

MICROSEC ZRT.



# MicroSigner Közvetítő Szerver fejlesztői dokumentáció

---

verzió: 1.0

Ivicsics Sándor, Máté Norbert, Vanczák Gergely

2016.06.09.

## Tartalom

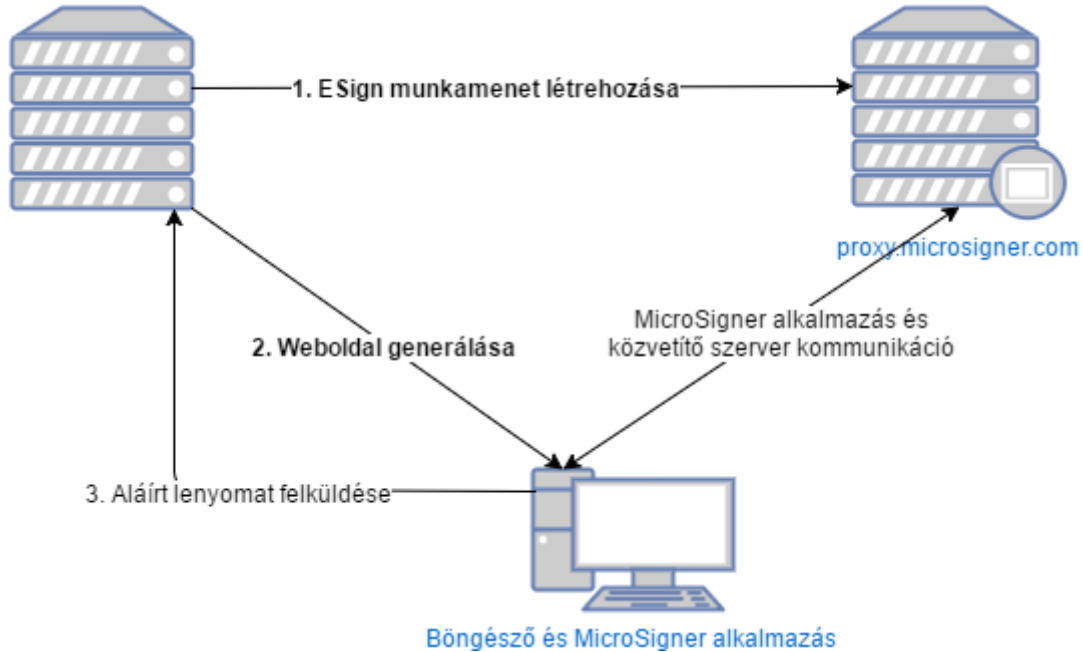
Általános információk.....	2
ESign munkamenet létrehozása .....	2
Elvégzendő programozási feladatok .....	2
Példa PHP-ban .....	3
Weboldal generálása.....	4
Elvégzendő programozási feladatok .....	4
Példa.....	5
Aláírt lenyomatok beillesztése .....	5
Elvégzendő programozási feladatok .....	5
Példa PHP-ban .....	5
Az ESign alapértelmezett viselkedésének megváltoztatása a weboldalon .....	7
onConfigAJAXFailure(jqXHR, textStatus).....	7
onConfigNotGot(counter) .....	7
onConfigGot() .....	7
onSignedHashesAJAXFailure(jqXHR, textStatus).....	7
onSignedHashesESignFailure(status) .....	7
onSignedHashesNotGot(counter) .....	7
Példa (JavaScript) .....	7

## Általános információk

A MicroSigner alkalmazás aktuális verziója az alábbi linkről tölthető le:

[https://proxy.microsigner.com/client/microsigner\\_setup.msi](https://proxy.microsigner.com/client/microsigner_setup.msi)

Informatív ábra a kialakítandó rendszer működéséről:



A kommunikáció a közvetítő szerverrel JSON formátumban történik HTTPS protokoll felett TLSv1-et használva.

## ESign munkamenet létrehozása

### Elvégzendő programozási feladatok

JSON objektum létrehozása az alábbi mezőkkel:

**mode:** A MicroSigner alkalmazás elvárt működési módja. Lehetséges értékek:

SELECT\_CERTIFICATE\_AND\_SIGN: tanúsítvány kiválasztása és lenyomatok aláírása (ez a dokumentum nem tér ki ennek a működési módnak a használtára)

SELECT\_CERTIFICATE: csak tanúsítvány kiválasztása (ez a dokumentum nem tér ki ennek a működési módnak a használtára)

SIGN: csak lenyomatok aláírása (a továbbiakban ezt mutatjuk be példák)

**SPName:** szolgáltató neve (szabad szöveges mező); a MicroSigner alkalmazás ablakának Szolgáltató mezőjében megjelenő információ

**message:** üzenet (szabad szöveges mező); a MicroSigner alkalmazás ablakának Üzenet mezőjében megjelenő információ

**certificate:** base64 kódolt tanúsítvány

**hashAlgorithm:** a lenyomatok típusát meghatározó OID

sha1 lenyomat aláírása esetén: 1.3.14.3.2.26

sha256 lenyomat aláírása esetén: 2.16.840.1.101.3.4.2.1

sha384 lenyomat aláírása esetén: 2.16.840.1.101.3.4.2.2

sha512 lenyomat aláírása esetén: 2.16.840.1.101.3.4.2.3

SSL/TLS kapcsolat kiépítésének folyamatában keletkező adat aláírása esetén üres sztringet kell megadni

