Sign interface

Introduction

Below will be detailed the SOAP request for sign, change password ecc...

All the methods described are on interface:

```
https://<IP-APPLIANCE>:8080/SignEngineWeb/sign-services?wsdl
```

The SOAP request examples are generated using SoapUI, you can use this guide to configure SoapUI on your pc.

In this guide will be described the example of Soap Requests.

Credentials Object

All methods for sign require the ojbect Credentials is used to specify the device signature are you using for sign. This object is composed by this variables:

SOAP-credentials-object

```
<credentials>
        <username>?</username>
        <password>?</password>
        <idOtp>?</idOtp>
        <otp>?</otp>
        <securityCode>?</securityCode>
    <sessionKey>?</sessionKey>
</credentials>
```

According the device signature (automatic or remote) are you using you should populate different fields.

Automatic Signature

Below the example of Credentials :

SOAP-Credentials-object-automatic-signature

```
<credentials>
       <username>AHI123456</username>
       <password>13572468</password>
</credentials>
```

Fileds required:

- username
- password

Remote Signature

If you sign with the remote there are two ways:

- specify "idOtp" and "otp"
- specify the sessionKey

Example with "idOtp" and "otp":

SOAP-Credentials-object-remote-signature-idotp-otp

```
<credentials>
        <username>RHIP1234567</username>
        <password>13572468</password>
        <idOtp>501719</idOtp>
        <otp>150259</otp>
</credentials>
```

Example with "sessionKey"

SOAP-Credentials-object-remote-signature-sessionKey

```
<credentials>
        <username>RHIP1234567</username>
        <password>13572468</password>
        <sessionKey>sadlijhdfkjslherpoufdblkhesljherihbfdoihejheroihger</sessionKey>
 </credentials>
```

If you decide to sign with idOtp and OTP you must obtain the OTP code for sign (from SMS, App and Token) and idOtp.

How obtain the idOtp and OTP code

Below will described with SOAP request how obtain idOtp (with method getOtpList) and OTP code.

Obtain the idOtp

You can obtain the idOtp with method getOtpList. Below the example of SoapRequest. In this example we are using the devicename: "RHIP20102336 019765":

REQUEST-getOTPList

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://service.ws.nam
/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:getOTPList>
         <credentials>
            <username>RHIP12345</username>
         </credentials>
      </ser:getOTPList>
   </soapenv:Body>
</soapenv:Envelope>
```

In output the SOAP response will be:

RESPONSE-getOTPList

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:getOTPListResponse xmlns:ns2="http://service.ws.nam/">
         <return>
            <idOtp>501719</idOtp>
            <serialNumber>20210113-091031RJ2L1</serialNumber>
            <type>SMS</type>
         </return>
         <return>
            <idOtp>537430</idOtp>
            <serialNumber>20210305-163726L0PYF</serialNumber>
            <type>OTP GENERATOR</type>
         </return>
         <return>
            <idOtp>537433</idOtp>
            <serialNumber>20210305-163726F0I75</serialNumber>
            <type>OTP PUSH</type>
         </return>
      </ns2:getOTPListResponse>
   </soap:Body>
</soap:Envelope>
```

During the signing process, it is possible to choose between these two idOtps: 501719 (associated with OTP SMS) and the idOTP: 537430 (associated with OTP GENERATOR).

It is not possible to use OTP PUSH, they are used for other purposes, not for signing.

For the signature we can choose two types of idOTP: 501719 or 537430.

Obtain the OTP code

With OTP SMS we can obtain the code using the method "sendOtpBySMS" like in this SOAP request:

REQUEST-sendOTPBySMS

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://service.ws.nam
/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:sendOtpBySMS>
         <credentials>
            <username>RHIP12345</username>
         </credentials>
      </ser:sendOtpBySMS>
   </soapenv:Body>
</soapenv:Envelope>
```

If everything is ok, in output response will be:

RESPONSE-sendOTPBySMS

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:sendOtpBySMSResponse xmlns:ns2="http://service.ws.nam/"/>
   </soap:Body>
</soap:Envelope>
```

On your mobile phone, you will receive an SMS containing the OTP code (composed of 6 numbers) for signature. Now, for example, we have received the code: “214196”.

While with OTP App and Token you don't require the method of SWS because you can read the OTP code on Token display or on your smartphone display (if you are using the App).

Manage the sessionKey

Below will be describe the SOAP request example for obtain the sessionKey, check if the sessionKey is valid and destroy the sessionKey

Obtain the sessionKey

Below the SOAP request example for create the openSession:

REQUEST-openSession

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
        xmlns:ser="http://service.ws.nam/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:openSession>
         <credentials>
            <idOtp>501719</idOtp>
            <otp>150259</otp>
            <password>13572468</password>
            <username>RHIP12345</username>
         </credentials>
      </ser:openSession>
   </soapenv:Body>
</soapenv:Envelope>
```

In output will obtain the value of sessionKey which will be used for the signature:

RESPONSE-REMOTE-openSession

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:openSessionResponse xmlns:ns2="http://service.ws.nam/">
                <return>
                        f4lf7bq/cCxW6mTgL3iGjFEST5cEAZjgLnXvV3hUFzFHcTvjlH3FOkJy+kv/0Zsv1
                        uNK0S7L6jMqHYSspBz+CZl7h3r5IEP2FqrK7WJQTVyrNfyr/trZmDgxYOLuACyoZVUFIlnck5Lkjihui
                        sv+gZeB68Spwm+cNDdQQdUS3ngzJavHXxo9ADCX6VDIKKMe
/AY0v+R51XWE90JF5LfKEThlv1OCpQC5nhnW8WKOFOm
                        P4vM90d79JhFYGVVSZWtnTQ9Dg8pOMvg9wwxNm3uGkKKaS7oTp1ewd+eCG/uSC9k3H2w9GB6vQLHQEbn6d
                        VVMcsIqJ0RMmZ2IgraD+scb4Q==
                </return>
      </ns2:openSessionResponse>
   </soap:Body>
</soap:Envelope>
```

The sessionKey just obtained is valid for three minutes (it is not possible to edit this value!). After it expires, you will need to generate another sessionKey using openSession method and new OTP code (it is not possible to use the same OTP already in use).

Check the sessionKey status

Below the SOAP request example for check the sessionKey status:

REQUEST-remote-getRemainingTimeForSession

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:ser="http://service.ws.nam/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:getRemainingTimeForSession>
         <credentials>
            <username>RHIP12345</username>
                   <sessionKey>
                              f4lf7bq/cCxW6mTgL3iGjFEST5cEAZjgLnXvV3hUFzFHcTvjlH3FOkJy+kv
/0Zsv1                              uNK0S7L6jMqHYSspBz+CZl7h3r5IEP2FqrK7WJQTVyrNfyr
/trZmDgxYOLuACyoZVUFIlnck5Lkjihui
sv+gZeB68Spwm+cNDdQQdUS3ngzJavHXxo9ADCX6VDIKKMe
/AY0v+R51XWE90JF5LfKEThlv1OCpQC5nhnW8WKOFOm
P4vM90d79JhFYGVVSZWtnTQ9Dg8pOMvg9wwxNm3uGkKKaS7oTp1ewd+eCG
/uSC9k3H2w9GB6vQLHQEbn6d                                          VVMcsIqJ0RMmZ2IgraD+scb4Q==
                  </sessionKey>
               </credentials>
      </ser:getRemainingTimeForSession>
   </soapenv:Body>
</soapenv:Envelope>
```

The SOAP response will be:

RESPONSE-remote-getRemainingTimeForSession

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:getRemainingTimeForSessionResponse xmlns:ns2="http://service.ws.nam/">
         <return>167</return>
      </ns2:getRemainingTimeForSessionResponse>
   </soap:Body>
</soap:Envelope>
```

Where 167 is the seconds until the session is active. After 180 seconds from creation, the session will be automatically deleted, but for good practice, close the session before it expires.

You can destroy the session manually before it expires with the method **closeSession**.

