

## **Authorization Parameters**

Whenever user needs to perform an API call which requires, User needs to pass the following details whether it is GET / POST / UPDATE.

Parameters:

```
{
  "role" : "user",
  "email" : "arjunthedragon@gmail.com",
  "auth_token" : "zhZAT_5WaKhT9gLsiJk-"
}
```

For eg: in the user update API, we include the user specific authentication parameters whenever the API requires it. Check the Authentication parameters needed section.

```
{
  "role" : "user",
  "email" : "arjunthedragon@gmail.com",
  "auth_token" : "zhZAT_5WaKhT9gLsiJk-",
  "user": {
    "current_password": "qwerty123",
    "profile_attributes": {
      "dob_day": 1,
      "dob_month": 6,
      "dob_year": 1992
    },
    "phone_number": "9633423388"
  }
}
```

## **User Profile Management APIs**

### **1) User Sign In**

This API is meant to perform the sign in. User needs provide his / her credentials and is responded with user session details. In the user session details, you will find the authentication token which needs to be used whenever user needs to perform API call which requires authentication.

Request URL : <Base URL>/users/sign\_in.json

Request Method: POST

Authentication Parameters needed: No

Request Body:

```
{
  "user": {
    "email" : "arjunthedragon@gmail.com",
    "password" : "qwerty123"
  }
}
```

Response:

User json response with authentication token, this authentication token needs to be used in every API calls wherever it asks for authentication parameters. Please refer the authorization parameters section above.

## 2) User Registration

This API is meant to perform user registration.

Request URL : <Base URL>/users.json

Request Method: POST

Authentication Parameters needed: No

Request Body:

```
{
  "user": {
    "email": "thor@gmail.com",
    "password": "qwerty123",
    "password_confirmation": "qwerty123",
    "profile_attributes": {
      "first_name": "Thor",
      "last_name": "Odinson",
      "dob_day": 5,
      "dob_month": 10,
      "dob_year": 1992
    },
    "phone_number": "9916233040"
  }
}
```

Response:

User json response with authentication token, this authentication token needs to be used in every API calls wherever it asks for authentication parameters. Please refer the authorization parameters section above.

### 3) User Update

This API is used to update the user profile. Include only those parameters which you want to update. *Please note that mandatory attributes are made bold and italic.*

Request URL : <Base URL>/users/<user\_id>.json

Request Method: PUT

Authentication Parameters needed: YES

Request Body:

```
{
  "user": {
    "current_password": "qwerty123",
    "profile_attributes": {
      "id": 3,
      "dob_day": 1,
      "dob_month": 6,
      "dob_year": 1992
    },
    "phone_number": "9633423388"
  }
}
```

Response:

Status 204 will be received from the server.

### 4) User Delete

This API is used to delete the user profile. *Please note that mandatory attributes are made bold and italic.*

Request URL : <Base URL>/users/<user\_id>.json

Request Method: DELETE

Authentication Parameters needed: YES

Request Body:

None

Response :

Status 204 will be received from the server.

**5) User Logout**

This API is used to logout a user profile. *Please note that mandatory attributes are made bold and italic.*

Request URL : <Base URL>/users/sign\_out.json

Request Method: DELETE

Authentication Parameters needed: YES

Request Body:

Response:

Status 204 will be received from the server.

**6) User Details**

This API is used to show a user his / her details. *Please note that mandatory attributes are made bold and italic.*

Request URL : <Base URL>/users/<user\_id>.json

Request Method: GET

Authentication Parameters needed: YES

Request Body:

None

Response:

Status 200 will be received from the server. Also the user details of that particular user will be displayed.

**7) Show Badges**

This API is used to show a user's badges. *Please note that mandatory attributes are made bold and italic.*

Request URL : <Base URL>/users/<user\_id>/badges.json

Request Method: GET

Authentication Parameters needed: YES

Request Body:

None

Response:

Status 200 will be received from the server. Also the user badges of that particular user will be displayed.

## 8) Show My Favourites

This API is used to show a user's favourites. *Please note that mandatory attributes are made bold and italic.*

Request URL : <Base URL>/users/<user\_id>/favourites.json

Request Method: GET

Authentication Parameters needed: YES

Request Body:

None

Response:

Status 200 will be received from the server. Also the user favourites of that particular user will be displayed.

## Post Management APIs

### 1. Create a Post

This API is meant to create a new post. Question parameters are mandatory to be sent. The tags ids here indicates the ids of the tag in the backend. You can call a separate API to list the tags you want to call.