**documents:** dokumentumok; JSON objektumok tömbje; a tömbben egy JSON objektumnak az alábbi mezőket kell tartalmaznia:

**name:** dokumentum neve

**unsignedHash:** base64 kódolt aláíratlan lenyomat

Sztringgé konvertált JSON objektum PKCS#7 formátumú aláírása. **Az aláíráshoz szükséges aláíró tanúsítványt a Microsec-től kell igényelni!**

PKCS#7 formátumú aláírás felküldése a közvetítő szerverre:

**URL:** <https://proxy.microsigner.com/esign/newSigningSession>

**Metódu:** POST

**Hitelesítés módja:** felhasználónév/jelszó

**Elvárt Content-Type:** application/x-pkcs7-mime

A szerver válasza egy JSON objektum, ha a HTTP státuszkód értéke 200 vagy 420.

A JSON objektum az alábbi mezőket tartalmazza 200-as HTTP státuszkód esetén:

**sessionUrl:** munkamenet általános URL-je

**sessionId:** munkamenet azonosítója

**mode:** működési mód

**documentIds:** dokumentumok azonosítói

A JSON objektum az alábbi mezőket tartalmazza 420-as HTTP státuszkód esetén:

**code:** kód; lehetséges értékek:

ERROR: a művelet végrehajtása közben hiba történt

**message:** üzenet

A szerver válaszolhat 500-as HTTP státuszkóddal is, ha a HTTP kérés feldolgozása során ismeretlen hiba történt.

## Példa PHP-ban

```
// Aláíratlan lenyomat(ok) előállítás
```

```
file_put_contents('signer_certificate.cer', $_POST['certificate']);
```

```
$last_line = exec("eszigno3 pdf_sign -in input.pdf -out half_signed.pdf  
-signer_cert signer_certificate.cer");
```

```
$last_line = exec("eszigno3 pdf_get_unsigned_hash -in half_signed.pdf -  
out unsigned_hash_base64_encoded.txt");
```

```
$unsignedHash = file_get_contents('unsigned_hash_base64_encoded.txt');
```

```
// JSON objektum létrehozása
```

```
$new_signing_session_request = array(
```

```
    'mode' => 'SELECT_CERTIFICATE_AND_SIGN',  
    'SPName' => 'Test',  
    'message' => 'Test',  
    'certificate' => $_POST['certificate'],  
    'hashAlgorithm' => '2.16.840.1.101.3.4.2.1',  
    'documents' => array(  
        0 => array(  
            'name' => 'Test.pdf',  
            'unsignedHash' => $unsignedHash  
        )  
    )  
);
```

```

    )
  );
  $new_signing_session_request_str =
  json encode($new_signing_session_request);

  // Karakterfüzérre konvertált JSON objektum PKCS#7 formátumú aláírása

  file_put_contents('tmp.json', $new_signing_session_request_str);

  openssl_pkcs7_sign('tmp.json', 'tmp.p7m',
  'signing_certificate_and_private_key.pem',
  array('file://signing_certificate_and_private_key.pem', '<passphrase>'));

  $data_to_post_str = file_get_contents('tmp.p7m');

  // PKCS#7 formátumú aláírás felküldése

  $ch = curl_init();

  curl_setopt($ch, CURLOPT_URL,
  'https://proxy.microsigner.com/esign/setUnsignedHashes');

  curl_setopt($ch, CURLOPT_USERPWD, '<username>:<password>');
  curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'Content-Type: application/pkcs7-signature',
    'Content-Length: ' . strlen($data_to_post_str)
  ));
  curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
  curl_setopt($ch, CURLOPT_POSTFIELDS, $data_to_post_str);
  curl_setopt($ch, CURLOPT_SSLVERSION, 1); // CURL_SSLVERSION_TLSv1
  curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

  $response_str = curl_exec($ch);
  curl_close($ch);

  $response = json_decode($response_str);
  $sessionId = $response->sessionId;
  $documentId = $response->documentIds[0];

```

## Weboldal generálása

### Elvégzendő programozási feladatok

jQuery és ESign javascript library-k meghívkozása

ESign objektum létrehozása, callback-ek megírása

**onSuccessGetSignedHashes:** ez a függvény fog meghívódni, ha a felhasználó aláírta az aláíratlan lenyomatokat; a függvénynek egyetlen paramétere az base64 kódolt aláírt lenyomatok tömbje

A közvetítő szerver válaszának elhelyezése egy javascript változóban, hogy paraméterként átadható legyen az ESign objektum sign metódusának

Hidden IFRAME és A tag-ek elhelyezése

Az ESign objektum sign metódusának meghívása (például egy BUTTON tag onclick eseménykezelőjének segítségével)

