EGIZ E-Government Innovationszentrum

www.egiz.gv.at
E-Mail: post@egiz.gv.at
Telefon: ++43 (316) 873 5514
Fax: ++43 (316) 873 5520
Inffeldgasse 16a / 8010 Graz / Austria

Dokumentation PDF-AS Schnittstelle

Erweiterung von PDF-AS um eine Schnittstelle zur Anbindung externer Web-Anwendungen

Version 1.1, 29.01.2009

Thomas Zefferer – thomas.zefferer@iaik.tugraz.at

Zusammenfassung:

Ziel dieses Projekts war die Erweiterung von PDF-AS zur Schaffung einer neuen Schnittstelle, über die externe Webapplikationen PDF-AS als Webservice nutzen können. Dadurch wird eine Einbindung der von PDF-AS zur Verfügung gestellten Funktionalität in andere Webanwendungen vereinfacht. Die implementierte Schnittstelle erlaubt die Übermittlung von beliebigen PDF Dokumenten an PDF-AS, wo diese dann signiert und zur weiteren Verarbeitung an die aufrufende Applikation retourniert werden.

Inhaltsverzeichnis:

Anmerkung: Zur besseren Lesbarkeit wurde in diesem Dokument teilweise auf geschlechtsspezifische Formulierungen verzichtet. Die verwendeten Formulierungen richten sich jedoch ausdrücklich an beide Geschlechter.

Revision History

Version	Datum	Autor(en)	
0.1	23.07.2007	Thomas Zefferer	Erster Entwurf
0.2	23.07.2007	Thomas Knall	Anmerkungen
0.3	23.07.2007	Thomas Zefferer	Überarbeitung
0.4	24.07.2007	Thomas Zefferer	Überarbeitung
1.0	28.01.2008	Thomas Zefferer	Letzte Anpassungen
1.1	29.01.2009	Thomas Knall	Dokumentation einiger Erweiterungen

1 Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden. Ausgangssituation und Zielsetzung

Mit PDF-AS steht eine Applikation zur Verfügung, mit welcher beliebige PDF Dokumente elektronisch signiert, bzw. bereits vorhandene Signaturen auf Dokumenten verifiziert werden können. PDF-AS kann dabei sowohl als Kommandozeilenprogramm als auch als Webapplikation sowie - allerdings mit Einschränkungen - als API verwendet werden. Die Signatur selbst kann über eine Vielzahl an Parametern den persönlichen Bedürfnissen angepasst werden. So kann beispielsweise neben einer Unterscheidung zwischen binärer und textueller Signatur unter anderem auch das Erscheinungsbild des Signaturblocks oder dessen Position im PDF Dokument angegeben werden.
Ziel war es nun, diese Funktionalität auch anderen Webapplikationen über eine definierte Schnittstelle zugänglich zu machen. Dadurch sollte es diesen Anwendungen ermöglicht werden, beliebige PDF Dokumente zur Verarbeitung an PDF-AS zu übermitteln.

Da die hier beschriebene Schnittstelle aus den Anforderungen eines laufenden Projekts heraus entwickelt wurde, ist diese Implementierung in der derzeitigen Version stark auf die Anforderungen dieses speziellen Projekts zugeschnitten. Dies gilt im Besonderen für die Wahl der an PDF-AS übermittelten Parameter. Eine eventuelle Anpassung der vorhandenen Version, sollte aber ohne großen Aufwand möglich sein.

2 Beschreibung der Schnittstelle

Die Grundstruktur der implementierten Schnittstelle ist in Abbildung 1 dargestellt.

