**Set up the form**

Create your login form with the elements you want to use.

In this case, we've added:

* email-or-username - the user's email address or username
* password - required for login with password

**Pass form data to Userfront.login()**

The [login()](https://userfront.com/docs/js#login-options) method allows you to pass in data to log in a user.

Our JavaScript needs to pass our form data into this method.

Userfront then does the following:

1. Verifies the user's credentials
2. Adds the user's access token as a cookie named access.demo1234
3. Redirects the page to the [After-login path](https://userfront.com/guide/glossary#after-login-path)

### **Example JavaScript**

In the example code here, we do the following:

1. Reference all the elements on the page
2. Define a custom formLogin() method that calls Userfront.login() with the form values
3. Add an event listener to call formLogin() when the form is submitted

### **Error handling**

Whenever the Userfront.login() method fails, we can catch its error in the promise chain.

This error contains a message property with what went wrong.

In this example, we use the setAlert() method to display the error message inside of our alert element

#### **Login after redirect**

Once the browser is redirected back to your login page after SSO approval, your application should call

Userfront.login({ method: "link" }).

You can set up your JS to call this method automatically by checking whether the URL contains the token and uuid parameters.

If your original SSO signup call contained a redirect parameter, it will be included in the URL and followed automatically