Integration Scenarios

This page will show you how to integrate eSignAnyWhere into your own products and systems. Moreover, some additional use cases are described:

- Basic Integration: Integrate eSignAnyWhere as middleware for remote signing
- Advanced Integration
    - In-Person Signing with SIGNificant Apps/Products
    - Integrate remote signing into your own (native) mobile Apps
    - Integrate remote signing into your own web portal
    - Integrate eSignAnyWhere document designer to define e-signing ceremonies in your own web application

- Basic Integration
- Advanced Integration
    - In-Person Signing with SIGNificant Apps/Products
    - Integrate remote signing into your own (native) mobile Apps
    - Integrate remote signing into your own web portal
    - Integrate eSignAnyWhere document designer to define e-signing ceremonies in your own web application
- Example of a complex integration
- Hybrid Integration (UI & API)
- Advanced Guides
    - Callbacks on Custom Events

Basic Integration

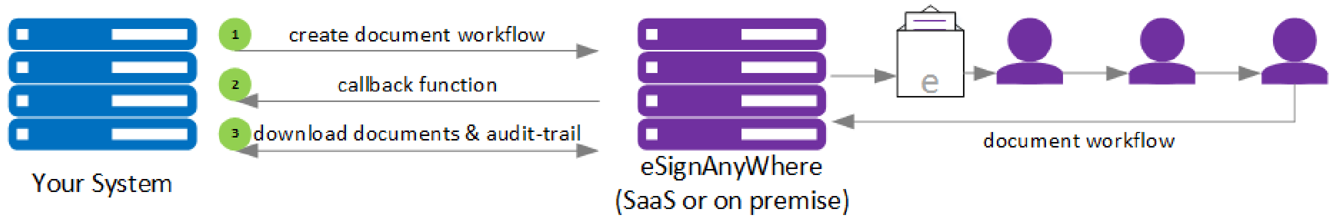
The basic integration is just using eSignAnyWhere as middleware for automating signing workflows and for the remote signing itself. Your integrating system configures envelopes, recipients and tasks via API (REST/SOAP) and uses callback functions to get the status of a finished (or rejected) envelope.

The callback function is calling the integrating system (basically a URL call with an envelope-id parameter). The integrated system gets the callback and can then check the status (signed or rejected) and init further actions, like downloading the documents and audit trail to store it in your archive system and delete the original documents from the eSignAnyWhere server.

Please note the following referring callbacks:

Only the envelope callback is fired, when the envelope is in a final state. The status update callback is fired by a sub-component and you may require to wait a post-processing time that the envelope reaches its final state. Therefore, please send back the HTTP 200 immediately!
If you are not returning the HTTP 200 immediately, eSignAnyWhere tries to recall the URL.
Attention: After some attempts the envelope will not be finished!

Due to these notes please send the message immediately after you receive the callback and before other callback-processing starts (e.g. download documents) to avoid a timeout of the callback!



Please also see the “HelloWorld” Tutorial for more information about the basic integration: visit HelloWorld_Tutorial.

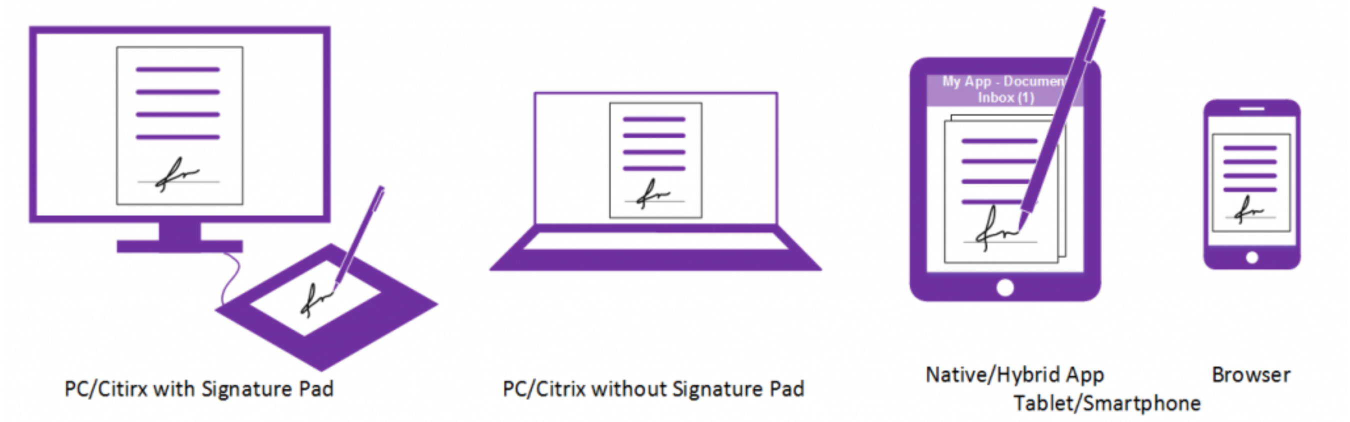
Advanced Integration

In-Person Signing with SIGNificant Apps/Products

In-Person Signing with SIGNificant Apps/Products requires a private SaaS or on-premise version of SignAnyWhere. For testing you can use our demo platform which is available at: https://demo.esignanywhere.net/

This is a typical point of sale scenario, where the customer signs the document via Signature-Pad or Touchscreen (Windows, Android, iOS). Therefore you can use the SIGNificant Products and Apps (www.xyzmo.com Solutions). So you can use in one step of the workflow the flexible SIGNificant Platform for signing. After signing the workflow continues automatically. You even can ask the customer at the point of sale if he wants to sign via eSignAnywhere remote on his mobile phone, via Signature-Pad or Tablet Computer.

