



Please read the following requirements carefully and submit your code in the form you think is best.

## Sr. Backend Developer

### Node.js Task



---

#### **Barcelona**

Calle Calabria 169  
10th floor  
08015 Barcelona  
SPAIN

#### **Madrid**

Av. Manoteras 26  
5th floor  
28050 Madrid  
SPAIN

#### **Menlo Park**

101 Jefferson Drive  
1st floor  
94025 Menlo Park, CA  
UNITED STATES OF AMERICA

Whisbi would like to build a simple events application where some users can create and post their events and other users can subscribe to them. We would like to see how would you build the backend of such a simple yet essential application,

### Functional Requirements

1. Users should be able to sign-in and sign-out of the application.
2. Only authenticated users should be able to post, edit or delete events.
3. Events can be edited or deleted after being published.
4. Users can only edit and delete their own events.
5. Users who are not logged in can only view some events but can not subscribe to an event.
6. Only users who are logged in can subscribe to any event.
7. Each event should have a headline, description, start date and location.
8. Each event has three states: Draft, Public and Private.
9. Users can only have one published event at a time and can subscribe to a maximum of 3 events.
10. Users should **not** be able to subscribe to their own events.
11. A user subscribes to an event by sending their personal details (name, email and any other details you think should be necessary) to the service.
12. Users should receive a notification 24 hours before the start date of each event reminding them that it's happening the next day.
13. Users should see a list of all events (as much as the rules allow) in the home.
14. Each event should have its own page to display all details and a subscribe functionality (as described in point #11) below the event details.

### Technical Requirements

1. Implement the REST API for this application using Node.js, Express and an ORM of your choice.
2. Implement proper security as you see fit for the application's authentication.
3. Use a middleware for the custom usability rules.
4. Use SOLID principle, linting and a convention for your code.
5. Add tests where / how you see fit.
6. Protect the application against Denial of Service attacks.
7. Send the reminders notifications using sockets.
8. Comment your code where you see is fit and include a README file with your submission.
9. Creating a bare UI to test the backend is **not** required but a nice to have.
10. Containerize your application if you think it is necessary..

*After reading these requirements please send us back the time you need to develop this solution. Keep in mind that a quick submission will give you an advantage.*