Request URL : <Base URL>/users/<user\_id>/1/posts.json

Request Method: POST

Authentication Parameters needed: YES

Request Body:

```
{  
  "role": "user",  
  "email": "arjunthedragon@gmail.com",  
}
```

```

"auth_token":"zhZAT_5WaKhT9gLsiJk-",
"post":{
  "title" : "How could be ML done with Cloud",
  "status" : "open",
  "post_category_id" : 1,
  "question_attributes" : {
    "description" : " I would like to know how to incorporate
ML with Cloud. Where do we find applications of ML in Cloud Computing"
  },
  "answer_requests_attributes": {
    "0":{
      "requested_user_id":4
    }
  },
  "tag_ids" : [1,2]
}
}

```

#### Response:

The response here is the list of posts of that particular user along with the newly created post, which mean these are the question or posts that this particular user has posted or asked. On successful creation, you will receive the status code 200.

## 2. Update a Post

This API is meant to update the post based on the post\_id mentioned in the url. You can even use this api to update the question details as well as the post details like post status, post\_category (if you want to move your post from one category to another).

Request URL : <Base URL>/users/<user\_id>/1/posts/<post\_id>.json

Request Method: PUT

Authentication Parameters needed: YES

#### Request Body:

```

{
  "role":"user",
  "email":"arjunthedragon@gmail.com",
  "auth_token":"zhZAT_5WaKhT9gLsiJk-",
  "post":{

```

```

        "title" : "How could be ML done with Cloud",
        "status" : "open",
        "post_category_id" : 1,
        "question_attributes" : {
            "id":1,
            "description" : " I would like to know how to incorporate
ML with Cloud. Where do we find applications of ML in Cloud Computing"
        },
        "answer_requests_attributes": {
            "0":{
                "requested_user_id":4
            }
        },
        "tag_ids" : [1,2]
    }
}

```

Response:

You will receive status 204 on successful update

### 3. Delete a Post

This API needs to be called to delete the post.

Request URL : <Base URL>/users/<user\_id>/1/posts/<post\_id>.json

Request Method: DELETE

Authentication Parameters needed: YES

Request Body:

None

Response:

The response here is the list of posts of that particular. I mean these are the question or posts that this particular user has posted or asked. On successful deletion, you will receive the status code 204.

### 4. List Post

This API will list all the posts. You can add additional parameters along to get the posts the way you want.

Request URL : <Base URL>/posts.json

Request Method: GET

Authentication Parameters needed: YES

Request Body:

```
{
  "role": "user",
  "email": "arjunthedragon@gmail.com",
  "auth_token": "zhZAT_5WaKhT9gLsiJk-",
}
```

Response:

The response here is the list all posts satisfying a particular criteria. I mean these are the question or posts that this particular user has posted or asked.

## 5. Show Post Details

This API is called to show the details of a post. You can call this API if you just want to see the details of a particular post instead of fetching the complete lists.

Request URL : <Base URL>/users/<user\_id>/1/posts/<post\_id>.json

Request Method: GET

Authentication Parameters needed: YES

Request Body:

```
{
  "role": "user",
  "email": "arjunthedragon@gmail.com",
  "auth_token": "zhZAT_5WaKhT9gLsiJk-",
}
```

Response:

The response here is the details of the particular (post with post\_id).

## 6. Favoured users



This API will list all the users who favours the post. It means these are the users which marked this post as favourite.

Request URL : <Base URL>/users/<user\_id>/posts/<post\_id>/favoured\_users.json

Request Method: GET

Authentication Parameters needed: YES

Request Body:

```
{
    "role": "user",
    "email": "arjunthedragon@gmail.com",
    "auth_token": "zhZAT_5WaKhT9gLsiJk-",
}
```

Response:

The response here is the list all users favouring the particular post.

#### 7. Mark as favourite

This API will mark the corresponding post as favourite for the user.

Request URL : <Base URL>/favourates.json

Request Method: POST

Authentication Parameters needed: YES

Request Body:

```
{
    "role": "user",
    "email": "arjunthedragon@gmail.com",
    "auth_token": "zhZAT_5WaKhT9gLsiJk-",
    "favourite": {
        "favoured_by_id": 1,
        "favourite_post_id": 13
    }
}
```

Response:

Here, status code 201 is returned for the successful operation. The response received is a JSON which gives information on the favourite entity which holds the post\_id and user\_id. It will also send back the favourite\_id which is needed when you want to delete the favourite from the backend.

8. Unmark as favourite

This API will unmark the corresponding post as favourite for the user.

Request URL : <Base URL>/favourites/<favourite\_id>.json

Request Method: POST

Authentication Parameters needed: YES

Request Body:

```
{
    "role":"user",
    "email":"arjunthedragon@gmail.com",
    "auth_token":"zhZAT_5WaKhT9gLsiJk-",
}
```

Response:

Here, status code 201 is returned for the successful operation. The response received is a JSON which gives information on the favourite entity which holds the post\_id and user\_id.

