Tutorial: Hello World

This tutorial shows how eSignAnyWhere can be implemented. This guide is intended to get to know the basics of a signature request workflow. After this guide you will be able to create envelopes and send them. For more information about the REST API you can also start with the Postman Tutorial. For more information about the XML configurations for the API please also have a look at our XML Guide.

eSignAnyWhere can be easily implemented. This tutorial shows you how to send your first envelope via api REST service of eSignAnyWhere. If you are using Postman for your REST calls please also have a look at the following Tutorial: Visit postman tutorial.
Please also see the developer mode for this tutorial: Visit developer mode.
For this tutorial you can use your desired programming language (with REST support) or any REST tool (e. g. Postman). Moreover you will need an active eSignAnyWhere account for the authorization. (Even a trial account will work).

If you want to use SOAP for your API calls you can use SoapUI. For this tool you can find a tutorial here: Visit SoapUI Sample. For SOAP you might be also interested in the envelope XML guide which explains you more about the XML configuration.

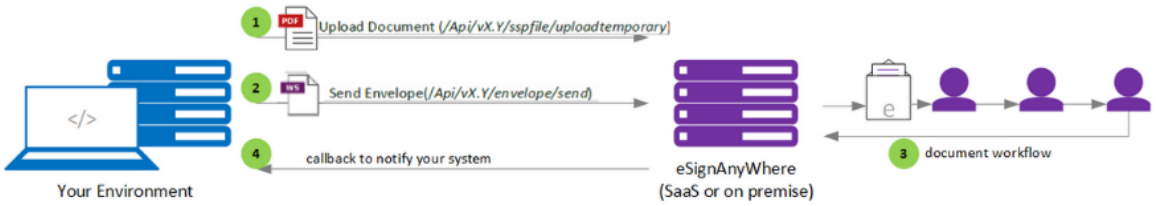
If you are using JAVA you may be interested in our Java Library.

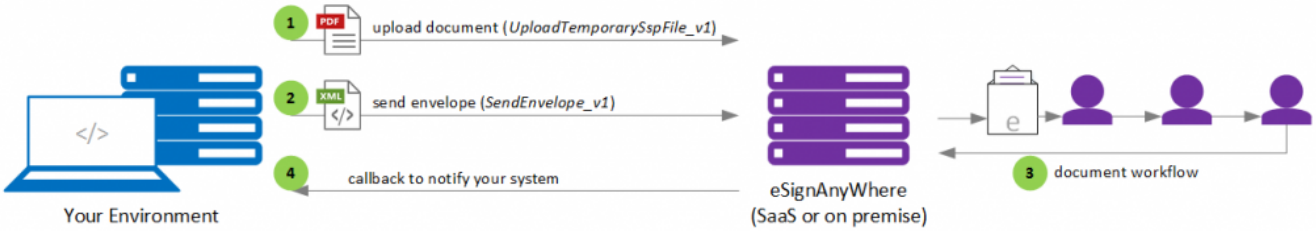
The REST URI of eSignAnyWhere is: https://demo.esignanywhere.net/Api/v4.0/
here you can find the Swagger: https://demo.esignanywhere.net/Api

The SOAP endpoint of eSignAnyWhere is: https://demo.esignanywhere.net/api.asmx

What the tutorial is about

The tutorial will show you the following basic use-case:





First we will have an easy api call to start with and then we will implement the above shown use case. Basically you simply upload a document. After the upload you will receive a documentId, which you need for creating an envelope with a JSON configuration. This configuration contains information about the envelope itself, workflow steps, signer information, policies and many more. After the envelope is created successful the workflow starts automatically. The eSignAnyWhere system will use callbacks (you can configure them in the envelope JSON configuration) to notify you if the envelope status has changed.

First we will have an easy api call to start with and then we will implement the above shown use case. Basically you simply upload a document. After the upload you will receive a documentId, which you need for creating an envelope with a JSON configuration. This configuration contains information about the envelope itself, workflow steps, signer information, policies and many more. After the envelope is created successful the workflow starts automatically. The eSignAnyWhere system will use callbacks (you can configure them in the envelope JSON configuration) to notify you if the envelope status has changed.

Your first api call

First we will try the connection with a version request. You can find the endpoints in the next table:

REST	SOAP
https://demo.esignanywhere.net/Api/v4.0/version*	GetVersion_v1*

*The version api call does not need an authorization.

The response should be like the following in REST:

```
{ "Success": true, "Version": "3.7.78.14788" }
```

and in SOAP:

```
<apiResult version="3.7.78.14788">
        <baseResult>ok</baseResult>
</apiResult>
```

Two ways of authorization

Almost all API calls are expecting an authorization. Therefore, your organization key and your email-address of a PowerUser is required or the apiToken. You can find the organization key in the Settings/Api Token and Apps in eSignAnyWhere. The email-address and the name of the PowerUser is used for sending the email.

