

GoTo Bus

Introduction

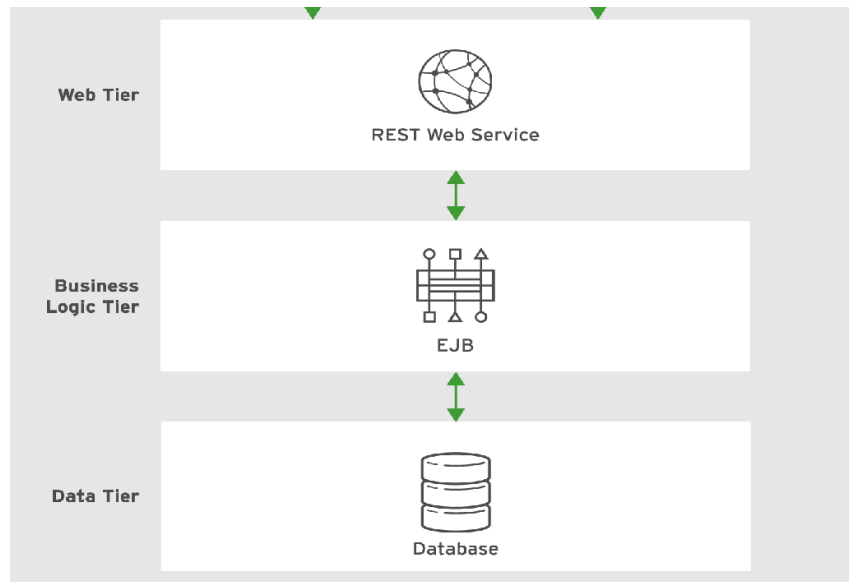
GoTo Bus is a company for operating bus trips in Egypt, They have several trips per day all over Egypt. They have one station in every Governorate so they need to reduce labor and human error by digitalizing the way they operate. So they decided to request from your team to start developing a BackEnd system that provides several Web Services that satisfies their business needs.

System Functionalities:

- System should provide user management subsystem for New User Registration and Login
- User can have two different roles Client and Administrator roles
- System provides a Station entity which describes the name of the station and its map coordinates
- System provides a trip entity which describes the from and to stations, datetime of the trip and number of seats
- Trip Entity Management is done **by the Admin role only**
- Users and Non-Users can search for trips using from and to stations and from and to dates showing the trip details including number of seats available
- Logged in Users can book a trip if available seats exists
- User should get a notification **asynchronously** when the booking is done successfully
- Entities of the system: User, Station, Trip, Notification

Important Note: GoTo Bus is providing API references for the requested functionalities, You can find them at the end of this document.

System Architecture



Project Guidelines (Please read carefully):

- You should **strictly follow** the API references, As We might run automatic test cases based on these APIs references so please try to avoid any misspelling and follow the naming conventions
- You should follow the system design with correct EJB types (stateless, stateful and message driven)
- You can use any relational database, but don't use in-memory database (Recommended DB: SQLite)
- You **must use** GIT for source code management and create a **private** repository on GitHub
- Teams should be of 3 - 4 members from the same group
- Each team member contribution will be calculated from GitHub, Team members with low contribution will get low grades
- Deadline is **01/06/2022 11:59 PM**
- You have to create a .docx file Named **Project_Information.docx** containing the Repository link and a table containing team members' information (Name, ID, Github Username),
- You have to compress the **.docx file and the Source code** to a zip file named **SLB_ID1_ID2_ID3_ID4.zip** and submit it on BB before the deadline.
- Plagiarism checks will run automatically on your code, Any cheating case, **Both teams will receive negative grades**
- Bonus Tasks:
 - Create a Frontend Application using any Javascript Framework (Angular, React, VueJS) including the GUI of all the requested functionalities **(+2 Marks)**
 - Deploy the Backend system on the cloud and allowing accessing it through the Public IP of the cloud virtual machine **(+2 Marks)**
 - Wrap the application into a .ear file **(+1 Mark)**
- **Note:** Please don't implement any of the bonus tasks unless you are delivering the full requested functionalities otherwise you will not get the grade of the bonus task