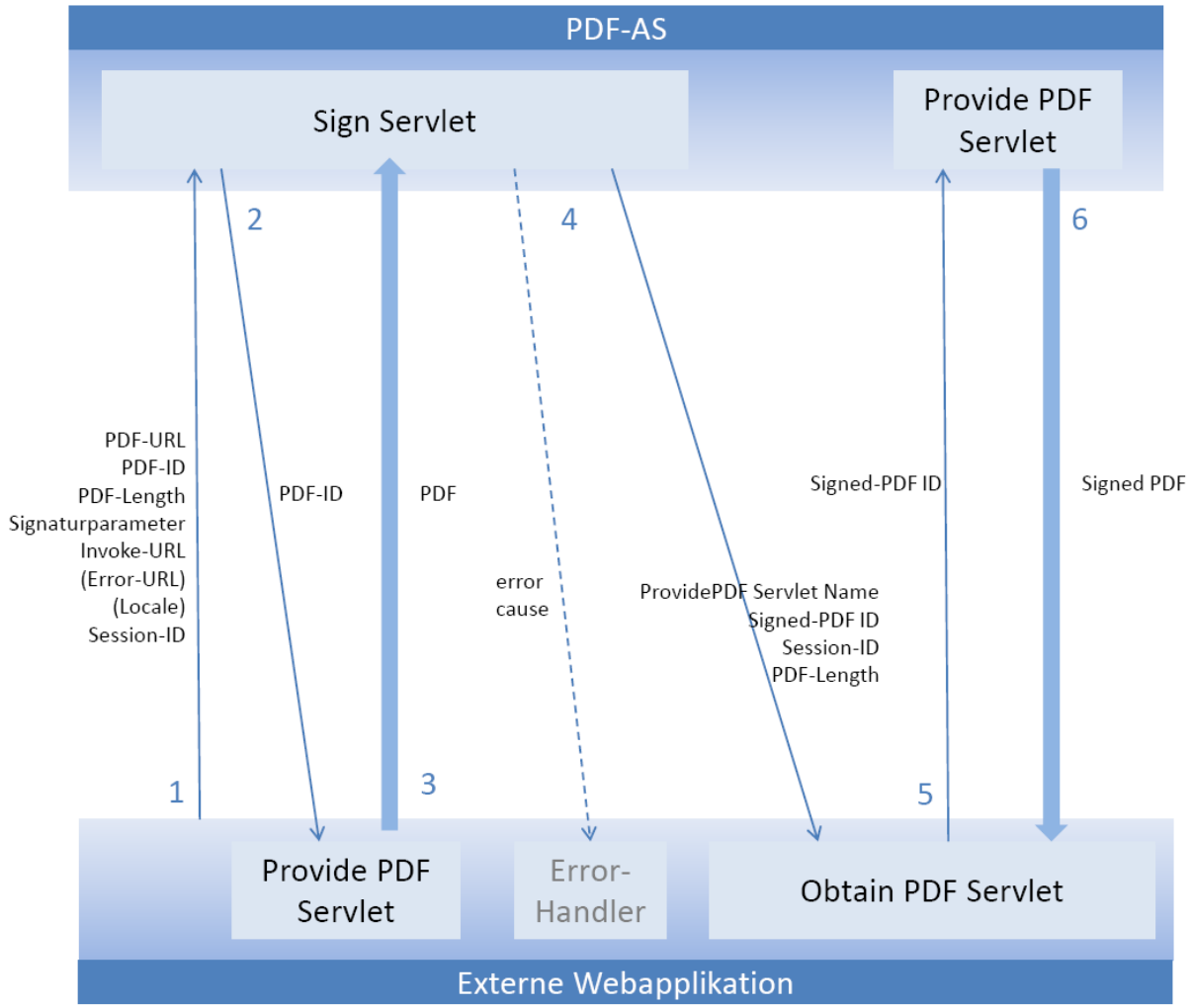

Abbildung 1 - Schnittstelle zwischen PDF-AS und externer Webanwendung

Die einzelnen Schritte der Kommunikation zwischen PDF-AS und der externen Webapplikation sind von 1 bis 6 durchnummeriert und geben die Abfolge des Informationsaustausches wieder. Im Folgenden werden die einzelnen Punkte kurz beschrieben.

1. Die externe Webanwendung, welche die Funktionalität von PDF-AS zur Signierung eines PDF Dokuments nutzen möchte, ruft das Sign-Servlet von PDF-AS über einen Redirect auf. Zu diesem Zeitpunkt muss sichergestellt sein, dass die zu signierende PDF Datei von der externen Webapplikation über eine URL zur Verfügung gestellt wird. Diese URL wird PDF-AS über einen Redirect-Parameter mitgeteilt. Optional ist die Angabe einer URL mit der PDF-AS Fehler an die aufrufende Applikation melden kann. Weitere Parameter definieren unter anderem die ID des PDF Dokuments, mit der dieses eindeutig identifiziert werden kann, diverse Signaturparameter, die von PDF-AS zur Erstellung der Signatur benötigt werden sowie auch optional die Angabe der zu nutzenden (Browser-)Sprache. Um nach erfolgter Signierung des Dokuments

die aufrufende Webanwendung wieder fortsetzen zu können, wird auch die Session-ID der externen Webapplikation als Parameter über den Redirect an PDF-AS übermittelt.

2. PDF-AS holt das zu signierende Dokument unter Nutzung der im ersten Schritt erhaltenen ID unter der angegebenen URL ab.

