

Migrate REST API clients from v5 to v6

The changes between REST API v5 and v6 require bigger changes of the API client implementations, as this version switch addressed many known disadvantages of the v3-v5 API versions. This migration guide should give an overview on how to migrate from API v5 to API v6. As there are no big changes between v4 and v5, this guide will also be helpful for integrations that are still on API v4. For those for projects that are still using API v3 or v4, we recommend to migrate to v5 first following the v3 to v5 guide, and then migrate in a 2nd step from v5 to v6 using this guide.



API v5 will remain part of the product beyond the announced removal of v3, while REST API v6 is already available. For existing integrations, staying on API v5 is ok as long as no new features of v6 are required, but our recommendation is to start early to migrate to API v6 even in consideration of the effort, to be prepared for the situation when you need a new functionality that is available on API v6 only.

For new customers, and also for new projects at existing customers, we strongly recommend to start with REST API v6.



Important information: If you have been utilizing the API v5 to create envelopes and are planning to retrieve these envelopes using API calls from v6, it is crucial to be aware of potential compatibility issues.

Due to changes and updates between API versions, certain calls made from v6 may result in errors related to invalid API version ("InvalidEnvelopeApiVersion") when attempting to retrieve envelopes created with API v5. A possible solution is to continue using the v5 calls to retrieve the envelopes created via v5 if one of the following API calls is needed to retrieve the envelope/template or draft. See following affected api call:

- Draft/GetElements
- Envelope/Get
- Envelope/GetConfiguration
- Envelope/GetElements
- Template/GetElements



Currently Non-Supported Functionality

Please keep using API v5 if you are using one of this features listed below:

- The mode SequenceOnlyRequiredTasks was removed from APIv6. A sender can still decide between NoSequenceEnforced and SequenceEnforced, and based on our experience SequenceEnforced can be used for those scenarios where SequenceOnlyRequiredTasks was used in some cases. We did this change to reduce complexity for integrators, and may decide in a future release if SequenceOnlyRequiredTasks needs to be added again.
- Support for Swisscom Automatic Signatures / Swisscom Organization Certificates will be added in a later release.
- Support for document links to bookmarks (i.e. within one document, or from one document to another). Only hyperlink areas to WWW resources are supported.
- Some functionality formerly possible to be set via eSAW API in undocumented data structures, e.g. parameters just interpreted in SIGNificant app, Kiosk etc but not available in SAW Viewer, are not contained in APIv6 models. Please check for such functionality, in case you used it before, already before starting to migrate to APIv6. This functionality will be added to APIv6 in a later product version, if it is considered as important product functionality. In case such functionality will be added to APIv6, then it will be available also in eSAW envelope creator and SAW Viewer.
- User Management - there is currently no /user/* endpoint in API v6. For user management, keep using API v5, until we add new user management methods to API v6 in a future product version
- OverrideHolderInCaseOfMismatch was removed from APIv6
- ViewerPreferences were removed from APIv6. Only configurable in the customization.zip: [SignAnyWhere Viewer - Customization#ViewerPreferences](#)
- SignatureRenderingLayout was removed from APIv6. It is therefore not possible to add the layout to the signature field.
- Signing certification via thumbprint was removed from APIv6. But please note that it is possible to add a custom sealing certificate via UI as well as via API - [Custom Sealing Certificate](#)
- The parameter "AttachSignedDocumentsToEnvelopeLog" was removed from APIv6 and therefore it is not possible to add signed documents to audit trail.

Already implemented:

See also [eSignAnyWhere Release News](#) for more detail.

