



A-Trust Gesellschaft für Sicherheitssysteme
im elektronischen Datenverkehr GmbH Landstraßer
Hauptstraße 5, A-1030 Wien
Tel: +43 (1) 713 21 51 - 0
Fax: +43 (1) 713 21 51 - 350
<https://www.a-trust.at>

A-Trust SignaturServer

Version: 1.2.2
Datum: 24.08.2015



Inhaltsverzeichnis

1	Signaturserver Kurzbeschreibung	2
1.1	Einzel-signatur	2
1.2	Stapel-signatur	2
1.3	P12 Signaturen	2
1.4	Verwaltung Templates	3
1.5	Allgemein	3
2	Ablauf der Handysignatur	4
3	Signatursiegel	6
3.1	Description	6
3.2	Position	6
3.3	B64bglImage	6
3.4	Optionale Elemente	6
3.4.1	Zusätzliche Bilder	6
3.4.2	Zusätzliche Text	8
4	Namespace Documentation	9
4.1	Package HandySignaturPdfSoap	9
4.2	Package LibHandySignaturPdf	9
4.2.1	Enumeration Type Documentation	9
5	Class Documentation	10
5.1	HandySignaturPdfSoap.IHandySignaturPdf Interface Reference	10
5.1.1	Detailed Description	11
5.1.2	Member Function Documentation	11

1 Signaturserver Kurzbeschreibung

1.1 Einzelsignatur

Eine Einzelsignatur wird mit einer der folgenden Methoden angestoßen:

- StartSignature (Ohne Signatursiegel)
- StartSignatureTemplate (Mit Signatursiegel)
- StartSignatureTemplateEx (Mit Signatursiegel und Positionsinformationen)

Nachdem die Handysignatur erfolgreich durchgeführt wurde, kann mit

- EndSignature

das signierte Dokument abgeholt werden.

1.2 Stapelsignatur

Eine Stapelsignatur wird mit dem folgenden Aufruf initiiert:

- StartBatchSignature

Anschließend können mit den folgenden Aufrufen Dokumente zu dem Signaturstapel hinzugefügt werden:

- AddDocument(Ohne Signatursiegel)
- AddDocumentTemplate(Mit Signatursiegel)
- AddDocumentTemplateEx(Mit Signatursiegel und Positionsinformationen)

Anschließend kann der Stapel mit der folgenden Methode geschlossen werden:

- EndBatchSignature

Nachdem die Handysignatur erfolgreich durchgeführt wurde, kann mit

- GetDocument

das signierte Dokument abgeholt werden.

1.3 P12 Signaturen

Eine Signatur wird mit einer der folgenden Methoden angestoßen:

- DoP12Signature (Ohne Signatursiegel)
- DoP12SignatureTemplate (Mit Signatursiegel)
- DoP12SignatureEx (Mit Signatursiegel und Positionsinformationen)

Wenn die Signatur erfolgreich war, wird das fertig signierte Dokument direkt mitgeliefert.

1.4 Verwaltung Templates

Die folgenden Funktionen dienen dazu die Templates auf dem Signaturserver zu verwalten:

- UploadTemplate: Läd ein Template auf den Signaturserver und liefert die ID zurück, mit der das Template verwendet werden kann
- ReplaceTemplate: Ersetzt das Template mit der übergebenen ID
- GetTemplate: Liefert das Template mit der übergebenen ID zurück
- DeleteTemplate: Liefert das Template mit der übergebenen ID zurück
- ListTemplates: Listet alle verfügbaren Templates auf

1.5 Allgemein

Nachdem eine Signatur erfolgreich durchgeführt wurde, kann mit

- GetCertificate

das Zertifikat des Signators geholt werden.

