Backend Challenge

We're thrilled that you are interested to join Gloris

We'd like to present you with a coding task that will allow us to get to know you and your coding skills a little better.

Background

Most of our clients use our API as the engine behind a customer facing bot integrated into their chats.
When one of their customers, also known as visitors, writes a message in the chat, which is sent to our servers. Our AI analyzes that message in order to predict the customer's intent, enabling our bot to return an appropriate reply according to the predicted intent. An example of a conversation:
> Visitor: "Hello"
AI understands that this is a Greeting
> Bot: "Hello :) How can I help you?"
To clarify, in this example:
● "Hello" is the visitor message
● "Greeting" is the predicted intent
● "How can I help you" is the reply
In order to analyze the intent behind a visitor message, we consume an AI API. A message is given to the AI and a list of intents is returned. A simplified version looks something like this:

```
[
  {
    "name": "Greeting",
    "confidence": 0.81
  },
  {
    "name": "Delivery status",
    "confidence": 0.18
  },
  {
    "name": "Refund possibility",
    "confidence": 0.01
  }
]
```

As you can see in the above example, the AI does not give one

definitive intent, but a list of intents with corresponding confidence values. The confidence means how sure was the AI that this is the correct intent behind what the user meant by a certain message. Usually the reply that is related to this intent is only given, if the confidence is above a certain threshold which is configurable for every reply. In other cases we give the visitor a default answer that can, for example, tell the visitor that the AI could not give the correct answer.

Your Challenge

Your task is to create a backend web server that exposes a single endpoint. This endpoint accepts a bot identifier and a visitor written message. It returns a single reply corresponding to the highest predicted intent above the confidence threshold, as per the description above.

Minimum Requirements

● A web server written in either NodeJS (TypeScript / JavaScript)or a JVM language.
● The web server should use a database from which it reads the replies from. We do not have a requirement for a certain technology, though we do prefer MongoDB as this is database of choice.
● Create one collection to have data from json attached and you should search typed string from translateData->messages. If there are any data matched or similar string there, the reply with reply field. If not, reply to default messages.
● Request and Response would be the same Structure that I shared on the swagger file. (not use that one but you can make Request and Response as like swagger)
● Tests written in a framework of your choice.

Further instructions

● If you have any questions regarding this task, please don't hesitate to ask our recruiters.
● Refer to the attached json file for examples of visitor messages and corresponding intents and replies.
● Please do not spend more than a few hours on this challenge; we do not expect a solution that is production ready.
● To share your solution with us, you can either send us a link to a repository where your code resides(GitHub, GitLab, etc..) or by sending us a zipped archive of your code.

What we look for

● Clean code that is easy to read and written with reusability and testability in mind.
● Some comments or documentation to help us understand the choices you made or next steps you would consider if you had more time.

Related source:
https://drive.google.com/drive/folders/1oNO19rsd1ty2tjjH2wwXNpz2qjCC2N8G?usp=sharing