

## Assignment C

In this assignment, there's a discrepancy between the keys of objects used in WebSocket and RESTful GET. The RESTful API employs Camel Case, while the WebSocket uses Pascal Case. Aligning the data structures across different API implementations is crucial to prevent errors and issues that may arise due to inconsistencies.

Enhancing API performance involves optimizing payload size, employing technologies like GZIP to reduce JSON size, and using concise yet descriptive variable/key names. Despite brevity, it's essential to maintain clarity for users to comprehend the data.

Comprehensive internal documentation is indispensable, especially in larger projects. For instance, in this assignment, the absence of clarification about units (e.g., altitude in feet or meters) for certain numeric values caused ambiguity.

Improving the API structure could involve adopting a more sophisticated communication protocol like the OPC UA standard, capable of managing sessions, message queues, sensor data, and handling communication failures more effectively.

Effective error handling is essential, alongside establishing predefined enumerations for elements like `messageStatus`. For instance, `{ messageStatus: "warning", message: "Temperature is low" }`, aiding in understanding data integrity or if certain parameters, like temperature, exceed acceptable limits.

To ensure secure interactions, encryption of information—especially when transmitting data to the rocket—is crucial. Implementing authentication systems like login procedures and utilizing technologies such as JWT and API Keys adds layers of security, preventing unauthorized access or malicious actions by third parties aiming to compromise the rocket or exploit the API.