

Ezidox Pty Ltd

ABN: 44 601 979 462

B2B Services Overview



Ver 1.11
20/9/2019

Table of Contents

AUTHENTICATION	5
REST APIS - PRODUCTION	6
TESTING CREDENTIALS	6
REST APIS - SANDBOX	ERROR! BOOKMARK NOT DEFINED.
TESTING CREDENTIALS	ERROR! BOOKMARK NOT DEFINED.
COMMON VALUES	6
CREATE APPLICATION REQUEST	7
SERVICE	7
PAYLOADPROPERTIES	7
PRIMARYSTAKEHOLDER AND OTHERSTAKEHOLDERSPROPERTIES	7
PRIMARYCOLLECTOR AND OTHERCOLLECTORS PROPERTIES	9
EXAMPLE PAYLOAD (SINGLE APPLICANT, USING EXISTING TEMPLATE)	10
EDIT APPLICATION REQUEST	11
SERVICE	11
PAYLOAD PROPERTIES	11
STAKEHOLDER PROPERTIES	11
SETTINGS	14
SETTING EXAMPLE	14
EXAMPLE FULL PAYLOAD – SCENARIO ADD NEW DOCUMENT	14
EXAMPLE FULL PAYLOAD – SCENARIO REMOVE DOCUMENT	15
EXAMPLE FULL PAYLOAD – ADD NEW STAKEHOLDER WITH OUTDOCUMENTS	16
SEARCH ATTACHMENTS	18
SERVICE	18
PAYLOAD PROPERTIES	18
EXAMPLE PAYLOAD	18
UPLOAD ATTACHMENTS AS BASE 64	19
SERVICE	19
PAYLOAD PROPERTIES	19
COLLECTOR PROPERTIES	19
EXAMPLE PAYLOAD – WITH ESIGNATURE	19
EXAMPLE PAYLOAD – WITHOUT ESIGNATURE	20
UPLOAD ATTACHMENTS	21
SERVICE	21
PAYLOAD PROPERTIES	21
COLLECTOR PROPERTIES	21
EXAMPLE PAYLOAD (IMPERSONATING CUSTOMER)	21
SEND ATTACHMENTS	22
SERVICE	22
PAYLOAD PROPERTIES	22
COLLECTOR PROPERTIES	22
MODELS PROPERTIES	22

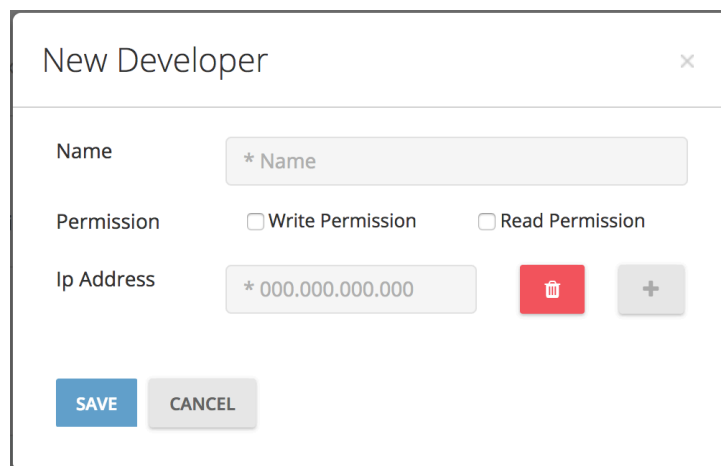
EXAMPLE PAYLOAD— WITH ESIGNATURE	23
EXAMPLE PAYLOAD (2 COLLECTORS) – WITHOUT ESIGNATURE	24
GET APPLICATIONS LIST	25
SERVICE	25
PAYLOAD PROPERTIES	25
EXAMPLE PAYLOAD (WITHOUT IMPERSONATION)	25
GET APPLICATION DETAILS	25
SERVICE	25
PAYLOAD PROPERTIES	25
EXAMPLE PAYLOAD (WITHOUT IMPERSONATION)	25
EXAMPLE PAYLOAD (IMPERSONATING CUSTOMER)	25
EXAMPLE PAYLOAD (IMPERSONATING COLLECTOR)	25
EXPORT DOCUMENTS	26
SERVICE	26
PAYLOAD PROPERTIES	26
EXAMPLE PAYLOAD (DOWNLOAD ALL DOCUMENTS AS SINGLE PDF)	26
EXAMPLE PAYLOAD (DOWNLOAD ALL DOCUMENTS AS ZIP ARCHIVE)	26
DOWNLOAD DOCUMENT	27
SERVICE	27
PAYLOAD PROPERTIES	27
EXAMPLE PAYLOAD (WITHOUT IMPERSONATION)	27
EXAMPLE PAYLOAD (IMPERSONATING CUSTOMER)	27
EXAMPLE PAYLOAD (IMPERSONATING COLLECTOR)	27
DOWNLOAD INDIVIDUAL DOCUMENT FILE	28
SERVICE	28
PAYLOAD PROPERTIES	28
EXAMPLE PAYLOAD (WITHOUT IMPERSONATION)	28
EXAMPLE PAYLOAD (IMPERSONATING CUSTOMER)	28
EXAMPLE PAYLOAD (IMPERSONATING COLLECTOR)	28
UPLOAD A FILE TO A DOCUMENT	29
SERVICE	29
PAYLOAD PROPERTIES	29
EXAMPLE PAYLOAD (WITHOUT IMPERSONATION)	29
EXAMPLE PAYLOAD (IMPERSONATING CUSTOMER)	29
EXAMPLE PAYLOAD (IMPERSONATING COLLECTOR)	29
WEBHOOKS	31
SETUP	31
SERVICE – ACTIVATE WEBHOOK	31
PAYLOAD PROPERTIES	31
EXAMPLE PAYLOAD	31
SERVICE – DEACTIVATE WEBHOOK	32
PAYLOAD PROPERTIES	32
EXAMPLE PAYLOAD	32
SERVICE – TEST WEBHOOK	32

