



TAXIZILLA

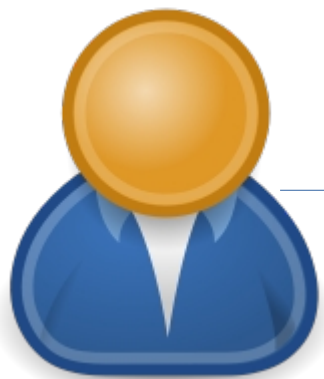
Проект №26



Какво представлява taxiZilla?



- Осъществява връзка между клиент, който си поръчва такси или доставка на покупки, и шофьор.



Каква целта на taxiZilla?



- TaxiZilla има за цел създаване на единно приложение, работещо с възможно най-много фирми и населени места.

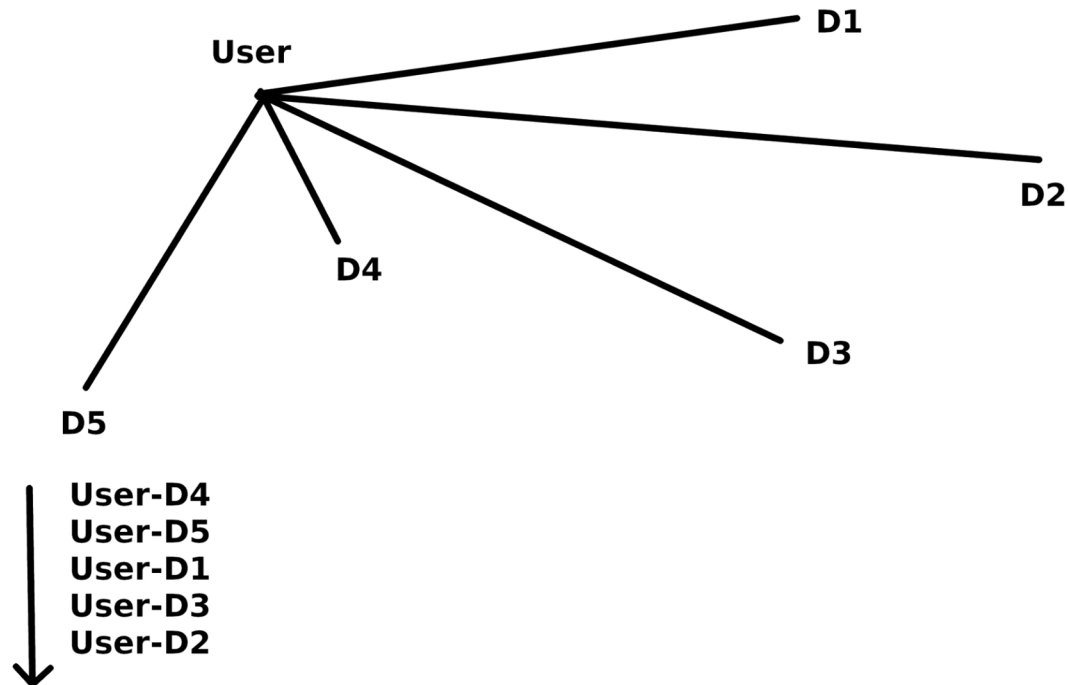




Какви са функциите на taxiZilla?

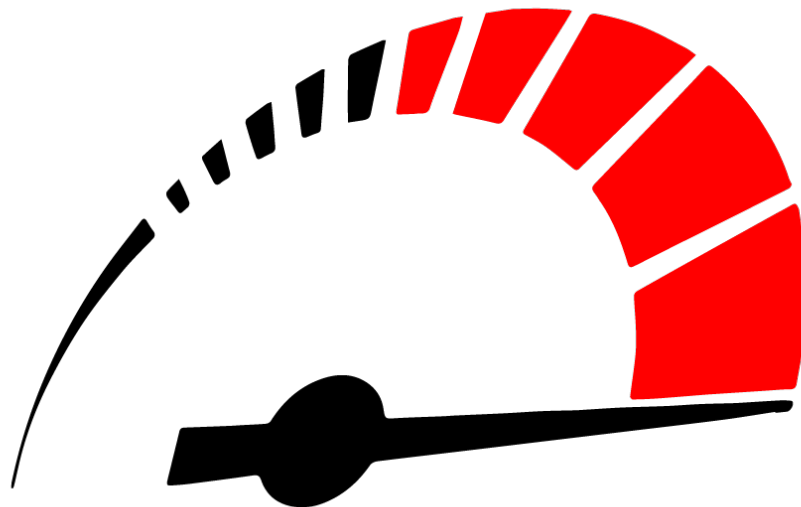


- Повикване на най-близкото такси;
- Поръчка за пазаруване с доставка до адрес.





