

TSOS To ESOS migration

API Change Notification: Transition from XML-DSIG to JSON Format

Introduction: This document outlines the changes introduced in our application's API, specifically the transition from XML-DSIG message format to JSON. These changes aim to enhance usability, improve performance, and provide a more modern and flexible interface for users.

Key Changes:

1. Message Format:

- **XML-DSIG (Old Format):**
 - Previously, our API accepted and returned data in XML-DSIG format, which encapsulates XML data along with digital signatures for verification purposes.
- **JSON (New Format):**
 - The updated API now utilizes JSON format for both request and response payloads. JSON offers a more lightweight and human-readable structure, facilitating easier data parsing and manipulation.

2. Data Structure:

- **XML-DSIG (Old Format):**
 - XML-DSIG required a specific hierarchical structure for data representation, often leading to complex and verbose payloads.
- **JSON (New Format):**
 - With JSON, data is represented in a simpler key-value pair format, promoting clarity and conciseness in the transmitted information.

3. Authentication and Security:

- **XML-DSIG (Old Format):**
 - Authentication and security were primarily handled through XML digital signatures, ensuring the integrity and authenticity of the transmitted data.
- **JSON (New Format):**
 - The transition to JSON does not compromise on security. Authentication mechanisms remain intact, although the format for conveying security-related information differs.

Migration Guide:

To adapt to the changes in the API, follow these steps:

1. Update Request and Response Handlers:

- Modify your application's request and response handlers to accommodate JSON payloads instead of XML-DSIG.

2. Adjust Data Parsing Logic:

- Revise any data parsing logic to correctly interpret JSON-formatted data.

3. Review Authentication Mechanisms:

- Ensure that authentication mechanisms align with the updated API specifications, considering any changes in security token formats or authentication endpoints.

4. Testing:

- Thoroughly test your application with the new API format to validate compatibility and functionality.

Mapping between the XML and JSON formats:

Use case	XML interface	JSON interface	Notes and comments
Authentication	All XML messages to E-Signing must be signed with a Nets issued certificate.	Customer needs to be configured as ESOS user and Nets will provide the credentials. Need to fetch the access_token and used this access_token as "Authorization: Bearer <access_token>" for all subsequent calls. <code>curl -L -X POST "https://www.sign-preprod1.nets.eu/esos/token" \</code> <code>-H "Content-Type: application/json" \</code> <code>-H "Authorization: Basic <Base64 encoded username:password>"</code> Response <pre>{ "access_token": "eyJhb... m1R5fAqw", "refresh_token": "c26c5c1f-2cac- 4888-9e2b-403e3168cb87", "token_type": "Bearer", "expires_in": 18000 }</pre>	

<p>Create sign orders</p>	<pre><?xml version="1.0" encoding="UTF-8"?> <TrustSignMessage xmlns="http://www.bbs.no/tt/trustsign/2023/03/tsm#" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <MerchantID>1001</MerchantID> <Time>2024-04-25T07:35:54+00:00</Time> <MessageID>1234</MessageID> <AdditionalInfo>testing additional info</AdditionalInfo> <InsertOrder> <OrderID>1224075384762</OrderID> </OrderID> <Documents>....</Documents> <Merchant> <Notification> <NotificationChannels> <NotificationChannel> <Channel> <Email> <EmailAddress>mac.sistest@gmail.com</EmailAddress> </Email> </Channel> <Triggers> <Trigger>OnOrderCompletion</Trigger> <Trigger>OnStepCompletion</Trigger> </Triggers> </NotificationChannel> </NotificationChannels> </Notification> </Merchant> <Signers>.....</Signers> <ExecutionDetails> <OrderDeadline>2024-04-26T07:35:54+00:00</OrderDeadline> <DisplayProcessInfo>NameStatusTime</DisplayProcessInfo> <GenerateOneTimeURLs>>false</GenerateOneTimeURLs> <Steps> <Step> <StepNumber>1</StepNumber> </Step> </Steps> <SigningProcess> <LocalDocumentReference>doc1</LocalDocumentReference> <LocalSignerReference>user1</LocalSignerReference> <TerminateOrderOnSignerRejection>true</TerminateOrderOnSignerRejection> </SigningProcess> <OutputFormat>SDO</OutputFormat> </ExecutionDetails> <MetaData> <NameValuePair> <Name>metaname-1</Name> <Value>metavalue-1</Value> </NameValuePair> <NameValuePair> <Name>metaname-2</Name> <Value>metavalue-2</Value> </NameValuePair> </MetaData> </InsertOrder> </TrustSignMessage></pre>	<pre>{ "requestId": "a2df74ff-8f4c-466b-8e99-72c8385b18b5", "objectId": "6456b020-e5f3-430b-909a-9a0521509946", "orderInfoDisplay": "NAMESTATUSTIME", "orderDescription": "My First Loan Order", "useOneTimeUrls": false, "orderDeadline": "2024-06-09T11:16:59.652Z", "outputFormat": "SDO", "orderMetadata": [{ "name": "metaname-1", "value": "metavalue-1" }, { "name": "metaname-2", "value": "metavalue-2" }], "notifications": [{ "objectId": "b1065735-5a82-468b-8e98-faa99e28a56f1", "method": "XML", "endpoint": "https://notifications.customer.com/signorder/notification", "triggers": ["ORDERCOMPLETE", "ORDERFAIL"] }] }</pre>	<p>InsertOrder / OrderID -> objectId InsertOrder / OrderDescription -> orderDescription ExecutionDetails/DisplayProcessInfo -> orderInfoDisplay ExecutionDetails/GenerateOneTimeURLs -> useOneTimeUrls ExecutionDetails/OrderDeadline -> orderDeadline ExecutionDetails/OutputFormat -> outputFormat InsertOrder / MetaData -> orderMetadata InsertOrder / Merchant / Notification -> notifications // triggers values are described in schema. eg. OnOrderCompletion -> ORDERCOMPLETE, OnStepCompletion -> STEPCOMPLETE</p>
---------------------------	---	---	---

