

# Sign Me API

## 1. Как пользоваться API

API предоставляет две возможности – подпись и проверка файла. В любом случае необходимо сформировать POST запрос. Все POST запросы необходимо отправлять с заголовком "Content-type": "application/x-www-form-urlencoded". **Для использования функций необходимо зарегистрировать свой сайт, с которого будет приходить запрос, в нашей системе и получить код доступа.** Далее подробнее о каждом:

### 1.1. Подпись файла

Необходимо сделать запрос к сайту подписи по url:

**https://server\_name/signapi/sjson**

В запросе необходимо отправить POST данными JSON-пакет формата:

```
{
  'filet' = base64_encode( содержимое файла ),
  'fname' = имя файла,
  'md5' = хеш содержимого для проверки, необязательно
  'key' = полученный ключ доступа,
  'url' = адрес возврата после подписания*, необязательно
  'user_email' = email пользователя, подписывающего файл, необязательно
  'user_ph' = телефон пользователя, подписывающего файл, необязательно
  'user_snils' = СНИЛС пользователя, подписывающего файл, необязательно
  'company_inn' = ИНН компании, подписывающей файл, необязательно
  'company_ogrn' = ОГРН компании, подписывающей файл, необязательно
  'poemail' = 1, если нужно не высылать пользователю емейл, необязательно
  'forcesms' = 1, если необходима двухфакторная авторизация, необязательно
  'IF' = 1, если необходимо уведомление на url-оповещения*** при каждом подписании**, необязательно
}
```

JSON пакет должен быть отправлен в виде строки:

```
rfile=json_encode(массив выше)
```

В ответ приходит добавочная часть url, которую нужно открыть в браузере или iframe текущего окна браузера в виде

**https://server\_name/signapi/sjson/добавочная\_часть**

По умолчанию *server\_name=sign.me* (можно делать запрос на <https://sign.me/> )

для того, чтобы пользователь ввел пароль для подписи файла, либо сообщение об ошибке.

При получении сообщения об ошибке не следует открывать с ним окно браузера.

Возможные сообщения об ошибке будут рассмотрены ниже.

\*Смотри пункт 7

\*\*На url будет осуществляться GET запрос с параметрами 'apikey': ваш api ключ, 'md5hash': md5 от содержимого файла, 'gosthash': ГОСТ 34.11 хеш от содержимого файла

\*\*\* url-оповещения задается один раз. Он сообщается сотрудникам Sign.Me

## 1.2. Проверка файла

Аналогично подписи, необходимо сделать POST запрос по url:

**https://server\_name/signaturecheck/json**

отправив JSON пакет:

```
{
    'file' = base64_encode( содержимое файла ), необязательно
    'md5' = md5( содержимое файла )
}
```

Отправляется строка

```
rfile=json_encode(массив выше)
```

В ответ приходит JSON пакет, содержащий массив, в котором есть вся необходимая информация о подписавших проверяемый файл пользователях:

```
{
    'count' = i,
    'signature_'i' = {
        'value' = base64_encode( значение подписи ),
        'pkcs7' = web_link (Ссылка на скачивание pkcs7-контейнера)
        'pkcs64' = base64_encode( pkcs7 контейнер электронной подписи)
        'sigdate' = дата подписи,
        'person' = {
            'public_key' = base64_encode( открытый ключ ),
            'key_id' = номер ключа,
            'crt_id' = номер сертификата,
            'crt_status' = 0 — действителен, 1 — ожидается, 2 — приостановлен, 3
— отозван, 4 — отсутствует, 5 — ошибка (чаще всего используются только 0 и 3),
            'cr_date' = дата создания dd.mm.yyyy,
            'exp_date' = дата истечения dd.mm.yyyy,
            'passport' = {
                'name' = имя,
                'lastname' = отчество,
                'surname' = фамилия,
            },
            'phone_number' = телефон,
            'email_address' = email,
        }
        'company' = { #если от имени компании
            'name' = название
            'inn' = ИНН
            'ogrn' = ОГРН
        }
    }
}
```

либо массив с 0 значением количества подписей, если файл не подписан:

```
{  
  'count' = 0,  
}
```

либо сообщение об ошибке.

