

Sign interface

- [Introduction](#)
 - [Credentials Object](#)
 - [Automatic Signature](#)
 - [Remote Signature](#)
 - [How obtain the idOtp and OTP code](#)
 - [Obtain the idOtp](#)
 - [Obtain the OTP code](#)
 - [Manage the sessionKey](#)
 - [Obtain the sessionKey](#)
 - [Check the sessionKey status](#)
 - [Destroy the sessionKey](#)
 - [Summarize](#)
- [Methods for sign](#)
 - [SignPades](#)
 - [SignPadesMultiFieldName](#)
 - [SignCades](#)
 - [SignCades Detached](#)
 - [SignXades](#)
 - [SignPkcs1](#)
- [Manage signer device](#)
 - [Method change password on automatic/eseal signature](#)
 - [Method change password on remote signature](#)
 - [Method getCertificate](#)
 - [Method getAvailableSignatures](#)
 - [Method getSignatures](#)
- [Manage errors in SWS](#)
 - [Method getErrors](#)
- [Methods for timestamp](#)
 - [Apply timestamp](#)
 - [Method getAvailableTimestamps](#)
- [Methods for utilities](#)
 - [Method getAllSignatureFieldsWithPreferences](#)
 - [Method getAvailableSignatureFields](#)
- [Examples \(source code\)](#)

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 object 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>
```

Fields 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>sadlijhdfkjslherpoufdblkhsljherihbfdoihejheroihger</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>
        <id0tp>501719</id0tp>
        <serialNumber>20210113-091031RJ2L1</serialNumber>
        <type>SMS</type>
      </return>
      <return>
        <id0tp>537430</id0tp>
        <serialNumber>20210305-163726L0PYF</serialNumber>
        <type>OTP GENERATOR</type>
      </return>
      <return>
        <id0tp>537433</id0tp>
        <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/cCxW6mTgL3iGjFEST5cEAZjgLnXvV3hUFzFHcTvjlH3F0kJy+kv/0Zsv1
        uNK0S7L6jMqHYSSpBz+CZl7h3r5IEP2FqrK7WJQTVyrNfyr/trZmDgxY0LuACyoZVUFIlncK5Lkjihui
        sv+gZeB68Spwm+cNDdQQdUS3ngzJavHXxo9ADCX6VDIKKMe
        /AY0v+R51XWE90JF5LfKETHlv10CpQC5nhnw8wK0F0m
        P4vM90d79JhFYGVVSZWtnTQ9Dg8p0Mvg9wwxNm3uGkKKaS7oTp1ewd+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>
          f41f7bq/cCxw6mTgL3iGjFEST5cEAZjgLnxvV3hUFzFHcTvjlH3F0kJy+kv
          uNK0S7L6jMqHYSSpBz+CZl7h3r5IEP2FqrK7WJQTVyrNfyr
/0Zsv1
/trZmDgxY0LuACyoZVUFILnck5Lkjihui
sv+gZeB68Spwm+cNDdQQdUS3ngzJavHXxo9ADCX6VDIKKMe
/AY0v+R51XWE90JF5LFKETHlv10CpQC5nhnW8WK0F0m
P4vM90d79JhFYGVVSZWtnTQ9Dg8p0Mvg9wwxNm3uGkKkAs7oTp1ewd+eCG
/uSC9k3H2w9GB6vQLHQEbn6d
          </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, c lose 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>f41f7bq/cCxW6mTgL3iGjFEST5cEAZjgLnXvV3hUFzFhcTvjlH3F0kJy+kv
/0Zsv1uNK0S7L6jMqHYSSpBz+CZl7h3r5IEP2FqrK7WJQTVyrNfyr
/trZmDgxYOLuACyoZVUFIlncK5Lkjihuisv+gZeB68Spwm+cNDdQQdUS3ngzJavHXxo9ADCX6VDIKKMe
/AY0v+R51XWE90JF5LfKETHlv10CpQC5nhnW8WK0F0m P4vM90d79JhFYGVVSZWtnTQ9Dg8p0Mvg9wwxNm3uGkKKaS7oTp1ewd+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>
  <id0tp>501719</id0tp>
