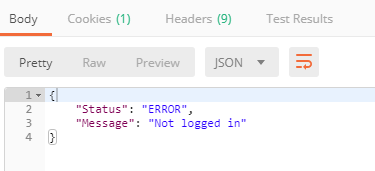
* /not\_logged\_in

**Method**: GET

**Description**: Default when not logged in

**Return**: Error response

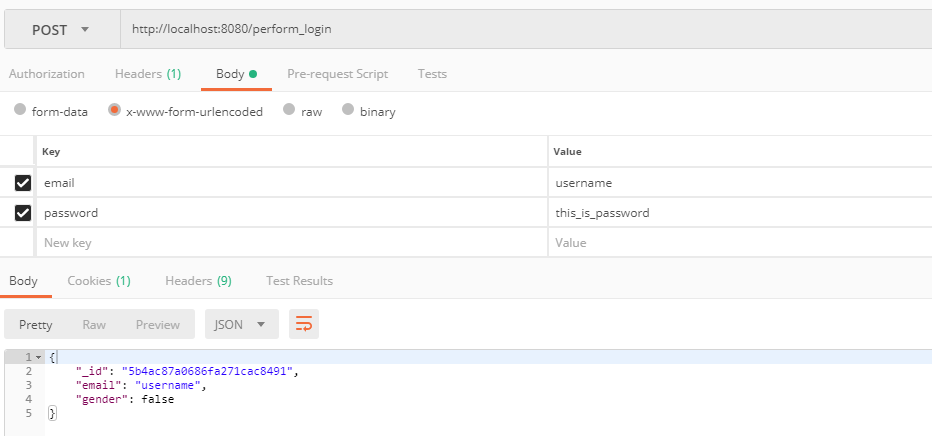


* /perform\_login

**Method**: POST

**Description**: Used to login

**Return**: redirect to /index



* /logout

**Method**: POST

**Description**: Used to logout

**Return**: redirect to /not\_logged\_in

* /account\_info (updated to account\_info instead user\_info)

