`useSignTestingJwt`

Overview

The `useSignTestingJwt` hook is a development-only utility that allows you to sign JSON Web Tokens (JWTs) directly in the client. It is intended for **testing and initial development** when backend infrastructure for signing tokens is not yet in place.

> ⚠️ **Warning:** *This method is **insecure for production** use. Exposing your private key in client-side code introduces serious security risks. For production scenarios, JWTs must be signed securely on your backend.*

If you are integrating Replyke with your own authentication system, see the section on **Authentication with External User Systems** for guidance on implementing secure token generation.

Usage Example

```
import { useEffect, useState } from "react";
import { useSignTestingJwt } from "@replyke/react-js";

const PROJECT_ID = import.meta.env.VITE_PUBLIC_REPLYKE_PROJECT_ID;
const PRIVATE_KEY = import.meta.env.VITE_PUBLIC_REPLYKE_SECRET_KEY;

function ExampleComponent() {
  const signTestingJwt = useSignTestingJwt();
  const [jwt, setJwt] = useState<string>();

  useEffect(() => {
    const sign = async () => {
      const payload = { id: "user_123", name: "Jane Doe" };

      const token = await signTestingJwt({
        projectId: PROJECT_ID,
        privateKey: PRIVATE_KEY,
        payload,
      });

      setJwt(token);
    };
```

```
    sign();
  }, []);

  return <div>JWT: {jwt || "Signing..."}</div>;
}
```

Parameters & Returns

Parameters

The `signTestingJwt` function returned by the hook accepts a single object with the following fields:

Parameter	Type	Description
`projectId`	`string`	The Replyke project ID for which the JWT is being signed.
`privateKey`	`string`	The private key used to sign the JWT. **For development only.**
`payload`	`Record<string, any>`	The payload object to include in the signed JWT.

Returns

The hook returns an asynchronous function:

Return Value	Type	Description
`jwt`	`string`	The signed JWT as a string. Returns `undefined` if an error occurs.