- GeneralPolicies were added to the UI as well as for the API v6 with version 23.49
- Support for modifying SMS-OTP message (language and content) was added with version 23.49

- [Conceptual Changes](#)
 - [Global Naming Changes](#)
 - [Envelope Status Enumeration](#)
 - [Disposable Certificate Enumeration](#)
 - [Identification type](#)

- Document type
 - Complexity Reduction
 - Structural Changes for Bulk Sending via API & for managing parent envelopes resulting from bulk-sending
 - Complexity reduction for envelope/find call
 - Changes on Method Level
 - Removed and Deprecated Methods
 - Changed and added Methods
 - Authorization
 - Changed method
 - Draft
 - Changed method
 - Added method
 - Envelope
 - Changed method
 - Envelope information
 - Download files referenced by the envelope
 - Restarting of an already expired envelope
 - Parsing a file to detect Advanced Document Tags or Form Fields contained in the PDF
 - Sending an Envelope
 - License
 - Organization
 - Recipient
 - SspFile
 - Team
 - Templates
 - System
- Changes in the Models
 - Draft Models
 - Envelope Models
 - Overview
 - Comparison
 - Signature Configuration
- New Functionalities in API v6
 - Define Workstep Parameters on Drafts
 - Work with Recipient Placeholders

Conceptual Changes

- The former option to use insecure organizationTokens (after they had been explicitly enabled on organization level) has been removed.
- Authentication
- All namings in the API, on method level and inside the models, have been reviewed (and most of them have been changed) because we think consistent and simple wording helps to speed up the learning curve of the API users
 - In API v6, we are calling elements as it is common in workflow context
 - We improved to get a consistent naming within the API
 - In consideration of clean-code principles, we tried to use "positive-wording" for boolean properties as much as possible
 - Where applicable, the naming of elements in the API is in sync with the naming in the eSignAnyWhere Web UI (in English language)
- We reviewed all default-values because with reasonable defaulting many elements don't need to be specified by the user of the API
- We removed historic relicts that come from technical architecture (e.g. „workstepConfiguration“) because API users shouldn't burn time by understanding internal concepts of our platform
- The API is no longer using „abstract types“ and the DiscriminatorType element in the API, but include the full structure in the method's model directly now because we wanted to focus on compatibility with various REST API code generators
- We are using only the verbs GET and POST; and strictly avoid using PUT, PATCH, DELETE and other HTTP verbs again for simplicity, but also to avoid issues with some coding platforms and infrastructure configurations where customers reported issues in the past.
- Clear rule for parameters in API calls:
 - in GET calls, the parameters are part of the URI path.
 - in POST calls, there are NO parameters in the URI path; all parameters are in the HTTP request body.
- Clear rules on datatype formats, e.g.:
 - Dates consistently following RFC 3339 Internet Date/Time Format "full-date" type
e.g. 2022-11-25
 - Timestamps consistently following RFC 3339 Internet Date/Time Format "date-time" type
e.g. 2022-11-25T09:11:12Z. Mind that a timezone information is mandatory, and Z can be used for "Zulu Time" (i.e. UTC+0)
 - Color values always as an html colorcode string (#rrggbb), e.g. #ff0000 for red

For the more detailed view on changes, where we are referring to old naming and the new replacement, we use following notation in this document:

- Old terminology is formatted in ***bold-italic***
- New terminology is formatted in **bold**.

Global Naming Changes

We take into account that the words "**whitelist**" and "**blacklist**" can be perceived as offensive and hurtful words. It has never been our intention to imply discriminatory or other-valued meanings in the use of these terminologies beyond the purely technical interpretation of including or excluding a subset of parameters from larger sets. We have changed our terminology to "**allowlist**" and "**blocklist**" to reflect the worldwide discussion of these terms over the past two years, or are now using descriptive namings for an inclusion lists such as in the **ActionCallbackSelection**.

In an eSignAnyWhere Envelope, we are now using the term "**activity**" instead of "**recipient**". The term "recipient" didn't work gracefully for situations where the "recipient" was just a background service activity such as an "automatic signing" or "automatic sealing" activity. We follow the terminology borrowed from the Business Process Modelling Notation (BPMN), as we consider an envelope as being processed in a signature workflow.

