dokuments fehlgeschlagen
    synchronous_failed = 8             " Synchroner Aufruf der Applikation
nicht möglich
    not_supported_by_gui = 9            " Nicht unterstützt von GUI
    OTHERS             = 10
).
IF sy-subrc <> 0.
MESSAGE ID sy-msgid TYPE sy-msgt y NUMBER sy-msgno
      WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.

CATCH cx_root INTO DATA(e_text).
  WRITE: /.
  WRITE: / 'Exception:', e_text->get_text( ).
ENDTRY.

```

5.2 REST ABAP Code

```

*&-----*
*& Report Z_WEBPDF_REST_CONVERT_FILE
*&-----*
*&

```

```

*&-----
REPORT Z_WEBPDF_REST_CONVERT_FILE.

* Web service Options

* Please check the webPDF server IP:
DATA lv_webpdf_url      TYPE string VALUE 'http://192.168.1.31:8080'.
* REST converter service: the parameters are passed to the JSON structures.
  DATA lv_ws_parameters   TYPE string VALUE '{"converter":'
  {"embedFonts":false,"jpegQuality":90,"pages":"*","page":'
  {"metrics":"mm","width":210,"height":297,"top":20,"right":20,"bottom":20,"left":20}}}'.
* User name and password:
  DATA lv_username        TYPE string VALUE 'user'.
  DATA lv_password         TYPE string VALUE 'user'.

* REST classes
  DATA lo_http_client     TYPE REF TO if_http_client.
  DATA lo_rest_client      TYPE REF TO cl_rest_http_client.
  DATA lo_json             TYPE REF TO /ui2/cl_json.

* REST help variables
  DATA lo_request          TYPE REF TO if_rest_entity.
  DATA lo_rest_multi       TYPE REF TO if_rest_multipart_entity.
  DATA lv_rest_url          TYPE string.
  DATA lo_response          TYPE REF TO if_rest_entity.
  DATA lv_json_response     TYPE string.
  DATA lv_http_status       TYPE string.

* Token and DocumentID for webPDF session
  DATA lv_token            TYPE string.
  DATA lv_docID             TYPE string.

* Fileopen Dialog
  DATA it_file_table        TYPE filetable.
  DATA is_file_table        TYPE file_table.
  DATA rc                   TYPE i.

* Files and Downloads
  DATA lv_filename_path     TYPE string.
  DATA lv_filename_full     TYPE string.
  DATA lv_filename           TYPE string.
  DATA lv_dirlen             TYPE i.
  DATA lt_data_tab           TYPE TABLE OF x255.
  DATA lv_bin_size            TYPE i.
  DATA lv_xstring_docx      TYPE xstring.
  DATA lv_xstring_pdf        TYPE xstring.
  DATA lt_data_tab_save      TYPE TABLE OF x255.
  DATA lv_bin_size_save       TYPE i.

* JSON->ABAP: Login
  TYPES: BEGIN OF ty_login,
    token TYPE string,
  END OF ty_login.
  DATA ls_login TYPE ty_login.

* JSON->ABAP: Info
  TYPES: BEGIN OF ty_info,
    documentId TYPE string,
    fileSize TYPE i,
    fileName TYPE string,
  END OF ty_info.
  DATA ls_info TYPE ty_info.