APIs Reference

Name: Create User
URL: /api/user
Method: POST
Parameters: None
Request <u>Body:</u> <pre>{ "username": "mohamed", "password": "test123", "full_name": "Mohamed Ahmed", "role": "client"}</pre>
Response <u>Status:</u> 200 OK

Name: Login
URL: /api/login
Method: POST
Parameters: None
Request <u>Body:</u> <pre>{ "username": "mohamed", "password": "test123",}</pre>
Response <u>Status:</u> 200 OK (If correct) <u>Status:</u> 400 (If not correct)

Name: Create Station
URL: /api/station
Method: POST
Parameters: None
Request <u>Body:</u> <pre>{ "name": "Aswan", "longitude": "23.5184815151", "latitude": "21.51548481515" }</pre>
Response <u>Status:</u> 200 OK (If correct) <u>Status:</u> 500 (If exception happened)

Name: Get Station
URL: /api/station/{id}
Method: GET
Parameters: id
Request <u>None</u>
Response <u>Status:</u> 200 OK (If correct) <u>Status:</u> 500 (If exception happened) Body: <pre>{ "name": "Aswan", "longitude": "23.5184815151", "latitude": "21.51548481515" }</pre>

Name: Create Trip
URL: /api/trip
Method: POST
Parameters: NONE
Request <u>Body:</u> <pre>{ "from_station": "Cairo", "to_station": "Aswan", "available_seats": 50, "departure_time": "25/05/2022 15:00:00", "arrival_time": "25/05/2022 20:00:00" }</pre>
Response <u>Status:</u> 200 OK (If correct) <u>Status:</u> 500 (If exception happened)

Name: Search Trips
URL: /api/searchtrips
Method: POST
Parameters: NONE
Request <u>Body:</u> <pre>{ "from_date": "25/05/2022", "to_date": "28/05/2022", "from_station": "<STATION_ID_OF_CAIRO>", "to_station": "<STATION_ID_OF_ASWAN>", }</pre>
Response <u>Status:</u> 200 OK (If correct) <u>Status:</u> 500 (If exception happened) <u>Body:</u> <pre>[{ "trip_id": "1", "from_station": "Cairo", "to_station": "Aswan", "departure_time": "26/05/2022 15:00:00", "arrival_time": "27/05/2022 01:30:00", "available_seats": 36 }, { "trip_id": "2", "from_station": "Cairo", "to_station": "Aswan", "departure_time": "26/05/2022 15:00:00", "arrival_time": "27/05/2022 01:30:00", "available_seats": 25 }, { "trip_id": "1", "from_station": "Cairo", "to_station": "Aswan", "departure_time": "26/05/2022 15:00:00", "arrival_time": "27/05/2022 01:30:00", "available_seats": 35 }]</pre>

Name: Book a trip
URL: /api/booktrip
Method: POST
Parameters: NONE
Request <u>Body:</u> <pre>{ "trip_id": 2, "user_id": 5 }</pre>
Response <u>Status:</u> 200 OK (If correct) <u>Status:</u> 500 (If exception happened)

Name: View User's Trips
URL: /api/viewtrips/<user_id>
Method: GET
Parameters: User ID
Request <u>Body:</u> NONE
Response <u>Status:</u> 200 OK (If correct) <u>Status:</u> 500 (If exception happened) <u>Body:</u> <pre>[{ "trip_id": "1", "from_station": "Cairo", "to_station": "Aswan", "departure_time": "26/05/2022 15:00:00", "arrival_time": "27/05/2022 01:30:00", "available_seats": 36 }]</pre>

Name: Show user notifications
URL: /api/notifications/{user_id}
Method: GET
Parameters: User ID
Request <u>Body:</u> NONE
Response <u>Status:</u> 200 OK (If correct) <u>Status:</u> 500 (If exception happened) Body: [{ "message": "You have booked trip from Cairo to Aswan successfully", "notification_datetime": "23/05/2022 12:00:00" }, { "message": "Sorry, Trip Cairo to Luxor have no available seats", "notification_datetime": "23/05/2022 12:05:00" }]