It is also possible to integrate the SIGNificant Biometric Server for **biometric verification in real-time** into your use-case. This requires devices capable of recording


My App - Document Inbox (1)
PC/Citirx with Signature Pad
PC/Citrix without Signature Pad
Native/Hybrid App
Tablet/Smartphone
Browser

a biometric signature (e.g. Signaturepads, Tablet PC with native pen, Smartphones with pen, etc.) and a SIGNificant biometric server. Because of the need of the biometric server, just a on-premise installation is supported.

How to Integrate SIGNificant Products and Apps

To connect eSignAnyWhere with one of the SIGNificant Products or App you have to suppress sending the email and catch the workstepId.

General steps:

REST	SOAP
/Api/v4.0/sspfile/uploadtemporary	UploadTemporarySspFile_v1
/Api/v4.0/envelope/send	SendEnvelope_v1
/Api/v4.0/envelope/{envelopeId}	GetEnvelopeById_v1
open in SAW-Viewer with the workstepId	open in SAW-Viewer with the workstepId

1) Suppress Sending Email

Envelope Config (JSON/XML) for suppressing sending email for one recipient/workflow-step:

```
...
"Recipients": [
        {
          "Email": "recipient@email.com",
          "FirstName": "Firstname",
          "LastName": "Lastname",
          "LanguageCode": "en",
          ...
          "DisableEmail": true,
          ...
          "AuthenticationMethods": []
        }
      ],
...
```

```
...
<recipient>
        <languageCode>en</languageCode>
        <eMail>recipient@email.com</eMail>
        <firstName>Firstname</firstName>
        <lastName>Lastname</lastName>
        ...
        <disableEmail>True</disableEmail>
        ...
        <!-- optional authentication methods for this recipient -->
</recipient>
...
```