PAYLOAD PROPERTIES	32
EXAMPLE PAYLOAD	33
B2B WEBVIEWS	34
TEMPLATESET SERVICES	38
SERVICE – SEARCH TEMPLATESETS	38
SERVICE – GET INDIVIDUAL TEMPLATESET	38
TEMPLATES SERVICES	38
SERVICE – SEARCH TEMPLATE	38
SERVICE – GET INDIVIDUAL TEMPLATE	38

Authentication

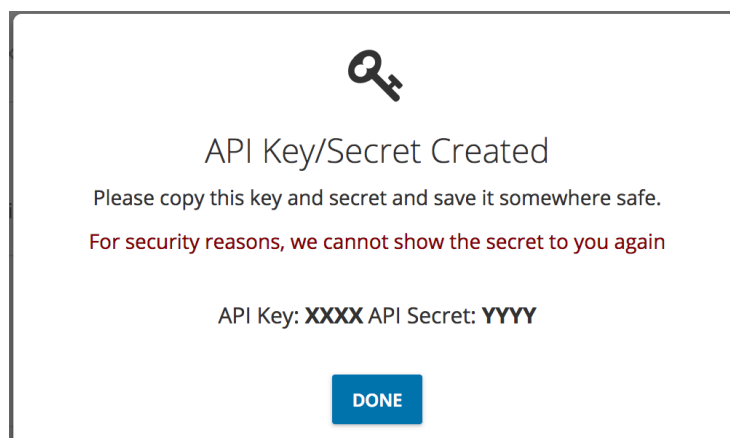
To authenticate to the ezidox system and access to the developer APIs, the first action to do is to:

- register an account
- go to Settings
- select “Company Details” and create a new Company
- select “Developers” and create a new Developer, including the IP Addresses to include in the whitelist (<https://portal.ezidox.com/#/admin/developers>)
- to bypass the IP Address validation, enter * in the IP Address field (**NOT RECOMMENDED**)



You will then receive:

- **ApiKey:** a randomly generated alphanumeric key
- **ApiSecret:** a randomly generated alphanumeric key



That you will need to pass each time a REST call is generated in the header of the request using the following format:

```
Authorization: amx {ApiKey}:{ApiSecret}
```

REST APIs - PRODUCTION

The full list of REST APIs can be found to the following address:

<https://api.ezidox.com/swagger/ui/index>

In order to access to the private APIs you will always need to be authenticated as per previous section.

The API call will accept and respond with JSON messages.

Testing Credentials

Below Collector account has been created:

- Portal: WWWWW
- Username:XXXX
- Password:YYYYY

Below Customer account has been created:

- Portal: WWWWW
- Username:XXXX
- Password:YYYYY

Below API Credentials have been generated for the demo account

- **API Key:** WWWWW
- **API Secret:** ZZZZZ

COMMON VALUES

UserType: When requested in the payloads/URL, the usertype values are:

- 1: Collector (Broker)
- 2: Sender (Customer)

IMPORTANT: In scenarios where the impersonated user doesn't have access right to access the functionality, the service will return a 401 (Not Authorised).

This will have to be improved by returning a more detailed error message.

Create Application Request

Service

POST /apiB2B/Applications

PayloadProperties

Name: Application Name (option), if not provided it's automatically generated

PrimaryStakeholder/OtherStakeholders: see next section

PrimaryCollector/OtherCollectors: see next section

IncludeDefaultCollectors: Share application with default Collectors

OtherCollectors: if you don't use IncludeDefaultCollectors (or in addition to it) you can also specify an array of collectors id to share

ExternalRef: external system identifier associated to the application. Can be used to query the application. This identifier is unique and is not visible in the UI.

ReferenceNumber: external identifier associated to the application. This identifier doesn't need to be unique, is visible in the UI and can be modified by the collectors.

DayUntilDueDate: used to set the DueDate parameter on the application (will add DayUntilDueDate to today)

WelcomeMessage: used set the Application Welcome Message. On order or priority the system will use (higher to lower):

- Welcome Message defined in the request
- Welcome Message associated to the template for each stakeholder

NotifyCustomers: do you want to notify the customer via email? (default true). If false, it will not send email to the customers for this transaction.

PrimaryStakeholder and OtherStakeholdersProperties

StakeholderTypeId: Stakeholder type, values from list:

- 1 - Primary Sender
- 2 - Co-Borrower
- 3 - Guarantor
- 4 - 3rd Party documents (e.g. accountant, lawyer)
- 5 - Company
- 6 - Company and Individual

List of values is configurable, and differs from the production ones

ShowOtherDocs: Is the stakeholder allowed to see the list of documents for the other stakeholders in the application?

IncludeDefaultOutdocuments: include yes/no the document to be sent to the customer when a new application is created. Currently no OutDocuments are set by default on any template.

OutDocuments: if you don't use IncludeDefaultOutdocuments (or in addition to it) you can also specify an array of document id to send to the customer

Documents available are:

- XXXX – To Be Added

TemplateSet section

You can either use the configuration of templates available on ezidox, or create an ad-hoc set of documents according to your business logic.

