

# Migrate REST API clients from v3 to v5

The changes between REST API v3 and v5 may require some smaller adjustments of the client implementations, but are in general not requiring big-effort changes. This migration guide should give an overview on how to migrate from API v3 to API v5. Please note that at the time of writing this guide the API v6 was already announced, but the changes from v3 or even v5 to v6 are of bigger nature. There will be a separate migration guide from v5 to v6 available. But for projects that are still using API v3, we recommend to migrate to v5 first and then migrate from v5 to v6 as already this 2nd step will be a big change.

**i** REST API /v3 and /v4 DEPRECATION: The 23.76 (published March 2024) will be the last LTS version that includes these API versions. By early June 2024, the REST API routes to v3/v4 will be deactivated on DEMO. Early December 2024, the REST API routes to v3/v4 will be removed from feature stream releases. Note that there is no date communicated yet to discontinue REST APIv5 (and where v5 refers to v4 routes, these will still remain); however we recommend to use the /v6 API specification already.

**i** We recommend to migrate now from REST API v3 to REST API v5 as long as new features of API v6 are not mandatory. API v5 will remain part of the product beyond the announced deprecation of v3. For new projects starting after Jul 2022, we strongly recommend to start with REST API v6.

## Conceptual Changes

Instead of having all routes strictly with /v3 in the path component, the v5 is redefining on a /v5 path only those methods where breaking changes between the API v4 and API v5 make it necessary to provide different implementations one beside the other. Therefore, the v5 API contains different versions in different methods, e.g. the GET /v4/authorization and the POST /v5/envelope/send.

With API v4 and v5, the **/envelope** methods have been split up in those dealing with envelopes (**/envelope**), and those just creating/reading/updating a draft (**/draft**).

## Changes on Method Level

### Removed and Deprecated Methods

The method **GET /v5/user/{email}** should be considered as DEPRECATED, and **GET /v5/userByEmail/{email}** should be used instead, as a GET on a user, with an "email" as path variable, violates the OAS3 specification because the "userId" is the primary identifier e.g. for the DELETE or PATCH verb.

### Changed Methods

The method POST /v3/envelope/create was renamed to POST /v5/draft/create.

The method POST /v3/envelope/createFromTemplate was renamed to POST /v5/draft/createFromTemplate.

While GET /v5/envelope/{envelopeld} remains with the purpose of reading details of an envelope (after it was sent), on an envelope in draft state the new method GET /v5/draft/{draftld} should be used instead of GET /v3/envelope/{envelopeld}.

### Added Methods

Methods to search for users have been introduced, which allow searching for users by both types of parameters and help to provide more easily an OAS3 compliant facade in case this is required.

The newly introduced methods therefore are:

- GET /v5/userById/{userId}
- GET /v5/userByEmail/{email}

In conjunction with the changed methods for draft management, following new methods have been added to provide new functionality on drafts via the API:

- POST /v5/draft/send/{draftld}
- PATCH /v5/draft/update/{draftld}

As result of allowing to unlock a locked envelope via API, following method was added:

- GET /v5/envelope/{envelopeld}/unlock

Envelopes which are expired can be restarted. To allow this also via the API, following method was added:

- POST /v5/envelope/{enveloped}/restart

For sending envelopes with automatic remote signatur/sealing, it is required to provide the sealing profile Id in the POST /envelope/send. To allow convinient selection of sealing profiles, the following method was implemented:

- GET /v5/organization/sealing

To allow OAuth2 use cases, following method was added (mainly for the purpose of offering data via a resource URI, e.g. to "authenticate via eSignAnyWhere"; somehow equivalent (but not fulfilling all requirements of) the /userinfo endpoint of OIDC)

- GET /v5/user/me

For active monitoring of the license consumption, following method was implemented:

- GET /v4/license

## Changes in the Models

Following changes has been made from v3 to v5 in JSON definitions referring to the /envelope/send api call.

Hyperlink navigation areas to WWW resources were added:

```

"Navigation": {
  "HyperLinks": [
    {
      "Id": "string",
      "DocRefNumber": 0,
      "PositionPage": 0,
      "Uri": "string",
      "Position": {
        "PositionX": 0,
        "PositionY": 0
      },
      "Size": {
        "Height": 0,
        "Width": 0
      }
    }
  ],
  "Links": [
    {
      "Id": "string",
      "DocRefNumber": 0,
      "PositionPage": 0,
      "TargetId": "string",
      "Position": {
        "PositionX": 0,
        "PositionY": 0
      },
      "Size": {
        "Height": 0,
        "Width": 0
      }
    }
  ],
  "LinkTargets": [
    {
      "Id": "string",
      "DocRefNumber": 0,
      "PositionPage": 0,
      "Position": {
        "PositionX": 0,
        "PositionY": 0
      },
      "Size": {
        "Height": 0,
        "Width": 0
      }
    }
  ]
}

```

Identification methods section was added to the JSON definition:

```

"IdentificationMethods": [
  {
    "Method": "OAuth",
    "Parameter": "string",
    "Filters": [
      {
        "CompareOperation": "Equals",
        "FilterId": "string",
        "FilterValue": "string"
      }
    ]
  }
],

```