2 Ablauf der Handysignatur

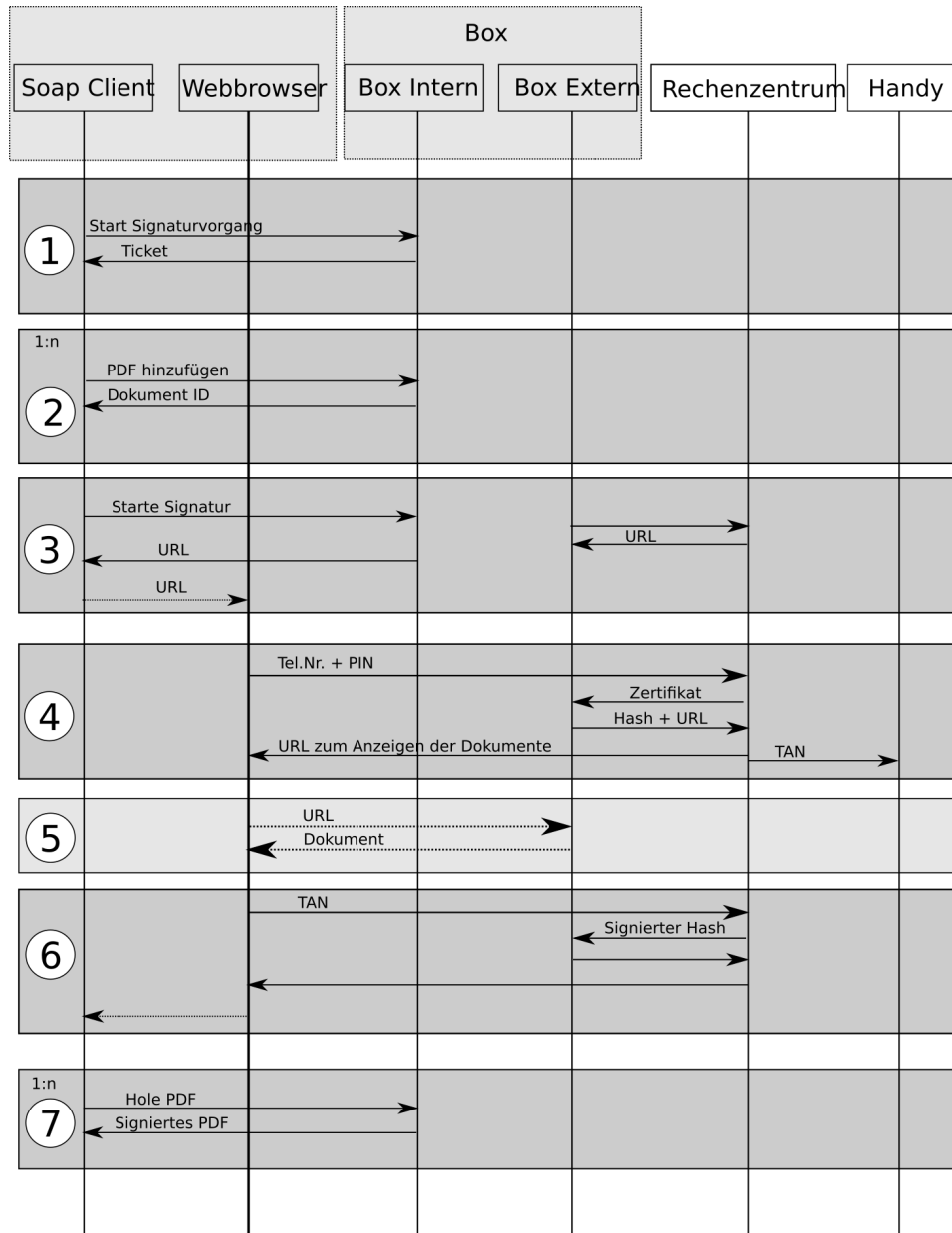


Abbildung 1: Sequenzdiagramm der Handysignatur

Die Signaturbox kann verwendet werden um einen Dokumentenstapel (mehrere PDF Dokumente) mit einer einzigen Handysignatur zu signieren. Bild 1 zeigt die Schritte auf die dafür notwendig sind. Im Detail sind diese:

1. Zuerst muss der Stapel mit der Methode *StartBatchSignature* geöffnet werden. Diese Methode liefert ein Ticket zurück welches bei allen weiteren Methodenaufrufen angegeben werden muss.
2. Nachdem der Stapel geöffnet ist, lassen sich Dokumente zu dem Stapel hinzufügen. Dies geschieht mit den Methoden *AddDocument* (nicht sichtbare Signatur), *AddDocumentTemplate* (sichtbare Signatur) oder *AddDocumentTemplateEx* (sichtbare Signatur mit Positionsangabe). In einem Stapel können auch nicht sichtbare und sichtbare Signaturen gemeinsam

durchgeführt werden. Als Ergebnis des Aufrufs bekommt man eine ID, unter der das Dokument abgeholt werden kann.

3. Anschließend muss der Signaturstapel mit dem Aufruf *EndBatchSignature* geschlossen werden. Durch den Aufruf wird der Handysignaturprozess angestoßen. Man bekommt eine URL die verwendet werden muss um die eigentliche Handysignatur durchzuführen.
4. Mit der erhaltenen URL wird der erste Teil der Handysignatur durchgeführt. Hierzu muss die Telefonnummer und das Passwort eingegeben werden. Wenn dies erfolgreich geschehen ist, wird eine TAN an die Handynummer gesendet. Außerdem wird eine URL zurück gegeben. Diese kann verwendet werden um eine Vorschau der zu signierenden Dokumente anzuzeigen.
5. Die erhaltene URL kann verwendet werden um eine Voransicht der zu signierenden Dokumente anzuzeigen. Dieser Schritt ist optional, d.h. die Handysignatur kann auch durchgeführt werden, ohne eine Vorschau anzuzeigen.
6. Als zweiten Schritt der Handysignatur muss die in Schritt 4 erhaltene TAN eingegeben werden. Dadurch wird die Handysignatur abgeschlossen.
7. Nach einer erfolgreichen Handysignatur können die signierten Dokument von der Signaturbox abgeholt werden. Dies geschieht mit der Methode *GetDocument*. Die jeweiligen Dokumente des Stapels werden einzeln mit ihrer ID von der Signaturbox geladen.

3 Signatursiegel

Signatursiegel werden als XML Dateien definiert, z.B.:

```
<?xml version="1.0" encoding="UTF-8"?>
<SignatureTemplateStyle>
<Description author="A-Trust GmbH" name="Sisi" description="A-Trust Standard Signatur
  Siegel" />
<Position type="relative" x="40.0" y="40.0" w="50.0" h="50.0" page="1" />
<B64bgImage>
iVBORw0KGgoAAAANSUUhEUgAAG [...] jhz/DwF8ylxAZ19
</B64bgImage>
<Fields>
<Image x="150.0" y="0.0" w="800.0" h="200.0">
/9j/4AAQSkZJRgAB [...] ABRRRRQAUUUUJAAFFFAH/9k=
</Image>
</Fields>
</SignatureTemplateStyle>
```

Die einzelnen Elemente werden im Folgenden beschrieben.

3.1 Description

Hier werden Metainformationen wie Autor, Name und eine Beschreibung gespeichert. Der Name wird von *ListTemplate* verwendet.

3.2 Position

Hier wird die Position des Signatursiegels im PDF festgelegt. Die Werte x und y legen eine Ecke des Signatursiegels fest. Hierbei ist zu beachten das sich der Koordinaten Ursprung in der linken unteren Ecke der PDF Datei befindet. Mit w und h wird die zweite Ecke des Siegels definiert. Die Werte werden in Prozent (der Seitenhöhe / Seitenbreite) angegeben. Zusätzlich ist die Seite auf der das Signatursiegel platziert werden soll festzulegen. Anstatt „relative“ kann die Position des Signatursiegels auch „absolute“ definiert werden.

3.3 B64bgImage

Hier wird das eigentliche Hintergrundbild als Base 64 gespeichert. Zulässige Dateitypen für Hintergrundbilder sind .png und .jpg.

3.4 Optionale Elemente

3.4.1 Zusätzliche Bilder

Zusätzlich zum Hintergrundbild können noch andere Bilder hinzugefügt werden. Diese werden innerhalb des <Fields> Tags definiert:

```
<Image x="0" y="0" w="150" h="150">Base64Content</Image>
```

Das Bild selbst wird in Base64 codiert in das XML Dokument geschrieben. Hierbei ist zu beachten, dass als zulässige Bildformate nur JPEG und PNG unterstützt werden. Die Werte x und y geben die Koordinaten des linken oberen Punktes des Bildes im Hintergrundbild an. Die Werte w und h bestimmen die Ausdehnungen das Bildes. Weichen sie von der Höhe oder Breite des Bildes selbst ab, wird es gestreckt bzw. gestaucht. Alle Werte (x, y, w, h) haben Pixel als Einheit.

