**API Documentation**

**Users & Groups Project**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Author** |
| 1.0 | 30-June-2019 | Peyman Ghader Kourehpaz |

## 

Index

[1. Login](#_bit7idv2ancx)

[Request](#_1cermjxbcuir)

[Response](#_8b7ijrpuamb8)

[2. Logout](#_l9zvgamyv7tu)

[Request](#_4g52516qq25k)

[Response](#_xj5121ptj10r)

[3. Get Groups](#_9t094e45sg90)

[Request](#_xll4ic1yxu5w)

[Response](#_47awg09ehquu)

[4. Add Group](#_9mc75gzfq3m3)

[Request](#_xtszbus9yomc)

[Response](#_xuhh3y4ufz7)

[5. Edit Group ( Get group info )](#_p8sbwmpv82e8)

[Request](#_iaps2dh7l2m2)

[Response](#_fvf6lulvf6wv)

[6. Update Group](#_ck649e58a33w)

[Request](#_yh0laj76z8yf)

[Response](#_at98vqz44vjx)

[7. Remove Group](#_rbn2ycy18z64)

[Request](#_eli1nsz7n7bv)

[Response](#_cns3unvmekn3)

[8. Get Group Users](#_75cxmkvwfdyt)

[Request](#_38rur58t3b6q)

[Response](#_fv28dmz2hp23)

[9. Get Users](#_8pc0kw1ty3cs)

[Request](#_bnz3k8uxw5my)

[Response](#_td8dtj6v8j77)

[10. Add User](#_c06sgkxyyov7)

[Request](#_p5l1uigi4w9o)

[Response](#_9d5ruzp9v5n6)

[11. Edit User ( Get User Info with Group Memberships )](#_jwmxife7q7lp)

[Request](#_t8547oxold7v)

[Response](#_63qvmscs4u0a)

[12. Update User](#_wqppncm6u3mt)

[Request](#_wiiy46g669v2)

[Response](#_8rxma6nupdot)

[13. Remove User](#_5tsmj5e046tm)

[Request](#_lavng8839tws)

[Response](#_uko0r49f98zu)

## 

## 

Methods

## 1. login

Authenticate the user with the system and return access token key , token type with expiration date.

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/auth/login |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| HEAD  HEAD  POST  POST  POST | Content-Type  X-Requested-With  email  Password  remember\_me | application/json  XMLHttpRequest  String (required)  String (required)  Boolean |

**Authorization**

Authorization must be sent with all client requests. Authorization helps the server to validate the request source.

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | Example response:  {  "access\_token": "token",  "token\_type": "Bearer",  "expires\_at": "2019-07-07 07:22:22"  }  Authorization (**string**) - include ( token\_type + " " + access\_token). all further API calls must have this key in header |
| 401 | {"error":"Unauthorised"} |

## 2. logout

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **ANY**  **-POST**  **-GET** | api/auth/logout |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| HEAD | Authorization | String:  "token\_type" + " " + "access\_token" |

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| success | {  "message": "Successfully logged out"  } |

## 3. Get Groups

Get groups list.

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **Get** | api/v1/groups |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| HEAD  GET | Authorization  page | String:  "token\_type" + " " + "access\_token"  number |

**Authorization**

The Authorization that was given in response to /api/auth/login

**page**

Groups are paginated by 50 , if request has more than 50 groups , then we should specify page value in our requests to paginate them by 50 until we get all groups.

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| success | **Response will be an object containing the list of Groups (array)**  **Paginated by 50 .**  Example response:  {  "success": 1,  "response": {  "current\_page": 1,  "data": [  {  "id": 2,  "name": "instagram",  "created\_at": "2019-06-29 14:21:26",  "updated\_at": "2019-06-30 07:29:22"  },  {  "id": 3,  "name": "facebook",  "created\_at": "2019-06-29 21:33:50",  "updated\_at": "2019-06-29 21:33:50"  }  ],  "first\_page\_url": "http://laravelchallenge.test/api/v1/groups?page=1",  "from": 1,  "last\_page": 1,  "last\_page\_url": "http://laravelchallenge.test/api/v1/groups?page=1",  "next\_page\_url": null,  "path": "http://laravelchallenge.test/api/v1/groups",  "per\_page": 50,  "prev\_page\_url": null,  "to": 6,  "total": 6  }  } |

## 4. Add Group

Add a new group.

