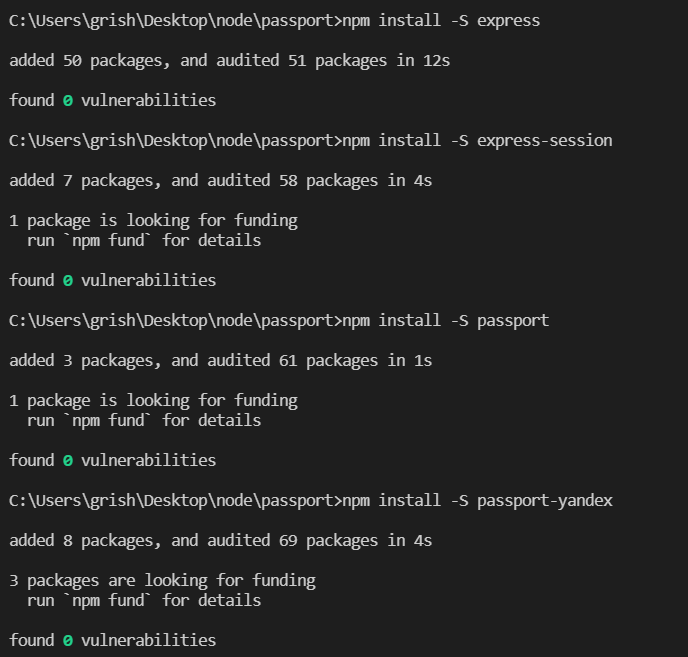
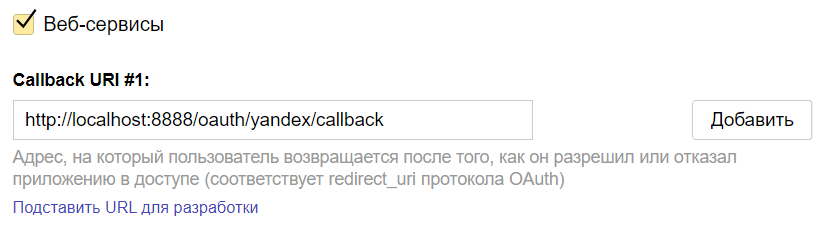
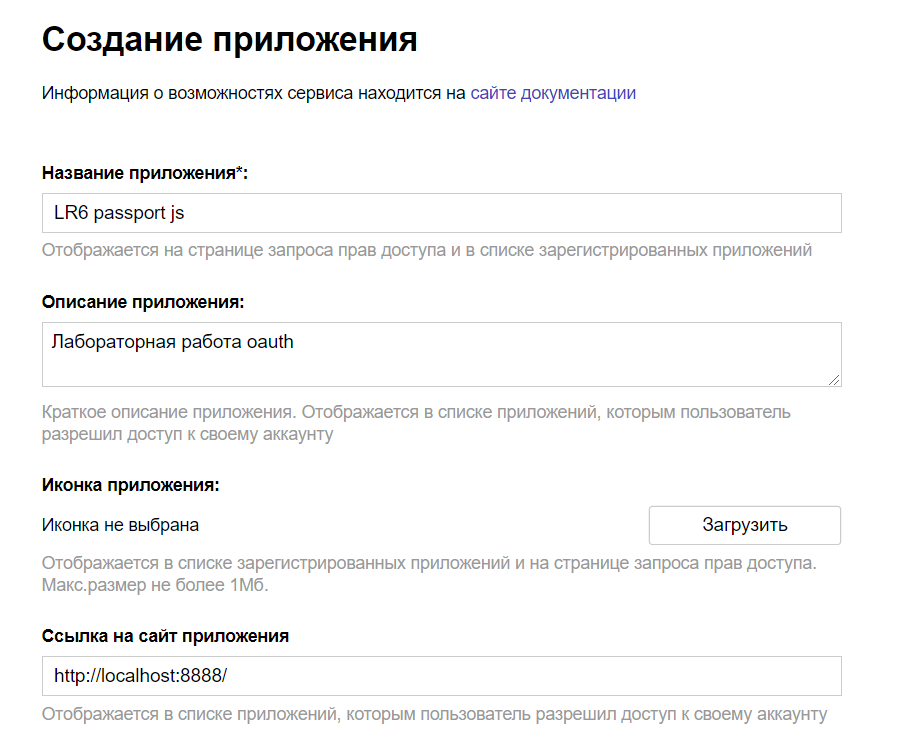
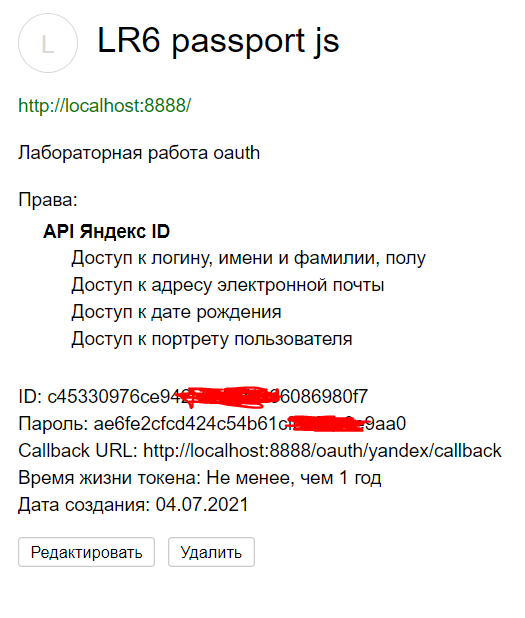
Установка