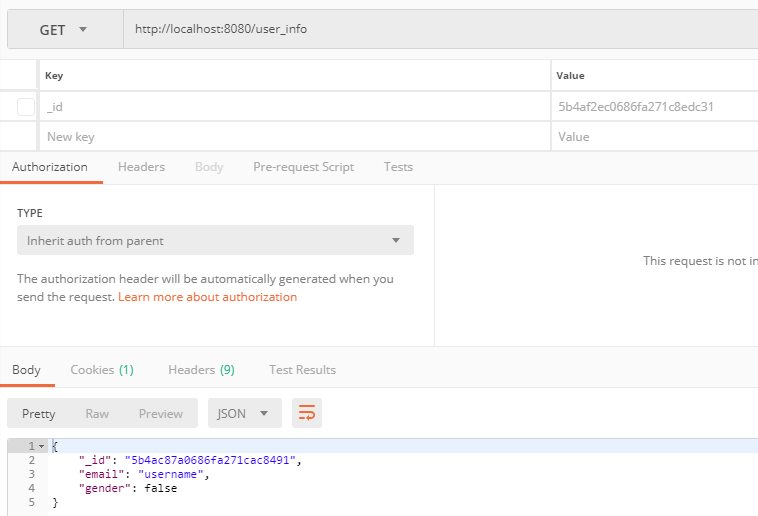
**Method**: GET

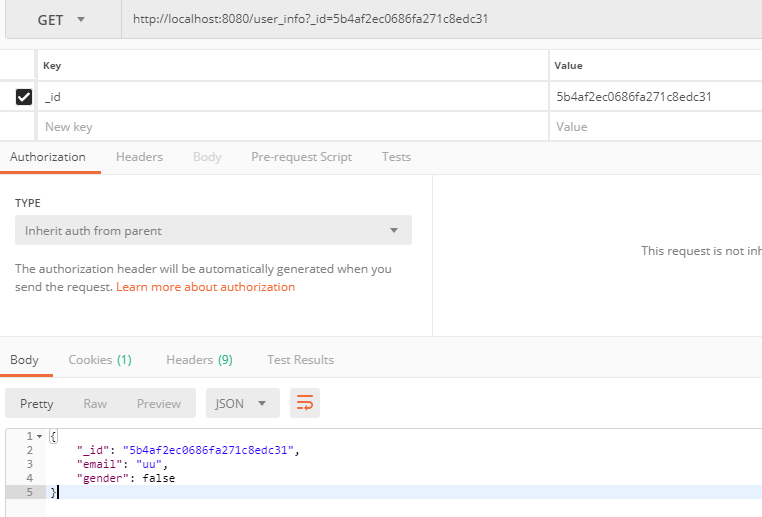
**Description**: get user’s information

**RequestParams**:

\_id (not required): user id

**Return**: return an AccountViewModel as result





* /changepass

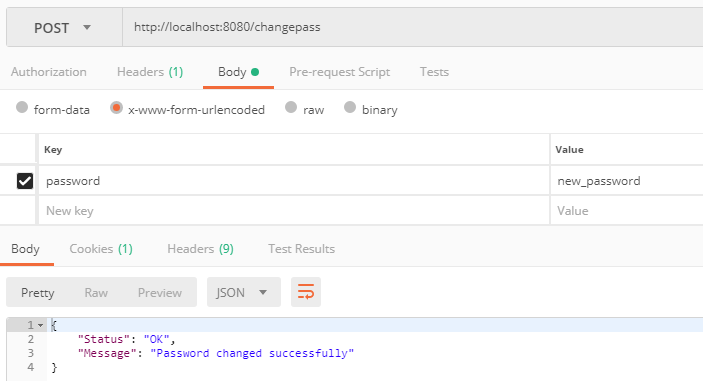
**Method**: POST

**Description**: Used to change password

**RequestParams**:

password: the new password

**Return**: response object



* /register (TEMPORARY)

**Method**: POST

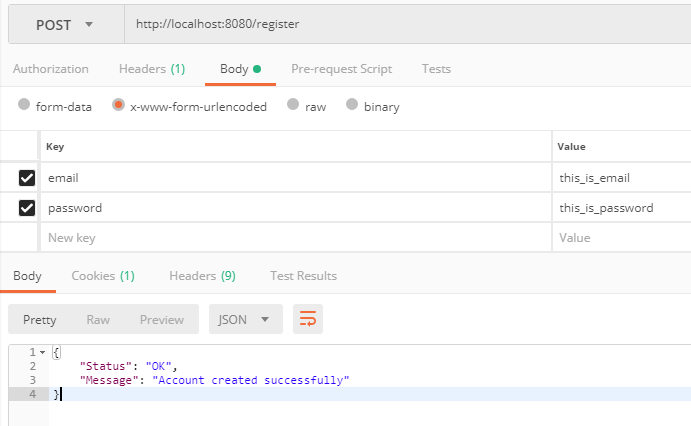
**Description**: Used to register new account

