

Спецификация
на
Локална услуга за електронно подписване в браузър

Версия 1.14

София, Септември 2017

Управление на документа	
Подготвен от	Валентин Иванов
Прегледан от	Димитър Николов
Одобен от	

ПРОМЕНИ НА ВЕРСИЯТА		
Версия	Промяна	Дата/Извършител
1.00	Начално създаване	26.08.2016/ВИ
1.01		
1.02	Добавено <ul style="list-style-type: none"> • Опции за сваляне и вдигане защитата на MS Edge • Сваляне защитата на MS Edge след инсталация 	03.10.2016/ВИ
1.03	Добавено <ul style="list-style-type: none"> • Подписани елементи в JSON 	07.10.2016/ВИ
1.04	Добавено <ul style="list-style-type: none"> • Проверка на подписаните елементи в JSON 	10.10.2016/ВИ
1.09	Добавено <ul style="list-style-type: none"> • Подписване без извикване на GetSigner 	01.03.2017/ВИ
1.12	Добавено <ul style="list-style-type: none"> • Timeout 15 минути за въвеждане на ПИН • Параметризирано е съобщението за потвърждаване на подписаните данни • Заявка за спиране на приложението 	13.04.2017/ВС
1.13	Добавено <ul style="list-style-type: none"> • Заявка за статуса на подписваните документи от масив • Сменено лого на фирмата по време на инсталация • Скрит бутон за смяна на сертификат • Сменен прозорец за избор на сертификат при MSCapi • Добавени съобщения за различни събития по време на подписване 	06.06.2017/ВС 12.07.2017/С. Славов
1.14	Добавено: <ul style="list-style-type: none"> • Разписване с различни видове карти • Пропърти файл за избиране на карта и нейният dll файл • Прозорец за избиране на dll в случай, че картата не присъства в пропърти файла • Премахнат е прозореца за избор на вида карта при PKCS11 подпис 	18.09.2017/С. Славов

	<ul style="list-style-type: none">• При инсталиране на BISS за Windows се инсталират и удостоверителните вериги на Борика• Сменена Java версия от 1.8.0_121 на 1.8.0_144	
--	---	--

Table of Contents

1. Описание	4
2. Системни изисквания	5
3. Процес по полагане на електронен подпис	5

СПЕЦИФИКАЦИЯ
на софтуер за електронно подписване

4.	Локална услуга за електронно подписване (BISS)	6
4.1.	Cross-Origin Resource Sharing (CORS) проверка	6
4.2.	Проверка на версията	7
4.2.1.	Запитване	7
4.2.2.	Отговор	8
4.3.	Избор на сертификат за подписване	9
4.3.1.	Запитване	9
4.3.2.	Отговор	10
4.4.	Създаване на подпис.....	11
4.4.1.	Запитване	11
4.4.2.	Отговор	13
4.5.	Статус кодове	14
4.6.	Заявка за спиране на приложението	15
4.7.	Заявка за статуса на подписваните документи от масив	16
4.7.1.	Запитване	16
4.7.2.	Отговор	17
5.	Извикване на локалната услуга за електронно подписване.....	17
5.1.	Javascript код за избиране на сертификат за подписване	17
5.2.	Javascript код за подписване	18

1. Описание

Приложения софтуер за електронно подписване изпълнява технология, която позволява локално полагане на електронен подпис без използване на активна компонента в Интернет браузър при Клиента. То се базира на спецификация [Signature Creation Service 1.0.1 \(June 30, 2015\)](#), която е допълнена с необходимите функции с цел гарантиране на работоспособността на основното Уеб-базирано приложение.

Приложния софтуер е реализиран като локален Web сървър, който позволява достъп до създаваща подписи функционалност посредством използване на защитен протокол (HTTPS).

При стартиране на Локалната услуга за електронен подпис (BISS) се проверява първият свободен порт от следните портове : 53952, 53953, 53954 и 53955. Следователно HTML приложението също трябва да сканира тези портова за намиране на BISS.