Регистрация





Код

const express = require('express');

const session = require('express-session');

const path = require('path');

const app = express();

const port = 8888;

const passport = require('passport');

const YandexStrategy = require('passport-yandex').Strategy;

app.use(session({ secret: "supersecret", resave: true, saveUninitialized: true}));

let Users = [{'login':'admin', 'email':'pavel.grishutenko@yandex.ru'}];

const findUserByLogin = (login) => {

    return Users.find((element)=> {

        return element.login == login;

    })

}

const findUserByEmail = (email) => {

    return Users.find((element)=> {

        return element.email.toLowerCase() == email.toLowerCase();

    })

}

app.use(passport.initialize());

app.use(passport.session());

passport.serializeUser((user, done) => {

    done(null, user.login);

  });

passport.deserializeUser((login, done) => {

    user = findUserByLogin(login);

        done(null, user);

});

passport.use(new YandexStrategy({

    clientID: YANDEX\_CLIENT\_ID,

    clientSecret: YANDEX\_CLIENT\_SECRET,

    callbackURL: "http://localhost:8888/auth/yandex/callback"

  },

  (accessToken, refreshToken, profile, done) => {

    let user = findUserByEmail(profile.emails[0].value);

    user.profile = profile;

    if (user) return done(null, user);

    done(true, null);

  }

));

const isAuth = (req, res, next) => {

    if (req.isAuthenticated()) return next();

    res.redirect('/sorry');

};

app.get('/', (req, res) => {

    res.sendFile(path.join(\_\_dirname, 'index.html'));

});

app.get('/sorry', (req, res) => {

    res.sendFile(path.join(\_\_dirname, 'sorry.html'));

});

app.get('/private', isAuth, (req, res) => {

    res.send(req.user);

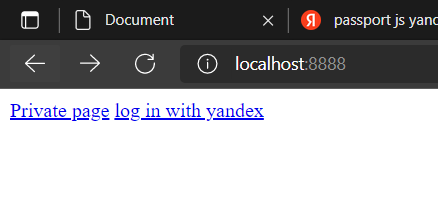
});

app.get('/auth/yandex.ru', passport.authenticate('yandex'));

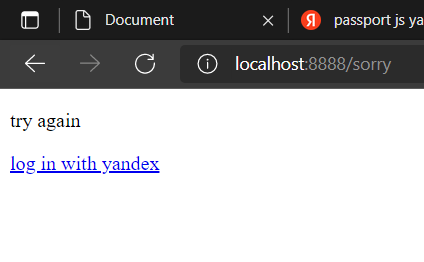
app.get('/auth/yandex/callback', passport.authenticate('yandex', { failureRedirect: '/sorry' , successRedirect: '/private'}))

app.listen(port, () => {console.log("server starting")});