- Да работи с фирми в цяла България;
- Оптимизиране на кода и бързодействието на платформата.
- Да бъдат добавени нови функции с цел подобряване работата на приложението.

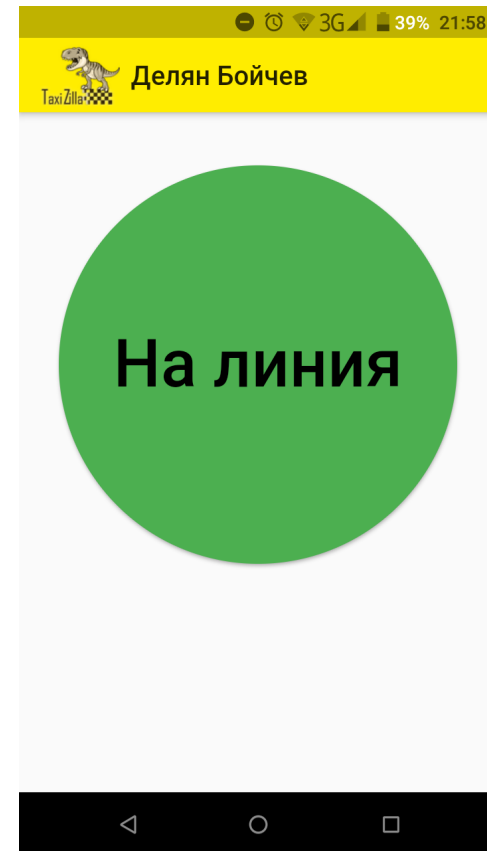
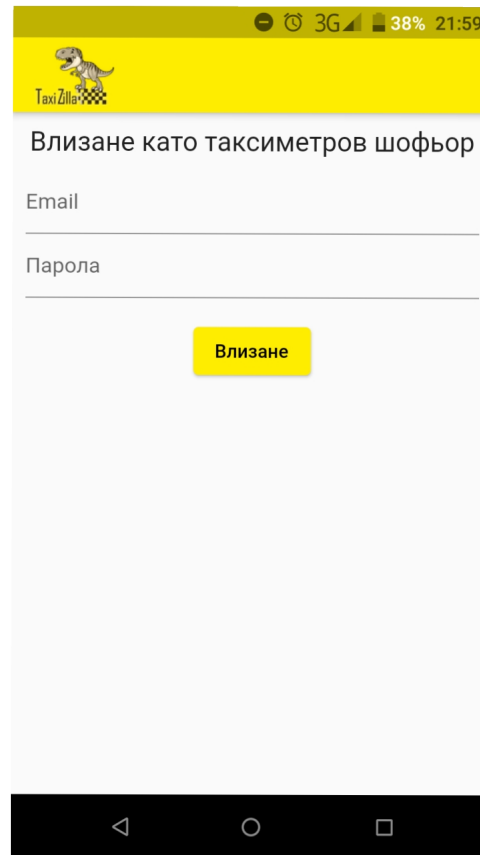




Системни изисквания за приложение за шофьор

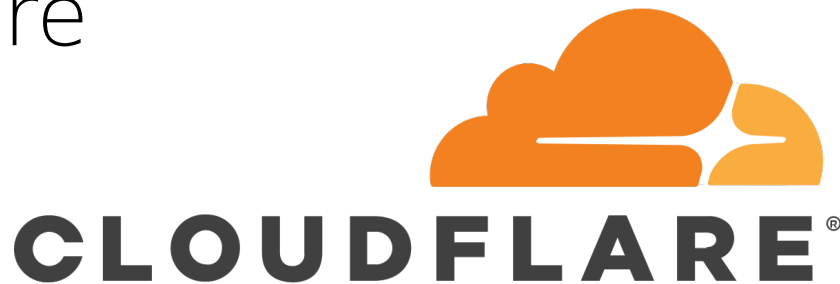


- Мобилно устройство с ОС Android 6.0 или по-нов;
- Постоянен достъп до местоположение;
- Постоянна връзка с интернет.