To emphasize the global aspect of the solution, we have adapted terminologies so that they are suitable for worldwide use. One example is the division of the name into the **firstName** and **lastName**, which is not common in every region of the world. Based on common recommendations, we are now using the terms **givenName** and **surname**.

Envelope Status Enumeration

The envelope status values have been reviewed, and status values are now aligned with the status value shown in the WebUI.

API v1-v5		API v6		
Status Values envelope/get	Status Value envelope /find	Status Value envelope /get	Status Value envelope /find	Comment
Started	Active	Active	Active	
InProgress				
			ActionRequired	ActionRequired is a status interpretation (derived from "Active") in context of the current user who calls envelope/find.
	WaitingForOthers		WaitingForOthers	WaitingForOthers is a status interpretation (derived from "Active") in context of the current user who calls envelope/find.
Completed	Completed	Completed	Completed	
CompletedWithWarnings				
Canceled	Canceled	Canceled	Canceled	
Rejected	Rejected	Rejected	Rejected	
(ExpiringSoon)	ExpiringSoon	Active	ExpiringSoon	ExpiringSoon was defined in the v5 model of envelope/get, but actually not used.
Expired	Expired	Expired	Expired	
BulkCompleted	(n/a)	(n/a)	(n/a)	Separated to API methods in "/envelopebulk" route
BulkPartlyCompleted	(n/a)	(n/a)	(n/a)	Separated to API methods in "/envelopebulk" route
Draft	Draft	(n/a)	(n/a)	Separated to API methods in "/draft" route
Template	Template	(n/a)	(n/a)	Separated to API methods in "/template" route

Disposable Certificate Enumeration

Please see the following changes referring to the identification types and the document types for disposable certificate:

Identification type

API v5	API v6
None	None
FOREIGN_TAX_CODE	ForeignTaxCode
PERSONAL_NUMBER	PersonalNumber
PASSPORT	Passport
NATIONAL_IDENTITY_CARD	NationalIdentityCard
ITALIAN_TAX_CODE	ItalianTaxCode
NO_SERIAL_NUMBER	NoSerialNumber

DRIVING_LICENSE	DrivingLicense
RESIDENCE_PERMIT	ResidencePermit
RESIDENCE_PERMIT_TEMP	TemporaryResidencePermit
EMBASSY_DOCUMENT	EmbassyDocument

Document type

API v5	API v6
CI	IdentityCard
PA	DriverLicense
PASS	Passport
RP	ResidencePermit
CIE	NationalElectronicIdentityCard
RT	TemporaryResidencePermit
DC	EmbassyDocument
PD*	
ID*	
PN*	
AT*	

*Not supported with api v6 and UI - but still available with older api versions than api v6.

Complexity Reduction

Structural Changes for Bulk Sending via API & for managing parent envelopes resulting from bulk-sending

While many `/v5/envelope/*` methods covered envelope functionality for both "standard envelopes" and "bulk sending" (or the "parent" envelope of a bulk-sending action), this was split up in v6. The `/v6/envelope/*` methods are now used to send a single envelope or get data of a single envelope (or one of the child envelopes that resulted from a bulk sending action), the `/v6/envelope/bulk/*` methods are now addressing the parent envelope from a bulk-sending action.

Complexity reduction for envelope/find call

To reduce the complexity of the envelope/find call, the call was modified as follows: Instead of allowing arrays of recipients, senders and signers, the arrays were replaced with simple string values. Therefore, with v6 it is possible to search for a combination of one sender, one signer and one recipient not for a combination of several of those mentioned before.

Changes on Method Level

Removed and Deprecated Methods

Following method was removed:

GET `/v5/envelope/{templateId}/copyFromTemplate`

GET `/v5/envelope/{envelopeId}/downloadPageImage/{docRefNumber}/{pageNumber}`

The method was implemented in v5 for internal reasons to the public API by mistake, existing methods of WorkstepController API should be used instead.