In the next section you can find a complete configuration:

```
{

  "SspFileIds": [
    "##FileId##"
  ],
  "SendEnvelopeDescription": {


  "Name": "test",
  "EmailSubject": "Please sign the enclosed envelope",
  "EmailBody": "Dear #RecipientFirstName# #RecipientLastName#\n\n#PersonalMessage#\n\nPlease sign the envelope
#EnvelopeName#\n\nEnvelope will expire at #ExpirationDate#",
  "DisplayedEmailSender": "",
  "EnableReminders": true,
  "FirstReminderDayAmount": 5,
  "RecurrentReminderDayAmount": 3,
  "BeforeExpirationDayAmount": 3,
  "DaysUntilExpire": 28,
  "CallbackUrl": "",
  "StatusUpdateCallbackUrl": "",
  "Steps": [
    {
      "OrderIndex": 1,
      "Recipients": [
        {
          "Email": "##EMAIL##",
          "FirstName": "##NAME##",
          "LastName": "##NAME##",
          "LanguageCode": "en",
          "EmailBodyExtra": "",
          "DisableEmail": false,
          "AddAndroidAppLink": false,
          "AddIosAppLink": false,
          "AddWindowsAppLink": false,
          "AllowDelegation": false,
          "AllowAccessFinishedWorkstep": false,
          "SkipExternalDataValidation": false,
          "AuthenticationMethods": [
            {
              "Method": "Pin",
              "Parameter": "1234"
            }
          ]
        }
      ],
      "EmailBodyExtra": "",
      "RecipientType": "Signer",
      "WorkstepConfiguration": {
        "WorkstepLabel": "test",
        "SmallTextZoomFactorPercent": 100,
        "FinishAction": {
          "ServerActions": [],
          "ClientActions": []
        },
        "ReceiverInformation": {
          "UserInformation": {
            "FirstName": "##NAME##",
            "LastName": "##NAME##",
            "EMail": "##EMAIL##"
          },
          "TransactionCodePushPluginData": []
        },
        "SenderInformation": {
          "UserInformation": {
            "FirstName": "##NAME##",
            "LastName": "##NAME##",
            "EMail": "##EMAIL##"
          }
        },
        "TransactionCodeConfigurations": [
          {
            "Id": "smsAuthTransactionCodeId",
            "HashAlgorithmIdentifier": "Sha256",
            "Texts": [

            ]
          }
        ],
        "SignatureConfigurations": [],
        "ViewerPreferences": {
          "FinishWorkstepOnOpen": false,
          "VisibleAreaOptions": {
            "AllowedDomain": "*",
            "Enabled": false
          }
        },
        "ResourceUris": {
          "SignatureImagesUri": "http://beta4.testlab.xyzmo.com//Resource/SignatureImages/?
link=1agjn5MvqNpSt2jFiZQySxLEiAO~ecLOxKqy3soEHk2F4Dz1MPSYLxRkpA21XMkYY"
        },
        "AuditingToolsConfiguration": {
          "WriteAuditTrail": false,
          "NotificationConfiguration": {}
        },
        "Policy": {
          "GeneralPolicies": {
            "AllowSaveDocument": true,
            "AllowSaveAuditTrail": true,
            "AllowRotatingPages": false,
            "AllowEmailDocument": true,
            "AllowPrintDocument": true,
            "AllowFinishWorkstep": true,
            "AllowRejectWorkstep": true,
            "AllowRejectWorkstepDelegation": false,
            "AllowUndoLastAction": false,
            "AllowAdhocPdfAttachments": false,
            "AllowAdhocSignatures": false,
            "AllowAdhocStampings": false,
            "AllowAdhocFreeHandAnnotations": false,
            "AllowAdhocTypewriterAnnotations": false,
            "AllowAdhocPictureAnnotations": false,
            "AllowAdhocPdfPageAppending": false
          },
          "WorkstepTasks": {
            "PictureAnnotationMinResolution": 0,
            "PictureAnnotationMaxResolution": 0,
            "PictureAnnotationColorDepth": "Color16M",
            "SequenceMode": "NoSequenceEnforced",
            "PositionUnits": "PdfUnits",
            "ReferenceCorner": "Lower_Left",
            "Tasks": [
              {
                "Texts": [
                  {
```

```
          "Language": "*",
          "Value": "Signature Disclosure Text"
        },
        {
          "Language": "en",
          "Value": "Signature Disclosure Text"
        }
      ],
      "Headings": [
        {
          "Language": "*",
          "Value": "Signature Disclosure Subject"
        },
        {
          "Language": "en",
          "Value": "Signature Disclosure Subject"
        }
      ],
      "IsRequired": false,
      "Id": "ra",
      "DisplayName": "ra",
      "DocRefNumber": 1,
      "DiscriminatorType": "Agreements"
    },
    {
      "PositionPage": 1,
      "Position": {
        "PositionX": 63.0,
        "PositionY": 603.0
      },
      "Size": {
        "Height": 80.0,
        "Width": 190.0
      },
      "AdditionalParameters": [
        {
          "Key": "enabled",
          "Value": "1"
        },
        {
          "Key": "positioning",
          "Value": "onPage"
        },
        {
          "Key": "req",
          "Value": "1"
        },
        {
          "Key": "fd",
          "Value": ""
        },
        {
          "Key": "fd_dateformat",
          "Value": "dd-MM-yyyy HH:mm:ss"
        },
        {
          "Key": "fd_timezone",
          "Value": "datetimeutc"
        },
        {
          "Key": "spcId",
          "Value": "tLevelId"
        }
      ],
      "AllowedSignatureTypes": [
        {
          "AllowedCapturingMethod": "Click2Sign",
          "Id": "679dd763-6e25-4a68-929d-cb1ce13dac7e",
          "DiscriminatorType": "SigTypeClick2Sign",
          "Preferred": false,
          "StampImprintConfiguration": {
```

```
                        "DisplayExtraInformation": true,
                        "DisplayEmail": true,
                        "DisplayIp": true,
                        "DisplayName": true,
                        "DisplaySignatureDate": true,
                        "FontFamily": "Times New Roman",
                        "FontSize": 11.0
                      }
                    }
                  ],
                  "UseTimestamp": false,
                  "IsRequired": true,
                  "Id": "1#XyzmoDuplicateIdSeperator#Signature_a1e940eb-bcd5-2222-9777-f3570faedf3f",
                  "DisplayName": "",
                  "DocRefNumber": 1,
                  "DiscriminatorType": "Signature"
                }
              ]
            }
          },
          "Navigation": {
            "HyperLinks": [],
            "Links": [],
            "LinkTargets": []
          }
        },
        "DocumentOptions": [
          {
            "DocumentReference": "1",
            "IsHidden": false
          }
        ],
        "UseDefaultAgreements": true
      },
      {
        "OrderIndex": 2,
        "Recipients": [
          {
            "Email": "##EMAIL##",
            "FirstName": "##NAME##",
            "LastName": "##NAME##",
            "LanguageCode": "en",
            "EmailBodyExtra": "",
            "DisableEmail": false,
            "AddAndroidAppLink": false,
            "AddIosAppLink": false,
            "AddWindowsAppLink": false,
            "AllowDelegation": false,
            "SkipExternalDataValidation": false,
            "AuthenticationMethods": []
          }
        ],
        "EmailBodyExtra": "",
        "RecipientType": "Cc",
        "DocumentOptions": [],
        "UseDefaultAgreements": false
      }
    ],
    "AddFormFields": {
      "Forms": {}
    },
    "OverrideFormFieldValues": {
      "Forms": {}
    },
    "AttachSignedDocumentsToEnvelopeLog": false
}
}
```

```
<envelope>
   <name>eSignAnyWhere Tutorial</name>
   <eMailSubject>Document of eSignAnyWhere Tutorial</eMailSubject>
   <eMailBody>Dear #RecipientFirstName#! Please sign this tutorial document.</eMailBody>
   <enableReminders>True</enableReminders>
   <firstReminderDayAmount>1</firstReminderDayAmount>
   <recurrentReminderDayAmount>1</recurrentReminderDayAmount>
   <beforeExpirationReminderDayAmount>1</beforeExpirationReminderDayAmount>
   <daysUntilExpire>2</daysUntilExpire>
   <!-- callback to your backend system on a completed envelope
   <callbackUrl>http://172.16.17.78:57550/default.aspx?EnvelopeId=##EnvelopeId##&myParamForMe=1234<
/callbackUrl>
   -->
   <callbackUrl />
   <steps>
      <step>
         <emailBodyExtra />
         <orderIndex>1</orderIndex>
         <recipientType>Signer</recipientType>
         <recipients>
            <recipient>
               <languageCode>en</languageCode>
               <eMail>##SIGNER-EMAIL##</eMail>
               <firstName>Alice</firstName>
               <lastName>Somename</lastName>
               <disableEmail>True</disableEmail>
            </recipient>
         </recipients>
         <workstepConfiguration>
<WorkstepLabel />
   <SmallTextZoomFactorPercent>100</SmallTextZoomFactorPercent>
   <WorkstepTimeToLiveInMinutes>11520</WorkstepTimeToLiveInMinutes>
   <FinishAction />
   <signatureTemplate>
      <version>1.2.0.2</version>
      <positionUnits>PdfUnits</positionUnits>
      <positionReferenceCorner>Lower_Left</positionReferenceCorner>
      <sig id="93cce567-ae5c-4e98-ac99-9f56ac034250">
         <positionPage>1</positionPage>
         <DocRefNumber>1</DocRefNumber>
         <positionX>80.22857</positionX>
         <positionY>158.8629</positionY>
         <width>171.4286</width>
         <height>68.57143</height>
         <param name="enabled">1</param>
         <param name="completed">0</param>
         <param name="sigType">Picture</param>
         <param name="positioning">onPage</param>
         <param name="allowedCapturingMethods">Click2Sign</param>
      </sig>
   </signatureTemplate>
   <Policy version="1.1.0.0">
      <GeneralPolicies>
         <AllowSaveDocument>1</AllowSaveDocument>
         <AllowSaveAuditTrail>1</AllowSaveAuditTrail>
      </GeneralPolicies>
      <WorkstepTasks SequenceMode="SequenceOnlyRequiredTasks" originalSequenceMode="
SequenceOnlyRequiredTasks">
         <Task enabled="1" completed="0" required="1" id="93cce567-ae5c-4e98-ac99-9f56ac034250" displayName="
SignField 1" DocRefNumber="1" type="SignField" internalAllConcernedDocRefNumbersList="1"
allRequiredFieldsFilledOnWorkstepCreation="0" />
      </WorkstepTasks>
   </Policy>
   <TransactionCodeConfigurations>
      <TransactionCodeConfiguration trConfId="">
         <Message>Please sign the document with the transactionId {tId} with the code: {Token}</Message>
         <hashAlgorithmIdentifier>Sha256</hashAlgorithmIdentifier>
      </TransactionCodeConfiguration>
      <TransactionCodeConfiguration trConfId="Trans1">
         <Message>Please accept the transactionId {tId} with the code: {Token}</Message>
         <hashAlgorithmIdentifier>Sha256</hashAlgorithmIdentifier>
      </TransactionCodeConfiguration>
   </TransactionCodeConfigurations>
</workstepConfiguration>
      </step>
   </steps>
</envelope>
```