TSOS To ESOS migration

<p>Adding documents</p>	<pre><Documents> <Document> <LocalDocumentReference>document1</LocalDocumentReference> <Presentation> <Title>Document Title</Title> <Description>PDF Document to review and Sign</Description> </Presentation> <DocType> <PDF> <B64DocumentBytes>JVBERi0xLjQNJeLjz9MNCjzIDA....</B64DocumentBytes> </PDF> </DocType> <RequiresAuthentication>false</RequiresAuthentication> </Document> </Documents></pre>	<pre>{ "requestId": "DocReqId001", "documents": [{ "objectId": "document1", "documentTitle": "Document Title", "documentDescription": "ESOS PDF Document to review and Sign", "requireAuthentication": false, "content": { "contentType": "APPLICATION_PDF", "b64data": "JVBERi0xLjQNJeLjz9M...." } }] }</pre>	<pre>DocType/ PDF "contentType": "APPLICATION_PDF" DocType/ TEXT "contentType": "TEXT_PLAIN" DocType/ XML "contentType": "APPLICATION_XML" RequiresAuthentication -> requireAuthentication</pre>
-------------------------	--	--	--

<p>Adding signers</p>	<p>Eg for NativeSigning :</p> <pre><Signers> <Signer> <EndUserSigner> <LocalSignerReference>user1</LocalSignerReference> <Name>Signer name</Name> <AcceptedPKIs> <BankID> <CertificatePolicy>PersonalQualified</CertificatePolicy> <SignerID> <IDType>SSN</IDType> <IDValue>28091195137</IDValue> </SignerID> </BankID> </AcceptedPKIs> </EndUserSigner> </Signer> </Signers></pre> <p>Eg for ABS :</p> <pre><Signers> <Signer> <EndUserSigner> <LocalSignerReference>user1</LocalSignerReference> <Name>Signer name</Name> <AcceptedPKIs> <Nets> <Authentication> <AuthenticationID>mitid</AuthenticationID> <SignerID> <IDType>SSN</IDType> <IDValue>0308451234</IDValue> </SignerID> </Authentication> </Nets> </AcceptedPKIs> </EndUserSigner> </Signer> </Signers></pre>	<p>Eg for NativeSigning :</p> <pre>{ "requestId": "369cf0e7-b175-4345-a789-b666909870c4", "signers": [{ "objectId": "566a79e2-7bf1-44cd-9233-dd8e48c82ef4", "signerName": "Signer name", "signerEids": { "signerIdScope": "BASIC", "signMethods": [{ "name": "NO_BANKID", "certPolicy": "PERSONALQUALIFIED", "ids": [{ "name": "SSN", "value": "28091195137" }] }] } }] }</pre> <p>Eg for ABS :</p> <pre>{ "requestId": "3c0cf0e7-b175-4345-a789-b666909870c4", "signers": [{ "objectId": "566a79e2-7bf1-44cd-9233-dd8e48c82ef4", "signerName": "Signer name", "signerEids": { "signerIdScope": "BASIC", "authMethods": [{ "name": "NO_BANKID", "ids": [{ "name": "SSN", "value": "20028713506" }] }, { "name": "SE_BANKID" }, { "name": "FI_TUPAS" }] } }] }</pre>	<p>LocalSignerReference objectId</p> <p>Name signerName</p> <p>AcceptedPKIs / BankID signMethods with "name": "NO_BANKID"</p> <p>AcceptedPKIs / BankIDSE signMethods with "name": "SE_BANKID"</p> <p>In ESOS native signing = "signMethods" and ABS signing = "authMethods"</p> <p>In TSOS (XML base solution), it is <Nets> tag for ABS In ESOS (REST API), it is "authMethods"</p> <p>AcceptedPKIs / Nets /Authentication (AuthenticationID="no_bankid") authMethods with "name": "NO_BANKID"</p> <p>AcceptedPKIs / Nets /Authentication (AuthenticationID="mitid") authMethods with "name": "DK_MITID"</p> <p>In TSOS for IncludeSSN <IncludeSSN>true</IncludeSSN></p> <p>In ESOS for IncludeSSN "signerIdScope": "SSN",</p> <p>for more detail about different IDP wise example, please check open api (.yaml) document.</p>
-----------------------	---	---	--