You can either authorize with the

- organizationKey and the UserLoginName or you can use
- the apiToken ≥20.42

for the authorization. For more information about the two different authorizations please have a look at the REST tutorial using Postman

The authorization XML has to be encoded via HTML special characters (“<” in “<” and “>” in “>”). If this encoding is not done you will receive a HTTP 400 Bad Request error.
Higher programming languages takes automatically care of the conversation, so just in lower languages it is required (or also SOAPUI). You can find a configuration for the authorization for SOAP (XML) in the next lines:

Note: The next lines show a authorization with the organizationKey and the userLoginName.

```
<authorization>
        <organizationKey>4647688a-xxxx-xxxx-xxxx-xxxxxxxx</organizationKey>
        <userLoginName>your@email.address</userLoginName>
</authorization>
```

Inline-XML (with HTML special characters):

```
&lt;authorization&gt;
&lt;organizationKey&gt;4647688a-xxxx-xxxx-xxxx-xxxxxxxx&lt;/organizationKey&gt;
&lt;userLoginName&gt;your@email.address&lt;/userLoginName&gt;
&lt;/authorization&gt;
```

You can also authorize with the apiToken. Please see the following configurations:

```
<authorization>
        <apiToken >hizit4enf8ellb6b5hwh5b------------------------------</apiToken >
</authorization>
```

Inline-XML (with HTML special characters):

```
&lt;authorization&gt;
        &lt;apiToken &gt;hizit4enf8ellb6b5hwh5b----------------------------&lt;/apiToken &gt;
&lt;/authorization&gt;
```

Errors:
For REST you get a HTTP status for the calls (for example: 200 OK or 401 unauthorized).

In case of errors, the error message is part of the response. In baseResult you can see the state of the call (ok/failed) and in the error-Info or ok-Info the response. The following code shows you an authorization error:

```
<apiResult version="2.2.458.6616">
        <baseResult>failed</baseResult>
        <errorInfo>
                <error>ERR0011</error>
                <errorMsg>ERR0011: Parameter apiAuthorizationXml is incorrect: Failed to parse</errorMsg>
        </errorInfo>
</apiResult>
```

In case of errors, the error message is part of the response. In baseResult you can see the state of the call (ok/failed) and in the error-Info or ok-Info the response. The following code shows you an authorization error:

Upload document

Note: This call needs the authorization.

First we have to upload a document. This will return a documentId, to use the document for creating an envelope. Therefore, we will use the following endpoints:

REST	SOAP
https://demo.esignanywhere.net/Api/v4.0/sspfile/uploadtemporary	UploadTemporarySspFile_v1

For this tutorial we just use a simple PDF document. You can download it here or use your own document.

You need to Base64 encode the file.

```
<file>
        <name>eSignAnyWhere_Tutorial.pdf</name>
        <data>##BASE64-FILE-CONTENT##</data>
</file>
```

This XML also has to be encoded via HTML special charaters (“<” in “`<`” and “>” in “`&gt`”).

The response after a successfull upload of the file is the document id (SspFileId).

You need the document Id for creating your first envelope. After uploading the file it is just temporary on the server. After 10 minutes it will be deleted and you are not able to use it again. The moment you are creating an envelope with the file, it gets the time-to-live of the envelope. Attention: The returned file id is just for creating an envelope. Once the envelope is created, the file id becomes invalid, so for downloading a finished file, you have to retrieve the id from the envelope status with the following URI for REST: https://demo.esignanywhere.net/Api/v4.0/envelope/##envelopeId## (Replace the placeholder ##envelopeId## with your envelope id.

Get Workstep Configuration

This step is only optional! But required if you do not have a workstep configuration. A workstep configuration is a definition of a signing task for one recipient. So it contains information about signature fields on the document, form-fields, etc. With the new Developer-Feature of eSignAnyWhere 2.6 it is not required anymore.

A workstep configuration is required for each signing step in your workflow. If you have already your workstep configuration you can skip this step. Moreover you can generate a workstep configuration with the SIGNificant Workstep Designer (Note: Only for XML). The SIGNificant Workstep Designer allows you as a developer to configure a SignAnyWhere workstep, set signature fields and types and many more. You can also use the developer mode for JSON (REST) and SOAP(XML)

To receive an adhoc workstep configuration you can use the following endpoint:

REST	SOAP
https://demo.esignanywhere.net/Api/v4.0/envelope/prepare	GetAdHocWorkstepConfiguration_v1