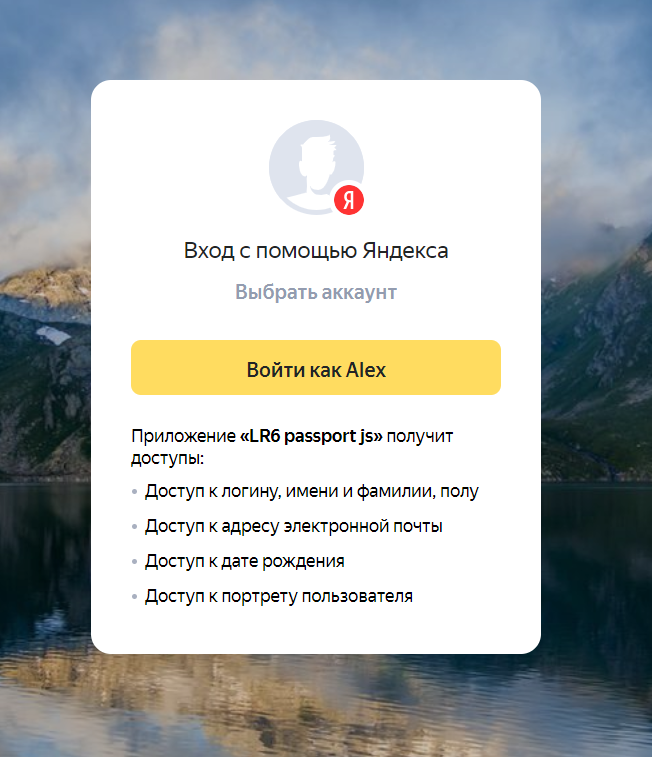
Тест входа через Yandex



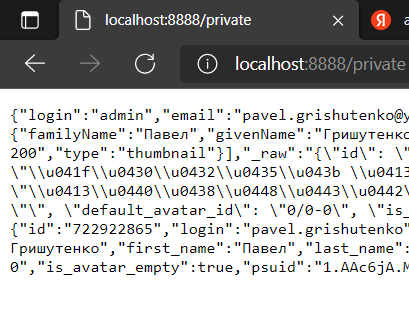
Попытка зайти на приватную страницу



Авторизация

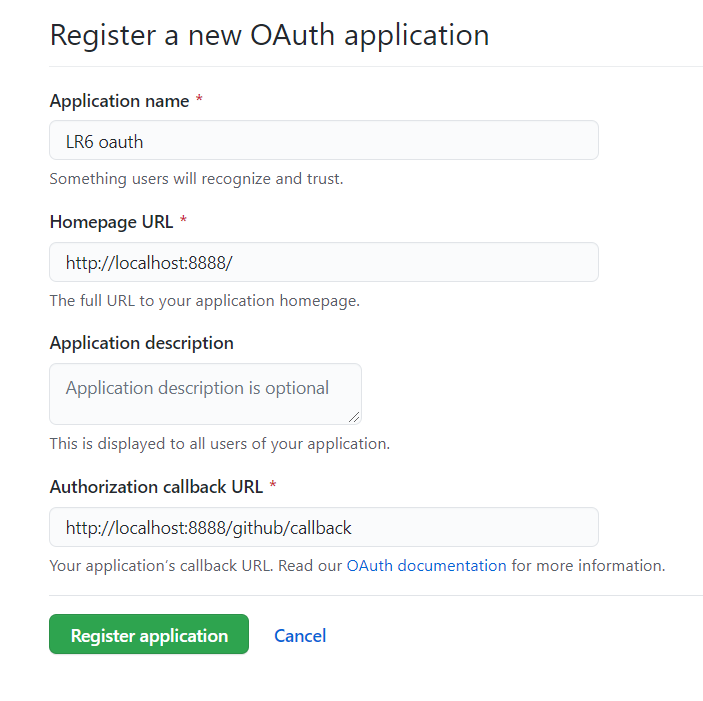


Private страница выдает информацию запрошенную у пользователя

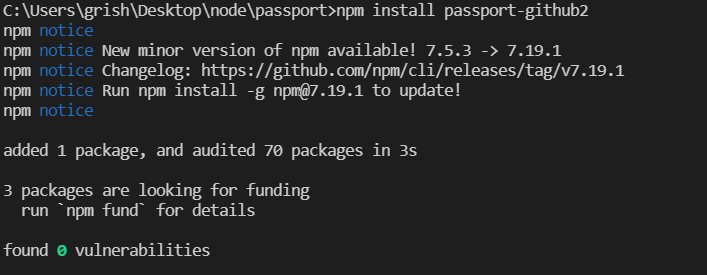


Авторизация через github

Регистрация приложения



Установка



Код

const express = require('express');