Option 1: If you want to use the configuration of templates available on ezidox, you need to specify the below:

TemplateSet:

- **Id:** specify the template id that you want to use
- **IncludeDefaultDocumentGroups:** do you want to include all the groups flagged as "Selected by Default?"
- **IncludeDefaultDocuments:** do you want to include all the documents flagged as "Selected by Default?" for all the groups
- **DocumentGroups:** If you want to request groups/documents in addition to the ones selected by default, populate this object.
 - **TemplateGroupId:** Id of the group
 - **Name/Description:** (Optional) used to override the default values
 - **IncludeDefaultDocuments:** do you want to include all the documents flagged as "Selected by Default?" for this group
 - **Documents:** If you want to request documents in addition to the ones selected by default, populate this object.
 - **TemplateId:** Id of the document
 - **Name/Description:** (Optional) used to override the default values

```
"TemplateSet": {
  "Id": 6815,
  "IncludeDefaultDocumentGroups": true,
  "IncludeDefaultDocuments": true,
  "DocumentGroups": [
    {
      "TemplateGroupId": 7267,
      "IncludeDefaultDocuments": true
    },
    {
      "TemplateGroupId": 7268,
      "Documents": [
        {
```



```

    "TemplateId": 17
  }
]
}
]
}

```

Option 2: If you want to specify your own list of documents, you need to specify the below:

TemplateSet:

- **DocumentGroups:** List of document groups that you want to request.
 - **Name:** Name of the group
 - **Description:** Description of the group
 - **Documents:** List of documents that you want to request in the group
 - **TemplateId:** Id of the document
 - **Name/Description:** (Optional) used to override the default values

```

"TemplateSet": {
  "DocumentGroups": [
    {
      "Name": "ID Documents",
      "Description": "Docs Description",
      "Documents": [
        {
          "TemplateId": 100
        }
      ]
    },
    {
      "Name": "Tax Documents",
      "Documents": [
        {
          "TemplateId": 17,
          "Name": "Tax Return",
          "Description": "Current year Tax Return"
        }
      ]
    }
  ]
}

```

[PrimaryCollector and OtherCollectors Properties](#)

PrimaryCollector is used to define the Collector assigned to the Application.
 OtherCollectors is defined to define the collectors sharing the Application.

Collectors can be identified via:

Id: Ezidox Id assigned to the collector

ExternalRef: External Reference assigned to the collector – currently assigned when the Collector is created via B2B

Email: collector email

Example payload (single applicant, using existing template)

```
{
  "Name": "Loan Application Test",
  "PrimaryStakeholder": {
    "Email": "customer@custemail.com",
    "Name": "Test",
    "Surname": "Test",
    "MobileNumber": "+610400000000",
    "IsCompany": false,
    "IncludeDefaultOutdocuments": true,
    "TemplateSet": {
      "Id": 6855,
      "IncludeDefaultDocumentGroups": true,
    },
    "IncludeDefaultDocuments": true
  },
  "IsUploader": true,
  "StakeholderTypeId": 1,
  "ShowOtherDocs": true
},
"PrimaryCollector": {
  "Email": "collector@company.com"
},
"IncludeDefaultCollectors": true,
"DayUntilDueDate": 15,
"IsHiddenFromStakeholder": false
}
```

Edit Application Request

Service

PUT /apiB2B/Applications/{applicationId}

Payload Properties

ApplicationId: Application id.

Publish: Boolean, true to publish the application when in Draft Status.

Name: Application Name.

DocsDueDate: Due date of the docs.

WelcomeMessage: used set the Application Welcome Message. On order or priority the system will use (higher to lower):

- Welcome Message defined in the request
- Welcome Message associated to the template for each stakeholder

NotifyCustomers: do you want to notify the customer via email? (default true). If false, it will not send email to the customers for this transaction.

Stakeholders: see next section

Settings: see next section

Stakeholder Properties

__EDIT_ACTION_TYPE: type of the action:

```
{  
  DELETED = -1,  
  EDITED = 0,  
  ADDED = 1  
}
```

__CAN_EDIT_TEMPLATESSET: Boolean, Indicates if you can select another templateset in place of the current one (it is false if at least a document has been uploaded on it).

Id: stakeholder id.(*)

Email: email of the stakeholder.(*)

Name: name of the stakeholder.

MobileNumber: mobile number.

IsCompany: Boolean that indicates if it's a company.

CompanyName: company name.

CompanyABN: company ABN.

IsUploader: Boolean is the stakeholder is uploader.**(*)**

IsPrimary: Boolean if stakeholder is primary.**(*)**

StakeholderTypeId: Stakeholder type, values from list:

- 1 - Primary Sender
- 2 - Co-Borrower
- 3 - Guarantor
- 4 - 3rd Party documents (e.g. accountant, lawyer)
- 5 - Company
- 6 - Company and Individual

List of values is configurable, and differs from the production ones

ShowOtherDocs: Is the stakeholder allowed to see the list of documents for the other stakeholders in the application?

OutDocuments: you can also specify an array of document id to send to the customer

(*): Mandatory param.

TemplateSet section

You can either use the configuration of templates available on ezidox, or create an ad-hoc set of documents according to your business logic.