3. Das entsprechende zu signierende PDF Dokument wird an PDF-AS übermittelt.

4. PDF-AS signiert das Dokument. Der Anwender wird über einen Redirect zur externen Web-Applikation zurück geleitet (Invoke-URL). Dazu wird der Name des Servlets, unter dem das signierte PDF Dokument abgeholt werden kann, sowie die ID des Dokuments übermittelt. Da die vollständige URL, unter der das PDF Dokument von PDF-AS zur Verfügung gestellt wird, von der Umgebung abhängt, in der die PDF-AS Webapplikation läuft, wird nur der Name des Servlets, welches für das Bereitstellen der Datei verantwortlich ist, übertragen. Der Rest der URL (PDF-AS Home-URL), muss von der externen Applikation ergänzt werden. Die PDF-ID entspricht jener ID, unter der das zu signierende PDF in Punkt 2 und 3 von der externen Webapplikation bereitgestellt wurde. Auch die in Schritt 1 übermittelte Session ID wird an die externe Anwendung übertragen. Wurde eine Fehler-URL beim Aufruf von PDF-AS angegeben, dann findet im Fehlerfall ein Redirect zu dieser Adresse statt wobei der Fehler als URL-Parameter "error" und die Ursache als URL-Parameter "cause" kodiert wird.

5. Unter der in Punkt 4 erhaltenen URL kann die externe Webanwendung nun das signierte PDF Dokument mit der entsprechenden ID von PDF-AS abholen.

6. Das signierte PDF Dokument wird an die externe Webanwendung übermittelt.

Die derzeitige Implementierung unterstützt ausschließlich die Signatur von PDF Dokumenten. Nichtsdestotrotz könnte das Prinzip dieser Schnittstelle auch zur Übermittlung von Dokumenten zur Verifikation vorhandener Signaturen angewandt werden.

3 Beschreibung der Implementierung

Im Folgenden wird die vorhandene Implementierung der in Abschnitt 2 definierten Schnittstelle beschrieben. Wie bereits erwähnt, ist diese Implementierung auf die Anforderungen eines speziellen Projekts zugeschnitten. Nichtsdestotrotz wird dadurch die Funktionalität der Schnittstelle illustriert.

3.1 Erweiterungen von PDF-AS

Vor der Erweiterung um die im vorigen Abschnitt beschriebene Schnittstelle konnten Daten über zwei Wege an PDF-AS übermittelt werden.

1. Über die Angabe von Parametern beim Aufruf der Applikation über die Kommandozeile

2. Über ein Webformular beim Betrieb von PDF-AS als Webapplikation

Die Implementierung der beschrieben Schnittstelle nützt die vorhandene Funktionalität der PDF-AS Webanwendung und erweitert diese, um eine Kommunikation mit externen Webapplikationen zu ermöglichen. Im Speziellen wurden die im Folgenden beschriebenen Klassen angepasst bzw. hinzugefügt.

3.1.1 Class SignServlet

Diese Klasse wurde ursprünglich ausschließlich vom Webformular, über das der Benutzer mit der Webapplikation PDF-AS kommuniziert, aufgerufen. In dieser Klasse wurden die vom Benutzer definierten Parameter gesammelt, aufbereitet und an die entsprechenden Klassen, welche die eigentliche Signatur des PDF Dokuments durchführen, weitergeleitet.
Da diese Klasse auch von einer externen Webanwendung über einen Redirect aufgerufen werden können soll, wurde sie dahingehend erweitert, als dass nun überprüft wird ob die URL einer eventuell vorhandenen externen Webanwendung als Parameter vorliegt. Ist dies der Fall, werden die für die Signatur benötigten Parameter über die URL anstelle des Webformulars bezogen.
Da diese Klasse auch für die weitere Verarbeitung des signierten Dokuments verantwortlich ist, wurde auch dieser Teil des Servlets angepasst. Bei Vorhandensein einer externen URL, ruft das Servlet über einen Redirect die externe Webapplikation auf, woraufhin das resultierende PDF Dokument schließlich an die externe Webapplikation übermittelt wird.

In der folgenden Tabelle sind die für einen erfolgreichen Aufruf des Servlets nötigen Parameter zusammengefasst. Einige Parameter sind für die Signierung des PDF Dokuments irrelevant, wurden jedoch in die Schnittstelle aufgenommen, um spätere Erweiterungen zu vereinfachen und nötige Adaptionen in bereits bestehenden Klassen von PDF-AS gering zu halten.