2) Links to SIGNificant Products or catch a WorkstepId

eSAW supports to send out links for the SIGNificant products automatically via notifications. Therefore, you just have to add to the recipient configuration (JSON/XML) the following parameters:

```
"Recipients": [
        {
          "AddAndroidAppLink": false,
          "AddIosAppLink": false,
          "AddWindowsAppLink": false,
        }
      ],
```

```
<envelope>
  <steps>
    <step>
      <recipients>
        <recipient>
          <addAndroidAppLink>0</addAndroidAppLink> <!-- 0 or 1 -->
          <addIosAppLink>0</addIosAppLink> <!-- 0 or 1 -->
          <addWindowsAppLink>0</addWindowsAppLink> <!-- 0 or 1 -->
        </recipient>
      </recipients>
    </step>
  </steps>
</envelope>
```

Otherwise you can connect one workstep of eSignAnyWhere with one of the SIGNificant Apps. After you have sent the envelope and you have got the envelopeId you can call the method /Api/v4.0/envelope/{envelopeId} for REST or`GetEnvelopeById_v1 for SOAP .` With the result of this method you get information about the envelope and you can find the `workstepRedirectionURL`. This URL forwards the SignAnywhere Viewer (Web-Client), but with the additional parameter `&responseType=returnWorkstepId` it returns the workstepId. Example:

`https://demo.xyzmo.com/workstepredirector/sign?identifier=8Wn..9dX&responseType=returnWorkstepId`

With this WorkstepId you can now connect the SIGNificant product to the document. If the document is finished the workflow continues automatically.

Other parameters are:

- `responseType=redirectToViewer` – redirects to SAW Viewer (default)
- `responseType=redirectToAndroidApp` – redirects to Android App
- `responseType=redirectToIOsApp` – redirects to iOS App
- `responseType=redirectToWindowsApp` – redirects to Windows App
- `responseType=returnWorkstepId` – returns the WorkstepId for other integration types

3) Client Redirect

If you want, that the recipient is not redirected to the significant.com webpage after signing the document, you can use the finish action to redirect the recipient to your preferred webpage.

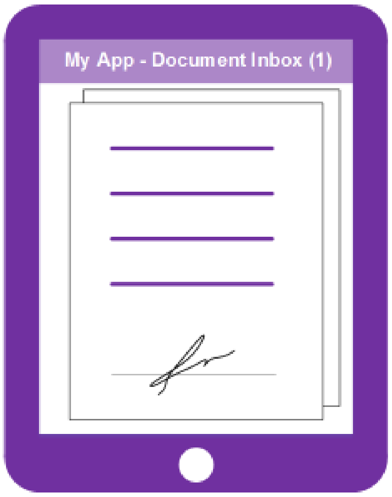
Therefore, you have to use the *FinishAction* in the *WorkstepConfiguration* of the recipient.

```
"FinishAction": {
          "ServerActions": [],
          "ClientActions": []
        },
<FinishAction>
<ClientAction clientName="SIGNificant SignAnywhere" closeApp="0" RemoveDocumentFromRecentDocumentList="0"
CallClientActionOnlyAfterSuccessfulSync="1">http://www.xyzmo.com</ClientAction>
</FinishAction>
```

The ClientAction in the SOAP section would forward the SignAnywhere Viewer, after the recipient has signed the document to www.xyzmo.com. You can replace the link with your own or even with parameter to recipient specific pages.