const session = require('express-session');

const path = require('path');

const app = express();

const port = 8888;

const passport = require('passport');

const YandexStrategy = require('passport-yandex').Strategy;

const GitHubStrategy = require('passport-github2').Strategy;

app.use(session({ secret: "supersecret", resave: true, saveUninitialized: true}));

let Users = [{'login':'admin', 'email':'grishutenkopp@gmail.com'},

            {'login':'admin', 'email':'pavel.grishutenko@yandex.ru'}];

const findUserByLogin = (login) => {

    return Users.find((element)=> {

        return element.login == login;

    })

}

const findUserByEmail = (email) => {

    return Users.find((element)=> {

        return element.email.toLowerCase() == email.toLowerCase();

    })

}

app.use(passport.initialize());

app.use(passport.session());

passport.serializeUser((user, done) => {

    done(null, user.login);

  });

  //user - объект, который Passport создает в req.user

passport.deserializeUser((login, done) => {

    user = findUserByLogin(login);

        done(null, user);

});

passport.use(new YandexStrategy({

    clientID: YANDEX\_CLIENT\_ID,

    clientSecret: YANDEX\_CLIENT\_SECRET,

    callbackURL: "http://localhost:8888/auth/yandex/callback"

  },

  (accessToken, refreshToken, profile, done) => {

    let user = findUserByEmail(profile.emails[0].value);

    if (user) return done(null, user);

    done(true, null);

  }

));

passport.use(new GitHubStrategy({

    clientID: GITHUB\_CLIENT\_ID,

    clientSecret: GITHUB\_CLIENT\_SECRET,

    callbackURL: "http://localhost:8888/auth/github/callback",

    scope: 'user:email'

  },

  (accessToken, refreshToken, profile, done) => {

    let user = findUserByEmail(profile.emails[0].value);

    console.log(profile);

    if (user) return done(null, user);

    done(true, null);

  }

));

const isAuth = (req, res, next) => {

    if (req.isAuthenticated()) return next();

    res.redirect('/sorry');

};

app.get('/', (req, res) => {

    res.sendFile(path.join(\_\_dirname, 'index.html'));

});

app.get('/sorry', (req, res) => {

    res.sendFile(path.join(\_\_dirname, 'sorry.html'));

});

app.get('/private', isAuth, (req, res) => {

    res.send(req.user);

});

app.get('/auth/yandex.ru', passport.authenticate('yandex'));

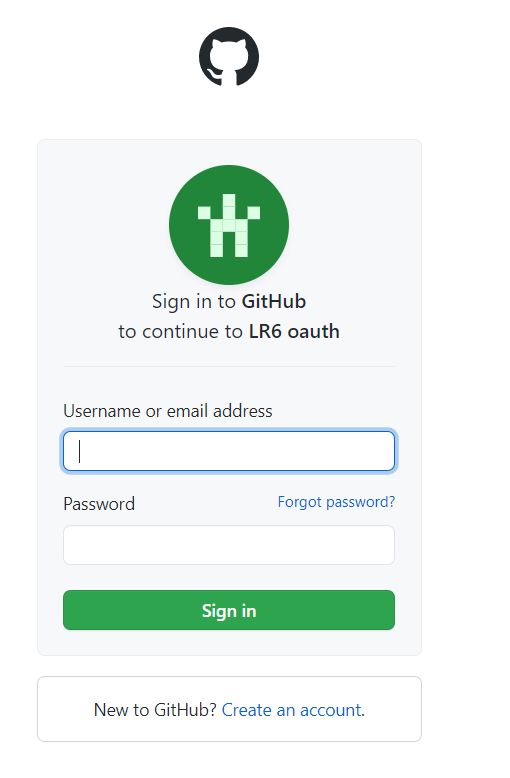
app.get('/auth/yandex/callback', passport.authenticate('yandex', { failureRedirect: '/sorry' , successRedirect: '/private'}))