POST `/v5/envelope/sendFromTemplate`

This method was implemented for internal usecases and just rarely used in customer integrations. A separate method for sending an envelope from a template was considered as confusing as in most cases anyhow placeholders etc need to be replaced, and we decided to remove this method from API v6 with the goal to keep the API simpler. For those rare use cases where envelopes should be sent directly from a template, a combination of **POST `/v6/{template}/createdraft`** and **POST `/v6/draft/send`** should be used instead.

Changed and added Methods

Authorization

Changed method

The method which can be used to validate if a user is authorized (GET /v4/authorization) got an extra path suffix and is now called GET /v6/authorization /validate to highlight better that it is not mandatory to call this method just for the next API calls.

The method to retrieve information about the user was merged from the following three api calls:

- GET /v5/user/me
- GET /v5/user/{email}
- GET /v5/userById/{userId}

to GET /v6/authorization/whoami. Beside of information like the surname and givenname, the response also provides information about the roles the user has assigned.

Draft

Changed method

The method to send an envelope from an existing draft was changed from POST /v5/draft/send/{draftId} to POST /v6/draft/send. Instead of sending the draftId within the URI, the draftId shall now be added in the body of the api call.

The method to create a draft from an existing template can now be find in the template section and was therefore changed from POST /v5/draft /createFromTemplate to POST /v6/template/createdraft

The method to update a draft and change it with the information provided in the call was changed from PATCH /v5/draft/update/{draftId} to POST /v6/draft /update.

Added method

The method GET /v6/draft/{draftId}/files was added in order to be able to get all files which are included in a draft.

Additionally it is possible to search for a draft with the method POST /v6/draft/find. The response lists all drafts which matches the search request. This was before possible with the envelope find call.

The methods POST /v6/draft/create and GET /v6/draft/{draftId} come with big changes in the model, which allows new use cases starting from a draft in the API and including the Designer UI on such a draft fully integrated in other business applications, using the existing AgentRedirect mechanism.

Envelope

Changed method

Please note that the following api calls changed just in the type of the call and/or in the URI:

- GET /v5/envelope/{envelope}/cancel changed to POST /v6/envelope/cancel
- GET /v5/envelope/{envelope}/remind changed to POST /v6/envelope/remind
- GET /v5/envelope/{envelope}/unlock changed to POST /v6/envelope/unlock

Envelope information

While GET /v5/envelope/{envelopeld} tried to return as much details as possible from an envelope in a specific status (including templates), the method was split up to address the situation specific needs:

- GET /v6/envelope/{envelopeld} returns the basic data of the envelope
- GET /v6/envelope/{envelopeld}/configuration returns the detailed configuration of an envelope
- GET /v6/envelope/{envelopeld}/files returns the file IDs referenced by a specific envelope / template
- GET /v6/envelope/{envelopeld}/forms returns the data related to form field filling
- GET /v6/envelope/{envelopeld}/viewerlinks returns the web URIs which have to be opened by a recipient (signer or needs-to-view) to open the SignAnyWhere Viewer

Download files referenced by the envelope

Depending on an envelope's status, the envelope provides references to several files (the signed document, the audit trail PDF, the audit trail XML, or legal documents such as the certificate request form for issuing a disposable certificate).

To avoid misleading interpretation, the method for downloading such files was changed from GET /v5/envelope/downloadCompletedDocument/{documentId} to GET /v6/file/{fileId}.

Restarting of an already expired envelope

The functionality of both methods

- POST /v5/envelope/{envelopeId}/restart
- GET /v5/envelope/{envelopeId}/restart/{expirationInDays} (where the verb GET was misleading)

is now covered by POST /v6/envelope/restartexpired

Parsing a file to detect Advanced Document Tags or Form Fields contained in the PDF

As the parsing for field information is done on a file BEFORE it is getting part of a draft or envelope, the method was changed from POST /v5/envelope/prepare to POST /v6/file/prepare

In API versions up to v5, the advanced document tag for signature fields did default to the signature method Click2Sign. Starting with v6, we decided to **consider the default signature method specified in the organization settings** when no signature methods have been defined in the Advanced Document Tag syntax.