### 1.3. Проверка файла через iframe

Необходимо сделать POST запрос на url

**https://server\_name/signaturecheck/request**

Достаточно прислать хеш файла, система вернет сгенерированный адрес, открыв который в iframe можно увидеть перечень подписавших этот файл.

Запрос должен быть следующего вида:

```
{  
  'key' = полученный ключ доступа,  
  'hash' = md5( содержимое файла )  
}
```

Отправляется строка

jsonrequest=json\_encode(*массив выше*)

В ответ приходит сообщение об ошибке, либо значение **UUID**, из которого необходимо сформировать адрес:

**https://server\_name/signaturecheck/result/UUID**

По этому адресу результат будет доступен для открытия только один раз.

Для повторного открытия необходимо повторить процедуру повторно.

По адресу **https://server\_name/signaturecheck/iframe/UUID** доступен javascript для отрисовки результата.

### 1.4. Быстрая проверка факта подписания запроса

Для проверки, подписан ли запрос из пункта **1.1**, необходимо открыть GET запросом адрес

**https://server\_name/signapi/check/добавочная\_часть\_1.1**

Ответ json, status: “0” - не подписан, “1” - подписан, “2” - отклонен; “comment” - причина отклонения, если есть, “3” - файл удален

### 1.5. Отказ от подписи запроса

Для отказа необходимо выполнить POST запрос на адрес

**https://server\_name/signapi/reject/**

С параметрами:

'rid' = *id запроса* (первая половина *добавочной\_части\_1.1*)

'key' = *полученный ключ доступа*,

'reason' = *текст причины отказа*

Ответ json:

```
'result': u'already_rejected', u'wrong_source', u'wrong_source_or_id', u'wrong_method', '0'  
'date': дата отказа (если already_rejected или 0)  
'reason': причина отказа (если already_rejected или 0)
```

## 2. Мультизапросы

Возможна отправка на подпись нескольких файлов сразу для одного адресата

### 2.1. Запрос на подпись

**https://server\_name/signapi/multijson/**

Аналогично одиночному запросу **1.1**, за исключением подачи файлов:

в корне json вместо filet, fname и md5 массив

```
files = [{  
    'filet' = base64_encode( содержимое файла ),  
    'fname' = имя файла,  
    'md5' = хеш содержимого для проверки, необязательно  
},  
{...}]
```

Запрос отправлять POST['mfile']=json\_encode(...)

В ответ придет **UUID** (uuid1, строка 8-4-4-4-12 символов)

Для открытия страницы подписания открыть url

**https://server\_name/signapi/multijson/UUID**

### 2.2. Проверка статуса

**https://server\_name/signapi/multicheck/UUID**

GET запрос, ответ аналогично одиночному **1.4**, за исключением

```
{'status'=4, 'comment'='Различные состояния отдельных файлов'}
```

(если, например, отдельно подписан один из файлов запроса)

### 2.3. Отказ

**https://server\_name/signapi/multireject/UUID**

POST запрос, аналогично **1.5**, но не нужен gid, вместо него UUID в url

Ответ json, вместо date – lastdate, reason – lastreason (для последнего файла из группы)

### 2.4. Проверка нескольких файлов

**https://server\_name/signaturecheck/multi**

POST запрос

```
{  
    'key' = полученный ключ доступа,  
    'files' = [{  
        'hash' = md5( содержимое файла )    }]
```

'name' = имя файла, необязательно  
,{...}]

}

Ответ **UUID** (uuid1, строка 8-4-4-4-12 символов)

