

# Реализация аутентификации по 3-D Secure 2.2 при помощи SDK (комплект для разработки ПО)

Корректное прохождение аутентификации по 3-D Secure 2.2 на странице предприятия можно реализовать при помощи SDK.

3DSWebSDK представляет собой JavaScript-код, позволяющий отправлять сообщения *3DS Method* и *Challenge Request* для аутентификаций через браузер плательщика.

- [Установка Web SDK](#)
- [Техническая документация](#)
- [Пример вызова init3DSMethod](#)
- [Пример вызова init3DSMethod с обработкой тайм-аута](#)
- [Пример вызова init3DSChallengeRequest](#)
- [Пример вызова init3DSChallengeRequest с обработкой тайм-аута](#)

## Установка Web SDK

Для установки 3DSWebSDK необходимо импортировать готовый [JavaScript](#) в HTML-страницу предприятия.

**Installation**

```
<!DOCTYPE html>
<html>
  <head>
    ...
    <script src="nca-3ds-web-sdk.js" type="text/javascript" />
    ...
  </head>
  <body>
    ...
  </body>
</html>
```

Скрипт прикрепит 3DSWebSDK к объекту окна браузера в JavaScript.

3DSWebSDK поддерживает следующие операции:

| Операция                               | Описание                                                                                                                                                                                                                                                                                                                                                                                                             |
|----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| init3DSMethod                          | Создает HTML-структуру, прикрепляет форму с одним полем ввода ( <i>threeDSMethodData</i> ) и автоматически отправляет его по адресу <i>threeDSMethodUrl</i> .                                                                                                                                                                                                                                                        |
| createIframeAndInit3DSMethod           | Создает iFrame со структурой HTML, прикрепляет к нему форму с одним полем ввода ( <i>threeDSMethodData</i> ), автоматически отправляет ее по адресу <i>threeDSMethodUrl</i> и прикрепляет iFrame к контейнеру. В случае необходимости, при загрузке iFrame будет выполнен callback. Кроме того, можно указать тайм-аут ожидания и callback для обработки случаев, когда выполнение 3DSMethod не завершается вовремя. |
| init3DSChallengeRequest                | Создает HTML-структуру, прикрепляет форму с одним полем ввода ( <i>creq</i> ) и автоматически отправляет его по адресу <i>acsUrl</i> .                                                                                                                                                                                                                                                                               |
| createIframeAndInit3DSChallengeRequest | Создает iFrame со структурой HTML, прикрепляет форму с одним полем ввода ( <i>creq</i> ), автоматически отправляет ее по адресу <i>acsUrl</i> и прикрепляет iFrame к контейнеру. Если указано, при загрузке iFrame будет выполнен callback. Дополнительно можно указать тайм-аут ожидания и callback для обработки случаев, когда страница ACS не загружена вовремя.                                                 |
| getBrowserData                         | Скрипт для сбора данных о браузере и устройстве плательщика, может быть использован на начальном этапе для заполнения соответствующих параметров в запросе на оплату через интерфейс silentpay.                                                                                                                                                                                                                      |

## Техническая документация

**Documentation**

```
/**
 * Attach all methods to window.
 */
let nca3DSWebSDK = {};
```

```

/**
 * Creates an iframe, attach it to the rootContainer and submit 3DS Method form.
 *
 * @param threeDSMethodUrl - a FQDN endpoint to submit the 3DS Method request
 * @param threeDSMethodData - Base64-encoded 3DS Method Data value
 * @param frameName - name of the frame container. if not set it will be set to 'threeDSMethodIFrame'
 * @param rootContainer - the container where the iframe will be attached to.
 *                        If not set defaults to the JavaScript document.body object
 * @param onFrameLoadCallback - callback function attached to the iframe.onload event
 * @param timeoutSeconds - timeout in seconds (default: 0 = no timeout)
 * @param onFrameTimeoutCallback - callback function called when timeout occurs
 * @throws {Error} - throws error if there is a validation error
 * @returns {HTMLIFrameElement} - returns the generated iframe element
 */
nca3DSWebSDK.createIframeAndInit3DSMethod = createIframeAndInit3DSMethod;

/**
 * Initiates a 3DS Method request and submits the form the 3DS Method URL. It will automatically hide the
 * container
 * when initiating a 3DS Method request.
 *
 * @param threeDSMethodUrl - a FQDN endpoint to submit the 3DS Method request
 * @param threeDSMethodData - Base64-encoded 3DS Method Data value.
 * @param container - the iframe container where the form will be attached to. The container must have the
 * 'name'
 * attribute set
 * @throws {Error} - throws error if there is a validation error
 * @returns {HTMLIFrameElement} - the container
 */
nca3DSWebSDK.init3DSMethod = init3DSMethod;

/**
 * Initiates a 3DS Challenge request and submits the form the ACS URL.
 *
 * @param acsUrl - the FQDN URL to submit the Challenge Request
 * @param creqData - Base64-encoded Challenge Request
 * @param container - the iframe container where the form will be attached to. The container must have the
 * 'name'
 * attribute set
 * @throws {Error} - throws error if there is a validation error
 * @returns {HTMLIFrameElement} - the container
 */
nca3DSWebSDK.init3DSChallengeRequest = init3DSChallengeRequest;

/**
 * Creates an iframe, attach it to the rootContainer and submits 3DS Challenge Request.
 * @param acsUrl - the FQDN URL to submit the Challenge Request
 * @param creqData - Base64-encoded Challenge Request
 * @param challengeWindowSize - EMVCo assigned window size.
 *
 * '01' -> 250px x 400px,
 * '02' -> 390px x 400px,
 * '03' -> 500px x 600px,
 * '04' -> 600px x 400px,
 * '05' -> Full screen, or full container content
 * @param frameName - name of the frame container. if not set it will be set to 'threeDSCReqIFrame'
 * @param rootContainer - the container where the iframe will be attached to.
 *                        If not set defaults to the JavaScript document.body object
 * @param onFrameLoadCallback - callback function attached to the iframe.onload event
 * @param timeoutSeconds - timeout in seconds (default: 0 = no timeout)
 * @param onFrameTimeoutCallback - callback function called when timeout occurs
 * @throws {Error} - throws error if there is a validation error
 * @returns {HTMLIFrameElement} - returns the generated iframe element
 */
nca3DSWebSDK.createIframeAndInit3DSChallengeRequest = createIframeAndInit3DSChallengeRequest;

/**
 * Collects browser-specific data.
 *
 * @returns {Object} An object containing various properties about the user's browser and device.

