

API Documentation - Addo Web Service v2.3

- 1. Introduction
 - 1.1. Purpose
 - 1.2. Allowed HTTPs requests
 - 1.3. Description of server responses
 - 1.4. WSDL document
 - 1.5. List of API methods
- 2. Usage of Addo API v2.3
 - 2.1. REST via Postman
 - 2.1.1. POST request via Postman
 - 2.1.2. GET request via Postman
 - 2.2. Public methods
 - 2.2.1. Login
 - Technical description
 - Parameters
 - Example
 - 2.2.2. InitiateSigning
 - Technical description
 - Parameters
 - Response
 - Types
 - Properties of the types
 - Example
 - 2.2.3. GenerateDocument
 - Technical description
 - Parameters
 - Properties of the types
 - 2.2.4. GetSigningTemplates
 - Technical description
 - Properties of types
 - 2.2.5. GetSigning
 - Technical description
 - Parameters
 - Properties of types
 - Example
 - 2.2.6. GetSigningStatus
 - Technical description
 - Parameters
 - Properties of types
 - Enum values
 - Example
 - 2.2.7. StartCampaignSigning
 - Technical description
 - Parameters
 - Response
 - Properties of types
 - 2.2.8. MergeDocuments
 - Technical description
 - Parameters
 - Response
 - Properties of types
 - 2.2.9. InitiateCampaign
 - Technical description
 - Parameters
 - Properties of types
 - Example
 - 2.2.10. GetAccountInfo
 - Technical description
 - Parameters
 - Response
 - Properties of types
 - 2.2.11. GenerateDocumentWithMergeData
 - Technical description
 - Parameters
 - Types
 - Response
 - 2.2.11. GetTransactionDetails
 - Technical description
 - Parameters

- Response
- 2.2.12. ReactivateTransaction
 - Technical description
 - Parameters
- 2.2.13. CancelTransaction
 - Technical description
 - Parameters
 - Example
- 2.2.14. UpdateTransaction
 - Technical description
 - Parameters
 - Properties of types
- 2.2.15. GetCampaigns
 - Technical description
 - Parameters
 - Response
 - Example
- 2.2.16. GetTemplateMessages
 - Parameters
 - Response
 - Example
- 2.2.17. GetRejectionComment
 - Parameters
 - Response
 - Example
- 2.2.18. CreateGroup
 - Parameters
 - Response
 - Example
- 2.2.19. GetGroups
 - Parameters
 - Response
 - Example
- 2.2.20. AddUserToGroup
 - Parameters
 - Example
- 2.2.21. CreateUser
 - Parameters
 - Response
 - Example
- 3. Security
 - 3.1. Client certificate and Server certificate
 - 3.2. Client account email and password with server certificate

1. Introduction

1.1. Purpose

This document describes Addo Web Service 2.3 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>

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]

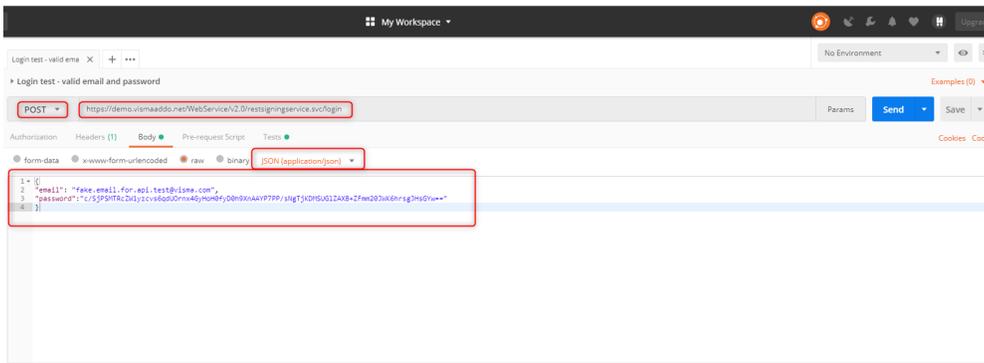
GET - [Base address]/[Public method name]?[Parameters]