9. Request users for a post

This API is called to request a user to answer a question for a post. Please note the **requested\_user\_id** is the id of the user whom you want to request to answer this question. Pass this **requested\_user\_id** within the **answer\_requests\_attributes** section with value as **id**. Please make sure to include the **\_destroy** parameter as **true** as well for that answer\_request's id.

Request URL : <Base URL>/users/<user\_id>/posts/<post\_id>.json

Request Method: PUT

Authentication Parameters needed: YES

Request Body:

```
{
    "role":"user",
    "email":"arjunthedragon@gmail.com",
    "auth_token":"zhZAT_5WaKhT9gLsiJk-",
    "post":{
        "answer_requests_attributes": {
            "0":{
```

```

        "requested_user_id" : 1,
      }
    }
  }
}

```

Response:

You will receive status code 204 on successful completion of this operation.

## 10. Remove user requests for a post

This API is called to remove any existing user requests for a post. Please note the **requested\_user\_id** when you obtain the response for that post details. Pass that **requested\_user\_id** along with this API within the **answer\_requests\_attributes** section with value as **id**. Please make sure to include the **\_destroy** parameter as **true** as well for that answer\_request's id.

Request URL : <Base URL>/users/<user\_id>/posts/<post\_id>.json

Request Method: PUT

Authentication Parameters needed: YES

Request Body:

```

{
  "role":"user",
  "email":"arjunthedragon@gmail.com",
  "auth_token":"zhZAT_5WaKhT9gLsiJk-",
  "post":{
    "answer_requests_attributes": {
      "0":{
        "id":2,
        "_destroy":true
      }
    }
  }
}

```

Response:

You will receive status code 204 on successful completion of this operation.

## 11. Add Tags to a Post

This API is called to assign tags to a post. This API add tags with id 3 and 4 to the post. To obtain the tag ids in you application, you need to call the tag ids API separately.

Request URL : <Base URL>/users/<user\_id>/posts/<post\_id>.json

Request Method: PUT

Authentication Parameters needed: YES

Request Body:

```
{
  "role": "user",
  "email": "arjunthedragon@gmail.com",
  "auth_token": "zhZAT_5WaKhT9gLsiJk-",
  "post": {
    "tag_ids": [3, 4]
  }
}
```

Response:

On successful completion of this operation, you will be received with status code 204. Please save those **tag\_ids, post\_ids and tag\_post\_ids** somewhere in your application. You will need them if you need to remove tags assigned to a post.

## 12. Remove Tags from a Post

This API needs to be called to delete the tag associated with the post. **tag\_ids** should have the value of the tag ids which needs to be removed associated from the post. Please note that this does not remove the tags as a whole. It just removes the association with the post.

Request URL : <Base URL>/tag\_posts/<tag\_post\_id>.json

Request Method: POST

Authentication Parameters needed: YES

Request Body:

```
{
  "role": "user",
  "email": "arjunthedragon@gmail.com",
  "auth_token": "zhZAT_5WaKhT9gLsiJk-",
  "tag_ids": [2,3]
}
```

Response:

On successful completion of the operation. The server returns with status code 200 along the updated tags associated with the posts.

### 13. List My Posts

This API could be called to list all tags of the currently signed in user. Here, <user\_id> mentioned in the URL should be the id of the user.

Request URL : <Base URL>/users/<user\_id>/posts.json

Request Method: GET

Authentication Parameters needed: YES

Request Body:

```
{
    "role": "user",
    "email": "arjunthedragon@gmail.com",
    "auth_token": "zhZAT_5WaKhT9gLsiJk-",
}
```

Response:

The response here is the list of posts of that particular user. I mean these are the questions or posts that this particular user has posted or asked.

### 14. List Posts I answered

This API could be called to list all tags the currently user has given an answer for. Here, <user\_id> mentioned in the URL should be the id of the user.

Request URL : <Base URL>/users/<user\_id>/answer\_posts.json

Request Method: GET

Authentication Parameters needed: YES

Request Body:

```
{
    "role": "user",
    "email": "arjunthedragon@gmail.com",
    "auth_token": "zhZAT_5WaKhT9gLsiJk-",
}
```

```
}
```

#### Response:

The response here is the list of posts that the particular user has given an answer.

15. List Interested Posts

16. List Popular Posts

## **Answer Management APIs**

### **1. Create an Answer**

This API is meant to create answer for a particular from a particular user. Question parameters are mandatory to be send. The tags ids here indicates the ids of the tag in the backend. You can call a separate API to list the tags you want to call. Here, description includes what answer you need to pass, and answered\_by\_id includes the id of the user who is answering.

