***Troubleshooting Guide for API Support Team***

* **401 Unauthorized Errors**

1. **Check API Key**: Ensure that the API key provided is valid and has not expired.

**Solutions**:

**Verify API Key:**

* + Ensure that the API key being used is correct and has not expired.
  + Check if the key has the necessary permissions for the requested resource.

1. **API Key Permissions**: Verify the API key has the necessary permissions in order to access the requested data.

**Solutions**:

**Check Credentials:**

* + Confirm that the correct username and password (if required) are being used.
  + Make sure the authentication method matches the API's requirements.

1. **Correct Endpoint**: Then confirm the endpoint URL is correct and matches the API documentation.

**Solutions**:

**Authorization Header:**

* + Verify the URL is correct and that the authorization header is included in the request and formatted correctly.]

1. **Rate Limits**: Check if the API key has covered, it’s rate limit and wait for the limit to reset.

**Solutions:**

* + Check if the API rate limit has been exceeded and wait for the limit to reset.
  + Consider implementing rate limiting in your application to avoid exceeding limits in the future.
* **Slow API Response Times**

1. **Network Issues**: First of all need to check the network connection to be sure doesn’t have ani issues can be responsible to cause delays

**Solutions**:

* + Test network connectivity and ensure there are no issues affecting data transmission.
  + Consider using a more stable network connection.

1. **Server Load**: Verify if the API server is experiencing high traffic, then must consider making requests during off-peak hours, that way can check two scenarios

**Solutions**:

* + Check the API server status for any maintenance or high traffic alerts.
  + Schedule requests during off-peak hours to avoid server congestion.

1. **Optimize Requests**: Need reduce the frequency of API ,calls and cache responses when possible.

**Solutions**:

* 1. Minimize the size of response payloads by requesting only the necessary data.
  2. Use pagination or filtering to reduce the amount of data returned in a single request.

1. **Check Response Size**: It is essential make sure. that the response size remains manageable to avoid slowing down the retrieval process.

**Solutions**:

* 1. Review and optimize the request methods being used (e.g., use GET for fetching data).
  2. Avoid making redundant or repetitive requests.
* **Incorrect JSON Responses (Missing Fields)**

1. **API endpoint issues or server-side bugs**: Review the API documentation to ensure that the correct fields are being requested.

**Solutions**:

**Review API Documentation:**

* + Ensure that the request parameters match the API documentation and are correctly formatted.
  + Verify that the requested fields are supported by the API endpoint.

1. **Field Validation**: Must validate the requested fields are available for the specific end-point and parameters.

**Solutions**:

**Validate Request Parameters:**

* + Check if the parameters being sent in the request are valid and meet the required criteria.

1. **Error Handling**: Implement error supervision to catch and manage missing fields on the right manner

**Solutions**:

**Error Handling:**

* + Implement error handling in your application to catch and manage missing or null fields gracefully.
  + Use fallback mechanisms to handle missing data and provide meaningful feedback to users.

1. **Update Code**: Keep the code updated to handle cases, where certain fields might be missing or null.

**Solutions**:

* + Verify Changes in the API Design
  + Verify Changes in the API Structure
  + Verify Changes in the API response codes