  <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>f41f7bq/cCxW6mTgL3iGjFEST5cEAZjgLnXvV3hUFzFHCtvjLH3F0kJy+kv
/0Zsv1uNK0S7L6jMqHYSSpBz+CZL7h3r5IEP2FqrK7WJQTVyrNfyr
/trZmDgxY0LuACyoZVUFILnck5Lkjihuisv+gZeB68Spwm+cNdDQQdUS3ngzJavHXxo9ADCX6VDIKKMe
/AY0v+R51XWE90JF5LFKETHlv10CpQC5nhnW8WK0F0m P4vM90d79JhFYGVVSWtnTQ9Dg8p0Mvg9wwxNm3uGkKKaS7oTp1ewd+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-T0-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-T0-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 `WSEException` 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 WSEException 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:WSEException xmlns:ns2="http://service.ws.nam/">
          <error>105</error>
          <message>Richiesta non valida</message>
        </ns2:WSEException>
      </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>MIIKVwYJKoZIhvcNAQcCoIIKSDCCCkQCAQExDzANBglgHkgBZQMEAgEFADA2BgkqhkiG9w0BBWGGKQQnVGhpcyBpcyB0aGUgZmlsZSB0byBiZSBzawduZWQgZm9yIHRlc3QuoIIg0DCCBswggW0oAMCAQICCFzwqTJXlRldMA0GCSqGSib3DQEBcWUAMH0xJjAKBgNVBAMMHU5hbWlyawFsIENBIEZpcm1hIFFfY1WxpZmljYXRhMSAwHgYDVQQQLDBdDZXJ0awZpY2F0aw9uIEF1dGhvcml0eTEkMCIIGA1UECgwbTmFtaXJpYWwgUy5wLkEuLzAyMDQ2NTcwNDI2MQswCQYDVQGEwJJVDAeFw0xODAxMjMxNjM3MDBaFw0yNDA0MjMxNjM3MDBaMIGnMQswCQYDVQGEwJJVDEVMBMGAlUECgWMTk90IFBSRVNFTlRFMRUwEwYDVQQEDAXERU1PIENPR05PTUUXEjAQBGNVBCoMCURFTU8gTk9NRTEcMBoGA1UEBRMTSVQ6RE1DRE5NM TVUMTBBMjcXZEFmB0GA1UEAwWwREVNTYBOT01FIERFTU8gQ09HTK9NRTEXMBUGA1UELhMOREVNTzEyMzQ1Njc40TAwggeiMA0GCSqGSib3DQEBQUAA4IBDwAwggEKaoIBAQD4i6k2DI5cXUjKDIPfx6qFL5kDz+wCpWRfnnWcL137vi4KhupGpcBEkvr298xZp82jFwroQ2I4NUkAuGBx7K3XZZw09mb+qZ2PfiEnZpJHTt/0UOMGe2wP0oEZY1cQdriXpg4tBbv5dPwMr0Trt/OA4/QuAth5Uekm0070C+Csecfyhklupjfabmjb4CsUIYf1qa4n4SaklE8Rtrbz9PjZxJbK902sTjaQnmK95Yv5CED6TtAniVtMKS/eLgoEFN8vx62Kh3V+4zXFmod3zv10KVL0KAXIFBShve93ohcr/EhspkR3/YsHwp6y5EwCkhhGJPM/SXWdhjdVqAbgMBAAGjggMjMIIDHxCBogYIKwYBBQUHAQEEGZUwgiWVAYIKwYBBQUHMAKGSgh0dHBz0i8vZG9jcy50ZXN0LmZpcm1hY2VydgEuaxQvZG9jdW1lbnRzL05hbWlyawFsQ0FGaXJtYVYfY1WxpZmljYXRhLmNydDA6BggrBgEFBQcwAYYuaHR0cDovL29jc3AudGVzdC5maXJtYWNlcnRhlml0L29jc3AvY2VydhN0YXR1czAdBgNVHQ4EFgQUvqgLdh6cX6Usi37Ymvet0DUQ+AcwHwYDVR0jBBgwFoAU22gIULe2L5bu0+w+tGZUwqWqd4wLwYIKwYBBQUHAQMEIzAhmAgGBgQAJkYBATALBgYEAi5GAQMCAQwCAYGBACORgEEMiIBggYDVR0gBIIBoTCCA0wggGZBgsrBgEEAYKaaWBAZCCAYgwMAYIKwYBBQUHAGewJGh0dHA6Ly93d3cuZmlybWVjZXJ0YS5pdC9tYw51YwXpLU1PLzCCAIVGCCsGAQUFBwICMIIIBRB6CAUAASQBSACAACABYAGUAcwBLAG4AdABLACAAYwB LAHIAdABpAGYAaQBjAGEAdABvACAA6AAgAHYAQBSAGKAZABvACAacwBvAGwAbwAgAAHAZQByACAAZgBpAHIAbQB LACAAYQBwAAHAbwBzAHQAZQAgAGMabwBuACAACABYAG8AYwBLAGQAdQByAGEAIAbAHUAdABvAG0AYQB0AGKAYwBhAC4AIABUAGgAaQBzACAAYwB LAHIAdABpAGYAaQBjAGEAdABLACAAbQBhAHKAIABvAG4AbAB5ACAAYgB LACAAdQBzAGUAZAAGAGYAbwByACAAdQBuAGEAdAB0AGUAbgBkAGUAZAavAGEAdQB0AG8AbQBhAHQAZQBkACAABZABpAGCAaQB0AGEAbAAGAHMAaQBnAG4AYQB0AHUAcgB