

API functions documentation

21th, Oct 2022

Contents

| | | |
|----------|---------------------------|----------|
| 1 | User API functions | 3 |
| 1.1 | User login | 3 |
| 1.2 | User register | 4 |
| 1.3 | User logout | 5 |
| 1.4 | User unregister | 6 |

1 User API functions

These functions concern the manipulation of the user database which is responsible for holding the users, user messages, friendships as well as other relations. The implemented abstractions behave in a stateless and are compliant with the REST design pattern. The following is the list with all the API functions:

1.1 User login

This request, called **user_login**, can be used to perform the login of the user by passing the JSON encoded credentials in the body of the request.

- *url endpoint: .../userservices/login.php*
- *method: POST*
- *possible HTTP status codes:*
 - *200 “OK”: request was successful.*
 - *400 “Bad Request”: request had missing or malformed parameters.*
 - *401 “Unauthorized”: user doesn’t exist.*
 - *403 “Forbidden”: wrong password for the specified username or user is already logged-in.*
 - *404 “Not found”: could not find the specified service.*
 - *500 “internal server error”: error in negotiation with the database.*
- *request parameters: JSON object containing two keys.*

```
{
  "username": String,
  "password": String
}
```

- *response parameters: JSON object containing different keys depending on the returned HTTP status code.*

```
{
  "user_already_loggedin": (Boolean | null),
  "payload": {
    "user_id": (Int | null)
  }
}
```

1.2 User register

This request, called **user_register**, can be used to perform the registration of the user by passing the JSON encoded credentials in the body of the request.

- *url endpoint: .../userservices/register.php*
- *method: POST*
- *possible HTTP status codes:*
 - *200 “OK”: request was successful.*
 - *400 “Bad Request”: request had missing or malformed parameters.*
 - *403 “Forbidden”: user already exists.*
 - *404 “Not found”: could not find the specified service.*
 - *500 “internal server error”: error in negotiation with the database.*
- *request parameters: JSON object containing two keys.*

```
{  
  "username": String,  
  "password": String  
}
```

- *response parameters: there is no response body.*

1.3 User logout

This request, called **user_logout**, can be used to perform the logout of the user by passing the JSON encoded username in the body of the request.

- *url endpoint: .../userservices/logout.php*
- *method: DELETE*
- *possible HTTP status codes:*
 - *200 “OK”: request was successful.*
 - *400 “Bad Request”: request had missing or malformed parameters.*
 - *401 “Unauthorized”: user doesn’t exist.*
 - *403 “Forbidden”: user is already offline.*
 - *404 “Not found”: could not find the specified service.*
 - *500 “internal server error”: error in negotiation with the database.*
- *request parameters: JSON object containing the username.*

```
{  
  "username": String  
}
```

- *response parameters: there is no response body.*

1.4 User unregister

This request, called **user_unregister**, can be used to perform the unregistration of the user by passing the JSON encoded credentials in the body of the request.

- *url endpoint: .../userservices/logout.php*
- *method: DELETE*
- *possible HTTP status codes:*
 - *200 “OK”: request was successful.*
 - *400 “Bad Request”: request had missing or malformed parameters.*
 - *401 “Unauthorized”: user doesn’t exist.*
 - *403 “Forbidden”: wrong password for the specified username.*
 - *404 “Not found”: could not find the specified service.*
 - *500 “internal server error”: error in negotiation with the database.*
- *request parameters: JSON object containing the credentials.*

```
{  
    "username": String,  
    "password": String  
}
```

- *response parameters: there is no response body.*