Destroy the sessionKey

Below the example of SOAP request for destroy the session:

REQUEST-remote-closeSession

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://service.ws.nam
/">
        <soapenv:Header/>
                <soapenv:Body>
                        <ser:closeSession>
                                <credentials>
                                  <username>RHIP12345</username>
<sessionKey>f4lf7bq/cCxW6mTgL3iGjFEST5cEAZjgLnXvV3hUFzFHcTvjlH3FOkJy+kv
/0Zsv1uNK0S7L6jMqHYSspBz+CZl7h3r5IEP2FqrK7WJQTVyrNfyr
/trZmDgxYOLuACyoZVUFIlnck5Lkjihuisv+gZeB68Spwm+cNDdQQdUS3ngzJavHXxo9ADCX6VDIKKMe
/AY0v+R51XWE90JF5LfKEThlv1OCpQC5nhnW8WKOFOm P4vM90d79JhFYGVVSZWtnTQ9Dg8pOMvg9wwxNm3uGkKKaS7oTp1ewd+eCG
/uSC9k3H2w9GB6vQLHQEbn6dVVMcsIqJ0RMmZ2IgraD+scb4Q==
                                        </sessionKey>
                                </credentials>
                        </ser:closeSession>
                </soapenv:Body>
</soapenv:Envelope>
```

The SOAP response will be ever like this:

RESPONSE-remote-closeSession

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
        <soap:Body>
                <ns2:closeSessionResponse xmlns:ns2="http://service.ws.nam/"/>
        </soap:Body>
</soap:Envelope>
```

Summarize

The credentials object for automatic signature is composed like in this example:

REQUEST-AUTOMATIC-Credentials

```
<credentials>
        <username>AHIP12345</username>
        <password>1357268</password>
</credentials>
```

With remote signature if you don't use the sessionKey the object Credentials will be:

REQUEST-Credentials-Remote-OTP-SMS

```
<credentials>
        <password>13572468</password>
        <username>RHIP12345</username>
        <idOtp>501719</idOtp>
        <otp>150259</otp>
</credentials>
```

While if you are using the sessionKey the object Credentials will be:

REQUEST-Credentials-Remote-OTP-SMS

```
<credentials>
        <password>13572468</password>
        <username>RHIP12345</username>
<sessionKey>f4lf7bq/cCxW6mTgL3iGjFEST5cEAZjgLnXvV3hUFzFHcTvjlH3FOkJy+kv
/0Zsv1uNK0S7L6jMqHYSspBz+CZl7h3r5IEP2FqrK7WJQTVyrNfyr
/trZmDgxYOLuACyoZVUFIlnck5Lkjihuisv+gZeB68Spwm+cNDdQQdUS3ngzJavHXxo9ADCX6VDIKKMe
/AY0v+R51XWE90JF5LfKEThlv1OCpQC5nhnW8WKOFOm P4vM90d79JhFYGVVSZWtnTQ9Dg8pOMvg9wwxNm3uGkKKaS7oTp1ewd+eCG
/uSC9k3H2w9GB6vQLHQEbn6dVVMcsIqJ0RMmZ2IgraD+scb4Q==
        </sessionKey>
</credentials>
```

Methods for sign

Below will be described the SOAP request example for every type of signature:

- Pades
- Cades
- Xades

SignPades

The SOAP request for create Pades signature:

signPades

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://service.ws.nam
/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:signPAdES>
         <credentials>
            <password>YOUR-DEVICE-PASSWORD</password>
            <username>YOUR-DEVICE-NAME</username>
         </credentials>
         <buffer>BASE64-TO-SIGN</buffer>
                <PAdESPreferences>
                    <level>B</level>
                    <signerImage>
                               <imageVisible>true</imageVisible>
                               <image>BASE64-IMAGE-LOGO</image>
                               <x>30</x>
                               <y>30</y>
                               <width>50</width>
                               <height>50</height>
                               <signerName>Name of Signer</signerName>
                   </signerImage>
                </PAdESPreferences>
    </ser:signPAdES>
   </soapenv:Body>
</soapenv:Envelope>
```

At this link is possible to see the full example (with file to sign and logo image) of signature Pades with appereance.

While at this link , you can find an example of Level T (signature+timestamp)

SignPadesMultiFieldName

The SOAP request for create Pades signature using signatures field:

signPadesMultiFieldName

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://service.ws.nam
/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:signPAdES>
         <credentials>
            <password>YOUR-DEVICE-PASSWORD</password>
            <username>YOUR-DEVICE-NAME</username>
                        <sessionKey>SESSION_KEY_RECEIVED_FROM_OPEN_SESSION</sessionKey>
          </credentials>
         <buffer>BASE64-TO-SIGN</buffer>
                <PAdESPreferences>
            <level>B</level>
            <signerImage>

                           <fieldsNameList>SignatureField-1</fieldsNameList>
               <fieldsNameList>SignatureField-2</fieldsNameList>

               <!-- THIS OPTION set to true allow to sign all signatures fields available -->
               <!-- <signAllFields>false</signAllFields> -->

               <signerName>NAME OF SIGNER</signerName>

            </signerImage>
            <withSignatureField>true</withSignatureField>
         </PAdESPreferences>
     </ser:signPAdES>
   </soapenv:Body>
</soapenv:Envelope>
```

At this link is possible to see the full example (with file to sign and logo image) of signature Pades with appereance.

NOTE: in this example the signatures fields: "SignatureField-1" and "SignatureField-2" already exist in a PDF

The response will be the complex object: "PadesWithMultiFieldName" or generate a WSException if don't sign at least one signature field.

Below the response ok, when all elements of "fieldsNameList" are signed (the file is fully signed):

SOAP-response-signPadesMultiFieldName-output-fully-signed

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:signPAdESMultiFieldNameResponse xmlns:ns2="http://service.ws.nam/">
         <return>
            <signedContent>BASE-64-OF-PDF-WITH-ELEMENTS-OF-LIST-FIELDSNAMELIST</signedContent>
         </return>
      </ns2:signPAdESMultiFieldNameResponse>
   </soap:Body>
</soap:Envelope>
```

While if the field are signed partially (for example the session key has expired) therefore the file is partially signed, the response will be:

SOAP-response-signPadesMultiFieldName-output-partially-signed

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:signPAdESMultiFieldNameResponse xmlns:ns2="http://service.ws.nam/">
         <return>
                <signedContent>BASE-64-OF-PDF-WITH-ELEMENTS-OF-LIST-FIELDSNAMELIST</signedContent>
            <serviceError>
               <code>69</code>
               <message>Session key scaduta</message>
            </serviceError>
            <remainingFieldNames>SignatureField-2</remainingFieldNames>
         </return>
      </ns2:signPAdESMultiFieldNameResponse>
   </soap:Body>
</soap:Envelope>
```

The tag:

- "signedContent" contains the partially signed file, because not all fields in a list have been signed
- "serviceError" contains the details about the cause because all signatures fields have been signed
- "remainingFieldNames" contains the list of field remaining to sign

In this case, the response will be the input of the new "signPadesMultiFieldName" request

While if you insert credentials.password, credentials.sessionKey, signature field not exist. In output will obtain a WSException like this:

SOAP-response-signPadesMultiFieldName-output-partially-signed

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <soap:Fault>
         <faultcode>soap:Server</faultcode>
         <faultstring>Richiesta non valida</faultstring>
         <detail>
            <ns2:WSException xmlns:ns2="http://service.ws.nam/">
               <error>105</error>
               <message>Richiesta non valida</message>
            </ns2:WSException>
         </detail>
      </soap:Fault>
   </soap:Body>
</soap:Envelope>
```

SignCades

The SOAP request for create Cades signature:

signCades

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://service.ws.nam
/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:signCAdES>
         <credentials>
            <password>YOUR-DEVICE-PASSWORD</password>
            <username>YOUR-DEVICE-NAME</username>
         </credentials>
         <buffer>VGhpcyBpcyB0aGUgZmlsZSB0byBiZSBzaWduZWQgZm9yIHRlc3Qu</buffer>
         <CAdESPreferences>
            <level>B</level>
         </CAdESPreferences>
      </ser:signCAdES>
   </soapenv:Body>
</soapenv:Envelope>
```

In this example the buffer to sign is "txt" files.

The SOAP response will be:

SOAP-response-signCades

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:signCAdESResponse xmlns:ns2="http://service.ws.nam/">
<return>MIIKVwYJKoZIhvcNAQcCoIIKSDCCCkQCAQExDzANBglghkgBZQMEAgEFADA2BgkqhkiG9w0BBwGgKQQnVGhpcyBpcyB0aGUgZmlsZ
SB0byBiZSBzaWduZWQgZm9yIHRlc3QuoIIG0DCCBswwggW0oAMCAQICCFzwqTJXlRldMA0GCSqGSIb3DQEBCwUAMH0xJjAkBgNVBAMMHU5hbW
lyaWFsIENBIEZpcm1hIFF1YWxpZmljYXRhMSAwHgYDVQQLDBdDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTEkMCIGA1UECgwbTmFtaXJpYWwgUy5
wLkEuLzAyMDQ2NTcwNDI2MQswCQYDVQQGEwJJVDAeFw0xODAxMjMxNjM3MDBaFw0yNDA0MjMxNjM3MDBaMIGnMQswCQYDVQQGEwJJVDEVMBMG
A1UECgwMTk9OIFBSRVNFTlRFMRUwEwYDVQQEDAxERU1PIENPR05PTUUxEjAQBgNVBCoMCURFTU8gTk9NRTEcMBoGA1UEBRMTSVQ6RE1DRE5NM
TVUMTBBMjcxTzEfMB0GA1UEAwwWREVNTyBOT01FIERFTU8gQ09HTk9NRTEXMBUGA1UELhMOREVNTzEyMzQ1Njc4OTAwggEiMA0GCSqGSIb3DQ
EBAQUAA4IBDwAwggEKAoIBAQD4i6k2DI5cXUjKDiPfx6qFl5kDz+wCpWRFnnWcL137vi4KhupGpcBEkvr298xZp82jFwroQ2I4NUkAuGBx7K3
XZZwO9mb+qZ2PfIeNzPjHTt/0UOMGe2wP0oEZY1cQdriXpg4tBbv5dPWMrOTrt/OA4
/QuAth5Uekm0O7OC+CsecfyhklupjfABmjb4CsUIYF1qa4n4SaklE8Rtrbz9PjZxJbK902sTjaQnmK95Yv5cED6TtANiVtMKs
/eLgoEFN8vx62Kh3V+dHHG+4zXfmod3zv1OkVL0KAXIFBShve93ohcr/EhspkR3/YsHWp6y5EwCkhbGJPM
/SXwDhjdVqAbAgMBAAGjggMjMIIDHzCBogYIKwYBBQUHAQEEgZUwgZIwVAYIKwYBBQUHMAKGSGh0dHBzOi8vZG9jcy50ZXN0LmZpcm1hY2Vyd
GEuaXQvZG9jdW1lbnRzL05hbWlyaWFsQ0FGaXJtYVF1YWxpZmljYXRhLmNydDA6BggrBgEFBQcwAYYuaHR0cDovL29jc3AudGVzdC5maXJtYW
NlcnRhLml0L29jc3AvY2VydHN0YXR1czAdBgNVHQ4EFgQUvqgLdh6cX6Usi37YMvetoDUQ+AcwHwYDVR0jBBgwFoAU2zgIULe2L5buO+w+tGZ
UwkqWqd4wLwYIKwYBBQUHAQMEIzAhMAgGBgQAjkYBATALBgYEAI5GAQMCARQwCAYGBACORgEEMIIBqgYDVR0gBIIBoTCCAZ0wggGZBgsrBgEE
AYKaawEBAzCCAYgwMAYIKwYBBQUHAgEWJGh0dHA6Ly93d3cuZmlybWFjZXJ0YS5pdC9tYW51YWxpLU1PLzCCAVIGCCsGAQUFBwICMIIBRB6CA
UAASQBsACAAcAByAGUAcwBlAG4AdABlACAAYwBlAHIAdABpAGYAaQBjAGEAdABvACAAcwBvAGwAbwAgAHAAZQ
ByACAAZgBpAHIAbQBlACAAYQBwAHAAbwBzAHQAZQAgAGMAbwBuACAAcAByAG8AYwBlAGQAdQByAGEIAbHAHUAdABvAG0AYQB0AGkAYwBhAC4
AIABUAGgAaQBzACAAYwBlAHIAdABpAGYAaQBjAGEAdABlACAAbQBhAHkAIABvAG4AbAB5ACAAYgBlACAAdQBzAGUAZAAgAGYAbwByACAAdQBu
AGEAdAB0AGUAbgBkAGUAZAAvAGEAdQB0AG8AbQBhAHQAZQBkACAAZABpAGcAaQB0AGEAbAAgAHMAaQBnAG4AYQB0AHUAcgBlAHMALjBJBgNVH
R8EQjBAMD6gPKA6hjhodHRwOi8vY3JsLnRlc3QuZmlybWFjZXJ0YS5pdC9GaXJtYUNlcnRhUXVhbGlmaWNhdGExLmNybDAOBgNVHQ8BAf8EBA
MCBkAwDQYJKoZIhvcNAQELBQADggEBAHnOmp8mICiahhf58HEcCjzWkjopGyaAERsWFScRBdg1k4Of3JOqi3QK47F83ai41jRNC+KBKYP
/mUSwEok3dERW7rVH3xKXo7GmCDUm7Hk1O7B8N+ITlSKniMksvSpkx3GEwedr1VD0Cgqj/MQa8wMP+xoXioBrdIIsTShk
/qi5ecrbdFXNhoeA3zO/vOn7WFPFC6xR+LKWnOEHW
/FtcOawcWV8hVNhG77CD+wyOnpypb1HKUOVSwFqDVvX7JF8u2809+m0ySqoZ621ITeTQNCw9km26bMKy7D4VefN3NIQbak8b0ftWzxWsngkvi
H5MFPSq5JWI0IZOjOhPiHntksxggMgMIIDHAIBATCBiTB9MSYwJAYDVQQDDB1OYW1pcmlhbCBDQSBGaXJtYSBRdWFsaWZpY2F0YTEgMB4GA1U
ECwwXQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkxJDAiBgNVBAoMG05hbWlyaWFsIFMucC5BLi8wMjA0NjU3MDQyNjELMAkGA1UEBhMCSVQCCFzw
qTJXlRldMA0GCWCGSAFlAwQCAQUAoIIBZzAYBgkqhkiG9w0BCQMxCwYJKoZIhvcNAQcBMBwGCSqGSIb3DQEJBTEPFw0yMjA4MTAwODAxNTZaM
C0GCSqGSIb3DQEJNDEgMB4wDQYJYIZIAWUDBAIBBQChDQYJKoZIhvcNAQELBQAwLwYJKoZIhvcNAQkEMSIEIIyS24577o6HAeqQ5OU2H1lF5J
MnJaRgd8ISguoanmBYMIHMBgsqhkiG9w0BCRACLzGBvDCBuTCBtjCBswQgxbkITFBriuSqr8M6xRqZCQN72rMbhDr7YtuXrw186f4wgY4wgYG
kfzB9MSYwJAYDVQQDDB1OYW1pcmlhbCBDQSBGaXJtYSBRdWFsaWZpY2F0YTEgMB4GA1UECwwXQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkxJDAi
BgNVBAoMG05hbWlyaWFsIFMucC5BLi8wMjA0NjU3MDQyNjELMAkGA1UEBhMCSVQCCFzwqTJXlRldMA0GCSqGSIb3DQEBCwUABIIBAIgAfXNg3
DxOPdOe5QTtfTMmV4zU/+cTGLT4xaB2x2/14o3NmkRjHCMq4NBDf8XifOo8YTrVQYqYNlxDaW5JpjTpqfbWun4wuIdkQqsg6TiRTiy3w
/v01oMk/X1sj7H+6wSffcRmIV5dwTIlHUx0MRXiPA00OaCsZTO852xwkXUB7z8/jaWfUK4bLoz
/ckgFmV3YhRLhth7sLWzVjgUELd6ukCiwCftP9R4KEwXEYu4YmBW9pknFDrDGZgTTYChsITLtJkNarzl/4JtxXcA7
/FALCxuyOHcnYNta+iCW4N3I/E0PIVzQ5XibNraAFO1ulJIzA1yC+hU4IjADJeAPoEE=</return>
      </ns2:signCAdESResponse>
   </soap:Body>
</soap:Envelope>
```

SignCades Detached

If you want make the Cades detached signature, SWS not require all files to sign, but only the hash. The tag "buffer" will be the hash of the file.

For example if we want the cades detached signature of this PDF the procedure is:

1) Calculate the hash of this file, for example with the openssl:

```
openssl dgst -sha256 -binary FILE_TO_BE_SIGN | openssl enc -a
```

And in output will obtain the hash to sign, will be:

```
HASH TO SIGN = msj3f4hJCSELbMkWjkFwNrf0XhkebTnAKaKhx4686DY=
```

2) Now can execute the method signCades, using the field "cadesPreferences.detached=true":

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://service.ws.nam
/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:signCAdES>
         <credentials>
            <username>YOUR-DEVICE-USERNAME</username>
            <password>YOUR-DEVICE-PASSWORD</password>
         </credentials>
         <buffer>msj3f4hJCSELbMkWjkFwNrf0XhkebTnAKaKhx4686DY=</buffer>
         <CAdESPreferences>
            <detached>true</detached>
         </CAdESPreferences>
      </ser:signCAdES>
   </soapenv:Body>
</soapenv:Envelope>
```

The SOAP response will be:

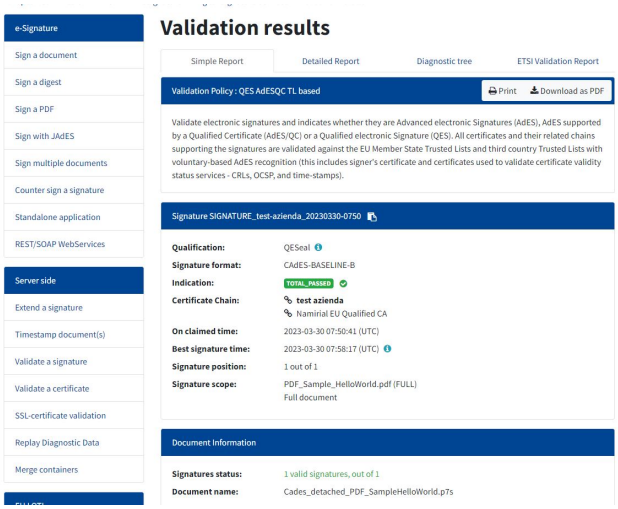
```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:signCAdESResponse xmlns:ns2="http://service.ws.nam/">
<return>MIIKKQYJKoZIhvcNAQcCoIIKGjCCChYCAQExDzANBglghkgBZQMEAgEFADALBgkqhkiG9w0BBwGggga2MIIGsjCCBJqgAwIBAgIIfl5W0
/FI108wDQYJKoZIhvcNAQELBQAwgYcxITAfBgNVBAMMGE5hbWlyaWFsIEVVIFF1YWxpZmllZCBDQTEfMB0GA1UECwwWVHJ1c3QgU2Vydmljc
BQcm92aWRlcjEYMBYGA1UECgwPTmFtaXJpYWwgUy5wLkEuMRowGAYDVQRhDBFWQVRJVC0wMjA0NjU3MDQyNjELMAkGA1UEBhMCSVQwHhcNMTk
wODA1MDc0NjAwWhcNMjUwODA1MDc0NjAwWjBmMQ0wCwYDVQQuEwRJRDkyMRUwEwYDVQQDDAx0ZXN0IGF6aWVuZGExFTATBgNVBAoMDHRlc3Qg
YXppZW5kYTEaMBgGA1UEYQwRVkFUSVQtMDAwMDAwMDAxCzAJBgNVBAYTAklUMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA4
TXG/VhgZhtRf8HkRmLa2NDEfG5CnloPR5yowp66oqQTovMqF
/nbvY+wTOXIxWHYbAL1mTeEqwNTWBwKrGDFe3JP3EJSAnRa8BMUJi5B2JUfud4fSZSaeeJkoCu2ggZ3/64kFdC2yxPlpwn
/oKSNJGiPyvhNW2JJV9NtlsUDdVOmHug5uFTQoo423CvmcRSz6kYhrWBKyQ9TGGPwukDqpKhiB+
/01pL7DVI4X4gNi1gnFQoBbBRG2rSAB8xLXxYP
/aWiMF8c3VZX5xELIGumQEfxOQRJ07kr1Qlh1MXFWwadRa8lZnQpg5vMAjthxXnr0GyPJcim5xOrbCRwGlhVsQIDAQABo4ICQDCCAjwwgYcGC
CsGAQUFBwEBBHsweTA+BggrBgEFBQcwAoYyaHR0cHM6Ly9kb2NzLm5hbWlyaWFsdHNwLmNvbS9kb2N1bWVudHMvTmFtQ0E0Sy5jcnQwNwYIKw
YBBQUHMAGGK2h0dHA6Ly9vY3NwLm5hbWlyaWFsdHNwLmNvbS9vY3NwL2NlcnRzdGF0dXMwHQYDVR0OBBYEFBT3hX0qb5n0V7hKLCuE5QNZCIl
1MB8GA1UdIwQYMBaAFGO4zbhJUuXnCXtXjPt6QQ5BqnhZMIHNBggrBgEFBQcBAwSBwDCBvTAIBgYEAI5GAQEwCwYGBACORgEDAgEUMAgGBgQA
jkYBBDATBgYEAI5GAQYwCQYHBACORgEGAjCBhAYGBACORgEFMHowOxY1aHR0cHM6Ly9kb2NzLm5hbWlyaWFsdHNwLmNvbS9kb2N1bWVudHMvU
ERTL1BEU19lbi5wZGYTAmVuMDsWNWh0dHBzOi8vZG9jcy5uYW1pcmlhbHRzcC5jb20vZG9jdW1lbnRzL1BEUy9QRFNfaXQucGRmEwJpdDBaBg
NVHSAEUzBRMDoGCysGAQQBgpprAQIDMCswKQYIKwYBBQUHAgEWHWh0dHBzOi8vZG9jcy5uYW1pcmlhbHRzcC5jb20vMAkGBwQAi+xAAQMwCAY
GBACPegECMDQGA1UdHwQtMCswKaAnoCWGI2h0dHA6Ly9jcmwubmFtaXJpYWx0c3AuY29tL0NBNEsuY3JsMA4GA1UdDwEB
/wQEAwIGQDANBgkqhkiG9w0BAQsFAAOCAgEAhlPP1QdsAqYEemj8MbHbF8a
/rAG4op1CbP4YsPr4Y5YKxr2TRuf4L4KcbFH8KpLPIXzw+2CvzeB2IGc00Ahsl0s0d1hQimLKL6SEkS3uN3sxq6f
/i2dMS9IuCLcwyn8ZpCmvp2bqqj
/fCqZdoiOIF1WIOuqRUfLSc95M+FMrWXzDovQS9Y8IerHiwbnq3fI3VuYdvKhx0Lril7VcpmvK5zfXxgsE9FmmH
/aJvdcrjPAjI8Sjzz1vsB74MmqtFFk5Pnd1NlOHGW8KaTeELpuOlmxsz2qPFO7inoSBuKMEFz3W3IvB3UnKuRbFEUtjyFYgYTpPTQcsX0kaJM
u/S76hRLTpvZ5zs+3AFgfCGbpH7kTCW1MTNT4oYlWEXYctF4Mqg8giMaBetfO6zB954Qqu3eqFvlDZOHC1RbL
/F2at61rmnSXSbzRkev+VevKUGjIqThMq5loQSP8mxCquUvaa4epJ
/tDzmLYdYwnoN7fzofA4ZkAQPLvt4Kv4IML612CM0kK+1mkEuhcTJN0h8l6Kax40q+Ld8qjgmvGAIMe
/28dfkIpQS7gm70N1nCcKdKMngcQ881LWYnbyY
/ba5BBCwXeqWKp1+oqorrqFYiWONSH9SmNfI0gG6leEjL5k8nArkKtNefScw5tv1LZXAC4uaZjD++iJ8jVMPn7khf9qKR8xggM3MIIDMwIBAT
CBlDCBhzEhMB8GA1UEAwwYTmFtaXJpYWwgRVUgUXVhbGlmaWVkIENBMR8wHQYDVQQLDBZUcnVzdCBTZXJ2aWNlIFByb3ZpZGVyMRgwFgYDVQQ
KDA9OYW1pcmlhbCBTLnAuQS4xGjAYBgNVBGEMEVZBVElULTAyMDQ2NTcwNDI2MQswCQYDVQQGEwJJVAIIfl5W0
/FI108wDQYJYIZIAWUDBAIBBQCgggFzMBgGCSqGSIb3DQEJAzELBgkqhkiG9w0BBwEwHAYJKoZIhvcNAQkFMQ8XDTIzMDMzMDA3NTA0MVowLQ
YJKoZIhvcNAQk0MSAwHjANBglghkgBZQMEAgEFAKENBgkqhkiG9w0BAQsFADAvBgkqhkiG9w0BCQQxIgQgmsj3f4hJCSELbMkWjkFwNrf0Xhk
ebTnAKaKhx4686DYwgdgGCyqGSIb3DQEJEAIvMYHIMIHFMIHCMIG/BCADIK96sQnXsOl13GxGXQqxA66Ryjv45gg1ab
/b9RM4PjCBmjCBjaSBijCBhzEhMB8GA1UEAwwYTmFtaXJpYWwgRVUgUXVhbGlmaWVkIENBMR8wHQYDVQQLDBZUcnVzdCBTZXJ2aWNlIFByb3Z
pZGVyMRgwFgYDVQQKDA9OYW1pcmlhbCBTLnAuQS4xGjAYBgNVBGEMEVZBVElULTAyMDQ2NTcwNDI2MQswCQYDVQQGEwJJVAIIfl5W0
/FI108wDQYJKoZIhvcNAQELBQAEggEAqEnv3NnVIrlLivnjrNyI85MjqZbaS3bt6FcfzREP+hmBSQ7DhXXqszRu2vJ+IkYod
/MhovvTwb2s6Y2uczugkVN3Jc
/uswUf6lIlCQn8F6bsDtl5xaTdZY75hu6mm8SJdNGyrTto7Zwfl9vm3yTgt5Z3M+ORM4aHqsBYHVZWlqTyXR58uz8udSMznN2Cfrk+JCbMGiv
TCTMhujCFLySocON/ZWB1KOEGK
/m7Ook9qZ4Ow9VQJOBbWEVLU4A2FhNSnJtnqky6jPEhnAZ9ssazJ4fhT8o4fUbs71AUnBz8A02BV5AcgK9pXvlrcx8pOwGEyzaxo26RFs9Aoc
pNhhE3rA==</return>
      </ns2:signCAdESResponse>
   </soap:Body>
</soap:Envelope>
```


