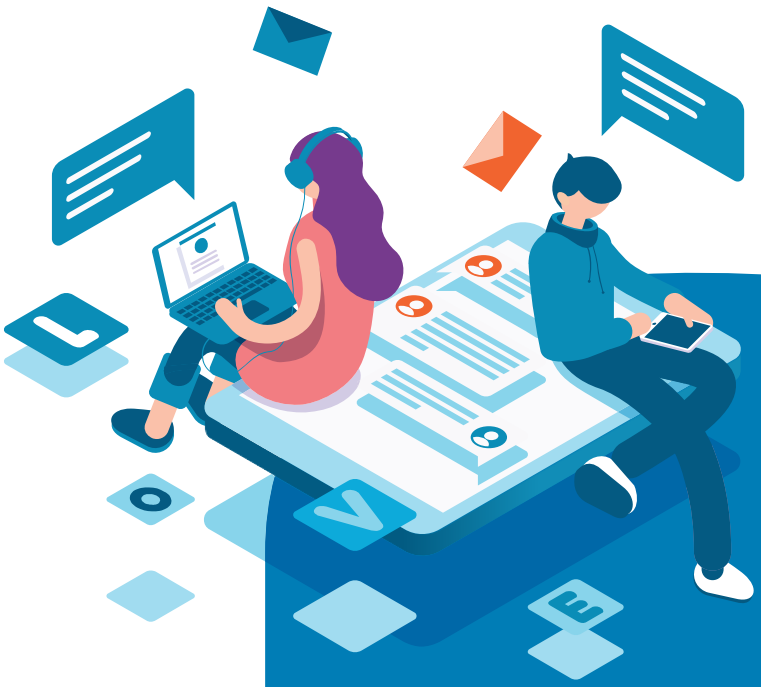
sogeti
Part of Capgemini

Java Backend Assessment



Introduction

This is the Sogeti Back-end developer assessment. The assessment is intended to give the reviewer insight into your technical abilities, development approach and general technical working habits. We view your performance on this assessment as indicative of the work you will deliver as a Sogeti Back-end developer.

The assessment can be made privately and is to be later discussed during an interview. We have prepared a business case, further described below. Read the case carefully and approach it as if it was a real project. We expect that you will need about 5 to 8 hours to finish the assessment. For the assessment discussion, please bring a laptop with a working development environment so that we can look at parts of the application together. Also, we expect you to provide the source code and some documentation before the interview.

The assignment should be implemented in Java.

Make sure to prepare yourself to tell something about the following subjects:

- the overall approach you took to implement the assignment
- the architecture of the solution
- the technologies used for your solution

Good luck with the assignment!

