**The School API**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description** |
| 1.0 | 22/09/2017 | Ethan Reubens | First version |

## 

Index

1. [Login](#_2s8eyo1)  
   [Request](#_17dp8vu)

[Response](#_3rdcrjn)

1. All requests
   1. Page components
   2. Requests on user action
2. Classes
3. Functions

## 

## 

## Login Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | login/login.php |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| HEAD  POST  POST | content-type  email  pass | application/x-www-form-urlencoded  String - from a variable ‘email’  string - from a variable ‘pass’ |

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | responseText |

## Owner:

Email: er-post@hotmail.com

Pass: 123456

## manager:

Email: user@gmail.com

Pass: 123456

## Sales:

Email: sales@sales.og

Pass: 1234

## All requests

## Page components

|  |  |  |
| --- | --- | --- |
| Login Request | | |
| **Method** | **URL** |  |
| **POST** | login/login.php |  |
| **Type** | **Params** | **Values** |
| HEAD  POST  POST | content-type  email  pass | application/x-www-form-urlencoded  String - from a variable ‘email’  string - from a variable ‘pass’ |
|  | Response |  |
| **Status** | **Response** |  |
| 200 | responseText |  |

## 

|  |  |  |
| --- | --- | --- |
| User data Request | | |
| **Method** | **URL** |  |
| **GET** | user.php |  |
|  | Response |  |
| **Status** | **Response** |  |
| 200 | userData | **= user variable** |

## 

|  |  |  |
| --- | --- | --- |
| Admin Request | | |
| **Method** | **URL** |  |
| **GET** | main.php |  |
|  | Response |  |
| **Status** | **Response** | **type** |
| 200 | Admin | string |

## 

|  |  |  |
| --- | --- | --- |
| Courses list Request | | |
| **Method** | **URL** | **location** |
| **POST** | actions/getinfo.php | getCourses() |
| **Type** | **Params** | **Values** |
| HEAD  POST | content-type  list | application/x-www-form-urlencoded  String - ‘courses’ (table name) |
|  | Response |  |
| **Status** | **Response** |  |
| 200 | data | **= courses variable** |

## 

## 

|  |  |  |
| --- | --- | --- |
| Students list Request | | |
| **Method** | **URL** | **location** |
| **POST** | actions/getinfo.php | getStudents() |
| **Type** | **Params** | **Values** |
| HEAD  POST | content-type  list | application/x-www-form-urlencoded  String - ‘students’ (table name) |
|  | Response |  |
| **Status** | **Response** |  |
| 200 | data | **= students variable** |

## 

|  |  |  |
| --- | --- | --- |
| Users list Request | | |
| **Method** | **URL** | **location** |
| **POST** | actions/getinfo.php | getUsers() |
| **Type** | **Params** | **Values** |
| HEAD  POST | content-type  list | application/x-www-form-urlencoded  String - ‘users’ (table name) |
|  | Response |  |
| **Status** | **Response** |  |
| 200 | data | **= users variable** |

## Requests on user action

|  |  |  |
| --- | --- | --- |
| Add new course/student/user | | |
| **Method** | **URL** | **location** |
| **POST** | actions/insert.php | insertNew() |
| **Type** | **Params** | **Values** |
| For course:  POST | dataToInsert = { list: name: description: image: }; | From form input |
| For student:  POST | dataToInsert = { list: string, name: name, phone: phone, email: email, image: image }; | From form input |
| For user:  POST | dataToInsert = { list: string, name: name, phone: phone, email: email, image: image, role: role }; | From form input |
|  | Response |  |
| **Status** | **Response** | **action** |
| 200 | result | **endOfEdit()** |

## 

|  |  |  |
| --- | --- | --- |
| Update course/student/user | | |
| **Method** | **URL** | **location** |
| **POST** | actions/update.php | updateItem() |
| **Type** | **Params** | **Values** |
| For course:  POST | dataToInsert = { ID: list: name: description: image: }; | From form input |
| For student:  POST | dataToInsert = { ID: list: name: phone: email: image: courses: }; | From form input |
| For user:  POST | dataToInsert = { ID: list: name: phone: email: image: role: }; | From form input  Restricted to max 1 role = owner |
|  | Response |  |
| **Status** | **Response** | **action** |
| 200 | result | **endOfEdit()** |

