

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ САНКТ-ПЕТЕРБУРГСКИЙ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
Факультет информационных технологий и программирования

Лабораторная работа №6
«Добавление авторизации и пользовательских сессий»
по дисциплине «Web программирование»

Выполнил: студент группы М33122 Белоногов Е.В.

В качестве поставщика услуг авторизации был выбран SuperTokens.

Это альтернатива Auth0 с полностью открытым исходным кодом, которая позволяет настроить аутентификацию менее чем за 30 минут. Есть возможность воспользоваться Managed Infrastructure, т.е. сам сервер авторизации (Identity Provider) доступен вам как облачное решение (используют AWS для размещения своих серверов). Комплексное решение, включающее вход в систему, регистрацию, управление пользователями и сеансами без сложностей протоколов OAuth.

В качестве способа авторизации был выбран вариант по логину и паролю. Для внедрения в приложение был создан модуль AuthModule содержащий имплементацию service, guard, middleware и exception filter. AuthModule является динамическим, т.е. он получает конфигурацию в момент регистрации модуля (с помощью метода forRoot).

Реализация:

auth.module.ts

```
src > auth > TS auth.module.ts > AuthModule
1  import { MiddlewareConsumer, Module, NestModule, DynamicModule } from '@nestjs/common';
2  import { AuthMiddleware } from './auth.middleware';
3  import { ConfigInjectionToken, AuthModuleConfig } from './config.interface';
4  import { SupertokensService } from './supertokens/supertokens.service';
5
6
7  @Module({
8    providers: [],
9    exports: [],
10    controllers: [],
11
12    export class AuthModule implements NestModule {
13      configure(consumer: MiddlewareConsumer) {
14        consumer.apply(AuthMiddleware).forRoutes('*');
15      }
16
17      static forRoot({ connectionURI, apiKey, appInfo }: AuthModuleConfig): DynamicModule {
18        return {
19          providers: [
20            {
21              useValue: {
22                appInfo,
23                connectionURI,
24                apiKey,
25              },
26              provide: ConfigInjectionToken,
27            },
28            SupertokensService, // ???
29          ],
30          exports: [],
31          imports: [],
32          module: AuthModule,
33        };
34      }
35    }
```

config.interface.ts:

```
src > auth > TS config.interface.ts > [🔗] AuthModuleConfig
1   import { AppInfo } from 'supertokens-node/lib/build/types';
2
3   export const ConfigInjectionToken = 'ConfigInjectionToken';
4
5   export type AuthModuleConfig = {
6     appInfo: AppInfo;
7     connectionURI: string;
8     apiKey?: string;
9   }
```

supertokens.service.ts:

```
src > auth > supertokens > TS supertokens.service.ts > [🔗] SupertokensService > [📦] constructor > [🔑] recipe
1   import { Inject, Injectable } from '@nestjs/common';
2   import supertokens from 'supertokens-node';
3   import Session from 'supertokens-node/recipe/session';
4   import EmailPassword from 'supertokens-node/recipe/emailpassword';
5   import { ConfigInjectionToken, AuthModuleConfig } from '../config.interface';
6
7
8   @Injectable()
9   export class SupertokensService {
10     constructor(@Inject(ConfigInjectionToken) private config: AuthModuleConfig) {
11       supertokens.init({
12         appInfo: config.appInfo,
13         supertokens: {
14           connectionURI: config.connectionURI,
15           apiKey: config.apiKey,
16         },
17         recipeList: [
18           EmailPassword.init({
19             // signUpFeature: {
20             //   formFields: [{
21             //     id: "email"
22             //   }, {
23             //     id: "password"
24             //   }, {
25             //     id: "name",
26             //     optional: true
27             //   }]
28             // }
29           }),
30           Session.init(),
31         ],
32       });
33     }
34   }
```

