API Documentation - Addo Web Service

(last updated 2021-02-26)

1. Introduction

1.1. Purpose

This document describes Addo Web Service and explains how to consume it.

1.2. Allowed HTTPs requests

POST - used to update resource and **GET** - used to get a resource or list of resources

1.3. Description of server responses

200 OK - the request was successful (some API calls may return 201 instead).

400 Bad Request - the request could not be understood or was missing required parameters.

401 Unauthorized - authentication failed or user doesn't have permissions for requested operation.

500 Internal Server Error - server encountered an unexpected condition that prevented it from fulfilling the request.

1.4. WSDL document

The WSDL document is supplied as a separate xml document named "SigningService.svc.wsdl.xml". The service exposes access to the wsdl document through metadata exchange.

Demo environment link - https://demo.vismaaddo.net/webService/v2.0/signingservice.svc

Production environment link - https://vismaaddo.net/webService/v2.0/signingservice.svc

Example application (.NET) can be downloaded from SOAP Example Application

1.5. List of API methods

Here you can find current used API methods:

Demo environment - https://demo.vismaaddo.net/webService/v2.0/restsigningservice.svc/help

Production environment - https://vismaaddo.net/webService/v2.0/restsigningservice.svc/help

2. Usage of Addo API v2.3

2.1. REST via Postman

Download Postman tool via link - https://www.getpostman.com/apps

To get your hash for password, use tools like this one: https://hash.online-convert.com/sha512-generator. Select **base64** for your password hash:

Request usage:

POST - [Base address]/[Public method name]

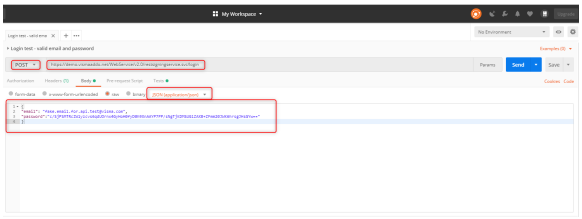
GET - [Base address]/[Public method name]?[Parameters]

2.1.1. POST request via Postman

Using Postman:

To login using REST service you need to call web service at address:
https://demo.vismaaddo.net/WebService/v2.0/restsigningservice.svc/login

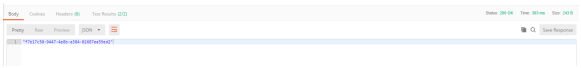
Use method **POST** and set content-type as **application/json**.



Code used here:

```
{ "email": "fake.email.for.api.test@visma.com", "password":"c
/SjPSMTRcZW1yzcvs6qdUOrnx4GyHoH0fyD0h9XnAAYP7PP
/sNgTjKDMSUGlZAXB+ZFmm20JWK6hrsgJHsGYw==" }
```

Below is a screenshot of **POST request** and response as described above:



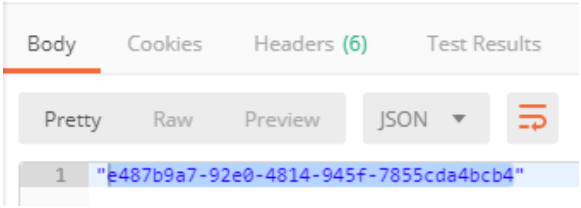
Server response of correct login is visible at the bottom with string in the body of **token** and **200 OK** status

2.1.2. GET request via Postman

Here is an example of a method with **GET request**, showing how parameters are passed through URL.

In Postman, we are using **GetRejectionComment** method (list of methods available here List of API methods). This requires us to pass **signingToken** and **token** parameters.

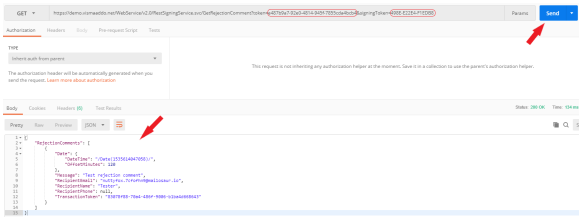
You can get the token after using POST for login example in previous paragraph, it's available in the response field "Body":



GetSigningStatus or GetSigning responses contains signing token.

Now we pass these parameters as a link in Postman and receive a response:

https://demo.vismaaddo.net/WebService/v2.0/RestSigningService.svc/GetRejectionComment?token=e487b9a7-92e0-4814-945f-7855cda4bcb4&signingToken=498E-E22E4-F1EDB8



2.2. Public methods

There are currently two Demo and Production environments, which might differ in variety of supported methods. Please take a look at the List of API methods for each environment before consuming our API.

Public methods available for REST API:

DEMO - https://demo.vismaaddo.net/WebService/v2.0/RestSigningService.svc/help

PRODUCTION - https://vismaaddo.net/WebService/v2.0/RestSigningService.svc/help

Base address for the service:

DEMO - https://demo.vismaaddo.net/WebService/v2.0/RestSigningService.svc

PRODUCTION - https://vismaaddo.net/WebService/v2.0/RestSigningService.svc

WCF (SOAP) available methods:

- *Guid Login(string email, string password);*
- *InitiateSigningResponse InitiateSigning(Guid token, InitiateSigningRequest request, TemplateOverride templateOverride);*
- *InitiateSigningSynchronouslyResponse InitiateSigningSynchronously(Guid token, InitiateSigningSynchronouslyRequest request, TemplateOverride templateOverride);*
- *GenworderateDocumentResponse GenerateDocument(Guid token, GenerateDocumentRequest request);*
- *GetSigningTemplatesResponse GetSigningTemplates(Guid token);*
- *GetSigningResponse GetSigning(Guid token, string signingToken);*
- *GetSigningStatusResponse GetSigningStatus(Guid token, string signingToken);*
- *StartCampaignSigningResponse StartCampaignSigning(Guid token, StartCampaignSigningRequest request);*
- *MergeDocumentsResponse MergeDocuments(Guid token, MergeDocumentsRequest request);*
- *void InitiateCampaign(Guid token, InitiateCampaignRequest request, TemplateOverride templateOverride)*
- *GetAccountInfo GetAccountInfo(Guid token)*
- *CancelSigningResponse CancelSigning(Guid token, string signingToken)*
- *string GenerateDocumentWithMergeData(Guid token, GenerateDocumentWithMergeDataRequest request, bool encryptDocument = false, string encryptionKey = null)*
- *GetTransactionDetailsResponse GetTransactionDetails(Guid token, Guid transactionToken)*
- *void CancelTransaction(Guid token, Guid transactionToken)Base address for service*
- *void ReactivateTransaction(Guid token, Guid transactionToken)*
- *void UpdateTransaction(Guid token, UpdateTransactionRequest updateTransactionRequest)*
- *void SaveRecipient(Guid token, Contact contact);*

- *void DeleteRecipient(Guid token, Contact contact);*
- *FileData ExportRecipients(Guid token, string separator = null);*
- *void ImportRecipients(Guid token, byte[] data, string separator = null);*
- *GetTemplateMessagesResponse GetTemplateMessages(Guid token, Guid templateId);*
- *GetCampaignsResponse GetCampaigns(Guid token, string externalReference);*
- *GetRejectionCommentResponse GetRejectionComment(Guid token,string signingToken);*
- *GroupResponse CreateGroup(Guid token, CreateGroupRequest request);*
- *GetGroupsResponse GetGroups(Guid token);*
- *void AddUserToGroup(Guid token, AddUserToGroupRequest request);*
- *CreateUserResponse CreateUser(Guid token, CreateUserRequest request);*
- *CreateAccountResponse CreateAccount(Guid token, CreateAccountRequest request);*

2.2.1. Login

Technical description

- A service consumer must call the Login method prior to any other operation.
- The credentials provided to the Login method must match a valid Addo user email as string and password hashed as SHA512 hash bytes to base64 string.
- In case of success the Login method will return a valid security token (type of Guid) that must be supplied subsequently in calls to the service methods.
- In case of authentication failure the Login method will return empty invalid security token (type of Guid).
- If email or password strings are null the service will throw a FaultException.
- If password is expired or account is disabled appropriate FaultException will be thrown.
- The token received on successful login is valid for a given time (00:05:00).
- Each time the token is used in an operation on the service this valid time window is refreshed.

Parameters

Name	Type	Description
email	string	Email of Addo user
password	string	SHA512 hash of Password as **base64** string of an Addo user.

Example

Request	Response

{
 	
"email": "fake.email.for.api.test@visma.com",
"password":"c/SjPSMTRcZW1yzcvs6qdUOrnx4GyHoH0fyD0h9XnAAYP7PP/sNgTjKDMSUGlZAXB+ZFmm20JWK6hrsgJHsGYw=="
}
	77d7f36e-3a90-462f-8d61-ee2b8a1f408d

2.2.2. InitiateSigning

Technical description

This method initiates a new Addo signing process.

An InitiateSigningRequest object must be supplied containing data must be supplied containing information needed for about how to create the signing along with a full signing dataset describing participating customers and their data.

The request also specifies signing template id which will be used while configuring signing.

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.

request	InitiateSigningRequest	Contains data about how to create signing.
templateOverride	TemplateOverride	Optional. Used in order to replace default account configuration presets (last reminder, signing method etc.).

Response

Method gives response (type of InitiateSigningResponse). Information if signing failed being initiated could be found there.

