# Users

## User management

|  |
| --- |
| **POST** |
| **Path**: /api/user |
| **Description**: Create a user |
| **Method**: POST |
| **Query Parameters**:   |  |  | | --- | --- | | name | **Required**: true  **Type**: string  **Description**: Users name | | birth\_date | **Required**: true  **Type**: string  **Description**: Birth date in format “dd/mm/yyyy” | | avatar\_path | **Required**: true  **Type**: string  **Description**: Picture of user | | password | **Required**: true  **Type**: Binary  **Description**: MD5 Hashed password | | user\_tags | **Required**: true  **Type**: string  **Description**: Comma separated user tags | |
| **Response:**  Application/json   |  |  | | --- | --- | | 200 | Successful user creation, return username  **{**  **“**username” :  **}** | | 401 | Invalid Credentials | |

|  |
| --- |
| **PUT** |
| **Path**: /api/user/{username} |
| **Description**: Update users name, avatar, points |
| **Path Parameters:**   |  |  | | --- | --- | | username | **Required**: true  **Type**: string  **Description**: Unique user username | |
| **Query Parameters**:   |  |  | | --- | --- | | name | **Required**: false  **Type**: string  **Description**: User full name | | avatar\_path | **Required**: false  **Type**: string  **Description**: | | reward\_points | **Required**: false  **Type**: int  **Description**: updated reward points | | tags | **Required**: false  **Type**: string  **Description**: full list of users tags, comma seperated | |
| **Response:**   |  |  | | --- | --- | | 200 | Successful update, return properties edited and new values  {  “name”:  “avatar”:  “reward\_points”:  “tags”  } | | 401 | Invalid credentials | |

|  |
| --- |
| **GET** |
| **Path**: /api/user/{username} |
| **Description**: Returns information on user |
| **Path Parameters:**   |  |  | | --- | --- | | username | **Required**: true  **Type**: string  **Description**: Unique user username | |
| **Response:**  Application/json   |  |  | | --- | --- | | 200 | Returns:   * Name * Date of birth * Avatar * Tags   For example:  {  “name” : ‘Samuel Thomas’  “birth\_date” : ‘04/12/1999’  “avatar” : binaryImage()  “user\_tags” : “cardio, gym, running”  } | | 401 | Invalid Credentials | |

|  |
| --- |
| **DELETE** |
| **Path**: /api/user/{username} |
| **Description**: |
| **Path Parameters:**   |  |  |  |  | | --- | --- | --- | --- | | username | **Required**: true  **Type**: string  **Description**: Unique user username |  | **Required**:  **Type**:  **Description**: | |
| **Response:**   |  |  | | --- | --- | | 200 | User successfully deleted, returns deleted user username  {  “username” :  } | | 401 | Invalid Credentials | |

## Reward Points

|  |
| --- |
| **PUT** |
| **Path**: /api/user/{username}/reward |
| **Description**: Update a users reward points |
| **Path Parameters:**   |  |  | | --- | --- | | username | **Required**: true  **Type**: string  **Description**: Unique user username | |
| **Query Parameters**:   |  |  | | --- | --- | | reward\_points | **Required**: true  **Type**: int  **Description**: New number of reward points | |
| **Response:**  Application/json   |  |  | | --- | --- | | 200 | User rewards updates successfully, returns updated points  {  “new\_points” : 12  } | | 401 | Invalid Credentials | |

|  |
| --- |
| **GET** |
| **Path**: /api/user/{username}/reward |
| **Description**: Get a users reward points |
| **Path Parameters:**   |  |  | | --- | --- | | username | **Required**: true  **Type**: string  **Description**: Unique user username | |
| **Response:**  Application/json   |  |  | | --- | --- | | 200 | Returns users reward points  {  “reward\_points” : 14  } | | 401 | Invalid Credentials | |

