

Get Started

Hello World tutorial to help you quickly get familiarized with Magic.

Get Template Code

Let's get started by building a simple interface to let users log in or log out via an email magic link experience. We've created a template with necessary html and styles on CodeSandBox.

  Go to our [Hello World Template](#) to continue this tutorial!

If you don't want to go through the walkthrough, feel free to skip to the [completed implementation](#) or watch the demo video below.

Magic Hello World Demo



Walkthrough

Install Magic SDK

Install Magic SDK to your template by simply including a script tag after the comment line in the CodeSandBox editor. You can start by copy and pasting the code snippet below to your editor under the comment.

```
1 <!--  Install Magic SDK -->
2 <script src="https://cdn.jsdelivr.net/npm/magic-sdk/dist/magic.js"></script>
```

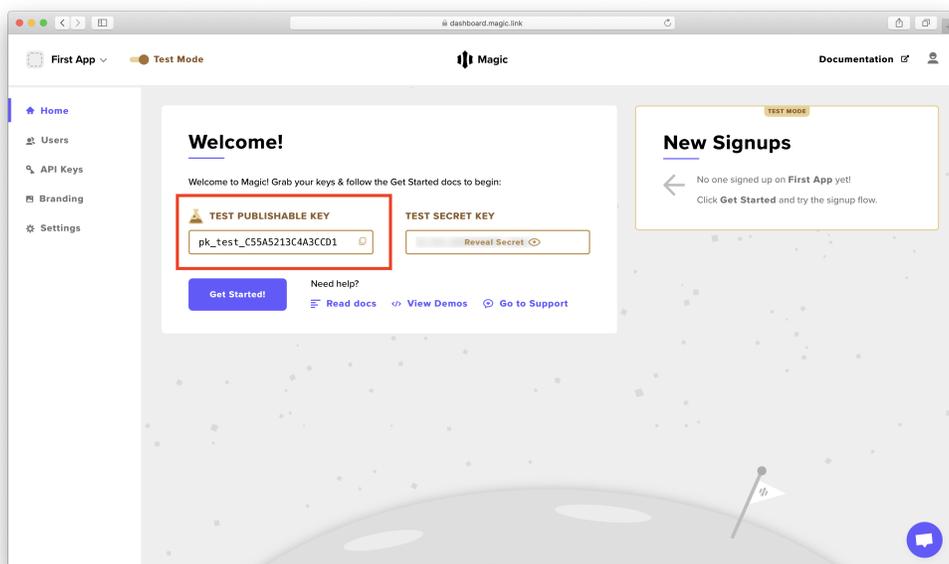
See other ways to [install Magic SDK](#) such as using yarn or npm.

2 Initialize Magic Instance

You can initialize a Magic instance with your Publishable API Key with the following code. Copy and paste this code snippet below to your editor under the comment.

```
1 /* 2 Initialize Magic Instance */
2 const magic = new Magic("YOUR_TEST_PUBLISHABLE_API_KEY");
```

You need to sign up or log in to the [Magic Dashboard](#) to view your API keys. Once you are logged in, you can then update the 'YOUR_TEST_PUBLISHABLE_API_KEY' string to your actual test *publishable* API key, which looks something like `pk_test_*****`.



3 Implement Render Function

Next, to complete the setup, let's write the logic on what the Hello World app should display when (1) users are not logged in, and when (2) users are logged in. Copy and paste this code snippet below to your editor under the comment.

```
1 /* 3 Implement Render Function */
2 const render = async () => {
3   const isLoggedIn = await magic.user.isLoggedIn();
4   /* Show login form if user is not logged in */
5   let html = `
6     <h1>Please sign up or login</h1>
7     <form onsubmit="handleLogin(event)">
8       <input type="email" name="email" required="required" placeholder="Enter your
9       <button type="submit">Send</button>
10    </form>
11  `;
12   if (isLoggedIn) {
13     /* Get user metadata including email */
14     const userMetadata = await magic.user.getMetadata();
15     html = `
16       <h1>Current user: ${userMetadata.email}</h1>
17       <button onClick="handleLogout()">Logout</button>
18     `;
19   }
20   document.getElementById("app").innerHTML = html;
```

```
21 };
```

4 Implement Login Handler

Now the real fun begins! Check out how you can implement user login with a **single line of code**, without needing to write any backend code! Copy and paste this code snippet below to your editor under the comment.

```
1 /* 4 Implement Login Handler */
2 const handleLogin = async e => {
3   e.preventDefault();
4   const email = new FormData(e.target).get("email");
5   if (email) {
6     /* One-liner login 🤖 */
7     await magic.auth.loginWithMagicLink({ email });
8     render();
9   }
10 };
```

5 Implement Logout Handler

To wrap it all up, it's super easy to implement user logout as well. Copy and paste this code snippet below to your editor under the comment.

```
1 /* 5 Implement Logout Handler */
2 const handleLogout = async () => {
3   await magic.user.logout();
4   render();
5 };
```

🎉 **Congratulations!** Now that you've completed the walkthrough, you should have a working version of the Hello World app! Don't worry if things are not working for some reason - you can find the [completed working version](#) on CodeSandbox.

👉 What's Next

Now that you have an understanding of how Magic works on the client-side, we *strongly recommend* you to take a look at an end-to-end, full-stack example to learn about how to connect it to a Node.js backend server and see the full potential of Magic!

→ 🍎 [Full Stack Node.js](#)

/tutorials/full-stack-node-js

Did you know you can actually customize the login experience (using your own UI instead of the default) and also have a branded magic link email? You can learn how to do that with the link below!

→ 🎨 [Customization](#)

/tutorials/customization

Magic provides much flexibility and composability to be combined with many other powerful platforms such as Firebase. You can learn how to plug Magic into the entire Firebase suite of tools here!

[→ !\[\]\(2bdfe261b986065ee0ac76460d6528c9_img.jpg\) **Firestore Integration**](#)

</tutorials/firebase-integration>