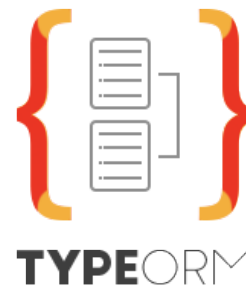
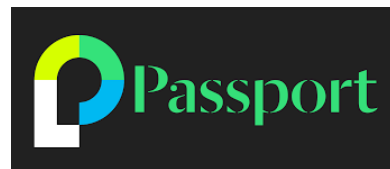
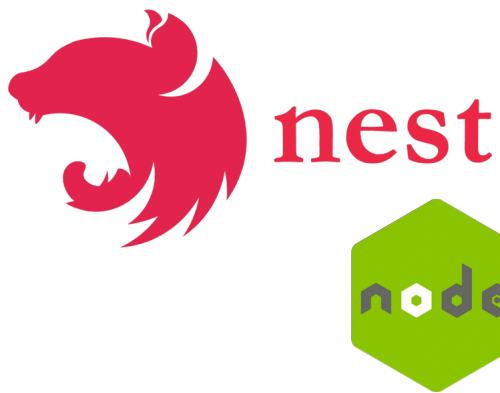


- Максимален брой заявки в секунда през ip адрес на сървъра - варират от 2000 до 2600;
- Максимален брой заявки в секунда през CloudFlare през един ip адрес - варират от 10 до 50;





- За реализиране на Backend са използвани: NestJS, MariaDB, TypeORM, Passport, NodeJS;
- За реализиране на Frontend са използвани: HTML, CSS, JavaScript, Bootstrap;
- За реализиране на приложение за шофьор е използван Flutter.





- Visual Studio Code OSS;
- LibreOffice;
- Gimp;
- NodeJS;
- Git;
- GitHub(Free);
- MariaDB.



LibreOffice®

loggedInPage.dart · workspace (Workspace) - Visual Studio Code

EXPLORER

- OPEN EDITORS
 - loggedInPage.dart taxi_zilla_driver · lib
- WORKSPACE (WORKSPACE)
 - res
 - AndroidManifest.xml
 - profile
 - AndroidManifest.xml
 - .classpath
 - .project
 - build.gradle
 - gradle
 - .gitignore
 - project
 - build.gradle
 - gradle.properties
 - gradlew
 - gradlew.bat
 - local.properties
 - settings.gradle
 - assets
 - img
 - sound
 - build
 - ios
 - lib
 - loggedInPage.dart
 - loginPage.dart
 - main.dart
 - newOrderPage.dart
 - orderConfirmed.dart
 - Session.dart
 - userFunctions.dart
 - flutter-plugins
 - flutter-plugins-dependencies
 - .gitignore
 - .metadata
 - packages
 - pubspec.lock
 - pubspec.yaml
 - README.md
 - OUTLINE
 - TIMELINE
 - DEPENDENCIES
 - JAVA PROJECTS

loggedInPage.dart

```

1 import 'dart:convert';
2 import 'package:flutter/material.dart';
3 import 'package:flutter/services.dart';
4 import 'main.dart';
5 import 'dart:async';
6 import 'userFunctions.dart';
7 import 'newOrderPage.dart';
8 import 'package:location/location.dart';
9 import 'package:http/http.dart' as http;
10
11 //Stranica kogato shofyoryt e vlrul
12 class loggedInPage extends StatefulWidget {
13   @override
14   _loggedInState createState() => _loggedInState();
15 }
16
17 //State na stranica
18 class _loggedInState extends State<loggedInPage> with WidgetsBindingObserver {
19   var statusButtonText = "На линия";
20   var statusButtonColor = Colors.green;
21   //Proverka za poruchki na vseki 3 sekundi
22   checkForOrders() async {
23     Location location = new Location();
24     bool _serviceEnabled;
25     var citySupported;
26     var locData;
27     var o;
28     while (true) {
29       SystemChrome.setEnabledSystemUIOverlays([SystemUIOverlay.bottom]);
30       _serviceEnabled = await location.serviceEnabled();
31       if (!_serviceEnabled) {
32         _serviceEnabled = await location.requestService();
33       } else {
34

