



# Cyberus Key Implementation QuickStart

Version 1.0 01.2020

## Table of Contents

<b>Introduction</b>	<b>2</b>
<b>Requirements</b>	<b>2</b>
<b>Integration steps</b>	<b>3</b>
Obtain Cyberus Key client credentials	3
Create a redirection	3
Embed the widget	3
Via package manager	3
NPM	3
Yarn	3
Via JSDelivr CDN	4
Handle the authentication result	4
<b>Testing / troubleshooting</b>	<b>6</b>
<b>Next steps</b>	<b>6</b>



## Introduction

The goal of this document is to provide information to our partners and clients to quickly integrate Cyberus Key with their system.

Cyberus Key eliminates passwords - the weakest link in cybersecurity nowadays. Our authentication protocol uses One-Time Passwords (OTP) transmitted by ultrasonic signal using an Out-of-Band connection.

The system enables users to log in to websites, mobile applications, execute transactions and exchange information with fully authenticated parties without User IDs or Passwords. Users benefit from fast, password-free authentication while eliminating fraud through direct authentication using one-time transaction tokens.

## Requirements

- Computer
  - Internet browser - Chrome, Firefox, Safari, Edge
  - Speakers - any speakers that cover the 20 Hz - 20 kHz band (the ultrasonic signal is at 17 kHz)
- Mobile device
  - Android version 4.3 (API level 18) or newer or iOS 8.0 or newer
  - Cyberus Key app ([Apple App Store](#), [Google Play](#))
- Web Server
  - Any server software that can receive and send web requests will work with Cyberus.
  - Server will need to be hosted with a publicly-reachable URL.

## Integration steps

### Obtain Cyberus Key client credentials

We will supply you with a Client ID, Client secret, Cyberus's public key (used for decoding JWT tokens) and a client account (with an email and password). We will also create transaction types based on your requirements.

We will also register one or more redirect URLs for the user to be redirected to after a successful login. So if your domain is www.mysite.com, the redirect URL could be www.mysite.com/authentication/success.

### Embed the widget

Using the Widget Javascript library (<https://www.npmjs.com/package/cyberuskey-widget>), create a new CyberusKeyWidget instance.

#### Via package manager

##### NPM

```
npm install cyberuskey-widget
```

##### Yarn

```
yarn add cyberuskey-widget
```

Then embed the button on your web page:

```
import { CyberusKeyWidget, HTML5GeoProvider } from "cyberuskey-widget";

$(document).ready(() => {
  const cyberusKeyButton = new CyberusKeyWidget({
    geoProvider: new HTML5GeoProvider(),
    // the client ID supplied to you
    clientId: 'CLIENT_ID',
    // the redirect URI registered in the previous step
    redirectUri: 'REDIRECT_URI'
  });
  // create the button, using the ID of the element that will contain it
  cyberusKeyButton.create('BUTTON_CONTAINER_ID');
});
```



## Via JSDelivr CDN

```
<script src="https://cdn.jsdelivr.net/npm/cyberuskey-widget@0.0.12/dist/widget.min.js"
type="text/javascript"></script>

<script type="text/javascript">
  const CyberusKeyWidgetNamespace = window['cyberuskey-widget'];
  const CyberusKeyWidget = CyberusKeyWidgetNamespace.CyberusKeyWidget;
  const HTML5GeoProvider = CyberusKeyWidgetNamespace.HTML5GeoProvider;

  $(document).ready(() => {
    const cyberusKeyButton = new CyberusKeyWidget({
      geoProvider: new HTML5GeoProvider(),
      // the client ID supplied to you
      clientId: 'CLIENT_ID',
      // the redirect URI registered in the previous step
      redirectUri: 'REDIRECT_URI'
    });
    // create the button, using the ID of the element that will contain it
    cyberusKeyButton.create('BUTTON_CONTAINER_ID');
  });
</script>
```

For more options, and full explanations of all parameters, see the [JS SDK documentation](#).

## Handle the authentication result

After the user authentication process finishes, the user will be redirected to your web server (specifically, the page you registered as a redirect URI). You can use the query parameters to determine the authentication result and load the user's info.

Example code is in Python and uses the Tornado web framework, but can be rewritten to use any language or framework.

```
class AuthorizationCallbackHandler(tornado.web.RequestHandler):
    def get(self):
        # check for errors
        error = self.get_argument('error', None)
        error_description = self.get_argument('error_description', None)
        if error:
            url=f'LOGIN_ERROR_PAGE_URL?error={error}&error_description={error_description}'
            self.redirect(url)
            return

        # exchange authorization code for token
        data = {
```

```

    'grant_type': 'authorization_code',
    'code': self.get_argument('code'),
    'redirect_uri': 'REDIRECT_URI'
}
token=base64.b64encode(str.encode("CLIENT_ID:CLIENT_SECRET"))
headers = {
    'Authorization': f'Basic {token.decode("utf-8")}'
}
url = 'https://production-api.cyberuskey.com/api/v2/tokens'
token_response = requests.post(url, data=data, headers=headers)
token_body = token_response.json()

# decode token
id_token = token_body['id_token']
access_token = token_body['access_token']
id_data = jwt.decode(id_token, CYBERUS_PUBLIC_KEY)

# validate token
if id_data.get('iss') != CLIENT_DOMAIN or id_data.get('aud') != CLIENT_ID:
    self.redirect(f'LOGIN_ERROR_PAGE_URL?error=Unauthorized')
    return

# save user data to session
session_id = self.get_cookie(options.cookie_name)
session = self._session_store.load_session(session_id)
session['user'] = {
    'first_name': id_data.get('given_name'),
    'last_name': id_data.get('family_name'),
    'email': id_data.get('email'),
    'openid_identifier': id_data.get('sub'),
    'exp': id_data.get('exp'),
    'access_token': access_token
}
self._session_store.save_session(session)

# redirect to success page
self.redirect('LOGIN_SUCCESS_PAGE_URL')
```

See the [full integration documentation](#) for further details on authentication options and parameters.

## Testing

1. Open your web page with the embedded widget.
2. Open the CyberusKey app and go to the “Listen” tab.
3. Click on the CyberusKey widget.
4. App should show the success message, and the web page should be redirected to your login success page.

## Troubleshooting

- Widget doesn't do anything when clicked
  - Check that your Client ID is correct (don't use your client secret for the widget!)
  - Check that your redirect URL matches the one you registered exactly.
- Widget does not play audio
  - Make sure your speakers are on and volume is up

## Next steps

- Integrate your users with Cyberus Key users - see full documentation
- Ensure all security settings are correct - see full documentation