You can find a sample configuration for this call in the next section


```
{
  "SspFileIds": [
    "##FileId##"
  ],
  "AdHocWorkstepConfiguration": {
    "WorkstepLabel": "string",
    "SmallTextZoomFactorPercent": 0,
    "WorkstepTimeToLiveInMinutes": 0,
    "FinishAction": {
      "ClientActions": [
        {
          "RemoveDocumentFromRecentDocumentList": true,
          "CallClientActionOnlyAfterSuccessfulSync": true,
          "ClientName": "string",
          "CloseApp": true,
          "Action": "string"
        }
      ]
    },
    "NoSequenceEnforced": true,
    "SigTemplate": {
      "Size": {
        "Height": 0,
        "Width": 0
      },
      "AllowedSignatureTypes": [

      ]
    },
    "ParseFormFields": {
      "MapRequiredFieldsToRequiredTask": true,
      "FormsGrouping": "PerPage",
      "ReturnSimplifiedConfig": true,
      "AddKeepExistingValueFlag": true,
      "ParseFormField": true
    },
    "AdhocPolicies": {
      "AllowModificationsAfterSignature": true
    },
    "ViewerPreferences": {
      "ShowPageNavigationBar": true,
      "ShowThumbnails": true,
      "SkipFinishConfirmDialog": true,
      "SkipDocumentDialog": true,
      "ShowImagesInFullWidth": true,
      "DisableGeolocation": true,
      "ShowDocumentDownloadDialogAfterAutomaticFinish": true,
      "AttachmentsMaxFileSize": 0,
      "SkipPreviewImageOnDisposableCertificate": true,
      "LoadCustomJs": true,
      "AllowCustomButtons": true,
      "GuidingBehavior": "GuideOnlyRequiredTasks",
      "FormFieldsGuidingBehavior": "AllowSubmitAlways",
      "ShowVersionNumber": true,
      "EnableWarningPopupOnLeave": true,
      "WarningPopupDisplayAfter": "FillOrSignField",
      "FinishWorkstepOnOpen": true,
      "AutoFinishAfterRequiredTasksDone": true,
      "GuidingBehaviorOnFinishedTask": "NoMove",
      "SkipThankYouDialog": true,
      "NativeAppsUrlScheme": "string",
      "DocumentViewingMode": "EndlessPaperAllDocuments",
      "ThumbnailMode": "ShowAllPages",
```

```
    "ShowTopBar": true,
    "DisplayRejectButtonInTopBar": true,
    "MultipleSignatureTypesAndBatchSigningSettings": {
      "IsUseBatchSigningCheckedByDefault": true,
      "IsRememberSignatureTypeCheckedByDefault": true,
      "IsRememberBatchSigningDecisionCheckedByDefault": true,
      "SkipMultipleSignatureTypesAndBatchSigningDialogIfBatchSigningPossible": true
    },
    "VisibleAreaOptions": {
      "AllowedDomain": "string",
      "Enabled": true
    },
    "ShowStartGuidingHint": true,
    "ShowStatusBar": true,
    "ShowZoomButtons": true,
    "ShowNoGeolocationWarning": true,
    "AutoStartGuiding": true,
    "ShowPageGap": true,
    "ShowPageNavigationButtons": true,
    "ShowFinishPossibleHint": true,
    "SkipRejectConfirmDialog": true,
    "BatchSigningType": "Basic",
    "BatchSigningDisableNextButtonUntilDialogScrolledToBottom": true
  },
  "SignatureConfigurations": [
    {
      "SpcId": "string",
      "PdfSignatureProperties": {
        "PdfAConformant": true,
        "PAdESPart4Compliant": true,
        "IncludeSigningCertificateChain": true,
        "SigningCertificateRevocationInformationIncludeMode": "DoNotInclude",
        "SignatureTimestampData": {
          "Uri": "string",
          "Username": "string",
          "Password": "string",
          "SignatureHashAlgorithm": "Sha1",
          "AuthenticationCertifiateDescriptor": {
            "Identifier": "string",
            "Type": "string"
          }
        },
        "EnableEutlVerification": true,
        "EnableValidateSigningCertificateName": true,
        "SigningCertificateNameRegex": "string"
      },
      "PdfSignatureCryptographicData": {
        "SignatureHashAlgorithm": "Sha1",
        "SigningCertificateDescriptor": {
          "Identifier": "string",
          "Type": "Sha1Thumbprint",
          "Csp": "Default"
        }
      },
      "CertificateFilter": {
        "KeyUsages": [
          "string"
        ],
        "ThumbPrints": [
          "string"
        ],
        "RootThumbPrints": [
          "string"
        ]
      }
    }
  ],
  "SigStringParsingConfiguration": {
    "SigStringsForParsings": [
      {
        "StartPattern": "string",
```

```
          "EndPattern": "string",
          "ClearSigString": true,
          "SearchEntireWordOnly": true
        }
      ]
    },
    "GeneralPolicies": {
      "AllowSaveDocument": true,
      "AllowSaveAuditTrail": true,
      "AllowRotatingPages": true,
      "AllowAppendFileToWorkstep": true,
      "AllowAppendTaskToWorkstep": true,
      "AllowEmailDocument": true,
      "AllowPrintDocument": true,
      "AllowFinishWorkstep": true,
      "AllowRejectWorkstep": true,
      "AllowRejectWorkstepDelegation": true,
      "AllowUndoLastAction": true,
      "AllowColorizePdfForms": true,
      "AllowAdhocPdfAttachments": true,
      "AllowAdhocSignatures": true,
      "AllowAdhocStampings": true,
      "AllowAdhocFreeHandAnnotations": true,
      "AllowAdhocTypewriterAnnotations": true,
      "AllowAdhocPictureAnnotations": true,
      "AllowAdhocPdfPageAppending": true,
      "AllowReloadOfFinishedWorkstep": true
    },
    "FinalizeActions": {

    },
    "TransactionCodeConfigurations": [
      {
        "Id": "string",
        "HashAlgorithmIdentifier": "Sha1",
        "Texts": [
          {
            "Language": "string",
            "Value": "string"
          }
        ]
      }
    ]
  },
  "PrepareSendEnvelopeStepsDescriptor": {
    "ClearFieldMarkupString": true
  }
}
```

```
<AdhocWorkstepConfiguration>
        <!-- not relevant-->
        <WorkstepLabel>workstepLabel</WorkstepLabel>
        <!-- not relevant-->
        <SmallTextZoomFactorPercent>100</SmallTextZoomFactorPercent>
        <!-- not relevant-->
        <WorkstepTimeToLiveInMinutes>0</WorkstepTimeToLiveInMinutes>
        <!--Configure the actions done by the server and the by the clients when the workstep is finished.-->
        <FinishAction>
                <!-- not relevant-->
                <ServerAction callSynchronous="0"></ServerAction>
                <!--A client action specifies the redirect, when a recipient clicks on finish.-->
                <ClientAction clientName="SIGNificant SignAnywhere" closeApp="0"
RemoveDocumentFromRecentDocumentList="0" CallClientActionOnlyAfterSuccessfulSync="1">https://www.significant.
com</ClientAction>
        </FinishAction>

        <!--Configure the adhoc workstep creation-->
        <NoSequenceEnforced>0</NoSequenceEnforced>
```

```
        <!-- Define default properties of signature fields / tasks-->
        <SigTemplate>
                <!--The elements width in points-->
                <width>50.5</width>
                <!--The elements height in points-->
                <height>100.5</height>
                <!--Parameter defining the signature type. Possible values: 'BiometricSignature',
'LocalCertificate', 'Picture', 'TransactionCode', 'TransactionCodeAndBiometricSignature',
'TransactionCodeAndLocalCertificate', 'TransactionCodeBiometricSignatureAndLocalCertificate' and
'BiometricSignature_and_LocalCertificate'. -->
                <param name="sigType">Picture</param>
                <!--Parameter that refines sigType "Picture": a list, seperated by "," from these values:
Draw2Sign,Type2Sign,Click2Sign -->
                <param name="allowedCapturingMethods">Draw2Sign,Type2Sign</param>

    </SigTemplate>
        <!--Configuration for parsing the form fields. Possible values: '1' parse the form fields, '0' do
not parse form fields-->
        <!--Attribute 'mapRequiredFieldsToRequiredTask': set the form filling task required when some of the