2. Системни изисквания

BISS може да работи на следните операционни системи:

- Windows 7/8/10, 32/64 bit;
- Linux Ubuntu
- MacOS Yosemite, El Capitan, Sierra

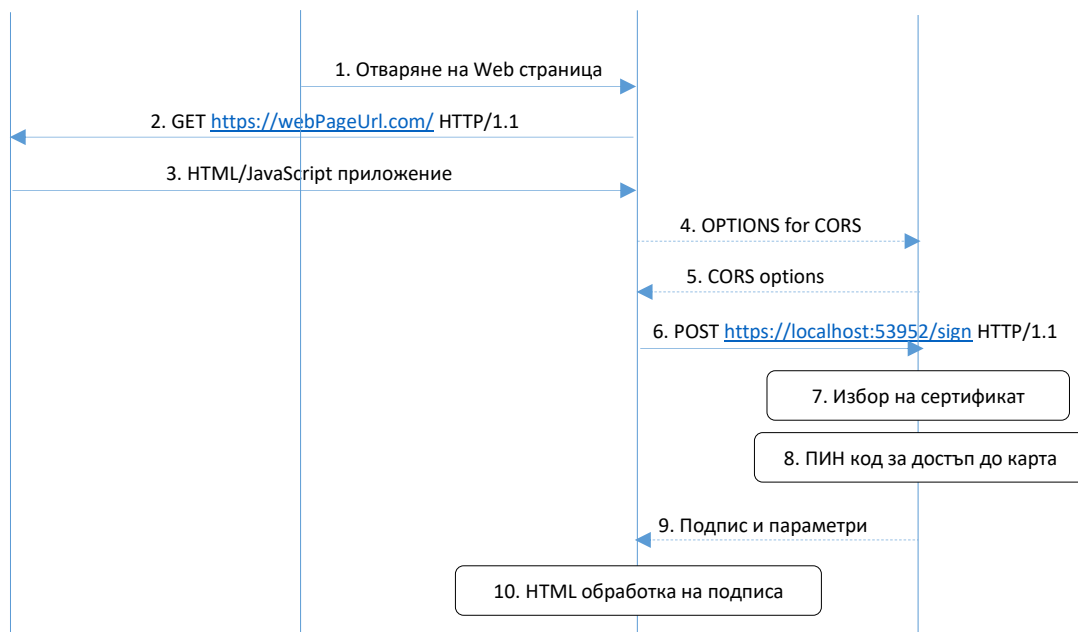
Задължително е да има инсталирана Java 7 или по висока версия.

3. Процес по полагане на електронен подпис

По – долу е дадена диаграмата описваща процеса по полагане на електронен подпис. Схемата е съсредоточена върху интерфейса между HTML приложението и BISS.



СПЕЦИФИКАЦИЯ
на софтуер за електронно подписване



1. Потребителят въвежда в браузъра URL адреса на HTML приложението.
2. Браузъра прави защитена заявка към сървъра на HTML приложението.
3. Браузъра зарежда HTML приложението от сървъра.
4. Браузъра прави OPTION заявка към BISS за да провери дали CORS заявките са позволени.
5. BISS връща CORS параметри позволявайки заявката да бъде направена от текущото HTML приложение (идентифицирано чрез Origin параметъра).
6. HTML приложението създава заявка за създаване на подпис.
7. Потребителят избира сертификата за подписване.
8. Потребителят въвежда ПИН код за достъп до смарт картата.
9. BISS връща отговор на HTML приложението. Отговорът съдържа подписа, алгоритъма на подписване и сертификационната верига.
10. HTML приложението обработва (валидира, създаване на контейнер) подписа.

4. Локална услуга за електронно подписване (BISS)

4.1. Cross-Origin Resource Sharing (CORS) проверка

Браузърите, които поддържат CORS ще направят проверка дали CORS заявките са позволени да бъдат изпращани към BISS. Това се прави като се изпраща OPTION заявка към BISS указваща Origin атрибута на HTML приложението, както и позволените методи и Header атрибути.