Types

Type	Description
Guid	A unique set of symbols, necessary for identification.
InitiateSigningResponse	Information if signing failed being initiated could be found there. Contains property: SigningToken (type of string).
InitiateSigningRequest	Contains properties: Name (type of string) to set signing name, StartDate (type of DateTime) to set starting date, SigningData (type of Signing) to set required parameters regarding documents, recipients, enclosures and signing sequence when initiating new signing, SigningTemplateId(type of Guid), DistributionUrl (type of string) to set callback url when signing is completed, RejectionUrl (type of string) to set callback url when signing is rejected, ExpirationUrl (type of string) to set callback url when signing expires, DocumentsSignedUrl (type of string) to set callback url when documents are signed, TransactionStateChangedUrl (type of string) to set callback url for all transaction state changes.
TemplateOverride	Contains properties: Duration (type of TimeSpan), MessageTypes (array of MessageTypeEnum), NotificationInterval (type of NotificationIntervalEnum), LastReminder (type of LastReminderEnum), SigningMethod (type of SigningMethodEnum), DocumentEncryption (type of DocumentEncryptionEnum), ShowSignerName (type of bool), ShowDate (type of bool) and CustomMessageTemplates (array of MessageTemplate), AuthenticationMethod (type of AuthenticationMethod), NotificationIntervalDays (type of int)
MessageTemplate	Used for sending customized messages (overrides default messages) for recipients. Contains properties: DocumentType (type of MessageTemplateType), UsageType (type of TemplateUsageType), Subject (type of string) used for email and Body (type of string) used for email or sms message.
Signing	Contains properties: Recipients (array of RecipientData) to define recients' data such as name, phone etc., Sender allows you to have sender name, email and company name other than your account name (type of SenderData) Documents (array of Document) to use these pregenerated documents if it is necessary while initiating signing, EnclosureDocuments (array of Document), SigningSequence (type of SigningSequenceOrder), ReferenceNumber (type of string), Sender (type of SenderData), SenderComment (type of string), AllowInboundEnclosures (type of bool), AllowRecipientComment (type of bool), ExternalReferenceId (type of string).
RecipientData	Information about the recipient. Contains properties: Id (type of Guid) relates to SigningSequence, Cpr (type of string), Name (type of string), PID (type of string), Address (type of string), Email (type of string), Phone (type of string), CVR (type of string), SSN (type of string), TupasSsn (type of string), TemplateData (type of TemplateDataContainer), SendWelcomeNotification (type bool?, true by default) which sets whether invitation mail /sms has to be sent to a recipient, SendDistributionNotification (type of bool?, true by default) which sets whether distribution mail/sms has to be sent to a recipient, SendDistributionDocument (type of bool?, true by default) which sets whether a receipt massage should include documents, SigningMethod (type of SigningMethodEnum?, if not set inherited from signing template), AuthenticationMethod (type of AuthenticationMethodEnum?, if not set inherited from signing template), Title (type of string), DistributionMethod (type of DistributionMethodEnum?, setting if recipient will get documents and how it will be sent).
SenderData	Information about the sender. Name (type of string) CompanyName (type of string) Email (type of string) This information is used only for display purpose, no data will be sent to this email address.
Document	Contains properties: Id (type of Guid) relates to SigningSequence, Data (type of string) which is encoded as base64 string, Name (type of string), MimeType (type of string), IsShared (type of bool) which tells if the document is shared between few recipients.

SequenceOrder	Defines order how the documents will be signed by recipients. Contains property: SigningSequenceItems (array of SequenceItem).
SequenceItem	States sequence number for a recipient. Contains properties: RecipientId (type of Guid) and SequenceNumber (type of int).
TemplateDataContainer	Used for pdf generation using XSLT templates. Contains property Items (list of TemplateDataItem).
TemplateDataItem	Represents XML. Contains properties: Name (type of string) which represents XML tag name, Value (type of string) which represents value inside that element and Items (list of TemplateDataItem) which contains nesting XML elements.
MessageTemplateType	Can have values: *Sms* (1), *Email* (2).
TemplateUsageType	Describes for what purpose the template message will be used. Can have values: *StartMessage* (0), *RemindMessage*(1), *Distribution* (4), CampaignStartMessage(5), CampaignRemindMessage(6).
MessageTypeEnum	Can have values: *None* (0), *Mail* (1), *SMS* (2), Eboks (3) (used for e-Boks private), SafeCloudPrivate (4), EboksCompany (5), SafeCloudCompany(6). Describes on which channel the campaign will send out messages to a recipient.
SigningMethodEnum	Can have values: *NemId* (1), *Stylus* (2), *BankID* (3), *NemIDMoces* (4), *SwedishBankID* (5), *Accept* (6), NoBankIDMobile (7 Depricated, same as BankId), NemIdPocesNoSsn (deprecated, does the same as NemId), FinnishTupas (9), Siths (10). Describes how the documents of the signing must be signed.
NotificationIntervalEnum	Can have values: *DontSend* (1), *EveryWeek (2), EveryTwoWeeks* (3), *EveryThreeWeeks* (4). Describes the frequency with which the campaign will send out notifications to its recipients.
LastReminderEnum	Can have values: *DontSend* (1), *DayBefore* (2), *TwoDaysBefore* (3). Describes when to send out last reminder to recipients of the signing.
DocumentEncryptionEnum	Can have values: *None* (1), *Encrypted* (2), Passcode (3). Specifies whether encryption is enabled for signing documents when distributing.
AuthenticationMethodEnum	Can have value: None (0), NemIdPrivate (1), NemIdPrivateNoSsn (deprecated, does the same as NemIdPrivate), TwoFactorVerification(3), Siths(4), NemIdEmployee(7), NorwegianBankId(8), SwedishBankId (9), NorwegianBankIdMobile(10), SecretCode(11), FinnishTupas(13), Freja(14), Idin(15), Eherkenning(16). Specifies which method should use on authentication before signing documents.
DistributionMethodEnum	Can have value: None (1), MessageWithDocuments (2), MessageWithoutDocuments (3) Eboks (4)
Callback	Information about callback: Url (type of string) Type (type of CallbackTypeEnum)
CallbackTypeEnum	Can have values: Distribution (1), Expiration (2), Rejection (3), DocumentOpened (4), DocumentSigned (5), TransactionStateChanged (6), Redirect (7)

Properties of the types

Type	Property	Type	Description
InitiateSigningResponse	SigningToken	string	Token of signing, unique identification.
	Name	string	Defines name of a signing. Mandatory, Max 100 chars.

InitiateSigningRequest	StartDate	DateTime	Defines when a signing will start. StartDate value cannot be 24 hours before request is created. Mandatory. Uses UNIX date format like "/Date (1573638173002)/"
	SigningData	Signing	Contains data about signing.
	SigningTemplateId	Guid	Defines a Guid which is associated with a template settings of a signing (duration, message type, notification interval, signing method etc.).
	DistributionUrl (obsolete)	string	Distribution callback url. On distribution for the provided url it will add query string parameter "signingToken"
	RejectionUrl (obsolete)	string	Rejection callback url. On rejection for the provided url it will add query string parameter "signingToken"
	ExpirationUrl (obsolete)	string	Expiration callback url. On expiration for the provided url it will add query string parameter "signingToken"
	DocumentSignedUrl (obsolete)	string	Document signed callback url. On document signed for the provided url it will add query string parameters "signingToken", "transactionToken", "externalDocumentId"
	DocumentOpenedUrl (obsolete)	string	Document opened callback url. On document opened for the provided url it will add query string parameters "signingToken", "transactionToken", "externalDocumentId"
	TransactionStateChangedUrl (obsolete)	string	Transaction state changed callback url. On transaction state change for the provided url it will add query string parameters "signingToken", "transactionToken", "referenceNumber", "transactionStateId"
	Callbacks	Callback[]	An array of callbacks. Explicit properties of the callbacks are obsolete.
TemplateOverride	Duration	TimeSpan?	Describes life time of the signing, must be longer than a day. Please see below in section 4 (Notes).
	MessageTypes	MessageTypeEnum[]	Describes on which channel the signing will send out messages to a recipient: *Mail* (1) and/or *SMS* (2).
	NotificationInterval	NotificationIntervalEnum?	Describes the frequency with which the signing will send out notifications to its recipient: *DontSend* (1), *EveryWeek* (2), *EveryTwoWeeks* (3), *EveryThreeWeeks* (4).
	LastReminder	LastReminderEnum?	Describes when to send out last reminder to recipients of the signing: *DontSend* (1), *DayBefore* (2), *TwoDaysBefore* (3).
	SigningMethod	SigningMethodEnum?	Describes how the documents of the signing must be signed: *NemId* (1), *Stylus* (2), *BankID* (3), *NemIDMoces* (4), *SwedishBankID* (5), *Accept* (6), *NoBankIDMobile* (7, depricted, same as BankId), *NemIdPocesNoSsn* (deprecated, does the same as NemId), *FinnishTupas* (9), Siths (10).
	DocumentEncryption	DocumentEncryptionEnum?	Specifies whether encryption is enabled for signing documents when distributing: *None* (1), *Encrypted* (2), *Passcode* (3).
	SmsVerification	bool?	[DEPRECATED, use authentication method] Specifies whether signing uses SMS verification when logging in.
	ShowSignerName	bool?	Specifies whether signer name should be displayed below signature (applies only for Touch signings)
	ShowDate	bool?	Specifies whether signing date should be displayed below signature (applies only for Touch signings)
	CustomMessageTemplates	MessageTemplate[]	Defines messages templates which would override default message templates. Message template contains subject, body text, usage type (ex. distribution message) and type how it will be distributed (ex. SMS).
	AuthenticationMethod	AuthenticationMethodEnum?	Authentication method when accessing transaction in SigningPortal. Possible values: None (0), NemIdPrivate (1), NemIdPrivateNoSsn (deprecated, does the same as NemIdPrivate), TwoFactorVerification(3).
	NotificationIntervalDays	int?	Notification interval in specific days count (not from enum).
MessageTemplate	Subject	string	Subject used for email message. Max 300 chars.
	Body	string	Message body. Max 1000 chars.
	UsageType	TemplateUsageType	What kind of message it is. Can be: StartMessage (0), RemindMessage (1), Distribution (4), CampaignStartMessage(5) for data campaign invitation messages, CampaignRemindMessage(6)