TRY.

```

```

* --- OPEN DIALOG
-----
-----

    cl_gui_frontend_services->FILE_OPEN_DIALOG (
*   exporting
*     WINDOW_TITLE           =      " Titel des Datei-Öffnen Dialogs
*     DEFAULT_EXTENSION       =      " Vorschlagserweiterung
*     DEFAULT_FILENAME        =      " Vorschlagsdateiname
*     FILE_FILTER             =      " Filterstring für Dateierweiterung
*     WITH_ENCODING           =      " file encoding
*     INITIAL_DIRECTORY       =      " Ausgangsverzeichnis
*     MULTISELECTION          =      " Mehrfachselektion möglich
*
*   changing
*     " Tabelle, die selektierte Dateien enthält
*     FILE_TABLE              = it_file_table
*     " Rückgabewert: Anzahl Dateien oder -1 falls Fehler auftritt
*     RC                      = rc
*     USER_ACTION              =      " Benutzeraktion( s. Kl.konstanten ACTION_OK,
ACTION_CANCEL)
*     FILE_ENCODING            =
*
*   exceptions
*     FILE_OPEN_DIALOG_FAILED = 1
*     CNTL_ERROR              = 2
*     ERROR_NO_GUI             = 3
*     NOT_SUPPORTED_BY_GUI     = 4
*     OTHERS                  = 5
).
if sy-subrc <> 0.
  message id sy-msgid type sy-msgtty number sy-msgno
    with sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
endif.

* Only one file selected?
IF rc <> 1.
  EXIT.
ENDIF.

* Get file name
LOOP AT it_file_table INTO is_file_table.
  MOVE is_file_table-filename TO lv_filename_full.
ENDLOOP.

* --- INIT
-----
-----

* Create HTTP client instance
CALL METHOD cl_http_client->CREATE_BY_URL
  exporting
    URL                  = lv_webpdf_url
*    PROXY_HOST           =      " logische Destination (Wird bei Funktionsaufruf
angegeben)
*    PROXY_SERVICE         =      " Portnummer
*    SSL_ID                =      " SSL Identität
*    SAP_USERNAME          =      " R/3-System, Anmeldename des Benutzers
*    SAP_CLIENT             =      " R/3-System, Mandantennummer aus Anmeldung
  importing
    CLIENT                = lo_http_client
  exceptions
    ARGUMENT_NOT_FOUND   = 1
    PLUGIN_NOT_ACTIVE    = 2
    INTERNAL_ERROR        = 3
    OTHERS                 = 4

```

```

.
if sy-subrc <> 0.
  message id sy-msgid type sy-msgty number sy-msgno
    with sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
endif.

lo_http_client->propertytype_logon_popup = lo_http_client->co_disabled.
CALL METHOD lo_http_client->AUTENTICATE
  exporting
*   PROXY_AUTHENTICATION = ' '      " Proxy-Logon (= 'X')
*   CLIENT                =      " R/3-System, Mandantennummer aus Anmeldung
    USERNAME               = lv_username
    PASSWORD               = lv_password
*   LANGUAGE               =      " R/3-System, aktuelle Sprache
.

* Create REST client instance
CREATE OBJECT LO_REST_CLIENT
  exporting
    IO_HTTP_CLIENT = lo_http_client.
.

* Class to convert a JSON structure to an ABAP object
CREATE OBJECT lo_json.

* --- LOGIN
-----
-----

* GET: /webPDF/rest/authentication/user/login
lv_rest_url = lv_webpdf_url && '/webPDF/rest/authentication/user/login'.
WRITE: / '1. LOGIN', /5(7) 'GET:', lv_rest_url.

* URL
cl_http_utility=>SET_REQUEST_URI(
  exporting
    REQUEST = lo_http_client->request      " HTTP Framework (iHTTP) HTTP Request
    URI     = lv_rest_url      " URI-String (in Form von /path?query-string)
).
.

* HTTP GET
lo_rest_client->if_rest_client~get( ).

* HTTP response
lo_response = lo_rest_client->if_rest_client~get_response_entity( ).

* HTTP return status
lv_http_status = lo_response->get_header_field('status_code').

* HTTP JSON return string
lv_json_response = lo_response->get_string_data( ).

* Get token
lo_json->DESERIALIZE( exporting JSON = lv_json_response changing DATA = ls_login ).
lv_token = ls_login-TOKEN.

WRITE: /5 'Result (Token):', lv_token.

* --- FILE
-----
-----

* load file from client
CALL FUNCTION 'GUI_UPLOAD'
  exporting
    filename = lv_filename_full

```

```

        filetype    = 'BIN'
        importing
            filelength = lv_bin_size
*      HEADER      =
        tables
            data_tab   = lt_data_tab.

