

Aon Pre-Interview Questionnaire

Part 1 Opinion Tasks

The first three tasks ask you about your opinion. You do not need to code here. Please restrict your answer to max 1500 characters per answer. If you can write a good answer with less text, that is even better.

Task 1 - Trends

Describe a trend which has happened with enterprise applications over the last two or three years. How do you see this trend influencing corporate application development? What are you doing to prepare for this trend?

Suggested effort: up to 15 minutes. Please do not write more than 1500 characters

Task 2 - Describe Software that Delivers User Value

Describe a product or project you worked on that delivered high value to the user. Which specific aspects did you think were critical in successfully creating value for the user?

Suggested effort: up to 10 minutes. Please do not write more than 1500 characters

Task 3 - Your Values

What are your core values / principals as a software engineering professional working in a team environment and why are they important to you?

Suggested effort: up to 5 minutes

Part 2 Programming Task

The next page describes the business requirements you need to develop a solution to meet. You will need to code the solution and any test cases you feel appropriately prove your solution meets the requirements.

Task 4 - Coding

Imagine a business insurance platform which facilitates a transaction between a customer and a panel of insurers. All insurers are able to offer quotes to the customer based on the questions answered by the customer.

A new requirement is that insurers don't want provide quotes when certain criteria are met, and thus should be removed from the panel for that transaction.

Your task is to create a restful **customer API** in Java or Groovy that allows the customer to request a quote by submitting all of the following required values:

- A **postcode**
- An **occupation** (e.g. "Plumber", "Butcher", etc.)
- The **annual turnover** of their business in dollars (e.g. \$400,000)

The customer API will return a list of quotes from 4 or more insurers. A quote simply contains the insurer's name and a (mock) price. Use a data structure suitable for a front-end to display.

The customer API will use set of business rules to check the customer values against criteria for each insurer to determine whether the insurer should be excluded from the results of the customer request.

Each insurer will have the following data stored in the application:

- The insurer's name
- A list of excluded postcodes (optional)
- A list of excluded occupations (optional)
- An minimum turnover that must be meet (e.g. \$100,000) (optional)

The business rules to exclude an insurer are:

- the postcode **and** occupation are **both** in an insurer's excluded lists **or**
- the turnover is less than the insurer's minimum turnover

Your design should consider the persistence of these criteria, but the implementation can be an in memory data structure or in memory DB.

Your design should also consider that the business rules will change in the future (i.e. new criteria or new rules). How could you reduce the cost of such changes?

Example insurer exclusion criteria:

Insurer	Excluded Postcodes	Excluded Occupations	Minimum Turnover
Insurer1			
Insurer2	2000, 2001	Butcher	
Insurer3	2000	Butcher, Plumber	200000
Insurer4			400000

Customer queries and the expected results based on the exclusion criteria above:

Query			Result			
Postcode	Occupation	Turnover	Insurer1	Insurer2	Insurer3	Insurer4
2000	Builder	\$600,000	✓	✓	✓	✓
2000	Builder	\$100,000	✓	✓	* turnover	* turnover
2000	Plumber	\$600,000	✓	✓	* postcode & occupation	✓
2001	Butcher	\$300,000	✓	* postcode & occupation	✓	* turnover

✓ Indicates a quote is returned. * Indicates a quote was not returned and the reason why.

The implementation is up to you: Spring MVC, Spring Boot, Vert.x or something else.

Email the source, build script, tests, etc., as zip of your local Git repo, or a zip of the files.