시험

전체적인 흐름을 이해,

출력결과가 뭔지, 코드가 무엇을 의미하는지 알고있으면 됨.

NodeJS 응용 III

USER AUTHENTICATION WITH EXPRESS SESSIONS

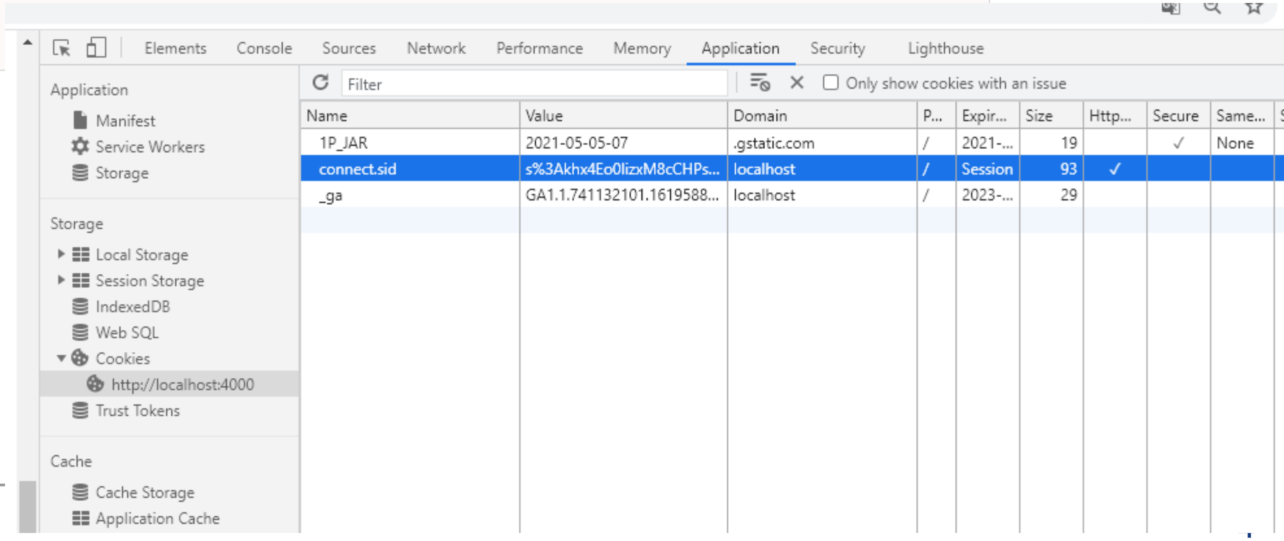
- **Sessions** are how we keep the user logged into our web app by keeping their information in the browser
  - The information kept on the user's browser is called **Cookies**

- **To implement Express sessions:**

```
> npm install express-session
```

- **In index.js :**
  - **secret** string is used by the express-session package **to sign and encrypt the session ID** cookie being shared the browser.

```
…
const expressSession = require('express-session');
…


 …
 app.use(expressSession({
    secret: 'keyboard cat'
 }))
 …
```