LAHMALjBJBgNVHR8EQjBAMD6gPKA6hjhodHRwOi8vY3JsLnRlc3QuZmlybWVjZXJ0YS5pdC9GaXJtYUNlcnRhUXVhbGlmawNhdGExLmNydA0BgNVHQ8BAf8EBAMCBkAwDQYJKoZIhvcNAQELBQADggEBAHnOmp8mICiahhf58HEcJzWkjopGyaAERSWFSrCBdg1k40f3J0qi3QK47F83ai41jRNC+KBKYP/mUSwEok3DERw7rVh3xKXo7GmCDum7Hk107B8N+ITlSKniMksvSpkx3GEwedr1VD0Cgqj/MQa8wMP+xoXioBrdIISTshk/qi5ecrbdFXNhoeA3z0/v0n7WFPFC6xR+LKwn0EHW/Ftc0awcwV8hVnhG77CD+wyOnpyb1HKU0VSwFqDvVx7JF8u2809+m0ySqoZ621ITeTQNCw9km26bMKy7D4VefN3NIQbak8b0ftWzXwsngkviH5MFPSq5JWI0IZ0j0hPiHntksxggMgMIIDHAIBATCBiT9MSYwJAYDVQQDDb10Yw1pcmlhbCBDQSBGaXJtYSBRdWFSawZpY2F0YTEgMB4GA1UECwwXQ2VydGlmawNhdGlvb1BBDXRob3JpdHkxJDAiBgNVBAoMG05hbWlyawFsIFMucC5BLi8wMjA0NjU3MDQyNjElMAKGA1UEBhMCSVQCCFzwqTJXlRldMA0GCGWCGSAFLawQCAQUAoIIBZZAYBgkqhkiG9w0BCQMxCwYJKoZIhvcNAQcCBMBwGCSqGSib3DQEJBTEPFw0yMjA4MTAwODAxNTZaM0GCSqGSib3DQEJNDEgMB4wDQYJYIZIAWUDBAIBBQChDQYJKoZIhvcNAQELBQAwLwYJKoZIhvcNAQkEMSIEIiYs24577o6HAeqQ50U2H1f5JMNjARgd8ISguoanmBYMIHMBgsqhkiG9w0BCRACLZGBVDCBTCBTjCBswQgxbkITfBrIuSqr8M6xRqZCQN72rMbhDr7YtuXrw186f4wgY4wgYGkfzB9MSYwJAYDVQQDDb10Yw1pcmlhbCBDQSBGaXJtYSBRdWFSawZpY2F0YTEgMB4GA1UECwwXQ2VydGlmawNhdGlvb1BBDXRob3JpdHkxJDAiBgNVBAoMG05hbWlyawFsIFMucC5BLi8wMjA0NjU3MDQyNjElMAKGA1UEBhMCSVQCCFzwqTJXlRldMA0GCSqGSib3DQEBcWUABIIABIAIgAfXNg3Dx0Pd0e5QTtftMmV4ZU/+cTGLT4xaB2x2/1403NmkRjHcmq4NBDF8X1f0o8YTrVQYqYNLxDaW5JpJTpqfbWun4wuIdkQqsg6TiRTiy3w/v01oMk/X1sJ7H+6wSffcRmIV5dwTILHUx0MRxiPA000aCsZT0852xwkXUB7z8/jawfUK4bLoz/ckgFmY3YhRLhth7sLwZVjgUeLD6ukCiwCftP9R4KEwXEYU4YmBW9pknFDrDGZgTTTYChsITLtkJkNarZl/4JtxXcA7/FALCxuy0HcnYnta+icW4N3I/E0PIVzQ5XibNraAF01ulJIzA1yC+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>MIiKKQYJKoZiHvcNAQcCoIiKgJCCChYCAQEXdZANBglgkhgBZQMEAgEFADALBgkqhkiG9w0BBwGggga2MIIGsjCCBJqgAwIBAGIIf
l5W0
/FI108wDQYJKoZiHvcNAQELBQAuGwYcXITAFBgNVBAMGE5hbWlyawFsIEVvIFF1YXpZml1ZCBBDQTEfMB0GA1UECwwVHj1c3QgU2VydmljZS
BQcm92aWRlcjEYMBYGA1UECgWPTmFtaXJpYWwgUy5wLkEuMR0wGAYDVRhDBFwQVRJVC0wMjA0NjU3MDQyNjEjLMAKGA1UEBhMCS0wHhCNMTk
w0DA1MDC0NjA1AwHhcnMjUw0DA1MDC0NjA1AwJwBmMQ0wCwYDVQQUeWRJRDRkYMRUwEwYDVQDDAx0XN0IGF6aWVuzGEXFTATBgNVBA0MDHRlC3Qg
YXppZ2W5KYTEaMBGGA1UEY9RVKfUSVQQTMDAOWpADMDAwMDAxCAZABGNVBAYTAKLUMiIBIjANBgkqhkiG9w0BAQEFAAOCQA8AMIIBCgKCAQEA4
TXG/VhgZhtRf8HkRmLa2NDEFg5CnLoPR5Yowpd66oqQTovMqF
/nbvY+wTOXiXwHYbAL1mTeEqwNTWBwKrGDFe3JP3EJSANRa8BMUJi5B2Jufud4fSZSaeeJkoCu2ggZ3/64kFdC2yxPlpwn
/oKSNJGiPyvhNW2JJV9NtlsUDDV0mHug5uFTQoo423CvmcRSz6kYhrWBKyQ9TG6PwukDqpKHiB+
/01pL7DVI4X4gNi1gnFQoBbBRG2rSAB8xLXxYP
/aWiMF83VZ5xELIGumQEfXoQRJ07kr1Qlh1MXFwwadRa8lZnQpg5vMAjthXnr0GyPjCim5X0rbCRwGlhVsQIDAQABo4ICQDCCAjwwgYcGC
CsGAQFUBwEBBHSweTA+BggprBgEFBQwAoYyaHR0CHM6Ly9kb2N2Lm5hbWlyawFsdlHNwLmNvbS9yY3NwL2N1cnRzdGF0dXMwHQYDVROBBEYFBT3hX0qb5n0V7hKLCuE5QNZCI1
1MB8GA1UdIwQYMBAAFG04zbhJUuXnCXtXjPt6QQ5BqnhZMIHNBggrBgEFBQwBAwSBwDCBvTAIBgYEA15GAQEwCwYGBACORgEDAgEUMAgBgQA
jYkYBBDATBgYEA15GAQYwCYHABACORgEGAjCbhAYGBACORgEFMHow0xY1aHR0cHM6Ly9kb2N2Lm5hbWlyawFsdlHNwLmNvbS9kb2N1bWVudHMvU
ERTL1BEU91b1b5wZGYTAmVuMdsWNWh0dHBz0i8vZG9jcy5uYw1pcmlhbHRzcC5jb20vZG9jdW1lbnRzL1BEUy9QRFNfaXQucGRmEwJpdDBaBg
NVHSAEUzBRMD0GcysGAQQBgpprAQIDMCswKQYIKwYBBQUHAgEWHw0dHBz0i8vZG9jcy5uYw1pcmlhbHRzcC5jb20vMAKGBWQAI+xAQMwCAY
GBACPEgECMDQGA1UdHwQTMCSwKaAnoCWI2h0dHA6Ly9jcmwubmFtaXJpYXw0c3AuY29tL0NBNEsuY3J5MA4GA1UdDwEB