Integrate remote signing into your own (native) mobile Apps

Integrate remote signing into your own (native) mobile Apps requires a private SaaS or on-premise version of eSignAnyWhere and the SIGNificant SDK for your desired platforms. You can test it on our demo platform which is available at: https://demo.esignanywhere.net/.



If you want to integrate the document signing for customers into your own Apps, we offer mobile SDKs for you. We support iOS, Android and Windows. Moreover, we have a 100% native SDK and a Hybrid SDK (for a Webbased-Document-Viewer).

So you can integrate into your App a user document-inbox, where the user find documents he/she has to sign. If the user has finished the document, the workflow (e.g. for internal review) continues automatically.

With this integration you can use your notification-channel (no emails via eSignAnyWhere), extend your App and users read & sign documents directly in your App. The use will not see, that he/she is using eSignAnyWhere, because it is seamless integrated into your App.

If you are interested in our mobile SDKs for integrating into your Apps, please contact us. We will provide you information about the Terms & Conditions and the SDK.

Integrate remote signing into your own web portal

This use case surpress sending emails via eSignAnyWhere, because you are sending to your customer the notifications and direct them to your web portal. The customer has to login at your web portal and navigate to the “inbox” or a “document section” and gets a list of all documents he/she has to sign. The eSignAnyWhere Viewer is integrated into your web portal.

How to integrate

Basically it is the same as In-Person Signing with SIGNificant Apps/Products. You have to surpress sending the email for the recipient (via JSON/XML) and instead of catching the WorkstepID you just take the `workstepRedirectURL` of the response of /Api/v4.0/envelope/{envelopeId} for REST or `GetEnvelopeById_v1 for SOAP`. This link you can show the user in your web portal for signing the document.

Integrate eSignAnyWhere document designer to define e-signing ceremonies in your own web application

If you are integrating eSignAnyWhere to be used within your web portal, you can use the eSignAnyWhere WYSIWYG document designer for defining e-signing ceremonies.

How to integrate

1) Create Draft and embed designer

General steps:

REST	SOAP
/Api/v4.0/sspfile/uploadtemporary	UploadTemporarySspFile_v1
/Api/v4.0/envelope/create	CreateDraft_v1
Open with SAW-Viewer	Open with SAW-Viewer

After uploading a file you have to create a draft and configure the option to allow an external designer (`allowAgentRedirect`). For the create draft call you need the documentId which you got from the upload a file call, an envelope description and the draft option.

```
{
  "SspFileIds": [
    "##FileId##"
  ],
  "SendEnvelopeDescription": {


  "Name": "test",
  "EmailSubject": "Please sign the enclosed envelope",
  "EmailBody": "Dear #RecipientFirstName# #RecipientLastName#\n\n#PersonalMessage#\n\nPlease sign the envelope
#EnvelopeName#\n\nEnvelope will expire at #ExpirationDate#",
  "DisplayedEmailSender": "",
  "EnableReminders": true,
  "FirstReminderDayAmount": 5,
  "RecurrentReminderDayAmount": 3,
  "BeforeExpirationDayAmount": 3,
  "DaysUntilExpire": 28,
  "CallbackUrl": "",
  "StatusUpdateCallbackUrl": "",
  "Steps": [
    {
      "OrderIndex": 1,
      "Recipients": [
        {
          "Email": "##EMAIL##",
          "FirstName": "##NAME##",
          "LastName": "##NAME##",
          "LanguageCode": "en",
          "EmailBodyExtra": "",
          "DisableEmail": false,
          "AddAndroidAppLink": false,
          "AddIosAppLink": true,
          "AddWindowsAppLink": false,
          "AllowDelegation": true,
          "SkipExternalDataValidation": false,
          "AuthenticationMethods": []
        }
      ],
      "EmailBodyExtra": "",
      "RecipientType": "Signer",



      "DocumentOptions": [
        {
          "DocumentReference": "1",
          "IsHidden": false
        }
      ],
      "UseDefaultAgreements": true
    },
    {
      "OrderIndex": 2,
      "Recipients": [
        {
          "Email": "##EMAIL##",
          "FirstName": "##NAME##",
          "LastName": "##NAME##",
          "LanguageCode": "en",
          "EmailBodyExtra": "",
          "DisableEmail": false,
          "AddAndroidAppLink": false,
          "AddIosAppLink": false,
          "AddWindowsAppLink": false,
          "AllowDelegation": false,
          "SkipExternalDataValidation": false,
          "AuthenticationMethods": []
        }
      ],
      "EmailBodyExtra": "",
      "RecipientType": "Cc",
      "DocumentOptions": [],
      "UseDefaultAgreements": false
    }
  ],
  "AddFormFields": {
    "Forms": {}
  },
  "OverrideFormFieldValues": {
    "Forms": {}
  },
  "AttachSignedDocumentsToEnvelopeLog": false
},
"CreateDraftOptions": {
    "AfterSendRedirectUrl": "http://www.mycallback.at/DraftAfterSendRedirect?
envelope=##EnvelopeId##&action=##Action##",
    "AfterSendCallbackUrl": "http://www.mycallback.at/DraftAfterSendCallback?
envelope=##EnvelopeId##&action=##Action##",
    "RedirectPolicy": "ToDesigner",
    "AllowAgentRedirect": true,
    "IframeWhiteList": "http://172.16.17.256;http://foo.org"
  }
}
```

```
<envelope>
   <name>eSignAnyWhere Tutorial</name>
   <eMailSubject>Document of eSignAnyWhere Tutorial</eMailSubject>
   <eMailBody>Dear #RecipientFirstName#! Please sign this tutorial document.</eMailBody>
   <enableReminders>True</enableReminders>
   <firstReminderDayAmount>1</firstReminderDayAmount>
   <recurrentReminderDayAmount>1</recurrentReminderDayAmount>
   <beforeExpirationReminderDayAmount>1</beforeExpirationReminderDayAmount>
   <daysUntilExpire>2</daysUntilExpire>
   <!-- callback to your backend system on a completed envelope
   <callbackUrl>http://172.16.17.78:57550/default.aspx?EnvelopeId=##EnvelopeId##&myParamForMe=1234<
/callbackUrl>
   -->
   <callbackUrl />
   <steps>
      <step>
         <emailBodyExtra />
         <orderIndex>1</orderIndex>
         <recipientType>Signer</recipientType>
         <recipients>
            <recipient>
               <languageCode>en</languageCode>
               <eMail>##EMAIL##</eMail>
               <firstName>Alice</firstName>
               <lastName>Somename</lastName>
               <disableEmail>false</disableEmail>
                <addIosAppLink>1</addIosAppLink>
            </recipient>
         </recipients>
      </step>
      <step>
         <emailBodyExtra />
         <orderIndex>2</orderIndex>
         <recipientType>CC</recipientType>
         <recipients>
            <recipient>
               <languageCode>en</languageCode>
               <eMail>##EMAIL##</eMail>
               <firstName>Charly</firstName>
               <lastName>Randomname</lastName>
            </recipient>
         </recipients>
      </step>
   </steps>
</envelope>
```