Результаты проверки по адресам

**https://server\_name/signaturecheck/result/UUID/1**

страница в дизайне signme

**https://server\_name/signaturecheck/multiiframe/UUID**

javascript для отрисовки iframe

### 3. Получение нескольких подписей в одном файле

К любому подписанному файлу можно будет выгрузить все подписи в одном контейнере по ссылке вида

**https://server\_name/signaturecheck/get\_grp\_pkcs/hash/[[md5hashoffile]]**

Пример:

**https://sign.me/signaturecheck/get\_grp\_pkcs/hash/B87F869D15B0FDC4C92A414FFBF5EA7**

### 4. Сообщения об ошибках при подписи

error 1: no file

Означает, что после раскодирования JSON пакета в нем не обнаружено элемента массива с ключом file

error 2: no filename

Означает, что после раскодирования JSON пакета в нем не обнаружено элемента массива с ключом fname

error 3: wrong file

Означает, что пришедшее в массиве содержимое по ключу file не получается раскодировать base64\_decode

error 4: wrong json

Означает, что не получается раскодировать JSON пакет, пришедший в POST запросе по ключу rfile

error 5: cant create file

Не получается записать файл на сервер, возможно проблема в имени файла, иначе свяжитесь с разработчиками

error 6: cant create request

Не получается создать запрос на подпись файла на сервере – в любом случае свяжитесь с разработчиками

Добавлено уточнение:

error 6: wrong user phone или wrong user email – пользователь с таким номером телефона/email не зарегистрирован или не активирован

error 6: wrong company inn или wrong company ogrn – компания с таким ИНН/ОГРН не зарегистрирована или не активирована

error 7: get api key

Апи ключ отсутствует или не найден в разрешенных

## 5. Сообщения об ошибках при проверке

error 1: wrong base64

Означает, что пришедшее в массиве содержимое по ключу filet не получается раскодировать base64\_decode

error 2: wrong md5

Не получается вычислить md5 от раскодированного файла на стороне сервера, свяжитесь с разработчиками

error 3: wrong file

Означает, что пришедшее в массиве содержимое по ключу filet не получается раскодировать base64\_decode

error 4: wrong hash

Означает, что пришедшее в пакете значение md5 не совпадает с вычисленным на сервере

error 5: no hash

Означает, что значение md5 не пришло на сторону сервера

error 6: wrong json

Означает, что не получается раскодировать JSON пакет, пришедший в POST запросе по ключу rfile

## 6. Сообщения об ошибках при запросе iframe

error 1: wrong json

Означает, что не получается раскодировать JSON пакет, пришедший в POST запросе по ключу rfile

error 2: wrong key

Означает, что присланное значение в поле key не соответствует заданному для сайта, с которого пришел запрос, в системе, либо сайт не зарегистрирован

error 3: wrong hash

Означает, что md5 прислан в неправильном формате (не соответствует регулярному выражению a-zA-Z0-9, содержит недопустимые символы)

error 4: request not created

Не получается создать запрос на проверку файла на сервере – в любом случае свяжитесь с разработчиками

*\*Во всех действиях ошибка error 0: wrong method означает, что необходимо использовать POST запрос*

## 7. Возврат на url

Если указан url, то после подписи произойдет возврат на этот url + добавочная часть вида ?signed=true или ?signed=false&error=%errno% (в случае, если в url уже есть символ ? часть будет начинаться с &)

errno:

1 - пароль неправильный

2 - смс код неправильный

3 - пользователь не имеет права подписывать

4 - в компании стоит запрет первой подписи

5 - запрос отклонен

6 - внутренняя ошибка сервера, запишите точное время ошибки

7 - попытка подписи от имени другой компании

Если url начинается с символов GET: (4 символа дословно), а далее идет сам url, например:

GET:<https://sandbox.sign.me/push>

**то произойдет curl опрос этого url с нашего сервера по тем же правилам (добавочная часть), но без переадресаций и открытия страниц. Это можно использовать для уведомлений при подписании с мобильного.**