

Coding Challenge: AutoScout24 Listing Report

Goal of this task

This task was prepared to give you a "taste" of how our domain looks like and how you develop a solution for the proposed case. The code will be used as a base for discussion on future interview steps. Two CSV files will be provided as data source for this exercise. You will be asked to provide four different reports based on this data. Under "Requirements", "Milestones" and later under "Acceptance Criteria" you will find further information about what to do. We kindly ask you to invest a maximum of **five hours** into this task. If you have any questions, please contact us.

Technical Info & Expectations

- Language: Feel free to pick any programming language you're comfortable with. We would prefer using Scala or TypeScript since those are our main programming languages. It might also be a good opportunity to learn something by trying one of those languages if you're not familiar with them.
- Web Frameworks: We use [Play Framework](#) for Scala and [Express](#) for Typescript. Both are excellent choices for this task.
- Test: Make sure your code is tested
- Please make sure it's easily executable on reviewer's machines without extensive setup
- Documentation: -- How to run the code -- How to run the tests -- If any code generator was used
- You may send us your code per email packaged in a ZIP file but we would prefer to have just a link of your repository containing all the code, test and documentation on GITHUB.

Requirements

Your product manager is reaching out to you and kindly requests some basic report on listings and leads on our platform. Ideally the report is a web-ui so the product manager can look into it on a regular basis.

The report should include:

- Average Listing Selling Price per Seller Type
- Distribution (in percent) of available cars by Make
- Average price of the 30% most contacted listings
- The Top 5 most contacted listings per Month

Milestones

Those milestones are meant to be guidance and provide you with ideas, not as hard requirements. You are not expected to achieve all milestones, however the report should be generated in some form. If you need to do a trade-off between code quality and feature set please focus on showing your best code.

1. Write a web application that displays / outputs the reports. This can be server generated html with no styling or some form of api depending on your preferences. Keep it simple! NOTE: This already fulfills the minimum requirements!
2. The project manager reaches out to you and mentions that they will want regularly to provide new input files

- Add an upload endpoint to the server that receives CSV files, validates their format and uses the data in the uploaded CSV to fulfill the above requirements for subsequent requests
3. Engineers from another team reach out and would like to re-use your aggregations. Add an api endpoint which exposes the data in a structured format.

Terms & Taxonomy

- Listing: that's the representation of a car being sold. At AutoScout24, "Listings" are the items you find on our list pages, like for example: [<https://www.autoscout24.de/lst>].
- Make: "Producer" of cars, for instance BWM, Audi, VW, etc.
- Seller Type: What kind of seller is behind that listing. If it's "private", that means the car will be sold by a private person. If it's dealer, it's means that the car is sold on a dealer shop. "Other" means another types of sellers that are not relevant in this context.
- Lead: Contact between a car buyer and car seller

Acceptance Criteria

Average Listing Selling Price per Seller Type

- There are three Seller Types: private, dealer and other.
- For each of these types, it should be provided an average selling price.
- The average price should be formatted as € #,-
- Output format is free for you to choose, but an example could be:

Seller Type	Average in Euro
private	€ 2.500,-
dealer	€ 3.529,-
other	€ 1.200,-

Percentual distribution of available cars by Make

- For each make, it should be reported the percentual amount of listings.
- The report should be sorted by distribution, where makes with biggest numbers stays on top.
- Output format is free for you to choose, but an example could be:

Make	Distribution
Audi	55%
BMW	35%
VW	10%

Average price of the 30% most contacted listings

- Using the "Contacts CSV list", report the average price(format: € #,-) of the 30% most contacted listings.
- Output format is free for you to choose, but an example could be:



Average price

€ 12.512,-

The Top 5 most contacted listings per Month

- Using the "Contacts CSV list", report which listing had more contacts in each month.
- Reported fields: Ranking, Listing Id, Make, Selling Price (format: € #,-), Mileage(format: # KM), Total Amount of contacts
- Output format is free for you to choose, but an example could be:

Month: 01.2020

Ranking	Listing Id	Make	Selling Price	Mileage	Total Amount of contacts
1	1000	BWM	€ 2.538,-	50.000 KM	15
2	1001	Audi	€ 4.300,-	20.000 KM	12
3	1002	Toyota	€ 18.250,-	35.000 KM	11
4	1003	VW	€ 25.080,-	45.678 KM	10
5	1004	Porsche	€ 102.000,-	2.000 KM	8

Month: 02.2020

Ranking	Listing Id	Make	Selling Price	Mileage	Total Amount of contacts
1	1004	Porsche	€ 102.000,-	2.000 KM	18
2	1001	Audi	€ 4.300,-	20.000 KM	17
3	1000	BWM	€ 2.538,-	50.000 KM	15
4	1003	VW	€ 25.080,-	45.678 KM	10
5	1002	Toyota	€ 18.250,-	35.000 KM	3

Definition of the CSV files

- Listing.csv

field	type	required
id	numeric	yes
make	alphanumeric	yes
price	numeric	yes
mileage	numeric	yes
seller_type	alphanumeric	yes

- contacts.csv

field	type	required
listing_id	numeric	yes
contact_date	UTC Timestamp(ms)	yes