auth.middleware.ts:

```
src > auth > TS auth.middleware.ts > AuthMiddleware > use
1  import { Injectable, NestMiddleware } from '@nestjs/common';
2  import { middleware } from 'supertokens-node/framework/express';
3
4
5  @Injectable()
6  export class AuthMiddleware implements NestMiddleware {
7    supertokensMiddleware: any;
8
9    constructor() {
10     this.supertokensMiddleware = middleware();
11   }
12
13   use(req: Request, res: any, next: () => void) {
14     return this.supertokensMiddleware(req, res, next);
15   }
16 }
```

auth.guard.ts:

```
src > auth > TS auth.guard.ts > AuthGuard > canActivate
1  import { CanActivate, ExecutionContext, Injectable } from '@nestjs/common';
2  import { Error as STError } from 'supertokens-node';
3  import { verifySession } from 'supertokens-node/recipe/session/framework/express';
4
5
6  @Injectable()
7  export class AuthGuard implements CanActivate {
8      async canActivate(context: ExecutionContext): Promise<boolean> {
9          const ctx = context.switchToHttp();
10
11          let err = undefined;
12          const req = ctx.getRequest();
13          const resp = ctx.getResponse();
14
15          await verifySession() (
16              req,
17              resp,
18              (res) => {
19                  err = res;
20              },
21          );
22
23          if (resp.headersSent) {
24              throw new STError({
25                  message: "RESPONSE_SENT",
26                  type: "RESPONSE_SENT",
27              });
28          }
29
30          if (err) {
31              throw err;
32          }
33
34          return true;
35      }
36  }
```

auth.filter.ts:

```
src > auth > TS auth.filter.ts > SupertokensExceptionFilter > catch
1  import { ExceptionFilter, HttpException, ArgumentsHost, Catch, ForbiddenException } from '@nestjs/common';
2  import { Request, Response, NextFunction, ErrorRequestHandler } from 'express';
3  import { errorHandler } from 'supertokens-node/framework/express';
4  import { Error as STError } from 'supertokens-node';
5
6
7  @Catch(STError)
8  export class SupertokensExceptionFilter implements ExceptionFilter {
9      handler: ErrorRequestHandler;
10
11      constructor() {
12          this.handler = errorHandler();
13      }
14      (parameter) exception: Error
15      catch(exception: Error, host: ArgumentsHost) {
16          const ctx = host.switchToHttp();
17          const resp = ctx.getResponse<Response>();
18          if (resp.headersSent) {
19              return;
20          }
21
22          this.handler(
23              exception,
24              ctx.getRequest<Request>(),
25              resp,
26              ctx.getNext<NextFunction>(),
27          );
28      }
29  }
```

Регистрация всех созданных компонент:

AppModule:

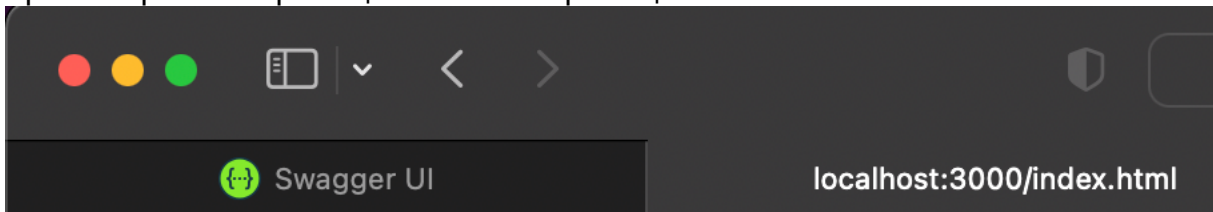
```
src > TS app.module.ts > AppModule
1  import { Module } from '@nestjs/common';
2  import { ConfigModule } from '@nestjs/config';
3  import { TypeOrmModule } from '@nestjs/typeorm';
4  import { Connection } from 'typeorm';
5  import { UsersModule } from './users/users.module';
6  import { CommentsModule } from './comments/comments.module';
7  import { PasswordsModule } from './passwords/passwords.module';
8  import { AppController } from './app.controller';
9  import { AppService, TypeOrmConfigService } from './app.service';
10 import { AuthModule } from './auth/auth.module';
11
12
13 @Module({
14   imports: [
15     ConfigModule.forRoot(),
16     AuthModule.forRoot({
17       connectionURI: process.env.SuperTokensConnectionURI,
18       apiKey: process.env.SuperTokensAPIKey,
19       appInfo: {
20         appName: "my-backend-project",
21         apiDomain: "http://localhost:3000",
22         websiteDomain: "http://localhost:3000",
23         apiBasePath: "/auth",
24         // websiteBasePath: "/auth/*",
25       },
26     }),
27     TypeOrmModule.forRootAsync({
28       useClass: TypeOrmConfigService
29     }), UsersModule, CommentsModule, PasswordsModule],
30   controllers: [AppController],
31   providers: [AppService]
32 })
33 export class AppModule {
34   constructor(private connection: Connection) {}
35 }
36
```