fields are required. Possible values: '1' required forms lead to required tasks, '0' required fields do not
enforce the task to be required-->
        <!--Attribute 'formsGrouping': Specify how the parsed forms should be grouped into tasks. Possible
values: 'PerPage' all forms on one page are grouped to one forms group, 'PerDocument' all forms of one
document are grouped to one forms group, 'PerEnvelope' all forms of all documents inside the envelope are
grouped to one forms group-->
        <ParseFormFields mapRequiredFieldsToRequiredTask="0" formsGrouping="PerDocument">1</ParseFormFields>
        <!--If the workstep is not generated by hand but automatically generated by the Workstep Controller
AdhocPolicies are specified-->
        <AdhocPolicies>
                <!-- not relevant-->
                <AllowModificationsAfterSignature>1</AllowModificationsAfterSignature>
        </AdhocPolicies>
        <!--Configurate the signatures for this workstep. One default configuration has to be defined. The
default configuration is used for flatten signatures, adhoc signatures and signature fields which do not
reference a special signature plugin configuration. The default configuration does not contain the attribute
'spcId'. If the attribute 'spcId' is defined the signature plugin configuration does only apply to signature
fields referencing the configuration by specifiying <param name="spcId">id</param>.-->
        <signaturePluginConfiguration>
                <!--Configurate the signature properties-->
                <PdfSignatureProperties_V1>
                        <!--Should the signatures be pdfA conformant. Note this setting does not convert a
document into pdfA, it only keeps it pdfA conformant if it already is. Possible values: '1' sign pdfA
conformant - in this case the file size will be bigger than without pdfA, '0' do not sign pdfA conformant.-->
                        <PdfAConformant>0</PdfAConformant>
                        <!--Defines if the signature should be PAdES part 4 compliant. Possible values: '1'
sign the document PAdES part 4 compliant, '0' sign the document with standard pdf signature. Default value:
'0'-->
                        <PAdESPart4Compliant>1</PAdESPart4Compliant>
                        <!--Defines if the certificate chain for the signing certificate should be embedded
into the signature. Possible values: '1' include the certificate chain, '0' do not include the certificate
chain. Default value: '0'-->
                        <IncludeSigningCertificateChain>1</IncludeSigningCertificateChain>
                        <!--Defines if and how the revocation information for the signing certificate chain
should be embedded. Possible values: 'DoNotInclude' no revocation information is included, 'Include' the
revocation information has to be included, if not possible the signature throws an exception, 'TryToInclude'
if the revocation information can be fetched, it should be included, if not the signature is done without
revocation information. Information about the signatures where the revocation information could not be
included is saved into the WorkstepStatus, 'CheckRevocationIncludeOcsp' the revocation information has to be
included when it is an OCSP, if checking of the revocation (OCSP or CRL) fails an exception is thrown.
Information about the signatures where the revocation information could not be included is saved into the
WorkstepStatus-->
                        <SigningCertificateRevocationInformationIncludeMode>Include<
/SigningCertificateRevocationInformationIncludeMode>
                </PdfSignatureProperties_V1>
                <!--Configurate the cryptographic data-->
                <PdfSignatureCryptographicData_V1>
                        <!--The hash algorithm used for the signatures. Possible values: 'Sha1', 'Sha256',
'Sha512'-->
                        <SignatureHashAlgorithm>Sha256</SignatureHashAlgorithm>
                        <!--The description of the signing certificate. More than one
SigningCertificateDescriptor can be defined by adding this node more than once. If more
```

```
SigningCertificateDescriptors are present, these configurations are used as backup if the selected
SigningCertificateDescriptor is not working. For example if no revocation information can be retrieved
although it should be included into the signature.-->
                        <SigningCertificateDescriptor>
                                <!--The certificates identifier-->
                                <Identifier>3b777446a35fca027cbed5f69e24995945a611cb</Identifier>
                                <!--The certificate identifier type. Possible values: 'Subject',
'Sha1Thumbprint'-->
                                <Type>Sha1Thumbprint</Type>
                                <!--The cryptographic service provider to locate the certificate. Possible
values: 'default' uses the servers certificate store, 'custom' uses the custom signature action-->
                                <Csp>Default</Csp>
                        </SigningCertificateDescriptor>
                </PdfSignatureCryptographicData_V1>
        </signaturePluginConfiguration>

        <!--Configure the signature string parsing pattern: Text in the document will be parsed for this
pattern and if found, a signature task is generated. -->
        <SigStringParsingConfiguration>
                <!--Defines a signature string to parse. Tag can be present more than once-->
                <SigStringsForParsing>
                        <!--The start pattern of the signature string if it has a start and end pattern.
Otherwise the whole word to parse-->
                        <StartPattern>`sig</StartPattern>
                        <!--End pattern if needed, otherwise empty-->
                        <EndPattern>`</EndPattern>
                        <!--Define if the signature strings should be cleared from the document. Possible
values: '1' remove the signature strings from the document, '0' do not change the document-->
                        <ClearSigString>1</ClearSigString>
                        <!--Define if only the entire word should be searched. For example if start pattern
is 'signature' only 'signature' but not 'signaturepad' is found. This option does only effect signature
string without end patterns. Possible values: '1' search only the entire word, '0' search words containing
the pattern as well.-->
                        <SearchEntireWordOnly>1</SearchEntireWordOnly>
                </SigStringsForParsing>
        </SigStringParsingConfiguration>
        <!--Defines general policies for this workstep-->
        <GeneralPolicies>
                <!--Is the client allowed to save the workstep document-->
                <AllowSaveDocument>1</AllowSaveDocument>
                <!--Is the client allowed to save the audittrail document-->
                <AllowSaveAuditTrail>1</AllowSaveAuditTrail>

                <!-- not relevant-->
                <AllowFinishWorkstep>1</AllowFinishWorkstep>
                <!--Is the client allowed to reject the workstep-->
                <AllowRejectWorkstep>1</AllowRejectWorkstep>

                <AllowAdhocPdfAttachments>1</AllowAdhocPdfAttachments>
        </GeneralPolicies>

        <ViewerPreferences>
                <ShowVersionNumber>1</ShowVersionNumber>
                <EnableThumbnailDisplay>1</EnableThumbnailDisplay>
                <EnableWarningPopupOnLeave>1</EnableWarningPopupOnLeave>
                <WarningPopupDisplayAfter>Always</WarningPopupDisplayAfter>
                <GuidingBehavior>GuideRequiredAndOptionalTasks</GuidingBehavior>

        </ViewerPreferences>

</AdhocWorkstepConfiguration>
```