In the tag "return" there is the cades detached signature, you MUST decode the content of this tag and will obtain this file: Cades_detached_PDF_SampleHelloWorld.p7s

3) Finally we have the cades detached signature and we ready to verify the signature at this link:

- field "signed file" upload the detached signature
- field "original file" upload the file "FILE_TO_BE_SIGN"

And the output will be:



SignXades

The SOAP request for create Xades signature:

SOAP-request-signXades

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://service.ws.nam
/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:signXAdES>
         <credentials>
                        <username>YOUR-DEVICE-NAME</username>
                        <password>YOUR-DEVICE-PASSWORD</password>
         </credentials>
         <buffer>PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4NCjxUZWxlbWF0
aWNvPg0KCTx0bz5Ub3ZlPC90bz4NCgk8ZnJvbT5KYW5pPC9mcm9tPg0KCTxoZWFk
aW5nPlJlbWluZGVyPC9oZWFkaW5nPg0KCTxib2R5PkRvbid0IGZvcmdldCBtZSB0
aGlzIHdlZWtlbmQhPC9ib2R5Pg0KPC9UZWxlbWF0aWNvPg==</buffer>
         <XAdESPreferences>
           <level>B</level>
         </XAdESPreferences>
      </ser:signXAdES>
   </soapenv:Body>
</soapenv:Envelope>
```

The SOAP response will be:

SOAP-response-signXades

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:signCAdESResponse xmlns:ns2="http://service.ws.nam/">

<return>MIIKVwYJKoZIhvcNAQcCoIIKSDCCCkQCAQExDzANBglghkgBZQMEAgEFADA2BgkqhkiG9w0BBwGgKQQnVGhpcyBpcyB0aGUgZmlsZ
SB0byBiZSBzaWduZWQgZm9yIHRlc3QuoIIG0DCCBswwggW0oAMCAQICCFzwqTJXlRldMA0GCSqGSIb3DQEBCwUAMH0xJjAkBgNVBAMMHU5hbW
lyaWFsIENBIEZpcm1hIFF1YWxpZmljYXRhMSAwHgYDVQQLDBdDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTEkMCIGA1UECgwbTmFtaXJpYWwgUy5
wLkEuLzAyMDQ2NTcwNDI2MQswCQYDVQQGEwJJVDAeFw0xODAxMjMxNjM3MDBaFw0yNDA0MjMxNjM3MDBaMIGnMQswCQYDVQQGEwJJVDEVMBMG
A1UECgwMTk9OIFBSRVNFTlRFMRUwEwYDVQQEDAxERU1PIENPR05PTUUxEjAQBgNVBCoMCURFTU8gTk9NRTEcMBoGA1UEBRMTSVQ6RE1DRE5NM
TVUMTBBMjcxTzEfMB0GA1UEAwwWREVNTyBOT01FIERFTU8gQ09HTk9NRTEXMBUGA1UELhMOREVNTzEyMzQ1Njc4OTAwggEiMA0GCSqGSIb3DQ
EBAQUAA4IBDwAwggEKAoIBAQD4i6k2DI5cXUjKDiPfx6qFl5kDz+wCpWRFnnWcL137vi4KhupGpcBEkvr298xZp82jFwroQ2I4NUkAuGBx7K3
XZZwO9mb+qZ2PfIeNzPjHTt/0UOMGe2wP0oEZY1cQdriXpg4tBbv5dPWMrOTrt/OA4
/QuAth5Uekm0O7OC+CsecfyhklupjfABmjb4CsUIYF1qa4n4SaklE8Rtrbz9PjZxJbK902sTjaQnmK95Yv5cED6TtANiVtMKs
/eLgoEFN8vx62Kh3V+dHHG+4zXfmod3zv1OkVL0KAXIFBShve93ohcr/EhspkR3/YsHWp6y5EwCkhbGJPM
/SXwDhjdVqAbAgMBAAGjggMjMIIDHzCBogYIKwYBBQUHAQEEgZUwgZIwVAYIKwYBBQUHMAKGSGh0dHBzOi8vZG9jcy50ZXN0LmZpcm1hY2Vyd
GEuaXQvZG9jdW1lbnRzL05hbWlyaWFsQ0FGaXJtYVF1YWxpZmljYXRhLmNydDA6BggrBgEFBQcwAYYuaHR0cDovL29jc3AudGVzdC5maXJtYW
NlcnRhLml0L29jc3AvY2VydHN0YXR1czAdBgNVHQ4EFgQUvqgLdh6cX6Usi37YMvetoDUQ+AcwHwYDVR0jBBgwFoAU2zgIULe2L5buO+w+tGZ
UwkqWqd4wLwYIKwYBBQUHAQMEIzAhMAgGBgQAjkYBATALBgYEAI5GAQMCARQwCAYGBACORgEEMIIBqgYDVR0gBIIBoTCCAZ0wggGZBgsrBgEE
AYKaawEBAzCCAYgwMAYIKwYBBQUHAgEWJGh0dHA6Ly93d3cuZmlybWFjZXJ0YS5pdC9tYW51YWxpLU1PLzCCAVIGCCsGAQUFBwICMIIBRB6CA
UAASQBsACAAcAByAGUAcwBlAG4AdABlACAAYwBlAHIAdABpAGYAaQBjAGEAdABvACAA6AAgAHYAYQBsAGkAZABvACAAcwBvAGwAbwAgAHAAZQ
ByACAAZgBpAHIAbQBlACAAYQBwAHAAbwBzAHQAZQAgAGMAbwBuACAAcAByAG8AYwBlAGQAdQByAGEIABhAHUAdABvAG0AYQB0AGkAYwBhAC4
AIABUAGgAaQBzACAAYwBlAHIAdABpAGYAaQBjAGEAdABlACAAbQBhAHkAIABvAG4AbAB5ACAAYgBlACAAdQBzAGUAZAAgAGYAbwByACAAdQBu
AGEAdAB0AGUAbgBkAGUAZAAvAGEAdQB0AG8AbQBhAHQAZQBkACAAZABpAGcAaQB0AGEAbAAgAHMAaQBnAG4AYQB0AHUAcgBlAHMALjBJBgNVH
R8EQjBAMD6gPKA6hjhodHRwOi8vY3JsLnRlc3QuZmlybWFjZXJ0YS5pdC9GaXJtYUNlcnRhUXVhbGlmaWNhdGExLmNybDAOBgNVHQ8BAf8EBA
MCBkAwDQYJKoZIhvcNAQELBQADggEBAHnOmp8mICiahhf58HEcCjzWkjopGyaAERsWFScRBdg1k4Of3JOqi3QK47F83ai41jRNC+KBKYP
/mUSwEok3dERW7rVH3xKXo7GmCDUm7Hk1O7B8N+ITlSKniMksvSpkx3GEwedr1VD0Cgqj/MQa8wMP+xoXioBrdIIsTShk
/qi5ecrbdFXNhoeA3zO/vOn7WFPFC6xR+LKWnOEHW
/FtcOawcWV8hVNhG77CD+wyOnpypb1HKUOVSwFqDVvX7JF8u2809+m0ySqoZ621ITeTQNCw9km26bMKy7D4VefN3NIQbak8b0ftWzxWsngkvi
H5MFPSq5JWI0IZOjOhPiHntksxggMgMIIDHAIBATCBiTB9MSYwJAYDVQQDDB1OYW1pcmlhbCBDQSBGaXJtYSBRdWFsaWZpY2F0YTEgMB4GA1U
ECwwXQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkxJDAiBgNVBAoMG05hbWlyaWFsIFMucC5BLi8wMjA0NjU3MDQyNjELMAkGA1UEBhMCSVQCCFzw
qTJXlRldMA0GCWCGSAFlAwQCAQUAoIIBZzAYBgkqhkiG9w0BCQMxCwYJKoZIhvcNAQcBMBwGCSqGSIb3DQEJBTEPFw0yMjA4MTAwODAxNTZaM
C0GCSqGSIb3DQEJNDEgMB4wDQYJYIZIAWUDBAIBBQChDQYJKoZIhvcNAQELBQAwLwYJKoZIhvcNAQkEMSIEIIyS24577o6HAeqQ5OU2H1lF5J
MnJaRgd8ISguoanmBYMIHMBgsqhkiG9w0BCRACLzGBvDCBuTCBtjCBswQgxbkITFBriuSqr8M6xRqZCQN72rMbhDr7YtuXrw186f4wgY4wgYG
kfzB9MSYwJAYDVQQDDB1OYW1pcmlhbCBDQSBGaXJtYSBRdWFsaWZpY2F0YTEgMB4GA1UECwwXQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkxJDAi
BgNVBAoMG05hbWlyaWFsIFMucC5BLi8wMjA0NjU3MDQyNjELMAkGA1UEBhMCSVQCCFzwqTJXlRldMA0GCSqGSIb3DQEBCwUABIIBAIgAfXNg3
DxOPdOe5QTtfTMmV4zU/+cTGLT4xaB2x2/14o3NmkRjHCMq4NBDf8XifOo8YTrVQYqYNlxDaW5JpjTpqfbWun4wuIdkQqsg6TiRTiy3w
/v01oMk/X1sj7H+6wSffcRmIV5dwTIlHUx0MRXiPA00OaCsZTO852xwkXUB7z8/jaWfUK4bLoz
/ckgFmV3YhRLhth7sLWzVjgUELd6ukCiwCftP9R4KEwXEYu4YmBW9pknFDrDGZgTTYChsITLtJkNarzl/4JtxXcA7
/FALCxuyOHcnYNta+iCW4N3I/E0PIVzQ5XibNraAFO1ulJIzA1yC+hU4IjADJeAPoEE=</return>
      </ns2:signCAdESResponse>
   </soap:Body>
</soap:Envelope>
```

