BLOG API

Post

Retrieve A Single Post

`GET  /post/get`

Retrieves the post with the given postID.

Query Parameters

Name	Description	Type
postID *required*	The ID of the post.	Integer

Responses

HTTP Code	Description	Response
200	The operation completed successfully.	The requested post.
404	The post with the given postID does not exist.	The error message.
500	An unexpected error has occurred.	The error message.

Retrieve Multiple Posts

```
GET  /post/gets
```

Retrieves an array of posts that satisfies the query parameters.

Query Parameters

Name	Description	Type
postIDStart *required*	The ID of the first post to retrieve.	Integer
count *required*	The number of posts to retrieve.	Integer
reverse *required*	Whether to retrieve the posts incrementally or decrementally.	Boolean

Responses

HTTP Code	Description	Response
200	The operation completed successfully.	The requested posts.
500	An unexpected error has occurred.	The error message.

Create A Post

```
POST  /post/create
```

Creates a new post.

Header Parameters

Name	Description	Type
Authorization *required*	The Google access token of the user.	String

Query Parameters

Name	Description	Type
jsonData *required*	The structure of the post as a JSON string.	String

Other Parameters

Name	Description	Type
file *optional*	The thumbnail image of the post. A default image is used when this parameter is not provided.	MultipartFile

Responses

HTTP Code	Description	Response
201	The operation completed successfully.	The created post.
400	The server received a bad request.	The error message.
401	The verification of the Google access token failed.	The error message.
403	The user making the post does not have the necessary role to create a post; or The user has been deleted.	The error message.
500	An unexpected error has occurred.	The error message.

Delete A Post

`DELETE  /post/delete`

Deletes a post.

Header Parameters

Name	Description	Type
Authorization *required*	The Google access token of the user.	String

Query Parameters

Name	Description	Type
postID *required*	The ID of the post to delete.	Integer

Responses

HTTP Code	Description	Response
204	The operation completed successfully.	No content.
400	The server received a bad request.	The error message.
401	The verification of the Google access token failed.	The error message.
403	The user deleting the post is not the author of the post; and The user is not an admin; or The user has been deleted.	The error message.
500	An unexpected error has occurred.	The error message.

Edit A Post

`PUT  /post/edit`

Edits a post.

Header Parameters

Name	Description	Type
Authorization *required*	The Google access token of the user.	String

Other Parameters

Name	Description	Type
body *required*	The new structure of the post as a JSON string.	String

Responses

HTTP Code	Description	Response
204	The operation completed successfully.	No content.
400	The server received a bad request.	The error message.
401	The verification of the Google access token failed.	The error message.
403	The user editing the post is not the author of the post; or The user has been deleted.	The error message.
500	An unexpected error has occurred.	The error message.

Upvote A Post

```
PUT  /post/upvote
```

Upvotes a post.
Any given user may only upvote once.
If the user has previously downvoted the post, then the downvote will be changed to an upvote.

Header Parameters

Name	Description	Type
Authorization *required*	The Google access token of the user.	String

Query Parameters

Name	Description	Type
postID *required*	The ID of the post.	Integer

Responses

HTTP Code	Description	Response
204	The operation completed successfully.	No content.
400	The user already upvoted the given post; or The server received a bad request.	The error message.
401	The verification of the Google access token failed.	The error message.
403	The user has been deleted.	The error message.
500	An unexpected error has occurred.	The error message.

Downvote A Post

```
PUT  /post/downvote
```

Downvotes a post.
Any given user may only downvote once.
If the user has previously upvoted the post, then the upvote will be changed to a downvote.

Header Parameters

Name	Description	Type
Authorization *required*	The Google access token of the user.	String

Query Parameters

Name	Description	Type
postID *required*	The ID of the post.	Integer

Responses

HTTP Code	Description	Response
204	The operation completed successfully.	No content.
400	The user already downvoted the given post; or The server received a bad request.	The error message.
401	The verification of the Google access token failed.	The error message.
403	The user has been deleted.	The error message.
500	An unexpected error has occurred.	The error message.

View A Post

```
PUT  /post/view
```

Increments the view counter of a post.

Query Parameters

Name	Description	Type
postID *required*	The ID of the post.	Integer

Responses

HTTP Code	Description	Response
204	The operation completed successfully.	No content.
400	The post does not exist; or The server received a bad request.	The error message.
500	An unexpected error has occurred.	The error message.

Search For Posts

```
GET  /post/search
```

Retrieves an array of posts that satisfies the search query.
A post is a match if the search pattern is a substring of the title or content of a post.
Search results can be sorted by newest, oldest, top-rated, or most-viewed.

Query Parameters

Name	Description	Type
pattern *optional*	The pattern to match for in the title and content of a post. Default Value: An empty string.	String
start *optional*	The index of the first result to retrieve. Default Value: 1	Integer
count *optional*	The number of posts to retrieve. Default Value: 100	Integer
sortBy *optional*	The sorting method for the search results: - new: newest post first - old: oldest post first - top: top-rated post first (upvotes - downvotes) - view: most viewed post first Default Value: new	String

Responses

HTTP Code	Description	Response
200	The operation completed successfully.	The requested posts.
400	An unsupported sorting method was passed into sortBy.	The error message.
500	An unexpected error has occurred.	The error message.