Option 1: If you want to use the configuration of templates available on ezidox, you need to specify the below:

TemplateSet:

- **Id:** specify the template id that you want to use
- **__EDIT_ACTION_TYPE:** type of the action: { **DELETED=-1, EDITED=0, ADDED=1**}
- **DocumentGroups:** If you want to request groups/documents in addition to the ones selected by default, populate this object.
 - **TemplateGroupId:** Id of the group
 - **__EDIT_ACTION_TYPE:** type of the action: { **DELETED=-1, EDITED=0, ADDED=1**}
 - **Name/Description:** (Optional) used to override the default values
 - **Documents:** If you want to request documents in addition to the ones selected by default, populate this object.
 - **TemplateId:** Id of the document
 - **__EDIT_ACTION_TYPE:** type of the action: { **DELETED=-1, EDITED=0, ADDED=1**}

- **Name/Description:** (Optional) used to override the default values

```
"TemplateSet": {
  "__EDIT_ACTION_TYPE": 0,
  "__CAN_DELETE": true,
  "Id": 6815,
  "IncludeDefaultDocumentGroups": true,
  "IncludeDefaultDocuments": true,
  "DocumentGroups": [
    {
      "__EDIT_ACTION_TYPE": 0,
      "__CAN_DELETE": true,
      "TemplateGroupId": 7267,
      "IncludeDefaultDocuments": true
    },
    {
      "TemplateGroupId": 7268,
      "Documents": [
        {
          "__EDIT_ACTION_TYPE": 0,
          "__CAN_DELETE": true,
          "TemplateId": 17
          "Name": "string",
          "Description": "string"
        }
      ]
    }
  ]
}
```

Option 2: If you want to specify your own list of documents, you need to specify the below:

TemplateSet:

- **DocumentGroups:** List of document groups that you want to request.
 - **Name:** Name of the group
 - **Description:** Description of the group
 - **Documents:** List of documents that you want to request in the group
 - **TemplateId:** Id of the document
 - **Name/Description:** (Optional) used to override the default values

```
"TemplateSet": {
  "__EDIT_ACTION_TYPE": 0,
  "__CAN_DELETE": true,
  "DocumentGroups": [
    {
      "__EDIT_ACTION_TYPE": 0,
      "__CAN_DELETE": true,
      "Name": "ID Documents",
      "Description": "Docs Description",
      "Documents": [
        {
          "__EDIT_ACTION_TYPE": 0,
          "__CAN_DELETE": true,
          "TemplateId": 100
        }
      ]
    }
  ],
}
```

```

    {
      "Name": "Tax Documents",
      "Documents": [
        {
          "__EDIT_ACTION_TYPE": 0,
          "__CAN_DELETE": true,
          "TemplateId": 17,
          "Name": "Tax Return",
          "Description": "Current year Tax Return"
        }
      ]
    }
  ]
}

```

Settings

You can also change settings related to the application:

NOTIFICATION_FREQUENCY: Frequency of recurring notifications (**integer**).

BEFORE_DUE_DATE_REMINDER: Send reminder 3 days before Due Date? (**boolean**).

DUE_DATE_REMINDER: Send reminder on due date? (**boolean**).

NOTIFY_COMPLETED_UPLOADED_PERCENTAGE: Notify sender when 100% documents have been uploaded? (**boolean**).

APP_DAILY_SUMMARY: Send daily digest email? (**boolean**).

Setting Example

```

"Settings":
  {
    "NOTIFICATION_FREQUENCY": "7",
    "BEFORE_DUE_DATE_REMINDER": "1",
    "DUE_DATE_REMINDER": "1",
    "NOTIFY_COMPLETED_UPLOADED_PERCENTAGE": "1",
    "APP_DAILY_SUMMARY": "1"
  }

```

Example FULL payload – Scenario Add New Document

__EDIT_ACTION_TYPE: To add a new document this is the hierarchy needed:

- **Level Stakeholder: 0 (EDITED)**
- **Level TemplateSet:0 (EDITED)**
- **Level DocumentGroup:0 (EDITED)**
- **Level Document:1 (ADDED)**

```

{
  "Id": 12878,
  "Name": "Prova Prova",
  "ReferenceNumber": "",
  "ExternalRef": "",
  "WelcomeMessage": null,
  "Stakeholders": [
    {
      "__EDIT_ACTION_TYPE": 0,
      "Id": 12827,
      "IsPrimary": true,
      "IsUploader": true,
      "Email": "test@gmail.com",
      "TemplateSet": {
        "__EDIT_ACTION_TYPE": 0,
        "Id": 0,
        "DocumentGroups": [
          {
            "__EDIT_ACTION_TYPE": 0,
            "Id": 13014,
            "Name": "Additional Documents",
            "Documents": [
              {
                "__EDIT_ACTION_TYPE": 1,
                "TemplateId": 8742,
                "Name": "CUSTOM EXTERNAL FORM",
                "Description": null
              }
            ]
          }
        ]
      }
    }
  ]
}

```

Example FULL payload – Scenario Remove Document

__EDIT_ACTION_TYPE : To remove a new document this is the hierarchy needed:

- **Level Stakeholder: 0 (EDITED)**
- **Level TemplateSet:0 (EDITED)**
- **Level DocumentGroup:0 (EDITED)**
- **Level Document: -1 (DELETED)**

```

{
  "Id": 12878,
  "Name": "Prova Prova",
  "ReferenceNumber": "",
  "ExternalRef": "",
  "WelcomeMessage": null,
  "Stakeholders": [
    {
      "__EDIT_ACTION_TYPE": 0,
      "Id": 12827,
      "IsPrimary": true,

```

```

    "IsUploader": true,
    "Email": "prova@gmail.com",
    "TemplateSet": {
      "__EDIT_ACTION_TYPE": 0,
      "Id": 0,
      "DocumentGroups": [
        {
          "__EDIT_ACTION_TYPE": 0,
          "Id": 13014,
          "Name": "Additional Documents",
          "Documents": [
            {
              "__EDIT_ACTION_TYPE": -1,
              "TemplateId": 8742,
              "Id": 22473,
              "Name": "CUSTOM EXTERNAL FORM",
              "Description": null
            }
          ]
        }
      ]
    }
  ]
}

```

Example FULL payload – Add New Stakeholder with OutDocuments

__EDIT_ACTION_TYPE: To add a new stakeholder this is the hierarchy needed:

- **Level Stakeholder: 1 (ADDED)**
- **Level TemplateSet: 0 (EDITED)**
- **Level DocumentGroup: 1 (ADDED)**
- **Level Document: 1 (ADDED)**

Stakeholders: Primary stakeholder is mandatory.

```

{
  "Id": 12878,
  "Name": "Prova Prova",
  "ReferenceNumber": "",
  "ExternalRef": "",
  "WelcomeMessage": null,
  "Stakeholders": [
    {
      "__EDIT_ACTION_TYPE": 0,
      "Id": 12827,
      "Email": "prova@gmail.com",
      "IsUploader": true,
      "IsPrimary": true
    },
    {
      "__EDIT_ACTION_TYPE": 1,
      "Email": "second@gmail.com",
      "NameSurname": "Second Second",

```



```

    "Name": "Second",
    "Surname": "Second",
    "MobileNumber": "61412345678",
    "IsUploader": true,
    "IsPrimary": false,
    "StakeholderTypeId": 4,
    "ShowOtherDocs": true,
    "OutDocuments": [
      {"Id": 3481},
      {"Id": 3425},
      {"Id": 3485}
    ],
    "TemplateSet": {
      "__EDIT_ACTION_TYPE": 0,
      "DocumentGroups": [
        {
          "__EDIT_ACTION_TYPE": 1,
          "Documents": [
            {
              "__EDIT_ACTION_TYPE": 1,
              "TemplateId": 8723,
              "Name": "MicTest",
              "Description": null
            }
          ]
        }
      ]
    }
  }
]
}

```

Search Attachments

Service

GET /apiB2B/OutDocuments/Search

Payload Properties

nameFilter: Apply e-signature functionality to the documents sent

templateExternalRef: Used only when ApplyESign is true. If true, the eSign request will generate a separate envelope for each stakeholder. Defaulted to false.

onlyESign: Used only when ApplyESign is false. If true, the documents sent will be attached to the email sent to customers. Defaulted to true.

Please note: only attachments shared at company level are returned.
If you don't specify any parameter, the full list of attachments will be returned.

Example payload

/apiB2B/OutDocuments/Search?nameFilter=Test

Upload Attachments as Base 64

Service

POST /apiB2B/OutDocuments/Upload/Base64

Payload Properties

Name: The name of the attachment

Description: The description of the attachment (optional)

IsReusable: True if you want to be able to use the attachment for multiple applications; false if you want to use it only for a specific application. Default true

ApplicationId/ApplicationExternalRef: Either ApplicationId or ApplicationExternalRef is required when IsReusable=false to identify the application associated to the attachment

EsignTemplateId: Id of the associated e-signature template; this is optional. Only to be passed when the document is associated to e-signature functionality.

Base64File: Base64 encoded file

Filename: Original filename with extension

Collector: See Collector section

Collector Properties

Collector is used to define the Collector sending the documents.

Collectors can be identified via:

Id: Ezidox Id assigned to the collector

ExternalRef: External Reference assigned to the collector – currently assigned when the Collector is created via B2B

Email: collector email

Example payload – With eSignature

```
{
  "Collector": {
    "ExternalRef": "XYZ",
  },
  "Name": "Loan Contract",
  "IsReusable": false,
  "ApplicationExternalRef": "ABC",
  "EsignTemplateId": 123,
  "Base64File": "string",
  "Filename": "LoanContractApp123.pdf"
}
```

Example payload – Without eSignature

```
{
  "Collector": {
    "ExternalRef": "XYZ",
  },
  "Name": "Loan Contract",
  "IsReusable": true,
  "Base64File": "string",
  "Filename": "LoanContractApp123.pdf"
}
```

Upload Attachments

Service

POST /apiB2B/OutDocuments/Upload

Payload Properties

name: The name of the attachment

description: The description of the attachment (optional)

isReusable: True if you want to be able to use the attachment for multiple applications; false if you want to use it only for a specific application. Default true

applicationId/applicationExternalRef: Either ApplicationId or ApplicationExternalRef is required when IsReusable=false to identify the application associated to the attachment

esignTemplateId: Id of the associated e-signature template; this is optional. Only to be passed when the document is associated to e-signature functionality.

collector: See Collector section

Collector Properties

Collector is used to define the Collector sending the documents.

Collectors can be identified via:

collectorId: Ezidox Id assigned to the collector

collectorExternalRef: External Reference assigned to the collector – currently assigned when the Collector is created via B2B

collectorEmail: collector email

Example payload (impersonating Customer)

```
/apiB2B/OutDocuments/Upload/95252?collectorEmail=custname@custemail.com&name=Test
```

Comments:

To execute the call we need to perform a multi-part form Content Type.

Below a simple HTML example to test this

Send Attachments

Service

POST /apiB2B/Applications/{applicationId}/OutDocuments

Payload Properties

ApplyESign: Apply e-signature functionality to the documents sent

SplitEnvelopes: Used only when ApplyESign is true. If true, the eSign request will generate a separate envelope for each stakeholder. Defaulted to false.

AttachToEmail: Used only when ApplyESign is false. If true, the documents sent will be attached to the email sent to customers. Defaulted to true.

NotifyCustomers: Notify customers via email/push notification when the request is sent. Defaulted to true.

Collector: See Collector section

Models: See Models section

Collector Properties

Collector is used to define the Collector sending the documents.

Collectors can be identified via:

Id: Ezidox Id assigned to the collector

ExternalRef: External Reference assigned to the collector – currently assigned when the Collector is created via B2B

Email: collector email

Models Properties

StakeholderEmail: email address of the stakeholder receiving the documents

OverrideStakeholderRole: this will match the signer role configured in DocuSign and overrides the one automatically generated by ezidox

OutDocuments: list of id of documents to be sent to each stakeholder

Currently documents available are:

- XXXX – To be added

Comment: Comment to send to the stakeholder

Placeholders: List of placeholders key/value to be replaced in external systems. Placeholders have a number of types, which reflect the types in the external eSign platform:

- Text

- Number
- Date
- Title
- Dropdown
- Checkbox

For each of them, we need to provide:

- Key: Key of the placeholder (field “Data Label” in DocuSign)
- Value: value of the placeholder

ASSOCIATE and/or CREDIT REP MEMBER APPLICATION

BUSINESS DETAILS

Full Member Business Name	Text
Contact Person	

INDIVIDUAL - MEMBER DETAILS / APPLICANT (PLEASE PROVIDE FULL LEGAL NAME)

First Name	First Name
Last Name	Last Name
Mercury CA Number (if allocated)	
Title	Text
Date of Birth	
Driver's License Number	
Email Address	

In our scenario all placeholders will be of type “Text” and can be referenced using the descriptions below – these should be used in DocuSign Data Label for the Text fields, and in the B2B API request

Example payload– With eSignature

```
{
  "Collector": {
    "Email": "collector@company.com"
  },
  "Models": [
    {
      "StakeholderEmail": "custname@custemail.com",
      "OverrideStakeholderRole": "B2B",
      "Placeholders": {
        "Text": [
          {
            "Key": "Business Entity Name",
            "Value": "Broker Company 1"
          },
          {
            "Key": "Title",
            "Value": "Mr"
          }
        ]
      }
    }
  ],
  "OutDocuments": [
    {
      "Id": 10392
    }
  ]
}
```

```
    }  
  ],  
  "ApplyESign": true,  
  "SplitEnvelopes": true,  
  "NotifyCustomers": true,  
}
```

Example payload (2 collectors) – Without eSignature

```
{  
  "Collector": {  
    "Email": "collector@company.com"  
  },  
  "Models": [  
    {  
      "StakeholderEmail": "custname@custemail.com",  
      "OutDocuments": [  
        {  
          "Id": 10392  
        }  
      ]  
    }  
  ],  
  "ApplyESign": false,  
  "NotifyCustomers": true,  
}
```


Get Applications List

Service

GET /apiB2B/Applications

Payload Properties

returnDeleted: Do you want to include deleted applications. Default false

Example payload (without impersonation)

/apiB2B/Applications

Get Application Details

Service

GET /apiB2B/Applications/{id}

Payload Properties

Id: The id of the application

userType: User Type (1: Collector/2: Sender)

userId: User Id

userExternalRef: External Ref of user (Collector Only)

userEmail: User email

includeDocuments: Do you want to include the documents in the application. Default false

Example payload (without impersonation)

/apiB2B/Applications/22391?includeDocuments=true

Example payload (impersonating Customer)

/apiB2B/Applications/22391?userType=2&userEmail=custname@custemail.com&includeDocuments=true

Example payload (impersonating Collector)

/apiB2B/Applications/22391?userType=1&userEmail=collector@company.com&includeDocuments=true

Export Documents

Service

GET /apiB2B/Applications/{applicationId}/PDF

GET /apiB2B/Applications/{applicationId}/ZIP

GET /apiB2B/Applications/Ext/{externalRef}/PDF

GET /apiB2B/Applications/Ext/{externalRef}/ZIP

Payload Properties

applicationId: The id of the application

externalRef: The external Reference of the application

Example payload (download all documents as single PDF)

/apiB2B/Applications/123/PDF

Example payload (download all documents as ZIP Archive)

/apiB2B/Applications/123/ZIP

Download Document

Service

GET /apiB2B/Documents/{documentId}/Download

Payload Properties

documentId: The id of the document

userType: User Type (1: Collector/2: Sender)

userId: User Id

userExternalRef: External Ref of user (Collector Only)

userEmail: user Email

Example payload (without impersonation)

/apiB2B/Documents/95252/Download

Example payload (impersonating Customer)

/apiB2B/Documents/95252/Download?userType=2&userEmail=custname@custemail.com

Example payload (impersonating Collector)

/apiB2B/Documents/95252/Download?userType=1&userEmail=collector@company.com

Download Individual Document File

Service

GET /apiB2B/Documents/DocumentFile/{documentFileId}/Download

GET /apiB2B/Documents/DocumentFile/{documentFileId}/Download/Base64

Payload Properties

documentFileId: The id of the document file

userType: User Type (1: Collector/2: Sender)

userId: User Id

userExternalRef: External Ref of user (Collector Only)

userEmail: user Email

Example payload (without impersonation)

/apiB2B/Documents/DocumentFile/39261/Download

Example payload (impersonating Customer)

/apiB2B/Documents/DocumentFile/39261/Download?userType=2&userEmail=custname@custemail.com

Example payload (impersonating Collector)

/apiB2B/Documents/DocumentFile/39261/Download?userType=1&userEmail=collector@company.com

Upload a file to a document

Service

POST /apiB2B/Documents/Upload/{documentId}

Payload Properties

documentId: The id of the document

userType: User Type (1: Collector/2: Sender)

userId: User Id

userExternalRef: External Ref of user (Collector Only)

userEmail: user Email

deleteOld: True if old document should be deleted, otherwise false. (default is true)

description: Description for the uploaded document, can be null.

Example payload (without impersonation)

```
/apiB2B/Documents/Upload/95252
```

Example payload (impersonating Customer)

```
/apiB2B/Documents/Upload/95252?userType=2&userEmail=custname@custemail.com
```

Example payload (impersonating Collector)

```
/apiB2B/Documents/Upload/95252?userType=1&userEmail=collector@company.com
```

Comments:

To execute the call we need to perform a multi-part form Content Type.

Below a simple HTML example to test this

IMPORTANT – This is only for explanation purposes. Key should never be included on client side!

```
<!DOCTYPE html>
<html>
<body>

<h1>Upload Document Multipart form</h1>

<form method="POST" enctype="multipart/form-data" id="fileUploadForm">
<input type="file" name="files"/><br/><br/>
<input type="submit" value="Submit" id="btnSubmit"/>
</form>

<h1>Upload Document Result</h1>
<span id="result"></span>
```

```

<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.4/jquery.min.js">
</script>
<script>
$(document).ready(function () {

    $("#btnSubmit").click(function (event) {

        //stop submit the form, we will post it manually.
        event.preventDefault();

        // Get form
        var form = $('#fileUploadForm')[0];

        // Create an FormData object
        var data = new FormData(form);

        // If you want to add an extra field for the FormData
        data.append("CustomField", "This is some extra data, testing");

        // disabled the submit button
        $("#btnSubmit").prop("disabled", true);

        $.ajax({
            url:
"https://previewapi.ezidox.com/apiB2B/Documents/Upload/95252?userType=1&userEmail=collector@company.com",
            type: "POST",
            headers: {Authorization: "amx XXXX:YYYY"},
            data: data, // this will get all the input fields of your form.
            enctype: 'multipart/form-data',
            processData: false, // tell jQuery not to process the data
            contentType: false, // tell jQuery not to set contentType
            dataType: 'json', // as you want
            success: function(response) {
                // success
            }
        });
    });
});
</script>
</body>
</html>

```

Webhooks

Webhook allow your platform to be notified of ezidox events in real time.

Events currently available are:

- Document Accepted
- Document Uploaded
- Document Rejected
- Document Not Relevant
- Application Status Updated
- Application Created
- Document Comment
- Application Note Created
- Application Note Updated
- Application Note Removed

Setup

Currently webhooks can be activated and configured only via B2B Web Services – UI to manage them will be added.

Service – Activate Webhook

PUT/apiB2B/Hooks/Subscribe

Payload Properties

targetUrl: The endpoint URL

event: Event Type:

- document_accepted
- document_uploaded
- document_rejected
- document_irrelevant
- application_status_updated
- application_created
- document_comment
- application_note_created
- application_note_updated
- application_note_removed

Example payload

/apiB2B/Hooks/Subscribe

```
{  
  "target_url": "https://myendpoint.mycompany.com",
```

```
"event": "document_accepted"
}
```

Service – Deactivate Webhook

POST /apiB2B/Hooks/Unsubscribe

Payload Properties

id: Webook Id

targetUrl: The endpoint URL

event: Event Type:

- document_accepted
- document_uploaded
- document_rejected
- document_irrelevant
- application_status_updated
- application_created
- document_comment
- application_note_created
- application_note_updated
- application_note_removed

You can specify in alternative:

- id
OR
- event and targetUrl

Example payload

/apiB2B/Hooks/Unsubscribe

```
{
  "id": 10
}

{
  "targetUrl": "https://myendpoint.mycompany.com",
  "event": "document_accepted"
}
```

Service – Test Webhook

POST/apiB2B/Hooks/Test

Payload Properties

targetUrl: The endpoint URL

webhookKind: Event Type:

- document_accepted
- document_uploaded
- document_rejected
- document_irrelevant
- application_status_updated
- application_created
- document_comment
- application_note_created
- application_note_updated
- application_note_removed

Example payload

/apiB2B/Hooks/Test

```
{  
  "target_url": "https://myendpoint.mycompany.com",  
  "webhookKind": "document_accepted"  
}
```

B2B Webviews

ezidox allows the generation of webviews via B2B services, that can be embedded in your systems and flows.

Below the list with some example requests, more are available.

Setup

Please note: you must contact support@ezidox.com in order to enable webviews to be hosted from your website, in order to enable the CORS validation to your host website.

Collector Service–Application View

POST /apiB2B/Collectors/{id}/Views/Application/{applicationId}/Details

POST /apiB2B/Collectors/Ext/{collExternalRef}/Views/Application/Ext/{appExternalRef}/Details

Collector Service – Full Portal View

POST /apiB2B/Collectors/{id}/Views/Full

Collector Service – Documents Viewer (All)

POST /apiB2B/Collectors/{id}/Views/Application/{applicationId}/Documents

Collector Service – Documents Viewer (Individual)

POST /apiB2B/Collectors/{id}/Views/Application/{applicationId}/Documents/{documentId}

Collector Service – Create Application

POST /apiB2B/Collectors/{id}/Views/Application/Create

POST /apiB2B/Collectors/Ext/{externalRef}/Views/Application/Create

Customer Service – Documents Viewer (All)