Request URL : <Base URL>/posts/<post\_id>/answers.json

Request Method: POST

Authentication Parameters needed: YES

#### Request Body:

```
{
  "role": "user",
  "email": "arjunthedragon@gmail.com",
  "auth_token": "zhZAT_5WaKhT9gLsiJk-",
  "answer": {
    "answered_by_id": 1,
    "description": "Yes, Masterwarlord, you are the emperor of this
universe"
  }
}
```

#### Response:

The response here is the is the newly created answer for that post by this user. On successful creation, you will receive the status code 201.

### **2. Update an Answer**

This API is meant to create answer for a particular from a particular user. Question parameters are mandatory to be send. Here, description includes what answer you need to pass, and answered\_by\_id includes the id of the user who is answering.

Request URL : <Base URL>/posts/<post\_id>/answers/<answer\_id>.json

Request Method: PUT

Authentication Parameters needed: YES

Request Body:

```
{
  "role": "user",
  "email": "arjunthedragon@gmail.com",
  "auth_token": "zhZAT_5WaKhT9gLsiJk-",
  "answer": {
    "description": "Yes, Masterwarlord, you are the emperor of this
universe. But, you must know, as great powers come, so great responsibility."
  }
}
```

Response:

On successful creation, you will receive the status code 204.

### 3. Delete the Answer

This API is meant to delete the answer based on the answer\_id we pass in the url.

Request URL : <Base URL>/posts/<post\_id>/answers.json

Request Method: DELETE

Authentication Parameters needed: YES

Request Body:

```
{
  "role": "user",
  "email": "arjunthedragon@gmail.com",
  "auth_token": "zhZAT_5WaKhT9gLsiJk-",
}
```

#### Response:

The response here is the newly created answer for that post by this user. On successful creation, you will receive the status code 201.

#### 4. Upvote an Answer

This API needs to be called to upvote an answer. Here, all the parameters should be mandatory. The voterable must be "Answer" if you are upvoting an answer. voterable\_id means the id of the answer you need to upvote.

Request URL : <Base URL>/votes.json

Request Method: POST

Authentication Parameters needed: YES

#### Request Body:

```
{
  "role": "user",
  "email": "arjunthedragon@gmail.com",
  "auth_token": "zhZAT_5WaKhT9gLsiJk-",
  "vote": {
    "voterable_type": "Answer",
    "voterable_id": 4,
    "vote_type": "upvote",
    "voter_id": 4
  }
}
```

#### Response:

The response here is vote json including the vote id. On successful operation, you will receive the status code 201.

#### 5. Downvote an Answer

This API needs to be called to downvote an answer. Here, all the parameters should be mandatory. The voterable must be "Answer" if you are downvoting an answer. voterable\_id means the id of the answer you need to upvote.

Request URL : <Base URL>/votes.json

Request Method: POST



Authentication Parameters needed: YES

Request Body:

```
{
  "role": "user",
  "email": "arjunthedragon@gmail.com",
  "auth_token": "zhZAT_5WaKhT9gLsiJk-",
  "vote": {
    "voterable_type": "Answer",
    "voterable_id": 4,
    "vote_type": "downvote",
    "voter_id": 4
  }
}
```

Response:

The response here is vote json including the vote id. On successful operation, you will receive the status code 201.

#### 6. Remove an answer vote

This API needs to be called to remove a vote made to an answer.

Request URL : <Base URL>/votes/<vote\_id>.json

Request Method: DELETE

Authentication Parameters needed: YES

Request Body:

```
{
  "role": "user",
  "email": "arjunthedragon@gmail.com",
  "auth_token": "zhZAT_5WaKhT9gLsiJk-",
}
```

Response:

On successful deletion, you will receive the status code 204.

#### 7. Accept an Answer

This API is used to mark an answer accepted.

Request URL : <Base URL>/posts/<post\_id>/answers/<answer\_id>/accept.json

Request Method: POST

Authentication Parameters needed: YES

Request Body:

```
{
    "role":"user",
    "email":"arjunthedragon@gmail.com",
    "auth_token":"zhZAT_5WaKhT9gLsiJk-",
}
```

Response:

The response here is the is the message that the answer is accepted. On successful operation, you will receive the status code 200.

#### 8. Unaccept an answer

This API is used to mark an answer unaccepted.

Request URL : <Base URL>/posts/<post\_id>/answers/<answer\_id>/unaccept.json

Request Method: POST

Authentication Parameters needed: YES

Request Body:

```
{
    "role":"user",
    "email":"arjunthedragon@gmail.com",
    "auth_token":"zhZAT_5WaKhT9gLsiJk-",
}
```

Response:

The response here is the is the message that the answer is made unaccepted. On successful operation, you will receive the status code 200.

#### 9. Add Comment to a Answer

This API is meant to add a answer comment by a user. Here, "commentable\_id" is the id of the Answer, commentable\_type is fixed i.e "Answer". "commented\_by\_id" is the id of the user who is commenting. "description" indicates the description you want to include in the comment. Please make sure the commentable "type" and "id" do not mismatch.