```

PROBLEMS 11 OUTPUT DEBUG CONSOLE TERMINAL

1: bash

```

[Next] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped (/firm/getAllFirms, GET) route +0ms
[Next] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped (/firm/addTaxiDriverByAdmin, POST) route +10ms
[Next] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped (/firm/changePassword, POST) route +1ms
[Next] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped (/order/resolver, POST) route +1ms
[Next] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped (/order/getOrderOneSender, GET) route +0ms
[Next] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped (/order/rejectOrderAfterAccept, POST) route +1ms
[Next] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped (/order/createsOrder, POST) route +0ms
[Next] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped (/order/acceptOrder, POST) route +1ms
[Next] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped (/order/rejectOrder, POST) route +0ms
[Next] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped (/order/finishOrder, POST) route +1ms
[Next] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped (/order/getAllOrders, GET) route +0ms
[Next] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped (/order/getOrdersByUser, GET) route +1ms
[Next] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped (/order/getOrdersByDriver, GET) route +0ms
[Next] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped (/order/removeOrder, POST) route +1ms
[Next] 14956 - 07.02.2021 r., 22:42:54 [NextApplication] Nest application successfully started +3ms

```

Ln 32, Col 51 Spaces: 2 UTF-8 LF Dart Flutter: 1.22.6 MDX (android-arm64) 23:29



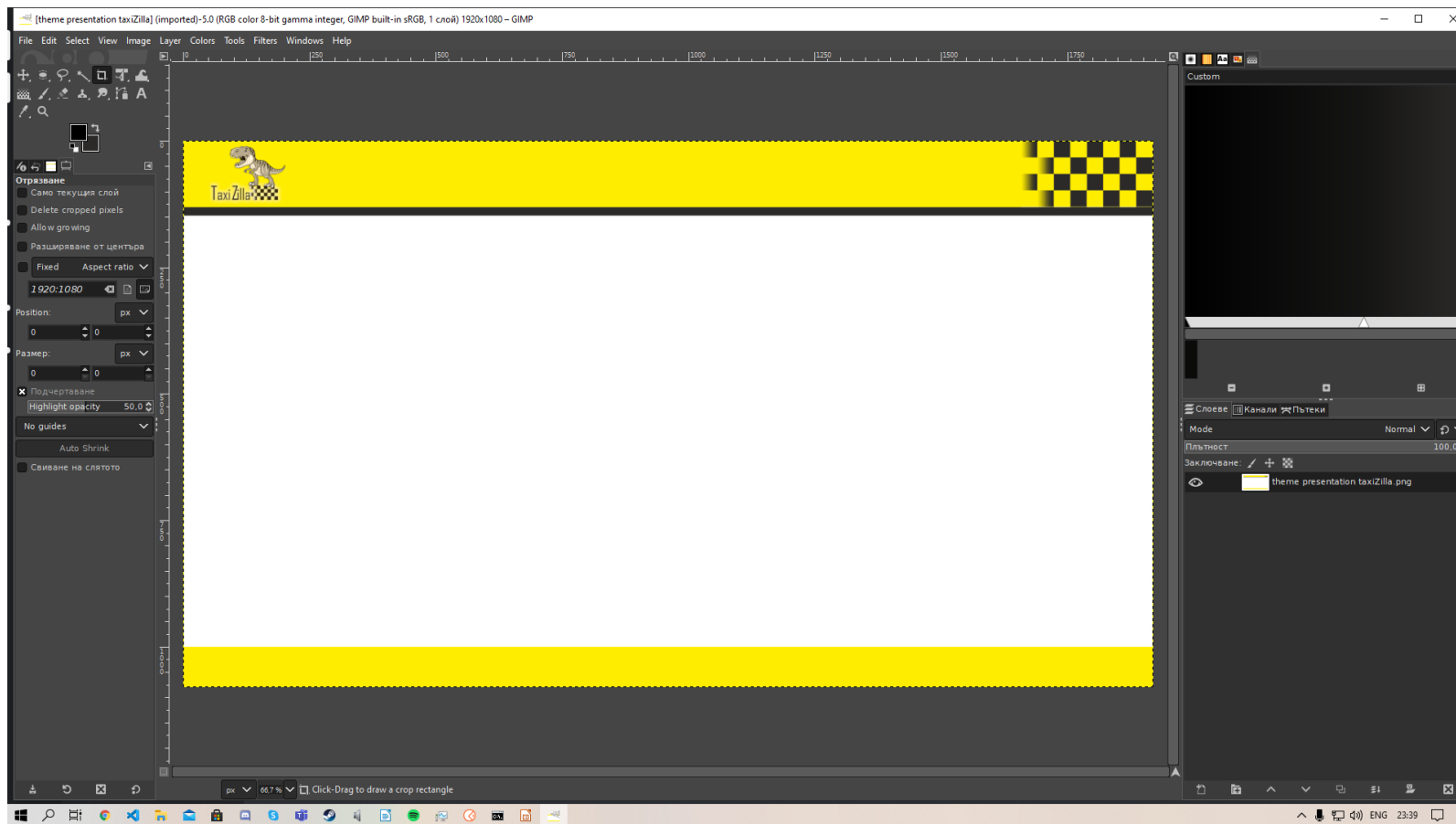
Visual Studio Code interface showing the development of the TaxiZilla application. The Explorer pane on the left displays the project structure, including the 'src' directory and the 'auth' module. The main editor displays the 'auth.controller.ts' file, which contains the implementation of the authentication and authorization logic. The code includes endpoints for getting cities by firm ID, changing status and location, and exiting a taxi driver. The code is written in TypeScript and uses the Express.js framework.