	MessageTemplateType	MessageTemplateType	Can be: Sms (1), Email (2). The value of this property is no longer in use since a signing can have recipients with different message types so the same text is used for all messages types.
Signing	Recipients	RecipientData[]	Contains recipients which are associated with the signing. Can be empty (but not null) if the signing template already has recipients defined.
	Documents	Document[]	Specifies documents which could be used for signing.
	EnclosureDocuments	Document[]	Contains enclosure documents which are added to signing.
	SigningSequence	SequenceOrder	Defines signing sequence of recipients to sign documents.
	ReferenceNumber	string	Defines an optional reference number for transactions. Max 36 chars.
	Sender	SenderData	Optional. Defines sender information. Contains properties name, email and company name to be displayed in emails and signing portal.
	SenderComment	string	Optional. Defines comment which is shown to recipient. Max 2000 chars.
	AllowInboundEnclosures	bool	Specifies whether inbound (recipient) enclosures are allowed.
	AllowRecipientComment	bool	Specifies whether a recipient can leave a comment.
	ExternalReferenceId	string	An identifier that could be used for integrations.
	BccRecipients	BccRecipientData[]	Specifies recipients which receive distribution email as BCC.
RecipientData	Id	Guid	Unique Guid to identify recipient (optional and used to match with SequenceItem).
	Cpr	string	Defines Cpr (personal identification) number. Max 10 chars
	PID	string	NemID public ID. Max 20
	Name	string	Defines name of the recipient. Max 255 chars
	Address	string	Defines address of the recipient. Max 150 chars
	Email	string	Defines email of the recipient. He will receive letters from the web service in this email. Max 255 chars
	Phone	string	Defines phone of the recipient. He will receive SMS from the web service in this phone, if distribution is set to SMS. Max 50 chars
	SignedDate	DateTimeOffset	Defines when the document was signed by the user (not specified by creator of the signing)
	CVR	string	Company identification ID. Max 12 chars.
	SSN	string	Social security number. Max 11 chars.
	TupasSsn	string	SSN for Tupas signing.
	TemplateDataContainer	TemplateData	Defines template for generating document (using XSLT transformation).
	SendWelcomeNotification	bool	If Null default True.
	SendDistributionNotification	bool	If Null default True. **OBSOLETE, use DistributionMethod instead**
	SendDistributionDocument	bool	If Null default True. **OBSOLETE, use DistributionMethod instead**
	SigningMethod	SigningMethodEnum?	Can be Null. Defines Recipient signing method, overrides selected Template signing method. If not defined (Null) Template signing method will be used
	AuthenticationMethod	AuthenticationMethodEnum?	Can be Null. Defines Recipient authentication method, overrides Template authentication method. If not defined (Null) Template authentication method is used

	Title	string	Recipient title. Max 255 chars.
	DistributionMethod	DistributionMethodEnum?	Can be null. Defines Recipient distribution method, overwrites values from template and fields SendDistributionNotification and SendDistributionDocument
	AllowApproveAllAttachments	bool	Defines if recipient will be able to approve all attachments at once
	IsQuestionnaireEnabled	bool	Defines if recipient will be asked to fill answers for questionnaire
	Secret	string	Secret is mandatory only when using authentication method SecretCcode. Max 255 characters
	NotificationMethods	MessageTypeEnum[]	Can be Null. Defines Recipient notification methods, overrides Template MessageTypes. If not defined (Null) Template MessageTypes is used.
	AdditionalData	string	Optional. Field to store any data per Recipient. Limited to 4000 characters
Document	Id	Guid	Identifier of the document, should be passed. Related to signing sequence association
	Data	string	Document data encoded as base64 string. Mandatory.
	Name	string	Document name. Mandatory. Max 255 chars.
	MimeType	string	Identifies what kind is the kind of document (ex. "PDF"). Max 255 chars.
	IsShared	bool	DEPRECATED. States whether document is shared between recipients.
SequenceOrder	SigningSequenceItems	SequenceItem[]	Contains signing sequence items.
SequenceItem	RecipientId	Guid	States for which recipient the sequence number applies.
	SequenceNumber	int	States sequence number for a recipient.
TemplateDataContainer	Items	TemplateDataItem[]	Used for pdf generation using XSLT templates.
TemplateDataItem	Name	string	Name of an XML tag.
	Value	string	Value of an XML element.
	Items	TemplateDataItem[]	Inner XML elements.
BccRecipientData	Name	string	Max chars 255.
	Email	string	Email of bcc recipient.
SenderData	Name	string	Senders name to display. Max chars 255
	Email	string	Senders email to display. Max chars 255
	CompanyName	string	Senders company name to display. Max chars 255
Callback	Type	CallbackTypeEnum	Cannot be null. Defines the type of the callback.
	Url	string	Cannot be empty. Defines the url of the callback.

Example

Request	Response

{
"token": "126d6f15-997e-4280-a670-e30fec346392",
"request": {
"DistributionUrl": null,
"DocumentSignedUrl": null,
"DocumentOpenedUrl": null,
"ExpirationUrl": null,
	{
	"SigningToken": "753-1584AE-102882"
	}

```
    "Name": "test signing",
    "RejectionUrl": null,
    "SigningData": {
      "AllowInboundEnclosures": true,
      "AllowRecipientComment": true,
      "BccRecipients": [],
      "Documents": [
        {
          "Data": "base64stringOfFile",
          "Id": "1fe93abb-5363-4eb5-a89f-
5210d38471d6",
          "IsShared": false,
          "MimeType": "PDF",
          "Name": "test.pdf"
        }
      ],
      "EnclosureDocuments": null,
      "ExternalReferenceId": null,
      "Recipients": [
        {
          "Address": null,
          "AuthenticationMethod": null,
          "CVR": null,
          "Cpr": "0101987654",
          "Email": "test_recipient1@visma.
com",
          "Id": "4c24818a-eae2-4eb0-aa2e-
f77ab355f77d",
          "Name": "TestName1",
          "PID": null,
          "Phone": null,
          "SSN": null,
          "SendDistributionDocument": null,
          "SendDistributionNotification":
true,
          "SendWelcomeNotification": true,
          "SigningMethod": 2,
          "TemplateData": {
            "Items": []
          },
          "Title": null,
          "TupasSsn": null
        },
        {
          "Address": null,
          "AuthenticationMethod": null,
          "CVR": null,
          "Cpr": "0101987654",
          "Email": "test_recipient2@visma.
com",
          "Id": "15271303-95a6-4865-b03f-
90582549697e",
          "Name": "TestName2",
          "PID": null,
          "Phone": "+123456789",
          "SSN": null,
          "SendDistributionDocument": null,
          "SendDistributionNotification":
true,
          "SendWelcomeNotification": true,
          "SigningMethod": 2,
          "TemplateData": {
            "Items": []
          },
          "Title": null,
          "TupasSsn": null
        }
      ],
      "ReferenceNumber": "666",
      "Sender": {
        "CompanyName": "SenderCompanyName",
```

```
        "Email": "sender.email@visma.com",
        "Name": "SenderTestName",
        "Phone": null
      },
      "SenderComment": "this is simple
comment",
      "SigningSequence": {
        "SigningSequenceItems": [
          {
            "RecipientId": "4c24818a-eae2-
4eb0-aa2e-f77ab355f77d",
            "SequenceNumber": 2
          },
          {
            "RecipientId": "15271303-95a6-
4865-b03f-90582549697e",
            "SequenceNumber": 1
          }
        ]
      }
    },
    "SigningTemplateId": "95b29391-c23b-46d2-
af51-ff9caa8edd9b",
    "StartDate": "/Date(1530791625)/",
    "Callbacks": [
        {
            "Url": "http://www.
distributionurl.com",
            "Type": 1
        },
        {
            "Url": "http://www.redirecturl.
com",
            "Type": 7
        }
    ],
  }
}
```


2.2.3. GenerateDocument

Technical description

- This method generates documents for signing by using XSLT templates.
- In the request a document template name or document template itself must be supplied along with a signing dataset which contains info about the recipients.
- The template determines how the document will be generated.
- The signing data describes which recipients the service must generate documents to, also describes each recipients' data.
- The method returns a response which contains validation data, generated documents.

Parameters

Name	Type	Description
token	Guid	Unique identifier to identify user.
request	GenerateDocumentRequest	Data which will be used when generating document.

Properties of the types

Type	Property	Type	Description
GenerateDocumentRequest	DocumentTemplate	string	XSLT format template
	RecipientData	RecipientData[]	Described in InitiateSigning. TemplateData is required at this point for DocumentTemplate XSLT data fields.
GenerateDocumentResponse	GeneratedDocuments	Document[]	Contains generated documents.
Document	RecipientId	Guid	Specifies with which recipient the document is associated.
	Data	byte[]	Contains the document data in byte array.

2.2.4. GetSigningTemplates

Technical description

This method allows to get signing templates of the user. It takes as a parameter user token and returns GetSigningTemplatesResponse with signing templates inside. This method could be accessed through both SOAP.

Properties of types

Type	Property	Type	Description
GetSigningTemplatesResponse	SigningTemplateItems	SigningTemplate[]	Contains SigningTemplate objects.
SigningTemplate	Id	Guid	Unique Guid to identify signing template.
	FriendlyName	String	Specifies a name for the signing template.
	Duration	TimeSpan	Defines duration of the signing.
	MessageTypes	MessageTypeEnum[]	See above in the *InitiateSigning* description.
	LastReminder	LastReminderEnum	
	SigningMethod	SigningMethodEnum	
	DocumentEncryption	DocumentEncryptionEnum	
	AuthenticationMethod	AuthenticationMethod	Described above.
	DistributionMethod	DeliveryMethodEnum	Described below.
	ShowSignerName	Bool	Specifies whether signer name should be displayed below signature (applies only for Touch signings)
	ShowDate	Bool	Specifies whether signing date should be displayed below signature (applies only for Touch signings)
	NotificationIntervalDays	Int	Value in days describes after how much days reminder should be sent repeatedly until signing is expired.
	LastReminderDay	Int	Value in days, describes when the last reminder is sent before a signing expires
	Recipients	RecipientData[]	Predefined recipients. RecipientData described above.
	BccRecipients	BccRecipientData[]	Predefined bcc recipients. BccRecipientData described above.
	SigningTemplateActions	SigningTemplateAction[]	Contains SigningTemplateAction objects.
SigningTemplateAction	MethodType	ActionTypeEnum	All values are reserved for future except SenderDistribution=5. Note that sender distribution may be different from DistributionMethod value above, which is for recipients.

	Method	DeliveryMethodEnum	Described below.
	MethodParameter	Int?	Reserved for future.

Enum values

Type	Name	Value
DeliveryMethodEnum	None	0
	Email	1
	EboksPrivate	5
	GoogleDrive	6
	Sftp	7
	CommonEmail	8
	VismaHrm	9
	VismaHr	10
	SafeCloudPrivate	11
	SafeCloudCompany	12
	EboksCompany	13

2.2.5. GetSigning

Technical description
This method returns info about the signing including recipients, documents etc.

Parameters

Name	Type	Description
token	Guid	Unique identifier to identify user.
signingToken	string	Token for identifying signing.