<p>Defining execution details</p>	<pre><WebContexts> <WebContext> <LocalWebContextRef>gui1</LocalWebContextRef> <SignURLBase>https://www.sign-preprod1.nets.eu/sign/index.html?sref=</SignURLBase> <ErrorURLBase>https://www.merchant.com/statusurlbase?status=</ErrorURLBase> <StyleURL>https://www.merchant.com/mystyle.css</StyleURL> <ExitURL>https://www.merchant.com/exit.html</ExitURL> <AbortURL>https://www.merchant.com/abort.html</AbortURL> </WebContext> </WebContexts> <ExecutionDetails> <OrderDeadline>2024-04-26T12:54:20+00:00</OrderDeadline> <DisplayProcessInfo>NameStatusTime</DisplayProcessInfo> <GenerateOneTimeURLs>false</GenerateOneTimeURLs> <Steps> <Step> <StepNumber>1</StepNumber> <SigningProcess> <LocalWebContextRef>gui1</LocalWebContextRef> <LocalDocumentReference>doc1</LocalDocumentReference> <LocalSignerReference>user1</LocalSignerReference> <TerminateOrderOnSignerRejection>true</TerminateOrderOnSignerRejection> </SigningProcess> </Step> </Steps> </ExecutionDetails></pre>	<pre>{ "requestId": "b25bf0e7-b175-4345-a789-b666909870c4", "webContexts": [{ "objectId": "dd6379e2-7bf1-44cd-9233-dd8e48c82ef4", "signUrlBase": "https://www.sign-preprod1.nets.eu/sign/index.html?sref=", "statusUrlBase": "https://www.merchant.com/statusurlbase?status=", "styleUrl": "https://www.merchant.com/mystyle.css", "exitUrl": "https://www.merchant.com/successurl" }], "steps": [{ "objectId": "616379e2-7bf1-44cd-9233-dd8e48c82ef4", "stepNumber": 1, "stepDeadline": "2024-05-10T10:09:19.525Z", "signProcesses": [{ "objectId": "746379e2-7bf1-44cd-9233-dd8e48c82ef4", "documentReference": "a6aa79e2-7bf1-44cd-9233-dd8e48c82ef4", "signerReference": "566a79e2-7bf1-44cd-9233-dd8e48c82ef4", "signerRole": "SIGNER", "organisationRef": "6e6f79e2-7bf1-44cd-9233-dd8e48c82ef4", "webContextReference": "dd6379e2-7bf1-44cd-9233-dd8e48c82ef4" }] }] }</pre>	<p>AbortURL and ErrorURL are replaced with StatusURL in ESOS</p> <ul style="list-style-type: none"> In case of error : It will be appended with status code query parameter. eg. https://www.merchant.com/statusurlbase?statuscode=insufficientsignpermission&sref=9A2304F76BA08E5FFDF698A37766B8A0FA4FF1C0 In case of abort : It will just redirect without any parameter so merchant has to capture further input from signer eg. https://www.merchant.com/statusurlbase?sref=F981692ACAEAF7C4F4C6C93854DF3417FB56A0F
<p>Committing an order</p>	<p>NA</p>	<pre>{ "requestId": "RequestID100101", "objectId": "dd6379e2-7bf1-44cd-9233-dd8e48c82ef3", "orderStatus": "READY" }</pre>	<p>In Rest api, order creation process is divided in different steps. so merchant needs to commit the order to make it ready for signing.</p> <p>So need to call</p> <p>PUT <code>/{mid}/orders/{orderid}</code> : Update sign order</p>
<p>Getting sign processes</p>	<pre><GetSigningProcesses> <OrderID>1230544491717</OrderID> <LocalSignerReference>user1</LocalSignerReference> <SigningProcessStatusFilter>Active</SigningProcessStatusFilter> </GetSigningProcesses></pre>	<p>For Getting sign process use GetSigner call of ESOS API</p> <p>Eg : <code>https://{{env-url}}/esos/v1/1001/signers/{{esosOrderID}}</code></p>	<p>GetSigner will return the Signer detail along with all the signing processes mapped with that signer.</p> <p>If one signer has multiple documents then response will contain all signing process detail with redundant signer detail in all elements.</p>
<p>Updating order state</p>	<p>Eg for Cancel Order TSOS :</p> <pre><CancelOrder> <OrderID>123456777</OrderID> </CancelOrder></pre> <p>Eg for FinalizeOrder TSOS :</p> <pre><FinalizeOrder> <OrderID>123456788</OrderID> </FinalizeOrder></pre>	<p>Eg for Cancel order ESOS :</p> <pre>{ "requestId": "RequestID100101", "objectId": "{{esosOrderID}}", "orderStatus": "CANCELLED" }</pre> <p>Eg for FinalizeOrder ESOS :</p> <pre>{ "requestId": "RequestID100101", "objectId": "{{esosOrderID}}", "orderStatus": "COMPLETED" }</pre>	<p>Need to call - PUT <code>/{mid}/orders/{orderid}</code> : Update sign order</p> <p>TSOS offers Cancellation of order with CancelOrder call see example</p> <p>ESOS offers Cancellation of order by changing "orderStatus" : "CANCELLED" with OrderUpdate call see example</p> <p>TSOS & ESOS also offers FinalizeOrder functionality The FinalizeOrder offers functionality to finalize the sign order before all signers have signed the document. One signature on each document in the sign order is required before the sign order can be finalized.</p>