Retrieve Posts Of A User

```
GET  /post/byUser
```

Retrieves the most recent posts of a given user.

Query Parameters

Name	Description	Type
userID *required*	The ID of the user to retrieve posts from.	String
count *optional*	The number of posts to retrieve. Default Value: 10	Integer

Responses

HTTP Code	Description	Response
200	The operation completed successfully.	The requested posts.
500	An unexpected error has occurred.	The error message.

Get The Largest Post ID

`GET  /post/highest`

DEPRECATED
Retrieves the largest postID that has been assigned thus far.

Responses

HTTP Code	Description	Response
200	The operation completed successfully.	The largest post ID.
500	An unexpected error has occurred.	The error message.

Comment

Retrieve A Single Comment

```
GET  /comment/get
```

Retrieves the comment with the given postID and commentID.

Query Parameters

Name	Description	Type
postID *required*	The ID of the post containing the comment.	Integer
commentID *required*	The ID of the comment.	Integer

Responses

HTTP Code	Description	Response
200	The operation completed successfully.	The requested comment.
404	The comment with the given postID and commentID does not exist.	The error message.
500	An unexpected error has occurred.	The error message.

Retrieve Multiple Comments

`GET  /comment/gets`

Retrieves an array of comments that satisfies the query parameters.

Query Parameters

Name	Description	Type
postID *required*	The ID of the post containing the comments.	Integer
commentIDStart *required*	The ID of the first comment to retrieve.	Integer
count *required*	The number of comments to retrieve.	Integer
reverse *required*	Whether to retrieve the comments incrementally or decrementally.	Boolean

Responses

HTTP Code	Description	Response
200	The operation completed successfully.	The requested comments.
500	An unexpected error has occurred.	The error message.

Create A Comment

```
POST  /comment/create
```

Creates a new comment.

Header Parameters

Name	Description	Type
Authorization *required*	The Google access token of the user.	String

Query Parameters

Name	Description	Type
body *required*	The structure of the comment as a JSON string.	String

Responses

HTTP Code	Description	Response
201	The operation completed successfully.	The created comment.
400	The server received a bad request.	The error message.
401	The verification of the Google access token failed.	The error message.
403	The user making the comment does not have the necessary role to create a comment; or The user has been deleted.	The error message.
500	An unexpected error has occurred.	The error message.

Delete A Comment

`DELETE  /comment/delete`

Deletes a comment.

Header Parameters

Name	Description	Type
Authorization *required*	The Google access token of the user.	String

Query Parameters

Name	Description	Type
postID *required*	The ID of the post containing the comment to delete.	Integer
commentID *required*	The ID of the comment to delete.	Integer

Responses

HTTP Code	Description	Response
204	The operation completed successfully.	No content.
400	The server received a bad request.	The error message.
401	The verification of the Google access token failed.	The error message.
403	The user deleting the comment is not the author of the comment; and The user is not an admin; or The user has been deleted.	The error message.
500	An unexpected error has occurred.	The error message.

Edit A Comment

```
PUT  /comment/edit
```

Edits a comment.

Header Parameters

Name	Description	Type
Authorization *required*	The Google access token of the user.	String

Other Parameters

Name	Description	Type
body *required*	The new structure of the comment as a JSON string.	String

Responses

HTTP Code	Description	Response
204	The operation completed successfully.	No content.
400	The server received a bad request.	The error message.
401	The verification of the Google access token failed.	The error message.
403	The user editing the comment is not the author of the comment ; or The user has been deleted.	The error message.
500	An unexpected error has occurred.	The error message.

Upvote A Comment

```
PUT  /comment/upvote
```

Upvotes a comment.
Any given user may only upvote once.
If the user has previously downvoted the comment, then the downvote will be changed to an upvote.

Header Parameters

Name	Description	Type
Authorization *required*	The Google access token of the user.	String

Query Parameters

Name	Description	Type
postID *required*	The ID of the post containing the comment.	Integer
commentID *required*	The ID of the comment.	Integer

Responses

HTTP Code	Description	Response
204	The operation completed successfully.	No content.
400	The user already upvoted the given comment; or The server received a bad request.	The error message.
401	The verification of the Google access token failed.	The error message.
403	The user has been deleted.	The error message.
500	An unexpected error has occurred.	The error message.

Downvote A Comment

```
PUT  /comment/downvote
```

Downvotes a comment.
Any given user may only downvote once.
If the user has previously upvoted the comment, then the upvote will be changed to a downvote.