/wQEAWiGQDANBgkqhkiG9w0BAQsFAAOCAgEAhlPP1QdsAqYEemj8MbHbF8a
/rAG4op1CbP4YsPr4Y5YKxr2TRuf4L4KcbFH8KpLPIXzw+2CvzeB2IGc00Ahs10s0d1hQimLKL6SEKS3UN3sxq6f
/i2dMS9IuCLcwyn8ZpCmvp2bqqj
/fCQZdoi0IF1WIOuqRufLSc95M+FMrWxzDovQs9Y8IerHiwbmq3fi3VuYdvKhx0Lr1l7Vcpmvk5zfXxsE9FmmH
/aJvdcrjPAJfT8Sjzz1vsB74MmqTfFk5Pnd1NlOHGw8KaTeLpu0lmsxz2qPF07inoSBuKMEFz3W3IvB3UnKuRbFEUTjyFYgYTPPTQcsX0kaJM
u/S76hRLTpvZ5zs+3AFgfcGbpH7kTCW1MTNT40yLwEXYctF4Mqg8giMaBetf06zB954Qqu3eqFvLDZ0HC1RbL
/F2at61rmnSXSbzRkev+VevKUGjIqThMq5lQSP8mxCquUvaa4epJ
/tDzmLyDYwnoN7fzoFA4ZkAQPLvt4Kv4Iml612CM0Kk+1mkEuhcTJN0h8l6Kax40q+Ld8qjgmVGAIME
/28dfIqP57gm70N1nCckDKMngcQ881LWYnbyY
/ba5BBcXeqwKp1+oqorrqFYiWONSH9SmmFI0gG6leEjL5k8nArKktNefScw5tv1LZXAC4uaZjD++iJ8jVMPn7khf9qKR8xggM3MIIDmWIBAT
CBLDCBzhEhMB8GA1UEAwYTMFtaXJpYWwgRVUgUXVhbGlmawVklENBMR8WHQYDVQQLDBZUcnVzdCBTZXJ2aWNlIFByb3ZpZGVyMRgwFgYDVQQ
KDA90Yw1pcmlhbCBTlNAuQs4xGjAYBgNVBGEtZBVElULTAyMDQ2NTcwNDI2MQswCQYDVQQUeWJJAiIfl5W0
/FI108wDQYJYIZIAUwDBAIBBQCgggFZMBG6CSsQsIb3DQeJAzelBgkqhkiG9w0BBwEwHAYJKoZiHvcNAQKFQM8XDTIzMDMzMMDA3NTA0MvowLQ
YJKoZiHvcNAQK0MSAwHjANBglgkhgBZQMEAgEFAKENBgkqhkiG9w0BAQsFADAvBgkqhkiG9w0BCQXxIggQmsj3f4hJCSELbMkwjKfWnrF0Xhk
eToNAkAkhx4686DYwdgGcYgGSib3DQeJEAIVMYHIMFIHMIHCIG/BCADIK96sqnXs0l13GxGXQqx466RyJv45gg1ab
/b9RM4PjCBmjCBjASBijCBzhEhMB8GA1UEAwYTMFtaXJpYWwgRVUgUXVhbGlmawVklENBMR8WHQYDVQQLDBZUcnVzdCBTZXJ2aWNlIFByb3Zp
pZGVyMRgwFgYDVQKDA90Yw1pcmlhbCBTlNAuQs4xGjAYBgNVBGEtZBVElULTAyMDQ2NTcwNDI2MQswCQYDVQQUeWJJAiIfl5W0
/FI108wDQYJKoZiHvcNAQELBQAEGgEAqEsv3NnVlrlLivnjrNyI85mjqZbaS3bt6FcfzREP+hmbSQ7DhXXqsRu2Vj+IkYod
/MhovvTwb2s6Y2uczugkVN3Jc
/uswUf6l1LCqn8F6bsDtL5xatZy75hu6mm8SJdNGyrTto7Zwf19vm3yTgt5Z3M+0RM4aHqsBYHVZwlqTyXR58uz8udSMzn2Cfrk+JCbMGiv
TCTMhujcFLySocN/ZWB1K0EGK
/m70ok9qZ40w9VQJ0BbwEVLU4A2FhNsnJtnqky6jPEhnAZ9ssazJ4fhT8o4fUbs71AunBz8A02BV5AcgK9pXvlrcx8p0WEyZaxo26RFs9Aoc
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 codes 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:

e-Signature

Sign a document

Sign a digest

Sign a PDF

Sign with JAdES

Sign multiple documents

Counter sign a signature

Standalone application

REST/SOAP WebServices

Server side

Extend a signature

Timestamp document(s)

Validate a signature

Validate a certificate

SSL-certificate validation

Replay Diagnostic Data

Merge containers

Export results

Validation results

Simple Report

Detailed Report

Diagnostic tree

ETSI Validation Report

Validation Policy: QES AdESQC TL based

Print

Download as PDF

Validate electronic signatures and indicates whether they are Advanced electronic Signatures (AdES), AdES supported by a Qualified Certificate (AdES/QC) or a Qualified electronic Signature (QES). All certificates and their related chains supporting the signatures are validated against the EU Member State Trusted Lists and third country Trusted Lists with voluntary-based AdES recognition (this includes signer's certificate and certificates used to validate certificate validity status services - CRLs, OCSP, and time-stamps).