Примерна заявка е показана по долу:

```
Request URL: https://localhost:53952/sign
Request method: OPTIONS
Remote address: 127.0.0.1:53952
Status code: ● 200 OK
Version: HTTP/1.1
```

Filter headers

▶ Response headers (0,322 KB)

▼ Request headers (0,404 KB)

Host: "localhost:53952"

User-Agent: "Mozilla/5.0 (Windows NT 10.0; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0"

Accept: "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8"

Accept-Language: "en-US,en;q=0.5"

Accept-Encoding: "gzip, deflate, br"

Access-Control-Request-Method: "POST"

Access-Control-Request-Headers: "content-type"

Origin: "http://b-trust:8888"

Connection: "keep-alive"

В резултат на OPTION заявката BISS трябва да върне HTTP отговор със указани CORS атрибути. Пример на такъв отговор е показан по долу:

```
Request URL: https://localhost:53952/sign
Request method: OPTIONS
Remote address: 127.0.0.1:53952
Status code: ● 200 OK
Version: HTTP/1.1
```

Filter headers

▼ Response headers (0,322 KB)

Accept: "application/json, */*"

Access-Control-Allow-Headers: "Accept, Content-Type"

Access-Control-Allow-Methods: "GET, POST"

Access-Control-Allow-Origin: "http://b-trust:8888"

Cache-Control: "no-cache"

Connection: "keep-alive"

Content-Length: "0"

Content-Type: "application/json"

Date: "Thu, 21 Jul 2016 06:40:37 GMT"

4.2. Проверка на версията

4.2.1. Запитване

HTML приложението може да изпрати запитване за версията на BISS. Запитването за версията на BISS може да бъде използвано от клиентското приложение за откриване на BISS и потвърждение, че е стартирано.

HTTP параметри:

- HTTP метод: GET
- Request-URI: /version
- Други HTTP параметри се добавят от браузъра по спецификацията на HTTP/1.1 и

CORS.

Примерно запитване:

```
Request URL: https://localhost:53952/version
Request method: GET
Remote address: 127.0.0.1:53952
Status code: ● 200 OK
Version: HTTP/1.1
```

Filter headers

▶ Response headers (0,275 KB)

▼ Request headers (0,294 KB)

Host: "localhost:53952"

User-Agent: "Mozilla/5.0 (Windows NT 10.0; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0"

Accept: "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8"

Accept-Language: "en-US,en;q=0.5"

Accept-Encoding: "gzip, deflate, br"

Connection: "keep-alive"

4.2.2. Отговор

BISS отговаря на HTML приложението със следните HTTP параметри:

- Status-Code: 200
- Content-Type: application/json
- Други HTTP параметри се добавят от браузъра както е по спецификацията на

HTTP/1.1 и CORS.

Както и със следният JSON отговор:

- **version** (string, задължителен): версията, която BISS поддържа.
- **httpMethods** (string, задължителен): позволените HTTP методи за изпращане на заявка за подписване.
- **contentTypes** (string, задължителен): подържаните типове данни, които могат да бъдат изпратени към BISS.
- **signatureTypes** (string, задължителен): подържаните типове подписване от BISS.
- **selectorAvailable** (Boolean, задължителен): указва дали BISS поддържа филтър при избора на сертификат.
- **hashAlgorithms** (string, задължителен): подържаните хеш алгоритми.


```
Request URL: https://localhost:53952/version
Request method: GET
Remote address: 127.0.0.1:53952
Status code: ● 200 OK
Version: HTTP/1.1

Filter headers

Response headers (0,275 KB)
Accept: "application/json, */*"
Access-Control-Allow-Headers: "Accept, Content-Type"
Access-Control-Allow-Methods: "GET, POST"
Cache-Control: "no-cache"
Connection: "keep-alive"
Content-Length: "175"
Content-Type: "application/json"
Date: "Tue, 12 Jul 2016 10:41:45 GMT"

JSON
version: "1.0"
httpMethods: "GET, POST"
contentType: "data, digest"
signatureTypes: "signature"
selectorAvailable: true
hashAlgorithms: "SHA1, SHA256, SHA384, SHA512"
```

4.3. Избор на сертификат за подписване

Преди подписването, HTML приложението изпраща заявка за избор на сертификат.

