Google Sign In - Front y Back

Generar API KEY y API de Google

- Google Identity
- Obtén tu ID de cliente de la API de Google
- Creo el nuevo proyecto y me da las credenciales
- En credenciales tengo:
 - Client ID
 - Client Secret
- Voy a la consola API
- Estoy en API y servicios
- En el menú de la izquierda accedo a la pantalla de consentimiento
- Selecciono Externos para que todo quisqui pueda entrar con google Guardar y Continuar
- Permisos Guardar y Continuar
- Info opcional Guardar y Continuar
- Resumen Guardar y Continuar
- Crear Credenciales
 - o Id de Oauth
 - Tipo de aplicacion- aplicacion Web
 - Origenes autorizados de JavaScript, agrego:
 - http://localhost
 - http://localhost:8080
 - **NOTA **En la documentación dice de agregar los 2!
- CREAR
- El secret ID es solo del lado del backend y dudo que lo usemos
- Creo 2 variables de entorno, con GOOGLE_CLIENT_ID y SECRET_GOOGLE_ID

Usuario de Google Frontend

- En la carpeta pública tengo mi index.html.
- En los middlewares ya está configurado para que lo muestre en '/'
- Busco en la web de Google Display the Sign In With Google Button
- Copio y pego

- El data-login-uri no lo voy a usar (es una redirección al endpoint)
- El data-auto-prompt lo dejo en false porque si no se le va a pedir autenticación al entrar
- Busco handle credential responses with javascript functions en la web de Google
 - o Copio el data callback (que lo llamaré cuando la auth es exitosa)
 - Lo coloco en el google id onload
- Uso mi client ID

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Acceso Denegado</title>
</head>
<body>
    <h1>Google Sign In</h1>
    <hr>>
    <div id="g id onload"
       data-client_id="316623172064-
bjqdtma10uf7e47f16rm2ne1permn984.apps.googleusercontent.com"
       data-auto_prompt="false"
       data-callback="handleCredentialResponse">
    </div>
    <div class="g_id_signin"</pre>
       data-type="standard"
       data-size="large"
       data-theme="outline"
       data-text="sign in with"
       data-shape="rectangular"
       data-logo alignment="left">
    </div>
    <script src="https://accounts.google.com/gsi/client" async defer></script>
</body>
</html>
```

Debo añadir la función del data-callback. Copio y pego

```
<h1>Google Sign In</h1>
    <hr>>
    <div id="g id onload"
       data-client id="316623172064-
bjqdtma10uf7e47f16rm2ne1permn984.apps.googleusercontent.com"
       data-auto prompt="false"
       data-callback="handleCredentialResponse">
    </div>
    <div class="g_id_signin"</pre>
       data-type="standard"
       data-size="large"
       data-theme="outline"
       data-text="sign_in_with"
       data-shape="rectangular"
       data-logo_alignment="left">
    </div>
    <script src="https://accounts.google.com/gsi/client" async defer></script>
    # Google Sign In - Front y Back
## Generar API KEY y API de Google
- Google Identity
- Obtén tu ID de cliente de la API de Google
- Creo el nuevo proyecto y me da las credenciales
- En credenciales tengo:
  - Client ID
  - Client Secret
- Vov a la consola API
- Estoy en API y servicios
- En el menú de la izquierda accedo a la pantalla de consentimiento
- Selecciono Externos para que todo quisqui pueda entrar con google - Guardar y
Continuar
- Permisos - Guardar y Continuar
- Info opcional - Guardar y Continuar
- Resumen - Guardar y Continuar
- Crear Credenciales
  - Id de Oauth
    - Tipo de aplicacion- aplicacion Web
    - Origenes autorizados de JavaScript, agrego:
      - http://localhost
      - http://localhost:8080
      - **NOTA **En la documentación dice de agregar los 2!
- El secret ID es solo del lado del backend y dudo que lo usemos
- Creo 2 variables de entorno, con GOOGLE_CLIENT_ID y SECRET_GOOGLE_ID
## Usuario de Google Frontend
- En la carpeta pública tengo mi index.html.
- En los middlewares ya está configurado para que lo muestre en '/'
```