Tabelle 1 - Benötigte Parameter

Parameter Name	Werte	Beschreibung / Anmerkungen
connector	<moa|bku>	Für die hier beschriebene Implementierung wurde bislang nur der Wert "bku" getestet
filename	<FILENAME>	Ist für den Signierungsvorgang ebenfalls irrelevant, kann jedoch bei Debug-Szenarien hilfreich sein

Parameter Name	Werte	Beschreibung / Anmerkungen
inline	<true|false>	Für die Signierung des Dokuments irrelevant. Wird der Vollständigkeit halber angegeben und sollte standardmäßig auf "false" gesetzt werden
invoke-app-url	<URL>	Gibt die URL zum Servlet an, unter dem die externe Webapplikation fortgesetzt werden kann
invoke-app-error-url	<URL>	OPTIONAL: Definiert eine Fehler-behandlungs-URL der aufrufenden Fach-applikation an die der Anwender im Fehlerfall weitergeleitet wird. Der Fehler wird hierbei URL-encoded im Parameter "error" und die Ursache, ebenfalls URL-encoded, im Parameter "cause" übergeben.
locale	<Locale>	OPTIONAL: Definiert die Sprache, die PDF-AS verwenden soll. Die Angabe der Sprache erfolgt durch Deklaration eines zweistelligen ISO-639 konformen Sprachcodes zusammen mit einem optionalen ISO-3166 konformen Ländercode. (Beispiele: "de", "de_AT", "en", "en_US")
mode	<textual|binary>	Gibt den Typ der Signatur an
num-bytes	<Integer>	Gibt die Anzahl der zu übertragenden Bytes an
pdf-id	<Long>	Eindeutige ID, mit der zu übermittelndes PDF Dokument identifiziert werden kann
pdf-url	<URL>	Gibt die URL zum Servlet an, unter dem PDF-AS das zu signierende Dokument beziehen kann
preview	false	Dieser Parameter muss auf "false" gesetzt sein um eine korrekte Übermittlung der Daten zu gewährleisten
session-id	<SESSION-ID>	Die Session-ID der externen Webanwendung wird übergeben um ein späteres Fortsetzen der Applikation zu gewährleisten
sig_type	<PROFILENAME>	Gibt den Namen des zu verwendenden PDF-AS Signaturprofils an
sig-pos-p	<Integer>	Gibt die Seite des PDF Dokuments an, auf dem der Signaturblock platziert werden soll
sig-pos-y	<Integer>	Gibt die vertikale Position des zu erstellenden Signaturblocks im PDF Dokument an

Das folgende Beispiel zeigt die Anwendung der oben beschriebenen Parameter:

```
https://server.foo/pdf-as/Sign
   ?preview=false
   &connector=bku
   &mode=textual
   &sig_type=SIGNATURBLOCK_DE
   &inline=false
   &filename=test.pdf
   &num-bytes=45916
   &pdf-url=http://localhost:8080/myapp/ProvidePDF
   &pdf-id=1956507909008215134
   &invoke-app-url=https://server.foo/myapp/ReturnSignedPDF
   &invoke-app-error-url=https://server.foo/myapp/ErrorHandler
   &session-id=9085B85B364BEC31E7D38047FE54577D
   &locale=de
```

Zu beachten ist, dass sämtliche Parameter URL-kodiert werden müssen.

```
https://server.foo/pdf-as/Sign
   ?preview=false
   &connector=bku
   &mode=textual
   &sig_type=SIGNATURBLOCK_DE
   &inline=false
   &filename=test.pdf
   &num-bytes=45916
   &pdf-url=http%3A%2F%2Flocalhost%3A8080%2Fmyapp%2FProvidePDF
   &pdf-id=1956507909008215134
   &invoke-app-url=https%3A%2F%2Fserver.foo%2Fmyapp%2FReturnSignedPDF
   &invoke-app-error-url=https%3A%2F%2Fserver.foo%2Fmyapp%2FErrorHandler
   &session-id=9085B85B364BEC31E7D38047FE54577D
   &locale=de
```

3.1.2 Class SessionInformation

Diese Klasse kapselt sämtliche Informationen, die vom Benutzer an die Webanwendung PDF-AS übermittelt werden. Da bei einem Aufruf von PDF-AS durch eine externe Webapplikation zusätzliche Daten anfallen, wurde diese Klasse um zwei Einträge erweitert.

1. In einem TablePos-Objekt werden Informationen über die Position der zu erstellenden Signatur gespeichert. In der aktuellen Version kann die Seite des PDF Dokuments auf dem sich die Signatur befinden soll, sowie deren vertikale Position angegeben werden.
2. In einem ExternAppInformation-Objekt werden Daten über die aufrufende Webapplikation gespeichert. Eine Beschreibung dieser Klasse ist in Abschnitt 3.1.3 gegeben.