4.3.1. Запитване

HTML параметри на заявката:

- HTTP метод: POST
- Request-URI: /getsigner
- Content-Type: application/json
- Други HTTP параметри се добавят от браузъра, както е по спецификация на HTTP/1.1 и CORS.

POST параметри на заявката:

- **Selector** (Object, по избор): съдържа параметрите за филтриране на сертификати:
 - **Issuers** (array, по избор): списък на позволени доставчици на удостоверителни услуги (ДУУ). Полето „Issuer“ задължително трябва да съответства на един от позволените ДУУ. Препоръчително е да се филтрират клиентските сертификати посредством селективния филтър „akis“ вчесто “Issuers“. Големината на буквите има значение.
 - **akis** (array, по избор): списък от „authority key identifiers“ на позволените ДУУ в base64 формат. Големината на буквите има значение.

СПЕЦИФИКАЦИЯ
на софтуер за електронно подписване

- **Keyusages** (array, по избор): списък на позволени стойности на атрибута „key Usage“ от клиентския сертификат (“digitalSignature”, “nonRepudiation”, “dataEncipherment”, “decipherOnly”, “encipherOnly”, “keyAgreement”, “keyEncipherment”, “keyCertSign”, “crlSign”). Ако има указани стойности, то клиентския сертификат трябва да ги има всички.

Примерна заявка за избор на сертификат:

```
Request URL: https://localhost:53952/getsigner
Request method: POST
Remote address: 127.0.0.1:53952
Status code: ● 200 OK
Version: HTTP/1.1
```

- ▶ Response headers (0,327 KB)
- ▼ Request headers (0,410 KB)
 - Host: "localhost:53952"
 - User-Agent: "Mozilla/5.0 (Windows NT 10.0; W....0) Gecko/20100101 Firefox/47.0"
 - Accept: "application/json, text/javascript, */*; q=0.01"
 - Accept-Language: "en-US,en;q=0.5"
 - Accept-Encoding: "gzip, deflate, br"
 - Content-Type: "application/json"
 - Referer: "http://localhost:8888/tester/index.html"
 - Content-Length: "58"
 - Origin: "http://localhost:8888"
 - Connection: "keep-alive"
- ▼ JSON
 - ▼ selector: Object
 - ▼ issuers: Array
 - 0: "CN=B-Trust Operational CA QES"

4.3.2. Отговор

BISS връща отговор със следните HTTP параметри:

- Status-code: 200
- Content-Type: application/json
- Други HTTP параметри се добавят от брауъра, както е по спецификацията на HTTP/1.1 и CORS.

Допълнителни параметри на отговора:

- **status** (string, задължителен): показва дали подписването е успешно.

Поддържаните стойности са “ok” и “filed”.

- **reasonCode** (string, задължителен): съдържа кода съответстващ на статуса на отговора. Всички кодове са описани в секция 1.4.

СПЕЦИФИКАЦИЯ
на софтуер за електронно подписване

- **reasonText** (string, задължителен): Съдържа текстово описание на статуса на отговора.
- **chain** (array, по избор): съдържа сертификационната верига на подписващия сертификат. Подписващият сертификат е на позиция 0. Всички сертификати са base64 кодирани. Наличен е ако подписването е успешно. Сертификационната верига може да не е пълна и да съдържа само подписващия сертификат.

```
Request URL: https://localhost:53952/getsigner
Request method: POST
Remote address: 127.0.0.1:53952
Status code: ● 200 OK
Version: HTTP/1.1
```

Filter headers

- Response headers (0,327 KB)
 - ACCESS-CONTROL-ALLOW-ORIGIN: "http://localhost:8888"
 - Accept: "application/json, */*"
 - Access-Control-Allow-Headers: "Accept, Content-Type"
 - Access-Control-Allow-Methods: "GET, POST"
 - Cache-Control: "no-cache"
 - Connection: "keep-alive"
 - Content-Length: "7967"
 - Content-Type: "application/json"
 - Date: "Tue, 02 Aug 2016 12:29:19 GMT"
- Request headers (0,410 KB)
 - JSON
 - status: "ok"
 - reasonCode: 200
 - reasonText: "Signer certificate is choosen"
 - chain: Array
 - 0: "MIIEHejCCBWKgAwIBAgIEAJolINDANB...5rqs1qom6xGNzAIKbFDSSyB20g=="
 - 1: "MIIIYzCCBkugAwIBAgIBAJANBgkqh...eNCiOycik3cPODHGPz8ZAvuQKNC2"
 - 2: "MIIHGDCCBQCgAwIBAgIBATANBgkq...eYfmn4Ttego0gsXTt+ENAgqn3bl="

