

1. What steps would you take to address the issue in *Ticket #0* where the API returned incorrect text recognition for a number plate?

>>>

The issue is likely related to the language parameter used in the OCR API. I would first replicate the issue by using the same image and API call. If the language parameter is set to "unk" or left empty, it can cause incorrect recognition. By setting the language parameter to "en" (English), the OCR correctly identifies the number plate as "UP27AP7762." I would inform the client to set the parameter correctly and provide a clear response with steps to test the fix.

2. How would you help a client resolve an "Access denied due to invalid subscription key or wrong API endpoint" error, as described in *Ticket #1*?

>>>

This error typically occurs when the API key is invalid or the API endpoint is incorrect. I would ask the client to double-check that they are using the correct subscription key and ensure that they are calling the correct regional endpoint that matches their API subscription. If the client is still having trouble, I would provide a valid test key and the correct endpoint details for troubleshooting.

3. A client is getting a reversed or distorted text response from an image, as described in Ticket #2. How would you troubleshoot this issue?

>>>

The issue could be related to the image orientation or the API misinterpreting the text due to the image format. I would first replicate the issue by processing the same image using the API. If the result is reversed (e.g., "umoa ep!sdn" instead of correct text), I would suggest ensuring the image is properly aligned and formatted. I would also recommend using a different text extraction method or specifying more accurate parameters in the API call.

4. How would you handle a situation in *Ticket #3* where a client received a response stating their location was not in India, but the user claims they were in India at the time?

>>>

I would analyze the API logs for the user (userId: 58f2336f-cd44-4ef4-bfd0-8d61c1fff8ae) and check the coordinates (latitude and longitude) sent in the request. I

would verify if there was an issue with the server or if the coordinates provided were outside India's geographical boundaries. I would also cross-check with the system's geolocation data to determine if any discrepancies occurred.

5. A client mentions that they were physically present with a customer when the API incorrectly returned that the user was not in India (as seen in *Ticket #4*). How would you verify this and address the client's concern?

>>>

I would start by filtering the API logs for the time frame between 9 am to 10 am IST on June 22, 2020. By checking the logs, I could identify the specific userId and review the request's lat/long coordinates to determine if they were within India's boundaries. Additionally, I would check for any potential server issues or temporary connectivity problems that could have affected the geolocation service during that period. Once identified, I would provide the correct userId and explain any findings to the client.

Here's an expanded list of at least 10 questions with their respective answers, based on the problem statement provided in the PDF. I'll also include important keywords and frame questions around them.

1. How can a client generate their own API key, as suggested in Problem Statement 1?

>>> Generating an API key typically involves signing up for the API service (in this case, Microsoft Vision API) and accessing the dashboard or portal where API keys are managed. I would advise the client to visit the Microsoft Azure portal, navigate to the API keys section, and create a new key for the Vision API. This new key can then be used in their requests.

2. A client is facing a "wrong API endpoint" error, as described in *Ticket #1*. How would you help them find the correct endpoint?

>>>

I would guide the client to refer to the Microsoft Vision API documentation. Typically, endpoints are regional (e.g., West Europe, East US). The client should use an endpoint that matches the region of their Azure subscription. I would help them identify the correct endpoint by checking their subscription region and providing the corresponding API URL.

3. In Ticket #2, why might the API return reversed or garbled text like "umoa ep!sdn"?

>>>

The reversed or garbled text could be due to image orientation issues or incorrect processing by the OCR engine. I would advise the client to check whether the image is upside down or mirrored and to use parameters in the API call to ensure proper orientation detection. Testing with different image orientations and formats can help resolve this.

4. How would you explain the use of the `language` parameter in the OCR API based on *Ticket #0*?

>>>

The `language` parameter specifies which language the OCR engine should expect when processing the text. If set to "unk" (unknown) or left empty, the engine may misinterpret the characters. By setting the parameter to a specific language (like "en" for English), the OCR can produce more accurate results. I would advise clients to always specify the `language` parameter if the text's language is known.