TemplateDataContainer

Properties of types

Type	Property	Type	Description
GetSigningResponse	Name	string	Name of the signing.
	SigningToken	string	Token of the signing.
	Recipients	Recipient[]	Contains recipient's info.
	Enclosures	Enclosure[]	Contains signing enclosure d
	ReferenceNumber	string	Contains optional reference
	Documents	SigningDocument[]	Contains signing documents
	CreatedOn	DateTime	Date of signing creation
	ExpiresOn	DateTime	Date of signing expiration
Recipient	Name	string	Contains Full Name of recipi
	XmlData	string	Contains XML data of the sig
	CustomId	Guid	Recipient Id which was optio
	SignatureSubject	string	Subject (signer name) extrac
	SignatureIdentifier	string	Signer identifier (PID/RID/etc

	Title	string	Recipient's title
	AdditionalData	string	Additional data that was pas
Document	Name	string	Name of the document.
	Data	byte[]	Data of the document in byte
	MimeType	string	Mime type of the document.
Enclosure(inherits from document)	Name	string	Name of the document.
	Data	byte[]	Data of the document in byte
	MimeType	string	Mime type of the document.
	IsInbound	bool	If true Enclore is was upload
	IsApproved	bool	Is enclosure approved in sig
SigningDocument(inherits from document)	Name	string	Name of the document.
	Data	byte[]	Data of the document in byte
	MimeType	string	Mime type of the document.
	XmlDSig	string	Contains XML digital signatu
	FlowId	int	Corresponding document flo
	State	DocumentFlowStateEnum	State of document. Possible
BccRecipient	Name	string	Contains Full Name of bcc re
	Email	string	Email of bcc recipient

Example

Request	Response

https://demo.vismaaddo.net/WebService/v2.0/restsigningservice.svc/GetSigning?signingToken=48C5-127B8C-F34D82&token=664dfc42-c5fd-4046-bf7d-731bdd04f12c	{
	"BccRecipients": [],
	
	"CampaignExternalReference": null,
	"Documents": [
	{
	"Data": [
	37,
	80,
	68,
	70,
	45,
	49,
	46,
	55,
	10
	],
	"MimeType": "application/pdf",
	"Name": "test_doc.pdf",
	"FlowId": 145818,
	"Id": "a9828bf3-fbee-4dac-80a0-6e7f2dc9b022",
	"XmlDSig": null
	}
	],
	"Enclosures": [],
	"Name": "Transaction with E-mail/Web Signature",
	"Recipients": [
	{
	"Name": "Test Name",
	"XmlData": null
	}
	],
	
	"ReferenceNumber": null,
	"SigningToken": "48C5-127B8C-F34D82"
	}

2.2.6. GetSigningStatus

Technical description

This method returns info about the signing including recipients, documents etc.

Parameters

Name	Type	Description
token	Guid	Unique identifier to identify user.

signingToken	string	Token for identifying signing.

Properties of types

Type	Property	Type	Description
GetSigningStatus	State	SigningStateEnum	Signing state
	Recipients	Recipient[]	Contains Recipients
	Transactions	Transaction[]	Contains signing transactions
	SigningToken	string	Token of the signing.
	CreatedOn	DateTime	Date of signing creation
	ExpiresOn	DateTime	Date of signing expiration
	ReferenceNumber	string	Reference number
Recipient	Name	string	Full Name of Recipient
	Title	string	Title of Recipient
	Email	string	Email of Recipient
	AdditionalData	string	Additional data of Recipient
	Transactions	Transactions[]	Contains Transactions
Transactions	TransactionToken	Guid	Token of the transaction.
	State	TransactionStateEnum	Describes transaction state

Enum values

Type	Value	Description
SigningStateEnum	*Failed (-1)*	Error occurred.
	Created (1)	Signing is created
	Started (2)	Signing reached his signing start date and sent notifications to recipients.
	Completed (3)	Finished distribution. Signing is completed.
	Expired (4)	Reached selected signing template expiration date
	Stopped (5)	Stopped by user
	CampaignStarted (6)	Related to Campaigns
	Rejected (7)	Recipient rejected to sign document
TransactionStateEnum	*Failed (-1)*	Error occurred.
	Created (1)	When transaction is created, and signing start date is not due.
	Pending (2)	Related to sequential signing. First document in sequence order is stated to Started as rest of them to Pending state.
	Started (3)	Signing start date is due and ready to sign document.
	Signed (4)	Recipient signed document - others may not have done that.
	ReadyForDistribution (5)	Start distributing transactions to recipients, and sender.
	Completed (6)	Finished distribution. Signing is completed.
	Expired (7)	Reached signing expiration date.
	Canceled (5)	Canceled by user.
	DataCollecting (9)	Relates to data campaigns. After campaign is initiated but not yet started, waiting for merged documents.

	Rejected (10)	Recipient rejected signing the document.

Example

Request

https://demo.vismaaddo.net/WebService/v2.0/restsigningservice.svc/GetSigningStatus?signingToken=48C5-127B8C-F34D82&token=664dfc42-c5fd-4046-bf7d-731bdd04f12c

Response


```
{
    "CreatedOn": "/Date(1582565108499)/",
    "ExpiresOn": "/Date(1583774701908)/",
    "Recipients": [
        {
            "AdditionalData": "additional data string to Test!!!",
            "Email": "karolis.vezelis@visma.com",
            "Name": "Carl",
            "Title": "Head manager",
            "Transactions": [
                {
                    "State": 3,
                    "TransactionToken": "95d7e929-f18e-4657-9ac7-
d9fb66beeb29"
                }
            ]
        },
        {
            "AdditionalData": null,
            "Email": "john.johnson@visma.com",
            "Name": "John",
            "Title": "System administrator",
            "Transactions": [
                {
                    "State": 2,
                    "TransactionToken": "68e71a96-37c8-42b8-8f1e-
24a2ae7a4840"
                }
            ]
        }
    ],
    "ReferenceNumber": "666",
    "SigningToken": "4F00-1D5FC2-D9F6B7",
    "State": 2,
    "Transactions": [
        {
            "State": 3,
            "TransactionToken": "95d7e929-f18e-4657-9ac7-d9fb66beeb29"
        },
        {
            "State": 2,
            "TransactionToken": "68e71a96-37c8-42b8-8f1e-24a2ae7a4840"
        }
    ]
}
```


2.2.7. StartCampaignSigning

Technical description

This method starts signing of a campaign which was created earlier by accepting merged documents (using mail merge functionality).

Parameters

Name	Type	Description
token	Guid	Unique identifier to identify user.
request	StartCampaignSigningRequest	Contains signing token and merged documents.

Response
Method gives response (type of StartCampaignSigningResponse) with information if the call had any errors and signing, transaction tokens.

Properties of types

Type	Property	Type	Description
StartCampaignSigningRequest	SigningToken	string	Unique token of the signing.
	MergedDocuments	MergeDocument[]	Array of merged documents.
	EnclosureDocuments	EnclosureDocument[]	Array of enclosure documents.
	TemplateData	TemplateDataContainer	Data, which will be converted as an recipients form data as XML. (Will be able to see that in GetSigning response near Recipient model)
MergeDocument	ContentType	string	Content type of the document. Mandatory. Max 255 chars.
	Name	string	Name of the document. Mandatory. Max 255 chars.
	Data	byte[]	Contains PDF document data in byte array.
	DocumentFlowId	int	ID of document flow which signing document will be replaced by this merged document.
EnclosureDocument	Name	string	Name of the document. Mandatory. Max 255 chars.
	Data	byte[]	Contains PDF document data in byte array.
StartCampaignSigningResponse	SigningToken	string	Unique token of the signing.
	Transaction Token	Guid	Unique token of the transaction which is used in SigningPortal.
TemplateDataContainer	Items	List<TemplateDataItem>	List of key-value pairs for an XML
TemplateDataItem	Name	string	Name of key-value pair for XML
	Value	string	Value for key-value pair for XML
	Items	List<TemplateDataItem>	Deeper level of key-value pairs for an XML

2.2.8. MergeDocuments

Technical description
This method takes text information which is merged with word documents (replacing merge fields).

Parameters

Name	Type	Description
token	Guid	Unique identifier to identify user.
request	MergeDocumentsRequest	Contains signing token and key-value pairs of information for merging with word documents.

Response
Method gives response (type of MergeDocumentsResponse) with information if the call had any errors and merged documents.

Properties of types

Type	Property	Type	Description
MergeDocumentsRequest	SigningToken	string	Unique token of the signing.
	MergeData	Dictionary<string,string>	Key-value pairs of information which will be used when merging with word documents having merge fields.
MergeDocumentsResponse	MergedDocuments	MergeDocument[]	Array of merged documents. MergeDocument described earlier.

2.2.9. InitiateCampaign

Technical description
This method initiates data campaign.

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.
request	InitiateCampaignRequest	Contains data about how to create signing.
templateOverride	TemplateOverride	Optional. Used in order to replace default account configuration presets (last reminder, signing method, messages etc.).

Properties of types

Type	Property	Type	Description
InitiateCampaignRequest	Name	string	set campaign name (used as signing reference number as well) - up to 36 chars
	Site	string	set web address where data will be collected - up to 100 chars
	RecipientsCsvData	byte[]	extract information about recipients - comma separated values that can be used - recipient name, email, cpr, phone, externalid, send welcome notification, send distribution notification, send documents. If you provide recipients as a list, then this should be null
	Recipients	CampaignRecipient	for adding recipients if not using CSV file
	MergeDocuments	MergeDocument[]	attach word documents for mail merge. If not provided a temporary document will be added which can later be replaced with PDF when using StartCampaignSigning
	SigningTemplateId	Guid	specify which template will be used
	DistributionUrl	string	
	RejectionUrl	string	
	ExpirationUrl	string	
	DocumentSignedUrl	string	Callback URL called when document is being opened. Adds additional query string parameters: "signingToken", "transactionToken", "externalDocumentId".
	DocumentOpenedUrl	string	Url that will be called if provided
	TransactionStateChangedUrl	string	callback url for each transaction state change

	AllowAttachEnclosure	bool	
	AllowAddingComment	bool	
	ExternalReference	string	
	BccRecipients	BccRecipientData[]	Array of bcc recipients
TemplateOverride	same as for InitiateSigning link	same as for InitiateSigning link	same as for InitiateSigning link
MergeDocument	Name	string	name of the document and Data
	Data	byte[]	document data in bytes
CampaignRecipient	Name	string	max 255 chars, Required
	Cpr	string	max 10 chars, required when encryption is using CPR
	Email	string	max 255 chars, required when message type is mail
	Phone	string	max 50 chars, required when message type is SMS
	Cvr	string	max 12 chars, Required when using NemID Employee signing method
	BankIdSsn	string	max 11 chars
	SendWelcomeNotification	bool	if not provided - true
	SendDocuments	bool	If not provided - true. **OBSOLETE, use DistributionMethod instead**
	SendDistributionNotification	bool	if not provided - true. **OBSOLETE, use DistributionMethod instead**
	SigningMethod	SigningMethodEnum	if not provided - used from signing template
	AuthenticationMethod	AuthenticationMethodEnum	if not provided - used from signing template
	Title	string	max 255 chars
	ExternalId	string	max 255 chars
	DistributionMethod	DistributionMethodEnum	Can be null. Defines Recipient distribution method. (see InitiateSigning part for DistributionMethodEnum)
	AllowApproveAllAttachments	bool	Defines if recipient will be able to approve all attachments at once
	IsQuestionnaireEnabled	bool	Defines if recipient will be asked to fill answers for questionnaire
	Secret	string	Secret code, which is mandatory only when using authentication method SecretCode. Max length is 255 characters.
	AdditionalData	string	Optional. Field to store any data per Recipient. Limited to 4000 characters
	TupasSsn	string	Optional. Field is used to provide a Finnish social security number
BccRecipientData	Name	string	Max 255 characters
	Email	string	Max 255 characters