4.4. Създаване на подпис

HTML приложението може да създава подпис чрез изпращане на заявка за подписване към BISS. В случай, че подписващият сертификат е наличен предварително, може да се извика метода за подписване без да се извиква метода /getsigner (за избор на подписващ сертификат).

4.4.1. Запитване

HTTP параметри на заявката:

- HTTP метод: POST
- Request-URI: /sign
- Content-Type: application/json

СПЕЦИФИКАЦИЯ
на софтуер за електронно подписване

- Други HTTP параметри се добавят от брауъра, както е по спецификацията на HTTP/1.1 и CORS.

POST параметри на заявката:

- **version** (string, по избор): съдържа версията, която се поддържа от HTML приложението и се очаква от BISS.
- **contents** (array, задължителен): съдържа base64 кодирани данни, които трябва да бъдат подписани.
- **signedContents** (array, задължителен): съдържа base64 кодирани данни, представляващи подписа на хеша на съответния елемент от атрибута **contents**.
- **signedContentsCert** (array, задължителен): съдържа base64 кодиран сертификат, който е подписал съответният елемент от атрибута **signedContents**.
- **contentType** (string, по избор): съдържа типа на данните за подписване. По подразбиране има стойност "digest". Подържаните стойности са "data" и "digest". Стойност "data" роказва, че параметъра "contents" съдържа информацията, която ще бъде хеширана(с указаната стойност в параметъра **hashAlgorithm**) преди да се подпише. Стойност "digest" показва, че параметъра "contents" съдържа информация(например hash), която трябва да се подпише без да се хешира. Ако стойността е "data" то **signedContents** не се валидира!
- **hashAlgorithm** (string, по избор): съдържа алгоритъма, който трябва да бъде използван за подписване. По подразбиране стойността е "SHA256". Подържаните стойности са "SHA1", "SHA256" и "SHA512". Да се обърне внимание, че ако стойността на параметъра "contentType" е "digest" то параметъра трябва да съдържа хеш, който е направен с хеш алгоритъма посочен в полето "hashAlgorithm".
- **signatureType** (string, по избор): съдържа формата на подписване. По подразбиране стойността е "signature", което означава, че е налице базово подписване. Поддържаните стойности са "signature".
- **signerCertificateB64** (string, задължителен): Base64 кодиран избраният сертификат за подписване.
- **confirmText** (array, незадължителен): Управява дали да се извежда диалоговия прозорец за потвърждаване на подписване на съдържанието и какъв текст да визуализира в него
 - Ако не е подадена стойност, диалоговия прозорец за потвърждаване на подписването не се визуализира
 - Ако е подаден "hash" като първи елемент от масива се визуализира част от хеш на съдържанието, което ще бъде подписано.
 - Ако се подаде произволен стринг като първи елемент от масива се визуализира подадения текст.

Примерна заявка за подписване към BISS е показана по долу:

СПЕЦИФИКАЦИЯ
на софтуер за електронно подписване

```
Request URL: https://localhost:53952/sign
Request method: POST
Remote address: 127.0.0.1:53952
Status code: ● 200 OK
Version: HTTP/1.1
```