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/SjP5MTRcZWlyzcvs6qdUOrnx4GyHoH0fyD0h9XnAAYP7PP/sNgTjKdMSUG1.ZAXB+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

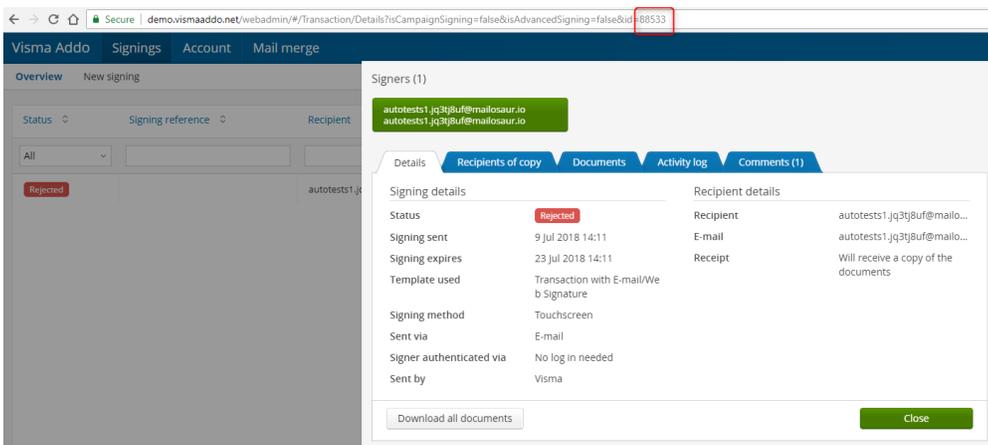
Here is an example of a method with **GET request**, showing how parameters are passed through URL.

In Postman, we are using **GetRejectionComment** method (list of methods available here List of API methods). This requires us to pass **transactionId** 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":

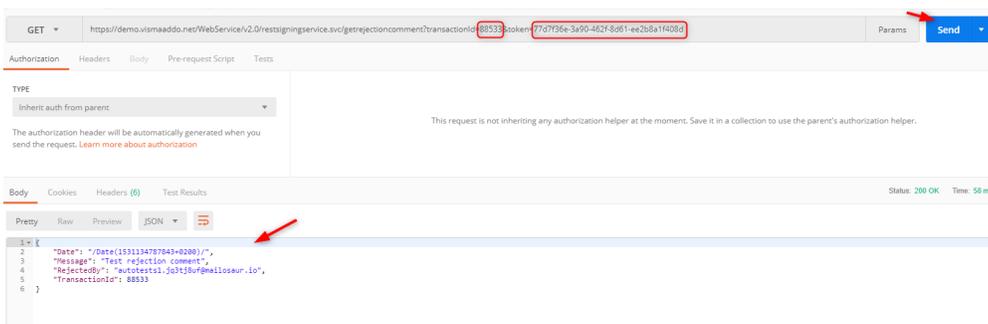


Transaction id can be seen in you web browser, in the link after opening rejected transaction in overview tab



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

<https://demo.vismaaddo.net/WebService/v2.0/restsigningservice.svc/getrejectioncomment?transactionId=88533&token=77d7f36e-3a90-462f-8d61-ee2b8a1f408d>



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.

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)*
- *void ReactivateTransaction(Guid token, Guid transactionToken)*
- *void UpdateTransaction(Guid token, UpdateTransactionRequest updateTransactionRequest)*
- *void SaveRecipient(Guid token, Contact contact)*
- *GetTemplateMessagesResponse GetTemplateMessages(Guid token, Guid templateId);*
- *GetCampaignsResponse GetCampaigns(Guid token, string externalReference);*
- *GetRejectionCommentResponse GetRejectionComment(Guid token, int transactionId);*
- *GroupResponse CreateGroup(Guid token, CreateGroupRequest request);*
- *GetGroupsResponse GetGroups(Guid token);*
- *void AddUserToGroup(Guid token, AddUserToGroupRequest request);*
- *CreateUserResponse CreateUser(Guid token, CreateUserRequest 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
<pre>{ "email": "fake.email.for.api.test@visma.com", "password": "c/SjPSMTRcZW1yzcv6qdUOrnx4GyHoH0fyD0h9XnAAYP7PP/sNgTjKDMSUGiZAXB+ZFmm20JWK6hrsgJHsGYw==" }</pre>	77d7f36e-