Example

Request	Response
{ "token": "06ef8f71-32c2-4cfe-	{ "CampaignSignings": [

```
b957-93e22c9e10e3",
  "request": {
    "Name": "override2",
    "Site": "http://demo.
vismaaddo.net/datacollection/",
    "SigningTemplateId":
"19e6fe5b-e96a-4ab0-8899-
012a932c1ddc",
    "TransactionStateChangedUrl":
"http://www.google.com",
    "Recipients": [
      {
        "Name": "TestCampaign",
        "Email":
"test_campaign_recipient@visma.
com",
        "SigningMethod": 6
      }
    ]
  },
  "templateOverride": {
    "LastReminder": 1,
    "SigningMethod": 1
  }
}
```

```
    {
      "SigningToken": "757-F53CF-
63B41C",
      "TransactionToken":
"101a925b-8680-41c6-bd2a-
a37cb2826f32"
    },
    {
      "SigningToken": "757-F53D2-
32BE9",
      "TransactionToken":
"7bc10e45-170c-4deb-8424-
9492ebd61914"
    }
  ]
}
```

2.2.10. GetAccountInfo

Technical description
This method returns account related information.

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.

Response
Method gives response (type of GetAccountInfo) which contains account related information.

Properties of types

Type	Property	Type	Description
GetAccountInfo	CreditsLeft	decimal	Number of credits left.
	IsEnterprise	bool	If value is True Account is Enterprise
	Name	string	Account Name
	Email	string	Account Email
	Phone	string	Account Phone
	AccountLevel	AccountLevel	Level of account. Possible values: *Now*(1), *Test*(2), *Enterprise*(3), *Internal* (4)
	AccountId	int	Addo account identifier
	ExternalId	string	External account identifier
	Contacts	List<Contact>	List of saved account contacts (commonly used for autofilling signing recipient info)
Contact	Name	string	Name of a contact
	Title	string	Title of a contact
	Phone	string	Phone of a contact
	Email	string	Email of a contact

	Cvr	string	Tax number of a contact

Example

Request

https://demo.vismaaddo.net/WebService/v2.0/restsigningservice.svc/GetAccountInfo?token=664dfc42-c5fd-4046-bf7d-731bdd04f12c

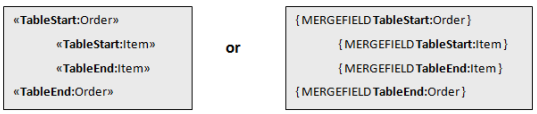
Response


```
{
    "AccountDistributionMethod": 1,
    "AccountId": 1613,
    "AccountLevel": 3,
    "CanValidateCpr": false,
    "Contacts": [
        {
            "Cvr": null,
            "Email": "addo.contact.user@visma.com",
            "Name": "ContactName1",
            "Phone": null,
            "Title": null
        },
        {
            "Cvr": null,
            "Email": "addo.contact.user2@visma.com",
            "Name": "Contact name 2",
            "Phone": null,
            "Title": "Manager"
        }
    ],
    "CreditsLeft": -19.40,
    "Email": "test@visma.com",
    "ExternalId": null,
    "IsEnterprise": true,
    "Name": "Test Name",
    "Phone": "+4500000000"
}
```

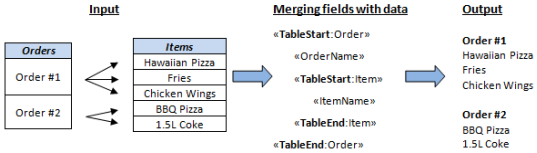

2.2.11. GenerateDocumentWithMergeData

Technical description

This method returns a base64 string of a PDF created from the provided information. The document must contain Merge Fields. Multiple objects (list of objects, dynamically generated tables and etc are also supported) and are called regions. Regions should look like this in the template document:



Example:



To insert a basic merge field in MS Word, click Insert -> Field -> MergeField and enter the same name as it will be used in your object (DataModel).

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.
request	GenerateDocumentWithMergeDataRequest	Contains the data about generating the document
encryptDocument	bool	Boolean if the document needs to be encrypted, default false
encryptionkey	string	Document encryption key if encryption is needed

Types

Type	Description
Guid	A unique set of symbols, necessary for identification.
GenerateDocumentWithMergeDataRequest	Contains properties: Document (type of string) - base64 string for the document template. Most common usage is MS Word document with merge fields. RegionDatas (type of IEnumerable<GenerateDocumentRegionData>) - list of region datas, the objects used on creating dynamic sections in document with merge fields. There can be sources for mergedata: **DataModel** (type of string) - main data object converted to JSON string. Object properties are used in adding data to the end document **XMLMergeData** - XML is used as main data source. If both data source are provided XMLMergeData will be used.
GenerateDocumentRegionData	Contains properties: RegionName (type of string) - name of the region where the objects will be placed. Objects (type of IEnumerable<string>) - list of objects converted to JSON string. Object properties are used in adding data to the end document.

Response
Method gives response (type of string) which contains the document bytes converted to a base64 string.

2.2.11. GetTransactionDetails

Technical description
Gets details of a transaction based on transaction token.

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.
transactionToken	Guid	Transaction token to identify transaction.

Response
Returns an object GetTransactionDetailsResponse with transaction information by the given transaction token.

2.2.12. ReactivateTransaction

Technical description
Reactivates transaction based on transaction token. Only transactions with state failed or expired can be reactivated.

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.
transactionToken	Guid	Transaction token to identify transaction.

2.2.13. CancelTransaction

Technical description
Cancel transaction based on transaction token.

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.
transactionToken	Guid	Transaction token to identify transaction.

Example

Request	Response

{
"token": "c05f50aa-35f4-4fb8-8d3c-b70a9169d28a",
"transactionToken": "abcf50aa-35f4-4fb8-8d3c-b70a9169d28f"
}
	-

2.2.14. UpdateTransaction

Technical description
Updates transaction based on transaction token. If Transaction is found by TransactionToken and updated with the new provided data. If any of the fields (RecipientName, RecipientPhoneNo, RecipientEmail) are null or empty - it will not be used to update transaction.

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.
request	UpdateTransactionRequest	An object with transaction information that needs updating

Properties of types

Type	Property	Type	Description
Guid			A unique set of symbols, necessary for identification.
UpdateTransactionRequest	TransactionToken		
	RecipientName		Name of the recipient, max 255 chars
	RecipientPhoneNo		Recipient phone number, max 50 chars
	RecipientEmail		Recipient email, max 255 chars

2.2.15. SaveRecipient

Technical description

Saves a contact into Address Book. A contact must have unique Email and Phone fields, at least one of them must be provided. If a contact with provided values already exists, it will be updated, otherwise a new one will be inserted.

Parameters

Name	Type	Description

token	Guid	User token, necessary for identification.
contact	Contact	An object with contact information.

Properties of types

Type	Property	Type	Description
Guid			A unique set of symbols, necessary for identification.
Contact	Name		Name of the recipient, max 255 chars.
	Title		Title of the recipient, max 255 chars.
	Email		Recipient email, max 255 chars.
	Phone		Recipient phone number, max 50 chars.
	Cvr		Company identification ID, max 12 chars.

2.2.16. DeleteRecipient

Technical description

Deletes contact from Address Book. Matching is done by Name, Email and Phone fields. If a match is not found - nothing is deleted, in either way returns success response.

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.
contact	Contact	An object with contact information.

Properties of types

Type	Property	Type	Description
Guid			A unique set of symbols, necessary for identification.
Contact	Name		Name of the recipient, max 255 chars.
	Title		Title of the recipient, max 255 chars (not used currently).
	Email		Recipient email, max 255 chars.
	Phone		Recipient phone number, max 50 chars.
	Cvr		Company identification ID, max 12 chars (not used currently).

2.2.17. ExportRecipients

Technical description

Exports all contacts from Address Book in CSV format.

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.
separator	string	Separator to be used for CSV fields (defaults to ';').

Response

This method returns UTF-8 encoded string containing CSV file data. Can be saved directly to file system.

2.2.18. ImportRecipients

Technical description

Import contacts from CSV file into Address Book. A contact must have unique Email and Phone fields, at least one of them must be provided. If there are contacts in Address Book that can be matched by Email or Phone they will be updated, otherwise a new contact will be inserted. For a correct CSV file format we recommend to first do ExportRecipients() call, and then edit the response file if needed.

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.
data	byte[]	Byte array of UTF-8 encoded string of CSV file data.
separator	string	Separator to be used for CSV fields (defaults to ';').