Beispiele Für die folgenden Beispiele wird jeweils ein Hintergrundbild mit der Auflösung 800x300 und ein zusätzliches Bild mit der Auflösung 100x100 verwendet.

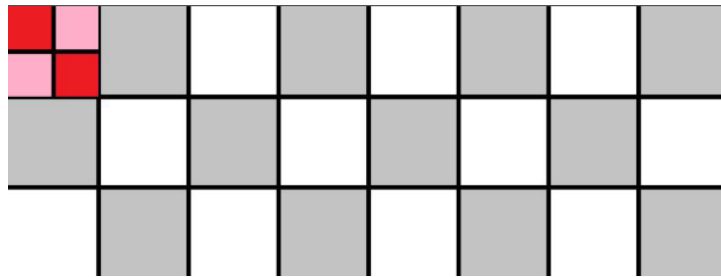


Abbildung 2: Bild in Originalgröße (X=0, Y=0, W=100, H=100)

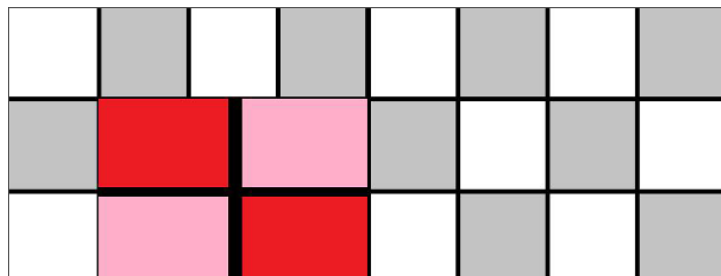


Abbildung 3: Bild gestreckt (X=100, Y=100, W=300, H=200), da Originalbild 100x100 groß

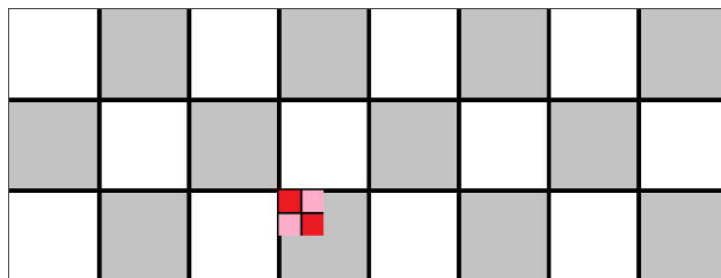


Abbildung 4: Bild gestaucht (X=300, Y=200, W=50, H=50), da Originalbild 100x100 groß

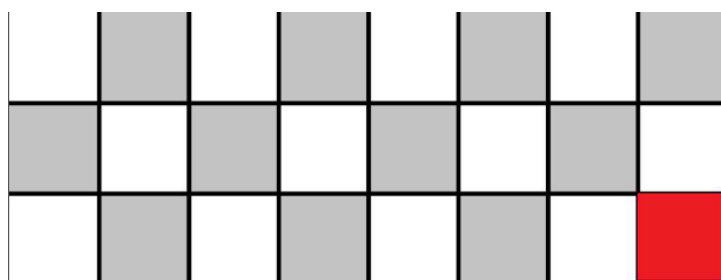


Abbildung 5: Bild außerhalb* (X=700, Y=200, W=200, H=200) Hier wird nur der linke obere Teil des Bildes angezeigt, der Rest ist außerhalb des sichtbaren Bereiches.

3.4.2 Zusätzliche Text

Zusätzlich kann auch Text definiert werden, der auf das Signatursiegel geschrieben wird. Es gibt verschiedene Typen von Text: „static“ und „dynamic“. Die folgenden Attribute sind notwendig:

- type - static, dynamic (signer, location, reason, timestamp können auch verwendet werden wenn bei einem der dynamischen Elemente kein zusätzlicher Text verwendet werden soll)
- x - X Position auf dem Hintergrundbild (Koordinatenursprung ist links oben)
- y - Y Position auf dem Hintergrundbild (Koordinatenursprung ist links oben)
- w - Weite des Textfeldes
- h - obsolete
- center - Wenn der Text kleiner wie das Textfeld ist kann er mit true zentriert werden
- wrap - obsolete
- fgColor - Als RGB Werte, also z.B. "(255, 255, 255)"
- bgColor - Als RGB Werte, also z.B. "(255, 255, 255)"

Erklärung type

- static - Unveränderlicher Text
- dynamic - Hier können veränderliche Informationen auf das Signatursiegel geschrieben werden. Aktuell werden folgende Informationen unterstützt:
 - %timestamp – Der Zeitpunkt an dem die Signatur durchgeführt wurde (im Format TT.MM.JJJJ HH:MM MEZ)
 - %signer – Der Signator der die Signatur ausgeführt hat (common name aus dem Zertifikat der Signatur)
 - %location – Der Ort an dem die Signatur durchgeführt wurde (aus Übergabewert)
 - %reason – Der Grund für die Signatur (aus Übergabewert)
- signer - Kurzschreibweise nur Signator
- location - Kurzschreibweise nur Ort
- reason - Kurzschreibweise nur Grund
- timestamp - Kurzschreibweise nur Zeitstempel

Beispiele Ein Textfeld mit Text und dem Signator:

```
<Label type="dynamic" x="50.0" y="50.0" w="0.0" h="0.0" center="false" wrap="false" fgColor="(0,0,0)" bgColor="(255, 255, 255)">Signiert von: %signer</Label>
```

Ein Textfeld nur mit dem Signator:

```
<Label type="signer" x="50.0" y="50.0" w="0.0" h="0.0" center="false" wrap="false" fgColor="(0,0,0)" bgColor="(255, 255, 255)"/>
```

4 Namespace Documentation

4.1 Package HandySignaturPdfSoap

Classes

- interface [IHandySignaturPdf](#)

Das Interface um auf einer Signatur-Box mit Hilfe der Handy-Signatur zu signieren.