5.What is the significance of the `statusCode` field in the API response (from Problem Statement 2)?

>>>

The `statusCode` field indicates the success or failure of an API request. A `200` status code means that the request was successful and processed correctly. Other status codes (e.g., `400` for bad requests or `500` for server errors) help identify issues in the request. I would explain to the client that checking the `statusCode` can quickly inform them whether the API call was successful or if troubleshooting is needed.

6. How can a client verify the coordinates used in an API request, as mentioned in Ticket #3?

>>>

I would recommend that the client checks the latitude (`lat`) and longitude (`long`) values passed in their API request. These coordinates can be verified by cross-referencing them with a mapping service like Google Maps. If the values are outside the

expected range, the API will return that the location is not in India. I would assist the client in reviewing the logs and confirm that the correct coordinates were submitted.

7. How would you guide a client in troubleshooting invalid API keys based on Ticket #1?

>>>

Invalid API keys typically result from expired or revoked subscriptions. I would ask the client to confirm that their subscription is active and that they are using a valid API key. If necessary, I would provide steps for generating a new key from their account dashboard and guide them through updating their application to use the new key.

8. What steps would you follow to investigate a potential server issue based on Ticket #4?

>>>

I would first analyze the logs from the specified time period (9 am to 10 am IST on June 22, 2020). By checking the `requestId`, `userId`, and the timestamp, I can verify whether the API responded correctly or if there were any server-side issues. If no issue is found in the logs, I would check for any network or infrastructure problems that may have occurred during that time.

9. How does the API log system work as described in *Problem Statement 2*?

>>>

The API logs capture essential details of each API request, including `appld` (client identifier), `userId` (customer identifier), `lat` and `long` (coordinates), the `status` of the request, and the `timestamp`. These logs are useful for debugging and auditing API requests. I would explain to the client how to read these logs and use them to troubleshoot specific requests.

10. How would you use the `userId` to investigate a customer complaint, as in *Ticket #3* and *Ticket #4*?

>>>

The `userId` uniquely identifies each customer within a client's application. By searching the logs for the `userId` provided, I can track all the API calls made by that

user and verify whether the API returned the correct result. If an error is found, I would use this information to correct the issue or suggest improvements.

Keywords-based Questions

11. What is the importance of the `requestId` in API logs?

>>>

The `requestId` is a unique identifier for each API call made to the server. It helps in tracking individual requests and is essential for debugging purposes. When a client reports an issue, using the `requestId` allows support engineers to quickly locate the specific API call and investigate its details.

12. How can the `inIndia` response field help clients validate their API requests?

>>>

The `inIndia` field in the response tells the client whether the provided latitude and longitude coordinates fall within India's geographical boundaries. This boolean response (`yes` or `no`) helps the client confirm whether the location is correctly identified by the API.

13. Why is the `timestamp` field in the logs important for troubleshooting?

>>>

The `timestamp` field records the exact time when an API request was made, using a UNIX timestamp. It is crucial for correlating issues with specific time periods, helping support engineers verify whether any external factors (e.g., server downtimes) affected the API performance during that time.

14. How can clients use the `lat` and `long` fields in API requests?

>>>

The `lat` (latitude) and `long` (longitude) fields are used to provide geographical coordinates to the API. These coordinates are essential for determining the location-based response, such as whether a given point lies within India. I would explain to

clients that accurate `lat` and `long` values are required for the API to return the correct result.

15. What steps would you take to analyze logs from a single day, such as June 22, 2020, as mentioned in *Problem Statement 2*?

>>>

To analyze the logs from a single day, I would first filter the log entries by the timestamp for June 22, 2020. Then, I would review the `requestId`, `userId`, `lat`, `long`, and `inIndia` fields to identify any irregularities. Additionally, cross-referencing with client-reported issues during that time would help in pinpointing specific problems.