Sending an Envelope

While POST /v5/envelope/send was supporting both "standard envelope sending" and "bulk sending", the POST /v6/envelope/send focuses on singular envelopes only.

Bulk sending is now covered by POST /v6/envelope/bulk/send

License

The method to retrieve the active license state was changed from GET /v4/license to GET /v6/organization/license.

Organization

The method to retrieve all eSealing profiles from the organization was changed from GET /v5/organization/sealing to GET /v6/organization/automaticprofile.

Recipient

The term "recipient" didn't work gracefully for situations where the "recipient" was just a background service activity such as an "automatic signing" or "automatic sealing" activity. We follow the terminology borrowed from BPMN, as we consider an envelope as being processed in a signature workflow. Therefore we are now using the wording "activity" instead of "recipient", when it isn't clear that a human person is involved. Therefore the following api calls changed (in type and URI)

- DELETE /v4/recipient/{recipientId}/fromEnvelope/{envelopeId} to POST /v6/envelope/activity/delete
- PUT /v4/recipient/{recipientId}/fromEnvelope/{envelopeId} to POST /v6/envelope/activity/replace

SspFile



Please note: It is not allowed to upload multiple documents at once. Related endpoint:

- POST /v6/file/upload
- POST /v5/user/{userId}/uploadSignatureImage
- POST /v4/sspfile/uploadtemporary

- Upload was changed from POST /v4/sspfile/uploadtemporary to POST /v6/file/upload
- Delete was changed from DELETE v4/sspfile/disposefile/{sspFileId} to /v6/file/delete with the fileId in the method's body data (JSON model)

Team

Instead of POST /v4/team, the method which allows changing the entire team configuration of an organization is now called POST v6/team/replace. The additional path component allows to introduce other POST methods on team level in the future.

Templates

The method GET /v5/envelope/{templateId}/copyFromTemplate was replaced by 2 methods: First of all use POST /v6/template/createdraft to create a new draft based on the template, and in a 2nd step use POST /v6/draft/send to make the draft an envelope.

System

To check the eSignAnyWhere version, also often used to perform a system availability check, the method GET /v4/version was moved into a group of "system" related methods and therefore changed to GET /v6/system/version

Changes in the Models

Draft Models

The JSON models of the /v6/draft methods have been changed widely, to be in sync between POST and GET as much as possible, and also be in sync with the envelope models. Please read the section about changes of the Envelope model carefully to see what changed in general.

In methods which create a draft, the ***DraftOptions*** have been renamed to **AgentRedirectConfiguration**.

Envelope Models

The JSON model of the /v6/envelope methods changed widely, not only to make API usage more easy for new customers. Some changes have been made to allow OAS3 compliance. Other changes have been done to avoid abstract types and simplify usage of the REST API with several REST API stub generators. By simplifying structures and rethinking defaults, it should also help existing customers after they migrated to v6.

For reasons of simplification, the POST /v6/envelope/send method data are now not separated in the SspFileId array and other SendEnvelopeDescription data any more. The data which were in SendEnvelopeDescription are now directly in the structure of the root element. contains 2 elements directly under the root level:

The definition of referenced documents was just an array called "***SspFileId***" in API v5. With API v6, it is now an array called "**Documents**" and contains, for each entry, a structure with the FileId and a DocumentNumber. The naming "FileId" in this structure already shows that it is referencing an element (or otherwise accessible) via one of the /v6/file/* methods. The document number is meant as unique reference of the file within the envelope; avoiding implicit numbering just based on an array index. Other elements e.g. for document visibility are referencing the document by this DocumentNumber.

