

# API Specification of Hello World

# Table of Contents

1. Overview .....	1
1.1. Version information .....	1
1.2. Contact information .....	1
1.3. License information .....	1
1.4. URI scheme .....	1
1.5. Tags .....	1
2. Definitions .....	1
2.1. ArithmeticController.BadRequestResponse .....	1
2.2. ArithmeticRequest .....	1
2.3. ArithmeticResponse .....	2
2.4. GreetingController.BadRequestResponse .....	2
2.5. GreetingRequest .....	2
2.6. GreetingResponse .....	2
2.7. TopResponse .....	2
3. Resources .....	3
3.1. Arithmetic Calculation .....	3
3.1.1. Add .....	3
3.1.2. Divide .....	4
3.1.3. Multiple .....	6
3.1.4. Subtract .....	7
3.2. Greeting .....	8
3.2.1. Top .....	8
3.2.2. Greet .....	9

# 1. Overview

API document of Hello World service

## 1.1. Version information

*Version* : 1.0.0

## 1.2. Contact information

*Contact* : dayflower

*Contact Email* : @dayflower

## 1.3. License information

*License* : PUBLIC DOMAIN

*Terms of service* : <http://example.com/>

## 1.4. URI scheme

*Host* : localhost

*BasePath* : /

## 1.5. Tags

- Arithmetic Calculation : Provides several basic arithmetic function.
- Greeting : Provides greeting methods.

# 2. Definitions

## 2.1. ArithmeticController.BadRequestResponse

Response model for bad request.

Name	Description	Schema
<b>message</b> <i>optional</i>	Error message. <b>Example</b> : "0 is not permitted."	string

## 2.2. ArithmeticRequest

Common request model for arithmetic calculation.

Name	Description	Schema
<b>value1</b> <i>optional</i>	Former operand. <b>Example</b> : 1	integer (int32)
<b>value2</b> <i>optional</i>	Latter operand. <b>Example</b> : 2	integer (int32)

## 2.3. ArithmeticResponse

Common response model for arithmetic calculation.

Name	Description	Schema
<b>result</b> <i>optional</i>	Result value. <b>Example</b> : 3	integer (int32)

## 2.4. GreetingController.BadRequestResponse

Response model for bad request.

Name	Description	Schema
<b>message</b> <i>optional</i>	Error message. <b>Example</b> : "0 is not permitted."	string

## 2.5. GreetingRequest

Request model for greeting.

Name	Description	Schema
<b>name</b> <i>optional</i>	Name of the person. <b>Example</b> : "dayflower"	string
<b>title</b> <i>optional</i>	Title of the person. <b>Example</b> : "Mr."	string

## 2.6. GreetingResponse

Response model for greeting.

Name	Description	Schema
<b>message</b> <i>optional</i>	Message from the system. <b>Example</b> : "Hello, Mr. dayflower"	string

## 2.7. TopResponse

Response model for top.

Name	Description	Schema
<b>alive</b> <i>optional</i>	Whether service is available or not. <b>Example</b> : <i>false</i>	boolean

## 3. Resources

### 3.1. Arithmetic Calculation

Provides several basic arithmetic function.

#### 3.1.1. Add

POST /arith/add

##### Description

Add two values.

##### Parameters

Type	Name	Description	Schema
<b>Body</b>	<b>request</b> <i>required</i>	request	<a href="#">ArithmeticRequest</a>

##### Responses

HTTP Code	Description	Schema
<b>200</b>	OK	<a href="#">ArithmeticResponse</a>

##### Consumes

- *application/json*

##### Produces

- *application/json; charset=UTF-8*

##### Example HTTP request

```
POST /arith/add HTTP/1.1
Content-Type: application/json
Content-Length: 34
```

```
{
  "value1" : 1,
  "value2" : 2
}
```

### Example HTTP response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 18
```

```
{
  "result" : 3
}
```

### Example Curl request

```
$ curl 'http://example.com/arith/add' -i -X POST -H 'Content-Type: application/json'
-d '{
  "value1" : 1,
  "value2" : 2
}'
```

## 3.1.2. Divide