4.2 Package LibHandySignaturPdf

Enumerations

- enum [eErrorCodes](#) {
[eErrorCodes.Error_Success](#) = 0, [eErrorCodes.Error_Failure](#) = -1, [eErrorCodes.Error_Database](#) = -1000, [eErrorCodes.Error_FileToBig](#) = -1001,
[eErrorCodes.Error_ConnectHandySignature](#) = -1002, [eErrorCodes.Error_InvalidParameter](#) = -1003, [eErrorCodes.Error_InvalidTicket](#) = -1004, [eErrorCodes.Error_Timeout](#) = -1005,
[eErrorCodes.Error_InvalidStatus](#) = -1006, [eErrorCodes.Error_InvalidResponseFromHandySig](#) = -1007, [eErrorCodes.Error_PreparePdf](#) = -1008, [eErrorCodes.Error_InvalidXml](#) = -1009,
[eErrorCodes.Error_TemplateNotFound](#) = -1010, [eErrorCodes.Error_AccessDenied](#) = -1011,
[eErrorCodes.Error_InvalidStylesheet](#) = -1012 }

Fehlercodes die bei Aufrufen der Signatur-Box zurück gegeben werden.

4.2.1 Enumeration Type Documentation

enum LibHandySignaturPdf.eErrorCodes Fehlercodes die bei Aufrufen der Signatur-Box zurück gegeben werden.

Enumerator

Error_Success Das aufgerufene Service war erfolgreich.

Error_Failure Ein unbekannter Fehler ist aufgetreten.

Error_Database Es konnte keine Verbindung zur Datenbank hergestellt werden oder die Datenbank liefert Fehler.

Error_FileToBig Das zu signierende bzw. signierte Dokument ist zu groß.

Error_ConnectHandySignature Der SignaturServer konnte keine SSL Verbindung mit dem Handysignatursystemen der A-Trust aufnehmen.

Error_InvalidParameter Einer der angegebenen Parameter enthält einen ungültigen Wert.

Error_InvalidTicket Das angegebene Ticket ist nicht vorhanden oder bereits geschlossen.

Error_Timeout Der betreffende Signaturauftrag ist ausgelaufen.

Error_InvalidStatus Der betreffende Signaturauftrag befindet sich in einem für diese Operation ungültigen Status.

Error_InvalidResponseFromHandySig Die A-Trust Handysignatur hat einen Fehler geliefert.

Error_PreparePdf Das hochgeladene Dokument konnte nicht verarbeitet werden.

Error_InvalidXml Das hochgeladene Template XML konnte nicht verifiziert werden.

Error_TemplateNotFound Die angegebene TemplateId existiert nicht.

Error_AccessDenied Die Funktion darf auf dem System bzw. vom Aufrufer nicht verwendet werden.

Error_InvalidStylesheet Das ausgewählte Template ist nicht valide.

5 Class Documentation

5.1 HandySignaturPdfSoap.IHandySignaturPdf Interface Reference

Das Interface um auf einer Signatur-Box mit Hilfe der Handy-Signatur zu signieren.

Public Member Functions

- int [StartSignature](#) (string APIKey, byte[] DocumentData, string DocumentName, string RedirectURL, string ErrorURL, string HandySignaturParameter, string Location, string Reason, out string Ticket, out string HandySignaturUrl)
Startet eine Einzelsignatur ohne Signaturtemplate.
- int [StartSignatureTemplate](#) (string APIKey, byte[] DocumentData, string DocumentName, string RedirectURL, string ErrorURL, string HandySignaturParameter, int SigTemplateld, string Location, string Reason, out string Ticket, out string HandySignaturUrl)
Startet eine Einzelsignatur mit einem bestehenden Signaturtemplate für sichtbare Signaturen.
- int [StartSignatureTemplateEx](#) (string APIKey, byte[] DocumentData, string DocumentName, string RedirectURL, string ErrorURL, string HandySignaturParameter, int SigTemplateld, string Location, string Reason, double X0, double Y0, double X1, double Y1, uint Page, out string Ticket, out string HandySignaturUrl)
Startet eine Einzelsignatur mit einem bestehenden Signaturtemplate und zusätzlichen Positionsinformationen.
- int [EndSignature](#) (string APIKey, string Ticket, out string DocumentName, out byte[] DocumentData, out string CertificateB64)
Beendet die Einzelsignatur mit dem angegebenen Ticket und gibt das fertig signierte Dokument zurück.
- int [StartBatchSignature](#) (string APIKey, string RedirectURL, string ErrorURL, out string Ticket)
Startet einen Signaturstapel.
- int [EndBatchSignature](#) (string APIKey, string Ticket, string HandySignaturParameter, out string HandySignaturUrl)
Schließt den Signaturstapel mit dem angegebenen Ticket.
- int [AddDocument](#) (string APIKey, string Ticket, byte[] DocumentData, string DocumentName, string Location, string Reason, out int DocumentId)
Fügt der Stapelsignatur mit dem angegebenen Ticket ein Dokument hinzu, welches nicht sichtbar signiert wird.
- int [AddDocumentTemplate](#) (string APIKey, string Ticket, byte[] DocumentData, string DocumentName, int Templateld, string Location, string Reason, out int DocumentId)
Fügt der Stapelsignatur mit dem angegebenen Ticket ein Dokument hinzu, welches mit einem bestehenden Signaturtemplate sichtbar signiert wird.
- int [AddDocumentTemplateEx](#) (string APIKey, string Ticket, byte[] DocumentData, string DocumentName, int Templateld, string Location, string Reason, double X0, double Y0, double X1, double Y1, uint Page, out int DocumentId)
Fügt der Stapelsignatur mit dem angegebenen Ticket ein Dokument hinzu, welches mit einem bestehenden Signaturtemplate und zusätzlichen Positionsinformationen sichtbar signiert wird.
- int [GetDocument](#) (string APIKey, string Ticket, int DocumentId, out string DocumentName, out byte[] DocumentData)
Fordert ein Dokument aus einem Signaturstapel an.
- int [GetCertificate](#) (string APIKey, string Ticket, out string CertificateB64)
Fordert das Zertifikat des Signators eines Dokumentes oder einer Stapelsignatur an.