Request URL : <Base URL>/comments.json

Request Method: POST

Authentication Parameters needed: YES

Request Body:

```
{
  "role": "user",
  "email": "arjunthedragon@gmail.com",
  "auth_token": "zhZAT_5WaKhT9gLsiJk-",
  {
    "role": "user",
    "email": "arjunthedragon@gmail.com",
    "auth_token": "zhZAT_5WaKhT9gLsiJk-",
    "comment": {
      "commentable_type": "Answer",
      "commentable_id": 1,
      "description": "This answer is superb",
      "commented_by_id": 1
    }
  }
}
```

Response:

On successful operation, you will receive status code 201 with the comment response. Please note down the id for later updating purpose.

## 10. Update Answer Comment

This API is meant to update a question comment.

Request URL : <Base URL>/comments/<comment\_id>.json

Request Method: PUT

Authentication Parameters needed: YES

Request Body:

```
{
  "role": "user",
  "email": "arjunthedragon@gmail.com",
  "auth_token": "zhZAT_5WaKhT9gLsiJk-",
  "comment": {
```

```
        "description": "This answer is hilarious"
    }
}
```

Response:

On successful operation, you will receive status code 201.

## 11. Delete Answer Comment

This API is meant to delete the answer comment.

Request URL : <Base URL>/comments/<comments\_id>.json

Request Method: DELETE

Authentication Parameters needed: YES

Request Body:

```
{
    "role": "user",
    "email": "arjunthedragon@gmail.com",
    "auth_token": "zhZAT_5WaKhT9gLsiJk-",
}
```

Response:

On successful operation, you will receive status code 204

## 12. Add an Answer Edit Suggestion

This API is meant to add a suggestion for a answer by a user. Here, “edit\_suggestionable\_id” is the id of the Answer, edit\_suggestionable\_type is fixed i.e “Answer”. “answer\_suggester\_id” is the id of the user who is suggesting it. “description” indicates the description you want to give for editing the description. Please make sure the edit\_suggestionable “type” and “id” do not mismatch. Please also ensure that you send a change back to the server. Otherwise, the system will respond for “no changes founded”.

Request URL : <Base URL>/edit\_suggestions.json

Request Method: POST

Authentication Parameters needed: YES

Request Body:

```

{
  "role":"user",
  "email":"arjunthedragon@gmail.com",
  "auth_token":"zhZAT_5WaKhT9gLsiJk-",
  "edit_suggestion":{
    "edit_suggestionable_type":"Answer",
    "edit_suggestionable_id":1,
    "answer_suggester_id":1,
    "description":"This is the updated description suggested by
user with user id 1 for answer with answer id 1"
  }
}

```

#### Response:

On successful operation, you will receive status code 201 with the edit\_suggestion response. Please note down the id for later updating purpose. Edit Suggestion made by default will be open.

### 13. Update Answer Edit Suggestion

This API is meant to update a suggestion for a answer by a user. This will update the edit suggestion with the edit\_suggestion\_id mentioned in the url. Please note that you are not allowed to update an edit suggestion which is already accepted.

Request URL : <Base URL>/edit\_suggestions/<edit\_suggestion\_id>.json

Request Method: PUT

Authentication Parameters needed: YES

#### Request Body:

```

{
  "role":"user",
  "email":"arjunthedragon@gmail.com",
  "auth_token":"zhZAT_5WaKhT9gLsiJk-",
  "edit_suggestion":{
    "title":"This is the updated title for the suggested title made",
    "description":"This is the updated description suggested by
user with user id for answer with answer id 1"
  }
}

```

#### Response:

On successful operation, you will receive status code 201.

#### 14. Delete Question Edit Suggestion

This API is meant to delete the edit\_suggestion.

Request URL : <Base URL>/edit\_suggestions.json

Request Method: DELETE

Authentication Parameters needed: YES

Request Body:

```
{
  "role": "user",
  "email": "arjunthedragon@gmail.com",
  "auth_token": "zhZAT_5WaKhT9gLsiJk-",
}
```

Response:

On successful operation, you will receive status code 204

#### 15. Accept Edit Suggestion

This API is mark an suggestion to be accepted. The basic usage whenever an edit suggestion is accepted.

Request URL : <Base URL>/edit\_suggestions/<edit\_suggestion\_id>/accept.json

Request Method: POST

Authentication Parameters needed: YES

Request Body:

```
{
  "role": "user",
  "email": "arjunthedragon@gmail.com",
  "auth_token": "zhZAT_5WaKhT9gLsiJk-",
}
```

Response:

On successful operation, you will receive status code 200 with a message that the edit suggestion is accept.

