

Adcuratio - Backend Assignment

Overview

Time to get your hands dirty! The assignment is a simple web API development problem - with a very limited and restricted scope so you do not have to spend more than a few hours on the overall assignment. If you need any help, please feel free to reach out for clarifications. You can use any object oriented language / framework you are most comfortable with. If you are familiar with Python (Django), using that is a definite plus.

This software development assignment helps you show us what you got, and helps us assess you on how you solve problems. More specifically, we look for a number of aspects in your submission. Among other things-

- How you approach the problem - What assumptions and trade-offs you choose to make - What design object-oriented/functional design decisions you make - please craft your best code!
- Code readability, language conventions, logical packaging/directory structure - Testing - Version control (frequent, atomic units are a huge plus) - Your submission must be a working solution

Note: We use a simplified mock API of our product to test for something that our candidates would actually be working on instead of something completely made up. Your code submissions will in no way be used in our product.

Problem Statement

Customers send in their invoices (PDF files) and the your system converts the (unstructured) data from the invoice into a structured format and saves it in an SQL database. At the minimum, this includes the vendor/seller, the purchaser/buyer, the invoice number and date, and each line item mentioned in the invoice. (Here are some examples of invoices on [Google Image search](#) to help you understand more about the type of information that Invoices generally contain. It is expected that you consider other useful information in your data model). These documents are then processed by our background processing engine using OCR and machine learning (or sometimes manually), structured information is generated **and validated**, after which the document is marked “digitized” and the structured information is accessible

to our customers.

We want to mock this process in this assignment. **Create a web service with the following requirements:**

API endpoints for the end customer

- To allow a customer to provide a PDF document (invoice) to process
- To allow a customer to track a document's digitization status
- To allow a customer to retrieve the structured invoice information for a specified

document, if the document status is digitized (simply return mock data in this case)

API endpoints for internal users / other microservices

- To allow a staff member (or another microservice) to manually add digitized / parsed **(structured)** information for a specific document

- For example, if a staff member views the document and wishes to record in the Structured data tables that the Invoice Number is INV1234, they should be able to do so
 - It should be possible to add/update more than one field at a time, if the caller

chooses to do so - It should not be mandatory to update all fields/information in the same call; it

should be possible to independently and partially add/update each piece of information - To allow marking a document as "digitized"

You are free to design the API (and associated data model) however you wish as long as it satisfies the criteria. Even though we only want to build these APIs, please make sure to think through the above-mentioned digitization process and think of other scenarios / considerations that might exist in the design for such a system - we want to know what all you come up with! :)

Out of Scope

To limit the amount of time you need to spend on this assignment, these things are explicitly out of scope:

- User authentication
- Any logic (parsing, OCR, etc) for actually digitizing a document
- Any other supporting endpoints/models not directly related to these endpoints

For any other questions, please reach out for

clarifications.