I chose the Google Authentication strategy, because it is my favorite way to log in to other websites that offer authentication and what I’d prefer to use in my own applications. I prefer Google, because it is such a ubiquitous application. Nearly everyone has a Gmail account at the very least, if not other Google applications. It’s a quick and easy way to login to services like Spotify, Meetup, and MyFitnessPal, and I like the convenience of not having to store so many usernames and passwords.

The first step is registering the application with Google through the Google Developers Console. Google will issue a client ID and client secret for the application, which will need to be provided to the strategy. A redirect URI will have to be configured which matches the route in the application.

The Google authentication strategy authenticates users using their Google account and OAuth tokens. The strategy requires the specific callback verify, which receives the access token and optional refresh token, as well as profile,which contains the authenticated user’s Google profile. The verify callback must call cb providing a user to complete authentication. Below is an example of the Configure Strategy:

**var** GoogleStrategy = **require**('passport-google-oauth20').Strategy;

passport.**use**(**new** GoogleStrategy({

clientID: GOOGLE\_CLIENT\_ID,

clientSecret: GOOGLE\_CLIENT\_SECRET,

callbackURL: "http://www.example.com/auth/google/callback"

},

**function**(accessToken, refreshToken, profile, cb) {

User.findOrCreate({ googleId: profile.id }, **function** (err, user) {

**return** cb(err, user);

});

}

));

In order to authenticate requests, we use passport.authenticate() specifying the ‘google’ strategy:

app.get('/auth/google',

passport.authenticate('google', { scope: ['profile'] }));

app.get('/auth/google/callback',

passport.authenticate('google', { failureRedirect: '/login' }),

function(req, res) {

// Successful authentication, redirect home.

res.redirect('/');

});

Both Google and Facebook use OAuth 2.0, making this a very flexible strategy to employ when using Passport to authenticate users through multiple social media account options.