## 16. Reject Edit Suggestion

This API is mark an suggestion to be reopened. The basic usage whenever an edit suggestion is rejected.

Request URL : <Base URL>/edit\_suggestions/<edit\_suggestion\_id>/reject.json

Request Method: POST

Authentication Parameters needed: YES

Request Body:

```
{
  "role": "user",
  "email": "arjunthedragon@gmail.com",
  "auth_token": "zhZAT_5WaKhT9gLsiJk-",
}
```

Response:

On successful operation, you will receive status code 200 with a message that the edit suggestion is reopened.

## 17. Mark Edit Suggestion as Open

This API is mark an suggestion to be reopened. The basic usage whenever an edit suggestion is rejected. It needs to be reopened.

Request URL : <Base URL>/edit\_suggestions/<edit\_suggestion\_id>/reopen.json

Request Method: POST

Authentication Parameters needed: YES

Request Body:

```
{
  "role": "user",
  "email": "arjunthedragon@gmail.com",
  "auth_token": "zhZAT_5WaKhT9gLsiJk-",
}
```

Response:

On successful operation, you will receive status code 200 with a message that the edit suggestion is reopened.

## Question Management APIs

### 1. Update a Question

This API is meant to update the question. You need to mention the `post_id` in the url. Please mention `id` parameter inside the `question_attributes` parameter.

Request URL : <Base URL>/users/<user\_id>/1/posts/<post\_id>.json

Request Method: PUT

Authentication Parameters needed: YES

Request Body:

```
{
  "role": "user",
  "email": "arjunthedragon@gmail.com",
  "auth_token": "zhZAT_5WaKhT9gLsiJk-",
  "post": {
    "title": "I am the title to be updated for this question",
    "question_attributes": {
      "id": 1,
      "description": "I would like to know how to incorporate
ML with Cloud. Where do we find applications of ML in Cloud Computing"
    }
  }
}
```

Response:

You will receive status code 204 on successful update.

### 2. Upvote a Question

This API needs to be called to upvote an question. Here, all the parameters should be mandatory. The voterable must be "Question" if you are upvoting an question. `voterable_id` means the id of the question you need to upvote.

Request URL : <Base URL>/votes.json



Request Method: POST

Authentication Parameters needed: YES

Request Body:

```
{
  "role": "user",
  "email": "arjunthedragon@gmail.com",
  "auth_token": "zhZAT_5WaKhT9gLsiJk-",
  "vote": {
    "voterable_type": "Question",
    "voterable_id": 4,
    "vote_type": "upvote",
    "voter_id": 4
  }
}
```

Response:

The response here is vote json including the vote id. On successful operation, you will receive the status code 201.

### 3. Downvote a Question

This API needs to be called to downvote an question. Here, all the parameters should be mandatory. The voterable must be "Question" if you are downvote an question. voterable\_id means the id of the question you need to downvote.

Request URL : <Base URL>/votes.json

Request Method: POST

Authentication Parameters needed: YES

Request Body:

```
{
  "role": "user",
  "email": "arjunthedragon@gmail.com",
  "auth_token": "zhZAT_5WaKhT9gLsiJk-",
  "vote": {
    "voterable_type": "Question",
    "voterable_id": 13,
    "vote_type": "downvote",
  }
}
```

```
        "voter_id":4
    }
}
```

Response:

The response here is vote json including the vote id. On successful operation, you will receive the status code 201.

4. Remove a question vote

This API needs to be called to remove a vote made to a question.

Request URL : <Base URL>/votes/<vote\_id>.json

Request Method: DELETE

Authentication Parameters needed: YES

Request Body:

```
{
    "role":"user",
    "email":"arjunthedragon@gmail.com",
    "auth_token":"zhZAT_5WaKhT9gLsiJk-",
}
```

Response:

On successful deletion, you will receive the status code 204.

5. Add Comment to a Question

This API is meant to add a question comment by a user. Here, "commentable\_id" is the id of the Question, commentable\_type is fixed i.e "Question". "commented\_by\_id" is the id of the user who is commenting. "description" indicates the description you want to include in the comment. Please make sure the commentable "type" and "id" do not mismatch.

Request URL : <Base URL>/edit\_suggestions.json

Request Method: POST

Authentication Parameters needed: YES

Request Body:

```
{
  "role":"user",
  "email":"arjunthedragon@gmail.com",
  "auth_token":"zhZAT_5WaKhT9gLsiJk-",
  {
    "role":"user",
    "email":"arjunthedragon@gmail.com",
    "auth_token":"zhZAT_5WaKhT9gLsiJk-",
    "comment":{
      "commentable_type":"Question",
      "commentable_id":1,
      "description":"This question is superb",
      "commented_by_id":1
    }
  }
}
```

#### Response:

On successful operation, you will receive status code 201 with the comment response. Please note down the id for later updating purpose. Edit Suggestion made by default will be open.

