

## LEGEND:

**PURPLE:** Code is over 80 characters.

**BLUE** : Please put comments

### tutorial\_routes.py

```
@app.route('/api/tutorial/<username>/<tutorial_uuid>', methods=['GET'])
def get_one_tutorial(username, tutorial_uuid):
    """
    TELL ME SOMETHING ABOUT THIS FUNCTION AS WELL AS THE PARAMS AND ITS RETURN VAL
    """
    sql_query = "SELECT * FROM diyup.tutorials WHERE author_username=%s AND uuid=%s"
    cur = mysql.connection.cursor()
    cur.execute(sql_query, (username, tutorial_uuid))
    tutorial = cur.fetchone()

    if not tutorial:
        return jsonify({'message' : 'No tutorial found!'}), 400

    sql_query = "SELECT * FROM diyup.steps WHERE tutorial_uuid=%s"
    cur.execute(sql_query, (tutorial[0],))
    steps = cur.fetchall()

    output_steps = []

    tutorial_data = {}
    tutorial_data['uuid'] = tutorial[0]
    tutorial_data['author_username'] = tutorial[1]
    tutorial_data['title'] = tutorial[2]
    tutorial_data['image'] = tutorial[3]
    tutorial_data['category'] = tutorial[4]
    tutorial_data['description'] = tutorial[5]
    tutorial_data['author_difficulty'] = str(tutorial[6])
    tutorial_data['viewer_difficulty'] = str(average_rating_type_for_tutorial('difficulty', tutorial[0]))
    tutorial_data['rating'] = str(average_rating_type_for_tutorial('score', tutorial[0]))

    for step in steps:
        step_data = {}
        step_data['index'] = step[1]
        step_data['content'] = step[2]
        step_data['image'] = step[3]
        output_steps.append(step_data)

    tutorial_data['steps'] = output_steps

    cur.close()

    return jsonify({'tutorial' : tutorial_data}), 200
```

## LEGEND:

**PURPLE:** Code is over 80 characters.

**BLUE** : Please put comments

### user\_routes.py

```
@app.route('/api/login', methods=['POST'])
def login():
    """
    TELL ME SOMETHING ABOUT THIS FUNCTION AND IT'S RETURN VAL
    """
    data = request.get_json()

    username = data['username']
    password = data['password']

    if not username:
        return make_response('Could not verify auth', 401, {'WWW-Authenticate' : 'Basic realm="Login required!"'})

    sql_query = "SELECT * FROM diyup.users WHERE username=%s"
    cur = mysql.connection.cursor()
    cur.execute(sql_query, (username,))
    user = cur.fetchone()
    cur.close()

    if not user:
        return make_response('Could not verify user', 401, {'WWW-Authenticate' : 'Basic realm="Login required!"'})

    if check_password_hash(user[2], password):
        token = jwt.encode({'email_address' : user[0]}, app.config['SECRET_KEY'])
        return jsonify({'token' : token.decode('UTF-8')})

    return make_response('Could not verify', 401, {'WWW-Authenticate' : 'Basic realm="Login required!"'})
```

## LEGEND:

**PURPLE:** Code is over 80 characters.

**BLUE** : Please put comments

### comment\_routes.py

```
@app.route('/api/comments/<tutorial_uuid>/create/<reply_comment_id>', methods=['POST'])
@token_required
def reply_to_tutorial_comment(current_user, tutorial_uuid, reply_comment_id):
    """
    TELL ME SOMETHING ABT THIS FUNCTION AND ITS VARIABLES AS WELL AS ITS RETURN VAL
    """
    data = request.get_json()

    cur = mysql.connection.cursor()

    cur.execute("SELECT * FROM diyup.comments WHERE tutorial_uuid=%s AND reply_to=%s", (tutorial_uuid, reply_comment_id,))
    uuid = cur.fetchall()

    if not uuid:
        return jsonify({'message' : 'No tutorial ID found!'}), 400

    cur.execute("SELECT * FROM diyup.comments ORDER BY id DESC LIMIT 1")
    index = cur.fetchone()
    id = int(index[0]) + 1

    content = data['content']
    image = data['image']
    reply_to = reply_comment_id
    edited = 0
    date = time.ctime(1574039538)
    timestamp = date

    cur.execute("INSERT INTO diyup.comments(comments.tutorial_uuid, username, content, created, timestamp, edited, image, reply_to) VALUES(%s, %s, %s, %s, %s, %s, %s, %s)",
                (tutorial_uuid, current_user[1], content, date, timestamp, edited, image, reply_to,))

    mysql.connection.commit()
    cur.close()

    return jsonify({'message' : 'Reply created!', 'comment id' : id}), 201
```