* get xstring
    CALL FUNCTION 'SCMS_BINARY_TO_XSTRING'
        exporting
            input_length = lv_bin_size
        importing
            buffer       = lv_xstring_docx
        tables
            binary_tab   = lt_data_tab.

* get filename and path
lv_filename = lv_filename_full.
WHILE lv_filename CA ':'.
    ADD 1 TO sy-fdpos.
    ADD sy-fdpos TO lv_dirlen.
    SHIFT lv_filename LEFT BY sy-fdpos PLACES.
ENDWHILE.
lv_filename_path = lv_filename_full+0(lv_dirlen).

* --- UPLOAD
-----
-----
* POST: /webPDF/rest/documents
lv_rest_url = lv_webpdf_url && '/webPDF/rest/documents'.
WRITE: / '2. UPLOAD', /5(7) 'POST:', lv_rest_url.

* Create a multipart
lo_rest_multi ?= lo_rest_client->if_rest_client~create_request_entity(
                           iv_multipart = abap_true
                         ).
* Set ContentType
lo_rest_multi->set_header_field(
    exporting
        iv_name  = if_http_header_fields->content_type
        iv_value = if_rest_media_type->gc_multipart_form_data
  ).

* Create a Multipart Entity
lo_rest_multi->create_entity(
    exporting
        iv_multipart          = abap_false
    RECEIVING
        ro_entity             = lo_request           " REST Entity
  ).

* Content-Disposition: form-data; name="filedata"
lo_request->set_header_field(
    exporting
        iv_name  = if_http_header_fields->content_disposition
        iv_value = 'form-data; name="filedata"'
  ).

* Binary Data
lo_request->SET_BINARY_DATA( lv_xstring_docx ).

* Header
CALL METHOD lo_rest_client->if_rest_client~set_request_header
    exporting
        iv_name  = 'token'

```

```

        iv_value = lv_token.
CALL METHOD lo_rest_client->if_rest_client~set_request_header
  exporting
    iv_name  = 'accept'
    iv_value = 'application/json'. "Set your header .

* URL
cl_http_utility=>SET_REQUEST_URI(
  exporting
    REQUEST = lo_http_client->request      " HTTP Framework (iHTTP) HTTP Request
    URI     = lv_rest_url      " URI-String (in Form von /path?query-string)
).
.

* POST
lo_rest_client->if_rest_resource~post( lo_rest_multi ).

* Collect response
lo_response = lo_rest_client->if_rest_client~get_response_entity( ).

* HTTP return status
lv_http_status = lo_response->get_header_field( '~status_code' ).

* Return redirect url
lv_rest_url = lo_response->get_header_field( IV_NAME = 'Location' ).

WRITE: /5 'Result (Redirect url):', lv_rest_url.

* --- 303 Redirect
-----
-----
* GET: Redirect URL: /webPDF/rest/converter/{docID}/Info
WRITE: / '3. 303 Redirect', /5(7) 'GET:', lv_rest_url.

* URL
cl_http_utility=>SET_REQUEST_URI(
  exporting
    REQUEST = lo_http_client->request      " HTTP Framework (iHTTP) HTTP Request
    URI     = lv_rest_url      " URI-String (in Form von /path?query-string)
).
.

* HTTP GET
lo_rest_client->if_rest_client~get( ).

* HTTP response
lo_response = lo_rest_client->if_rest_client~get_response_entity( ).

* HTTP return status
lv_http_status = lo_response->get_header_field( '~status_code' ).

* HTTP JSON return string
lv_json_response = lo_response->get_string_data( ).

* Read JSON
lo_json->DESERIALIZE( exporting JSON = lv_json_response changing DATA = ls_info ).
lv_docID = ls_info-DOCUMENTID.

WRITE: /5 'Result (DocID):', lv_docID.

* --- CONVERT
-----
-----
* POST: /webPDF/rest/converter/{docID}
lv_rest_url = lv_webpdf_url && '/webPDF/rest/converter/' && lv_docID.
WRITE: / '4. CONVERT', /5(7) 'POST:', lv_rest_url.

```

```

* Clear request entity
lo_rest_client->REFRESH_REQUEST( ).

