Signing Tool API Anatomy (Draft) v-0.2.1

This document is intended for Portal / Developers / Websites to integrate the API in their Web application and Desktop Application.

Standard Disclaimer

- The specification can always change at last minute.
- This is intended for selected people, and is confidential.

Anatomy of Command

Request Command
This is a root element it contains the following elements

All the element are sent in **<request>**

<command>
<ts>
<txn>
<certificate>
<file>
<pdf>
<data>

Element Details

<command>
This is the actual command to help you perform operations using our Application.
(This is Mandatory Element)

Sl.	Command	Description
01.	pkiNetworkSign	This command is used to Sign, XML, TEXT, PDF, the data has to be passed in Base64.

02.	pkiNetworkSignHash	This command is used to sign Hash instead of sending the Base64 File.
03.	pkiNetworkVerSign	This command is used to Verify Signature.
04.	pkiNetworkEnc	This command is used for Encryption of Data.
05.	pkiNetworkDeEnc	This command is used for De Encryption of Data.

<ts>
This is Time Stamping element, which the server would pass to the Signing Tool.
(This is Optional Element)

The signing tool would compare with the Client system Date and time and see the difference, if there is difference of 30 minutes from the Server Date and Time it would not Sign.

<txn>
A unique ID could be passed with each Signing Request, it would help Web Application to get the response and map it to correct request.
(This is Optional Element)

<certificate>
The application could also pass values, to prompt the signing tool to match for a certificate present in the Browser Store and operations could be performed.
(This is Optional Element)

- If there are more than One Signature matching the criteria, then a Box showing all the matching signature would be shown, and the user could select the signature.
- If only One Signature is found, then our application would only show the pop up for Password and it would automatically sign.
- If no Parameters are sent then all allowed certificate would be shown.

The <certificate> can used with the following attributes with this element.

<attribute name="CN">John</attribute>
<attribute name="O">John & Company</attribute>
<attribute name="OU">Management</attribute>
<attribute name="T">CEO</attribute>
<attribute name="E">SomeEmail@domain.com</attribute>
<attribute name="SN">A123</attribute>
<attribute name="CA">Capricorn CA</attribute>
<attribute name="TC">SG</attribute>
<attribute name="AP">application id</attribute>

<attribute name="VD">Application Validity Date </attribute>

CN = Subject Name
O = Organization name.
OU = Organization Unit.
T = Title
E = eMail
SN = Serial number.
CA = Certifying Authority.
TC = Type of certificate, allowed variables **SG** (Signing) **EN** (Encryption)
AP = Application Id (This is intended for application developers)
VD= Application Validity Date

<file>
The Type of Data which needs to used by the commands
(This is Mandatory Element)
Allowed variables **PDF**, **TXT** or **XML**

<pdf>
This Element is used when <File> element is PDF
(This is Optional Element)

- This Element is is used when we would want to sign the PDF file in particular Page and in Particular Place.
- In case no values are passed, the signing would happen in the file invisible form (No Physical footprint would left in the file.
- If the values are passed with <File> not equal to PDF the values present would not be affect the Signing process.
- This element is used for Signing only.

The attributes available are

<page> - Only Numeric, Zero is not allowed
<cood>- Only Numeric allowed, two numeric value allowed, it would try to sign the pixels mentioned (depending on the resolutions of the file saved), else it would generate an error. Please separate the value by a comma, The first figure is horizontal and the next figure is Vertical on the page (eg. 79,100)
<size> - Only Numeric allowed, two numeric value allowed, and it would print the signature in the box mentioned here. Please separate the value by a comma, The first figure is horizontal and the next figure is Vertical on the page (eg. 79,100)

<data>
The actual data which is encoded in Base64 format.
(This is Mandatory Element)

A Sample Structure

```
<request>
 <command>pkiNetworkSign</command>
        <ts>2017-03-22T12:23:11.3820412+05:30</ts>
        <txn>unique id</txn>
        <certificate>
                <attribute name="CN"></attribute>
                <attribute name="O"></attribute>
                <attribute name="OU"></attribute>
                <attribute name="T"></attribute>
                <attribute name="E"></attribute>
                <attribute name="SN"></attribute>
                <attribute name="CA"></attribute>
                <attribute name="TC">sg/en</attribute>
                <attribute name="AP"></attribute>
                <attribute name="VD"></attribute>

        </certificate>
        <file>
                <attribute name="type">xml/pdf/text</attribute>
        </file>
        <pdf>
                <page></page>
                <cood>78,56</cood>
                 <size>100x200</size>
         </pdf>
        <data>
                base64 encoded
        </data>
</request>


<response>
        <command>pkiNetworkSign<command>
        <ts>2017-03-22T12:23:11.3820412+05:30</ts>
        <txn>unique id</txn>
        <status>ok</status>
```

```
        <data>base64 encoded</data>
</response>
Or
<response>
        <command>pkiNetworkSign<command>
        <ts>2017-03-22T12:23:11.3820412+05:30</ts>
        <txn>unique id</txn>
        <status>failed</status>
        <error code="">error message</error>
</response>
```

ERROR Codes

Our Errors are divided in three different Categories

- All Error starting from ER - are related to Application
- All Error Starting from CR - are related to Certificate and Licensing related.
- All Error Starting from PE - are related to API Data and before Signing.
- All Error Starting from OT- are related to Actual Signing.

Error code	**Error Message**	**Action**
ER-01	Latest version available	Download new version of the application.
ER-02	Licence expired	Renew your Software licence
ER-03	Internal error	Send details to support by mail
ER-04	Operation not allowed	An illegal Operation was attempted.
ER-05	Invalid command name	The command not supported in this version of the software.
ER-06	Licence file not found	Please acquire Software Licenses or refresh
ER-07	Unable to Execute Commands	Please restart the application
ER-08	Software License Corrupted	Please get a copy of the Software license.
ER-09	The System Data and Time is not correct	There is error in Data and Time or Time Zone.
ER-10	Invalid CRL or OCSP	Please delete the files and Sync the software.
ER-11	The Sync is not	

Error code	**Error Message**	**Action**
CR-01	No License Found	There were no Certificate License found.
CR-02	Chain certificate missing	
CR-03	Root certificate is missing	

CR-04	Invalid CRL or OCSP	
CR-05	Invalid or corrupted Licenses	Please delete all the Licenses
CR-06	Certificate not found	Please acquire Software Licenses or refresh

Error code	**Error Message**	**Action**
PE-01	Incomplete request xml	Correct the request xml
PE-02	Missing or invalid data element value	
PE-03	Invalid file type value	Pass any of them XML, PDF, TEXT
PE-04	Page number is out of range	
PE-05	Invalid page number	
PE-06	Invalid coordinate	
PE-07	Invalid box size	
PE-08	Invalid type of certificate	SG : for signing EN: for encryption or Decryption

Error code	**Error Message**	**Action**
OT-01	Invalid password supplied	
OT-02	User cancel the request	
OT-03	Private key not found	
OT-04	USB Device Not Responding	
OT-05	Signing Time Out.	
OT-06	Signing Malfunction	
OT-07	Invalid Page Number	
OT-08	Invalid Coordinates	

OT-09	Invalid Box Size	
OT-10	Invalid base64 encoded data	
OT-11	Base64 data and element mismatch	
OT-13	Signature not found	