🔍 Filter headers

- ▶ Response headers (0,326 KB)
- ▼ Request headers (0,406 KB)
 - Host: "localhost:53952"
 - User-Agent: "Mozilla/5.0 (Windows NT 10.0; WOW64; rv:49.0) Gecko/20100101 Firefox/49.0"
 - Accept: "application/json, text/javascript, */*; q=0.01"
 - Accept-Language: "en-US,en;q=0.5"
 - Accept-Encoding: "gzip, deflate, br"
 - Referer: "http://localhost:8080/WSP/singleUpload"
 - Content-Type: "application/json"
 - Content-Length: "6491"
 - Origin: "http://localhost:8080"
 - Connection: "keep-alive"
- ▼ JSON
 - contentType: "digest"
 - ▼ contents: Array
 - 0: "MYIB+TAYBgkqhkiG9w0BCQMxCwYJKoZIh...ECszNTkgMiA5IDlxNSAxMDACBACc6jY="
 - hashAlgorithm: "SHA256"
 - signatureType: "signature"
 - ▼ signedContents: Array
 - 0: "WfqLj29hJkU/nMzTnB12G+sVI4ZozjeCIM...o5e8uQ7yB1dFw3ofaYzC49v7krtvMRO3g=="
 - ▼ signedContentsCert: Array
 - 0: "MIIH7jCCBdagAwIBAgIDHpHYMA0GCSqG...1XBCe1WbrHd4QhsTRpwym6QOy3Y98N0"
 - signerCertificateB64: "MIIHgzCCBWugAwIBAgIEAJzqNjA...Cs/+yujggivoy2tO9XTUtb1rw=="
 - version: "1.0"

4.4.2. Отговор

BISS връща отговор със следните HTTP параметри:

- Status-Code: 200
- Content-Type: application/json
- Други HTTP параметри се добавят от браузъра, както е по спецификацията на HTTP/1.1 и CORS.

Допълнителни параметри на отговора:

- **status** (string, задължителен): показва дали подписването е успешно. Поддържаните стойности са "ok" и "filed".
- **reasonCode** (string, задължителен): съдържа кода съответстващ на статуса на отговора. Всички кодове са описани в секция 1.4.

СПЕЦИФИКАЦИЯ
на софтуер за електронно подписване

- **reasonText** (string, задължителен): Съдържа текстово описание на статуса на отговора.
- **signatures** (array, по избор): съдържа base64 кодиран подпис. Наличен е ако подписването е успешно.
- **signatureType** (string, по избор): съдържа типа на подписа. Поддържаните стойности са "signature". Наличен е ако подписването е успешно.

```
Request URL: https://localhost:53952/sign
Request method: POST
Remote address: 127.0.0.1:53952
Status code: ● 200 OK
Version: HTTP/1.1

Filter headers

Response headers (0,326 KB)
ACCESS-CONTROL-ALLOW-ORIGIN: "http://localhost:8888"
Accept: "application/json, */*"
Access-Control-Allow-Headers: "Accept, Content-Type"
Access-Control-Allow-Methods: "GET, POST"
Cache-Control: "no-cache"
Connection: "keep-alive"
Content-Length: "824"
Content-Type: "application/json"
Date: "Tue, 02 Aug 2016 12:40:15 GMT"

Request headers (0,407 KB)
JSON
status: "ok"
reasonCode: 200
reasonText: "Signature generated"
signatures: Array
  0: "JRy9nt1qDJp5kvNQxoobhhPpjKvHvQr...ExVQpiqsFXlv0LZCLRjFjt2/Ghrrg=="
  1: "JRy9nt1qDJp5kvNQxoobhhPpjKvHvQr...ExVQpiqsFXlv0LZCLRjFjt2/Ghrrg=="
signatureType: "signature"
```

4.5. Статус кодове

Статус кодовете, описващи процеса на подписване, в отговора от BISS са както следва:

- 2xx – показва успешно подписване
- 4xx – показва, че подписването е прекъснало поради някаква причина
- 5xx – показва, че BISS не работи правилно или не поддържа заявената функционалност.