* Set token
CALL METHOD lo_rest_client->if_rest_client~set_request_header
  exporting
    iv_name  = 'token'
    iv_value = lv_token.

* Create Request Entity
lo_request = lo_rest_client->if_rest_client~create_request_entity( ).
lo_request->set_content_type( iv_media_type = if_rest_media_type=>GC_APPL_JSON ).

* Converter Properties
lo_request->SET_STRING_DATA( lv_ws_parameters ).

* URL
cl_http_utility=>set_request_uri(
  exporting
    request = lo_http_client->request      " HTTP Framework (iHTTP) HTTP Request
    uri     = lv_rest_url                   " URI String (in the Form of /path?query-
string)
).

* POST
lo_rest_client->if_rest_resource~post( lo_request ).

* Collect response
lo_response = lo_rest_client->if_rest_client~get_response_entity( ).

* HTTP return status
lv_http_status = lo_response->get_header_field( '~status_code' ).

WRITE: /5 'Result (HTTP Status):', lv_http_status.

* --- DOWNLOAD
-----
-----
* GET: /webPDF/rest/documents/{docID}
lv_rest_url = lv_webpdf_url && '/webPDF/rest/documents/' && lv_docID.
WRITE: / '5. DOWNLOAD', /5(7) 'GET:', lv_rest_url.

* Clear request entity
lo_rest_client->REFRESH_REQUEST( ).

* Set token
CALL METHOD lo_rest_client->if_rest_client~set_request_header
  exporting
    iv_name  = 'token'
    iv_value = lv_token.

* URL
cl_http_utility=>set_request_uri(
  exporting
    request = lo_http_client->request
    uri     = lv_rest_url
).
.

* HTTP GET
lo_rest_client->if_rest_client~get( ).

* HTTP response
lo_response = lo_rest_client->if_rest_client~get_response_entity( ).

```

```

* HTTP return status
lv_http_status = lo_response->get_header_field( '~status_code' ).

* HTTP return binary string
lv_xstring_pdf = lo_response->GET_BINARY_DATA( ).

* Save Result PDF
CALL FUNCTION 'SCMS_XSTRING_TO_BINARY'
  exporting
    buffer      = lv_xstring_pdf
  importing
    output_length = lv_bin_size_save
  tables
    binary_tab   = lt_data_tab_save
  exceptions
    OTHERS           = 1.

if sy-subrc <> 0.
  message id sy-msgid type sy-msgtty number sy-msgno
            with sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
endif.

* Save dialog
lv_filename = lv_filename && '.pdf'.
cl_gui_frontend_services=>FILE_SAVE_DIALOG(
  exporting
    WINDOW_TITLE          = "Fenstertitel"
    DEFAULT_EXTENSION     = 'pdf' " Vorschlagserweiterung
    DEFAULT_FILE_NAME     = lv_filename " Vorschlagsdateiname
  *  WITH_ENCODING        =
    FILE_FILTER           = '*.pdf' " Filter Tabelle für Dateityp
    INITIAL_DIRECTORY     = lv_filename_path " Ausgangsverzeichnis
  *  PROMPT_ON_OVERWRITE = 'X'
  changing
    FILENAME             = lv_filename " Dateiname für Sichern
    PATH                 = lv_filename_path " Pfad zu Datei
    FULLPATH             = lv_filename_full " Pfad + Dateiname
    USER_ACTION          = rc " Benutzeraktion ( K.Konst. ACTION_OK,
ACTION_OVERWRITE usw.)
  *  FILE_ENCODING        =
  exceptions
    CNTL_ERROR           = 1
    ERROR_NO_GUI         = 2
    NOT_SUPPORTED_BY_GUI = 3
    INVALID_DEFAULT_FILE_NAME = 4
    OTHERS               = 5
).
if sy-subrc <> 0.
  message id sy-msgid type sy-msgtty number sy-msgno
            with sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
endif.

if rc = CL_GUI_FRONTEND_SERVICES=>ACTION_CANCEL.
  EXIT.
endif.