2.2.19. GetCampaigns

Technical description
Get campaigns with same externa reference identifier

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.
externalReference	string	External campaign identifier

Response

Type	Description
GetCampaignsResponse	Contains properties: **Campaigns** - list of **Campaign** object **Campaign** contains: **CampaignStateEnum** - possible values Started = 1, Completed = 2, Canceled = 3 **Signings** - list of **Signing** objects **Signing** contains: **Token -** Signing token, public signing identifier **State -** State of Signing, **SigningStateEnum** **TransactionState** - State of Transaction, **TransactionStateEnum**

Example

Request

```
{
  "token": "db8b5d64-6f45-4748-a93a-
80022d150286",
  "externalReference": null
}
```

Response

```
{
  "Campaigns": [
    {
      "Signings": [
        {
          "State": 4,
          "Token": "4726-
16AAA-6A9A",

"TransactionState": 7
        }
      ],
      "State": 2
    },
    {
      "Signings": [
        {
          "State": 4,
          "Token": "47F4-
14F61-4E20",
```

```
"TransactionState": 7
        }
      ],
      "State": 2
    }
  ]
}
```


2.2.20. GetTemplateMessages

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.
templateId	Guid	Template ID

Response

Type	Description
GetTemplateMessagesResponse	Contains properties: **ID**- Guid of template **TemplateMessages -** list of **SigningTemplateMessage** **SigningTemplateMessage** contains: **Subject**, subject of email message **Body**, content of message **UsageType**, enum, StartMessage = 0, RemindMessage = 1, Distribution = 4, CampaignStartMessage = 5, CampaignRemindMessage = 6 **Type**, enum, Sms = 1, Email = 2, SigningPortalReceipt = 3

Example

Request

https://demo.vismaaddo.net/WebService/v2.0/restsigningservice.svc/GetTemplateMessages?templateId=f2e2e62b-7d67-4695-8961-b2411dbd20a7&token=664dfc42-c5fd-4046-bf7d-731bdd04f12c

Response


```
{
  "Id": "f2e2e62b-7d67-4695-8961-b2411dbd20a7",
  "TemplateMessages": [
    {
      "Body": "<p>Dear %recipientname%</p><p>%sender% has sent you a
document which awaits your signing.You can sign the document by
following the link below: %signingportalurl%.</p><p>Best Regards,<br /> %
sender%<br /> %senderaccount%</p>",
      "Subject": "Document awaits your signing",
      "Type": 2,
      "UsageType": 0
    },
    {
      "Body": "<p>Dear %recipientname%,</p><p>%sender% has sent you a
document which awaits your signing. You can sign the document by
following the link below: %signingportalurl%.</p><p>Best Regards,<br /> %
sender%<br />%senderaccount%</p>",
      "Subject": "Reminder: Document awaits your signing",
```

```
      "Type": 2,
      "UsageType": 1
    },
    {
      "Body": "<p>Dear %recipientname%</p><p>A copy of the signed
documents is attached this e-mail.</p><p>Best Regards,<br /> %sender%<br
/> %senderaccount%</p>",
      "Subject": "Receipt for signed document",
      "Type": 2,
      "UsageType": 4
    },
    {
      "Body": "Dear %recipientname%, <br/><br/> %sender% has added you
to a campaign. You can participate by adding your information here %
campaignsite%. Thank you. <br/><br/> Best regards <br/> %sender%<br/>%
senderaccount%",
      "Subject": "Invitation to participate",
      "Type": 2,
      "UsageType": 5
    },
    {
      "Body": "Dear %recipientname%, <br/><br/> Please go to %
campaignsite% and add your information to generate the document for you
to sign. Thank you. <br/><br/> Best regards <br/> %sender%<br/>%
senderaccount%",
      "Subject": "Reminder",
      "Type": 2,
      "UsageType": 6
    },
    {
      "Body": "Dear %recipientname%\n\n \n\nThis is test version of
email from %sender% on url %signingportalurl%\n\n \n\nBest Regards\n",
      "Subject": null,
      "Type": 3,
      "UsageType": 8
    },
    {
      "Body": "%sender%%signingportalurl%%recipientname% abcde\n\nthis
is test demo template for no receipt text\n",
      "Subject": null,
      "Type": 3,
      "UsageType": 9
    }
  ]
}
```


2.2.21. GetRejectionComment

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.
signingToken	string	Signing Token

Response

Type	Description
GetRejectionCommentResponse	Contains properties: **RejectionComments - RejectionComment[]** - array of rejection commentsRejectionComment contains:**TransactionToken - [Guid]** - transaction Token **Message** - **[string] -** rejection comment/message **RecipientName- [string] -** full name of the user who rejected transaction

	Date - [DateTimeOffset] - rejection date**RecipientEmail - [string] -** email of the user who rejected transaction **RecipientPhone - [string] -** phone number of the user who rejected transaction

Example

Request
https://demo.vismaaddo.net/WebService/v2.0/RestSigningService.svc/GetRejectionComment?token=e487b9a7-92e0-4814-945f-7855cda4bcb4&signingToken=498E-E22E4-F1EDB8

Response

```
{
  "RejectionComments": [
    {
      "Date": {
        "DateTime": "/Date(1535614047058)/",
        "OffsetMinutes": 120
      },
      "Message": "Test rejection comment",
      "RecipientEmail": "nuttyfox.7cfofhn9@mailosaur.io",
      "RecipientName": "Tester",
      "RecipientPhone": null,
      "TransactionToken": "83078f88-70a4-486f-9006-b1ba4d668643"
    }
  ]
}
```

2.2.22. CreateGroup

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.
request	CreateGroupRequest	CreateGroupRequest contains two properties: **Name - [string] -** name of the group **Description - [string] -** group description

Response

Type	Description
GroupResponse	Contains properties: **Id- [int] -** Id of new group **Name - [string] -** name of the group **Description - [string] -** group description

Example

Request

```
{
  "token": "cbf214f0-114e-4ead-a837-10088002869e",
  "request": {
    "Name": "TestGroup2",
    "Description": "this is
```

Response

```
{
  "CreateGroupResult": {
    "Description": "this is postman created group1",
    "Id": 7789,
    "Name": "TestGroup2"
  }
}
```

```
postman created group1"
  }
}
```


2.2.23. GetGroups

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.

Response

Type	Description
GetGroupsResponse	Contains properties: **List<GroupResponse> -** list of GroupResponse model, which contains same properties as CreateGroup method response model.

Example

Request	Response
https://demo.vismaaddo.net/WebService/v2.0/RestSigningService.svc/GetGroups?token=cbf214f0-114e-4ead-a837-10088002869e	`{ "Groups": [ { "Description": "This is for custom templates testing", "Id": 5074, "Name": "Custom templates group" } ] }`

2.2.24. CreateUser

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.
request	CreateUserRequest	CreateUserRequest contains these properties: **Email - [string] -** email of user **FullName - [string] -** full name of user **Phone - [string] -** users phone **LanguageId - [Language {int}] -** enum value of language **RoleId - [UserRole {int}]** - enum value of user role **SendWelcomeNotification - [bool] -** should user get invitation email. If not, response will have password included Enum values: **Language -** Danish = 1, English = 2, Norwegian = 3, Swedish = 4, Finnish = 5, Dutch = 6 **UserRole -** Administrator = 2, Standard = 3, GroupAdministrator = 4

Response

Type	Description
CreateUserResponse	Contains properties: **Id- [int] -** Id of new user **FullName - [string] -** full name of new user **Email - [string] -** email of new user **Password - [string]** - if create user is used without sending welcome notification, Addo generates password and returns it in response, so web service user can send it by himself

Example

Request	Response

<pre>{
  "token": "9ecdee94-3388-4fe4-baab-cf1f3eb0ac1b",
  "request": {
  "Email": "test.tester27@visma.com",
  "FullName": "Test Namer27",
  "LanguageId": 2,
  "Phone": "+450000000",
  "RoleId": 3,
  "SendWelcomeEmail": false
  }
}</pre>
	<pre>{
	  "Email": "test.tester27@visma.com",
	  "FullName": "Test Namer27",
	  "Id": 8912,
	  "Password": "FtJP0NuA"
	}</pre>

2.2.25. AddUserToGroup

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.
request	AddUserToGroupRequest	AddUserToGroupRequest contains two properties: **Email - [string] -** email of user in the account you want to add to group **GroupId- [int] -** group ID in which user has to be added

Example

Request	Response

{
"token": "c05f50aa-35f4-4fb8-8d3c-b70a9169d28a",
"request":{
"Email": "firstname.lastname@visma.com",
"GroupId": 7789
}
}
	-

2.2.26. CancelSigning

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.
transactionToken	Guid	Transaction token to identify transaction.

Example

Request	Response

{
"token": "c05f50aa-35f4-4fb8-8d3c-b70a9169d28a",
"signingToken": "abcf50aa-35f4-4fb8-8d3c-b70a9169d28f"
}
	-

2.2.27. CreateAccount (secured)

Some of our integration parties can create accounts, but to do that, they have to contact Visma Addo Team and get approval.Also, the limitation for created account is enabled.

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.
request	CreateAccountRequest	CreateAccountRequest contains these properties: **Email - [string] -** email of user **FullName - [string] -** full name of user **CompanyName - [string] -** account company name **CVR - [string] -** account CVR number **ZipCode - [string] -** account zip code **Address - [string] -** account address **Phone - [string] -** users phone **Locale - [Language {int}] -** enum value of language **City - [string] -** account city **Country - [string] -** country code of country (3 letters, e.g. "DNK" or "LTU") **ExternalId- [string] -** external identifier that can be added to new account (max length 255 characters) Enum values: **Language -** Danish = 1, English = 2, Norwegian = 3, Swedish = 4, Finnish = 5, Dutch = 6

Response

Type	Description
CreateAccountResponse	Contains properties: **Id- [int] -** Id of new account **FullName - [string] -** full name of new user **Email - [string] -** email of new user **Password - [string]** - if create user is used without sending welcome notification, Addo generates password and returns it in response, so web service user can send it by himself

Example

Request	Response

{
"token": "9ecdee94-3388-4fe4-baab-cf1f3eb0ac1b",
"request": {
"Address": "Test st. 11-4",
"CVR": "12345678",
"City": "Test City",
"CompanyName": "Company name of new account",
"Country": "LTU",
"Email": "new_account@email.com",
"FullName": "New account name",
"Locale": 2,
"Phone": "+555000123456",
"ZipCode": "12345-POST"
}
}
	{
	"Email": "new_account@email.com",
	"FullName": "New account name",
	"Id": 8912,
	"Password": "FtJP0NuA"
	}

2.2.28. GetSignings

Get a list of signings, using small filter and pagination functions

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.
request	GetSigningsRequest	GetSigningsRequestcontains these properties: **Page- [int, required] -** number of page, min value is 1 **PageSize - [int, required] -** number of elements in response, min value is 1, max value is 200 **OrderType - [OrderTypeEnum, required] -** order by created date ascending or descending **CreatedOnFrom - [DateTime, optional] -** optional parameter to set the date signings list should be filtered from **CreatedOnTo - [DateTime, optional] -**optional parameter to set the date signings list should be filtered to Enum values: **OrderTypeEnum -** Ascending = 0, Descending = 1 **Note**:

Response

Type	Property	Type	Description
GetSigningsResponse	Signings	List<SigningItem>	Contains list of **SigningItem** which has some information about signing
SigningItem	Token	string	Signing token which can be used for other actions, like GetSigning or GetSigningStatus
	State	SigningStateEnum	Signing state. For the values you can check GetSigningStatus method
	CreatedOn	DateTime	Date when signing was created
	ExpiresOn	DateTime	Date when signing will expire or has expired
	ReferenceNumber	string	Reference number of the signing which was passed on creation as ability to separate signings

Example

Request

```
{
  "token": "{{token}}",
  "request": {
    "Page": 1,
    "PageSize": 20,
    "CreatedOnFrom": "/Date
(1574850672141)/",
    "CreatedOnTo": "/Date
(1574936930225)/",
    "OrderType": 1
  }
}
```

Response

{
"Signings": [

```
[
  {
    "CreatedOn": "/Date
(1574936535824)/",
    "ExpiresOn": "/Date
(1580210447029)/",
    "ReferenceNumber":
"R12134581",
    "State": 1,
    "Token": "2EA98-12A604-
EC95E4"
  },
  {
    "CreatedOn": "/Date
(1574857155100)/",
    "ExpiresOn": "/Date
(1576066755100)/",
    "ReferenceNumber": null,
    "State": 2,
    "Token": "2EA97-15A7A5-
```

```
C446D1"
  }
]
```

2.2.29. RefreshToken

Extends token for 5 more minutes in order you want to use the same token

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.

