Resiliency JSON Structure

Resiliency combination request:

Here's the structural notation of request for better understanding:

```
{ "operators":
        [ { "BW": <integer>,
                “delay": <float>,
                "packet_loss": <float>,
                "name": <string> }, ...
        ],
        "required_bandwidth": <integer>,
        "resiliency_level": <integer>
}
```

Explanation of Request Fields:

1. **`operators`**:

    - A list of dictionaries, where each dictionary represents an operator.
    - Each operator has:
        - `"BW"`: Bandwidth as an integer.
        - `"delay"`: Delay as a float.
        - `"packet_loss"`: Packet loss as a float.
        - `"name"`: Operator name as a string.

2. **`required_bandwidth`**:

    - An integer specifying the minimum required bandwidth.

3. **`resiliency_level`**:

    - An integer indicating the desired resiliency level (e.g., 1 for low, 2 for medium, 3 for high).

Example of 3 operators with Given Data:

```
{
 "operators": [
   {
    "BW": 200,
    "delay": 20.0,
    "packet_loss": 2.0,
    "name": "Operator200"
   },
   {
    "BW": 105,
    "delay": 10.5,
    "packet_loss": 1.05,
    "name": "Operator105"
   },
   {
    "BW": 80,
    "delay": 8.0,
    "packet_loss": 0.8,
    "name": "Operator80"
   }
 ],
 "required_bandwidth": 100,
 "resiliency_level": 2
}
```

Resiliency combination response:

The JSON response structure provided is a **list of lists**, where each inner list contains dictionaries representing operator details. Here's the structural notation for better understanding:

```
{
 [
  [
    {
      "BW": <integer>,
      "delay": <float>,
      "packet_loss": <float>,
      "name": <string>
    },
    …
  ],
  ...
 ]
"ErrorCode": <integer>,
"ErrorMessage": <string>
}
```

Explanation of Response Fields:

1. **Outer List**:
    - The outer list contains multiple inner lists.

2. **Inner Lists**:
    - Each inner list represents a combination of operators.
3. **Dictionaries**:
    - Each dictionary in the inner list represents an operator with the following keys:
        - `"BW"`: Bandwidth as an integer.
        - `"delay"`: Delay as a float.
        - `"packet_loss"`: Packet loss as a float.
        - `"name"`: Operator's name as a string.

4. **ErrorCode**:
    - An integer error code specifying error return similar to HTTP/REST error codes.
    - 2xx error codes is for success cases
    - 3xx error codes for client suppled wrong input conditions
    - 5xx error codes are server side processing errors
5. **ErrorMessage**:
    - Descriptive of error message in plain English supplied as string

Response example for three operator example request:

```
[
  {
    "BW": 200,
    "delay": 20.0,
    "packet_loss": 2.0,
    "name": "Operator200"
  },
  {
    "BW": 105,
    "delay": 10.5,
    "packet_loss": 1.05,
    "name": "Operator105"
  },
  {
    "BW": 80,
    "delay": 8.0,
    "packet_loss": 0.8,
    "name": "Operator80"
    }
],
[
    {
        "BW": 200,
        "delay": 20.0,
        "packet_loss": 2.0,
        "name": "Operator200"
    },
    {
        "BW": 105,
        "delay": 10.5,
        "packet_loss": 1.05,
        "name": "Operator105"
    }
]
```