**RequestParams**:

Email: email, act as username

Password: the password

**Return**: response object



* /list/comments/

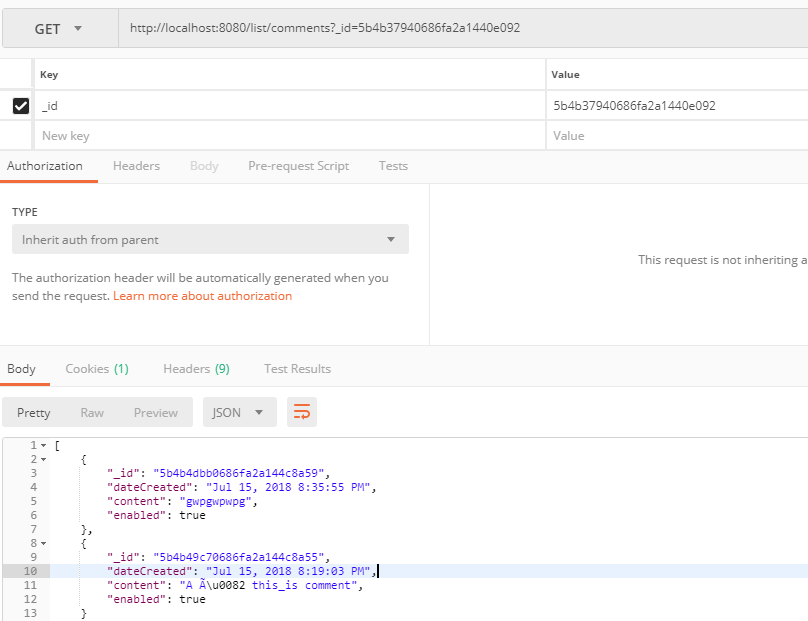
**Method**: GET

**Description**: Get comments of a post

**RequestParams**:

\_id: post id

**Return**: Array of CommentViewModel



* /comment

**Method**: POST

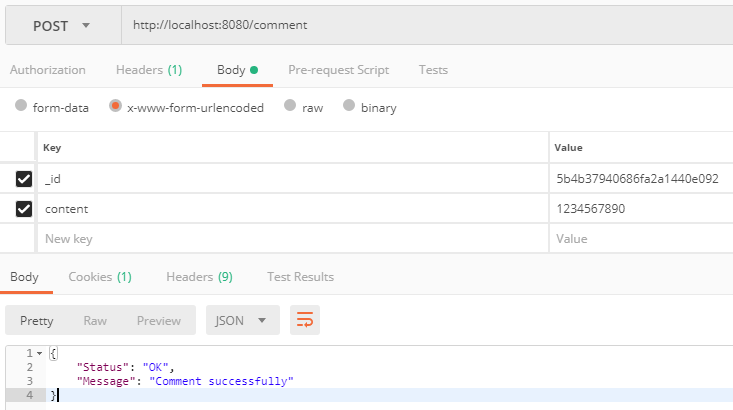
**Description**: Comment

**RequestParams**:

\_id: post id

Content: content of comment

**Return**: response object



* /upload

**Method**: POST

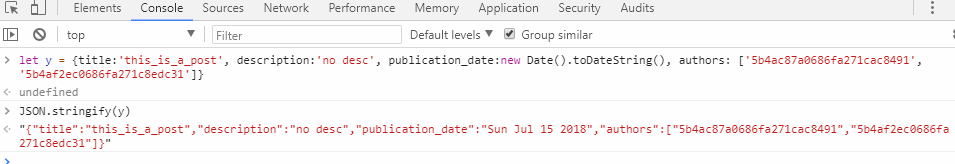
**Description**: Used to post a post

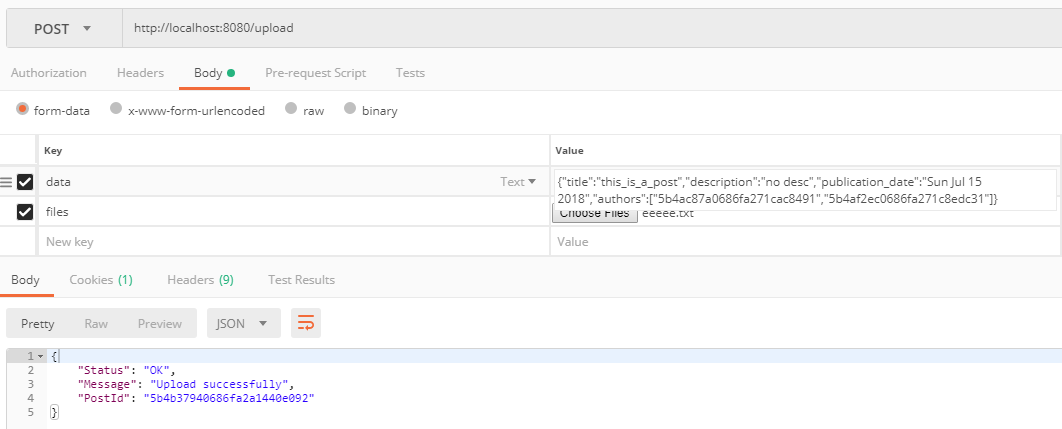
**RequestParams**:

Data: JSON string like bellow

Files: enctype multipart/form-data file

**Return**: response object with PostId





* /download/{post\_id}

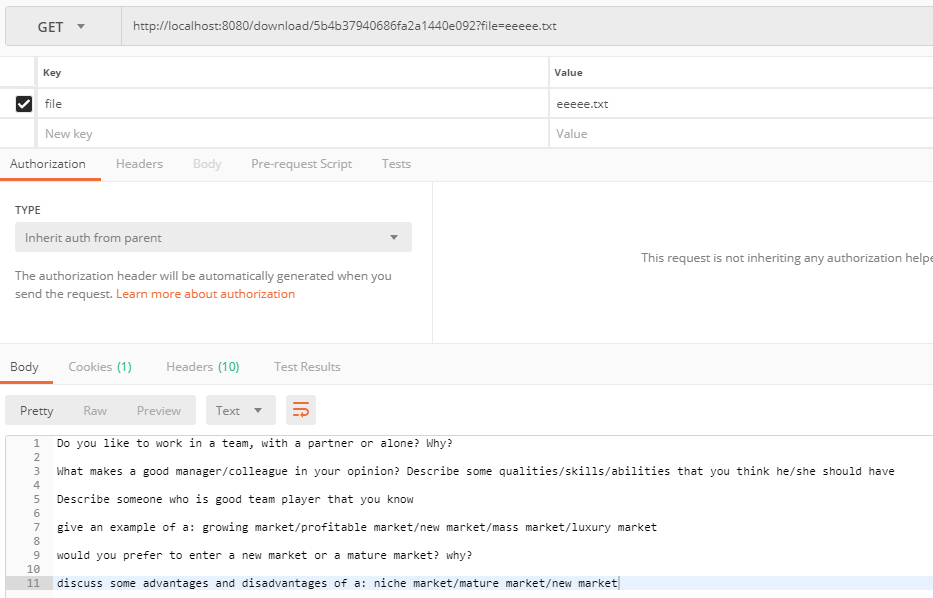
**Method**: GET

**Description**: Used to download

**RequestParams**:

File: file name

**Return**: File data as application/octet-stream to download

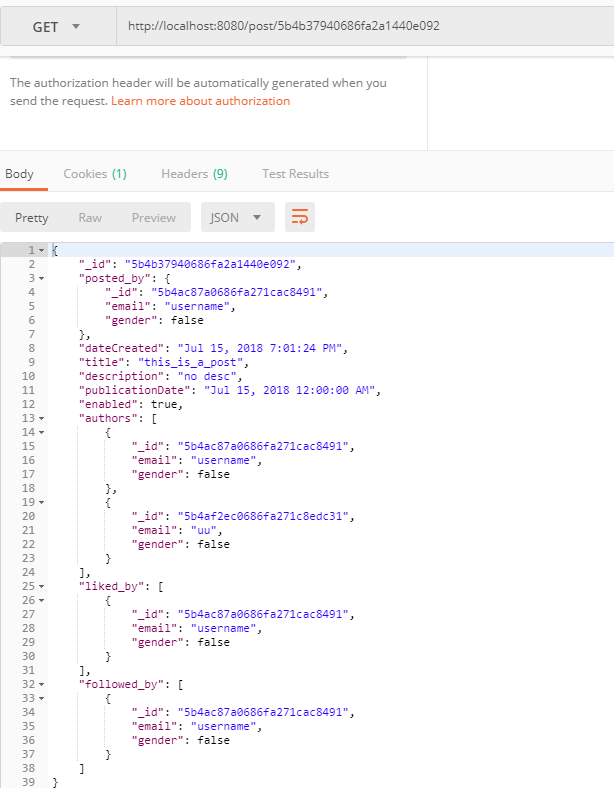


* /post/{post\_id}