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/v1/add\_group |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| HEAD  Post | Authorization  group\_name | String:  "token\_type" + " " + "access\_token"  String (required) |

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| success | **A Json array**  Example response:  {  "success": 1,  "response": {  "name": "My new group",  "updated\_at": "2019-06-30 08:25:59",  "created\_at": "2019-06-30 08:25:59",  "id": 10  }  } |
| error | {"err": 1,"msg": "Error , Wrong Parameters"} |

## 5. Edit Group

Get Information of a group for editing.

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | api/v1/edit\_group/{id} |

**ID**

Id of the group.

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| HEAD | Authorization | String:  "token\_type" + " " + "access\_token" |

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| success | **A json array**  Example response:  {  "success": 1,  "response": {  "id": 10,  "name": "Group Name",  "created\_at": "2019-06-30 08:25:59",  "updated\_at": "2019-06-30 08:25:59"  }  } |
| error | {"err": 1,"msg": "Group not found"} |

## 

## 6. Update Group

Update Information of a group for editing.

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/v1/edit\_group/{id} |

**ID**

Id of the group.

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| HEAD  post | Authorization  group\_name | String:  "token\_type" + " " + "access\_token"  String (required) |

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| success | **A json array**  Example response:  {  "success": 1,  "response": {  "id": 10,  "name": "My Updated Group",  "created\_at": "2019-06-30 08:25:59",  "updated\_at": "2019-06-30 08:25:59"  }  } |
| error | {"err": 1,"msg": "Group not found"} |

## 7. Remove Group

Remove a group.

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **ANY - POST - GET** | api/v1/remove\_group/{id} |

**ID**

Id of the group.

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| HEAD | Authorization | String:  "token\_type" + " " + "access\_token" |

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| success | **A json array**  Example response:  {  "success": 1,  "msg": "Group has been removed successfully"  } |
| error | {"err": 2,"msg": "Error : the selected group has members ! we can not remove a group with members ! make it empty and try again."} |
| error | {"err": 1,"msg": "Group not found"} |

## 8. Get Group Users

Get User members of a group with detail ( every users group membership ).

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | api/v1/group\_users/{id} |

**ID**

Id of the group.

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| HEAD  GET | Authorization  page | String:  "token\_type" + " " + "access\_token"  number |

**page**

Users are paginated by 10 , if request has more than 10 users , then we should specify page value in our requests to paginate them by 10 until we get all users with their group membership relationships for a group.

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| success | **A json array**  Example response:  {  "success": 1,  "response": {  "current\_page": 1,  "data": [  {  "id": 10,  "name": "payam ghader kourehpaz",  "created\_at": "2019-06-29 22:09:07",  "updated\_at": "2019-06-29 22:09:07",  "groups": [  {  "id": 2,  "name": "instagram",  "created\_at": "2019-06-29 14:21:26",  "updated\_at": "2019-06-30 07:29:22",  "pivot": {  "suser\_id": 10,  "sgroup\_id": 2  }  },  {  "id": 7,  "name": "kurdish boys",  "created\_at": "2019-06-30 07:29:06",  "updated\_at": "2019-06-30 07:29:06",  "pivot": {  "suser\_id": 10,  "sgroup\_id": 7  }  }  ]  }  ],  "first\_page\_url": "http://laravelchallenge.test/api/v1/group\_users/2?page=1",  "from": 1,  "last\_page": 1,  "last\_page\_url": "http://laravelchallenge.test/api/v1/group\_users/2?page=1",  "next\_page\_url": null,  "path": "http://laravelchallenge.test/api/v1/group\_users/2",  "per\_page": 10,  "prev\_page\_url": null,  "to": 2,  "total": 2  }  } |
| error | {"err": 1,"msg": "Group not found"} |

## 9. Get Users

Get User lists with pagination .

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **ANY**  **-GET**  **-POST** | api/v1/users |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| HEAD  GET | Authorization  page | String:  "token\_type" + " " + "access\_token"  number |

**page**

Users are paginated by 10 , if request has more than 10 users , then we should specify page value in our requests to paginate them by 10 until we get all users with their group membership relationships for a group.

