

Connecting to the Private eXecution Environment (PXE)

PXE is a component of the Aztec Protocol that provides a private execution environment for your application.

As an app developer, the [PXE](#) interface provides you with access to the user's accounts and their private state, as well as a connection to the network for accessing public global state.

During the Sandbox phase, this role is fulfilled by the [Aztec Sandbox](#), which runs a local PXE and an Aztec Node, both connected to a local Ethereum development node like Anvil. The Sandbox also includes a set of pre-initialized accounts that you can use from your app.

In this section, we'll connect to the Sandbox from our project.

Create PXE client

We'll use the `createPXEClient` function from `aztec.js` to connect to the Sandbox. Sandbox exposes a HTTP JSON-RPC interface of PXE. By default it runs on `localhost:8080`. To test the connection works, we'll request and print the node's chain id.

Let's create our first `src/index.mjs` with the following contents:

```
all import
{ createPXEClient }
from
'@aztec/aztec.js' ;
const
{
  PXE_URL
=
'http://localhost:8080'
}
= process . env ;
async
function
main ( )
{ const pxe =
createPXEClient ( PXE_URL ) ; const
{ chainId }
=
await pxe . getNodeInfo ( ) ; console . log ( `Connected to chain { chainId } ` ) ; }
main ( ) . catch ( err
=>
{ console . error ( `Error in app: { err } ` ) ; process . exit ( 1 ) ; } ) Source code: yarn-project/end-to-end/src/sample-dapp/connect.mjs#L1-L16 Run this example as node src/index.mjs and you should see the following output:
```

Connected to chain 31337 info Should the above fail due to a connection error, make sure the Sandbox is running locally and on port 8080.

Load user accounts

In sandbox PXE comes with a set of pre-initialized accounts that you can use from your app. Let's try loading the accounts:

```
showAccounts const accounts =
```

```
await pxe . getRegisteredAccounts ( ) ; console . log ( ` User accounts:\n { accounts . map ( a
```

```
=> a . address ) . join ( '\n' ) } ` ) Source code: yarn-project/end-to-end/src/sample-dapp/index.mjs#L10-L13 Run again the above, and we should see:
```

```
User accounts: 0x0c8a6673d7676cc80aaebe7fa7504cf51daa90ba906861bfad70a58a98bf5a7d  
0x226f8087792beff8d5009eb94e65d2a4a505b70baf4a9f28d33c8d620b0ba972  
0x0e1f60e8566e2c6d32378bdcadb7c63696e853281be798c107266b8c3a88ea9b
```

Next steps

With a working connection to PXE, let's now setup our application by [compiling and deploying our contracts](#) . [Edit this page](#)

[Previous Setting up your project](#) [Next Contract Deployment](#)