## User tags

|  |
| --- |
| **GET** |
| **Path**: /api/user/{username}/tags |
| **Description**: Returns all tags associated with a user |
| **Path Parameters:**   |  |  | | --- | --- | | username | **Required**: true  **Type**: string  **Description**: Unique user username | |
| **Response:**  Application/json   |  |  | | --- | --- | | 200 | Returns comma separated variable string tags  For example:  {  “user\_tags” : “gym, cardio, run”  } | | 401 | Invalid Credentials | |

|  |
| --- |
| **PUT** |
| **Path**: /api/user/{username}/tags |
| **Description**: Post new tags |
| **Path Parameters:**   |  |  | | --- | --- | | username | **Required**: true  **Type**: string  **Description**: Unique user username | |
| **Query Parameters**:   |  |  | | --- | --- | | user\_tags | **Required**: true  **Type**: string  **Description**: Comma separated new tags to add | |
| **Response:**  Application/json   |  |  | | --- | --- | | 200 | Returns comma separated list of all tags (original + yours)  For example:  {  “user\_tags” : “gym, cardio, run”  } | | 401 | Invalid Credentials | |

|  |
| --- |
| **DELETE** |
| **Path**: /api/user/{username}/tags |
| **Description**: Delete all tag |
| **Path Parameters:**   |  |  | | --- | --- | | username | **Required**: true  **Type**: string  **Description**: Unique user username | |
| **Response:**  Application/json   |  |  | | --- | --- | | 200 | All tags deleted, ie cleared | | 401 | Invalid Credentials | |

|  |
| --- |
| **DELETE** |
| **Path**: /api/user/{username}/tag |
| **Description**: Delete a single tag, case sensitive |
| **Path Parameters:**   |  |  | | --- | --- | | username | **Required**: true  **Type**: string  **Description**: Unique user username | |
| **Query Parameters**:   |  |  | | --- | --- | | user\_tag | **Required**: true  **Type**: string  **Description**: Single tag to be removed | |
| **Response:**  Application/json   |  |  | | --- | --- | | 200 | Tag succesfully deleted, returns updated list of (all) tags  {  “user\_tags” :  } | | 401 | Invalid Credentials | |

# Posts and comments

## Posts

|  |
| --- |
| **POST** |
| **Path**: /api/post |
| **Description**: Add a new post |
| **Query Parameters**:   |  |  | | --- | --- | | username | **Required**: true  **Type**: string  **Description**: Username of user posting comment | | message | **Required**: true  **Type**: string  **Description**: Message of user posting comment | | photo\_path | **Required**: false  **Type**: string  **Description**: Video or image attached | | video\_path | **Required**: false  **Type**: string  **Description**: Video or image attached | | post\_tags | **Required**: true  **Type**: string  **Description**: Comma separate post tags | |
| **Response:**  Application/json   |  |  | | --- | --- | | 200 | Post successfully created, returns post\_id  Example:  {  “post\_id” : “1AFG83AERD”  } | | 401 | Invalid credentials | |

|  |
| --- |
| **PUT** |
| **Path**: /api/post/{post\_id}/likes |
| **Description**: Add likes to a post (increment by 1) |
| **Path Parameters:**   |  |  | | --- | --- | | post\_id | **Required**: true  **Type**: string  **Description**: ID associated with a post | |
|  |
| **Response:**  Application/json   |  |  | | --- | --- | | 200 | Likes successfully updated by 1, returns new likes  For example:  {  “post\_likes” : 1  } | | 401 | Invalid Credentials | |

|  |
| --- |
| **PUT** |
| **Path**: /api/post/{post\_id}/tags |
| **Description**: Add add new tags to a post |
| **Path Parameters:**   |  |  | | --- | --- | | post\_id | **Required**: true  **Type**: string  **Description**: new comma-seperated post tags | |
|  |
| **Response:**  Application/json   |  |  | | --- | --- | | 200 | List of comma separated post tags, old ones plus new ojes  For example:  {  “post\_tags” :  } | | 401 | Invalid Credentials | |

|  |
| --- |
| **GET** |
| **Path**: /api/post/{post\_id} |
| **Description**: Get post (information) |
| **Path Parameters:**   |  |  | | --- | --- | | post\_id | **Required**: true  **Type**: string  **Description**: ID associated with a post | |
| **Response:**   |  |  | | --- | --- | | 200 | Returns information on post  {  “username”:  “media”: highleg  “message” : “Here’s how to do some fitness”  “likes”: 0  } | | 401 | Invalid credentials | |

|  |
| --- |
| **GET** |
| **Path**: /api/post/{post\_id}/comments |
| **Description**: Get all comments for a specific post |
| **Path Parameters:**   |  |  | | --- | --- | | post\_id | **Required**: true  **Type**: string  **Description**: ID associated with a post | |
| **Response:**   |  |  | | --- | --- | | 200 | Returns list of all comments for particular post  [  {  “comment\_id”:  “username”:  “message”  },  {  “comment\_id”:  “username”:  “message”  }  ] | | 401 | Invalid credentials | |