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), MessageType (type 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 recipients' 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 of 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, 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).
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: <i>Sms</i> (1), <i>Email</i> (2).
TemplateUsageType	Describes for what purpose the template message will be used. Can have values: <i>StartMessage</i> (0), <i>RemindMessage</i> (1), <i>Distribution</i> (4), <i>CampaignStartMessage</i> (5), <i>CampaignRemindMessage</i> (6).
MessageTypeEnum	Can have values: <i>Mail</i> (1), <i>SMS</i> (2). Describes on which channel the campaign will send out messages to a recipient.
SigningMethodEnum	Can have values: <i>NemId</i> (1), <i>Stylus</i> (2), <i>BankID</i> (3), <i>NemIDMoces</i> (4), <i>SwedishBankID</i> (5), <i>Accept</i> (6), <i>NoBankIDMobile</i> (7), <i>NemIdPocesNoSsn</i> (deprecated, does the same as NemId), <i>FinnishTupas</i> (9). Describes how the documents of the signing must be signed.

NotificationIntervalEnum	Can have values: <i>DontSend</i> (1), <i>EveryWeek</i> (2), <i>EveryTwoWeeks</i> (3), <i>EveryThreeWeeks</i> (4). Describes the frequency with which the campaign will send out notifications to its recipients.
LastReminderEnum	Can have values: <i>DontSend</i> (1), <i>DayBefore</i> (2), <i>TwoDaysBefore</i> (3). Describes when to send out last reminder to recipients of the signing.
DocumentEncryptionEnum	Can have values: <i>None</i> (1), <i>Encrypted</i> (2), <i>Passcode</i> (3). Specifies whether encryption is enabled for signing documents when distributing.
AuthenticationMethod	Can have value: <i>None</i> (0), <i>NemIdPrivate</i> (1), <i>NemIdPrivateNoSsn</i> (deprecated, does the same as <i>NemIdPrivate</i>), <i>TwoFactorVerification</i> (3). Specifies which method should use on authentication before signing documents.

Properties of the types

Type	Property	Type	Description
InitiateSigningResponse	SigningToken	string	Token of signing, unique identification.
InitiateSigningRequest	Name	string	Defines name of a signing. Mandatory, Max 100 chars.
	StartDate	DateTime	Defines when a signing will start. StartDate value cannot be 24 hours before request is created. Mandatory.
	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	string	Distribution callback url. On distribution for the provided url it will add query string parameter "signingToken"
	RejectionUrl	string	Rejection callback url. On rejection for the provided url it will add query string parameter "signingToken"
	ExpirationUrl	string	Expiration callback url. On expiration for the provided url it will add query string parameter "signingToken"
	DocumentSignedUrl	string	Document signed callback url. On document signed for the provided url it will add query string parameters "signingToken", "transactionToken", "externalDocumentId"
	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"
TemplateOverride	Duration	TimeSpan?	Describes life time of the signing, must be longer than a day.
	MessageType	MessageTypeEnum?	Describes on which channel the signing will send out messages to a recipient: <i>Mail</i> (1) or <i>SMS</i> (2).
	NotificationInterval	NotificationIntervalEnum?	Describes the frequency with which the signing will send out notifications to its recipient: <i>DontSend</i> (1), <i>EveryWeek</i> (2), <i>EveryTwoWeeks</i> (3), <i>EveryThreeWeeks</i> (4).
	LastReminder	LastReminderEnum?	Describes when to send out last reminder to recipients of the signing: <i>DontSend</i> (1), <i>DayBefore</i> (2), <i>TwoDaysBefore</i> (3).
	SigningMethod	SigningMethodEnum?	Describes how the documents of the signing must be signed: <i>NemId</i> (1), <i>Stylus</i> (2), <i>BankID</i> (3), <i>NemIdMoces</i> (4), <i>SwedishBankID</i> (5), <i>Accept</i> (6), <i>NoBankIDMobile</i> (7), <i>NemIdPocesNoSsn</i> (deprecated, does the same as <i>NemId</i>), <i>FinnishTupas</i> (9).
	DocumentEncryption	DocumentEncryptionEnum?	Specifies whether encryption is enabled for signing documents when distributing: <i>None</i> (1), <i>Encrypted</i> (2), <i>Passcode</i> (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).
Signing	Recipients	RecipientData[]	Contains recipients which are associated with the signing.
	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.
	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.
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 file "application/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

