**Coding and Design**

Used Web API and created single endpoint to accept calculation. The approach lessen the impact on additional end point to support any new operation. A structure format for Response is used to have a standard response of successful and unsuccessful calls. Payload validation is added to ensure invalid requests doesn’t break the API.

**Request Payload**

1. **Operation : string** – “add”, “subtract”, “multiply” and “divide”
2. **Value1 : double** – numeric value
3. **Value 2 : double** – numeric value

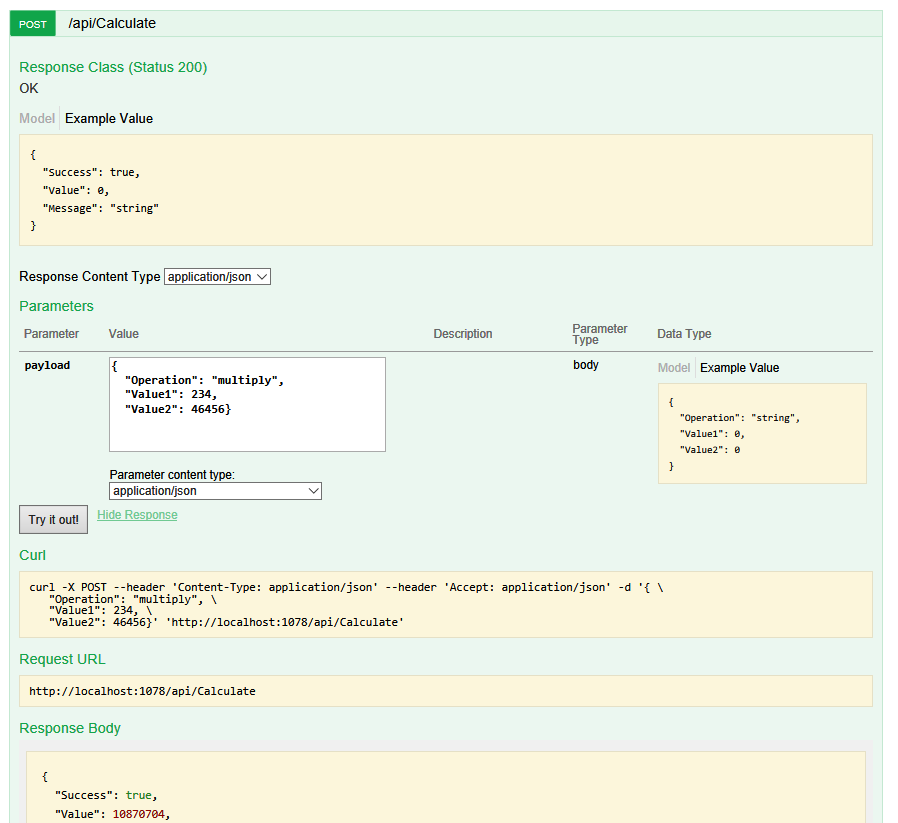
**Response**

1. **Success : bool** – true or false
2. **Message : string** – error message
3. **Value : double** – computation result

**Technologies**

1. Logging – used log4net to log. All requests and response were captured and dump into myapp.txt to capture
2. Unit Testing – implemented Controller testing
3. Swagger – used swagger for api documentation

URL: <http://localhost:1078/swagger/ui/index#!/Calculate/Calculate_Post>



**Example Payload 1**

{

"Operation": "multiply",

"Value1": 234,

"Value2": 123

}

**Example Payload 2**

{

"Operation": "add",

"Value1": 234,

"Value2": 123

}

**Example Payload 3**

{

"Operation": "subtract",

"Value1": 234,

"Value2": 123

}

**Example Payload 4**

{

"Operation": "divide",

"Value1": 234,

"Value2": 123

}