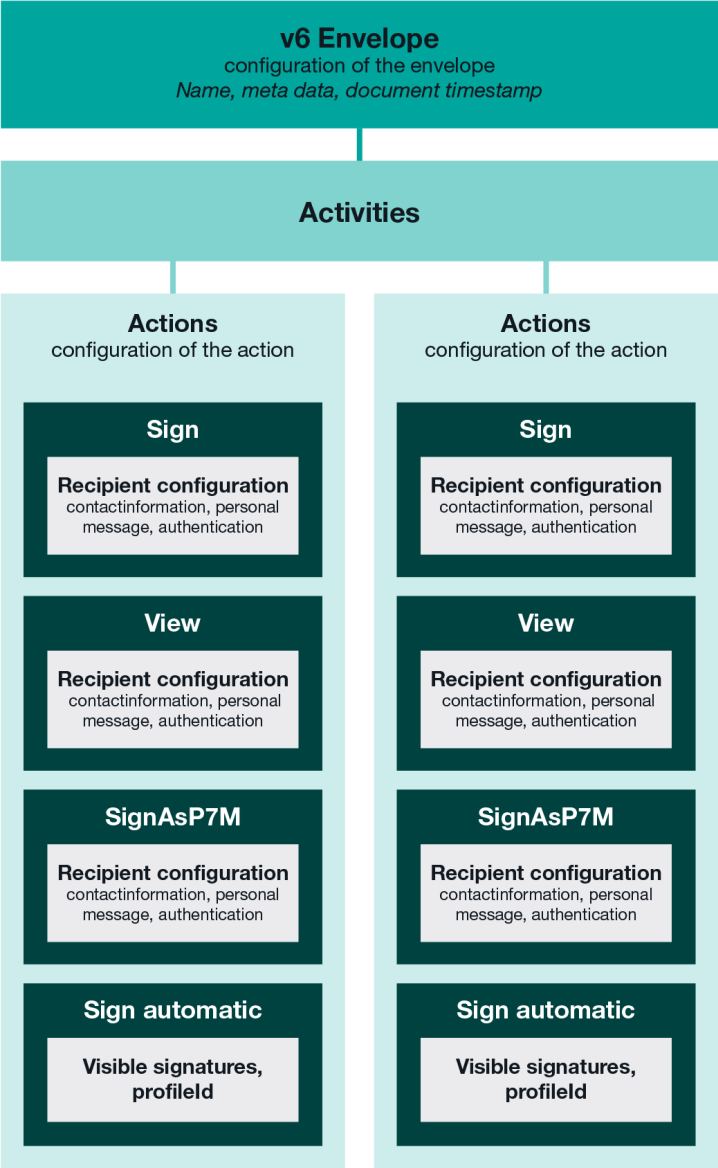
Instead of having all form fields defined in the "***AddFormFields***" array of API v5, and define an extra WorkstepTask for those who are assigned to a signer somewhere inside the recipient's workstepConfiguration data, the form fields are now directly defined in the activity definition. Only those form fields which are not assigned to any activity (only signer or automatic signing/sealing activities can have fields assigned) have to be defined in the new "**UnassignedElements**" array.

Instead of the API v5 "***Steps***" array, where each step usually had to have exactly one "***Recipient***" entry regardless of being a human recipient or an automatic service activity, there is now the "**Activity**" array (and one hierarchy level less). As explained above, we are using the term "Activity" now to follow the common naming of the Business Process Modelling Notation (BPMN) standard, as this fits best for such workflow activities.

Instead of the API v5 "***RecipientType***", which was just a string datatype and expected values such as "Signer", "Cc" and "Acknowledge", the Activity has now an **Action** element where exactly one child element (**Sign**, **View**, **SendCopy**, **SignAutomatic** or **SignAsP7M**) has to be set with an object. All action-specific configuration is defined inside this action object. The structure of the different action objects differs therefore. There is no separate **workstepConfiguration** available any more, as all these data belong to the action.

