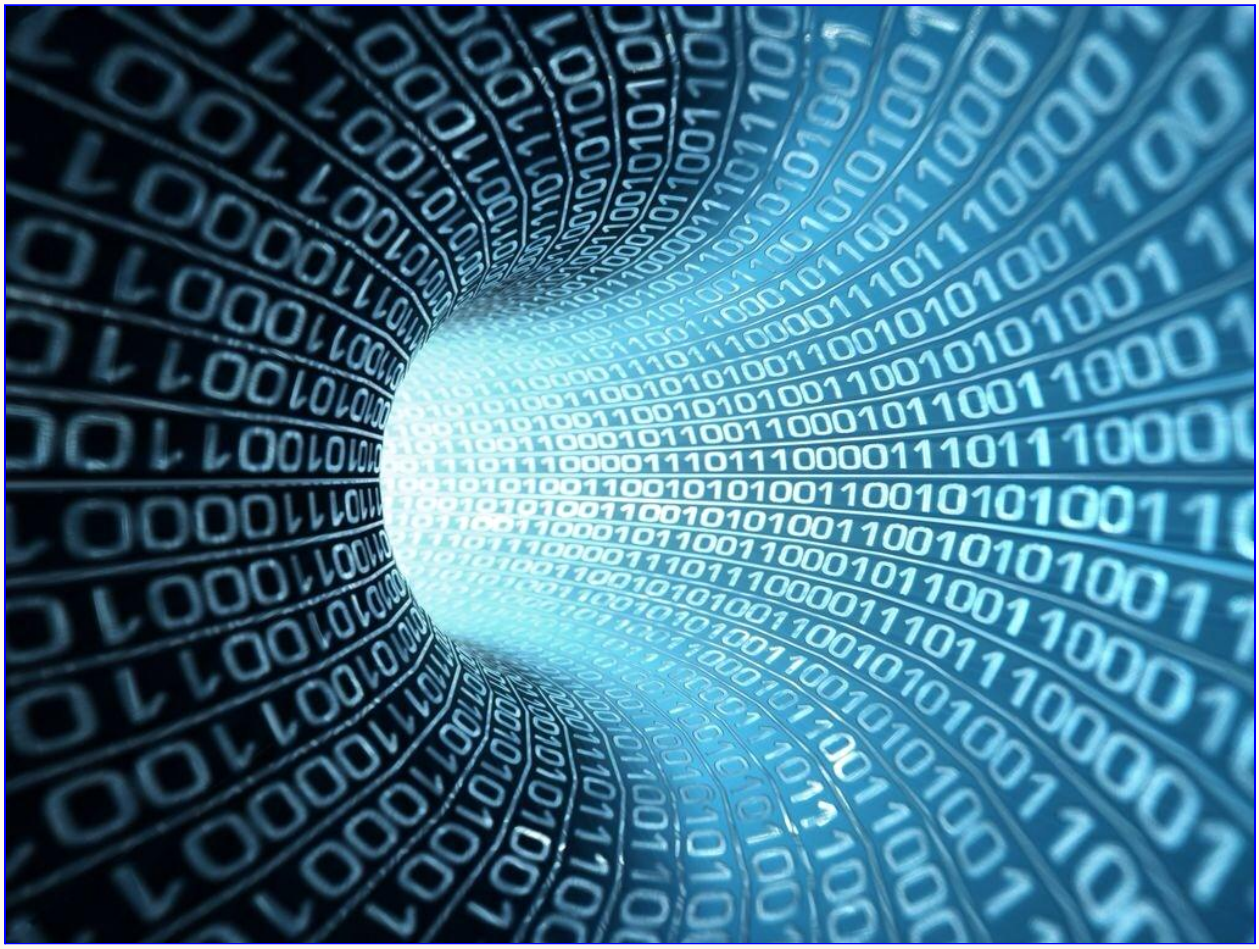
DeepCoding

Senior Engineer Candidate Challenge 1



INTRODUCTION

DeepCoding runs machine learning on human work logs that are retrieved from ITSM systems. With DeepCoding technology we can provide strategic insights as well as real-time recommendations. The recommendations improve overall delivery performance and reduce IT costs.