Example

Request	Response
<pre>{ "token": "126d6f15-997e-4280-a670-e30fec346392", "request": { "DistributionUrl": null, "DocumentSignedUrl": null, "ExpirationUrl": null, "Name": "test signing", "RejectionUrl": null, "SigningData": { "AllowInboundEnclosures": true, "AllowRecipientComment": true, "BccRecipients": [], "Documents": [{ "Data": "base64stringOfFile", "Id": "1fe93abb-5363-4eb5-a89f-5210d38471d6", "IsShared": false, "MimeType": "application/pdf", "Name": "test.pdf" }], "EnclosureDocuments": null, "ExternalReferenceId": null, "Recipients": [{ "Address": null, "AuthenticationMethod": null, "CVR": null, "Cpr": "0101987654",</pre>	<pre>{ "SigningToken": "753-1584AE-102882" }</pre>

```
"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)/",
```

```
"TransactionStateChangedUrl": null
}
}
```

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.

MessageType	MessageTypeEnum	See above in the <i>InitiateSigning</i> description.
LastReminder	LastReminderEnum	
SigningMethod	SigningMethodEnum	
DocumentEncryption	DocumentEncryptionEnum	
AuthenticationMethod	AuthenticationMethod	
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.

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.

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 documents.
	ReferenceNumber	string	Contains optional reference number.
	Documents	SigningDocument[]	Contains signing documents.
Recipient	Name	string	Contains Full Name of recipient
	XmlData	string	Contains XML data of the signing document, which have been submitted in TemplateData.
Document	Name	string	Name of the document.
	Data	byte[]	Data of the document in bytes.
	MimeType	string	Mime type of the document.
Enclosure (inherits from document)	Name	string	Name of the document.
	Data	byte[]	Data of the document in bytes.

	MimeType	string	Mime type of the document.
	IsInbound	bool	If true Encloure is was uploaded by recipient, if false Enclouse created by Sender
SigningDocument (inherits from document)	Name	string	Name of the document.
	Data	byte[]	Data of the document in bytes.
	MimeType	string	Mime type of the document.
	XmlDSig	string	Contains XML digital signature data.
	FlowId	int	Corresponding document flow id
BccRecipient	Name	string	Contains Full Name of bcc recipient
	Email	string	Email of bcc recipient

Example

Request

<https://demo.vismaaddo.net/WebService/v2.0/restsigningservice.svc/GetSigning?signingToken=48C5-127B8C-F34D82&token=664dfc42-c5fd-4C>

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.
Recipient	Name	string	Full Name of Recipient
	Transactions	Transactions[]	Contains Transactions
Transactions	TransactionToken	Guid	Token of the transaction.
	State	TransactionStateEnum	Describes transaction state

Enum values

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

<i>DataCollecting (9)</i>	Relates to data campaigns. After campaign is initiated but not yet started, waiting for merged documents.
<i>Rejected (10)</i>	Recipient rejected signing the document.

Example

Request

<https://demo.vismaaddo.net/WebService/v2.0/restsigningservice.svc/GetSigningStatus?signingToken=48C5-127B8C-F34D82&token=664dfc42-c>

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.
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.
	TransactionToken	Guid	Unique token of the transaction which is used in SigningPortal.