**Method**: GET

**Description**: get post information

**Return**: return PostViewModel



* /post/react

**Method**: POST

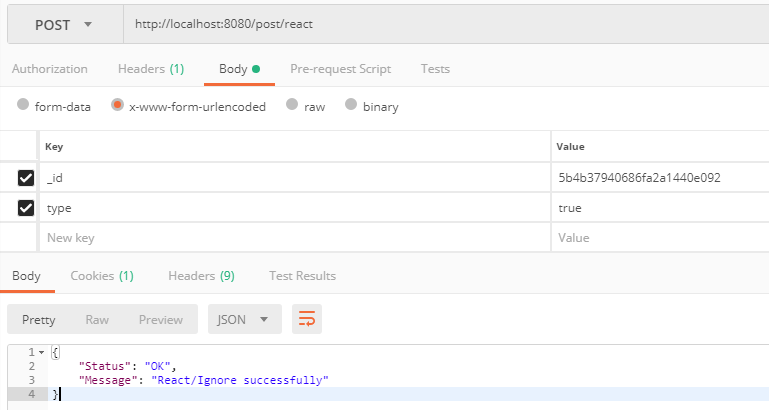
**Description**: React a post

**RequestParams**:

\_id: Post id

Type: type of state of reaction, false->unlike, true->like

**Return**: response object



* /post/follow

**Method**: POST

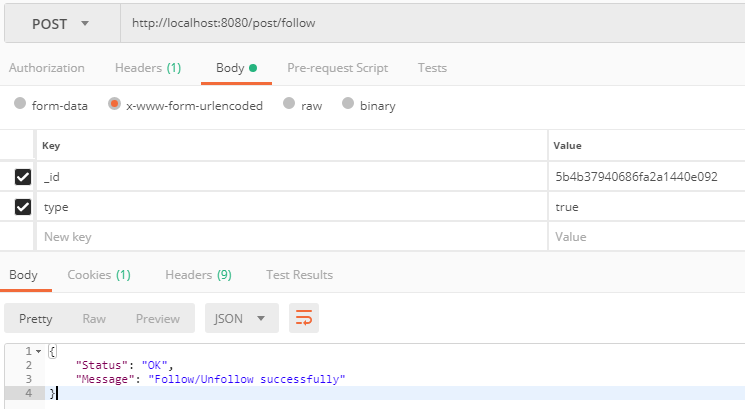
**Description**:

**RequestParams**:

\_id: Post id

Type: type of state of follow, false->unfollow, true->follow

**Return**: response object



* /user/friend\_list

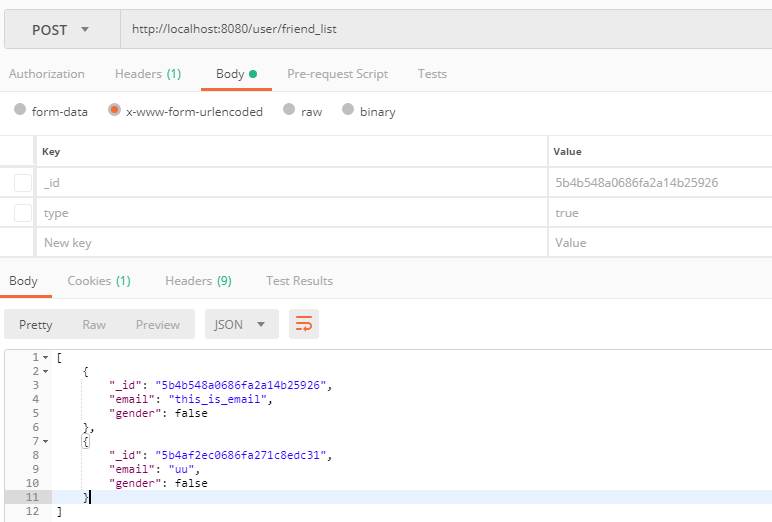
**Method**: POST

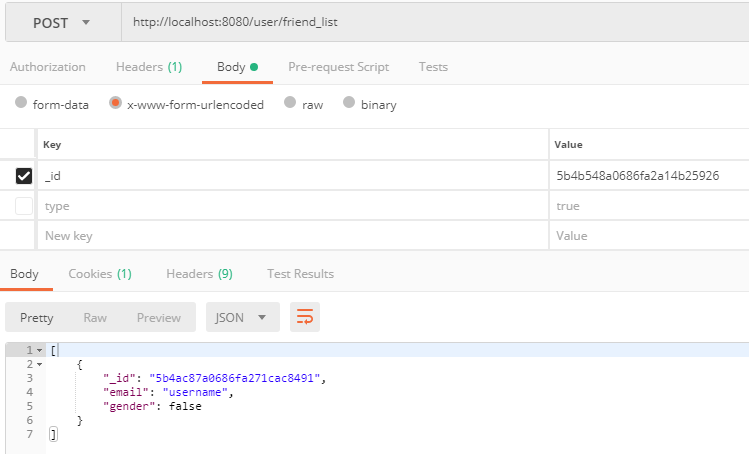
**Description**: Get friend list of user