|  |
| --- |
| **GET** |
| **Path**: /api/user/{username}/posts |
| **Description**: Get all posts associated with user |
| **Path Parameters:**   |  |  | | --- | --- | | username | **Required**: true  **Type**: string  **Description**: Username of a valid user | |
| **Response:**   |  |  | | --- | --- | | 200 | Returns list of all users posts  [  {  “post\_id”:  “username”:  “message”  },  {  “post\_id”:  “username”:  “message”  }  ] | | 401 | Invalid credentials | |

|  |
| --- |
| **GET** |
| **Path**: /api/posts |
| **Description**: Get all posts, filtered by query parameters if needed |
| **Query parameters:**   |  |  | | --- | --- | | tags | **Required**: false  **Type**: string  **Description**: Comma separated list of search tags | | name | **Required**: false  **Type**: string  **Description**: Name to search | |
| **Response:**   |  |  | | --- | --- | | 200 | Returns list of all users posts. If query parameters set, returns filtered list of posts  [  {  “post\_id”:  “username”:  “message”  },  {  “post\_id”:  “username”:  “message”  }  ] | | 401 | Invalid credentials | |

|  |
| --- |
| **DELETE** |
| **Path**: /api/post/{post\_id} |
| **Description**: Delete a post, including associated comments |
| **Path Parameters:**   |  |  | | --- | --- | | post\_id | **Required**: true  **Type**: string  **Description**: ID associated with a post | |
| **Response:**  Application/json   |  |  | | --- | --- | | 200 | Post successfully deleted, returns id of deleted post  {  “post\_id”  } | | 401 | Invalid Credentials | |

|  |
| --- |
| **DELETE** |
| **Path**: /api/comment/{comment\_id} |
| **Description**: Delete a single comment |
| **Path Parameters:**   |  |  | | --- | --- | | comment\_id | **Required**: true  **Type**: string  **Description**: ID associated with a comment | |
| **Response:**  Application/json   |  |  | | --- | --- | | 200 | Comment successfully deleted, returns id of deleted comment  {  “comment\_id”  } | | 401 | Invalid Credentials | |

|  |
| --- |
| **POST** |
| **Path**: /api/post/{post\_id}/comment |
| **Description**: Add a new comment to a post |
| **Path Parameters**:   |  |  | | --- | --- | | post\_id | **Required**: true  **Type**: string  **Description**: ID associated with a comment | |
| **Query parameters**:   |  |  | | --- | --- | | username | **Required**: true  **Type**: string  **Description**: Username of user posting comment | | message | **Required**: true  **Type**: string  **Description**: Message of user posting comment | |
| **Response:**  Application/json   |  |  | | --- | --- | | 200 | Comment successfully created, returns comment\_id  Example:  {  “comment\_id” : “1AFG83AERD”  } | | 401 | Invalid credentials | |

|  |
| --- |
| **DELETE** |
| **Path**: /api/post/{post\_id}/tag |
| **Description**: Remove tag from post |
| **Path Parameters:**   |  |  | | --- | --- | | post\_id | **Required**: true  **Type**: string  **Description**: new comma-seperated post tags | |
| **Query Parameters:**   |  |  | | --- | --- | | post\_tag | **Required**: true  **Type**: string  **Description**: tag to remove | |
| **Response:**  Application/json   |  |  | | --- | --- | | 200 | List of comma separated post tags, ie old ones minues the deleted tag  For example:  {  “post\_tags” :  } | | 401 | Invalid Credentials | |

# Advert

|  |
| --- |
| **GET** |
| **Path**: /api/advert |
| **Description**: Get an advert |
| **Response:**   |  |  | | --- | --- | | 200 | Returns random advert. Video\_path and Photo\_path only returned if avaliable  {  “advert\_id”:  “name”:  “message”:  “photo\_path”:  “link”:  “video\_path”:  } | | 401 | Invalid credentials | |

|  |
| --- |
| **POST** |
| **Path**: /api/advert |
| **Description**: Post an advert |
| **Response:**   |  |  | | --- | --- | | 200 | Returns random advert. Video\_path and Photo\_path only returned if avaliable  {  “name”:  “message”:  “photo\_path”:  “link”:  “video\_path”:  “frequency”:  } | | 401 | Invalid credentials | |

|  |
| --- |
| **DELETE** |
| **Path**: /api/advert/{advert\_id} |
| **Description**: Delete an advert |
| **Response:**   |  |  | | --- | --- | | 200 | Returns deleted advert ID | | 401 | Invalid credentials | |