## 6. Update Question Comment

This API is meant to update the question comment.

Request URL : <Base URL>/comments/<comment\_id>.json

Request Method: PUT

Authentication Parameters needed: YES

#### Request Body:

```
{
  "role":"user",
  "email":"arjunthedragon@gmail.com",
  "auth_token":"zhZAT_5WaKhT9gLsiJk-",
  "comment":{
    "description":"This question is hilarious"
  }
}
```

#### Response:

On successful operation, you will receive status code 201.

## 7. Delete Question Comment

This API is meant to delete the question comment.

Request URL : <Base URL>/comments/<comments\_id>.json

Request Method: DELETE

Authentication Parameters needed: YES

Request Body:

```
{
    "role":"user",
    "email":"arjunthedragon@gmail.com",
    "auth_token":"zhZAT_5WaKhT9gLsiJk-",
}
```

Response:

On successful operation, you will receive status code 204

## 8. Add a Question Edit Suggestion

This API is meant to add a suggestion for a question by a user. Here, "edit\_suggestionable\_id" is the id of the Question, edit\_suggestionable\_type is fixed i.e "Question". "answer\_suggester\_id" is the id of the user who is suggesting it. "description" indicates the description you want to give for editing the description. Please make sure the edit\_suggestionable "type" and "id" do not mismatch. Please also ensure that you send a change back to the server. Otherwise, the system will respond for "no changes founded".

Request URL : <Base URL>/edit\_suggestions.json

Request Method: POST

Authentication Parameters needed: YES

Request Body:

```
{
    "role":"user",
    "email":"arjunthedragon@gmail.com",
    "auth_token":"zhZAT_5WaKhT9gLsiJk-",
    "edit_suggestion":{
        "edit_suggestionable_type":"Question",
        "edit_suggestionable_id":13,
```

```

        "question_suggester_id":1,
        "description":"This is the updated description suggested by
user with user id 1 for question with question id 13"
        "title":"this is the updated title suggested by user with user id 1
for question with question id 13"
    }
}

```

Response:

On successful operation, you will receive status code 201 with the edit\_suggestion response. Please note down the id for later updating purpose. Edit Suggestion made by default will be open.

## 9. Update Question Edit Suggestion

This API is meant to update a suggestion for a question by a user. This will update the edit suggestion with the edit\_suggestion\_id mentioned in the url. Please note that you are not allowed to update an edit suggestion which is already accepted.

Request URL : <Base URL>/edit\_suggestions/<edit\_suggestion\_id>.json

Request Method: PUT

Authentication Parameters needed: YES

Request Body:

```

{
    "role":"user",
    "email":"arjunthedragon@gmail.com",
    "auth_token":"zhZAT_5WaKhT9gLsiJk-",
    "edit_suggestion":{
        "description":"This is the updated description suggested by
user with user id for answer with answer id 1"
    }
}

```

Response:

On successful operation, you will receive status code 201.

## 10. Delete Question Edit Suggestion

This API is meant to delete the edit\_suggestion.

Request URL : <Base URL>/edit\_suggestions.json

Request Method: DELETE

Authentication Parameters needed: YES

Request Body:

```
{
    "role":"user",
    "email":"arjunthedragon@gmail.com",
    "auth_token":"zhZAT_5WaKhT9gLsiJk-",
}
```

Response:

On successful operation, you will receive status code 204

#### 11. Accept Question Edit Suggestion

This API is mark an suggestion to be accepted. The basic usage whenever an edit suggestion is rejected. It needs to be reopened. This will update the corresponding titles and description of Question.

Request URL : <Base URL>/edit\_suggestions/<edit\_suggestion\_id>/accept.json

Request Method: POST

Authentication Parameters needed: YES

Request Body:

```
{
    "role":"user",
    "email":"arjunthedragon@gmail.com",
    "auth_token":"zhZAT_5WaKhT9gLsiJk-",
}
```

Response:

On successful operation, you will receive status code 200 with a message that the edit suggestion is accepted.

#### 12. Reject Question Edit Suggestion

This API is mark an suggestion to be rejected. The basic usage whenever an edit suggestion is rejected. It needs to be reopened.

Request URL : <Base URL>/edit\_suggestions/<edit\_suggestion\_id>/reopen.json

Request Method: POST

Authentication Parameters needed: YES

Request Body:

```
{  
    "role":"user",  
    "email":"arjunthedragon@gmail.com",  
    "auth_token":"zhZAT_5WaKhT9gLsiJk-",  
}
```

Response:

On successful operation, you will receive status code 200 with a message that the edit suggestion is rejected.