- int [UploadTemplate](#) (string APIKey, byte[] TemplateData, out int SigTemplateId)
Lädt ein Signaturtemplate auf den SignaturServer.
- int [ReplaceTemplate](#) (string APIKey, byte[] TemplateData, int SigTemplateId)
Ersetzt ein Signaturtemplate auf dem SignaturServer.
- int [GetTemplate](#) (string APIKey, int SigTemplateId, out byte[] TemplateData)
Lädt ein Signaturtemplate vom SignaturServer.
- int [DeleteTemplate](#) (string APIKey, int SigTemplateId)
Löscht ein Template vom SignaturServer.
- int [ListTemplate](#) (string APIKey, out string list)
Liefert eine XML-Liste der am SignaturServer installierten Templates.
- int [GetSmsCount](#) (string APIKey, int Year, int Month, out int Count)
Liefert die Anzahl der verschickten SMS im angegebenen Jahr/Monat zurück.
- int [GetSmsRemaining](#) (string APIKey, int Year, out int Count)
Liefert die Anzahl der im aktuellen SMS-Paket verbleibenden SMS.
- int [GetSmsPackage](#) (string APIKey, int Year, out int Count)
Liefert die Anzahl der im Sms-Paket enthaltenen SMS.
- int [DoP12Signature](#) (string APIKey, byte[] DocumentData, string DocumentName, string Location, string Reason, out byte[] SignedDocumentData)
Signiert das Dokument nicht sichtbar mit dem hinterlegten Schlüssel.
- int [DoP12SignatureTemplate](#) (string APIKey, byte[] DocumentData, string DocumentName, int SigTemplateId, string Location, string Reason, out byte[] SignedDocumentData)
Signiert das Dokument sichtbar mit einem bestehenden Signaturtemplate und dem hinterlegten Schlüssel.
- int [DoP12SignatureTemplateEx](#) (string APIKey, byte[] DocumentData, string DocumentName, int SigTemplateId, string Location, string Reason, double X0, double Y0, double X1, double Y1, uint Page, out byte[] SignedDocumentData)
Signiert das Dokument sichtbar mit einem bestehenden Signaturtemplate, zusätzlichen Positionsinformationen und dem hinterlegten Schlüssel.

5.1.1 Detailed Description

Das Interface um auf einer Signatur-Box mit Hilfe der Handy-Signatur zu signieren.

5.1.2 Member Function Documentation

int HandySignaturPdfSoap.IHandySignaturPdf.AddDocument (string APIKey, string Ticket, byte[] DocumentData, string DocumentName, string Location, string Reason, out int DocumentId) Fügt der Stapelsignatur mit dem angegebenen Ticket ein Dokument hinzu, welches nicht sichtbar signiert wird.

Remarks

Location und Reason werden auf Standardwerte gesetzt.

Parameters

<i>APIKey</i>	Der erhaltene APIKey
<i>Ticket</i>	Das Ticket vom Signaturstapel, zu dem das Dokument hinzugefügt werden soll.
<i>Document-Data</i>	Das zu signierende PDF-Dokument in binärer Form.
<i>Document-Name</i>	Der Name des zu signierenden Dokumentes.
<i>Location</i>	Ort der Signatur
<i>Reason</i>	Grund der Signatur
<i>DocumentId</i>	[out] Innerhalb der Stapelsignatur eindeutige ID, mit dem das Dokument nach der Signatur mittels GetDocument abgeholt werden kann.

Returns

Status Code siehe eErrorCodes

int HandySignaturPdfSoap.IHandySignaturPdf.AddDocumentTemplate (string *APIKey*, string *Ticket*, byte[] *DocumentData*, string *DocumentName*, int *Templateld*, string *Location*, string *Reason*, out int *DocumentId*) Fügt der Stapelsignatur mit dem angegebenen Ticket ein Dokument hinzu, welches mit einem bestehenden Signaturtemplate sichtbar signiert wird.

Remarks

Location und Reason werden auf Standardwerte gesetzt.

Parameters

<i>APIKey</i>	Der erhaltene APIKey
<i>Ticket</i>	Das Ticket vom Signaturstapel, zu dem das Dokument hinzugefügt werden soll.
<i>Document-Data</i>	Das zu signierende PDF-Dokument in binärer Form.
<i>Document-Name</i>	Der Name des zu signierenden Dokumentes.
<i>Templateld</i>	ID des zu verwendenden Signatursiegels.
<i>Location</i>	Ort der Signatur
<i>Reason</i>	Grund der Signatur
<i>DocumentId</i>	[out] Innerhalb der Stapelsignatur eindeutige ID, mit dem das Dokument nach der Signatur mittels GetDocument abgeholt werden kann.

Returns

Status Code siehe eErrorCodes

int HandySignaturPdfSoap.IHandySignaturPdf.AddDocumentTemplateEx (string *APIKey*, string *Ticket*, byte[] *DocumentData*, string *DocumentName*, int *Templateld*, string *Location*, string *Reason*, double *X0*, double *Y0*, double *X1*, double *Y1*, uint *Page*, out int *DocumentId*) Fügt der Stapelsignatur mit dem angegebenen Ticket ein Dokument hinzu, welches mit einem bestehenden Signaturtemplate und zusätzlichen Positionsinformationen sichtbar signiert wird.

Parameters

<i>APIKey</i>	Der erhaltene APIKey
<i>Ticket</i>	Das Ticket vom Signaturstapel, zu dem das Dokument hinzugefügt werden soll.
<i>Document-Data</i>	Das zu signierende PDF-Dokument in binärer Form.
<i>Document-Name</i>	Der Name des zu signierenden Dokumentes.
<i>TemplateId</i>	ID des zu verwendenden Signatursiegels.
<i>Location</i>	Ort der Signatur
<i>Reason</i>	Grund der Signatur
<i>X0</i>	Linke untere X Koordinate des Signatursiegels (in user space units)
<i>Y0</i>	Linke untere Y Koordinate des Signatursiegels (in user space units)
<i>X1</i>	Rechte obere X Koordinate des Signatursiegels (in user space units)
<i>Y1</i>	Rechte obere Y Koordinate des Signatursiegels (in user space units)
<i>Page</i>	Die Seite auf der das Signatursiegel angebracht wird.
<i>DocumentId</i>	[out] Innerhalb der Stapelsignatur eindeutige ID, mit dem das Dokument nach der Signatur mittels GetDocument abgeholt werden kann.

Returns

Status Code siehe eErrorCodes

int HandySignaturPdfSoap.IHandySignaturPdf.DeleteTemplate (string APIKey, int SigTemplateId) Löscht ein Template vom SignaturServer.