Response

Type	Description
DateTime	Returns datetime value of token validity

Example

Request	Response
https://localhost/WebService/v2.0/RestSigningService.svc/RefreshToken?token={{token}}	"/Date (1576508250275)/"

2.2.30. Logout

Suspend session token validity

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.

Response

There is no response object for this method. All you have to check is response Status, it should be 200, which means your token has been revoked and you are no longer able to use it for other requests

Example

Request
https://localhost/WebService/v2.0/restsigningservice.svc/Logout?token={{token}}

2.2.31. GetSecureTransferTemplates

Technical description

This method allows to get secure transfer signing templates of the user. It takes as a parameter user token and returns GetSigningTemplatesResponse with secure transfer signing templates inside. This method can be accessed through both SOAP and REST.

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.

Response

This method returns GetSigningTemplatesResponse. See 2.2.4. GetSigningTemplates for more info about response type.

2.2.32. InitiateSecureTransfer

Technical description

This method initiates a secure file transfer process. Secure file transfer is used to transfer documents safely and efficiently to customers without without necessity to sign a document. Secure file transfer can be used in two ways. The first way is to send an invitation link to recipient and provide documents to download on our own Signing Portal. The other way is to send documents directly to chosen distribution integration (e-Boks or Safecloud at the moment). Only secure transfer templates can be used for secure file transfer (see 2.2.31 GetSecureTransferTemplates).
SigningMethod for all RecipientData in a request must be set to None (0).

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.
request	InitiateSecureTransferRequest	Contains data about how to create secure transfer.

Response

Method gives response (type of InitiateSecureTransferResponse).

Type	Property	Property type	Description
InitiateSecureTransferResponse	SigningToken	string	Token of signing, unique identification.

Properties of the types

Type	Property	Type	Description
InitiateSecureTransferResponse	SigningToken	string	Token of signing, unique identification.
InitiateSecureTransferRequest	Name	string	Defines name of a secure transfer. Mandatory, Max 100 chars.
	StartDate	DateTime	Defines when a signing will start. StartDate value cannot be 24 hours before request is created. Mandatory. Uses UNIX date format like "/Date (1573638173002)/"
	SecureTransferData	SecureTransferData	Contains data about secure transfer. (Similar as Signing in InitiateSigningRequest except this one excludes these properties: BccRecipients, SigningSequence, AllowInboundEnclosures, ExternalReferenceId, EnclosureDocuments. See 2.2.2. InitiateSigning).
	SigningTemplateId	Guid	Defines a Guid which is associated with a template settings of a secure transfer template (authentication, message type etc.).
	ExpirationUrl	string	Expiration callback url. On expiration for the provided url it will add query string parameter "signingToken"
	TransactionStateChangedUrl	string	Transaction state changed callback url. On transaction state change for the provided url it will add query string parameters "signingToken", "transactionToken", "referenceNumber", "transactionStateId"

Send documents directly

To send documents as secure file transfer directly to chosen channel (e.g. e-Boks) you need to set NotificationMethods of RecipientData to None (0). Then web service will use DistributionMethod as a sending method. For example, if NotificationMethods is set to None and DistributionMethod to Eboks (4) secure file transfer will be sent directly to recipient's e-Boks. This way is only available when you have e-Boks or Safecloud activated for your account.Allowed methods for RecipientData when sending directly:

Property	AllowedMethods
AuthenticationMethod	Only None (0) method is allowed.
SigningMethod	This method does not affect anything since there is no signing of documents in signing portal.
DistributionMethod	Only e-Boks method (4) or null are allowed to be set as DistributionMethod. If you want to use Safecloud set DistributionMethod to null and select signing template with Safecloud method. Setting RecipientData Cpr or Cvr decides which of e-Boks methods will be used (private or company). e-Boks or Safecloud must be activated for the account to use them.
NotificationMethods	This **must** set to None (0). Otherwise you will send a link of signing portal to recipient.

Example

Request	Response

{
  "token": "126d6f15-997e-4280-a670-e30fec346392",
  "request": {
  "ExpirationUrl": null,
  "Name": "test signing",
  "SecureTransferData": {
  "AllowRecipientComment": true,
  "Documents": [
  {
  "Data": "base64stringOfFile",
  "Id": "1fe93abb-5363-4eb5-a89f-5210d38471d6",
  "IsShared": false,
  "MimeType": "application/pdf",
  "Name": "test.pdf"
  }
  ],
  "Recipients": [
  {
  "Address": null,
  "AuthenticationMethod": 0,
  "CVR": null,
  "Cpr": "0101987654",
  "Email": "test_recipient1@visma.com",
  "Id": "4c24818a-eae2-4eb0-aa2e-f77ab355f77d",
  "Name": "TestName1",
  "PID": null,
  "Phone": null,
  "SSN": null,
  "SendDistributionDocument": null,
  "SendWelcomeNotification": true,
  "DistributionMethod": 4,
  "NotificationMethods": [0],
  "SigningMethod": 0,
  "Title": null
  }
  ],
  "ReferenceNumber": "666",
  "Sender": {
  "CompanyName": "SenderCompanyName",
  "Email": "sender.email@visma.com",
  "Name": "SenderTestName",
  "Phone": null
  }
  },
  "SigningTemplateId": "95b29391-c23b-46d2-af51-ff9caa8edd9b",
  "StartDate": "/Date(1530791625)/",
  "TransactionStateChangedUrl": null
  }
}
	{
	"SigningToken": "753-1584AE-102882"
	}

Send link

To send link of secure file transfer to signing portal you need to set DistributionMethod of RecipientData to None (1). Then web service will use NotificationMethods from RecipientData or MessageTypes from SigningTemplate to send an invitation.
Allowed methods for RecipientData when sending a link:

Property	AllowedMethods
AuthenticationMethod	All methods described for AuthenticationMethodEnum are allowed except None (0).
SigningMethod	This method does not affect anything since there is no signing of documents in signing portal.
DistributionMethod	DistributionMethod must be set to None(1).
DistributionType	DistributionType must be set to None(0)
NotificationMethods	All methods are allowed except None(0).

It is possible to pass null with all of these methods in both ways of secure file transfer process. In that case a value from signing template will be used. SecureTransferData

Example

Request	Response

{
"token": "126d6f15-997e-4280-a670-e30fec346392",
"request": {
"ExpirationUrl": null,
"Name": "test signing",
"SecureTransferData": {
"AllowRecipientComment": true,
"Documents": [
{
"Data": "base64stringOfFile",
"Id": "1fe93abb-5363-4eb5-a89f-5210d38471d6",
"IsShared": false,
"MimeType": "application/pdf",
"Name": "test.pdf"
}
],
"Recipients": [
{
"Address": null,
"AuthenticationMethod": 1,
"CVR": null,
"Cpr": "0101987654",
"Email": "test_recipient1@visma.com",
"Id": "4c24818a-eae2-4eb0-aa2e-f77ab355f77d",
"Name": "TestName1",
"PID": null,
"Phone": null,
"SSN": null,
"SendDistributionDocument": null,
"SendWelcomeNotification": true,
"DistributionMethod": 1,
"NotificationMethods": [1],
"SigningMethod": 0,
"Title": null
}
],
"ReferenceNumber": "666",
"Sender": {
"CompanyName": "SenderCompanyName",
"Email": "sender.email@visma.com",
"Name": "SenderTestName",
"Phone": null
},
"SenderComment": "this is simple comment"
},
"SigningTemplateId": "95b29391-c23b-46d2-af51-ff9caa8edd9b",
	{
	"SigningToken": "753-1584AE-102882"
	}

```
    "StartDate": "/Date(1530791625)/",
    "TransactionStateChangedUrl": null
  }
}
```


2.2.33. GetTransactionToken

Technical description

The method is used to get transactionToken if you got transactionId from integration like SFTP or Google Drive (transactionId is added to document name)

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.
transactionId	int	Id of transaction which token you want to retrieve

Response

Type	Description
Guid?	TransactionToken is a response. Null is a response with status 200, when that transaction does not belong to this account or is not found at all.

Exampl

Request	Response
https://demo.vismaaddo.net/WebService/v2.0/RestSigningService.svc/GetTransactionToken?token={{token}}&transactionId=123456	"4c24818a-eae2-4eb0-aa2e-f77ab355f77d",

2.2.34. GetSigningWithoutDocuments

Technical description

This method is almost duplicate for GetSigning except it does not generate and return documents, so it retrieves the same data, only Documents and Enclosures arrays are empty.Example and request and response can be found at 2.2.5 GetSigning method

2.2.35. GetAccounts

Get a list of your sub accounts.

