Lista de comandos do Assinador SERPRO

URL: wss://assinador-desktop.serpro.gov.br:65166/signer/

Comando: Version

Retorna versão do Assinador

{"command": "version"}

Comando: List

Retorna a lista de comandos/funcionlidades do Assinador

{"command": "list"}

Comando: Sign

Gera uma assinatura digital.

Parâmetros:

requestId:
- Identificador da requisição

type:
- file: assina um arquivo local
- text
- hash
- base64
- pdf

listOfInputData:
De acordo com o type
- file (null, branco)
- text: “texto puro”
- hash: []bytes do hash em base64
- base64: []bytes do conteúdo em base64
- pdf: []bytes do arquivo PDF em base64, neste caso o retorno são os []bytes do PDF já assinado

textEncoding:
Valido para type do tipo Text. (ex: UTF-8,ISO-8859-1), senão informado o padrão é UTF-8

attached =
- true (anexa o conteúdo à assinatura)
- false

signaturePolicy =
-RB
-RT
-RV
politica de Assinatura ex: RT→ AD_**RT**_CADES_2_3, RV → AD_**RV**_CADES_2_3 (conforme o componente Demoiselle-Signer), default é **RB**

outputDataType=
- base64: (default) da assinatura em base64
- file: gera um arquivo com a assinatura

Os parâmetros abaixo, são usados apenas para o type: PDF

pdfStampPosX: Valor da posição da linha onde inserir o selo
pdfStampPosY: Valor da posição da coluna onde inserir o selo
pdfStampPage: Valor correspondente á pagina onde inserir o selo
pdfInvisibleSignature: true indica que não haverá selo

Exemplos (quase todos os parâmetros podem ser combinados):

Assinar um texto:
{"command": "sign","type": "text", "listOfInputData": ["teste"]}
{"command": "sign","type": "text", "listOfInputData":["teste","teste"]}

Assinar um texto com politica de carimbo do tempo
{"command": "sign","type": "text", "listOfInputData": ["teste"], "signaturePolicy":"RT"}
Assinar um texto usando enconding ISO-8859-1
{"command": "sign","type": "text", "listOfInputData": ["teste"], "textEncoding": "ISO-8859-1"}
Assinad um texto e inclui o contéudo na assinatura
{"command": "sign","type": "text", "listOfInputData": ["teste"],"attached": "true"}
Assinar um hash
{"command": "sign","type": "hash", "listOfInputData": ["hash em base64"]}
Assinar um hash com carimbo do tempo
{"command": "sign","type": "hash", "listOfInputData":["hash em base64","hash em base64"], "signaturePolicy":"RT"}
Assinar um contéudo em base 64
{"command": "sign","type": "base64", "listOfInputData": [“conteúdo em base64"]}
{"command": "sign","type": "base64", "listOfInputData": ["conteúdo em base64"]}
Assinar um arquivo PDF
{"command": "sign","requestId":"NNNN","type":"PDF","listOfInputData":["PDF em base64"],"outputDataType":"file","fileName":"teste","pdfStampPosX":"NN","pdfStampPosY":"NN","pdfStampPage":"NN","signaturePolicy":"RT"}
{"command": "sign","requestId":"NNNN","type":"PDF","listOfInputData":["PDF em base64","PDF em base64"],"pdfStampPosX":"NN","pdfStampPosY":"NN","pdfStampPage":"NN","signaturePolicy":"RT"}

Retorno:


```
{
   "original": "para compatibilidade, até a versão 2.7.1 do Assinador",
   "signature": "para compatibilidade, até a versão 2.7.1 do Assinador",
   "publicKey": "ABC...",
   "by": {
      "alias": "ABC...",
      "provider": "ABC...",
      "subject": "ABC...",
      "notBefore": "DD MMM YYYY HH:MM:SS GMT",
      "notAfter": "DD MMM YYYY HH:MM:SS GMT"
   },
   "originalFileName": "",
   "notAfterDateCertificate": "MMM DD, YYYY HH:MM:SS PM/AM",
   "msgError": "",
   "msgWarn": "",
   "hasError": true/false,
   "listOfOriginalContents": [
      "ABC..."
   ],
   "listOfSignatures": [
```

```
      "ABC..."
   ],
   "command": "sign",
   "requestId": NNNNN,
   "actionCanceled": false/true
}
```

Para executar os testes, recomendamos o uso do Firefox com a extenção: Weasel
https://addons.mozilla.org/pt-BR/firefox/addon/websocket-weasel/?utm_source=addons.mozilla.org&utm_medium=referral&utm_content=search

Comando: Verifiy

Valida uma assinatura digital

Parâmetros:

requestId:
- Identificador da requisição

type:
- text
- base64
- hash
- pdf

inputData:
De acordo com o type
- text: texto puro
- base64: []bytes do contéudo em base64
- hash: []bytes do hash em base64
- pdf: []bytes do arquivo PDF em base64

algorithmOIDHash (para validar por hash)
- oid do algoritmo 2.16.840.1.101.3.4.2.3 = sha512 ou 2.16.840.1.101.3.4.2.1 = sha256

inputSignature
- []bytes da Assinatura em base 64

Exemplos:

Verifica conteúdo em texto puro.
```
{"command": "verify","type": "text", "inputData": "texto", "inputSignature":" Assinatura em base64"}
```

Verifica qualquer contéudo em base64
```
{"command": "verify","type": "base64", "inputData": " Contéudo em base64", "inputSignature":" Assinatura em base64"}
```

Verifica assinatura com contéudo anexado
```
{"command": "verify","type": "base64", "inputSignature":" Assinatura em base64"}
```

Verifica por hash
```
{"command": "verify","type": "hash", "inputData": "hash em base64", "algorithmOIDHash": " oid do algoritmo"}
```

Verifica PDF assinado
```
{"command": "verify","type": "pdf", "inputData": "arquivo PDF em base64"}
```

Retorno:

```
{
   "isValid": true/false,
   "hasError": false/true,
   "numberOfSignatures": N,
   "signerSignatureValidations": [
      {
         "hasError": false/true,
         "signDate": "DD-MMM-YYYY HH:MM:SS:s GMT",
         "signaturePolicy": "ABC...",
         "assinante": "ABC...",
         "cadeiaCertificado": [
            "Autoridade Certificadora 3 nível",
            "Autoridade Certificadora 2 nível",
            "Autoridade Certificadora Raiz Brasileira vN"
         ],
         "validatorErrors": [],
         "validatorWarnings": [],
         "isPF": true/false,
         "valCPF_CNPJ": "NNN…."
      }
   ],
   "command": "verify",
   "requestId": nnn,
   "actionCanceled": false/true
}
```

Comando: TimeStamp

Gera um carimbo do tempo (sem assinatura)

Parâmetros:

requestId:
- Identificador da requisição

type:
- signature
- haw
- hash

inputContent
De acordo com o Type
- signature: []bytes de uma assinatura em base64
- haw: []bytes do contéudo a ser assinado em base64
- hash: []bytes do hash em base64 a ser assinado em base64

algorithmOIDHash
- oid do algoritmo 2.16.840.1.101.3.4.2.3 = sha512 ou 2.16.840.1.101.3.4.2.1 = sha256

inputSignature
- []bytes da Assinatura em base 64

Exemplos:

Gera carimbo para conteúdo

```
{"command": "TimeStamp","inputContent":"Conteúdo em base64", "type":"raw"}
```

Gera carimbo para uma assinatura
{"command": "TimeStamp","inputContent":"Conteúdo em base64", "type":"Signature"}

Gera carimbo para hash de um contéudo
{"command": "TimeStamp","inputContent":"Conteúdo em base64", "type":"hash"}

Comando: Attached

Recupera o conteúdo em uma assinatura com conteúdo anexado

Parâmetros:

requestId:
- Identificador da requisição

inputSignature
- []bytes da Assinatura em base 64

Exemplo

{"command": "attached","inputSignature":"Assinatura em base64"}

Comando: CoSign

Gera uma co-assinatura em uma assinatura

Parâmetros:

requestId:
- Identificador da requisição

type:
- hash
- base64

inputData:
De acordo com o type
- hash: []bytes do hash em base64
- base64: [] bytes do contéudo em base64

signatureToCoSign = []bytes da Assinatura anterior em base 64

signaturePolicy =
-RB
-RT
-RV
politica de Assinatura ex: RT→ AD_**RT**_CADES_2_3, RV → AD_**RV**_CADES_2_3 (conforme o componente Demoiselle-Signer), default é **RB**

Exemplos:

Co-assinar com hash

{"command": "cosign", "type": "hash","inputData": "hash em base64",
"signatureToCoSign":"Assinatura em base64"}

Co-assinar com base64 (arquivo todo em base64)

```
{"command": "cosign", "type": "base64","inputData": "Contéudo em base64",
"signatureToCoSign":"Assinatura em base64"}
```

Co-assinar com base64 (arquivo todo em base64), com politica de carimbo do tempo

```
{"command": "cosign", "type": "base64","inputData": "Contéudo em base64",
"signatureToCoSign":"Assinatura em base64","signaturePolicy":"RT"}
```

