

## Test entrance

---

Hey hello! This is a simple test application that should take no more than a few hours of your time. The package should be written using pure PHP or Node.js. This small application will have 3 endpoints:

- /api/sayHello
- /api/getBase64
- /api/sayHelloInLanguage

**First endpoint is /api/sayHello .**

This endpoint should take one parameter: name as POST parameter and based on input return success or error response.

```
1 Example of success response:
2
3 curl -X POST -H "Content-Type: application/x-www-form-urlencoded" -d 'name=Test'
  "http://127.0.0.1/api/sayHello"
4
5 Response:
6
7 {
8   "status": "Success",
9   "msg": "Hello Test"
10 }
11
12 Also, this endpoint can return one of the following errors in different circumstances.
13 Error cases for this package that should return 500 error is:
14
15 1. The name have less than 2 characters and more than 15. The error message should be
   "You name is too short" or "Your name is too long". Examples:
16
17 curl -X POST -H "Content-Type: application/x-www-form-urlencoded" -d 'name=a'
   "http://127.0.0.1/api/sayHello"
18
19 Response:
20
21 {
22   "status": "Error",
23   "msg": "Name is too short"
24 }
25
26 curl -X POST -H "Content-Type: application/x-www-form-urlencoded" -d
   'name=ThisIsNotCorrectName' "http://127.0.0.1/api/sayHello"
27
28 Response:
29
30 {
31   "status": "Error",
32   "msg": "Name is too long"
33 }
34
35 2. Should contain at least 2 different letters. For example: aaaa and bbb aren't valid
   names. The error message should be "Please enter valid name".
36
```

```

37 curl -X POST -H "Content-Type: application/x-www-form-urlencoded" -d 'name=aaaa'
   "http://127.0.0.1/api/sayHello"
38
39 Response:
40
41 {
42   "status": "Error",
43   "msg": "Name is invalid"
44 }
45
46 3. Names like "aa" or "bbbbbbbbbbbbbbbbbbbb" should return both messages. E.g:
47 - "Your name is too long" or "You name is too short"
48 - "Please enter valid name"
49
50 curl -X POST -H "Content-Type: application/x-www-form-urlencoded" -d 'name=aa'
   "http://127.0.0.1/api/sayHello"
51
52 Response:
53
54 {
55   "status": "Error",
56   "msg": "Name is too short, Please enter valid name"
57 }
58
59 or
60
61 curl -X POST -H "Content-Type: application/x-www-form-urlencoded" -d
   'name=bbbbbbbbbbbbbbbbbbbb' "http://127.0.0.1/api/sayHello"
62
63 Response:
64
65 {
66   "status": "Error",
67   "msg": "Name is too long, Please enter valid name"
68 }

```

### Second endpoint is /api/getBase64.

This endpoint should receive file and return base64 representation of this file. Example:

```

1 curl -X POST -H "Content-Type: application/json" --data @fileName
   "http://127.0.0/api/getBase64"
2
3 Response:
4
5 {
6   "status": "Success",
7   "base64": "BASE64_HERE"
8 }

```

### Third endpoint is /api/sayHelloInLanguage.

This endpoint take 2 parameters: language and name as POST parameters. It will do the same thing as /api/sayHello, but "Hello" in the response will be based on the language selected. You should use Google Translate API here. See example:

```

1 curl -X POST -H "Content-Type: application/x-www-form-urlencoded" -d
   'name=Test&language=Heb' "http://127.0.0.1/api/sayHello"
2
3 Success response:
4
5 {

```

```
6  "status": "Success",
7  "msg": "שלום Test"
8  }
9
10 Error cases the same as for /api/sayHello endpoint.
```

**Good luck!**