Instead of the API v5 "***DocumentOptions***" in the Steps element, document visibility is now managed in the "**VisibilityOptions**" of the Activity. The structure is very similar, but as mentioned above, the DocumentNumber now references to a defined number which is set in the Documents array instead of an implicit numbering based on array position.

Overview



Comparison

Figure

REST Api v5

```
{
  "SapFileIds": [
    "string"
  ],
  "SendEnvelopeDescription": {
    "Name": "string",
    "EmailSubject": "string",
    "EmailBody": "string",
    "DisplayedEmailSender": "string",
    "EnableReminders": true,
    "FirstReminderDayAmount": 0,
    "RecurrentReminderDayAmount": 0,
    "BeforeExpirationDayAmount": 0,
    "DaysUntilExpire": 0,
    "ExpirationDate": "2022-09-22T08:47:55.048Z",
    "ExpirationInSecondsAfterSending": 0,
    "CallbackUrl": "string",
    "StatusUpdateCallbackUrl": "string",
    "LockFormFieldsAtEnvelopeFinish": true,
    "BeforeSendDraftRedirectUri": "string",
    "WorkstepEventCallback": {
      "Url": "string",
      "Blacklist": [
        "string"
      ],
      "Whitelist": [
        "string"
      ]
    },
    "Steps": [
      {
        ...
      }
    ],
    "AddFormFields": {
      "Forms": {
        ...
      }
    },
    "OverrideFormFieldValues": {
      "Forms": {}
    },
    "MetadataXml": "string",
    "AttachSignedDocumentsToEnvelopeLog": true
  }
}
```

REST Api v6

```
{
  "Documents": [
    {
      "FileId": "string",
      "DocumentNumber": 0
    }
  ],
  "Name": "string",
  "Metadata": "string",
  "AddDocumentTimestamp": true,
  "ShareWithTeam": true,
  "LockFormFieldsOnFinish": true,
  "UnassignedElements": {
    ...
  },
  "Activities": [
    {
      ...
    }
  ],
  "EmailConfiguration": {
    ...
  },
  "ReminderConfiguration": {
    ...
  },
  "ExpirationConfiguration": {
    ...
  },
  "CallbackConfiguration": {
    ...
  },
  "AgreementConfiguration": {
    ...
  }
}
```

Signature Configuration

For automatic remote signatures, specifying the resulting stamp imprint appearance with on/off switches (DisplayExtraInformation, DisplayEmail, DisplayTransactionId, DisplayTransactionToken, DisplayPhoneNumber, DisplayIp, DisplayName, DisplaySignatureDate) is no longer supported. Compared to an APIv5 integration, all those fields are now ENABLED. To keep the API simpler, we decided to remove this functionality as the [Stamp Imprint Configuration](#) feature meanwhile offers more powerful options and still allows to hide single data rows.

New Functionalities in API v6

In this section we will describe some "new use cases" which have not been possible via REST API v5 in before. The descriptions below will help to get the maximum out of the new API structure. Some of the concepts will be described just in short summaries. Note that this section doesn't aim for completeness.

Define Workstep Parameters on Drafts

In APIv5 the settings on a draft have been very restrictive, now in APIv6 you can set lots of additional parameters. This allows to predefine even those details on a draft, then open the draft using AgentRedirect link and allow the sender to modify some parameters of the envelope-to-be-sent in a what-you-see-is-what-you-get editor right before sending. Parameters such as Client Actions (i.e. redirects after signing) can now be set on drafts already.

Work with Recipient Placeholders

A template can define placeholder values for recipients. In APIv5, there was no strict distinction between drafts/templates/envelopes and therefore it was difficult to work with placeholders. Now the APIv6 allows to define placeholders on a template, create a draft still with unresolved placeholders using the POST `v6/template/createdraft` method and retrieve placeholder information from the draft using GET `/v6/draft/{id}`. Placeholders need to be replaced before sending the draft.