3.1.3 Class ExternAppInformation

Diese Klasse wurde der Applikation hinzugefügt und kapselt Informationen über die aufrufende Webanwendung. So werden die Session-ID, Invoke-URL, Fehlerbehandlungs-URL und die PDF-ID des zu signierenden Dokuments gespeichert.

3.1.4 Class ProvidePDFServlet

Die Klasse ProvidePDFServlet wurde dem Projekt ebenfalls hinzugefügt. Wird die Klasse aufgerufen, sucht sie in allen evident gehaltenen, signierten Dokumenten nach dem PDF mit der entsprechenden ID, die der Klasse als Parameter übergeben wird. Nach erfolgreicher Übermittlung des PDF Dokuments wird dieses zusammen mit seiner ID gelöscht.

In der folgenden Tabelle ist der für einen erfolgreichen Aufruf des Servlets nötige Parameter beschrieben.

Tabelle 2 - Benötigter Parameter

Parameter Name	Werte	Beschreibung / Anmerkungen
pdf-id	<Long>	Eindeutige ID, mit der PDF Dokument identifiziert und von der externen Webanwendung abgeholt werden kann

3.1.5 Class PDFContainer

Die Klasse PDFContainer enthält ein signiertes PDF und die dazugehörige ID. Sie wird verwendet um alle bereits signierten PDF Dokumente einer ID zuzuordnen und diese evident zu halten bis sie von der aufrufenden Applikation abgeholt werden.

3.2 Anbindung der externen Web-Anwendung an PDF-AS

Um eine Kommunikation mit PDF-AS zu ermöglichen, muss die externe Webapplikation zwei Servlets zur Verfügung stellen, welche von PDF-AS aufgerufen werden. Da die URLs zu diesen Servlets ohnehin als Parameter an PDF-AS übermittelt werden, sind die hier angegebenen Namen der Servlets im Gegensatz zu den Namen der Parameter keinesfalls zwingend.

3.2.1 ProvidePDFServlet

Ähnlich dem gleichnamigen Servlet in PDF-AS, stellt diese Klasse eine Methode zur Verfügung mit der ein PDF Dokument mit einer bestimmten ID abgeholt werden kann. Für einen erfolgereichen Dateitransfer muss die URL zu diesem Servlet zusammen mit der entsprechenden ID PDF-AS als Parameter übergeben werden. Das File selbst wird über den OutputStream des Response-Parameters des Servlets gestreamt und so an PDF-AS übertragen.

In der folgenden Tabelle ist der für einen erfolgreichen Aufruf des Servlets nötige Parameter beschrieben.

Tabelle 3 - Benötigter Parameter

Parameter Name	Werte	Beschreibung / Anmerkungen
pdf-id	<Long>	Eindeutige ID, mit der PDF Dokument identifiziert und von PDF-AS abgeholt werden kann

3.2.2 ObtainPDFServlet

Über dieses Servlet wird die externe Webapplikation mit Hilfe der in Schritt 1 übermittelten Session-ID von PDF-AS wieder fortgesetzt. Aus dem übermittelten Namen des Servlets, unter dem PDF-AS das signierte Dokument zur Verfügung stellt, muss zunächst durch Voranstellen des PDF-AS Home-URL eine vollständige URL erstellt werden. Über diese URL und der ebenfalls übermittelten ID kann das Servlet schließlich das signierte PDF Dokument von PDF-AS abholen. Das entsprechende PDF-AS Servlet stellt die Daten dabei über den OutputStream des Response-Parameters des Servlets zur Verfügung. Das PDF Dokument kann einfach über einen InputStream von der übermittelten URL eingelesen werden.
In der folgenden Tabelle sind die für einen erfolgreichen Aufruf des Servlets nötigen Parameter zusammengefasst.

Tabelle 1 - Benötigte Parameter

Parameter Name	Werte	Beschreibung / Anmerkungen
num-bytes	<Integer>	Gibt die Anzahl der zu übertragenden Bytes an
pdf-id	<Long>	Eindeutige ID, mit der zu übermittelndes PDF Dokument identifiziert werden kann
pdf-url	<URL>	Gibt den Namen des Servlets an, unter dem die externe Webapplikation das zu signierende Dokument beziehen kann
session-id	<SESSION-ID>	Wird als Parameter übergeben um die ursprüngliche Session fortzusetzen