```

```

* @property {boolean} browserJavaEnabled - Whether Java is enabled in the browser.
* @property {boolean} browserJavaScriptEnabled - Always true; indicates JavaScript is functioning.
* @property {string} browserLanguage - The browser's preferred language setting (e.g., 'en-US').
* @property {number} browserColorDepth - Screen color depth in bits per pixel.
* @property {number} browserScreenHeight - Total height of the screen in pixels.
* @property {number} browserScreenWidth - Total width of the screen in pixels.
* @property {string} browserTZ - Time zone of the browser in IANA format (e.g., 'America/New_York').
* @property {string} browserUserAgent - The full User-Agent string of the browser.
*/
nca3DSWebSDK.getBrowserData = getBrowserData;
window.nca3DSWebSDK = nca3DSWebSDK;

```

## Пример вызова *init3DSMethod*

### init3DSMethod

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <script src="nca-3ds-web-sdk.js" type="text/javascript"></script>
  <title>Init 3DS Method - Example</title>
</head>
<body>
<!-- The first example shows how to generate a form and attach to an already defined iframe -->
<iframe name="iframeContainer" id="iframe"></iframe>

<!-- In the second example it will generate an iframe and attach to the #frameContainer element -->
<div id="frameContainer"></div>

<script type="text/javascript">
  iframe = document.getElementById('iframe');
  frameContainer = document.getElementById('frameContainer');
  nca3DSWebSDK.init3DSMethod('http://example.com', 'base64-encoded-3dsmethod-request', iframe);

  nca3DSWebSDK.createIframeAndInit3DSMethod(
    'http://example.com',
    'base64-encoded-3dsmethod-request',
    'threeDSMethodIFrameContainer',
    frameContainer,
    () => {
      console.log('Iframe loaded, form created and submitted');
    }
  );
</script>

</body>
</html>

```

## Пример вызова *init3DSMethod* с обработкой тайм-аута

### init3DSMethod with Timeout

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <script src="nca-3ds-web-sdk.js" type="text/javascript"></script>
  <title>Init 3DS Method with Timeout - Example</title>
</head>
<body>
<div id="methodFrameWithTimeout"></div>

<script type="text/javascript">
  const methodContainer = document.getElementById('methodFrameWithTimeout');

  // 3DS Method with 10 second timeout
  nca3DSWebSDK.createIframeAndInit3DSMethod(
    'http://example.com/3dsmethod',
    'base64-encoded-3dsmethod-request',
    'methodIFrameTimeout',
    methodContainer,
    () => {
      console.log('3DS Method completed successfully');
    },
    10, // 10 second timeout
    () => {
      console.log('3DS Method timed out after 10 seconds');
    }
  );
</script>

</body>
</html>
```

### Пример вызова *init3DSChallengeRequest*

### init3DSChallengeRequest

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <script src="nca-3ds-web-sdk.js" type="text/javascript"></script>
  <title>Init 3DS Challenge Request - Example</title>
</head>
<body>
<!-- This example will show how to initiate Challenge Reqeuests for different window sizes. -->
<div id="frameContainer05"></div>

<script type="text/javascript">
  container05 = document.getElementById('frameContainer05');

  nca3DSWebSDK.createIFrameAndInit3DSChallengeRequest(
    'http://localhost:8080/acs/challenge',
    'base64-encoded-challenge-request',
    '05',
    'threeDSCReq05',
    container05,
    () => {
      console.log('Iframe loaded, form created and submitted');
    }
  );
</script>

</body>
</html>
```

**Пример вызова *init3DSChallengeRequest* с обработкой тайм-аута**

### init3DSChallengeRequest with Timeout

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <script src="nca-3ds-web-sdk.js" type="text/javascript"></script>
  <title>Init 3DS Challenge Request with Timeout - Example</title>
</head>
<body>
<div id="challengeFrameWithTimeout"></div>

<script type="text/javascript">
  const challengeContainer = document.getElementById('challengeFrameWithTimeout');

  // 3DS Challenge with 60 second timeout
  nca3DSWebSDK.createIFrameAndInit3DSChallengeRequest(
    'http://localhost:8080/acs/challenge',
    'base64-encoded-challenge-request',
    '03',
    'challengeIFrameTimeout',
    challengeContainer,
    () => {
      console.log('Challenge authentication completed successfully');
    },
    60, // 60 second timeout
    () => {
      console.log('Challenge authentication timed out after 60 seconds');
    }
  );
</script>

</body>
</html>
```