Below is the example of Xades Signature Level B:

signXadesList-Level-B.txt

Below, there is an example of Xades using the preferences:

- signElement
- signatureId

We sign the XML parts with "Id=tagToSign" specified on Soap request by:

<signElement>tagToSign</signElement>

And we set the id of the digital signature to:

<signatureId>idOfSignature</signatureId>

The full example:

signXadesList-on-specifiedTagId.xml

SignPkcs1

The SOAP request for create raw signature (PKCS1):

SOAP-request-signPkcs1

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://service.ws.nam
/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:signPkcs1>
         <credentials>
            <password>YOUR-DEVICE-PASSWORD</password>
            <username>YOUR-DEVICE-NAME</username>
         </credentials>
         <hash>c1fbd2b034abaea9ec99535394f21bb556edcf1833c680ebdfcc76e1e635844b</hash>
         <preferences>
            <hashAlgorithm>SHA256</hashAlgorithm>
         </preferences>
      </ser:signPkcs1>
   </soapenv:Body>
</soapenv:Envelope>
```

The SOAP response will be:

SOAP-response-signPkcs1

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:signPkcs1Response xmlns:ns2="http://service.ws.nam/">
      <return>p1kGoLorh0dGXKOFV4NAAZSsFxEl3YEbSXSI8tlWCSkoDwKLN9wHeKaosAuuJMuL6Vl61AxiEoBFGh5
/ufQLFKWaiMqvB5VYD62yXdp8f2dtMeCfqZGwLEV4ci/kd1iMvE2eYiGornXk9NoF+eg/+h6W8TZWSnYrjp0YlF1oJxOG1
/r5qr+OVvWQ7n73fo3Qe6Rjw
/TuSI5V+WDaboctuCIwlK5gM8R4cT552PrNLsnVYmwR4epSUTx5VYwag6IhEHYPUtUkbMGpvN+0C0cZY7NqOHPfqgrqks0HkMr4Z99DQAKOqS
ZHg+h4AIGvgqFGsLxSRWCbT2G7ve+qX7IVgg==</return>
      </ns2:signPkcs1Response>
   </soap:Body>
</soap:Envelope>
```

Manage signer device

In this section you can find the example of SOAP request associated to the information about signer device, timestamp, errors

Method change password on automatic/eseal signature

Below an example of change password on automatic signer device (AHIP22021318589386):

REQUEST-AUTOMATIC/ESEAL-changePassword

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://service.ws.nam
/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:changePassword>
         <credentials>
            <password>13572468</password>
            <securityCode>8214260012</securityCode>
            <username>AHIP12345</username>
         </credentials>
         <newPassword>NEWPASSWORD123</newPassword>
      </ser:changePassword>
   </soapenv:Body>
</soapenv:Envelope>
```

After this execution, the password/PIN of the device signature will be changed from "13572468" (old password) to "NEWPASSWORD123".

Method change password on remote signature

Below an example of change password on remote signer device (RHI3644468199007):

REQUEST-AUTOMATIC/ESEAL-changePassword

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://service.ws.nam
/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:changePassword>
         <credentials>
                        <username>YOUR-DEVICE-NAME</username>
            <password>YOUR-DEVICE-PASSWORD</password>
            <idOtp>YOUR-ID-OTP</idOtp>
            <otp>876321</otp>
         </credentials>
         <newPassword>NEWPASSWORD123</newPassword>
      </ser:changePassword>
   </soapenv:Body>
</soapenv:Envelope>
```

After this execution the password/PIN of the device signature will be changed from "847291742" (old password) to "NEWPASSWORD123".

Method getCertificate

Below the SOAP request example for obtain the certificate associate to signer device: "SHI7493852568871"

SOAP-request-getCertificate

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://service.ws.nam
/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:getCertificate>
         <credentials>
            <username>SHI12345</username>
         </credentials>
      </ser:getCertificate>
   </soapenv:Body>
</soapenv:Envelope>
```

The SOAP response will be:

SOAP-response-getCertificate

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:getCertificateResponse xmlns:ns2="http://service.ws.nam/">
         <return>MIIG1jCCBL6gAwIBAgIIKSO
/4EWMpWgwDQYJKoZIhvcNAQELBQAwgYwxJjAkBgNVBAMMHVRlc3QgTmFtaXJpYWwgRVUgUXVhbGlmaWVkIENBMR8wHQYDVQQLDBZUcnVzdCBT
ZXJ2aWNlIFByb3ZpZGVyMRgwFgYDVQQKDA9OYW1pcmlhbCBTLnAuQS4xGjAYBgNVBGEMEVZBVElULTAyMDQ2NTcwNDI2MQswCQYDVQQGEwJJV
DAeFw0xNzExMjQwOTAwMDBaFw0yMzExMjQwOTAwMDBaMG0xCzAJBgNVBAYTAklUMRowGAYDVQRhDBFWQVRJVC0wMjA0NjU3MDQyOTEYMBYGA1
UECgwPTmFtaXJpYWwgcy5wLmEuMRgwFgYDVQQDDA9uYW1pcmlhbCBzLnAuYS4xDjAMBgNVBC4TBUlEMTMzMIIBIjANBgkqhkiG9w0BAQEFAAO
CAQ8AMIIBCgKCAQEApGhU87UKgtDTBTy13rv9GKZk1+YUfuHPGJxFI3TxZrimmDtymOhu1l400Y1mMLe5DZVgFXkRFlw0+Gk0ZLJYkG4FvTo5
khqu+vsd0wyh/Hkpei0wLfnKhCWhYOpmlwahh3a31U1qFVlZJNu1ybRRf8N4+yAQQ2DlNL7
/B1VAgq1gVt1uXjxvas8MAUjjDbg3eQYXkSn2FJbveDRs127eeXUu+uabqt/GU/Y77Rvd7IKW6aH+dOF0oU/s7
/dto7q393rPU3OpWfvA3A1107C/jwFaSgIDdYhtGviT6Jbakk
/SM26QfkNQShrHsS9S9hCn3DZUfg53I4YGnOHtjFntKwIDAQABo4ICWDCCAlQwgZQGCCsGAQUFBwEBBIGHMIGEMEMGCCsGAQUFBzAChjdodHR
wczovL2RvY3MudGVzdC5uYW1pcmlhbHRzcC5jb20vZG9jdW1lbnRzL05hbUNBNEsuY3J0MD0GCCsGAQUFBzABhjFodHRwczovL29jc3AudGVz
dC5uYW1pcmlhbHRzcC5jb20vb2NzcC9jZXJ0c3RhdHVzMB0GA1UdDgQWBBRM2g4FW+kgJ7XcAbdsY2fK3wqG8TAfBgNVHSMEGDAWgBT8Hvd
/XQEv+XufWSmAjOaa1hw+njCBzgYIKwYBBQUHAQMEgcEwgb4wCAYGBACORgEBMAsGBgQAjkYBAwIBFDATBgYEAI5GAQYwCQYHBACORgEGAjCB
jwYGBACORgEFMIGEMEAWOmh0dHBzOi8vZG9jcy50ZXN0Lm5hbWlyaWFsdHNwLmNvbS9kb2N1bWVudHMvUERTL1BEU19lbi5wZGYTAmVuMEAWO
mh0dHBzOi8vZG9jcy50ZXN0Lm5hbWlyaWFsdHNwLmNvbS9kb2N1bWVudHMvUERTL1BEU19pdC5wZGYTAml0MF8GA1UdIARYMFYwPwYLKwYBBA
GCmmsBAgEwMDAuBggrBgEFBQcCARYiaHR0cHM6Ly9kb2NzLnRlc3QubmFtaXJpYWx0c3AuY29tLzAJBgcEAIvsQAEBMAgGBgQAj3oBATA5BgN
VHR8EMjAwMC6gLKAqhihodHRwOi8vY3JsLnRlc3QubmFtaXJpYWx0c3AuY29tL0NBNEsuY3JsMA4GA1UdDwEB
/wQEAwIGQDANBgkqhkiG9w0BAQsFAAOCAgEAE2+uvSjsQZwx2R+tH76IfrPcWfGuJYlFAh044gu7evJ6
//h7EQc0Y6wSzMHM91mfOpnuHD2BP9NftA+qBqqZnwpcLn3S+3WiM7L7wBG0LJE20Ji
/fw0JUzTojtDQ24h64kQUv+u9cygB4JtFWAZ74WbMjmeGl5WtBbo9zUx5Z59qsMl+BuXUW23u7lbzIyZLKAL6w5qSrBJhafBuKtwNIYfMojtW
K3kOSikhktoA1K/s74xp9ofUpM4EHtjXHkQXN754dUbXbh2zYtDC4qw7LvkUnx2y2Yh8tOsv+N0c5kRMHvr0IjMzuuY7
/EuO0ivR7ncUg5ABJA9TQ8RA7pNw9ID+t9MHrsbEMGLYRWAsylARVbXpDpgZcCZxas6HE+JJWGn+LEBFDJiEPdSuvYY/bLML3G5wan
/cTYic0TK/HGJxzqoAxUB1gks3eUmvsTxzOzyTmJxjJBBWSTU5ulRjK
/oexLEjYprXeiPjx0gB6J+2+LoXsBAj1KMgHLWZ9NBAqZ7EvvZi3SDcOA1ijtjDRRNbHvI1naV3e
/1WlYLEJaAUghNQBYzf2hyv8ikDv1Wb3YGML2ruu5BDAn7YpM0+sMFMmcjU/ImN64L1oLYJJ8d79UqJ0AS5bc+OzbncpTtd5sncvCKWh
/xg5Qnc2ZY9djDgk/HhDUqTVRkGLua8gN4=</return>
      </ns2:getCertificateResponse>
   </soap:Body>
</soap:Envelope>
```