TSOS To ESOS migration

Updating order detail	ModifyOrderDeadline	{ "requestId": "RequestID100101", "objectId": "a8102ead-ebb0-442e-8e59-6ab31cd8ed23", "orderDeadline": "2024-06-10T12:23:40.976Z" // New Deadline }	Need to call - PUT <code>/{mid}/orders/{orderid}</code> : Update sign order with new orderDeadline

Notable changes to system behaviour and features:

Feature	Changes
Authentication / IncludeSSN	"signerIdScope": "SSN". It will be applicable for all authMethods.
Step / SigningProcess / TerminateOrderOnSignerRejection	SignatureRequired If true, the sign order state is set to Rejected if a signer rejects this sign process. If false, the signer may reject this sign process without "terminating" the sign order. The sign process is considered Complete even if the signer rejected to sign the document.
GetOrderDetailsResponse/ SigningProcessDetails/ RejectText	RejectText which contained the freetext the user can write when rejecting This will be returned as "GET <code>/{mid}/exec-details/{orderid}</code> " signProcesses/signerText
SigningTime	In GET <code>/{mid}/exec-details/{orderid}</code> lastUpdate (if signProcessStatus is COMPLETED then lastUpdated is a SigningTime)
AbortURL and ErrorURL	StatusURL <ul style="list-style-type: none"> in case of error : it will be appended with statusCode query parameter. eg. https://www.merchant.com/statusurlbase?statusCode=insufficientsignpermission&sref=9A2304F76BA08E5FFDF698A37766B8A0FA4FF1C0 in case of abort : it will just redirect without any parameter so merchant has to capture further input from signer eg. https://www.merchant.com/statusurlbase?sref=F981692ACAE AFC7C4F4C6C93854DF3417FB56A0F
GetPAdES <IncludeSSN>false</IncludeSSN>	GetPAdES returnPIL=true
LocalSignerReference	signerReference
LocalDocumentReference	documentReference
LocalWebContextRef	webContextReference
AcceptedPKIs eg : <AcceptedPKIs> <BankID/> </AcceptedPKIs>	"signerEids": { "signerIdScope": "SSN", "signMethods": [{ "name": "NO_BANKID" }] } In ESOS AcceptedPKIs tag is replaced by "signerEids" where we can mention both signMethods(Native Signing) and authMethods (ABS Signing)

Resources:

For additional guidance or support during the migration process, refer to the following resources:

- API Documentation: <https://www.nets.eu/developer/E-Signing/getstarted/Pages/E-Signing-APIs.aspx>
- Support Channels: <https://www.nets.eu/developer/E-Signing/contactus/Pages/default.aspx>

Conclusion:

The transition from XML-DSIG to JSON format signifies our commitment to enhancing the user experience and keeping pace with evolving industry standards. We appreciate your cooperation and understanding during this transition period.

If you have any questions or require further clarification, please don't hesitate to reach out to our support team.

Thank you for your continued partnership.

--

Add any tips and hints that you think need to be added to this document in the table below:
