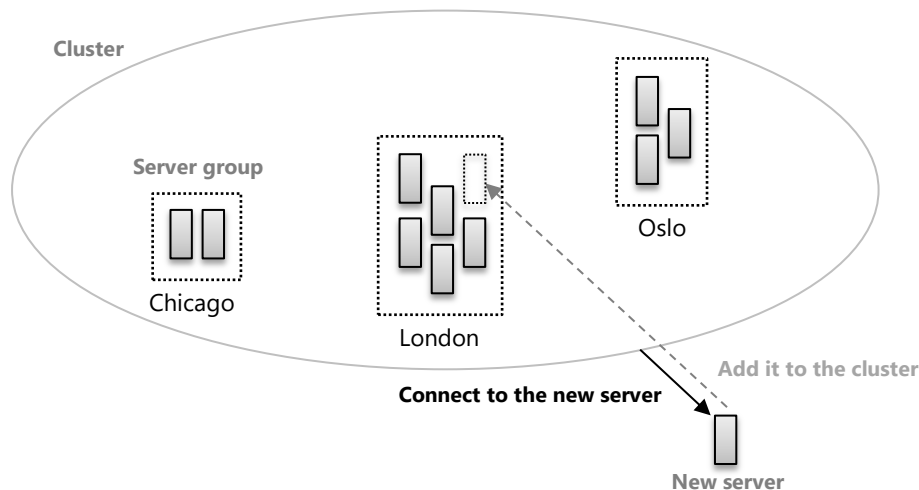


Scenario

Our product comprises a cluster of servers, grouped together by geographical location (e.g. office building, site, city or country). The mobile app is already connected to the cluster.

You are creating a simplified version of the *'add server'* flow from our desktop application. The goal is to enable an administrator to connect a new server to the cluster.

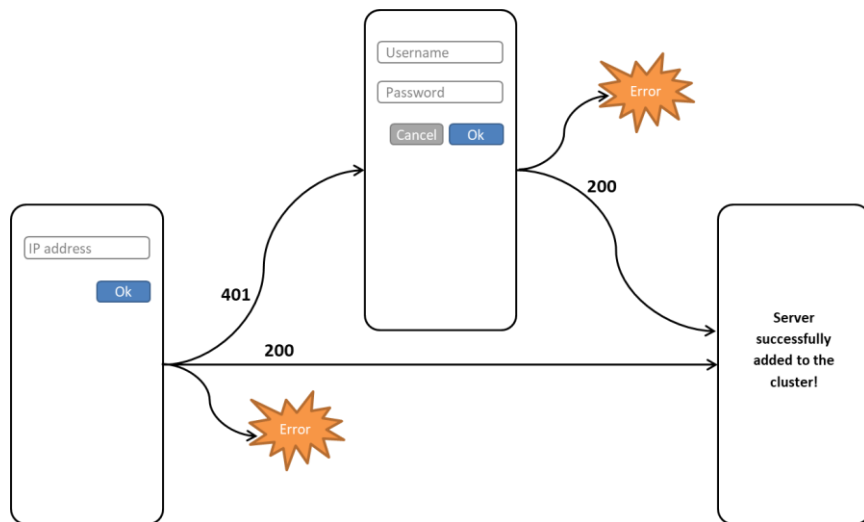


The server may be brand new out of the box (no credentials required), or it may already have been configured with a username and password.

When attempting to connect to a server, the http response code will indicate the server state:

- 200 (success): credentials correct or not required
- 401 (unauthorized): credentials invalid or required

The app should prompt the user for the IP address of the server. On submission, it should try to connect without credentials. If successful, it should show a confirmation page. If the server returns 401 (unauthorised), it should prompt the user to enter a username and password and then reattempt connection.



Resources

You are provided with a **Networking** class to act as your networking layer, which contains mocked functionality to connect to the new server. Use the **connectToServer** function to attempt to connect to the new server.

The networking class contains additional functions, but you won't need them for this exercise.

It has also been configured with two fake servers which you can test with:

- **192.168.1.10** – no credentials required
- **192.168.1.11** – requires credentials (username: **vaion**, password: **password**)

Challenge

Create a simple three-page application for this flow, using any tech stack you like.

The app should meet the following criteria:

- Allows the user to add a new server through the flow described above.
- Must use the *Networking* layer provided, but feel free to use this as is or wrap it up in a layer of your own
- Handles basic error cases.
- Requests can be slow, it should manage the delay between request and response

Our Guidance

The challenge should not take any more than 2 hours. You do not need to complete it in one go.

Although the app might be relatively straightforward, please try to write the code as if you were building something more complex. We would like to gain an idea of how you would go about structuring the application.

Feel free to make any changes to the UI you see fit, although please don't prioritise styling! Something basic will do just fine.

Do you test your code? We'd love to see any indicator of TDD.

When you are finished, you should send us a link to the codebase, preferably via git, showing multiple commits so we can see its evolution.