Send envelope

For sending an envelope we are using the following endpoints:

REST	SOAP

https://demo.esignanywhere.net/Api/v4.0/envelope/send	SendEnvelope_v1

This call requires the authorization, the file id(s) and the envelope configuration. The envelope configuration defines the envelope and the steps in the workflow. For each signing step you need to define a workstep configuration, which defines what the signer has to do in his/her singing task.

You can finde a sample configuration for this call in the next section:

```
{

  "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>
               <authentications>
                  <authentication>
                     <method>Pin</method>
                     <parameter>1234</parameter>
                  </authentication>
               </authentications>
            </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>
      <step>
         <emailBodyExtra />
```

```
         <orderIndex>2</orderIndex>
         <recipientType>CC</recipientType>
         <recipients>
            <recipient>
               <languageCode>en</languageCode>
               <eMail>##COPYRECEIVER-MAIL##</eMail>
               <firstName>Charly</firstName>
               <lastName>Randomname</lastName>
            </recipient>
         </recipients>
      </step>
   </steps>
</envelope>
```

This is the simplest form of a workstep configuration. With the Developer Feature of eSAW 2.6+ you can download the envelope JSON or the envelope XML including the workstep configuration of any envelope designed in eSAW UI. The second signer only receives a copy, so he/she does not need to have a workstep configuration.

If the creation of the envelope was successful, you will get the envelope id as response.
In REST:

```
{ "EnvelopeId": "56db6133-xxxx-xxxx-xxxx-xxxxxxxxxxxx" }
```

```
<apiResult version="2.2.458.6616">
  <baseResult>ok</baseResult>
  <okInfo>
    <envelopeId>56db6133-xxxx-xxxx-xxxx-xxxxxxxxxxxx</envelopeId>
  </okInfo>
</apiResult>
```

After the successful creation of the envelope, it is sent to the first recipient. The envelope id is used for managing the envelopes (send reminder, cancel delete, reject envelope,…).

The recipient language must be supported by your organization (Settings->Localization).

Find the envelope

After sending the envelope you can use the following api call to search for the envelope:

REST	SOAP
https://demo.esignanywhere.net/Api/v4.0/envelope/find	FindEnvelopes_v1 or FindEnvelopes_v2

In the next section you can find a sample configuration which you need for the api call:

```
{
  "StartDate": "2020-04-17T13:09:50.089Z",
  "EndDate": "2020-04-17T13:09:50.089Z",
  "SearchText": "string",
  "Status": "Draft",
  "InStatusSinceDays": 0,
  "Senders": [
    "string"
  ],
  "Signers": [
    "string"
  ],
  "Recipients": [
    "string"
  ],
  "WaitingForRecipient": "string",
  "Bulk": "string"
}
```

```
<findEnvelopesDescriptor>
<status>Draft|Started|InProgress|Canceled|Completed|Expired|Rejected|Template|ActionRequired|
WaitingForOthers|ExpiringSoon|Active</status>
        <inStatusSinceDays>30</inStatusSinceDays>
        <sentAfterDate>2017-02-08T12:18:37.2415657Z</sentAfterDate>
        <sentBeforeDate>2017-05-10T11:18:37.2415657Z</sentBeforeDate>
        <searchText/>
        <hasSender>
                <eMail/>
        </hasSender>
        <hasSigner>
                <eMail/>
        </hasSigner>
        <hasRecipient>
                <eMail/>
        </hasRecipient>
        <waitingForRecipient/>
</findEnvelopesDescriptor>
```

You can search for the following status:

REST	SOAP
Draft	Draft
Canceled	Canceled
Completed	Completed
Expired	Expired
Rejected	Rejected
Template	Template
Active	Active
ExpiringSoon	ExpiringSoon
WaitingForOthers	WaitingForOthers
ActionRequired	
	Started
	InProgress

If you search for templates you will get the templateId which you need for the following api calls:

REST	SOAP
https://demo.esignanywhere.net/Api/v4.0/envelope/sendFromTemplate	CreateDraftFromTemplate_v1
https://demo.esignanywhere.net/Api/v4.0/envelope/createFromTemplate	SendEnvelopeFromTemplate_v1
https://demo.esignanywhere.net/Api//v4.0/envelope/{templateId}/copyFromTemplate	CopyDocumentFromTemplate_v1

For more information about the process of finding the template and send an envelope from the template please also have a look at the template use case.