main.ts:

```
21 app.enableCors({
22   origin: ['http://localhost:3000'],
23   allowedHeaders: ['content-type', ...supertokens.getAllCORSHeaders()],
24   credentials: true,
25 });
26
27 app.useGlobalFilters(new SupertokensExceptionFilter());
28
```

Пример работы:

При запросе страницы без авторизации:



```
{"message": "unauthorised"}
```


Форма авторизации:

Авторизация

Email (*)

Email

Password (*)

Password

Name

Ivan

Surname

Вторник 17 мая 2022 г.

Ivanov

Sign in

Sign up

После прохождения процесса авторизации получаем access token и refresh token:

token

Все хранилище		Базы данных		Файлы cookie		Фильтровать		+ ↻ 🗑	
Локальная память — localhost	Имя	Значение	Domain	Path	Expires	Размер	Secure	HttpOnly	SameSite
Файлы cookie — localhost	sAccessToken	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCIsImVudCI6IjE5OTk0MTUzIiwiaWF0Ij0iMTU5OTk0MTUzIn0=	localhost	/	17.05.2022, 18:59:38	828 Б		✓	Lax
Хранилище сеанса — localhost	sIdRefreshToken	b0544884-96ef-4ee6-adfd-8eb10a173...	localhost	/	25.08.2022, 17:59:38	51 Б		✓	Lax

И доступ к страницам сайта (вместе с отображением имени пользователя из собственной БД):

Свagger UI

Portfolio Belonogov Evgeny M3...

js innertext — Яндекс: нашлос...

Exception filters | NestJS - A pr...

FDI | 1.13.2 | supertokens | Swa...

node.js - Accessing handlebars...

Добро пожаловать на мой сайт!

Это содержимое тега <main>:

Список достижений:

- Достижение 1
- Достижение 2
- Достижение 3

Код, которым можно гордиться:

```
int main(int argc, char* argv[]) {
    std::cout << "Lorem Ipsum";
    return 0;
}
```

МЕНЮ САЙТА

Главная

Галерея фотографий

Таблица

Страница с DHTML

Страница с Promise

Библиотека CodeMirror

Статус авторизации:
авторизован как Evgeny

Выйти

Список пользователей с сайта SuperTokens:

dev-users

userId	email	isEmailVerified	phoneNumber	timeJoined	thirdPartyProvider	thirdPartyUserId	metadata
4eb50908-bffb-4274-bdee-d49834d6c42b	admin@mail.ru	false		2022-05-16T14:25:34.644Z			{}
ca207ef9-f318-4286-a08d-805b58d09199	testuser@mail.ru	false		2022-05-16T16:43:20.895Z			{}
8e8ddbca-9e21-47c6-b6a5-095b4bc04031	evgeny@mail.ru	false		2022-05-17T13:54:23.805Z			{}

Ссылка на приложение:

<https://evgeny-backend-itmo.herokuapp.com/index.html>