In the tag "return" there is the base64 associated to the signer device.

Method getAvailableSignatures

Below the SOAP request example for obtain the certificate associate to signer device: "SHI7493852568871"

SOAP-request-getAvailableSignatures

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://service.ws.nam
/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:getAvailableSignatures>
         <credentials>
            <username>SHI12345</username>
         </credentials>
      </ser:getAvailableSignatures>
   </soapenv:Body>
</soapenv:Envelope>
```

The SOAP response will be:

SOAP-response-getAvailableSignatures

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:getAvailableSignaturesResponse xmlns:ns2="http://service.ws.nam/">
         <return>996413</return>
      </ns2:getAvailableSignaturesResponse>
   </soap:Body>
</soap:Envelope>
```

In the tag "return" there is the number of signatures available.

Method getSignatures

Below the SOAP request example for obtain the certificate associate to signer device: "SHI7493852568871"

SOAP-request-getSignatures

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://service.ws.nam
/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:getSignatures>
         <credentials>
            <username>SHI12345</username>
         </credentials>
      </ser:getSignatures>
   </soapenv:Body>
</soapenv:Envelope>
```

The SOAP response will be:

SOAP-response-getSignatures

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:getSignaturesResponse xmlns:ns2="http://service.ws.nam/">
         <return>3587</return>
      </ns2:getSignaturesResponse>
   </soap:Body>
</soap:Envelope>
```

In the tag "return" there is the number of signatures apposed since the device has been created.

Manage errors in SWS

In this section will be described how manage the errors in SWS and obtain the info about errors.

If the SOAP request is not correct in output will obtain the SOAP response with this structure:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <soap:Fault>
         <faultcode>soap:Server</faultcode>
         <faultstring>Codice OTP errato, riprovare con il prossimo codice</faultstring>
         <detail>
            <ns2:WSException xmlns:ns2="http://service.ws.nam/">
               <error>44</error>
               <message>Codice OTP errato, riprovare con il prossimo codice</message>
            </ns2:WSException>
         </detail>
      </soap:Fault>
   </soap:Body>
</soap:Envelope>
```

This SOAP response contains:

- error code = 44
- error message = "Codice OTP errato, riprovare con il prossimo codice"

By default SOAP response on SWS contains the error message in italian, but is possible to obtain the error message in other different languages using the method "getErrors".

Method getErrors

This method permits to obtain the list of all errors in a specified language or all languages.

For example if we want obtain the list of all errors in english language the SOAP request will be:

SOAP-request-getErrors

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://service.ws.nam
/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:getErrors>
         <lang>EN</lang>
      </ser:getErrors>
   </soapenv:Body>
</soapenv:Envelope>
```

In output will obtain a list of all errors in a specified language:

SOAP-response-getErrors

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:getErrorsResponse xmlns:ns2="http://service.ws.nam/">
         <return>
            <errorCode>0</errorCode>
            <errorLanguage>EN</errorLanguage>
            <errorLanguage2>ENG</errorLanguage2>
            <errorText>No errors found</errorText>
         </return>
         <return>
            <errorCode>1</errorCode>
            <errorLanguage>EN</errorLanguage>
            <errorLanguage2>ENG</errorLanguage2>
            <errorText>Generic error</errorText>
         </return>
.........................
.........................
.........................
         <return>
            <errorCode>1007</errorCode>
            <errorLanguage>EN</errorLanguage>
            <errorLanguage2>ENG</errorLanguage2>
            <errorText>The OTP device was not activated</errorText>
         </return>
         <return>
            <errorCode>1009</errorCode>
            <errorLanguage>EN</errorLanguage>
            <errorLanguage2>ENG</errorLanguage2>
            <errorText>Unavailable attempts for the OTP device</errorText>
         </return>
         <return>
            <errorCode>1016</errorCode>
            <errorLanguage>EN</errorLanguage>
            <errorLanguage2>ENG</errorLanguage2>
            <errorText>The OTP device was not associated to the holder</errorText>
         </return>
      </ns2:getErrorsResponse>
   </soap:Body>
</soap:Envelope>
```

With this method is possible to obtain the description associated to a specified error code (in this example 44). Below the example:

SOAP-request-getErrors-on-specific-errorCode

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://service.ws.nam
/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:getErrors>
         <lang>EN</lang>
         <errorCode>44</errorCode>
      </ser:getErrors>
   </soapenv:Body>
</soapenv:Envelope>
```

The SOAP response will be:

SOAP-request-getErrors-on-specific-errorCode

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:getErrorsResponse xmlns:ns2="http://service.ws.nam/">
         <return>
            <errorCode>44</errorCode>
            <errorLanguage>EN</errorLanguage>
            <errorLanguage2>ENG</errorLanguage2>
            <errorText>Wrong OTP code, try again with the next code</errorText>
         </return>
      </ns2:getErrorsResponse>
   </soap:Body>
</soap:Envelope>
```

Methods for timestamp

Below the SOAP request for apply timestamp and get the timestamps available

Apply timestamp

Below an example of SOAP request for apply timestamp. In output will have the timestamp in TSD format

SOAP-request-timestamp

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://service.ws.nam
/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:timestamp>
                  <content>BASE64-FILE-TO-APPLY-TIMESTAMP</content>
         <preferences>
            <outputAsTSD>true</outputAsTSD>
            <timestampHashAlgo>SHA-256</timestampHashAlgo>
            <timestampUrl>TSA-URL</timestampUrl>
            <timestampUsername>TSA-USERNAME</timestampUsername>
            <timestampPassword>TSA-PASSWORD</timestampPassword>
         </preferences>
      </ser:timestamp>
   </soapenv:Body>
</soapenv:Envelope>
```

In output will obtain the TSD.