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.
filter	GetAccountsFilter	Filter itself is optional, if null, will return all sub accounts available GetAccountsFilter contains these properties: **AccountId - [int, optional] -** account id **AccountExternalId - [string, optional] -** number of elements in response, min value is 1, max value is 200 **UserExternalProviderType- [ExternalProviderEnum, optional] -** user external login type **UserExternalProviderId- [string, optional] -** user external login id, works (together with **UserExternalProviderType** only. Search for a sub account, which has user, containing specific external login identifier)

		CreatedOnFrom - [DateTime, optional] - filter taccounts created after this date **CreatedOnTo - [DateTime, optional] -** filter taccounts created before this date Enum values: **ExternalProviderEnum-** Azure= 1, VismaConnect = 2 **Note**: If you do not provide UserExternalProviderType and UserExternalProviderId they will be null in response even if they exist

Response

Type	Property	Type	Description
GetAccountsResponse	Accounts	List<AccountInfo>	Contains list of **AccountInfo** which has some information about account
AccountInfo	AccountId	int	Addo account id
	CompanyName	string	Company name of an account, might be null
	CreatedOn	DateTime	Date when account was created
	ExternalId	string	Account external identifier, which has been passed on account creation
	UserExternalProviderType	ExternalProviderEnum	External user login from the filter
	UserExternalProviderId	string	External user login identifier from the filter

Example

Request

```
{
  "token": "126d6f15-997e-4280-
a670-e30fec346392",
  "filter": {
    "AccountId": null,
    "UserExternalProviderType":
null,
    "UserExternalProviderId":
null,
    "CreatedFrom": "/Date
(1574850672141)/",
    "CreatedTo": "/Date
(1574936930225)/"
  }
}
```

Response

```
{
    "Accounts": [
        {
            "AccountId": 1710,
            "CompanyName": "",
            "CreatedOn": "/Date
(1551736800000+0200)/",
            "ExternalId": "Test
external Id 123456789,

"UserExternalProviderId": null,

"UserExternalProviderType": null
        },
        {
            "AccountId": 1724,
            "CompanyName": "Company
name of test",
            "CreatedOn": "/Date
(1557003600000+0300)/",
            "ExternalId": null,

"UserExternalProviderId": null,

"UserExternalProviderType": null
        }
    ]
}
```

2.2.36. GetAccountCreditsUsage

Get a list of your sub accounts.

Parameters

Na	Type	Description

me		
token	Guid	User token, necessary for identification.
filter	AccountCreditsUsageFilter	Filter is mandatory AccountCreditsUsageFilter contains these properties: **AccountId - [int, required] -** account id **DateFrom - [DateTime, required] -** filters credits used/reserved since specified date **DateTo - [DateTime, required] -** filters credits used/reserved before specified date

Response

Type	Property	Type	Description
AccountCreditsUsageResponse	AccountId	int	Addo account id
	UsedCredits	decimal	Number of credits used. This number is shown credits spent for already completed, expired, failed, cancelled, rejected.
	DateFrom	DateTime	DateFrom used in filter
	DateTo	DateTime	DateTo used in filter

Example

Request

```
{
  "token": "126d6f15-997e-4280-a670-
e30fec346392",
    "filter": {
        "AccountId": 1816,
        "DateFrom": "/Date
(1551736800000)/",
        "DateTo": "/Date
(1603720913016)/"
    }
}
```

Response

{
"Signings": [

```
{
    "AccountId": 1816,
    "DateFrom": "/Date
(1548972000000)/",
    "DateTo": "/Date
(1603720913016)/",
    "UsedCredits": 151.25
}
```

2.2.37. CalculateTransactionsCount (paired with GetTransactionsCount)

This method will initiate calculation for your account and all your sub accounts transactions per specific period providen in the request. Also it can be filtered for specific transaction state. Each signing type will be separated in the response (see more in 2.2.38 GetTransactionsCount). This implementation of two separate methods is done because you may want to search database for all time entries and it can take longer than timeout period can handle, so we generate this data and store it for specific time (currently it is set to 1 week, might be changed in the furute).

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.
req	CalculateTransactionsCountRequest	CalculateTransactionsCountRequest contains these properties: **DateFrom - [DateTime, required] -** filters Transaction.LastUpdatedStateDate after specified DateFrom **DateTo - [DateTime, required] -** filters Transaction.LastUpdatedStateDate before specified DateTo

u e st		**CallbackUrl - [string, optional] -** callback url which to call after transactions counts for all accounts are calculated. Your providen callback url will be attached with additional parameter "getTransactionsCountToken" which can be later used in GetTransactionsCount endpoint, to retrieve data **TransactionState - [int, optional] -** you can filter specific states of transactions, for example completed only or expired only. Null will not add filter and count all transactions, no matter what transaction state is

Response

Type	Property	Type	Description
CalculateTransactionsCountResponse	GetTransactionsCountToken	Guid	A special token which should be used in GetTransactionsCount endpoint to retrieve generated data
	TokenValidTo	DateTime	Date value until when GetTransactionsCountToken can be used

Example

Request

```
{
  "token": "126d6f15-997e-4280-a670-e30fec346392",
    "request": {
        "CallbackUrl": "https://example.com",
        "TransactionState": 6,
        "DateFrom": "/Date(1551736800000)/",
        "DateTo": "/Date(1603720913016)/"
    }
}
```

Response

{
"Signings": [

```
{
    "GetTransactionsCountToken": "49bb38cd-f24d-4367-af37-b88c5e68dc97",
    "TokenValidTo": "/Date(1614765778068)/"
}
```

2.2.38. GetTransactionsCount

This method is used to retrieve data, calculated by CalculateTransactionsCount endpoint with that approriate response token.

Parameters

Name	Type	Description
token	Guid	Unique identifier to identify user.
getTransactionsCountToken	Guid	Token for retrieving stored data of transaction counts for accounts and sub accounts.

Properties of types

Type	Property	Type	Description
TransactionsCountResponse	GeneratedOn	DateTime?	Date when data was generated. If it is null, it is not generated yet
	ExpiresOn	DateTime	Date when this entry expires. If you call this endpoint after this expiration date is expired, you will not get this response object
	Filter	CalculateTransactionsCountRequest	Filter of the previously called request to see what filter values has been used for this data. (see CalculateTransactionsCount endpoint for more info about this field)
	AccountsTransactionsCounts	AccountTransactionsResponse[]	List of transactions count for account for specific signing type

AccountTransactionsResponse	AccountId	int	Account identifier
	AccountExternalid	string	Account external identifier (the same which might be added on creation of sub account)
	SigningType	SigningTypeEnum	Signing type of Addo workflow
	TransactionsCount	int	Number of transactions

Enum values

Type	Value	Description
SigningTypeEnum	*Signing(1)*	Regular document signing type
	SecureTransfer(2)	Secure file transfer signing type
	Form(3)	Addo forms signing type
	CprValidation(4)	Cpr validation signing type

Example

Request

https://localhost/WebService/v2.0/RestSigningService.svc/GetTransactionsCount?token={{token}}&getTransactionsCountToken=49bb38cd-f24d-4367-af37-b88c5e68dc97

Response


```
{
    "AccountsTransactionsCounts": [
        {
            "AccountExternalId": null,
            "AccountId": 1613,
            "SigningType": 1,
            "TransactionsCount": 218
        },
        {
            "AccountExternalId": null,
            "AccountId": 1613,
            "SigningType": 2,
            "TransactionsCount": 1
        },
        {
            "AccountExternalId":
"ID_EXT_123456_RandomNumbersCompany_001",
            "AccountId": 1816,
            "SigningType": 1,
            "TransactionsCount": 1
        }
    ],
    "ExpiresOn": "/Date(1614765778068)/",
    "Filter": {
        "CallbackUrl": "https://www.google.com/",
        "DateFrom": "/Date(1551002577498)/",
        "DateTo": "/Date(1614160977497)/",
        "TransactionState": 6
    },
    "GeneratedOn": "/Date(1614161028645)/"
}
```

3. Security

The Addo service is exposed as a WS-* specifications compatible service.The WS-* specifications enable the service to support reliable messaging and security by default.Clients using .net version older than 3.0 will not be able to consume the service.Soap version: 1.2The Addo service is exposed on two different security configurations.

3.1. Client certificate and Server certificate

This is secured as Transport with message credentials.This means the service is exposed over https (ssl).All messages are secured using client credential type certificate, which means the client provides a certificate as proof for its identity.This enabled the Addo service to determine the validity of the identity of the client.The service identifies itself by a certificate which is also used in the SSL communication taking place.This enables the clients to determine the validity of the identity of the Addo service.In Addo a client must be registered as using this type of connection before it will succeed.

3.2. Client account email and password with server certificate

This is secured as Transport with message credentials.This means the service is exposed over https (ssl).All messages are secured using client credential type username which means a username and password, which is a client's account email and hashed password (SH A512) which has no correlation to Addo credentials.The email and hashed password are handed to clients which then must configure the service consumption on their end using these client credentials.This enabled the Addo service to determine the validity of the identity of the client.The service identifies itself by a certificate which is also used in the SSL communication taking place.This enables the clients to determine the validity of the identity of the Addo service.In Addo a client must be registered as using this type of connection before it will succeed.

4. Notes

4.1 Dates

Please note that dates, unless otherwise noted, should be in the format:

```
/Date(xxx)/
```

(also known as the microsoft format).

The **xxx** part should be specified in milli-seconds. PHP users - the **time()** function returns the time in seconds so you have to multiply this value by 1000.

4.2 Durations

The format of the Duration in the TemplateOverrides to the /InitiateSigning call must have this format:

```
[-]P[{days}D][T[{hours}H][{min}M][{sec}S]]
```

... where:

- - Indicates negative timespan, omitted for positive values
- P must be the first character (unless negative time value)
- T must precede the time portion of the timespan.
- [] = optional part that may be omitted if 0.

Example. If you like to specify a duration of two days then the string would be:

```
P2D
```

.. if you would like to specify 3 days, 7 hours and 2 seconds, then:

```
P3DT7H2S
```

Please note that unless otherwise noted durations in the Addo API is always expressed in whole positive days (PxD, where x is the number of days).

4.3 MIME-types

Currently the MIME-type of the documents should be set to "PDF".

4.4 Creating invitation link by hand

You shouldn't be doing this. If you cannot wait for us to get it implemented then this is how it is done:

1. Obtain a valid `SESSIONTOKEN` and a valid `SIGNINGTOKEN` (by calling /Login and /InitiateSigning).

2. Call /GetSigningStatus:

```
GET https://vismaaddo.net/WebService/v2.0/restsigningservice.svc
/GetSigningStatus?signingToken=SIGNINGTOKEN&token=SESSIONTOKEN
```

3. Convert the result JSON to an object and then you f.ex. iterate the recipients:

```
foreach (var recipient in result.Recipients) {
    var signingUrl = "https://vismaaddo.net/SigningPortal/?token=" +
recipient.Transactions[0].TransactionToken;
    // Do something with signingUrl - include it in an email, redirect in a
browser window ... whatever is good for you!
}
```

Big fat warning! It is unlikely that we will change this. But things will change from time to time ....