If you have not finished the envelope then search for “Active” envelopes, if the envelope is already finished please search for “Completed” envelopes to find the envelope you have sent in the api call above.
As response from this api call you will get a list of all envelopes which fulfill the status.

Envelope Status & Callback

The envelope id can be used any time to receive the envelope status. Therefore, you simply call:

REST	SOAP
https://demo.esignanywhere.net/Api/v4.0/envelope/##envelopeId##	GetEnvelopeById_v1

You can find a sample response for this call in the next section:


```
{
    "Status": "InProgress",
    "SendDate": "2020-03-17T08:24:47.62Z",
    "ExpirationDate": "2020-04-14T08:24:47.62",
    "Bulks": [
        {
            "Status": "InProgress",
            "Email": "",
            "LogDocumentId": "",
            "FinishedDocuments": [],
            "Steps": [
                {
                    "Id": "c211d550-25c5-4f61-94f9-9eb5695685fd",
                    "FirstName": "##NAME##",
                    "LastName": "##NAME##",
                    "OrderIndex": 1,
                    "Email": "##EMAIL##",
                    "LanguageCode": "en",
                    "Status": "NotSigned",
                    "StatusReason": "",
                    "RecipientType": "Signer",
                    "OpenedDate": "2020-03-17T08:24:59.383Z",
                    "WorkstepRedirectionUrl": "http://beta4.testlab.xyzmo.com/workstepredirector/sign?
identifier=##RedirectionURL##",
                    "AllowAccessFinishedWorkstep": false,
                    "Authentication": [
                        {
                            "Method": "Pin",
                            "Parameter": "1234",
                            "Filters": []
                        }
                    ],
                    "IsParallel": false,
                    "WorkstepConfiguration": {
                        "WorkstepLabel": "test",
                        "SmallTextZoomFactorPercent": 100,
                        "FinishAction": {
                            "ServerActions": [
                                {
                                    "CallSynchronous": false,
                                    "Action": "http://beta4.testlab.xyzmo.com//workstepredirector
/setfinishaction?wid=##WorkstepId##"
                                }
                            ],
                            "ClientActions": []
                        },
```

```
"ReceiverInformation": {
    "UserInformation": {
        "FirstName": "##NAME##",
        "LastName": "##NAME##",
        "EMail": "##EMAIL##"
    },
    "TransactionCodePushPluginData": []
},
"SenderInformation": {
    "UserInformation": {
        "FirstName": "##NAME##",
        "LastName": "##NAME##",
        "EMail": "##EMAIL##"
    }
},
"TransactionCodeConfigurations": [
    {
        "Id": "otpSignatureSmsText",
        "HashAlgorithmIdentifier": "Sha256",
        "Texts": [

        ]
    }
],
"SignatureConfigurations": [
    {
        "PdfSignatureProperties": {
            "PdfAConformant": false,
            "PAdESPart4Compliant": false,
            "IncludeSigningCertificateChain": false,
            "SigningCertificateRevocationInformationIncludeMode": "DoNotInclude"
        },
        "PdfSignatureCryptographicData": {
            "SignatureHashAlgorithm": "Sha256",
            "SigningCertificateDescriptor": {
                "Identifier": "14527A6BCFA8B4D7D0183FCA6B735B1C246D14AE",
                "Type": "Sha1Thumbprint",
                "Csp": "Default"
            }
        }
    },
    {
        "SpcId": "timestampSigningId",
        "PdfSignatureProperties": {
            "PdfAConformant": false,
            "PAdESPart4Compliant": false,
            "IncludeSigningCertificateChain": false,
            "SigningCertificateRevocationInformationIncludeMode": "DoNotInclude",
            "SignatureTimestampData": {
                "Uri": "https://timestamp.test.namirialtsp.com",
                "Username": "xyzmo",
                "Password": "xyzmo",
                "SignatureHashAlgorithm": "Sha256"
            }
        },
        "PdfSignatureCryptographicData": {
            "SignatureHashAlgorithm": "Sha256",
            "SigningCertificateDescriptor": {
                "Identifier": "14527A6BCFA8B4D7D0183FCA6B735B1C246D14AE",
                "Type": "Sha1Thumbprint",
                "Csp": "Default"
            }
        }
    },
    {
        "SpcId": "padesSigningId",
        "PdfSignatureProperties": {
            "PdfAConformant": false,
            "PAdESPart4Compliant": true,
            "IncludeSigningCertificateChain": false,
            "SigningCertificateRevocationInformationIncludeMode": "IncludeDss"
```

```
        },
        "PdfSignatureCryptographicData": {
            "SignatureHashAlgorithm": "Sha256",
            "SigningCertificateDescriptor": {
                "Identifier": "14527A6BCFA8B4D7D0183FCA6B735B1C246D14AE",
                "Type": "Sha1Thumbprint",
                "Csp": "Default"
            }
        }
    },
    {
        "SpcId": "padesTimestampSigningId",
        "PdfSignatureProperties": {
            "PdfAConformant": false,
            "PAdESPart4Compliant": true,
            "IncludeSigningCertificateChain": false,
            "SigningCertificateRevocationInformationIncludeMode": "IncludeDss",
            "SignatureTimestampData": {
                "Uri": "https://timestamp.test.namirialtsp.com",
                "Username": "xyzmo",
                "Password": "xyzmo",
                "SignatureHashAlgorithm": "Sha256"
            }
        },
        "PdfSignatureCryptographicData": {
            "SignatureHashAlgorithm": "Sha256",
            "SigningCertificateDescriptor": {
                "Identifier": "14527A6BCFA8B4D7D0183FCA6B735B1C246D14AE",
                "Type": "Sha1Thumbprint",
                "Csp": "Default"
            }
        }
    },
    {
        "SpcId": "padesRemoteCertificateSigningId",
        "PdfSignatureProperties": {
            "PdfAConformant": false,
            "PAdESPart4Compliant": true,
            "IncludeSigningCertificateChain": false,
            "SigningCertificateRevocationInformationIncludeMode": "IncludeDss"
        },
        "PdfSignatureCryptographicData": {
            "SignatureHashAlgorithm": "Sha256",
            "SigningCertificateDescriptor": {
                "Identifier": "14527A6BCFA8B4D7D0183FCA6B735B1C246D14AE",
                "Type": "Sha1Thumbprint",
                "Csp": "Default"
            }
        }
    },
    {
        "SpcId": "padesRemoteCertificateTimestampSigningId",
        "PdfSignatureProperties": {
            "PdfAConformant": false,
            "PAdESPart4Compliant": true,
            "IncludeSigningCertificateChain": false,
            "SigningCertificateRevocationInformationIncludeMode": "IncludeDss",
            "SignatureTimestampData": {
                "Uri": "https://timestamp.test.namirialtsp.com",
                "Username": "xyzmo",
                "Password": "xyzmo",
                "SignatureHashAlgorithm": "Sha256"
            }
        },
        "PdfSignatureCryptographicData": {
            "SignatureHashAlgorithm": "Sha256",
            "SigningCertificateDescriptor": {
                "Identifier": "14527A6BCFA8B4D7D0183FCA6B735B1C246D14AE",
                "Type": "Sha1Thumbprint",
                "Csp": "Default"
            }
```

```
        }
    },
    {
        "SpcId": "automaticSigningId",
        "PdfSignatureProperties": {
            "PdfAConformant": false,
            "PAdESPart4Compliant": false,
            "IncludeSigningCertificateChain": false,
            "SigningCertificateRevocationInformationIncludeMode": "DoNotInclude"
        },
        "PdfSignatureCryptographicData": {
            "SignatureHashAlgorithm": "Sha256",
            "SigningCertificateDescriptor": {
                "Identifier": "14527A6BCFA8B4D7D0183FCA6B735B1C246D14AE",
                "Type": "Sha1Thumbprint",
                "Csp": "Default"
            }
        }
    },
    {
        "SpcId": "automaticTimestampSigningId",
        "PdfSignatureProperties": {
            "PdfAConformant": false,
            "PAdESPart4Compliant": false,
            "IncludeSigningCertificateChain": false,
            "SigningCertificateRevocationInformationIncludeMode": "DoNotInclude",
            "SignatureTimestampData": {
                "Uri": "https://timestamp.test.namirialtsp.com",