Parameters

<i>APIKey</i>	Der erhaltene APIKey
<i>SigTemplateId</i>	Die TemplateId des gewünschten Templates.

Returns

Status Code siehe eErrorCodes

int HandySignaturPdfSoap.IHandySignaturPdf.DoP12Signature (string APIKey, byte[] DocumentData, string DocumentName, string Location, string Reason, out byte[] SignedDocumentData) Signiert das Dokument nicht sichtbar mit dem hinterlegten Schlüssel.

Parameters

<i>APIKey</i>	Der erhaltene APIKey
<i>Document-Data</i>	Das zu signierende PDF-Dokument in binärer Form.
<i>Document-Name</i>	Der Name des zu signierenden Dokumentes.
<i>Location</i>	Ort der Signatur
<i>Reason</i>	Grund der Signatur
<i>Signed-Document-Data</i>	Das signierte PDF-Dokument in binärer Form.

Returns

Status Code siehe eErrorCodes

int HandySignaturPdfSoap.IHandySignaturPdf.DoP12SignatureTemplate (string *APIKey*, byte[] *DocumentData*, string *DocumentName*, int *SigTemplateId*, string *Location*, string *Reason*, out byte[] *SignedDocumentData*) Signiert das Dokument sichtbar mit einem bestehenden Signaturtemplate und dem hinterlegten Schlüssel.

Parameters

<i>APIKey</i>	Der erhaltene APIKey
<i>Document-Data</i>	Das zu signierende PDF-Dokument in binärer Form.
<i>Document-Name</i>	Der Name des zu signierenden Dokumentes.
<i>SigTemplateId</i>	ID des zu verwendenden Signaturtemplates.
<i>Location</i>	Ort der Signatur
<i>Reason</i>	Grund der Signatur
<i>Signed-Document-Data</i>	Das signierte PDF-Dokument in binärer Form.

Returns

Status Code siehe eErrorCodes

int HandySignaturPdfSoap.IHandySignaturPdf.DoP12SignatureTemplateEx (string *APIKey*, byte[] *DocumentData*, string *DocumentName*, int *SigTemplateId*, string *Location*, string *Reason*, double *X0*, double *Y0*, double *X1*, double *Y1*, uint *Page*, out byte[] *SignedDocumentData*) Signiert das Dokument sichtbar mit einem bestehenden Signaturtemplate, zusätzlichen Positionsinformationen und dem hinterlegten Schlüssel.

Parameters

<i>APIKey</i>	Der erhaltene APIKey
<i>Document-Data</i>	Das zu signierende PDF-Dokument in binärer Form.
<i>Document-Name</i>	Der Name des zu signierenden Dokumentes.
<i>SigTemplateId</i>	ID des zu verwendenden Signaturtemplates.
<i>Location</i>	Ort der Signatur
<i>Reason</i>	Grund der Signatur
<i>X0</i>	Linke untere X Koordinate des Signatursiegels (in user space units)
<i>Y0</i>	Linke untere Y Koordinate des Signatursiegels (in user space units)
<i>X1</i>	Rechte obere X Koordinate des Signatursiegels (in user space units)
<i>Y1</i>	Rechte obere Y Koordinate des Signatursiegels (in user space units)
<i>Page</i>	Die Seite auf der das Signatursiegel angebracht wird.
<i>Signed-Document-Data</i>	Das signierte PDF-Dokument in binärer Form.

Returns

Status Code siehe eErrorCodes

int HandySignaturPdfSoap.IHandySignaturPdf.EndBatchSignature (string *APIKey*, string *Ticket*, string *HandySignaturParameter*, out string *HandySignaturUrl*) Schließt den Signaturstapel mit dem angegebenen Ticket.

Precondition

Der Aufruf muss innerhalb eines 5 min Timeouts nach dem letzten AddDocument/AddDocument-Ex aufgerufen werden.

Der Aufruf von StartBatchSignature muss SUCCESS geliefert haben.

Parameters

<i>APIKey</i>	Der erhaltene APIKey
<i>Ticket</i>	Das Ticket des zu schließenden Signaturauftrages.
<i>Handy-Signatur-Parameter</i>	Parameter, welche der Handysignaturseite übergeben werden. Diese Parameter müssen als GET-Parameter in der Form ¶m1=value1¶m2=value2... formatiert werden. Mögliche Parameter siehe unter http://labs.-a-trust.at/developer/ShowSource.aspx?id=113 .
<i>Handy-SignaturUrl</i>	[out] Die URL zur Handysignatur Seite auf der der Benutzer PIN/TAN eingeben muss.

Returns

Status Code siehe eErrorCodes

int HandySignaturPdfSoap.IHandySignaturPdf.EndSignature (string APIKey, string Ticket, out string DocumentName, out byte[] DocumentData, out string CertificateB64) Beendet die Einzelsignatur mit dem angegebenen Ticket und gibt das fertig signierte Dokument zurück.

Precondition

Der Aufruf von StartSignature muss SUCCESS geliefert haben.

Postcondition

Nach einem erfolgreichen Aufruf wird der Signaturauftrag als finalisiert markiert. Das Dokument kann ab diesem Zeitpunkt jederzeit gelöscht werden.

Remarks

Wird EndSignature aufgerufen, bevor der Signator die Signatur auf der A-Trust Handysignaturseite durchgeführt hat, wird das unsignierte Dokument zurückgegeben.

Parameters

<i>APIKey</i>	Der erhaltene APIKey
<i>Ticket</i>	Das Ticket des zu finalisierenden Signaturauftrages.
<i>Document-Name</i>	[out] Der Name des signierten Dokumentes. Dieser ist der originale Dokumentenname mit einem "-signed" postfix.
<i>Document-Data</i>	[out] Das signierte Dokument in binärer Form.
<i>CertificateB64</i>	[out] Das Zertifikat, welches zum signieren verwendet wurde. Base64 kodiert.

Returns

Status Code siehe eErrorCodes

int HandySignaturPdfSoap.IHandySignaturPdf.GetCertificate (string APIKey, string Ticket, out string CertificateB64) Fordert das Zertifikat des Signators eines Dokumentes oder einer Stapelsignatur an.