Signature SIGNATURE_test-azienda_20230330-0750

Qualification:

Signature format:

Indication:

Certificate Chain:

On claimed time:

Best signature time:

Signature position:

Signature scope:

QESal

CADES-BASELINE-B

100%

Pass

test-azienda

Hammerill EU Qualified CA

2023-03-30 07:50:41 (UTC)

2023-03-30 07:58:17 (UTC)

1 out of 1

PDF_Sample_HelloWorld.pdf (FULL)

Full document

Document Information

Signatures status:

Document name:

1 valid signatures, out of 1

Cades_detached_PDF_SampleHelloWorld.p7s

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.name/">
 <soapenv:Header/>
 <soapenv:Body>
 <ser:signXAdES>
 <credentials>
 <username>YOUR-DEVICE-NAME</username>
 <password>YOUR-DEVICE-PASSWORD</password>
 </credentials>
 <buffer>PD94bWwgdmlVyc2lrbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4NCjxUZWxlbWFO
aWNvPg0KCTx0bz5Ub3ZlPC90bz4NCgk8ZnJvbT5KYW5pPC9mcm9tPg0KCTxoZWFK
aW5nPlJlbWluZGVyPC9oZWFKaW5nPg0KCTxib2R5PkRvbid0IGZvcmdldCBtZSB0
aGlzIHdlZWt1bmQhPC9ib2R5Pg0KPC9UZWxlbWFOaWNvPg==</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>MIIKvWYJKoZIhvcNAQcCoIIKSDCCCkQCAQEXdZANBglgHkgBZQMEAGFADA2BgkqhkiG9w0BBwGgKQqnVGhpcyBpcyB0aGUgZmlsZSB0byBiZSBzawduZWQgZm9yIHRlc3QuoIIg0DCCBSwwggw0oAMCAQICCFzwqTJXlRldMA0GCSqGSib3DQEBCwUAMH0xJjAkBgNVBAMMHU5hbWlyawFsIENBIEZpcm1hIFFf1YwXpZmljYXRhMSAwHgYDVQQLDBdDZXJ0awZpY2F0aw9uIEF1dGhvcm10eTEKMCIGA1UECgwBTmFtaXJpYWwgUy5wLkEuLzAyMDQ2NTcwNDI2MQswCQYDVQQGEWJJVDAeFw0xODAxMjMxNjM3MDBaFw0yNDA0MjMxNjM3MDBaMIGnMQswCQYDVQQGEWJJVDEVMBMGAG1UECgwMTk90IFBSRVNFTlRFMRUwEwYDVQEDAXERU1PIENPR05PTUUXEjAQBgNVBComCURFTU8gTk9NRTEcMBoGA1UEBRMTSVQ6RE1DRE5NM TVUMTBBMjxcTzEfMB0GA1UEAwwWREVENTYB0T01FIERFTU8gQ09HTk9NRTEXMBUGA1UELhMOREVNTzEyMzQ1Njc40TAwggEiMA0GCSqGSib3DQEBAQUAA4IBDwAwggEKAoIBAQD4i6k2DI5cXUjKDIPfx6qF15kdZ+wCpWRfnnWcL137vi4KhupGpcBEkvr298xZp82jFwroQ2I4NUKAuGBx7K3XZZw09mb+qZ2PfIeNzPjHTt/0UOMGe2wP0oEZY1cQdriXpg4tBbv5dPWMr0Trt/0A4
/QuAth5Uekm0070C+CsecfyhklupjFABmbj4CsUIYF1qa4n4SakLE8RtrbZp9PjZxJbK902sTjaQnmK95Yv5cED6TtAniVtMKs
/eLgoEFN8vx62Kh3V+dHHG+4Zxfmod3zv10kVL0KAXIFBShve93ohcr/EhspkR3/YsHWP6y5EwCkHbGJPM
/SXwDhjdvQAbAgMBAAGjggMjMII DHZCBogYIKwYBBQUHAQEEGZUwgZiWVAYIKwYBBQUHMAKGSgh0dHBz0i8vZG9jcy50ZXN0LmZpcm1hY2VydGEuaxQvZG9jdW1lbnRzL05hbWlyawFsQ0FGaXJtYVY1YwXpZmljYXRhLmNyNDAA6BgrBgEFBQcwAYYuaHR0cDovL29jc3AudGVzdC5maxJtYwNlcnRhLm10L29jc3AvY2VydHNOYXR1czAdBgNVHQ4EFgQUvqgLDh6cX6Usi37YmVetoDUQ+AcwHwYDVR0jBBgwFoAU2ZgIULe2L5bu0+w+tGZUwkqWqd4wLwYIKwYBBQUHAQMEIzAhMAgGBGQAjkyBATALBgYEAISGAQMCAQwCAYGBACORgEEMIIBqgYDVR0gBIIBoTCCA0wggGZBgsrBgEEAYKaawEBAZCCAYgWMAyIKwYBBQUHAGewJGH0dHA6Ly93d3cuZmlybWljZXJ0YS5pdC9tYW51YwXpLU1PLZCCAIVGCCsGAUUFBwICMII