Възможните кодове са описани в следната таблица:

reasonCode	Текст	Описание
200	Успех	Подписването е успешно
400	Лоша заявка	Лоша заявка, например:

СПЕЦИФИКАЦИЯ
на софтуер за електронно подписване

		<ul style="list-style-type: none"> • Непознат адрес (Request-URI) • Лошо форматиране на JSON • Липсващи задължителни параметри на заявката • Параметъра съдържа не поддържана стойност
401	Непозволен	<p>Подписването е отказано от потребителя:</p> <ul style="list-style-type: none"> • Не е избран сертификат за подписване • Потребителският сертификат няма заявените “key Usage” атрибути (грешен ПИН, Блокиран ПИН) • Потребителя няма сертификати отговарящи на филтъра в заявката
403	Забранен	<p>Заявката да подписване е отказана, заради:</p> <ul style="list-style-type: none"> • Друга заявка е в процен на обработка • Заявката за подписване е направена от непознат източник (не е използван HTTPS протокол за отваряне на HTML приложението изискващо подпис)
413	Заявката е твърде голяма	Размера на заявката е твърде голяма
500	Вътрешна грешка	BISS не работи правилно. Възможна причина е неправилното конфигуриране
501	Не се поддържа	Заявената функционалност не се поддържа или не е налична

4.6. Заявка за спиране на приложението

Заявката се използва за спиране на приложението.

Request URL: https://localhost:53952/close
Request Method: POST
Status Code: 200 / OK
Request Headers
Accept: application/json, text/javascript, */*; q=0.01
Accept-Encoding: gzip, deflate
Accept-Language: bg, en-US; q=0.7, en; q=0.3
Cache-Control: no-cache
Connection: Keep-Alive
Content-Length: 0
Host: localhost:53952
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko
Response Headers
Accept: application/json, */*
Access-Control-Allow-Headers: Accept, Content-Type
Access-Control-Allow-Methods: GET, POST
Access-Control-Allow-Origin: *
Cache-Control: no-cache
Connection: keep-alive
Content-Length: 38
Content-Type: application/json
Date: Mon, 12 Jun 2017 14:01:01 GMT

4.7. Заявка за статуса на подписваните документи от масив

Заявката служи за следене на текущия статус на подписваните документи от масив

4.7.1. Запитване

HTTP параметри:

- HTTP метод: GET
- Request-URI: /status
- Други HTTP параметри се добавят от брауъра по спецификацията на HTTP/1.1 и

CORS.

Примерно запитване:

Request URL: https://localhost:53952/status?_=1497337337282
Request Method: GET
Status Code: ■ 200 / OK
Request Headers
Accept: application/json, text/javascript, */*; q=0.01
Accept-Encoding: gzip, deflate
Accept-Language: bg, en-US; q=0.7, en; q=0.3
Connection: Keep-Alive
Host: localhost:53952
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko
Response Headers
Accept: application/json, */*
Access-Control-Allow-Headers: Accept, Content-Type
Access-Control-Allow-Methods: GET, POST
Access-Control-Allow-Origin: *
Cache-Control: no-cache
Connection: keep-alive
Content-Length: 73
Content-Type: application/json
Date: Tue, 13 Jun 2017 07:02:57 GMT

4.7.2. Отговор

Отговорът е в JSON формат, подобен на примера показан по-долу:

`{"status":100,"statusCode":"PROCESSING","totalCount":20,"currentCount":1}`, където

Status	Описание	statusCode
- 100	Грешка	ERROR
0	Не е започнало изпълнение	NOT STARTED
100	В процес на изпълнение	PROCESSING
200	Обработката е приключила	DONE

- `totalCount`: общия брой елементи от масива за подписване
- `currentCount`: брой подписани до момента елементи от масива