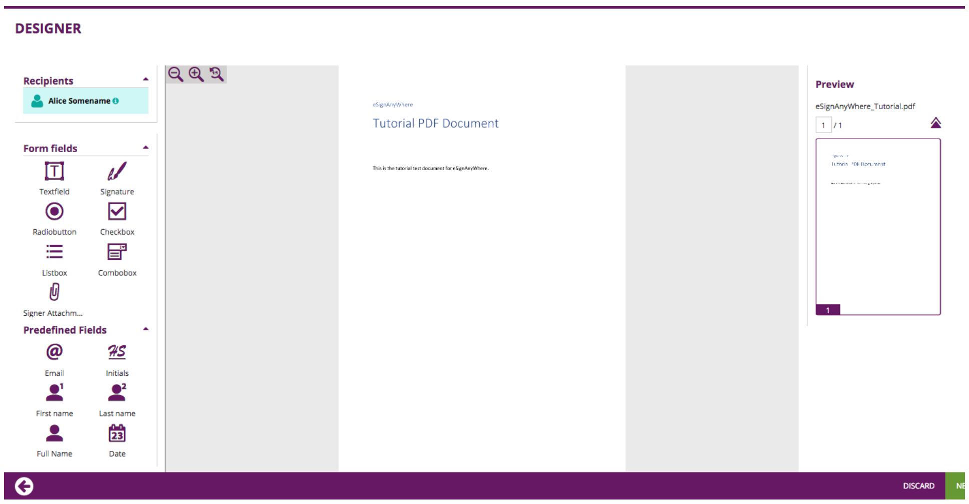
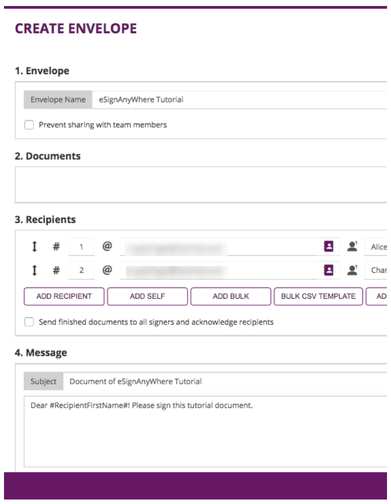
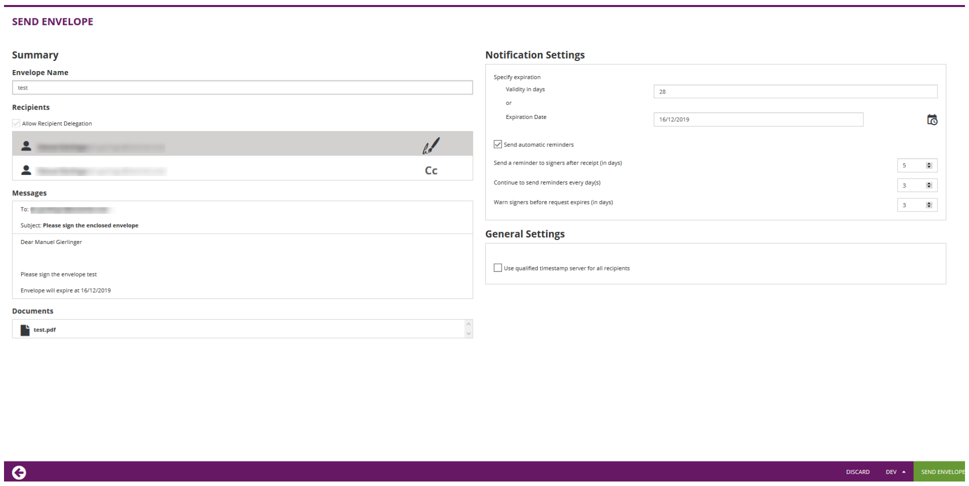
In REST the CreateDraftOptions is already given in the envelope description.

```
You have to add the following section for the CreateDraftOptions:

<draftOptions>
        <afterSendRedirectUrl>http://www.mycallback.at/DraftAfterSendRedirect?
envelope=##EnvelopeId##&action=##Action##</afterSendRedirectUrl>
        <afterSendCallbackUrl>http://www.mycallback.at/DraftAfterSendCallback?
envelope=##EnvelopeId##&action=##Action##</afterSendCallbackUrl>
        <redirectPolicy>ToDesigner</redirectPolicy>
        <allowAgentRedirect>true</allowAgentRedirect>
        <iFrameWhiteList>http://172.16.17.256;http://foo.org</iFrameWhiteList>
</draftOptions>
```