```
POST /arith/div
```

### Description

Divide one value by the other value.

### Parameters

Type	Name	Description	Schema
Body	<b>request</b> <i>required</i>	request	<a href="#">ArithmeticRequest</a>

### Responses

HTTP Code	Description	Schema
200	OK	<a href="#">ArithmeticResponse</a>
400	Bad Request	<a href="#">ArithmeticController.BadRequestResponse</a>

### Consumes

- `application/json`

### Produces

- `application/json; charset=UTF-8`

### Example HTTP request

```
POST /arith/div HTTP/1.1
Content-Type: application/json; charset=UTF-8
Content-Length: 35

{
  "value1" : 35,
  "value2" : 5
}
```

### Example HTTP response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 18

{
  "result" : 7
}
```

### Example Curl request

```
$ curl 'http://example.com/arith/div' -i -X POST -H 'Content-Type:
application/json; charset=UTF-8' -d '{
  "value1" : 35,
  "value2" : 5
}'
```

### 3.1.3. Multiple

POST /arith/mul

#### Description

Multiply two values.

#### Parameters

Type	Name	Description	Schema
Body	<b>request</b> <i>required</i>	request	<a href="#">ArithmeticRequest</a>

#### Responses

HTTP Code	Description	Schema
200	OK	<a href="#">ArithmeticResponse</a>

#### Consumes

- `application/json`

#### Produces

- `application/json; charset=UTF-8`

#### Example HTTP request

```
POST /arith/mul HTTP/1.1
Content-Type: application/json; charset=UTF-8
Content-Length: 34

{
  "value1" : 3,
  "value2" : 5
}
```

#### Example HTTP response



```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 19
```

```
{
  "result" : 15
}
```

### Example Curl request

```
$ curl 'http://example.com/arith/mul' -i -X POST -H 'Content-Type:
application/json;charset=UTF-8' -d '{
  "value1" : 3,
  "value2" : 5
}'
```

### 3.1.4. Subtract

POST /arith/sub

#### Description

Subtract one value from the other value.

#### Parameters

Type	Name	Description	Schema
Body	<b>request</b> <i>required</i>	request	<a href="#">ArithmeticRequest</a>

#### Responses

HTTP Code	Description	Schema
200	OK	<a href="#">ArithmeticResponse</a>

#### Consumes

- `application/json`

#### Produces

- `application/json;charset=UTF-8`

### Example HTTP request

```
POST /arith/sub HTTP/1.1
Content-Type: application/json;charset=UTF-8
Content-Length: 34

{
  "value1" : 1,
  "value2" : 2
}
```

### Example HTTP response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 19

{
  "result" : -1
}
```

### Example Curl request

```
$ curl 'http://example.com/arith/sub' -i -X POST -H 'Content-Type:
application/json;charset=UTF-8' -d '{
  "value1" : 1,
  "value2" : 2
}'
```

## 3.2. Greeting

Provides greeting methods.

### 3.2.1. Top

```
GET /
```

#### Description

Get availability of the service.

#### Parameters

Type	Name	Description	Schema	Default
Query	<b>stage</b> <i>required</i>	Stage.	enum (DEVELOPING, STAGING, PRODUCTION)	"PRODUCT ION"

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">TopResponse</a>

## Consumes

- `application/json`

## Produces

- `application/json; charset=UTF-8`

## Example HTTP request

```
GET /?stage=DEVELOPING HTTP/1.1
```

## Example HTTP response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 20

{
  "alive" : true
}
```

## Example Curl request

```
$ curl 'http://example.com/?stage=DEVELOPING' -i
```

## 3.2.2. Greet

```
POST /greet
```

## Description

Get greeting message for the request.

## Parameters

Type	Name	Description	Schema
Body	<b>request</b> <i>required</i>	request	<a href="#">GreetingRequest</a>

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">GreetingResponse</a>
400	Bad Request	<a href="#">GreetingController.BadRequestResponse</a>

## Consumes

- `application/json`

## Produces

- `application/json; charset=UTF-8`

## Example HTTP request

```
POST /greet HTTP/1.1
Content-Type: application/json
Content-Length: 45

{
  "title" : "Mr.",
  "name" : "dayflower"
}
```

## Example HTTP response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 41

{
  "message" : "Hello, Mr. dayflower!"
}
```

## Example Curl request

```
$ curl 'http://example.com/greet' -i -X POST -H 'Content-Type: application/json' -d '{
  "title" : "Mr.",
  "name" : "dayflower"
}'
```