Parameters

<i>APIKey</i>	Der erhaltene APIKey
<i>Ticket</i>	Das Ticket eines Signaturauftrages oder einer Stapelsignatur.
<i>CertificateB64</i>	[out] Das Zertifikat, welches zum signieren verwendet wurde. Base64 kodiert.

Returns

Status Code siehe eErrorCodes

int HandySignaturPdfSoap.IHandySignaturPdf.GetDocument (string *APIKey*, string *Ticket*, int *DocumentId*, out string *DocumentName*, out byte[] *DocumentData*) Fordert ein Dokument aus einem Signaturstapel an.

Precondition

Der Aufruf von StartBachtSignature und AddDocument muss SUCCESS geliefert haben.

Postcondition

Nach einem erfolgreichen Aufruf wird das Dokument als finalisiert markiert. Das Dokument kann ab diesem Zeitpunkt jederzeit gelöscht werden.

Remarks

Wird GetDocument aufgerufen, bevor der Signator die Signatur auf der A-Trust Handysignaturseite durchgeführt hat, wird das unsignierte Dokument zurückgegeben.

Parameters

<i>APIKey</i>	Der erhaltene APIKey
<i>Ticket</i>	Das Ticket des gewünschten Signaturstapels
<i>DocumentId</i>	Die ID des zu finalisierenden Dokumentes.
<i>Document-Name</i>	[out] Der Name des signierten Dokumentes. Dieser ist der originale Dokumentenname mit einem "-signed" postfix.
<i>Document-Data</i>	[out] Das signierte Dokument in binärer Form.

Returns

Status Code siehe eErrorCodes

int HandySignaturPdfSoap.IHandySignaturPdf.GetSmsCount (string *APIKey*, int *Year*, int *Month*, out int *Count*) Liefert die Anzahl der verschickten SMS im angegebenen Jahr/Monat zurück.

Parameters

<i>APIKey</i>	Der erhaltene APIKey
<i>Year</i>	
<i>Month</i>	
<i>Count</i>	[out] Anzahl der verschickten SMS im angegebenen Zeitraum.

Returns

Status Code siehe eErrorCodes

int HandySignaturPdfSoap.IHandySignaturPdf.GetSmsPackage (string *APIKey*, int *Year*, out int *Count*) Liefert die Anzahl der im Sms-Paket enthaltenen SMS.

Parameters

<i>APIKey</i>	Der erhaltene APIKey
<i>Year</i>	
<i>Count</i>	[out] Anzahl der im SMS-Paket enthaltenen SMS.

Returns

Status Code siehe eErrorCodes

int HandySignaturPdfSoap.IHandySignaturPdf.GetSmsRemaining (string *APIKey*, int *Year*, out int *Count*) Liefert die Anzahl der im aktuellen SMS-Paket verbleibenden SMS.

Parameters

<i>APIKey</i>	Der erhaltene APIKey
<i>Year</i>	
<i>Count</i>	[out] Anzahl der im aktuellen SMS-Paket verbleibenden SMS.

Returns

Status Code siehe eErrorCodes

int HandySignaturPdfSoap.IHandySignaturPdf.GetTemplate (string *APIKey*, int *SigTemplateId*, out byte[] *TemplateData*) Lädt ein Signaturtemplate vom SignaturServer.

Parameters

<i>APIKey</i>	Der erhaltene APIKey
<i>SigTemplateId</i>	Die TemplateId des gewünschten Templates.
<i>TemplateData</i>	[out] Das Template-XML.

Returns

Status Code siehe eErrorCodes

int HandySignaturPdfSoap.IHandySignaturPdf.ListTemplate (string *APIKey*, out string *list*) Liefert eine XML-Liste der am SignaturServer installierten Templates.

Parameters

<i>APIKey</i>	Der erhaltene APIKey
<i>list</i>	[out] Der Xml-Response.

Returns

Status Code siehe eErrorCodes

```
* <ArrayOfKeyValueOfintstring xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns="
* http://schemas.microsoft.com/2003/10/Serialization/Arrays">
*   <KeyValueOfintstring>
*     <Key>1</Key>
*     <Value>SiSi</Value>
*   </KeyValueOfintstring>
*   <KeyValueOfintstring>
*     <Key>2</Key>
*     <Value>Description Name</Value>
*   </KeyValueOfintstring>
* </ArrayOfKeyValueOfintstring>
*
```

int HandySignaturPdfSoap.IHandySignaturPdf.ReplaceTemplate (string *APIKey*, byte[] *TemplateData*, int *SigTemplateId*) Ersetzt ein Signaturtemplate auf dem SignaturServer.

Das hochgeladene Dokument wird gegen ein xsd geprüft. Wenn diese Prüfung erfolgreich ist, wird das Template mit der angegebenen ID ersetzt.

Parameters

<i>APIKey</i>	Der erhaltene APIKey
<i>TemplateData</i>	Das Template-XML.
<i>SigTemplateId</i>	Eine eindeutige TemplateId.

Returns

Status Code siehe eErrorCodes

int HandySignaturPdfSoap.IHandySignaturPdf.StartBatchSignature (string *APIKey*, string *RedirectURL*, string *ErrorURL*, out string *Ticket*) Startet einen Signaturstapel.

Postcondition

Nach dem Aufruf von StartBatchSignature ist der Stapel im Status Offen. Es können nun Dokumente mit AddDokument und AddDokumentEx hinzugefügt werden. Nach jedem Aufruf von StartBatchSignature, AddDokument oder AddDokumentEx wird ein Timeout von fünf Minuten gesetzt. Wird dieses Timeout überschritten wird die Stapelsignatur abgebrochen.

Parameters

<i>APIKey</i>	Der erhaltene APIKey
<i>RedirectURL</i>	Die URL, auf die der Signator im Falle einer erfolgreichen Signatur, von der Handysignaturseite weiterleiten werden soll.
<i>ErrorURL</i>	Die URL, auf die der Signator im Falle einer fehlgeschlagenen Signatur, von der Handysignaturseite weiterleiten werden soll.
<i>Ticket</i>	[out] Ticket, welche den Signaturauftrag auf der Box identifiziert.

Returns

Status Code siehe eErrorCodes

int HandySignaturPdfSoap.IHandySignaturPdf.StartSignature (string *APIKey*, byte[] *DocumentData*, string *DocumentName*, string *RedirectURL*, string *ErrorURL*, string *HandySignaturParameter*, string *Location*, string *Reason*, out string *Ticket*, out string *HandySignaturUrl*) Startet eine Einzelsignatur ohne Signaturtemplete.