```
- Busco en la web de Google Display the Sign In With Google Button
- Copio y pego
- El data-login-uri no lo voy a usar ( es una redirección al endpoint )
- El data-auto-prompt lo dejo en false porque si no se le va a pedir autenticación
al entrar
- Busco handle credential responses with javascript functions en la web de Google
  - Copio el data callback ( que lo llamaré cuando la auth es exitosa)
  - Lo coloco en el google_id_onload
- Uso mi client ID
~~~html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Google Sign In</title>
</head>
<body>
    <h1>Google Sign In</h1>
    <hr>>
    <div id="g_id_onload"
       data-client_id="316623172064-
bjqdtma10uf7e47f16rm2ne1permn984.apps.googleusercontent.com"
       data-auto prompt="false"
       data-callback="handleCredentialResponse">
    </div>
    <div class="g id signin"</pre>
       data-type="standard"
       data-size="large"
       data-theme="outline"
       data-text="sign in with"
       data-shape="rectangular"
       data-logo_alignment="left">
    </div>
    <script src="https://accounts.google.com/gsi/client" async defer></script>
    <script>
  function handleCredentialResponse(response) {
     // decodeJwtResponse() is a custom function defined by you
     // to decode the credential response.
     const responsePayload = decodeJwtResponse(response.credential);
     console.log("ID: " + responsePayload.sub);
     console.log('Full Name: ' + responsePayload.name);
     console.log('Given Name: ' + responsePayload.given_name);
     console.log('Family Name: ' + responsePayload.family_name);
     console.log("Image URL: " + responsePayload.picture);
     console.log("Email: " + responsePayload.email);
  }
</script>
```

```
</body>
</html>
```

- La función decodeJwtResponse no viene definida en el script (la dejan abierta)
- No se recomienda hacer la decodificación aqui. Nos daría la info de la persona, el nombre, la foto, etc
- En lugar de decodificarlo, voy a enviar directamente el token
- NOTA No funciona con una función de flecha

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Acceso Denegado</title>
</head>
<body>
    <h1>Google Sign In</h1>
    <hr>>
    <div id="g_id_onload"
       data-client id="316623172064-
bjqdtma10uf7e47f16rm2ne1permn984.apps.googleusercontent.com"
       data-auto_prompt="false"
       data-callback="handleCredentialResponse">
    </div>
    <div class="g_id_signin"</pre>
       data-type="standard"
       data-size="large"
       data-theme="outline"
       data-text="sign_in_with"
       data-shape="rectangular"
       data-logo_alignment="left">
    </div>
    <script src="https://accounts.google.com/gsi/client" async defer></script>
    <script>
  function handleCredentialResponse(response) {
     // decodeJwtResponse() is a custom function defined by you
     // to decode the credential response.
     const googleToken = response.credential
     console.log(googleToken)
 }
</script>
</body>
</html>
```

- Ahora tengo el token en la consola
- No tengo implementado un sign out
- Puedo hacerlo manualmente borrando todas las cookies en la consola Aplicación/cookies/botón de texto+prohibido
- Me pide que ingrese el correo y la doble auth
- La idea es mandar este idToken al backend y creemos un usuario basado ene ste token

Ruta para manejar autenticación de Google

- Manejo la ruta para recibir el token de autenticación de google
- auth.router.js

```
router.post('/google', [
    check('googleToken', 'id_token es necesario').not().isEmpty(),
    validarCampos
], auth.loginGoogle)
```

auth.controller.js

Creo una petición POST en Thunder Client a este endpoint

http://localhost:8080/api/auth/google

• En el body le paso el googleToken