                "Username": "xyzmo",
                "Password": "xyzmo",
                "SignatureHashAlgorithm": "Sha256"
            }
        },
        "PdfSignatureCryptographicData": {
            "SignatureHashAlgorithm": "Sha256",
            "SigningCertificateDescriptor": {
                "Identifier": "14527A6BCFA8B4D7D0183FCA6B735B1C246D14AE",
                "Type": "Sha1Thumbprint",
                "Csp": "Default"
            }
        }
    },
    {
        "SpcId": "swissComSigningId",
        "PdfSignatureProperties": {
            "PdfAConformant": false,
            "PAdESPart4Compliant": true,
            "IncludeSigningCertificateChain": false,
            "SigningCertificateRevocationInformationIncludeMode": "IncludeDss"
        },
        "PdfSignatureCryptographicData": {
            "SignatureHashAlgorithm": "Sha256",
            "SigningCertificateDescriptor": {
                "Identifier": "14527A6BCFA8B4D7D0183FCA6B735B1C246D14AE",
                "Type": "Sha1Thumbprint",
                "Csp": "Default"
            }
        }
    },
    {
        "SpcId": "tLevelId",
        "PdfSignatureProperties": {
            "PdfAConformant": false,
            "PAdESPart4Compliant": true,
            "IncludeSigningCertificateChain": true,
            "SigningCertificateRevocationInformationIncludeMode": "DoNotInclude",
            "SignatureTimestampData": {
                "Uri": "https://timestamp.test.namirialtsp.com",
                "Username": "xyzmo",
                "Password": "xyzmo",
                "SignatureHashAlgorithm": "Sha256"
```

```
                                }
                            },
                            "PdfSignatureCryptographicData": {
                                "SignatureHashAlgorithm": "Sha256",
                                "SigningCertificateDescriptor": {
                                    "Identifier": "14527a6bcfa8b4d7d0183fca6b735b1c246d14ae",
                                    "Type": "Sha1Thumbprint",
                                    "Csp": "Default"
                                }
                            }
                        }
                    ],
                    "ViewerPreferences": {
                        "FinishWorkstepOnOpen": false,
                        "VisibleAreaOptions": {
                            "AllowedDomain": "*",
                            "Enabled": false
                        }
                    },
                    "ResourceUris": {
                        "SignatureImagesUri": "http://beta4.testlab.xyzmo.com//Resource/SignatureImages/?
link=1WStxYx9aVh2bKgPnnOoA2x56aVWCf/oMP6GirOkH5eOattOs60C8lVAATUnqhUHa"
                    },
                    "Policy": {
                        "GeneralPolicies": {
                            "AllowSaveDocument": true,
                            "AllowSaveAuditTrail": true,
                            "AllowRotatingPages": false,
                            "AllowAppendFileToWorkstep": false,
                            "AllowAppendTaskToWorkstep": false,
                            "AllowEmailDocument": true,
                            "AllowPrintDocument": true,
                            "AllowFinishWorkstep": true,
                            "AllowRejectWorkstep": true,
                            "AllowRejectWorkstepDelegation": false,
                            "AllowUndoLastAction": false,
                            "AllowColorizePdfForms": 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": []
                    }
                }
            },
            {
                "Id": "ff4b740e-f0a8-4e69-ad32-54cae6104993",
                "FirstName": "##NAME##",
                "LastName": "##NAME##",
                "OrderIndex": 2,
                "Email": "##EMAIL##",
                "LanguageCode": "en",
                "Status": "NotSigned",
                "StatusReason": "",
                "RecipientType": "Cc",
                "WorkstepRedirectionUrl": "",
                "AllowAccessFinishedWorkstep": false,
                "IsParallel": false,
                "WorkstepConfiguration": {
                    "SmallTextZoomFactorPercent": 100,
                    "ViewerPreferences": {},
                    "Policy": {
                        "GeneralPolicies": {
                            "AllowRotatingPages": true,
                            "AllowFinishWorkstep": true,
                            "AllowUndoLastAction": true
                        },
                        "WorkstepTasks": {
                            "PictureAnnotationMinResolution": 0,
                            "PictureAnnotationMaxResolution": 0,
                            "PictureAnnotationColorDepth": "Color16M",
                            "SequenceMode": "NoSequenceEnforced",
                            "PositionUnits": "PdfUnits",
                            "ReferenceCorner": "Lower_Left",
                            "Tasks": []
                        }
                    }
                }
            }
        ]
    }
],
"Documents": [
    {
        "PageSizesInPoints": [
            {
                "Height": 792.0,
                "Width": 612.0
            }
        ],
        "DocRefNumber": 1,
        "FileName": "test.pdf",
        "FormFields": []
    }
],
"Id": "62ce2b28-deb9-40a0-9656-88609353494b",
"Bulk": "",
"BasicOptions": {
    "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#",
    "EnableReminders": true,
```

```
        "FirstReminderDayAmount": 5,
        "RecurrentReminderDayAmount": 3,
        "BeforeExpirationDayAmount": 3
    }
}
```

```
<apiResult version="2.2.458.6616">
        <baseResult>ok</baseResult>
        <okInfo>
                <envelopeStatus>
                        <id>13ad0518-xxxx-xxxx-xxxx-xxxxxxxxxxx</id>
                        <name>eSignAnyWhere Tutorial</name>
                        <status>Completed</status>
                        <sendDate>2017-03-15T12:13:13.453Z</sendDate>
                        <expirationDate>2017-03-17T12:13:13.453Z</expirationDate>
                        <bulkRecipients>
                                <bulkRecipient eMail="">
                                        <status>Completed</status>
                                        <recipients>
                                                <recipient>
                                                        <orderIndex>1</orderIndex>
                                                        <eMail>first.signer@email.com</eMail>
                                                        <status>Signed</status>
                                                        <signedDate>2017-03-15T12:13:42.717Z</signedDate>
                                                        <recipientType>Signer</recipientType>
                                                        <workstepRedirectionUrl/>
                                                </recipient>
                                                <recipient>
                                                        <orderIndex>2</orderIndex>
                                                        <eMail>cc@email.com</eMail>
                                                        <status>NotSigned</status>
                                                        <signedDate>2017-03-15T12:13:43.183Z</signedDate>
                                                        <recipientType>Cc</recipientType>
                                                        <workstepRedirectionUrl/>
                                                </recipient>
                                        </recipients>
                                        <completedDocuments>
                                                <logDocumentId>52cba71e-xxxx-xxxx-xxxx-xxxxxxxxxxx<
/logDocumentId>
                                                <completedDocument>
                                                        <documentId>77851da5-xxxx-xxxx-xxxx-xxxxxxxxxxx<
/documentId>
                                                        <fileName>eSignAnyWhere_Tutorial.pdf</fileName>
                                                        <fields/>
                                                </completedDocument>
                                        </completedDocuments>
                                </bulkRecipient>
                        </bulkRecipients>
                </envelopeStatus>
        </okInfo>
</apiResult>
```