**RequestParams**:

\_id (not required): User id

**Return**: List of AccountViewModel





* /user/friend\_request\_list

**Method**: POST

**Description**: Get list of user that received friend request from current user

**RequestParams**:

\_id (not required): User id

**Return**: List of user that received friend request from current user

**Return format**: same with /user/friend\_list

* /user/friend\_requested\_by\_list

**Method**: POST

**Description**: Get list of user that sent friend request to current user

**RequestParams**:

\_id (not required): User id

**Return**: List of user that sent friend request to current user

**Return format**: same with /user/friend\_list

* /user/is\_friend

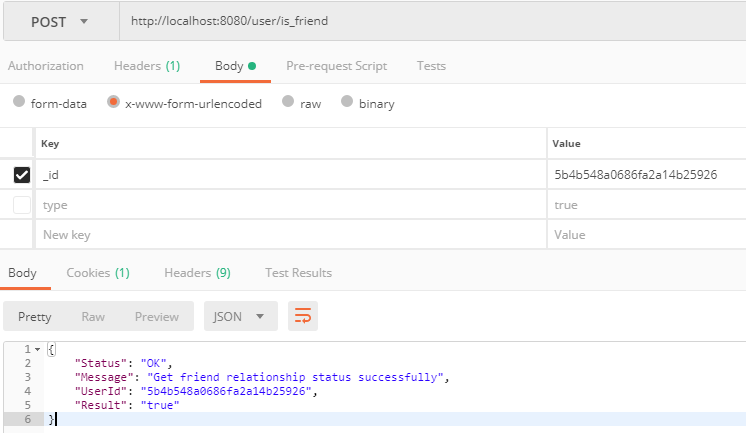
**Method**: POST

**Description**: Check if user is friend of current user

**RequestParams**:

\_id (not required): User id

**Return**: response object with addition field named “Result”



* /user/is\_friend\_request\_sent

**Method**: POST

**Description**: Check if user has received friend request sent by current user

**RequestParams**:

\_id (not required): User id

**Return**: response object with addition field named “Result”

**Return format**: same with /user/is\_friend

* /user/is\_friend\_request\_received

**Method**: POST

**Description**: Check if user has sent friend request to current user

**RequestParams**:

\_id (not required): User id

**Return**: response object with addition field named “Result”

**Return format**: same with /user/is\_friend

* /post/is\_followed

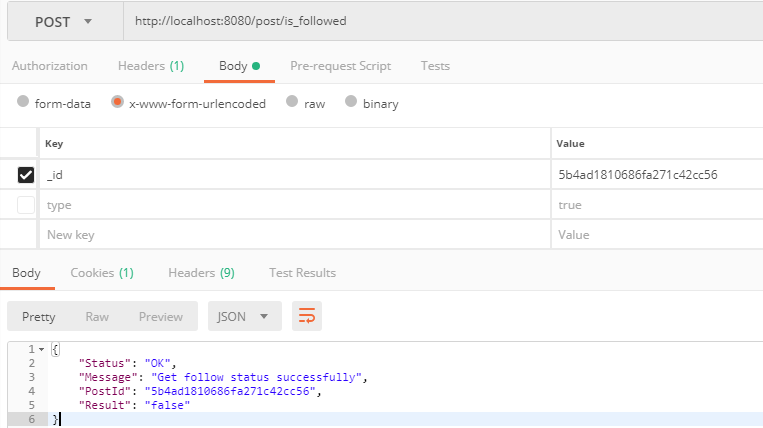
**Method**: POST,

**Description**: Check if user has followed post

**RequestParams**:

\_id : Post id

**Return**: response object with addition field named “Result”



* /post/is\_reacted

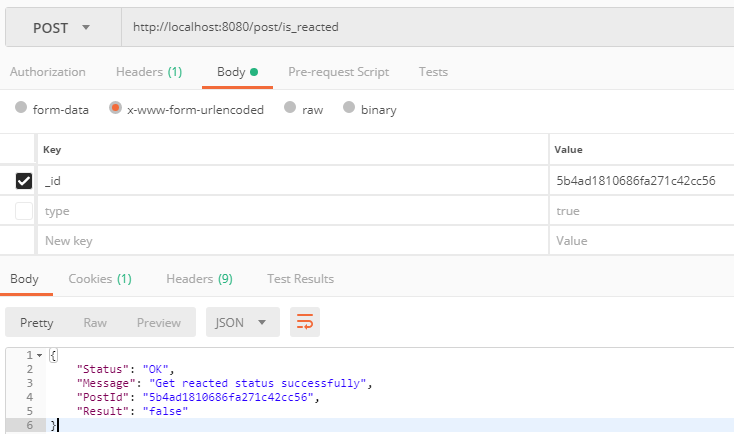
**Method**: POST,

**Description**: Check if user has reacted post

**RequestParams**:

\_id : Post id

**Return**: response object with addition field named “Result”



* /user/friend\_request\_process

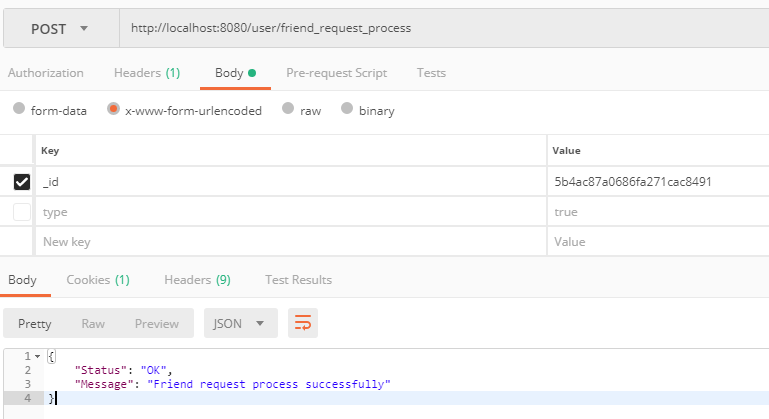
**Method**: POST

**Description**: Process friend request, included send friend request, confirm friend request if received, cancel if existed