```
"googleToken":
"eyJhbGciOiJSUzI1NiIsImtpZCI6ImM5YWZkYTM2ODJlYmYwOWViMzA1NWMxYzRiZDM5Yjc1MWZiZjgxO
TUiLCJ0eXAiOiJKV1QifQ.eyJpc3MiOiJodHRwczovL2FjY291bnRzLmdvb2dsZS5jb20iLCJuYmYiOjE2
ODMxMTY5OTUsImF1ZCI6IjMxNjYyMzE3MjA2NC1ianFkdG1hMTB1ZjdlNDdmMTZybTJuZTFwZXJtbjk4NC
5hcHBzLmdvb2dsZXVzZXJjb250ZW50LmNvbSIsInN1YiI6IjExMTIwMjEyNjI0NzUzMTE1OTQ1MiIsImVt
YWlsIjoiYmVyY2FzdDgxQGdtYWlsLmNvbSIsImVtYWlsX3ZlcmlmaWVkIjp0cnVlLCJhenAiOiIzMTY2Mj
MxNzIwNjQtYmpxZHRtYTEwdWY3ZTQ3ZjE2cm0ybmUxcGVybW450DQuYXBwcy5nb29nbGV1c2VyY29udGVu
dC5jb20iLCJuYW1lIjoiSXNtYWVsIEJlcsOzbiIsInBpY3R1cmUiOiJodHRwczovL2xoMy5nb29nbGV1c2
VyY29udGVudC5jb20vYS9BR05teXhhOThJcVZpZ3JuWmprTXdzVXRtSFUzOC0zV3Eta0NhVUFGa3VJeT1z
OTYTYYISImdpdmVuX25hbWUiOiJJc21hZWwiLCJmYW1pbHlfbmFtZSI6IkJlcsOzbiIsImlhdCI6MTY4Mz
ExNzI5NSwiZXhwIjoxNjgzMTIwODk1LCJqdGkiOiJiMDM4OWUzN2RmODR1YjEzMTM5ODc2NWZkZjM5YjNk
ZmEwNzUzNjA2In0.EMkWvZQb-e2yCuZT5NjDmZ6BWYkyVEi2VFobpIad8LT3 2JYT0z158 V-
bDvJLntTY-
hCwlCX0Bz86yBmD6sIEbHOXpUQH7JDf0mPiP4fEe1AQMHyqT6tZffKZP8Y VT8XSMG1E8YCLX5dff-
BtKj7P1RLi05SCeM1UFx-
P5uB8gkkYfbMwr0fEG0lrH1GQBy7iE6AiBiQ6yUUQh34q8qyMP_Cuw0VD0c6YeAZ1Pihm4IjdHov86u2F3
17yf8DyHNQNAaR-DHuM-LkAQvDwHM25pc35MHMtvaZXnqQ5ANHoOrEJVLh02YCwfK0h730i27Dd-
xuXBanhWTrmmuxI2Qw"
```

 Aunque por ahora hubiera valido que le pasara caulquier cosa como token porque no hay validación del token todavía

- Vamos a enviar el token desde el frontend
- Uso el fetch
- Esta promesa, al usar fetch API no me regresa el body directamente, si no que un readable string. Habrá que serializarlo
- debo especificar el método ya que es POST, con los headers y parsear el body

•

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Google Sign in</title>
</head>
<body>
    <h1>Google Sign In</h1>
    <hr>>
    <div id="g id onload"
       data-client id="316623172064-
bjqdtma10uf7e47f16rm2ne1permn984.apps.googleusercontent.com"
       data-auto_prompt="false"
       data-callback="handleCredentialResponse">
    </div>
    <div class="g_id_signin"</pre>
       data-type="standard"
       data-size="large"
       data-theme="outline"
       data-text="sign in with"
       data-shape="rectangular"
       data-logo_alignment="left">
    </div>
    <script src="https://accounts.google.com/gsi/client" async defer></script>
  function handleCredentialResponse(response) {
     // decodeJwtResponse() is a custom function defined by you
     // to decode the credential response.
     const body = {googleToken: response.credential} //paso el token al body. Paso
este body al JSON.stringify
     fetch('http://localhost:8080/api/auth/google',{
      method: "POST",
      headers:{'Content-Type': 'application/json'},
      body: JSON.stringify(body) //Aquí paso el body!!!
     })
            .then(resp=> resp.json())
```

```
.then(resp=>{
          console.log(resp) // esta es la respuesta que mi backend va a
regresar
})
     .catch(console.warn) //por si algo sale mal

}
</script>
</body>
</html>
```