Callback

The configured callback URL is used to notify your system if the state of an envelope or workstep is changing.
So you can track the status of envelopes without polling the eSignAnyWhere server regulary.

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!

Download a finished document

To download a finished document you simply call the function:

REST	SOAP
https://demo.esignanywhere.net/Api/v4.0/envelope/downloadCompletedDocument/##documentId##	DownloadCompletedDocument_v1

You have to use the envelope id and the document id to download the file from the server (Attention: just documents of finished envelopes can be downloaded and the document id is different from the uploaded-file-id!)
You can receive the document id from the response of the getEnvelope call.

Download respoonse in REST would be for example a 200 OK status.

```
<apiResult version="2.2.458.6616">
  <baseResult>ok</baseResult>
  <okInfo>
    <file>
      <name>eSignAnyWhere_Tutorial.pdf</name>
      <data>##BASE64_FILE##</data>
    </file>
  </okInfo>
</apiResult>
```

Additional Information

Depending on the signature method used, it might be necessary or recommended to store additional documents beside the signed document, to ensure having the necessary evidence. For advanced electronic signatures, you might have to store also the audit trail document; and for Namirial Disposable Certificate, also the Certificate Request Form by which the signer requested issuance of the QEC. Those documents can be downloaded using the same method, just with the different file IDs.