* Download file
CALL FUNCTION 'GUI_DOWNLOAD'
  exporting
    bin_filesize = lv_bin_size_save
    filename     = lv_filename_full
    filetype     = 'BIN'
  tables
    data_tab     = lt_data_tab_save
  exceptions
    OTHERS           = 1.

```

```

if sy-subrc <> 0.
  message id sy-msgid type sy-msgty number sy-msgno
    with sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
endif.

WRITE: /5 'Result (Binary Data):', lv_bin_size_save, 'Bytes'.

* --- LOGOUT
-----
-----

* /webPDF/rest/authentication/user/logout
lv_rest_url = lv_webpdf_url && '/webPDF/rest/authentication/user/logout'.
WRITE: / '6. LOGOUT', /5(7) 'GET:', lv_rest_url.

* Clear request entity
lo_rest_client->REFRESH_REQUEST( ).

* Set token
CALL METHOD lo_rest_client->if_rest_client~set_request_header
  exporting
    iv_name  = 'token'
    iv_value = lv_token.

* URL
cl_http_utility=>set_request_uri(
  exporting
    request = lo_http_client->request
    uri     = lv_rest_url
  ).

* HTTP GET
lo_rest_client->if_rest_client~get( ).

* HTTP response
lo_response = lo_rest_client->if_rest_client~get_response_entity( ).

* HTTP return status
lv_http_status = lo_response->get_header_field('~status_code').

WRITE: /5 'Result (HTTP Status):', lv_http_status.

* --- SHOW PDF FILE (ShellExecute)
-----
-----

WRITE: / '7. SHOW PDF FILE'.
WRITE: /5 'File name:', lv_filename_full.

cl_gui_frontend_services=>EXECUTE(
  exporting
    DOCUMENT      = lv_filename_full" Pfad+Dokumentname
*     APPLICATION   = " Pfad + Anwendungsname
*     PARAMETER     = " Parameter für Anwendung
*     DEFAULT_DIRECTORY = " Vorschlagsverzeichnis
*     MAXIMIZED     = " Fenster maximiert anzeigen
*     MINIMIZED     = " Fenster minimiert anzeigen
*     SYNCHRONOUS    = " Wenn 'X': die Anwendung läuft synchron
    OPERATION      = 'OPEN'    " Reserviert: Verb für ShellExecute
  exceptions
    CNTL_ERROR      = 1
    ERROR_NO_GUI    = 2
    BAD_PARAMETER    = 3
    FILE_NOT_FOUND   = 4
    PATH_NOT_FOUND   = 5
    FILE_EXTENSION_UNKNOWN = 6
    ERROR_EXECUTE_FAILED = 7

```

```

SYNCHRONOUS_FAILED      = 8
NOT_SUPPORTED_BY_GUI    = 9
OTHERS                  = 10
).

if sy-subrc <> 0.
  message id sy-msgid type sy-msgty number sy-msgno
           with sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
endif.

CATCH cx_root INTO DATA(e_text).
  WRITE: /.
  WRITE: / 'Exception:', e_text->get_text( ).
ENDTRY.

```

5.3 Links

Website	https://www.webpdf.de
Dokumentation	https://www.webpdf.de/dokumentation.html
Addressen	https://portal.webpdf.de/webPDF/help/doc/de/adressen.htm
Webservices	https://portal.webpdf.de/webPDF/help/doc/de/webservice_general.htm
Installation für Windows	https://download.softvision.de/?product=webpdf
Pakete für Linux	https://packages.softvision.de/
Docker Container	https://github.com/softvision-dev/webpdf-docker
VMware Image	https://download.softvision.de/?product=webpdf_vm
webPDF-Portal (Live Demo)	https://portal.webpdf.de/webPDF/
Support	https://www.webpdf.de/support
SoftVision Development Website	https://www.softvision.de