- Ahora tengo el token en el backend!!!
- Falta abrirlo y extraer la info que me interesa

Validar Token de Google en el backend

- Busco verify Google id token on your server side
- Instalación necesaria

npm install google-auth-library --save

- Copio el código de la página que es la función verifyldToken en un archivo en /helpers/google-verify.js
- Cambio el nombre a googleVerify
- Introduzco la variable de entorno GOOGLE_CLIENT_ID donde pone CLIENT_ID
- Debo proveer a googleVerify el token, lo inicializo con un string vacío para tener el error controlado

```
import {OAuth2Client} from 'google-auth-library';

const client = new OAuth2Client(process.env.GOOGLE_CLIENT_ID);
async function googleVerify(token = "") {
  const ticket = await client.verifyIdToken({
    idToken: token,
    audience: process.env.GOOGLE_CLIENT_ID, // Specify the CLIENT_ID of the app
that accesses the backend
    // Or, if multiple clients access the backend:
    //[CLIENT_ID_1, CLIENT_ID_2, CLIENT_ID_3]
  });
  const payload = ticket.getPayload();

console.log(payload)
}

export default googleVerify
```

Voy al controlador, importo googleVerify

```
const loginGoogle = async (req, res)=>{
    const {googleToken} = req.body
    try {
        const googleUser = await googleVerify(googleToken)
         console.log(googleUser) //recibo nombre, img y correo (mirar código
siguiente, lo desestructuro y renombro en el return)
                                 //de googleVerify
        res.json({
            msg: "Todo bien!",
            googleUser
        })
    } catch (error) {
        console.log(error)
            res.status(400).json({
            msg: "El token no se pudo verificar"
        })
    }
}
```

- Si hago la autenticación en consola, el console.log del payload en verify-google me devuelve esto
- NOTA algunos datos se han omitido con ***

```
iss: 'https://accounts.google.com',
 nbf: 168*126**,
 aud: '316****064-bjqdtma10uf7e47f1****1permn984.apps.googleusercontent.com',
 sub: '1112021262475*****2', //id único
  email: '********,
 email_verified: true,
  azp: '31662317**********16rm2ne1permn984.apps.googleusercontent.com',
//procedimiento de autenticación con el CLIENT ID
 name: 'Miguel Castaño',
  picture: 'https://lh3.googleusercontent.com/a/AGNmyxa98IqVigrnZjk******3Wq-
kCaUAFkuIy=s96-c',
  given_name: 'Miguel'
 family name: 'Castaño',
 iat: 1683127***,
 exp: 1683130***,
 jti: '46da4d7ef9fbee58229f9c86*****63de0ad'
}
```

- De aquí voy a extraer la info
- · google-verify.js

```
import {OAuth2Client} from 'google-auth-library';
const client = new OAuth2Client(process.env.GOOGLE_CLIENT_ID);
async function googleVerify(token="") {
  const ticket = await client.verifyIdToken({
      idToken: token,
      audience: process.env.GOOGLE_CLIENT_ID, // Specify the CLIENT_ID of the app
that accesses the backend
     // Or, if multiple clients access the backend:
      //[CLIENT_ID_1, CLIENT_ID_2, CLIENT_ID_3]
  });
  const payload = ticket.getPayload();
  const {name, picture, email} = payload
  console.log({name, picture, email})
 return {
   nombre: name,
  img: picture,
   correo: email
  }
}
export default googleVerify
```

