Consider an application, a simpler version of our application.

We have those entities:
- Client - is a company with a name, a description, type(DOMESTIC, INTERNATIONAL) and phone fields;
- Client can have one or more Investors that they represent;
- Investor manages one or more funds;

Backend
We would want couple of REST end-points:
- Get list of clients;
- Get list of investors for a client;
- Get list of funds for an investor;
- Update or create a client.

Additional requirements:
- It's expected to have server side filtering for list of clients (e.g. triggered by user from UI);
- If Client is INTERNATIONAL, it's expected that client will have phone number provided;

Please feel free to make assumptions on any other key factors. Make sure to document your assumptions in the README.txt file in the project.

Expectation for this task:
1. Spring boot application with single module;
2. Standard Java (and /or apache) libraries and collections are to be used;
3. Choice of JDK 1.8+;
4. Rest API end-point to build and retrieve the data;
5. Simple JSON output is expected;
6. Unit tests to test the key aspects of the logic is expected (Client related logic test coverage is expected);
   - Unit test for service layer;
   - Unit test for repository layer;
7. Feel free to use any SQL DB of your choice or in-memory DB;
8. Build with maven or gradle;
9. Check into git repo of your choice and send us the link once done.

BONUS points:
- Add validation in place for client, investor and fund entities. Their names and description cannot start with character '-', '+', '='
- Create a UI app to work along with Backend. Use provided basic angular app (modify/improve as needed) or use another framework of your choice (such as React)

UI requirements.
Have a simple dashboard with links that would allow:
- Display list of clients;
- When client is clicked load list of Investors;
- When Investor is clicked load list of funds;
- Ability to create and edit a new client.