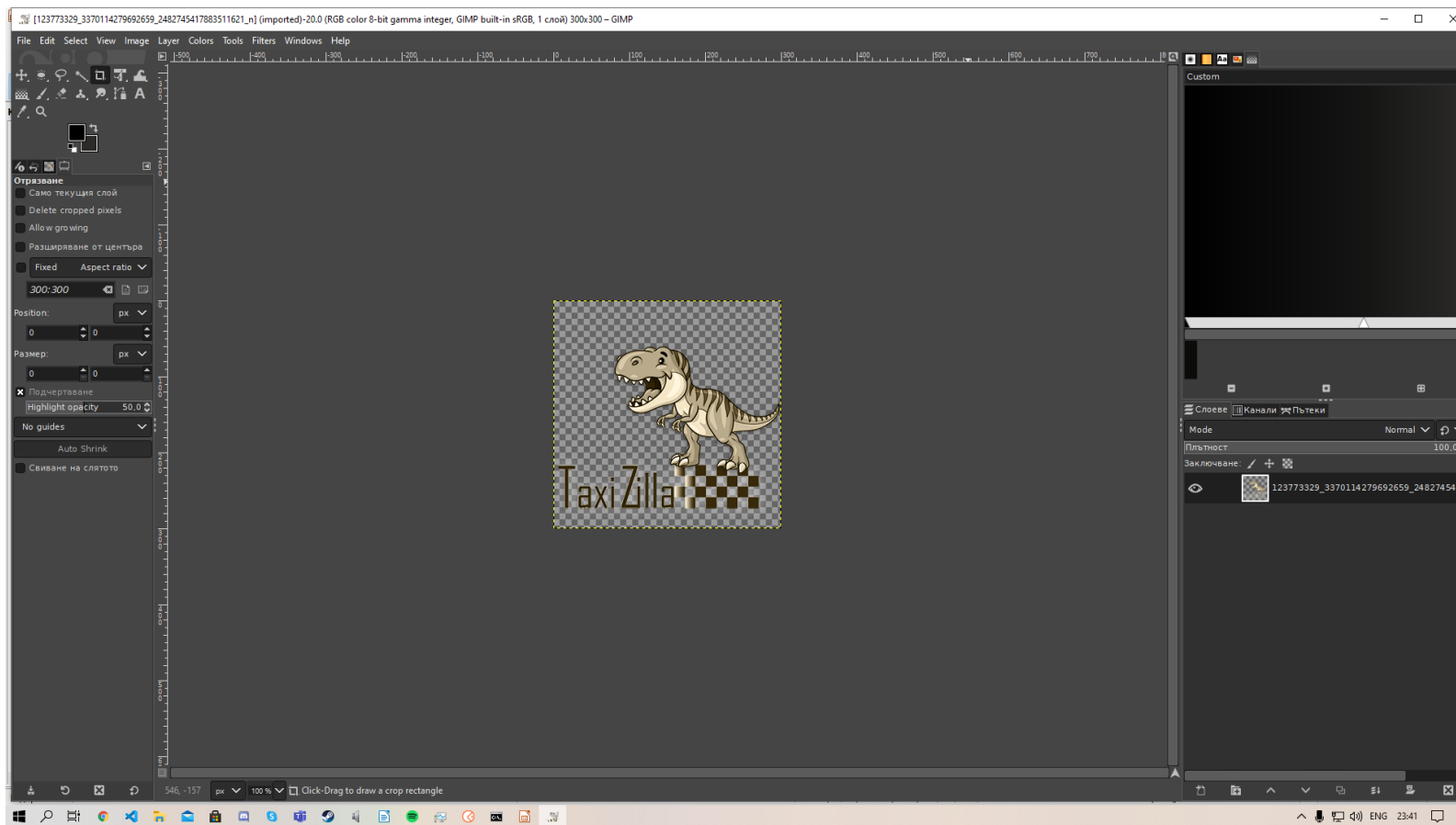
```
taxiZilla > src > auth > TS auth.controller.ts > AuthController > changeStatusAndLocation

168 @Post("/getCitiesByFirmId")
169 async getCitiesByFirmId(@Session() session:{token?:string}, @Body("firmID") firmID:string)
170 {
171     if(!session.token) throw new UnauthorizedException();
172     return await this.authService.getCitiesByFirmId(parseInt(firmID));
173 }
174 //Промяна на статус и проверка за поръчки. Само за шофьори
175 @Post("/changeStatusAndCheckForOrders/")
176 async changeStatusAndLocation(@Session() session:{token?:string, role?:UserRoles}, @Body("newStatus")newStatus:UserStatus, @Body("x") x:string, @Body("y") y:string)
177 {
178     if(!session.token && session.role != UserRoles.DRIVER)throw new UnauthorizedException();
179     let date = new Date();
180     return this.authService.changeStatusAndLocation(session,newStatus,parseFloat(x),parseFloat(y));
181 }
182 @Post("/exitTaxiDriver/")
183 exitTaxiDriver( @Body("key") key:string, @Body("driverID") driverID:string)
184 {
185     if(!key) throw new UnauthorizedException();
186     if(key.length!=19) throw new UnauthorizedException();
187     const date = new Date();
188     const d = new Date(Date.UTC(date.getUTCFullYear(), date.getUTCMonth(), date.getUTCDate(), date.getUTCHours(), date.getUTCMinutes(), date.getUTCSeconds()+3 ));
189     const str = this.authService.decode(key);
190     const d2 = new Date(Date.UTC(parseInt(str.substr(0, 4)), parseInt(str.substr(6, 2))-1, parseInt(str.substr(4, 2)), parseInt(str.substr(8, 2)), parseInt(str.substr(10, 2)), parseInt(str.substr(12, 2)), parseInt(str.substr(14, 2))));
191     const d3 = new Date(Date.UTC(date.getUTCFullYear(), date.getUTCMonth(), date.getUTCDate(), date.getUTCHours(), date.getUTCMinutes(), date.getUTCSeconds()-3));
192     if(d2.toString() == "Invalid date") throw new UnauthorizedException();
193     if(d2.getTime()>d.getTime() || d3.getTime()>d.getTime()) throw new UnauthorizedException();
194     return this.authService.exitTaxiDriver(parseInt(driverID));
195 }
196 //Взимане на всички потребители
197 @Get("/getAllUsers")
198 async getAllUsers(@Session() session:{token?:string, role?:UserRoles, type?:string})
199 {
200     if(!session.token && session.role != UserRoles.ADMIN)throw new UnauthorizedException();
201 }
```

