gatsby-auth0-functions

This example shows how to use @serverless-jwt with Gatsby and Gatsby Hosted Functions.

Inspiration

This code was inspired by this article and repo.

🚀 Quick start

To test this Example locally you'll need to:

1. Create an application in Auth0 of type "Single Page Application" and configure `http://localhost:8000` as the Allowed Callback URL, Allowed Logout URL, Allowed Web Origins and Allowed CORS.
2. Create an API in Auth0 (e.g. with the name of `tv-shows` and identifier of `https://api/tv-shows`) and create a permission for that API (e.g. `read:shows`)
3. Rename the `.env-template` file to `.env.development` and update all of the settings there with the domain and clientId from the application you made in Auth0.
4. Run `yarn run start` which will run the Gatsby application and the Gatsby functions.

How does example this work?

Gatsby

The Gatsby application uses @auth0/auth0-react to authenticate the user. Once the user is authenticated, the Gatsby application will receive an `id_token` and `access_token` from Auth0;

The `access_token` is then provided to our Functions to authenticate the request.

Gatsby Functions

In the Gatsby Functions we use @serverless-jwt/jwt-verifier to secure our functions.

The `JwtVerifier` serves as a way to verify your token. If the token is not valid, then we return an error to the client. If it is valid, it will expose all of the claims to the current function and you'll have the guarantee that the request is authenticated.

```
const {
  JwtVerifier,
  getTokenFromHeader,
} = require("@serverless-jwt/jwt-verifier")

const jwt = new JwtVerifier({
  issuer: process.env.JWT_ISSUER,
  audience: process.env.JWT_AUDIENCE,
})

const shows = async (req, res) => {
  const scope = "read:shows"
  const token = getTokenFromHeader(req.get("authorization"))
  const claims = await jwt.verifyAccessToken(token)
```

```
  if (!claims || !claims.scope || claims.scope.indexOf(scope) === -1) {
    return res.status(403).json({
      error: "access_denied",
      error_description: `Token does not contain the required '${scope}' scope`,
    })
  }
}
```