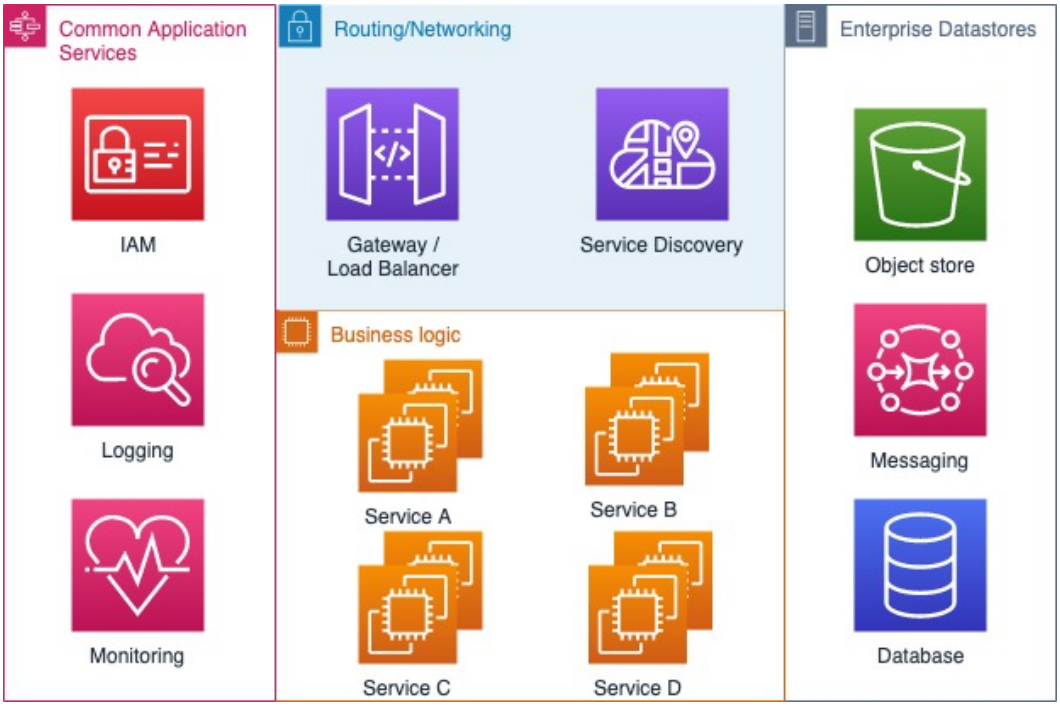
The assessment

The goal of this assessment is to build a simple first version of a **Car-lease Platform API**. **The Car-lease Platform API** should be built according to the microservices principles.

What is a microservice?

Microservices are small, autonomous services that work together. Let's break that definition down a bit and consider the characteristics that make microservices different.

Below you see a generic architecture on how you can build and assemble a microservice application.



Car-lease Platform API

The **Car-lease Platform API** is a REST API that allows to maintain vehicle versions, customers and other data needed to service a broker.

The end-users of the **Car-lease Platform API** are:

- Brokers that calculate the leaserate for a customer and maintain customer data.
- Leasing company that maintains data to make an accurate calculation.

Functional Requirements

The **Car-lease Platform API** has the following *functional requirements*:

- You can maintain (read, add, change, and delete) basic customer attributes:
  - Name
  - Street
  - House number
  - Zip code
  - Place
  - Email address
  - Phone number

- You can maintain basic car attributes:
    - Make
    - Model
    - Version
    - Number of doors
    - CO2-emission
    - Gross price
    - Nett price
- The leaserate is depending on the following parameters:
    - Mileage - the amount of kilometers on annual base
    - Duration - the number of months in the contract
    - Interest rate with startdate
    - Nett price
- Leaserate is calculated as:
    - (((mileage / 12)*duration)/Nett price) + (((Interest rate / 100) * Nett price) / 12)
- Example calculation:
    - Mileage: 45000 km/yr
    - Duration: 60 months
    - Interest rate: 4.5%
    - Nett Price: € 63000
    - **Leaserate: € 239,76 per month**
- The broker and employees who keep track of the data must log into the API before any subsequent call can be made.
- The identity should be validated with every call.

Non-functional Requirements

- The **Car-lease Platform API** is built using a modern Java version with a current Spring Boot version.
- Maintainability is favoured over performance. No complex performance optimisations should be needed. Another developer should be able to continue where you left off.
- It should contain service-2-service communication.
- Add token-based security (oauth/jwt).
- An in-memory datastore can be used in this first version.
- Document your code (javadoc).
- Write unit tests for your code.
- Use Google Java Style Guide.

Bonus objectives

The following are not required, but can be seen as nice-to-have's:

- Use Swagger to document to REST API.
- Store the data in a persistent datastore.
- Make the service ready for containerisation.

Deliverables

This assessment should be delivered in the following way:

- All code is pushed to a git-like (personal) repository (github/gitlab/bitbucket).
- Documentation is provided in the README.md on how the API works, and how to run it.
- Any information, (dummy)-data, files, and other assets that are needed to run this API, are provided in this repository.

Tips and remarks

- Think of what you're building is a real **Product**. Think of your end-users and what they want.
- Choose a suitable data structure for modeling the car and customer data.
- Design your API in an intuitive way, design it according to the best practices of the API technology you choose.
- Work in an agile way! You might not be able to implement all the features for this assessment in the allotted time, so be smart in picking the features you work on first.

About Sogeti

Sogeti is a leading provider of technology and engineering services. Sogeti delivers solutions that enable digital transformation and offers cutting-edge expertise in Cloud, Cybersecurity, Digital Manufacturing, Digital Assurance & Testing, and emerging technologies. Sogeti combines agility and speed of implementation with strong technology supplier partnerships, world class methodologies and its global delivery model, Rightshore®. Sogeti brings together more than 25,000 professionals in 15 countries, based in over 100 locations in Europe, USA and India. Sogeti is a wholly-owned subsidiary of Capgemini SE, listed on the Paris Stock Exchange.

Learn more about us at www.sogeti.com