RedirectPolicy values:

- “ToDesigner” (like it is in the example)
- “ToRecipients” -> in JSON, “ToCreateEnvelope” in XML
- “ToSend”

ToDesigner	ToCreateEnvelope
	
ToSend	
	

DESIGNER

CREATE ENVELOPE

1. Envelope

Envelope Name: eSignAnyWhere Tutorial

☐ Prevent sharing with team members

2. Documents

3. Recipients

1 @ [redacted] Alice

2 @ [redacted] Char

☐ Send finished documents to all signers and acknowledge recipients

4. Message

Subject: Document of eSignAnyWhere Tutorial

Dear #RecipientFirstName#! Please sign this tutorial document.

SEND ENVELOPE

Summary

Envelope Name: test

Recipients

☑ Allow Recipient Delegation

[redacted]

[redacted] Cc

Messages

To: [redacted]

Subject: **Please sign the enclosed envelope**

Dear Manuel Gierlinger

Please sign the envelope test

Envelope will expire at 16/12/2019

Documents

test.pdf

Notification Settings

Specify expiration

Validity in days: 28

or

Expiration Date: 16/12/2019

☑ Send automatic reminders

Send a reminder to signers after receipt (in days): 5

Continue to send reminders every day(s): 3

Warn signers before request expires (in days): 3

General Settings

☐ Use qualified timestamp server for all recipients

On the create envelope page you may think that you have to upload the document again but you do not have to. The document which you have uploaded before via the api call is still there. If go forward to the “Designer” page you can see the document.

Note: With the redirectPolicy you can decide if the SAW-Viewer starts with the “Designer” page, with the “CreateEnvelope” page or with the “SendEnvelope” page.

The option `allowAgentRedirect` enables an anonymous designer integration (without eSignAnyWhere Login) and `iFrameWhiteList` extends the HTTP header with a list to integrate in your web application or portal (via `X-FRAME-OPTIONS`).

2) Integrate Designer

The designer can be embedded by modifying the following string:

```
http://www.significant.com/AgentRedirect/index?draftid=##draftid##
```

or

```
https://demo.xyzmo.com/AgentRedirect/index?draftid=##draftid##
```

If the draft is finished you can start the envelope.