## 

|  |  |  |
| --- | --- | --- |
| Delete a course/student/user | | |
| **Method** | **URL** | **location** |
| **POST** | actions/delete.php | deleteItem() |
| **Type** | **Params** | **Values** |
| POST | table:  name: | From form clicked item |
|  | Response |  |
| **Status** | **Response** | **action** |
| 200 | data | **Back to main view** |

## Classes All classes can be found in the ‘includes’ folder

|  |  |  |
| --- | --- | --- |
| SchoolDB | | |
| **variables** | **type** | **description** |
| **$servername $username $password $database** | Private Private Private private | Server name Server user name Server password Database name |
| **methods** | **type** | **description** |
| connect () | protected | If variables are undrfined creates mysqli connection to: **$this->servername = 'localhost';**  **$this->username = 'root';**  **$this->password = 'root';**  **$this->database = 'the\_school';** |

|  |  |  |  |
| --- | --- | --- | --- |
| Actions | | |  |
| **variables** | **type** | **description** | |
| **$servername $username $password $database** | Private Private Private private | Server name Server user name Server password Database name | |
| **methods** | **type** | **Required** | **description** |
| LogIn () | protected | $email | Log in validation |
| getAllInfo () | Protected | $table | Returns all info from $table |
| insertCourse () | Protected | $name, $description, $image, $table | Inserts a row to courses $table |
| insertStudent () | Protected | ($name, $phone, $email, $image, $table | Inserts a row to students $table |
| insertUser () | Protected | $name, $phone, $email, $image, $table, $role | Inserts a row to users $table |
| update\_Course () | Protected | $ID, $name, $description, $image, $table | Updates a row to courses $table |
| update\_Student () | Protected | $ID, $name, $phone, $email, $image, $table, $courses | Updates a row to students $table |
| update\_User () | Protected | $ID, $name, $phone, $email, $image, $table, $role | Updates a row to users $table- restricted to only 1 owner role |
| RemoveItem () | Protected | $table, $name | Deletes rom from $table |
| GetAssigned () | Protected | $ID | Returns all assigned courses to a specific student by $ID |

## 

|  |  |  |
| --- | --- | --- |
| Application | | |
| **methods** | **type** | **description** |
| All Actions methods | public | Interface to use Action methods as public |

## Functions

|  |  |  |
| --- | --- | --- |
| Application | | |
| **methods** | **type** | **description** |
| All Actions methods | public | Interface to use Action methods as public |

## 

|  |  |  |
| --- | --- | --- |
| Application | | |
| **methods** | **type** | **description** |
| All Actions methods | public | Interface to use Action methods as public |

## 

|  |  |
| --- | --- |
| fLetterUpperCase | |
| **requires** | **description** |
| string | Returns the same string as String (first letter uppercase) |

## 

|  |  |
| --- | --- |
| clearSelected | |
| **requires** | **description** |
|  | Clears the ‘selected’ class from <li> item |

## 

|  |  |
| --- | --- |
| showCancel | |
| **requires** | **description** |
|  | Shows the X icon to that triggers the onClick cancel |

## 

|  |  |
| --- | --- |
| flip | |
| **requires** | **description** |
| element | RotateX an element 90deg and returns after 300ms |

Glossary

## Conventions

* All requests and responses are in JSON format.

## Status Codes

All status codes are standard HTTP status codes. The below ones are used in this API.

2XX - Success of some kind

4XX - Error occurred in client’s part

5XX - Error occurred in server’s part

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 200 | OK |
| 201 | Created |
| 202 | Accepted (Request accepted, and queued for execution) |
| 400 | Bad request |
| 401 | Authentication failure |
| 403 | Forbidden |
| 404 | Resource not found |
| 405 | Method Not Allowed |
| 409 | Conflict |
| 412 | Precondition Failed |
| 413 | Request Entity Too Large |
| 500 | Internal Server Error |
| 501 | Not Implemented |
| 503 | Service Unavailable |