

[Configuration](#)[Providers](#)[Credentials](#)

Version: v4

# Credentials

## How to

The Credentials provider allows you to handle signing in with arbitrary credentials, such as a username and password, two-factor authentication or hardware device (e.g. YubiKey U2F / FIDO).

It is intended to support use cases where you have an existing system you need to authenticate users against.

[pages/api/auth/\[...nextauth\].js](#)

```
import CredentialsProvider from "next-auth/providers/credentials"
...
providers: [
  CredentialsProvider({
    // The name to display on the sign in form (e.g. 'Sign in with...')
    name: 'Credentials',
    // The credentials is used to generate a suitable form on the sign in page.
    // You can specify whatever fields you are expecting to be submitted.
    // e.g. domain, username, password, 2FA token, etc.
    // You can pass any HTML attribute to the <input> tag through the object.
    credentials: {
      username: { label: "Username", type: "text", placeholder: "jsmith" },
      password: { label: "Password", type: "password" }
    },
    async authorize(credentials, req) {
      // You need to provide your own logic here that takes the credentials
      // submitted and returns either a object representing a user or value
      // that is false/null if the credentials are invalid.
      // e.g. return { id: 1, name: 'J Smith', email: 'jsmith@example.com' }
      // You can also use the `req` object to obtain additional parameters
      // (i.e., the request IP address)
      const res = await fetch("/your/endpoint", {
        method: 'POST',
        body: JSON.stringify(credentials),
```

```
headers: { "Content-Type": "application/json" }
}))
const user = await res.json()

// If no error and we have user data, return it
if (res.ok && user) {
  return user
}
// Return null if user data could not be retrieved
return null
}
})
]
...
```

See the [Credentials provider documentation](#) for more information.

**NOTE**

The Credentials provider can only be used if JSON Web Tokens are enabled for sessions. Users authenticated with the Credentials provider are not persisted in the database.

Options

Name	Description	Type	Required
id	Unique ID for the provider	string	Yes
name	Descriptive name for the provider	string	Yes
type	Type of provider, in this case credentials	"credentials"	Yes
credentials	The credentials to sign-in with	Object	Yes
authorize	Callback to execute once user is to be authorized	(credentials, req) => Promise<User>	Yes

 [Edit this page](#)

*Last updated on **May 16, 2024** by **Codie Newark***