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| success | **A json array**  Example response:  {  "success": 1,  "response": {  "current\_page": 1,  "data": [  {  "id": 10,  "name": "peyman ghader kourehpaz",  "created\_at": "2019-06-29 22:09:07",  "updated\_at": "2019-06-29 22:09:07",  "groups": [  {  "id": 2,  "name": "instagram",  "created\_at": "2019-06-29 14:21:26",  "updated\_at": "2019-06-30 07:29:22",  "pivot": {  "suser\_id": 10,  "sgroup\_id": 2  }  }  }  ]  }  ],  "first\_page\_url": "http://laravelchallenge.test/api/v1/users?page=1",  "from": 1,  "last\_page": 1,  "last\_page\_url": "http://laravelchallenge.test/api/v1/users?page=1",  "next\_page\_url": null,  "path": "http://laravelchallenge.test/api/v1/users",  "per\_page": 20,  "prev\_page\_url": null,  "to": 3,  "total": 3  }  } |

## 

## 10. Add User

Add a new User. with group memberships .

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/v1/add\_user |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| HEAD  POST  POST | Authorization  User\_name  groups.\* | String:  "token\_type" + " " + "access\_token"  String (required)  Array [integer] |

**groups.\***

An array of specified groups with group value as ID , through this array value parameter , we add user membership to the specified groups through group IDs in the array value .

Example parameter:

groups[] = 2

Groups[] = 3

Here we add User to groups with IDs of 2 and 3 .

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| success | **A Json array**  Example response:  {  "success": 1,  "response": {  "name": "new user",  "updated\_at": "2019-06-30 12:50:40",  "created\_at": "2019-06-30 12:50:40",  "id": 19  }  } |

## 11. Edit User

Get Information of a user for editing with group memberships.

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | api/v1/edit\_user/{id} |

**ID**

Id of the user.

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| HEAD | Authorization | String:  "token\_type" + " " + "access\_token" |

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| success | **A json array**  Example response:  {  "success": 1,  "response": {  "id": 19,  "name": "new user",  "created\_at": "2019-06-30 12:50:40",  "updated\_at": "2019-06-30 12:50:40",  "groups": [  {  "id": 2,  "name": "instagram",  "created\_at": "2019-06-29 14:21:26",  "updated\_at": "2019-06-30 07:29:22",  "pivot": {  "suser\_id": 19,  "sgroup\_id": 2  }  }  ]  }  } |
| error | {"err": 1,"msg": "User not found"} |

## 

## 12. Update User

Update a user or it’s group memberships .

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/v1/edit\_user/{id} |

**ID**

Id of the user.

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| HEAD  POST  POST | Authorization  User\_name  groups.\* | String:  "token\_type" + " " + "access\_token"  String (required)  Array [integer] |

**groups.\***

An array of specified groups with group value as ID , through this array value parameter , we add user membership to the specified groups through group IDs in the array value .

Example parameter:

groups[] = 2

Groups[] = 3

Here we add User to groups with IDs of 2 and 3 .

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| success | **A json array**  Example response:  {  "success": 1,  "response": {  "id": 19,  "name": "update name and membership",  "created\_at": "2019-06-30 12:50:40",  "updated\_at": "2019-06-30 12:59:27",  "groups": [  {  "id": 2,  "name": "instagram",  "created\_at": "2019-06-29 14:21:26",  "updated\_at": "2019-06-30 07:29:22",  "pivot": {  "suser\_id": 19,  "sgroup\_id": 2  }  },  {  "id": 3,  "name": "facebook",  "created\_at": "2019-06-29 21:33:50",  "updated\_at": "2019-06-29 21:33:50",  "pivot": {  "suser\_id": 19,  "sgroup\_id": 3  }  }  ]  }  } |
| error | {"err": 1,"msg": "User not found"} |

## 13. Remove User

Remove a User with it’s membership relationships.

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **ANY - POST - GET** | api/v1/remove\_user/{id} |

**ID**

Id of the user.

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| HEAD | Authorization | String:  "token\_type" + " " + "access\_token" |

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| success | **A json array**  Example response:  {  "success": 1,  "msg": "User has been removed successfully"  } |
| error | {"err": 1,"msg": "User not found"} |