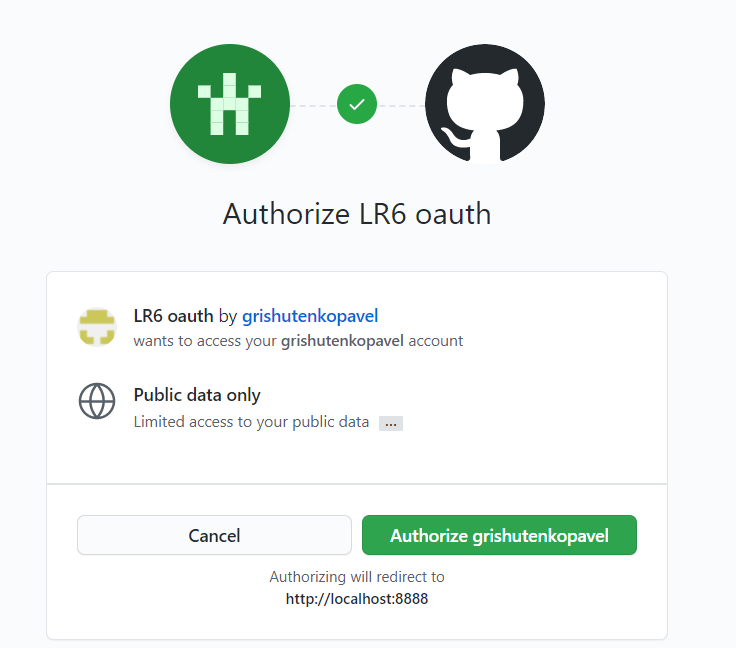
app.get('/auth/github', passport.authenticate('github', { scope: [ 'user:email' ] }));

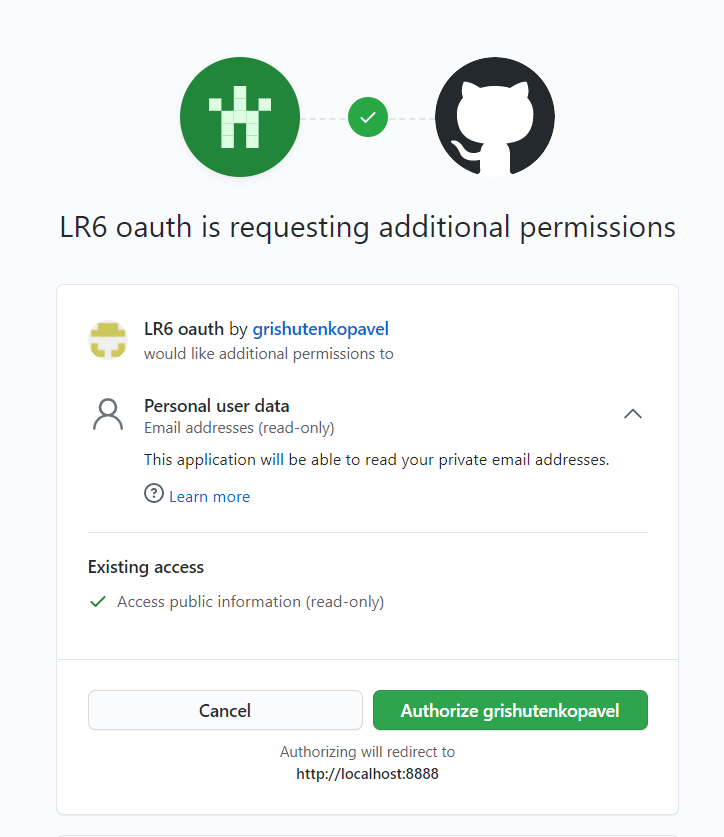
app.get('/auth/github/callback', passport.authenticate('github', { failureRedirect: '/sorry' , successRedirect: '/private'}))

app.listen(port, () => {console.log("server starting")});

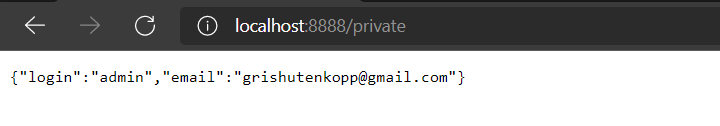
Вход







Приватная страница



Полученные данные от github