Response:

On successful operation, you will receive status code 201 with the edit\_suggestion response. Please note down the id for later updating purpose. Edit Suggestion made by default will be open.

### 13. Mark Question Edit Suggestion as Open

This API is mark an suggestion to be reopened. The basic usage whenever an edit suggestion is rejected. It needs to be reopened.

Request URL : <Base URL>/edit\_suggestions/<edit\_suggestion\_id>/reopen.json

Request Method: POST

Authentication Parameters needed: YES

Request Body:

```
{  
    "role":"user",  
    "email":"arjunthedragon@gmail.com",  
    "auth_token":"zhZAT_5WaKhT9gLsiJk-",  
}
```

Response:

On successful operation, you will receive status code 200 with a message that the edit suggestion is reopened.

## Global APIs

### 1. Add Feedback

This API is meant to give feedback by a user. Here, "feedbackable\_id" is the id of the Question / Answer / Comment, commentable\_type is either "Question", "Answer" or "Comment". "user\_id" is the id of the user who is giving the feedback. Please make sure the commentable "type" and "id" do not mismatch. Here, feedback type indicates the type of feedback you are giving. The supported feedback\_types

- helpful
- abusive
- reopen
- close
- protect
- not\_constructive
- duplicate

Request URL : <Base URL>/feedbacks.json

Request Method: POST

Authentication Parameters needed: YES

Request Body:

```
{
  "role": "user",
  "email": "arjunthedragon@gmail.com",
  "auth_token": "zhZAT_5WaKhT9gLsiJk-",
  "feedback": {
    "feedbackable_type": "Question",
    "feedbackable_id": 13,
    "feedback_type": "abusive",
    "user_id": 4
  }
}
```

Response:

On successful operation, you will receive status code 201 with the feedback response. Please note down the id for later updating purpose.

### 2. Remove Feedback



This API is meant to delete a particular feedback.

Request URL : <Base URL>/feedbacks.json

Request Method: POST

Authentication Parameters needed: YES

Request Body:

```
{
    "role": "user",
    "email": "arjunthedragon@gmail.com",
    "auth_token": "zhZAT_5WaKhT9gLsiJk-",
}
```

Response:

On successful operation, You will receive status code 204.

### 3. List all Departments

This API is meant to list all the departments.

Request URL : <Base URL>/departments.json

Request Method: GET

Authentication Parameters needed: YES

Request Body:

```
{
    "role": "user",
    "email": "arjunthedragon@gmail.com",
    "auth_token": "zhZAT_5WaKhT9gLsiJk-",
}
```

Response:

On successful operation, You will receive status code 200 with the list of Departments list as json.

### 4. Show Department Details

This API is meant to list all the departments.

Request URL : <Base URL>/departments/<department\_id>.json

Request Method: GET

Authentication Parameters needed: YES

Request Body:

```
{
    "role":"user",
    "email":"arjunthedragon@gmail.com",
    "auth_token":"zhZAT_5WaKhT9gLsiJk-",
}
```

Response:

On successful operation, You will receive status code 200 with the department entity as json.

## 5. List all Tags

This API is meant to list all the departments.

Request URL : <Base URL>/tags.json

Request Method: GET

Authentication Parameters needed: YES

Request Body:

```
{
    "role":"user",
    "email":"arjunthedragon@gmail.com",
    "auth_token":"zhZAT_5WaKhT9gLsiJk-",
}
```

Response:

On successful operation, You will receive status code 200 with the list of tag list as json.

## 6. List tag details

This API is meant to list the details of a particular tag.

Request URL : <Base URL>/departments.json

Request Method: GET

Authentication Parameters needed: YES

Request Body:

```
{  
    "role":"user",  
    "email":"arjunthedragon@gmail.com",  
    "auth_token":"zhZAT_5WaKhT9gLsiJk-",  
}
```

Response:

On successful operation, You will receive status code 200 with the list of tags list as json.

#### 7. List all Post categories

This API is meant to list all the departments.

Request URL : <Base URL>/post\_categories.json

Request Method: GET

Authentication Parameters needed: YES

Request Body:

```
{  
    "role":"user",  
    "email":"arjunthedragon@gmail.com",  
    "auth_token":"zhZAT_5WaKhT9gLsiJk-",  
}
```

Response:

On successful operation, You will receive status code 200 with the list of Departments list as json.

#### 8. Show Post Category Details

This API is meant to list all the departments.

Request URL : <Base URL>/post\_category/<post\_category\_id>.json

Request Method: GET

Authentication Parameters needed: YES

Request Body:

```
{  
    "role":"user",  
    "email":"arjunthedragon@gmail.com",  
    "auth_token":"zhZAT_5WaKhT9gLsiJk-",  
}
```

Response:

On successful operation, You will receive status code 200 with the details of the particular post id.

Analytics API

1. Visit Post