Comando: signxml

Gera uma assinatura em formato XML

Parâmetros:

requestId:
- Identificador da requisição

type:
- file: assina um arquivo XML local
- xml: assina um XML em texto puro
- hash: assina um hash em base64, gera uma assinatura desanexada
- xmlBase64: conteúdo de um XML codificado em base64
- contentBase64: um conteúdo qualquer em base64, para gerar uma assinatura desanexada

listOfInputData:
- De acordo com o type
- {Para "file" não informar}
- xml: texto puro contendo o arquivo XML
- hash: []bytes do hash em base64
- xmlBase64: conteúdo do XML codificado em base64
- contentBase64: []bytes do conteúdo a ser assinado em base64

signaturePolicy =
-RB
-RT
-RV
politica de Assinatura ex: RT→ AD_**RT**_CADES_2_3, RV → AD_**RV**_CADES_2_3 (conforme o componente Demoiselle-Signer), default é **RB**

outputDataType:
- file : gera um arquivo localmente (somenta para o Type file)
- xml : devolve a assinatura em formato texto representando o xml
- base64 : devolve a assinatura em formato base64 (byte[]) representando o xml

algorithm:
- "SHA256withRSA"
- "SHA512withRSA"

Exemplos:

```
{"command": "signxml","type": "xml", "listOfInputData":["<?xml version=\"1.0\"
encoding=\"UTF-8\"?>\n<raiz>\n <documento>documento um</documento>\
n<conteudo>xml para assinar</conteudo>\n</raiz>"]}
```

{"command": "signxml","type": "xml", "listOfInputData":["<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<raiz>\n <documento>documento um</documento>\n<conteudo>xml para assinar</conteudo>\n</raiz>","<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<raiz>\n <documento>documento dois</documento>\n<conteudo>segundo xml para assinar</conteudo>\n</raiz>"]}

{"command": "signxml","type": "file","outputDataType":"file", "signaturePolicy":"RT"}

{"command": "signxml","type": "xmlBase64","listOfInputData": ["PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPHJhaXo+CiA8ZG9jdW1lbnRvPnVtIGRvY3VtZW50bzwvZG9jdW1lbnRvPgogPGNvbnRldWRvPnhtbCBwYXJhIGFzc2luYXI8L2NvbnRldWRvPgo8L3JhaXo+CjwvcmFpej4K"], "outputDataType":"base64"}

{"command": "signxml","type": "hash","listOfInputData": ["4NaTO1LGBGer9yN0G7Jm6wOs66TuSeOaNAlPyEapRMw="], "outputDataType":"base64", "algorithm":"SHA256withRSA"}
{"command": "signxml","type": "contentBase64", "listOfInputData": ["PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPHJhaXo+CiA8ZG9jdW1lbnRvPnVtIGRvY3VtZW50bzwvZG9jdW1lbnRvPgogPGNvbnRldWRvPnhtbCBwYXJhIGFzc2luYXI8L2NvbnRldWRvPgo8L3JhaXo+CjwvcmFpej4K"]}

Comando: preparesign

Prepara a assinatura para assinaturas acima de RB

Parâmetro:

type:
- text: texto puro
- hash: hash em base64
- base64: contéudo em base64

listOfInputData:
- De acordo com o type

textEncoding:
Valido para type do tipo Text. (ex: UTF-8,ISO-8859-1)

attached =
- true (anexa o conteúdo à assinatura)
- false

signaturePolicy =
-RB
-RT
-RV
politica de Assinatura ex: RT→ AD_**RT**_CADES_2_3, RV → AD_**RV**_CADES_2_3 (conforme o componente Demoiselle-Signer), default é **RB**

Exemplos:

{"command": "preparesign","type": "text", "listOfInputData":["teste1"], "attached": "true", signaturePolicy = "RT"}
{"command": "preparesign","type": "text", "listOfInputData":["teste1"], signaturePolicy = "RT"}

{"command": "preparesign","type": "text", "listOfInputData":["teste1","teste2"], signaturePolicy = "AD_RT_CADES_2_3"listOfInputData}
{"command": "preparesign","type": "hash", "listOfInputData": ["cL35UO6xkjJbz2nn+l465BIduzF0b1dEfBEq02SR5x+lvz+dxYvRNYqbT4+h54CaYQ0Wx5xD1+J7hGGUUENmlg=="],signaturePolicy = "AD_RT_CADES_2_3"}
{"command": "preparesign","type": "hash", "listOfInputData": ["i19O8wXZgrnPHHNIEXa9+elCbv7PuQnoX25a+LzfLThcLc2YR53KM3EmeSJGYwhrM84rFUShcH6MsZmHV427ew==","i19O8wXZgrnPHHNIEXa9+elCbv7PuQnoX25a+LzfLThcLc2YR53KM3EmeSJGYwhrM84rFUShcH6MsZmHV427ew=="],signaturePolicy = "AD_RT_CADES_2_3"}
{"command": "preparesign","type": "base64", "listOfInputData": ["Y29udGXDumRvIGF0YWNoYWRvLgo="],signaturePolicy = "AD_RT_CADES_2_3"}
{"command": "preparesign","type": "base64", "listOfInputData": ["Y29udGXDumRvIGF0YWNoYWRvLgo="], "attached": "true"}
{"command": "preparesign","type": "base64", "listOfInputData": ["Y29udGXDumRvIGF0YWNoYWRvLgo=","Y29udGXDumRvIGF0YWNoYWRvLgo="],signaturePolicy = "AD_RT_CADES_2_3"}