• Ahora si vuelvo a hacer la autenticación con google recibo en consola el nombre, la foto y el email - Puedo desestructurarlo directamente de googleVerify en el controlador

```
})
}
```

Creo un usuario personalizado con las credenciales de Google

• Si el usuario no existe debo crearlo

```
const loginGoogle = async (req, res)=>{
   const {googleToken} = req.body
   try {
        const {nombre, img, correo} = await googleVerify(googleToken)
        //console.log(googleUser) //recibo nombre, img y correo
        let usuario = await Usuario.findOne({correo})
         //Si el usuario no existe debo crearlo
        if(!usuario){
            const data ={
                nombre,
                img,
                correo,
                password: ':P', //Hay que mandar el password porque es
obligatorio. Nunca hará match porque se compara contra un hash
                google: true
                //el USER ROLE viene como default en el modelo
            usuario = new Usuario(data)
            await usuario.save() //uso el await porque es una interacción con la
DB!!
        }
        //Si el usuario en DB tiene el estado en false voy a negar su
autenticación
        if(!usuario.estado) return res.status(401).json({msg: "Hable con el
administrador. Usuario bloqueado"})
        //genero el jwt
        const token = await generarJWT(usuario.id)
        res.json({
            msg: "Todo bien!",
            usuario,
            token
```

```
})

} catch (error) {
    console.log(error)
    res.status(400).json({
        msg: "El token no se pudo verificar"
    })
}
```

- No hace falta que especifique el rol porque viene USER_ROLE marcado por defecto
- models/user.js

```
rol:{
    type: String,
    required: true,
    enum: ['ADMIN_ROLE', 'USER_ROLE', 'VENTAS_ROLE'],
    default: 'USER_ROLE'
}
```

SIGN OUT

- La manera más fñacil es borrar todas las cookies
- Hay otra forma de hacerlo de la parte de google
- Creo un botón para el logout
- Tengo disponible google.account.id
- Para hacer el logout necesito el correo de la persona que se desautenticó
- Después de hacer la autenticación (en el then) puedo grabarlo en el LocalStorage

```
data-auto_prompt="false"
       data-callback="handleCredentialResponse">
    </div>
    <div class="g_id_signin"</pre>
       data-type="standard"
       data-size="large"
       data-theme="outline"
       data-text="sign in with"
       data-shape="rectangular"
       data-logo_alignment="left">
    </div>
    <button id="sign_out">SignOut</button>
    <script src="https://accounts.google.com/gsi/client" async defer></script>
    <script>
 function handleCredentialResponse(response) {
     // decodeJwtResponse() is a custom function defined by you
     // to decode the credential response.
     const body = {googleToken: response.credential} // este body lo paso al
JSON.stringify
     fetch('http://localhost:8080/api/auth/google',{
      method: "POST",
      headers:{'Content-Type': 'application/json'},
      body: JSON.stringify(body)
     })
            .then(resp=> resp.json())
            .then(resp=>{
               console.log(resp) // esta es la respuesta que mi backend va a
regresar
               //después de hacer la autenticación puedo grabarla en el
LocalStorage
               localStorage.setItem('email', resp.usuario.correo)
            })
            .catch(console.warn) //por si algo sale mal
  }
 //selecciono el botón de sign out
  const button = document.getElementById('sign out')
  //le añado el evento
 button.onclick =()=>{
  //tengo acceso al estar loggeado al google.account.id
  console.log(google.account.id)
  }
</script>
</body>
</html>
```

- Si voy a la consola/Aplicación/LocalStorage tengo el email guardado
- Uso el método revoke para hacer el logout
- Limpio el localStorage