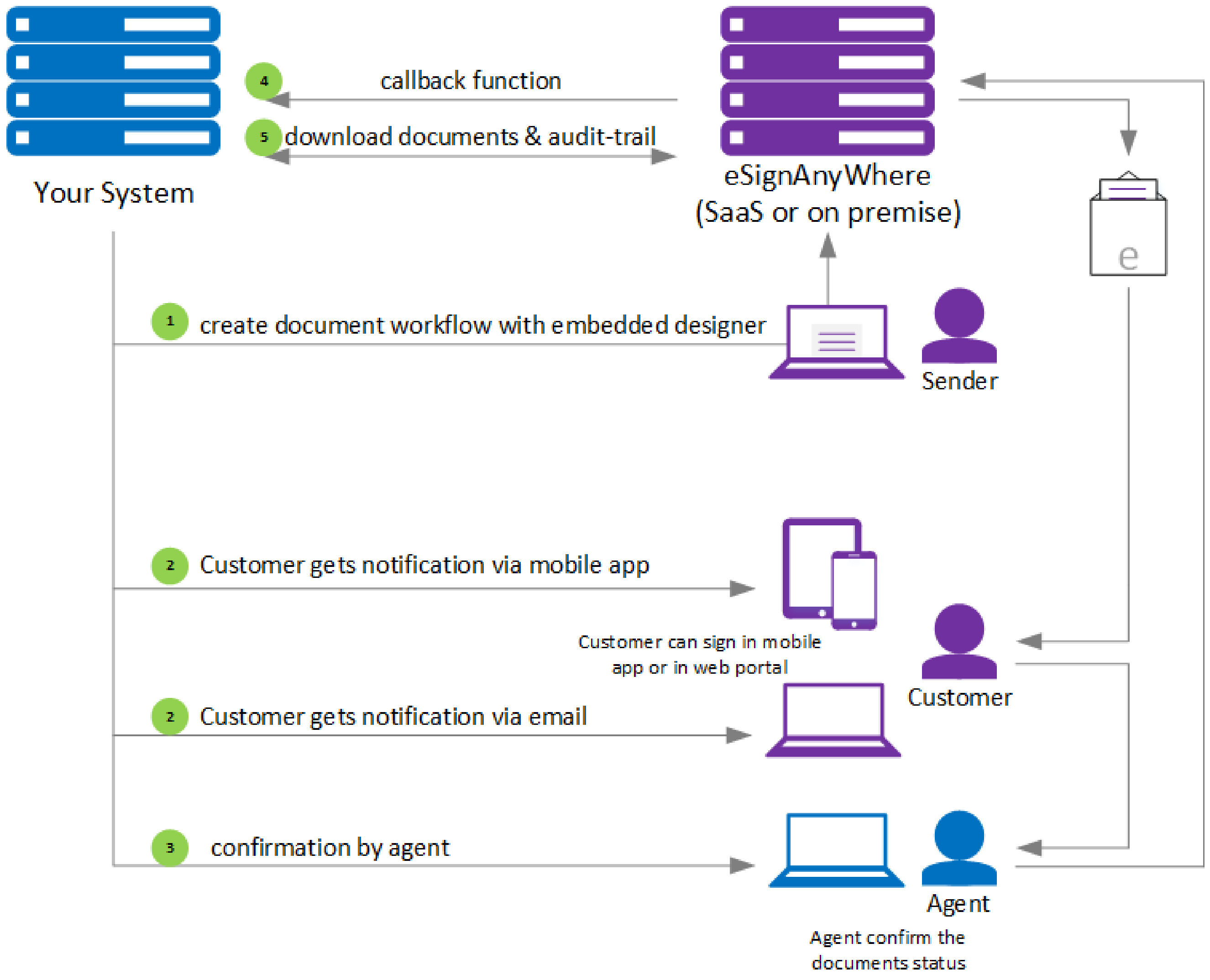
ⓘ Make sure to open the link above (to embed the designer) in a private window (In Firefox->File->New Private Window. Otherwise the browser will remember the session and you will not have access to significant or https://demo.esignanywhere.net/ after the draft is finished.

The following video shows the process from opening the draft to finish and send the envelope.

create/send envelope

Example of a complex integration

The following picture show you an example of a complex integration/use case.

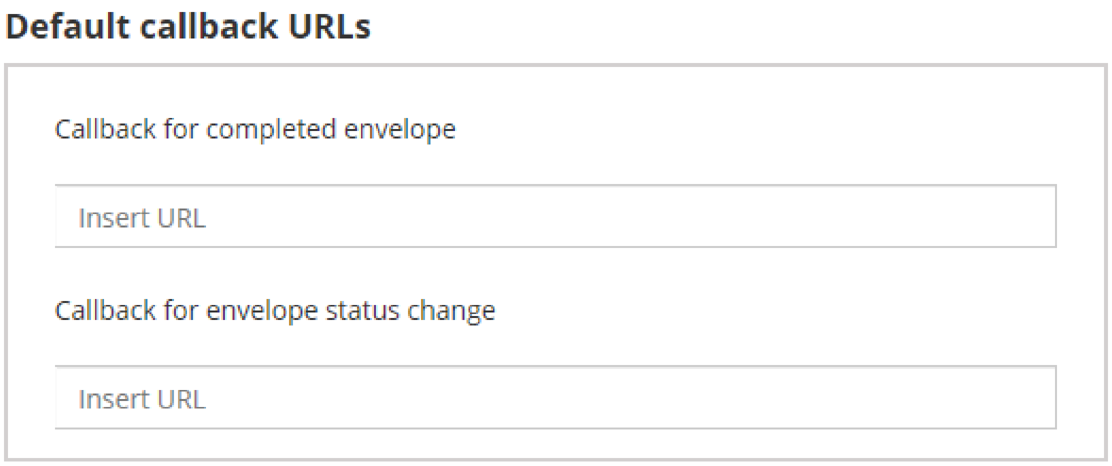

callback function
4
download documents & audit-trail
5
Your System
eSignAnyWhere
(SaaS or on premise)
e
1
create document workflow with embedded designer
Sender
2
Customer gets notification via mobile app
Customer can sign in mobile app or in web portal
Customer
2
Customer gets notification via email
3
confirmation by agent
Agent
Agent confirm the documents status

Description:

1. An agent designs in your web portal via integrated document designer the e-signing ceremony
2. The first signer (customer) gets a notification via mobile app on his smartphone/tablet and gets an email that a document can be signed in the web portal. The customer is able to sign in the app or in the web-portal after login.
3. An agent has to confirm the signed document. The agent gets an email from the eSignAnyWhere System and signs it in the browser.

Hybrid Integration (UI & API)

In some scenarios it makes sense to send the envelopes via UI by the user and have an automated post processing, when the envelope gets finished. Therefore, you can configure in the organization settings a default callback, which is used by using the UI. You still can overwrite it via API.


Default callback URLs

Callback for completed envelope

Callback for envelope status change

Recommended is to use the envelope status change callback to get detailed information about the callbacks. See API Reference - SOAP for details how to configure the callback.

So you can allow users to send envelopes via eSAW UI and integrate a post processing, e.g. for automated archiving.

Advanced Guides

Callbacks on Custom Events

You can define specific callbacks on specific events (e.g. if you want to get notified when a signer rejects the agreement text). A detailed description of this feature you can find here.