POST /apiB2B/Stakeholders/{id}/Views/Application/{applicationId}/Documents

URL Parameters

Collectors/{id}: The id of the collector

Collectors/Ext/{collExternalRef}: The external Reference of the collector (assigned for Collectors created via B2B)

Application/{applicationId}: The id of the application

Application/Ext/{appExternalRef}: The external Reference of the application (assigned for Applications created via B2B)

Stakeholders/Ext/{staExternalRef}: The email address of the customer

Payload Properties

OneOff:Set to true if you don't want the user to be able to extend the session and re-login

SessionDurationTime:Duration of the session in minutes before expiring - if null it will use the default value

SessionIdleTime:Automatic session renewal timespan (in minutes), ignored if OneOff is set to true; if null it will use the default value

RedirectUrl:Url where the user will be redirected to when the session is expired or the page is closed

HideHeader:Set to true if you want to hide the page header

HideFooter:Set to true if you want to hide the page footer

IsIFrame:Set to true if the link is used in an iFrame (if you need to use generated links in IFrames, please contact us providing the hostname(s) that will embed it to enable them)

AdditionalParameters: Key/Value list of additional Parameters, listed below:

- **Application Level Parameters:**
 - referenceNo: mortgage application reference number provided by the lender
 - docsDueDate: when the docs are due in format dd/mm/yyyy
 - externalReference: application external reference
- **Primary Stakeholder:**
 - primaryEmail: email address
 - primaryMobile: mobile number
 - primaryName: name
 - primarySurname: surname
 - primaryType: stakeholder type
- **Additional Customers:**
 - stk1Name
 - stk1Surname
 - stk1Email
 - stk1Mobile
 - stk1Type
 - stk1Relationship
 - stk2Name
 - stk2Surname
 - stk2Email
 - stk2Mobile
 - stk2Type
 - stk2Relationship
 - stk3Name
 - stk3Surname
 - stk3Email
 - stk3Mobile
 - stk3Type
 - stk3Relationship

Customer Service – View Application – Anonymous/Registered
POST /apiB2B/Stakeholders/Views/Application/Details

Payload Properties

ApplicationId: The id of the application

AppExternalRef: The external Reference of the application (assigned for Applications created via B2B)

StaExternalRef: The external reference or email of the stakeholder

StakeholderId: The id of the customer

ShowAnonymousView: Set to true if you want anonymous view, false otherwise (registered user)

ViewOptions.OneOff: Set to true if you don't want the user to be able to extend the session and re-login

ViewOptions.SessionDurationTime: Duration of the session in minutes before expiring - if null it will use the default value

ViewOptions.SessionIdleTime: Automatic session renewal timespan (in minutes), ignored if OneOff is set to true; if null it will use the default value

ViewOptions.RedirectUrl: Url where the user will be redirected to when the session is expired or the page is closed

ViewOptions.HideHeader: Set to true if you want to hide the page header

ViewOptions.HideFooter: Set to true if you want to hide the page footer

ViewOptions.IsIFrame: Set to true if the link is used in an iFrame (if you need to use generated links in IFrames, please contact us providing the hostname(s) that will embed it to enable them)

Example payload

/apiB2B/Collectors/17890/Views/Application/Create

```
{
  "OneOff": false,
  "SessionDurationTime": 10,
  "SessionIdleTime": 2,
  "IsIFrame": false,
  "IsIFrame": false,
  "AdditionalParameters":
  {
    "referenceNo ":"ABC123",
    "externalReference ":"XYZ456"
  }
}
```

/apiB2B/Collectors/17890/Views/Application/Details

```
{
  "OneOff": false,
  "SessionDurationTime": 10,
  "SessionIdleTime": 2,
  "IsIFrame": false
}
```

/apiB2B/Stakeholders/Views/Application/Details

```
{
  "StaExternalRef": "test@test.com",
  "AppExternalRef": "ABC123",
  "ShowAnonymousView": true,
  "ViewOptions": {
    "OneOff": true,
    "HideHeader": true,
  }
}
```

```
    "HideFooter": true,  
    "IsIFrame": true  
  }  
}
```

The services will return a one-off URL that can be used in the iFrame.

```
<iframe id="myIframe" src="oneOffUrlLink"></iframe>
```

Using the functionality in <https://github.com/davidjbradshaw/iframe-resizer#typical-setup> you can automatically resize the iFrame, as per example below.

This package can be installed via NPM (npm install iframe-resizer --save).

```
<style>  
  iframe {  
    width: 1px;  
    min-width: 100%;  
  }  
</style>  
<iframe id="myIframe" src="oneOffUrlLink"></iframe>  
<script>  
  iFrameResize({ log: false }, '#myIframe')  
</script>
```

TemplateSet Services

ezidox allows to retrieve the list of TemplateSet available for use. Only the TemplateSet shared at company level are returned.

Service – Search TemplateSets

GET /apiB2B/TemplateSets/Search?nameFilter={value}

URL Parameters

nameFilter: The name or part of the TemplateSet name

Service – Get Individual TemplateSet

GET /apiB2B/TemplateSets/{id}&populateGroupAndTemplates=true

URL Parameters

id: The id of the TemplateSet

populateGroupAndTemplates: Populate the list of groups and templates in the TemplateSet; default to false (optional)

Templates Services

ezidox allows to retrieve the list of Templates available for use. Only the Templates shared at company level are returned.

Service – Search Template

GET /apiB2B/Templates/Search?nameFilter={value}

URL Parameters

nameFilter: The name or part of the Template name

Service – Get Individual Template

GET /apiB2B/Templates/{id}

URL Parameters

id: The id of the Template