Parameters

<i>APIKey</i>	Der erhaltene APIKey
<i>DocumentData</i>	Das zu signierende PDF-Dokument in binärer Form.
<i>DocumentName</i>	Der Name des zu signierenden Dokumentes.
<i>RedirectURL</i>	Die URL, auf die der Signator im Falle einer erfolgreichen Signatur, von der Handysignaturseite weiterleiten werden soll.
<i>ErrorURL</i>	Die URL, auf die der Signator im Falle einer fehlgeschlagenen Signatur, von der Handysignaturseite weiterleiten werden soll.
<i>HandySignaturParameter</i>	Parameter, welche der Handysignaturseite übergeben werden. Diese Parameter müssen als GET-Parameter in der Form ¶m1=value1¶m2=value2... formatiert werden. Mögliche Parameter siehe unter http://labs.a-trust.at/developer/ShowSource.aspx?id=113 .
<i>Location</i>	Ort der Signatur.
<i>Reason</i>	Grund der Signatur.
<i>Ticket</i>	[out] Ticket, welche den Signaturauftrag auf der Box identifiziert.
<i>HandySignaturUrl</i>	[out] Die URL zur Handysignatur Seite auf der der Benutzer PIN/TAN eingeben muss.

Returns

Status Code siehe eErrorCodes

int HandySignaturPdfSoap.IHandySignaturPdf.StartSignatureTemplate (string *APIKey*, byte[] *DocumentData*, string *DocumentName*, string *RedirectURL*, string *ErrorURL*, string *HandySignaturParameter*, int *SigTemplateId*, string *Location*, string *Reason*, out string *Ticket*, out string *HandySignaturUrl*) Startet eine Einzelsignatur mit einem bestehenden Signaturtemplete für sichtbare Signaturen.

Parameters

<i>APIKey</i>	Der erhaltene APIKey
<i>DocumentData</i>	Das zu signierende PDF-Dokument in binärer Form.
<i>DocumentName</i>	Der Name des zu signierenden Dokumentes.
<i>RedirectURL</i>	Die URL, auf die der Signator im Falle einer erfolgreichen Signatur, von der Handysignaturseite weiterleiten werden soll.

<i>ErrorURL</i>	Die URL, auf die der Signator im Falle einer fehlgeschlagenen Signatur, von der Handysignaturseite weiterleiten werden soll.
<i>Handy-Signatur-Parameter</i>	Parameter, welche der Handysignaturseite übergeben werden. Diese Parameter müssen als GET-Parameter in der Form ¶m1=value1¶m2=value2... formatiert werden. Mögliche Parameter siehe unter http://labs.a-trust.at/developer/ShowSource.aspx?id=113 .
<i>SigTemplateId</i>	ID des zu verwendenden Signaturtemplates.
<i>Location</i>	Ort der Signatur.
<i>Reason</i>	Grund der Signatur.
<i>Ticket</i>	[out] Ticket, welche den Signaturauftrag auf der Box identifiziert.
<i>Handy-SignaturUrl</i>	[out] Die URL zur Handysignatur Seite auf der der Benutzer PIN/TAN eingeben muss.

Returns

Status Code siehe eErrorCodes

int HandySignaturPdfSoap.IHandySignaturPdf.StartSignatureTemplateEx (string APIKey, byte[] DocumentData, string DocumentName, string RedirectURL, string ErrorURL, string HandySignaturParameter, int SigTemplateId, string Location, string Reason, double X0, double Y0, double X1, double Y1, uint Page, out string Ticket, out string HandySignaturUrl) Startet eine Einzelsignatur mit einem bestehenden Signaturtemplate und zusätzlichen Positionsinformationen.

Parameters

<i>APIKey</i>	Der erhaltene APIKey
<i>Document-Data</i>	Das zu signierende PDF-Dokument in binärer Form.
<i>Document-Name</i>	Der Name des zu signierenden Dokumentes.
<i>RedirectURL</i>	Die URL, auf die der Signator im Falle einer erfolgreichen Signatur, von der Handysignaturseite weiterleiten werden soll.
<i>ErrorURL</i>	Die URL, auf die der Signator im Falle einer fehlgeschlagenen Signatur, von der Handysignaturseite weiterleiten werden soll.
<i>Handy-Signatur-Parameter</i>	Parameter, welche der Handysignaturseite übergeben werden. Diese Parameter müssen als GET-Parameter in der Form ¶m1=value1¶m2=value2... formatiert werden. Mögliche Parameter siehe unter http://labs.a-trust.at/developer/ShowSource.aspx?id=113 .
<i>SigTemplateId</i>	ID des zu verwendenden Signaturtemplates.
<i>Location</i>	Ort der Signatur
<i>Reason</i>	Grund der Signatur
<i>X0</i>	Linke untere X Koordinate des Signatursiegels (in user space units)
<i>Y0</i>	Linke untere Y Koordinate des Signatursiegels (in user space units)
<i>X1</i>	Rechte obere X Koordinate des Signatursiegels (in user space units)
<i>Y1</i>	Rechte obere Y Koordinate des Signatursiegels (in user space units)



<i>Page</i>	Die Seite auf der das Signatursiegel angebracht wird.
<i>Ticket</i>	[out] Ticket, welche den Signaturauftrag auf der Box identifiziert.
<i>Handy-SignaturUrl</i>	[out] Die URL zur Handysignatur Seite auf der der Benutzer PIN/TAN eingeben muss.

Returns

Status Code siehe eErrorCodes

int HandySignaturPdfSoap.IHandySignaturPdf.UploadTemplate (string *APIKey*, byte[] *TemplateData*, out int *SigTemplateld*) Lädt ein Signaturtemplate auf den SignaturServer.

Das hochgeladene Dokument wird gegen ein xsd geprüft. Wenn diese Prüfung erfolgreich ist, wird es hinzugefügt und eine Templateld dafür generiert und zurückgeliefert.

Parameters

<i>APIKey</i>	Der erhaltene APIKey
<i>TemplateData</i>	Das Template-XML.
<i>SigTemplateld</i>	[out] Eine eindeutige Templateld.

Returns

Status Code siehe eErrorCodes