**RequestParams**:

\_id : User id

**Return**: response object



* /search/documents

**Method**: GET

**Description**: Search documents

**RequestParams**:

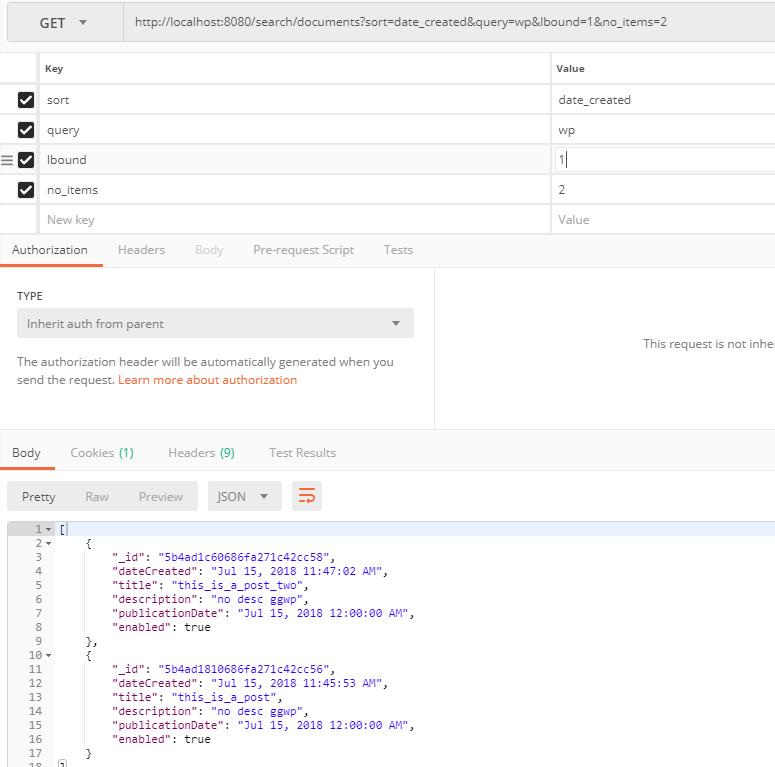
Sort (not required): 1 of 3 values: “date\_created”, “publication\_date”, “most\_followed”. If not specified, the default is “date\_created”, sort order is descending

Query: string to query

Lbound: lower bound to get item

No\_items: size of result array

**Return**: Array of PostViewModel



* /search/users

**Method**: GET

**Description**: Used to search user by name, email

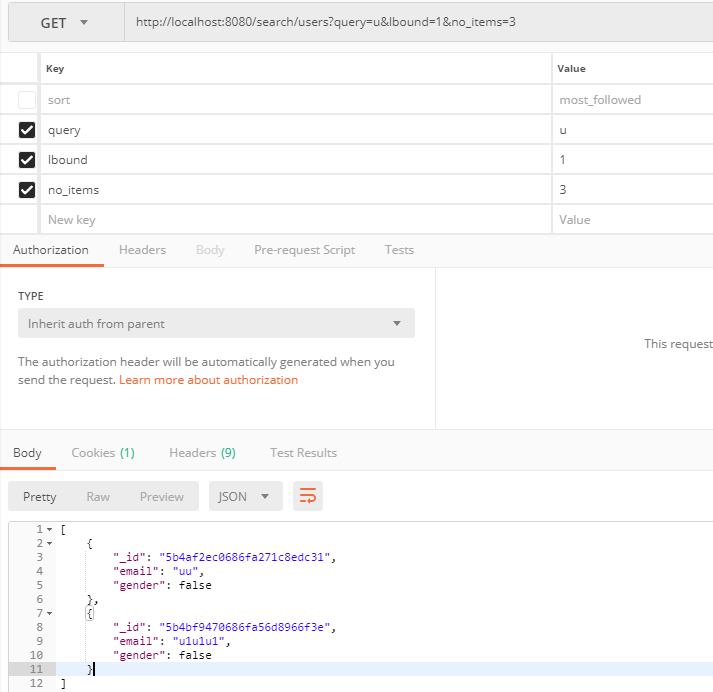
**RequestParams**:

Query: string to query

Lbound: lower bound to get item

No\_items: size of result array

**Return**: Array of AccountViewModel



* /user/posted\_posts
* /user/followed\_posts
* /users/author\_of\_posts
* <<placeholder>>