BRB6CAUAAASQBsACAAcABYAGUAacwBLAG4AdABLACAAyWbLAIHAdABpAGYAaQBjAGEAdABLACAAbQBhAHKAIABvAG4AbAB5ACAAYgBLACAAAdQBzAGUAZAAGAGYAbwByACAAdQBuAGEAdAB0AGUAbgBkAGUAZAavAGEAdQB0AG8AbQBhAHQAZQBkACAABZABpAGcAaQB0AGEAbAAGAHMAaQBNAG4AYQB0AHUAcgBLAHMALjBjBgNVHR8EQjBAMd6gPKA6hjhodHRw0i8vY3JsLnRlc3QuZmlybWljZXJ0YS5pdC9GaXJtYUNlcnRhUXVhbGlmawNhdGEXLmNybdA0BgnVHQ8BAf8EBAACBkAwDQYJKoZIhvcNAQELBQADggEBAHn0mp8mICiahhf58HEcCjzWkjopGyaAERSwFScRBdg1k40f3J0qi3QK47F83ai41jRNC+KBKYP
/mUSwEok3dERw7rVH3KX07GmCDUm7Hk107B8N+ITlSKniMksvSpkx3GEwedr1VD0Cgqj/MQa8wMP+oxXioBrdIIsTShk
/qi5ecrdbFXNhoeA3z0/v0n7WFPFC6xR+LKWn0EHw
/Ftc0awcwV8hVNHg77CD+wy0Npyyb1HKU0VSsFqDVvX7JF8u2809+m0ySsqoZ621ITeTQNCw9km26bMKy7D4VefN3NIQbak8b0ftWzxwsngkviH5MFPSq5JWI0IZ0j0hPiHntksxggGMIIIDHAIBATCBiTB9MSYwJAYDVQDDb10Yw1pcmlhbCBDQSBGaXJtYSBRdWFSawZpY2F0YTEgMB4GA1UECwwXQ2VydGlmawNhdGlvbiBBdXRob3JpdHkxJDAiBgNVBAoMG05hbWlyawFsIFMucC5BLi8wMjA0NjU3MDQyNjELMAKGA1UEBhMCSVQCCFZwqTJXlRldMA0GCSqGSaF1awQCAUaoIIBZzAYBgkqhkiG9w0BCQMxwCwYJKoZIhvcNAQcBMBwGCSqGSib3DQEBJTEPFw0yMjA4MTAwODAxNTZaMCOGCSqGSib3DQEJNDEgMB4wDQYJYIZIAWUDBAIBBQChDQYJKoZIhvcNAQELBQAwLWYJKoZIhvcNAQkEMSIEIIyS24577o6HAeqQ50U2H1LF5JMnJaRgd8ISguoanmBYMIHMBgsqhkiG9w0BCRACLzGBVDCBuTCBTjCBswQgxbkITFBrIuSqr8M6xRqZCQN72rMbhdR7YtuXrw186f4wgY4wgYGkfzB9MSYwJAYDVQDDb10Yw1pcmlhbCBDQSBGaXJtYSBRdWFSawZpY2F0YTEgMB4GA1UECwwXQ2VydGlmawNhdGlvbiBBdXRob3JpdHkxJDAiBgNVBAoMG05hbWlyawFsIFMucC5BLi8wMjA0NjU3MDQyNjELMAKGA1UEBhMCSVQCCFZwqTJXlRldMA0GCSqGSib3DQEBCwUABIIBAIgAfXNg3DxOPd0e5QTtftTmV4Zu/+cGLT4xaB2x2/14o3NmKrjHcMq4NBDf8XiF0o8YTrVQYqYNLxDaw5JpjTpqfbwun4wuIdkQqsg6tIRTIy3w
/v01oMk/X1sj7H+6wSffCRmIV5dwTILHUX0MRXiPA000aCsZT0852xwkXUB7z8/jawfUK4bLoz
/ckgFmV3YhRLhth7sLwZVjgUELd6ukCiwCftP9R4KEwXEYu4YmBW9pknFDrDGZgTTYChsITLtkJkNarzL/4JtxXcA7
/FALCxy0HcnYNta+iCw4N3I/E0PIVZQ5XibNraAF01ulJIZa1yC+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>p1kGoLorh0dGXK0FV4NAAZSsFxE13YEBXSXI8t lWCSkoDwKLN9wHeKaosAuuJMuL6Vl61AxiEoBFgh5
/ufQLFKwaiMqvB5VYD62yXdp8f2dtMeCfqZGwLEV4ci/kd1iMvE2eYiGornXk9NoF+eg/+h6W8TZwSnYrjp0YlF1oJxOG1
/r5qr+OVvwQ7n73fo3Qe6Rjw
/TuSI5V+wDaboctuCIwlK5gM8R4cT552PrNLsnVYmwR4epSUTx5VYwag6IhEHYPUTUkbMGpvN+0C0cZY7Nq0HPfqgrqks0HkMr4Z99DQAK0qS
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>MIIG1jCCBL6gAwIBAgIIKS0
/4EWmpWgWdQYJKoZIhvcNAQELBQAwwJjAKBgNVBAMMHVlc3QgTmFtaXJpYWwRVUgUXVhbGlmawVkJENBMR8wHQYDVQQLEDBZUcnVzdCBT
ZXJ2aWNLIFByb3ZpZGVyMRgwFgYDVQQKDA90YW1pcmlhbCBTlnAuQS4xGjAYBgNVBGEVZBVElULTAyMDQ2NTcwNDI2MQswCQYDVQGEwJJV
DAeFw0xNzExMjQwOTAwMDBaFw0yMzExMjQwOTAwMDBaMG0xCzAJBgNVBAYTAlUMRowGAYDVQRhDBFWQVRJVC0wMjA0NjU3MDQyOTEYMBYGA1
