

Java Coding Challenge

The goal of this challenge is to give you an opportunity to show us what you know and how you would work in a real-life project.

General aspects

We expect you to fork this github repository that contains the instructions, project structure and basic assets.

Please always commit and push over your fork and when you are done create a pull request to this repository to deliver the results.

In order to be practical, we ask you to make one commit per challenge part and use the comment to identify it. I.e: commit description "A" should have all the changes to answer the challenge part A.

To simplify things we added some necessary assets and configuration needed in `src/main/resources` folder.

After cloning the repository, you should be ready to start up the cars application at port 8080 with the command **`mvn springboot:run`**

The application technology

Here is the list of the technologies we are providing in the application structure:

Java 8

Spring boot 2.1.4

Maven 3.3.9 or above

Tomcat server 9 (embedded)

H2 embedded database

Spring Data JPA

Be free to add any necessary dependency you need to achieve the challenge but please, keep the above technologies and versions in the project as they already are.

Challenge introduction

Our customer is a big car vendor in Bangalore with many local branches all over the India.

He asked for a service that enables the current seller's application in the different local branches to consume the car specifications.

Nowadays the car specifications comes from the different car factories in different formats, the idea is to unify the data and expose it with a standard interface like a RESTful webservice.

A - The entities

You can find an example of Ford's car file specification named `ford-example.xml` within the main resources of the car application.

Create a Database schema diagram to explain how you will represent the information in that file. Remember to create a normalized data structure, entities and relationships.

Then proceed to create the Models/Entities.

Please provide the diagram as an image and include it in this answer commit.
You can also use the ANSWERS.md file to explain in more detail if you want.

B - Ingest the data

Our customer wants to start ingesting the Ford's car specifications that comes in XML format.

Use Spring Interface **CommandLineRunner** to insert **basic data of the cars present in the xml file**.

We recommend to implement all the necessary using JPA entities and Spring repositories in order to persist the data.

C - Expose data with a RESTful API

Create the following RESTful API endpoints to:

1. Create a new Car
2. Get a car specification by **id**
3. Get all the car specifications by **brand**
4. Get All cars **list**
5. Update a Car entity by **id**
6. Update a Car entity by **name**
7. Delete a car by **id**

All the endpoints must expose the results in JSON format.

D - Adding images

Explain (no code needed) how you would do to add provided image files to the part B process.
Consider that they want to include them as part of the responses of the endpoints in part C.

E - Improvements

Is there anything that you consider that can be improved in your solution? Please explain

Deliverables

The client is going to code review the changes, so you must create a pull request.

FAQ

- Should I deliver clean code and apply design patterns and good practices?
 - Definitely yes! We will evaluate those aspects.
- Do I need to add tests?
 - Tests are always important! In the case of this challenge they are not mandatory, but always test what you consider important.
- Do I need to use some specific libraries?
 - No, build the application with the tools you prefer, there are no limits. Just keep the ones that are already provided.