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	
	TransactionStateChangedUrl	string	callback url for each transaction state change
	AllowAttachEnclosure	bool	
	AllowAddingComment	bool	
ExternalReference	string		
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	string	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
	SendDistributionNotification	bool	if not provided - true
	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

Example

Request	Response
<pre>{ "token": "06ef8f71-32c2-4cfe-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 } }</pre>	<pre>{ "CampaignSignings": [{ "SigningToken": "757-F53CF-63B41C", "TransactionToken": "101a925b-8680-41c6-bd2a-a37cb2826f32" }, { "SigningToken": "757-F53D2-32BE9", "TransactionToken": "7bc10e45-170c-4deb-8424-9492ebd61914" }] }</pre>

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: <i>Now(1)</i> , <i>Test(2)</i> , <i>Enterprise(3)</i> , <i>Internal(4)</i>
	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

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	<p>Contains properties: Document (type of string) - base64 string for the document template. Most common usage is MS Word document with merge fields. RegionDats (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:</p> <p>DataModel (type of string) - main data object converted to JSON string. Object properties are used in adding data to the end document</p> <p>XMLMergeData - XML is used as main data source.</p> <p>If both data source are provided XMLMergeData will be used.</p>
GenerateDocumentRegionData	<p>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</p>

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

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. 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	<p>Contains properties:</p> <p>Campaigns - list of Campaign object</p> <p>Campaign contains: CampaignStateEnum - possible values Started = 1, Completed = 2, Canceled = 3</p> <p>Signings - list of Signing objects</p> <p>Signing contains:</p> <p>Token - Signing token, public signing identifier</p> <p>State - State of Signing, SigningStateEnum</p> <p>TransactionState - State of Transaction, TransactionStateEnum</p>

Example

Request	Response

```

{
  "token": "db8b5d64-6f45-4748-a93a-80022d150286",
  "externalReference": null
}
{
  "Campaigns": [
    {
      "Signings": [
        {
          "State": 4,
          "Token": "4726-16AAA-6A9A",
          "TransactionState": 7
        }
      ],
      "State": 2
    },
    {
      "Signings": [
        {
          "State": 4,
          "Token": "47F4-14F61-4E20",
          "TransactionState": 7
        }
      ],
      "State": 2
    }
  ]
}

```

2.2.16. GetTemplateMessages

Parameters

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

Response

Type	Description
GetTemplateMessagesResponse	<p>Contains properties:</p> <ul style="list-style-type: none"> ID- Guid of template TemplateMessages - list of SigningTemplateMessage <p>SigningTemplateMessage contains:</p> <ul style="list-style-type: none"> 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-b2411dbd20>

2.2.17. GetRejectionComment

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.
transactionId	int	Transaction ID

Response

Type	Description
GetRejectionCommentResponse	Contains properties: Message - [string] - rejection comment/message TransactionId - [int] - transaction ID RejectedBy - [string] - full name of the user who rejected transaction Date - [DateTime] - rejection date

Example

Request

<https://demo.vismaaddo.net/WebService/v2.0/restsigningservice.svc/getrejectioncomment?transactionId=49480&token=b42ee72d-29d3-41b4-b7>

2.2.18. 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	Response
<pre>{ "token": "cbf214f0-114e-4ead-a837-10088002869e", "request": { "Name": "TestGroup2", "Description": "this is postman created group1" } }</pre>	<pre>{ "CreateGroupResult": { "Description": "this is postman created group1", "Id": 7789, "Name": "TestGroup2" } }</pre>

2.2.19. 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	<pre>{ "Groups": [{ "Descriptio "This is for custom templates testing", "Id": 5074, "Name": "Custom templates group" }] }</pre>

2.2.20. 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
<pre>{ "token": "c05f50aa-35f4-4fb8-8d3c-b70a9169d28a", "request": { "Email": "firstname.lastname@visma.com", "GroupId": 7789 } }</pre>	-

2.2.21. CreateUser

Parameters

Name	Type	Description
token	Guid	User token, necessary for identification.
model	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", "model": { "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>

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: 1.2

The 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.