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Google Sign in</title>
</head>
<body>
    <h1>Google Sign In</h1>
    <hr>>
    <div id="g_id_onload"
       data-client_id="316623172064-
bjqdtma10uf7e47f16rm2ne1permn984.apps.googleusercontent.com"
       data-auto prompt="false"
       data-callback="handleCredentialResponse">
    </div>
    <div class="g_id_signin"</pre>
       data-type="standard"
       data-size="large"
       data-theme="outline"
       data-text="sign in with"
       data-shape="rectangular"
       data-logo alignment="left">
    </div>
    <button id="sign out">SignOut</button>
    <script src="https://accounts.google.com/gsi/client" async defer></script>
    <script>
  function handleCredentialResponse(response) {
     // decodeJwtResponse() is a custom function defined by you
     // to decode the credential response.
     const body = {googleToken: response.credential} // este body lo paso al
JSON.stringify
     fetch('http://localhost:8080/api/auth/google',{
      method: "POST",
      headers:{'Content-Type': 'application/json'},
      body: JSON.stringify(body)
     })
            .then(resp=> resp.json())
            .then(resp=>{
               console.log(resp) // esta es la respuesta que mi backend va a
```

```
regresar
               //después de hacer la autenticación puedo grabarla en el
LocalStorage
               localStorage.setItem('email', resp.usuario.correo)
            })
            .catch(console.warn) //por si algo sale mal
 }
 //selecciono el botón de sign_out
 const button = document.getElementById('sign_out')
 //le añado el evento
 button.onclick =()=>{
  //tengo acceso al estar loggeado al google.account.id
  //console.log(google.account.id)
  //para hacer el logout llamo al método revoke
  google.account.id.revoke(localStorage.getItem('email'), done=>{
     //cuando acaba el proceso llama al callback, limpio el localStorage
     localStorage.clear()
      location.reload()//recargar la página y vaciarla
  })
 }
</script>
</body>
</html>
```

NOTA este código no funciona. ACTUALIZACIÓN

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Google Sign in</title>
</head>
<body>
    <h1>Google Sign In</h1>
    <hr>>
    <div id="g_id_onload"
       data-client_id="316623172064-
bjqdtma10uf7e47f16rm2ne1permn984.apps.googleusercontent.com"
       data-auto_prompt="false"
       data-callback="handleCredentialResponse">
```

```
</div>
    <div class="g_id_signin"</pre>
       data-type="standard"
       data-size="large"
       data-theme="outline"
       data-text="sign in with"
       data-shape="rectangular"
       data-logo alignment="left">
    </div>
   <button id="sign_out" onClick="cerrarSesion()">SignOut</button>
   <script src="https://accounts.google.com/gsi/client" async defer></script>
    <script>
 function handleCredentialResponse(response) {
     // decodeJwtResponse() is a custom function defined by you
     // to decode the credential response.
     const body = {googleToken: response.credential} // este body lo paso al
JSON.stringify
     fetch('http://localhost:8080/api/auth/google',{
     method: "POST",
     headers:{'Content-Type': 'application/json'},
     body: JSON.stringify(body)
     })
            .then(resp=> resp.json())
            .then(resp=>{
               console.log(resp) // esta es la respuesta que mi backend va a
regresar
               //después de hacer la autenticación puedo grabarla en el
LocalStorage
               localStorage.setItem('email', resp.usuario.correo)
            })
            .catch(console.warn) //por si algo sale mal
 }
 //selecciono el botón de sign out
 const button = document.getElementById("sign_out")
 //le añado el evento
 /*button.onclick =()=>{
  //tengo acceso al estar loggeado al google.account.id
  //console.log(google.account.id)
  //para hacer el logout llamo al método revoke
   google.account.id.revoke(localStorage.getItem('email'), done=>{
     //cuando acaba el proceso llama al callback, limpio el localStorage
     localStorage.clear()
      location.reload()//recargar la página y vaciarla
   })
```

```
function cerrarSesion(){
  google.accounts.id.disableAutoSelect()
  google.accounts.id.revoke(localStorage.getItem('email'), done=>{
    localStorage.clear(
        location.reload()
    )
  })
  })
}

</script>
</body>
</html>
```