Header Parameters

Name	Description	Type
Authorization *required*	The Google access token of the user.	String

Query Parameters

Name	Description	Type
postID *required*	The ID of the post containing the comment.	Integer
commentID *required*	The ID of the comment.	Integer

Responses

HTTP Code	Description	Response
204	The operation completed successfully.	No content.
400	The user already downvoted the given comment; or The server received a bad request.	The error message.
401	The verification of the Google access token failed.	The error message.
403	The user has been deleted.	The error message.
500	An unexpected error has occurred.	The error message.

User

Retrieve A Single User

```
GET  /user/get
```

Retrieves the user with the given Google access token or userID.
When both the Google access token and userID is given, the userID takes precedence.
Either the Google access token or the userID must be provided.

Header Parameters

Name	Description	Type
Authorization *optional*	The Google access token of the user to retrieve.	String

Query Parameters

Name	Description	Type
userID *optional*	The ID of the user to retrieve.	String

Responses

HTTP Code	Description	Response
200	The operation completed successfully.	The requested user.
400	The server received a bad request.	The error message.
401	The verification of the Google access token failed.	The error message.
403	The user with the given userID has been deleted.	The error message.
404	The user with the given userID does not exist.	The error message.
500	An unexpected error has occurred.	The error message.

Delete A User

`DELETE  /user/delete`

Deletes a user.

Header Parameters

Name	Description	Type
Authorization *required*	The Google access token of the user performing the deletion.	String

Query Parameters

Name	Description	Type
userID *required*	The ID of the user to delete.	String

Responses

HTTP Code	Description	Response
204	The operation completed successfully.	No content.
400	The server received a bad request.	The error message.
401	The verification of the Google access token failed.	The error message.
403	The user deleting is not the same user as the one being deleted; and The user is not an admin; or Either user has been deleted.	The error message.
500	An unexpected error has occurred.	The error message.

Edit A User

`PUT  /user/edit`

Edits a user.

Header Parameters

Name	Description	Type
Authorization *required*	The Google access token of the user performing the edit.	String

Other Parameters

Name	Description	Type
body *required*	The new structure of the user as a JSON string.	String

Responses

HTTP Code	Description	Response
204	The operation completed successfully.	No content.
400	The server received a bad request.	The error message.
401	The verification of the Google access token failed.	The error message.
403	The user editing is not the same user as the one being edited; or The user has been deleted.	The error message.
500	An unexpected error has occurred.	The error message.

Request A Promotion

```
PUT  /user/requestPromotion
```

Creates a request for a promotion or updates an existing promotion request.

Header Parameters

Name	Description	Type
Authorization *required*	The Google access token of the user.	String

Other Parameters

Name	Description	Type
body *required*	The structure of the promotion request as a JSON string.	String

Responses

HTTP Code	Description	Response
201	The operation completed successfully.	The created/updated promotion request.
400	The user already has a user level equal to or higher than the requested user level; or The server received a bad request.	The error message.
401	The verification of the Google access token failed.	The error message.
403	The user has been deleted.	The error message.
500	An unexpected error has occurred.	The error message.

Admin

Retrieve Multiple Promotion Requests

```
GET  /admin/gets
```

Retrieves an array of promotion requests that satisfies the query parameters.

Header Parameters

Name	Description	Type
Authorization *required*	The Google access token of the user.	String

Query Parameters

Name	Description	Type
requestIDStart *required*	The ID of the first promotion request to retrieve.	Integer
count *required*	The number of promotion requests to retrieve.	Integer
reverse *required*	Whether to retrieve the promotion requests incrementally or decrementally.	Boolean

Responses

HTTP Code	Description	Response
200	The operation completed successfully.	The requested promotion requests.
401	The verification of the Google access token failed.	The error message.
403	The user is not an admin; or The user has been deleted	The error message.
500	An unexpected error has occurred.	The error message.

Perform A Promotion

`PUT  /admin/promote`

Promotes a user to a given user level.
The same endpoint can be used for demotions.
Will also remove all promotion requests by the promoted user that is equal to or lower than the new user level of the promoted user.
This effectively approves a promotion request.

Header Parameters

Name	Description	Type
Authorization *required*	The Google access token of the user performing the promotion.	String

Other Parameters

Name	Description	Type
body *required*	The structure of the promotion as a JSON string. Contains the userID of the user to promote and the user level to promote them to.	String

Responses

HTTP Code	Description	Response
200	The operation completed successfully.	No Content.
400	The server received a bad request.	The error message.
401	The verification of the Google access token failed.	The error message.
403	The user is not an admin; or Either user has been deleted.	The error message.
500	An unexpected error has occurred.	The error message.

Delete A Promotion Request

`DELETE  /admin/delete`

Deletes a promotion request.
This effectively denys a promotion request.

Header Parameters

Name	Description	Type
Authorization *required*	The Google access token of the user performing the deletion.	String

Query Parameters

Name	Description	Type
requestID *required*	The ID of the promotion request to delete.	Integer

Responses

HTTP Code	Description	Response
204	The operation completed successfully.	No content.
400	The server received a bad request.	The error message.
401	The verification of the Google access token failed.	The error message.
403	The user is not an admin; or The user has been deleted.	The error message.
500	An unexpected error has occurred.	The error message.