Problems (11) OUTPUT DEBUG CONSOLE TERMINAL

```
[Nest] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped {/firm/getAllFirms, GET} route +0ms
[Nest] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped {/firm/addTaxiDriverByKdmin, POST} route +10ms
[Nest] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped {/firm/changePassword, POST} route +1ms
[Nest] 14956 - 07.02.2021 r., 22:42:54 [RoutesResolver] OrderController {/order}: +1ms
[Nest] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped {/order/getOrderOneSender, GET} route +0ms
[Nest] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped {/order/rejectOrderAfterAccept, POST} route +1ms
[Nest] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped {/order/createOrder, POST} route +0ms
[Nest] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped {/order/acceptOrder, POST} route +1ms
[Nest] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped {/order/rejectOrder, POST} route +0ms
[Nest] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped {/order/finishOrder, POST} route +1ms
[Nest] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped {/order/getAllOrders, GET} route +0ms
[Nest] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped {/order/getOrdersByUser, GET} route +1ms
[Nest] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped {/order/getOrdersByDriver, GET} route +0ms
[Nest] 14956 - 07.02.2021 r., 22:42:54 [RouterExplorer] Mapped {/order/removeOrder, POST} route +1ms
[Nest] 14956 - 07.02.2021 r., 22:42:54 [NestApplication] Nest application successfully started +3ms
```







Благодарим за
вниманието!