5. Извикване на локалната услуга за електронно подписване

5.1. Javascript код за избиране на сертификат за подписване

```
function chooseSignerCertificate() {  
  
var port = $("#port").val();  
  
var request = {  
  selector: {}  
};  
  
console.log(request);  
  
if (keyusage!=null) {  
  request.selector.keyusages = new Array();  
  request.selector.keyusages.push(keyusage);  
}  
if (issuer!=null) {  
  request.selector.issuers = new Array();  
  request.selector.issuers.push(issuer);  
}  
if (thumbprint!=null){  
  request.selector.thumbprint=thumbprint;  
}  
  
var payload = null;  
var querystr = "";  
if (method=="POST") {  
  payload = JSON.stringify(request);  
} else {  
  querystr = "?version="+request.version;  
}  
  
var url = "https://localhost:" + port + "/getsigner" + querystr;
```

```
$.ajax({
  url: url,
  type: method,
  crossDomain: true,
  data: payload,
  contentType: "application/json",
  dataType: "json"
})
.done(function (response) {
  console.log(response);
  if (response.status=="ok") {
    var div = $("#result");
    $("#signerCertificate").html(response.chain[0])
    for (var i=0;i<response.chain.length;i++)
    div.append("<p><pre>Certificate#" + i + ":\n"+rowlize(response.chain[i])+"</pre></p>");
    response.content = data;
    response.contentType = type;
    validate(response);
  } else
  $("#result").html("<p><span class='label label-danger'>"+response.reasonCode+"</span> "+response.reasonText+"</p>");
  $("#sign").removeAttr("disabled");
})
.fail(function () {
  console.log("failed");
  $("#result").html("<p><b>Failed!</b> Most likely the SCS server is not running or browser's configuration policy does not allow cross origin requests. For instance, IE 10 and 11 do not allow requests to localhost unless the origin server is listed as trusted: see IE Settings > Internet Options > Security tab > click Trusted sites > click Sites buttons.</p>");
  $("#sign").removeAttr("disabled");
});
}
```

5.2. Javascript код за подписване

За извикване на BISS се използват показаните по - долу JavaScript кодове.

```
function sign(datas, sigalg) {

var port = $("#port").val();

$("#sign").attr("disabled", "disabled");

var request = {
  signerCertificateB64: $("#signerCertificate").html(),
  contents: datas,
  contentType: type,
  hashAlgorithm: sigalg,
  signatureType: "signature",
  version: "1.0"
};

console.log(request);
```

СПЕЦИФИКАЦИЯ
на софтуер за електронно подписване

```

/*
request.selector.validate = validation;

if (keyusage!=null) {
request.selector.keyusages = new Array();
request.selector.keyusages.push(keyusage);
}
if (issuer!=null) {
request.selector.issuers = new Array();
request.selector.issuers.push(issuer);
}
if (thumbprint!=null){
request.selector.thumbprint=thumbprint;
}
*/

var payload = null;
var querystr = "";
if (method=="POST") {
payload = JSON.stringify(request);
} else {
querystr = "?version="+request.version;
querystr += "&contentType="+escape(request.contentType);
querystr += "&hashAlgorithm="+escape(request.hashAlgorithm);
querystr += "&signatureType="+escape(request.signatureType);
querystr += "&content="+escape(request.content);
}

var url = "https://localhost:" + port + "/sign" + querystr;

$.ajax({
url: url,
type: method,
crossDomain: true,
data: payload,
contentType: "application/json",
dataType: "json"
})
.done(function (response) {
console.log(response);
if (response.status=="ok") {
var div = $("#result");
div.html("<p><b>"+response.reasonText+"</b> <font
id=\"signature_result\"><span class='label label-warning'>validating
signature on server...</span></font></p>");
for (step = 0; step < response.signatures.length; step++) {
div.append("<p><pre>Signature
("+response.signatureAlgorithm+"):\\n"+rowlize(response.signatures[step])+"<
/pre></p>");
}
//for (var i=0;i<response.chain.length;i++)
//
div.append("<p><pre>Certificate#"+i+":\\n"+rowlize(response.chain[i])+"</pre
></p>");
response.content = data;
response.contentType = type;
validate(response);
} else

```

СПЕЦИФИКАЦИЯ
на софтуер за електронно подписване

```
$("#result").html("<p><span class='label label-  
danger'>" + response.reasonCode + "</span> " + response.reasonText + "</p>");  
$("#sign").removeAttr("disabled");  
}  
.fail(function () {  
console.log("failed");  
$("#result").html("<p><b>Failed!</b> Most likely the SCS server is not  
running or browser's configuration policy does not allow cross origin  
requests. For instance, IE 10 and 11 do not allow requests to localhost  
unless the origin server is listed as trusted: see IE Settings > Internet  
Options > Security tab > click Trusted sites > click Sites buttons.</p>");  
$("#sign").removeAttr("disabled");  
});  
}
```