## Példa

```
<!DOCTYPE html>
<html>
  <head>
    <title>Test</title>
    <!-- jQuery javascript library meghivatkozása -->
    <script src="/jquery.js"></script>
    <!-- ESign javascript library meghivatkozása -->
    <script
src="https://proxy.microsigner.com/esign/js/esign.js"></script>
    <script>
esign_log = true;
// ESign objektum létrehozása
var esign = new ESign({'baseUrl':'https://proxy.microsigner.com/esign'});
// A közvetítő szerver válaszáának elhelyezése egy javascript változóban
// Az itt szereplő JSON sztring a PHP példában a $response_str változó
értéke
var newSigningSessionResponse =
{"sessionId":"zgp6zvlgxj89", "mode":" SIGN", "documentIds":["85obxs7ev961"]};
// callback-ek
function testOnSuccessGetSignedHashes(signedHashes) {
  sendSignedHash(signedHashes[0]);
}
    </script>
  </head>
  <body>
    <h1>Test</h1>
    <!-- Hidden IFRAME és A tag-ek elhelyezése -->
    <iframe name="iframe_esign" style="display: none;"></iframe>
    <a target="iframe_esign" href="javascript:void(0)" id="a_esign"></a>
    <!-- Az ESign objektum sign metódusának meghívása (például egy BUTTON tag
onclick eseménykezelőjének segítségével) -->
    <button onclick="esign.sign(newSigningSessionResponse, 'a_esign',
testOnSuccessGetSignedHashes);">Sign</button>
  </body>
</html>
```

## Aláírt lenyomatok beillesztése

### Elvégzendő programozási feladatok

Aláírt lenyomatok beillesztése

### Példa PHP-ban

```
// Aláírt lenyomatok beillesztése

$last_line = exec("eszigno3 pdf_set_signed_hash -in half_signed.pdf -
out signed.pdf -hash '" . $_POST['signedHash'] . "'");
```

**Figyelem!** A fenti példa parancs beszúrási támadásra ad lehetőséget, ezért a kliens által felküldött paramétert validálni kell!

Bővebb információ:

[https://www.owasp.org/index.php/Command\\_Injection](https://www.owasp.org/index.php/Command_Injection)

## Az ESign alapértelmezett viselkedésének megváltoztatása a weboldalon

Lehetőség van megváltoztatni az ESign alapértelmezett viselkedését a weboldalon bizonyos függvényeinek felüldefiniálásával.

### **onConfigAJAXFailure(jqXHR, textStatus)**

Akkor hívódik meg, ha kapcsolódási hiba történt annak ellenőrzése közben, hogy a MicroSigner alkalmazás elindult-e.

### **onConfigNotGot(counter)**

Alapértelmezetten 1000 ms-ként meghívódik, amíg a MicroSigner alkalmazás el nem indul. A counter paraméter tartalmazza, hogy eddig hányszor hívódott meg a függvény. A függvénynek false-szal vagy true-val kell visszatérnie. Ha a függvény false-szal tér vissza, akkor a folyamat megszakításra kerül.

### **onConfigGot()**

Akkor hívódik meg, ha a MicroSigner alkalmazás elindult.

### **onSignedHashesAJAXFailure(jqXHR, textStatus)**

Akkor hívódik meg, ha kapcsolódási hiba történt annak ellenőrzése közben, hogy a lenyomatok aláírásra kerültek-e.

### **onSignedHashesESignFailure(status)**

Akkor hívódik meg, ha a lenyomatok aláírása során felhasználói megszakítás vagy hiba történt. A paraméterként kapott status objektum code és message mezőinek segítségével további információhoz juthatunk. Pl.: ha status.code értéke CANCEL, akkor a felhasználó kérésére szakadt meg a folyamat.

### **onSignedHashesNotGot(counter)**

Alapértelmezetten 1000 ms-ként meghívódik, amíg a lenyomatok aláírásra nem kerülnek. A counter paraméter tartalmazza, hogy eddig hányszor hívódott meg a függvény. A függvénynek false-szal vagy true-val kell visszatérnie. Ha a függvény false-szal tér vissza, akkor a folyamat megszakításra kerül.

## Példa (JavaScript)

```
esign.onConfigAJAXFailure = esign.onSignedHashesAJAXFailure = function (jqXHR,
textStatus) {
    removeProgressbarWithIndeterminateValue();
    showModalDialog('AJAX error: ' + textStatus);
}
```



```
esign.onSignedHashesESignFailure = function (status) {
    removeProgressbarWithIndeterminateValue();
    if (status.code !== 'CANCEL') {
        showModalDialog('ESign error!\nCode: ' + status.code + '\nMessage: ' +
status.message);
    }
}

esign.onConfigNotGotDefault = function(counter) {
    if (counter % 30 === 0) {
        if (confirm('It seems, that e-Sign application did not start. Would you like
to wait more?')) {
            return true;
        }
        else {
            removeProgressbarWithIndeterminateValue();
            return false;
        }
    }
    return true;
}

esign.onSignedHashesNotGotDefault = function(counter) {
    if (counter % 30 === 0) {
        if (confirm("It seems, that no hashes were signed. Would you like to wait
more?")) {
            return true;
        }
        else {
            removeProgressbarWithIndeterminateValue();
            return false;
        }
    }
    return true;
}
```