



B2 - Introduction to Web Development

B-WEB-200

API

EPyTodo





API

Some reading about [API](#).

Here, we're using a [REST API](#) which is slightly modified to only use GET and POST methods.

To use PUT and DELETE methods, you need to understand Ajax requests. That's why this is considered as a bonus.

ROUTES

You need to implement the following routes:

route	method	prerequisites	description
/	GET	-	main page
/register	POST	user not logged in	register a new user
/signin	POST	user not logged in	connect a user
/signout	POST	user already logged in	disconnect user
/user	GET	user already logged in	view all user information
/user/task	GET	user already logged in	view all user tasks
/user/task/id	GET	user already logged in existing id task	view a specific task
/user/task/id	POST	user already logged in existing id task	update a specific task
/user/task/add	POST	user already logged in	create a new task
/user/task/del/id	POST	user already logged in existing id task	delete a specific task

All the parameters and responses are explained below:

route	parameters	response if OK	response if KO
/	-	-	-
/register	REGISTER_POST	REGISTER_RES	REGISTER_ERR
/signin	SIGNIN_POST	SIGNIN_RES	SIGNIN_ERR
/signout	-	SIGNOUT_RES	-
/user	-	USER_RES	USER_ERR
/user/task	-	USER_TASK_RES	USER_TASK_ERR
/user/task/id	-	TASK_ID_RES	TASK_ID_ERR
/user/task/id	TASK_ID_MOD_POST	TASK_ID_MOD_RES	TASK_ID_MOD_ERR
/user/task/add	TASK_ID_ADD_POST	TASK_ID_ADD_RES	TASK_ID_ADD_ERR
/user/task/del/id	TASK_ID_DEL_POST	TASK_ID_DEL_RES	TASK_ID_DEL_ERR



DEFINITIONS OF FORMAT

Every route can have multiple methods, parameters or responses.
In this documentation, the definitions of format are capitalized.

+ DEFAULTS

By default, each error will be treated as follows.
If there are more details, that will be explained into the related part.

1. If the user is not logged in :

```
{  
  "error" : "you must be logged in"  
}
```

2. If the task id does not exist :

```
{  
  "error" : "task id does not exist"  
}
```

3. If there's another error :

```
{  
  "error" : "internal error"  
}
```



+ REGISTER

REGISTER_POST

```
{
  "key1" : "value1",
  "key2" : "value2",
  "keyN" : "valueN"
}
```

Where **keyX** is the information sent into the HTML form.
Reminder : at least a **username** and a **password** are required.

REGISTER_RES

```
{
  "result" : "account created"
}
```

REGISTER_ERR

- If the account already exists

```
{
  "error" : "account already exists"
}
```

- If there's another error

```
{
  "error" : "internal error"
}
```



+ SIGN-IN / SIGN-OUT

SIGNIN_POST

```
{
  "username" : "_username",
  "password" : "_password"
}
```

SIGNIN_RES

```
{
  "result" : "signin successful"
}
```

SIGNIN_ERR

- If the account credentials are incorrect :

```
{
  "error" : "login or password does not match"
}
```

- If there's another error :

```
{
  "error" : "internal error"
}
```

SIGNOUT_RES

```
{
  "result" : "signout successful"
}
```



+ USER INFORMATIONS

USER_RES

```
{
  "result" :
  {
    "key1" : "value1",
    "key2" : "value2",
    "keyN" : "valueN"
  }
}
```

Where **keyX** are the information sent when registered or fulfill on his profile page.

USER_ERR

Refer to defaults 1 and 3.

+ VIEW ONE TASK

TASK_ID_RES

```
{
  "result" :
  {
    "title" : "_title",
    "begin" : "_begin",
    "end" : "_end",
    "status" : "_status"
  }
}
```

TASK_ID_ERR

Refer to all defaults.



[+ VIEW ALL TASKS](#)

USER_TASK_RES

```
{
  "result" :
  {
    "tasks" :
    [
      {
        "_id1" :
        {
          "title" : "_title",
          "begin" : "_begin",
          "end" : "_end",
          "status" : "_status"
        }
      },
      {
        "_idN" :
        {
          "title" : "_title",
          "begin" : "_begin",
          "end" : "_end",
          "status" : "_status"
        }
      }
    ]
  }
}
```

Where `_idX` is the id of each task.



`_begin` and `_end` values will have the following format :
YYYY-MM-DD HH:mm:ss

USER_TASK_ERR

Refer to defaults 1 and 3.



+ UPDATE ONE TASK

TASK_ID_MOD_POST

```
{
  "title" : "_title",
  "begin" : "_begin",
  "end" : "_end",
  "status" : "_status"
}
```

TASK_ID_MOD_RES

```
{
  "result" : "update done"
}
```

TASK_ID_MOD_ERR

Refer to all defaults.

+ CREATE A NEW TASK

TASK_ID_ADD_POST

```
{
  "title" : "_title",
  "begin" : "_begin",
  "end" : "_end",
  "status" : "_status"
}
```

TASK_ID_ADD_RES

```
{
  "result" : "new task added"
}
```




TASK_ID_ADD_ERR

Refer to defaults 1 and 3.

+ DELETE A TASK

TASK_ID_DEL_POST

This is actually the id of your URL.

TASK_ID_DEL_RES

```
{  
  "result" : "task deleted"  
}
```

TASK_ID_DEL_ERR

Refer to all defaults.