UECgwPTmFtaXJpYWwgcyswLmEuMRgwFgYDVQQDDA9uYW1pcmlhbCBzLnAuYS4xOjAMBGNVBC4TBULEMTMzMIIBIjANBgkqhkiG9w0BAQEFAAO
CAQ8AMIIBCgKCAQEApgHu8U7UKgtDTBTy13rv9GKZk1+YUfuHPGJxFI3TxZrImmDtyM0hu1l400Y1mML5DZVgFXkRFLw0+Gk0ZLJYkG4FvTo5
khqu+vsd0wyh/Hkpei0wLfnKhCWhY0pmlwahh3a31U1qFVLZJNu1ybRRF8N4+yAQQ2DlNL7
/B1VAgq1gVt1uXjxvas8MAUjjDbg3eQYXkSn2FJbveDRs127eeXUu+uabqt/GU/Y77Rvd7IKW6aH+d0F0oU/s7
/dto7q393rPU30pwfVA3A1107C/jwFaSgIDdYhtGviT6Jbakk
/SM26QfKnQShrHsS9S9hCn3DZUfg53I4Ygn0HtjFntKwIDAQAB04ICWDCCAlQwgZQGccsGAQUFBwEBBIGHMIGEMEMGCCsGAQUFBzACHjdodHR
wcZovL2RvY3MudGVzdC5uYW1pcmlhbHRzcC5jb20vZG9jdW1lbnRzL05hbUNBNesuY3J0MD0GCCsGAQUFBzABHjFodHRwcZovL29jc3AudGVz
dC5uYW1pcmlhbHRzcC5jb20vb2NzcC9jZXJ0c3RhdHVzMBOGA1UdDgQWBBRM2g4Fw+kgJ7XcAbdsY2fK3wqG8TAFBgNVHSMEGDAWgBT8Hvd
/XQE+XufwSmAaj0aa1hw+njCBZgYIKwYBBQUHAQMEgcEwgb4wCAYGBACORgEBMAsGBGQAjkyBAwIBFDATBgYEAI5GAQYwCQYHBACORgEAGjCB
jwYGBACORgEFMIGEMEAw0mh0dHBZ0i8vZG9jcy50ZXN0Lm5hbWlyawFsdHNwLmNvbS9kb2N1bWVudHMvUERTL1BEU19lb15wZGYTAuMEAwO
mh0dHBZ0i8vZG9jcy50ZXN0Lm5hbWlyawFsdHNwLmNvbS9kb2N1bWVudHMvUERTL1BEU19pdC5wZGYTAuL0MF8GA1UdIARYMFYwPwYkKwYBBA
GCmmsBAGAwMDAUBgggrBgEFBQcCARYiaHR0cHM6Ly9kb2N2LnRlc3QubmFtaXJpYXw0c3AuY29tLzAJBgECAIvsQAEBMAGBGBGQAj3oBATA5BgN
VHR8EMjAwMc6gLKAqh1hodHRw0i8vY3JsLnRlc3QubmFtaXJpYXw0c3AuY29tL0NBNEsuY3JsMA4GA1UdDwEB
/wQEAWIGQDANBgkqhkiG9w0BAQsFAAOCAgEAE2+uvSjsQZwx2R+tH76IfrPcwFguJYlFAh044gu7evJ6
//h7EQc0Y6wSzMHM91mfOpnuHD2BP9NftA+qBqQZnwpLn3S+3WiM7L7wBG0LJE20Ji
/fw0JUzTojtDQ24h64kQUv+u9cygB4JtFWAZ74WbMjmeG15WtBbo9zUx5Z59qsMl+BuXUW23u7lbzIyZLKAL6w5qSrBJhafBuKtwNIYfMojtw
K3k0SiKhktoA1K/s74xp9ofUpM4EhtjXhkQXN754dUbXbh2zYtDC4qw7LvKUnx2y2Yh8t0sv+N0c5kRMHvr0IjMzuuY7
/Eu00ivR7ncUg5ABJA9TQ8RA7pNw9ID+t9MhrsBEMGLYRWAsylARVbXpDpgZcCZxas6HE+JJWgn+LEBFDJiEPdSuvYY/bLML3G5wan
/cTYic0TK/HGJxzqoAxUB1gks3eUmvstxz0zyTmJxjJBBWSTU5ulRjk
/oexLEjYprXeipjx0gB6J+2+LoXsBAj1KMgHLWZ9NBAQZ7EvvZi3SDc0A1ijtjDRRNbHvI1naV3e
/1wLYLEJaAughNBQYzf2hyv8ikDv1Wb3YGML2ruu5BDAn7YpM0+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:WSEException xmlns:ns2="http://service.ws.nam/">
          <error>44</error>
          <message>Codice OTP errato, riprovare con il prossimo codice</message>
        </ns2:WSEException>
      </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-T0-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:

tsp-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 utilities to extract 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>MIIGZDCCBbSgAwIBAgIIXPCpMleVGv0wDQYJKoZIhvcNAQELBQAwFTEuMCQGA1UEAwddTmFtaXJpYWwgQ0EgRmlybWEGUxVh
