

## Assignment 6.3 – API Gateway Part IV

### Instructions

1. Open the user.js file and add a new query for finding individual users by email address
  - a. user.js

```
module.exports.getOne = (e, callback) => {  
  var query = {email: e};  
  User.findOne(query, callback);  
};
```

2. Add a new function to the authController to handle user login requests
  - a. authController.js

```
exports.user_login = function(req, res) {  
  User.getOne(req.body.email, function(err, user) {  
    if (err) return res.status(500).send('Error on server.');
```

3. Update the api-catalog.js routes to allow user login requests
  - a. /routes/api-catalog.js
    - i. router.post('auth/login', auth\_controller.user\_login);

```
router.post('/auth/login', auth_controller.user_login);
```



4. Add a new function to the authController to handle user logout requests
  - a. /controllers/authController.js

```
exports.user_logout = function(req, res) {  
  res.status(200).send({ auth: false, token: null});  
};
```

5. Update the api-catalog.js routes to allow user logout requests
  - a. /routes/api-catalog.js
    - i. router.get('auth/logout', auth\_controller.user\_logout);

```
router.get('/auth/logout', auth_controller.user_logout);
```

6. Build, run, and verify there are no errors

- |   | Method | Endpoint              | Resource        |
|---|--------|-----------------------|-----------------|
|    | POST   | http://localhost:3000 | /api/auth/login |

- Media Type  ☐ Post QueryString
- grant\_type=password&username=test&password=test&email=test@hotmail.com

- ```
1 {  
2   "auth": true,  
3   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCJ6IjVlYmF0eS5kaW8udnNpdjEwLmlkZWQucmVzcw%3D"  
4 }
```

- ```
1 {  
2   "auth": false,  
3   "token": null  
4 }
```

- Note: The logout API has zero parameters and simply returns a 200-status code. This is far from production ready and is only intended to illustrate the logout process.