As part of customers' policy, sensitive data that is being retrieved from their ITSM systems need to be masked before consumption.

While you work on the assignment we encourage you to share your progress with us, let us understand your thinking process and do not hesitate to ask for clarifications.

For any question and help please approach via WhatsApp the contact point that will provide you:

DeepCoding wishes you success and joy :-)

Assignment - Masking Proxy Server

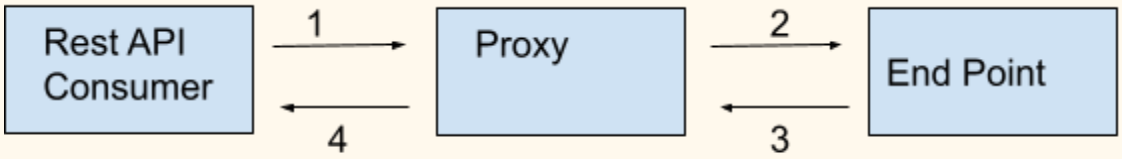
You should create an application server that acts as a simple proxy for masking purposes.

Development environment and tools

- You can pick any development tool, platform, and environment for solving the task.
- Please use Python (or Node.JS) for coding.

Assignment details

You should expose in your server rest API that accepts text forward it to another server and mask the response. For simplicity we will use a public end-point without authentication.



1, HTTP request with text in payload
2. The request is forwarded to the end-point
3, The response is received by the proxy
4. The proxy apply masking and send back a masked response

For simplicity the end-point will be https://postman-echo.com/post

This end-point implements “echo” functionality - reply with a body that contains the input that was sent to him. For example:

Request:

Response:


```
{
  "args": {},
  "data": "The laptop of Mike (242-22-5322) patch can not be applied in the last week",
  "files": {},
  "form": {},
  "headers": {
    "x-forwarded-proto": "https",
    "x-forwarded-port": "443",
    "host": "postman-echo.com",
    "x-amzn-trace-id": "Root=1-60c99f07-6daa57fe0933a4373c8d61fa",
    "content-length": "74",
    "content-type": "text/plain",
    "user-agent": "PostmanRuntime/7.28.0",
    "accept": "*/*",
    "cache-control": "no-cache",
    "postman-token": "d507089d-8179-49b5-824d-0ce994e83e90",
    "accept-encoding": "gzip, deflate, br",
```

```
    "cookie":
"sails.sid=s%3Akv7N-8ZHTfIj2u8riD4yh7M-vYB_-7Ke.Z9tw%2BXFTlBmIz1icdIO2PmOcscbyDh0%2FjLJl3O3sCmg"
  },
  "json": null,
  "url": "https://postman-echo.com/post"
}
```


The Masking proxy should apply two types of masking:
1. Mask all **valid** SSN occurrences in the response (“data” field).

2. Mask all occurrences of names (list of names will be loaded to the server on start-up from a CSV file in the local file storage)

The Masking operation replaces the chars in the original tokens with the char X
In the above example the Data value “"The laptop of **Mike** (**242-22-5322**) patch can not be applied in the last week" will be masked like that: “"The laptop of **XXXX (XXX-XX-XXXX)** patch can not be applied in the last week"

Notes:

- Potential SSN token can be identified with RegEx.
- You can start your implementation by identification of potential SSN tokens and improve it with ensuring the validity of the SSN token.
- For testing you can use online SSN validator
- The list of names is provided in simple CSV file..
  Please take into account that there may be a very big number of names.
  Please think about suitable data structure.

Deliverables

- Please write what were your assumptions you took in the process.
- Please write your code like it is part of a company production product (have a readable log, add comments; clear, organized and maintainability code).
- Provide access to your code so we can test it separately.
  Bonus - host your proxy server in the cloud (e.g. GCP, AWS VM) and expose its URL.
- Consider the DataStructure that you using and server development considerations (concurrency, performance, logging)
- Please be ready to present your implementation and to be asked questions about it.