          bGlmawNhdGEeIDAeBgNVBAsMF0NlcnRpZmJlYXRpb24qQXV0aG9yaXR5MSQwIgYDVQKQKDBtOYW1pcmlhbCBTlNaUQS4vMDIwNDY1NzA0MjYxYXc
          ZAJBgNVBAYTAKLUMB4XDTE4MDEyMzE2MzcwMFOxDTI0MDQyMzE2MzcwMFOwGACxCzAJBgNVBAYTAKLUMRUwEwYDVQKQKDAxOT04gUUFJFU0VOVE
          UxFTATBgNVBAQMDERFTU8gQ09HTk9NRTESMBAGA1UEKgwJREVNTyB0T01FMRwwGgYDVQQFEhNjVDpETUNETk0xNVQxMEEyNzFPMR8wHQYDVQQ
          DDBZERU1PIE5PTUUGREVNTyB0T0d0T01FMRcwFQYDVQQUeW5ERU1PMTIzNDU2Nzg5MDCCASiwdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
          APiLqTYMjlxDSM0I9/HqowXmQPP7AKLZEWedZwvXfu+LgqG6kalwESS+vb3zFmznaMXCuhDYjg1SQc4YHHSrddlnA72Zv6pnY98h43M+Md03
          /RQ4wZ7bA/SgRljVxB2uJemDi0Fu/l09Yys50u384Dj9C4C2HLR6SbQ7s4L4Kx5x
          /KGSW6mN8AGaNgvKxQhgXwprifhJqSUTxG2tvP0+NnElSr3TaxONpCeYr3li/lwQPP00A2JW0wqz94uCGQU3y/HrYqHdX50ccb7jNd+ah3f0
          /U6RUvQoBcgUFG973eiFyv8SGymRHf9iwdanrLkTAKSFsYk8z9JfA0GN1WoBsCAwEAaAOCAYMwggMfMIGiBggrBgEFBQcBAQSB1TCBkjBUBg
          grBgEFBQcAwAoZiAHR0cHM6Ly9kb2NzLnRlc3QuZmlybWJFJXJ0YS5pdC9kb2N1bWVudHMvTmFtaXJpYXWxDQUZpcmlhUXVhbGlmawNhdGEuY3J
          0MDoGCCsGAQUFBzABhi5odHRwOi8vb2NzcC50ZXN0LmZpcm1hY2VydgEuaXQvb2NzcC9jZXJ0c3RhZHVzMB0GA1UdDgQWBBS+qAt2HpxfpSyL
          ftgy962gNRD4BzAfBgNVHSMEGDAwgbTb0AhQt7Yvlu477D60ZLTCSpap3jAvBggrBgEFBQcBAwQjMCEwCAYGBACORgEBMASGBGQAJkYBAwIBF
          DAIBgYEAi5GAQQwgGgBgNVHSAEggGHMIIbNtCCAZkGCysGAQQBgpprAQEDMIIbIDAwwBggrBgEFBQcCARYkaHR0cDovL3d3dy5maXJtYWNlcn
          RhLml0L21hbGVhbGktTU8vMIIbUgYIKwYBBQUHAgIwggFEHoIBQABJAGwAIAABwAHIAZQBZAGUAbgB0AGUAIABjAGUAcgB0AGkAZgBpAGMAYQB
          0AG8AIAADoACAAAdgBhAGwAaQbKAG8AIAABZAG8ABvACAACABLAHIAIAbMAgKAcgBtAGUAIABhAHAACABvAHMAAdABLAACAyWBvAG4AIAABwAHIA
          bwBjAGUAZAB1AHIAIYQAGAGEAdQb0AG8ABQbAHQAaQbJAGEALGAgAFQAaABpAHMAIAABjAGUAcgB0AGkAZgBpAGMAYQB0AGUAIABtAGEAeQAgA
          G8AbgBSAHkAIABIAGUAIAB1AHMAZQBKACAAZgBVAHIAIAAB1AG4AYQB0AHQAZQBBAQGAZQBKAC8AYQB1AHQABwBtAGEADABLAGQAIABkAGkAZw
          BpAHQAYQBSACAACwBpAGcAbgBhAHQAdQByAGUAcwAUMEkGA1UdHwRCEAwPqA8oDqG0Gh0dHA6Ly9jcmVudGVzdC5maXJtYWNlcnRhLml0L0Z
          pcm1hQ2VydgFRdWFSawZpY2F0YTEuY3JsMA4GA1UdDwEB/wQEAWIGQDANBgkqhkiG9w0BAQsFAAOCQAQEAec6anyYgKJqGF
          /nwcRwKPNAS0iKbJoARGxYVJxEF2DWTg5/ck6qLdArjsXzdqLjWNE0L4oEpg
          /+ZRLASiTD0RFbutUffEpejsaYINSbseTU7sHw34h0VIqeIySy9KmTHcYTB52vVUPQKcQp8xBrzAw
          /7GheKgGt0gixNKGt+qLl5ytt0Vc2Gh4DfM7+86ftYU8ULrFH4spac4Qdb8W1w5rBxZXYFU2EbvsIP7DI6enKlvUcpQ5VLAw0Nw9fskXy7bzT
          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:DMCDNM15T10A2710,
          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>