NOTE:

The TSA-URL for PROD environment is:

tsa-url-prod

```
https://timestamp.namirialtsp.com
```

While the TSA-URL for TEST environment is:

tsa-url-test

```
https://timestamp.test.namirialtsp.com
```

Method getAvailableTimestamps

This method permits to obtain the timestamp available. Below an example:

SOAP-request-getAvailableTimestamps

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://service.ws.nam
/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:getAvailableTimestamps>
         <preferences>
            <timestampUrl>https://timestamp.test.namirialtsp.com/enquiry</timestampUrl>
            <timestampUsername>TSA-USERNAME</timestampUsername>
            <timestampPassword>TSA-PASSWORD</timestampPassword>
         </preferences>
      </ser:getAvailableTimestamps>
   </soapenv:Body>
</soapenv:Envelope>
```

The SOAP response will contain the number of timestamp available associate to TSA-USERNAME.

NOTE: if you are checking the PROD TSA account the timestampURL will be:

timestampUrl-PROD

```
https://timestamp.namirialtsp.com/enquiry
```

Methods for utilities

Below the utilitiest to extraxt info about file

Method getAllSignatureFieldsWithPreferences

This method permits to obtain the extract all info about signature fields of a PDF document available. Below an example:

SOAP-request-getAllSignatureFieldsWithPreferences

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://service.ws.nam
/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:getAllSignatureFieldsWithPreferences>
         <buffer>BASE64-FILE-TO-EXTRACT-INFO</buffer>
         <preferences>
            <encryptionPassword>string</encryptionPassword>
            <withCertificate>boolean</withCertificate>
            <withDetails>boolean</withDetails>
         </preferences>
      </ser:getAllSignatureFieldsWithPreferences>
   </soapenv:Body>
</soapenv:Envelope>
```

The response will be:

SOAP-response-getAllSignatureFieldsWithPreferences

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:getAllSignatureFieldsWithPreferencesResponse xmlns:ns2="http://service.ws.nam/">
         <return>
            <identifier>SignatureField-1</identifier>
            <signed>false</signed>
         </return>
         <return>
            <identifier>SignatureField-2</identifier>
            <signatureDetails>
               <appearance>
                  <height>50.0</height>
                  <width>200.0</width>
                  <x>100.0</x>
                  <y>100.0</y>
               </appearance>

<certificate>MIIGzDCCBbSgAwIBAgIIXPCpMleVGV0wDQYJKoZIhvcNAQELBQAwfTEmMCQGA1UEAwwdTmFtaXJpYWwgQ0EgRmlybWEgUXVh
bGlmaWNhdGExIDAeBgNVBAsMF0NlcnRpZmljYXRpb24gQXV0aG9yaXR5MSQwIgYDVQQKDBtOYW1pcmlhbCBTLnAuQS4vMDIwNDY1NzA0MjYxC
zAJBgNVBAYTAklUMB4XDTE4MDEyMzE2MzcwMFoXDTI0MDQyMzE2MzcwMFowgacxCzAJBgNVBAYTAklUMRUwEwYDVQQKDAxOT04gUFJFU0VOVE
UxFTATBgNVBAQMDERFTU8gQ09HTk9NRTESMBAGA1UEKgwJREVNTyBOT01FMRwwGgYDVQQFExNJVDpETUNETk0xNVQxMEEyNzFPMR8wHQYDVQQ
DDBZERU1PIE5PTUUgREVNTyBDT0dOT01FMRcwFQYDVQQuEw5ERU1PMTIzNDU2Nzg5MDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
APiLqTYMjlxdSMoOI9/HqoWXmQPP7AKlZEWedZwvXfu+LgqG6kalwESS+vb3zFmnzaMXCuhDYjg1SQC4YHHsrddlnA72Zv6pnY98h43M+MdO3
/RQ4wZ7bA/SgRljVxB2uJemDi0Fu/l09Yys5Ou384Dj9C4C2HlR6SbQ7s4L4Kx5x
/KGSW6mN8AGaNvgKxQhgXWprifhJqSUTxG2tvP0+NnElsr3TaxONpCeYr3li/lwQPpO0A2JW0wqz94uCgQU3y/HrYqHdX50ccb7jNd+ah3fO
/U6RUvQoBcgUFKG973eiFyv8SGymRHf9iwdanrLkTAKSFsYk8z9JfAOGN1WoBsCAwEAAaOCAyMwggMfMIGiBggrBgEFBQcBAQSBlTCBkjBUBg
grBgEFBQcwAoZIaHR0cHM6Ly9kb2NzLnRlc3QuZmlybWFjZXJ0YS5pdC9kb2N1bWVudHMvTmFtaXJpYWxDQUZpcm1hUXVhbGlmaWNhdGEuY3J
0MDoGCCsGAQUFBzABhi5odHRwOi8vb2NzcC50ZXN0LmZpcm1hY2VydGEuaXQvb2NzcC9jZXJ0c3RhdHVzMB0GA1UdDgQWBBS+qAt2HpxfpSyL
ftgy962gNRD4BzAfBgNVHSMEGDAWgBTbOAhQt7Yvlu477D60ZlTCSpap3jAvBggrBgEFBQcBAwQjMCEwCAYGBACORgEBMAsGBgQAjkYBAwIBF
DAIBgYEAI5GAQQwggGqBgNVHSAEggGhMIIBnTCCAZkGCysGAQQBgpprAQEDMIIBiDAwBggrBgEFBQcCARYkaHR0cDovL3d3dy5maXJtYWNlcn
RhLml0L21hbnVhbGktTU8vMIIBUgYIKwYBBQUHAgIwggFEHoIBQABJAGwAIABwAHIAZQBzAGUAbgB0AGUAIABjAGUAcgB0AGkAZgBpAGMAYQB
0AG8AIADoACAAdgBhAGwAaQBkAG8AIABzAG8AbABvACAAcABlAHIAIABmAGkAcgBtAGUAIABhAHAAcABvAHMAdABlACAAYwBvAG4AIABwAHIA
bwBjAGUAZAB1AHIAYQAgAGEAdQB0AG8AbQBhAHQAaQBjAGEALgAgAFQAaABpAHMAIABjAGUAcgB0AGkAZgBpAGMAYQB0AGUAIABtAGEAeQAgA
G8AbgBsAHkAIABiAGUAIAB1AHMAZQBkACAAZgBvAHIAIAB1AG4AYQB0AHQAZQBuAGQAZQBkAC8AYQB1AHQAbwBtAGEAdABlAGQAIABkAGkAZw
BpAHQAYQBsACAAcwBpAGcAbgBhAHQAdQByAGUAcwAuMEkGA1UdHwRCMEAwPqA8oDqGOGh0dHA6Ly9jcmwudGVzdC5maXJtYWNlcnRhLml0L0Z
pcm1hQ2VydGFRdWFsaWZpY2F0YTEuY3JsMA4GA1UdDwEB/wQEAwIGQDANBgkqhkiG9w0BAQsFAAOCAQEAec6anyYgKJqGF
/nwcRwKPNaSOikbJoARGxYVJxEF2DWTg5/ck6qLdArjsXzdqLjWNE0L4oEpg
/+ZRLASiTd0RFbutUffEpejsaYINSbseTU7sHw34hOVIqeIySy9KmTHcYTB52vVUPQKCqP8xBrzAw
/7GheKgGt0gixNKGT+qLl5ytt0Vc2Gh4DfM7+86ftYU8ULrFH4spac4Qdb8W1w5rBxZXyFU2EbvsIP7DI6enKlvUcpQ5VLAWoNW9fskXy7bzT
36bTJKqhnrbUhN5NA0LD2SbbpswrLsPhV583c0hBtqTxvR+1bPFayeCS+IfkwU9KrklYjQhk6M6E+Iee2Sw==</certificate>
               <location>Casa</location>
               <name>DEMO NOME DEMO COGNOME</name>
               <page>-1</page>
               <reason>prova nuovo metodo</reason>
               <signDate>2023-10-19T15:24:37+02:00</signDate>
               <subjectDN>DNQ=DEMO1234567890, CN=DEMO NOME DEMO COGNOME, SERIALNUMBER=IT:DMCDNM15T10A271O,
GIVENNAME=DEMO NOME, SURNAME=DEMO COGNOME, O=NON PRESENTE, C=IT</subjectDN>
            </signatureDetails>
            <signed>true</signed>
         </return>
         <return>
            <identifier>SignatureField-3</identifier>
            <signed>false</signed>
         </return>
      </ns2:getAllSignatureFieldsWithPreferencesResponse>
   </soap:Body>
</soap:Envelope>
```

Method getAvailableSignatureFields

This method permits to obtain the extract all info about signature fields of a PDF document available. Below an example:

SOAP-request-getAvailableSignatureFields

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://service.ws.nam
/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:getAvailableSignatureFields>
         <buffer>BASE64-FILE-TO-EXTRACT-INFO</buffer>
         <encryptionPassword>string</encryptionPassword>
      </ser:getAvailableSignatureFields>
   </soapenv:Body>
</soapenv:Envelope>
```

The response will be:

SOAP-response-getAvailableSignatureFields

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:getAvailableSignatureFieldsResponse xmlns:ns2="http://service.ws.nam/">
         <return>SignatureField-1</return>
         <return>SignatureField-3</return>
      </ns2:getAvailableSignatureFieldsResponse>
   </soap:Body>
</soap:Envelope>
```

For example, you can test this request using this pdf

Examples (source code)

Below will find the links contains the source code with examples.

Java:

To add on CMS repo

Php:

C#: https://cms.firmacerta.it/download/sws_cnet.zip

C# (for SaaS instance): https://cms.firmacerta.it/download/SignEngineWebClientSaaS.zip