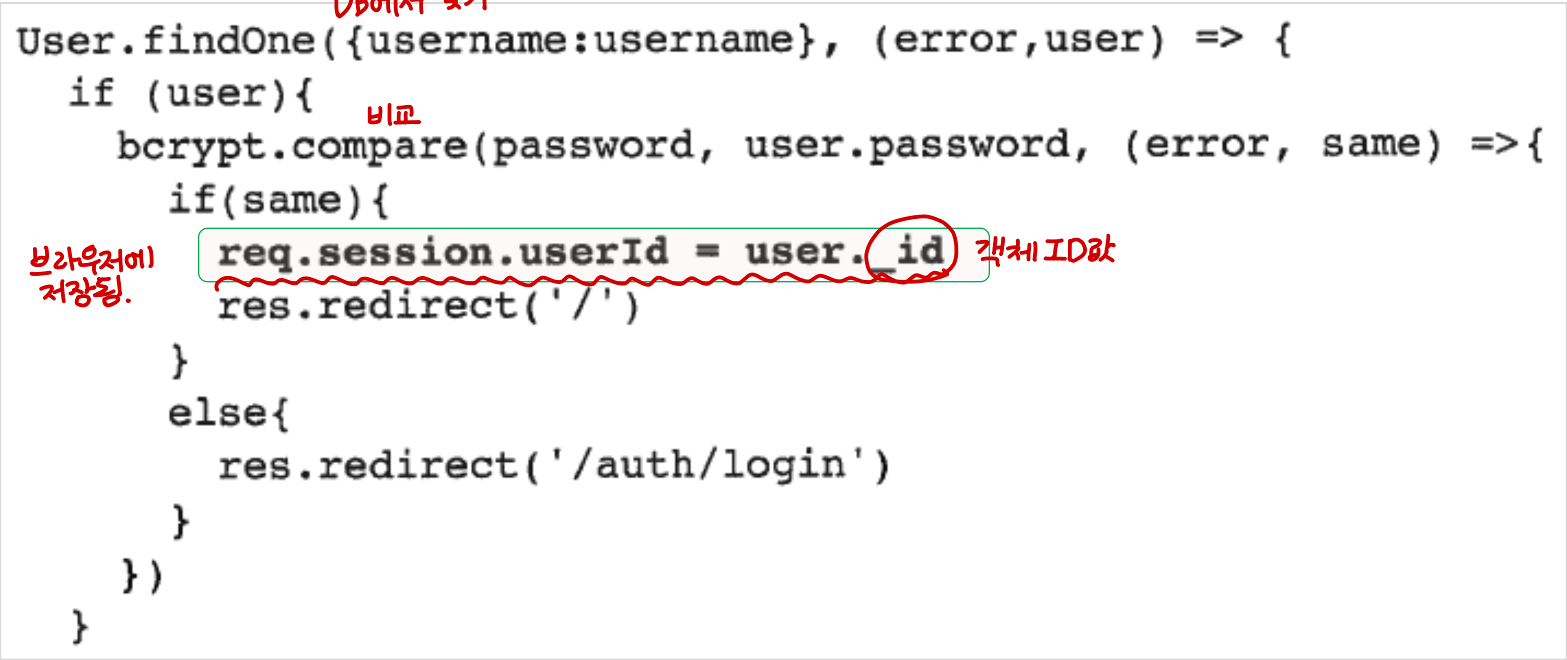


Name	Value	Domain	P...	Expir...	Size	Http...	Secure	Same...
1P_JAR	2021-05-05-07	.gstatic.com	/	2021-...	19		✓	None
connect.sid	s%3Akhx4Eo0lizxM8cCHPs...	localhost	/	Session	93	✓		
_ga	GA1.1.741132101.1619588...	localhost	/	2023-...	29			

Implementing User Sessions

- **In loginUser.js**
  - We assign the user _id to the session.
  - The session package saves this data on the user's browser

DB에서 찾기



```
User.findOne({username:username}, (error,user) => {
  if (user){
    bcrypt.compare(password, user.password, (error, same) =>{
      if(same){
        req.session.userId = user._id
        res.redirect('/')
      }
      else{
        res.redirect('/auth/login')
      }
    })
  }
```

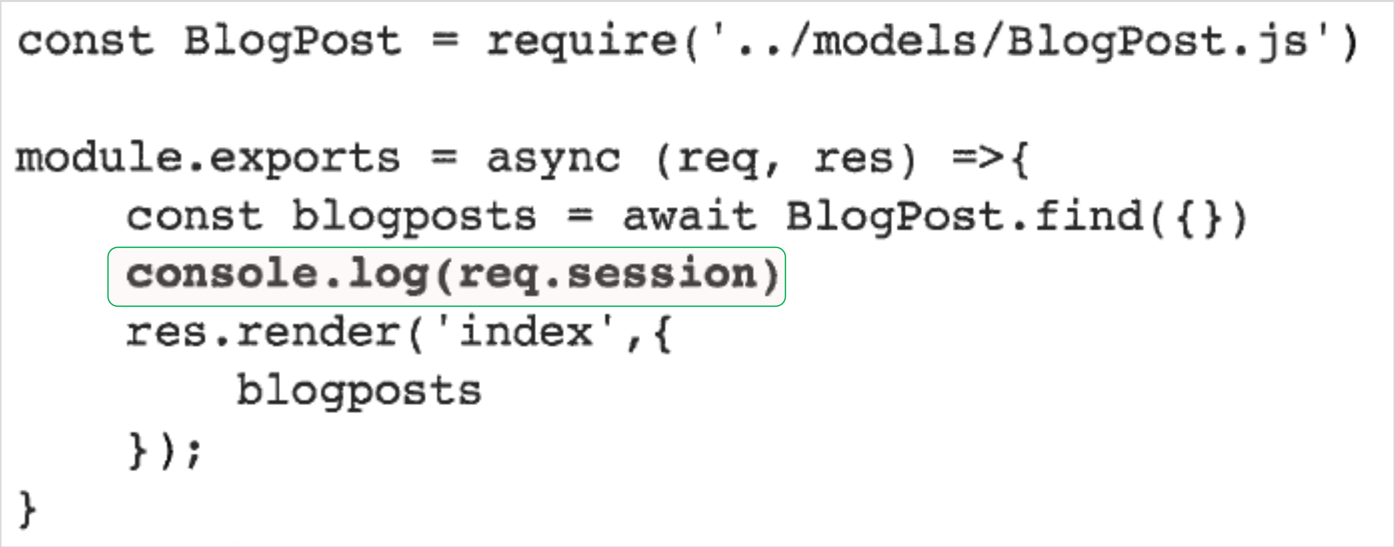
비교

브라우저에 저장됨.

객체 ID값

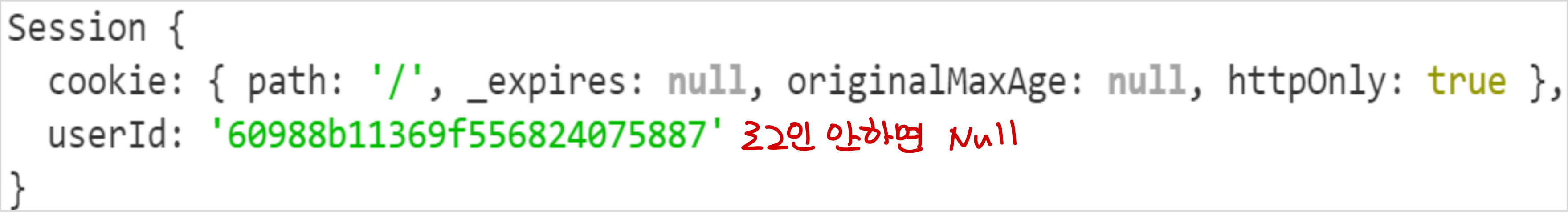
In home.js :

- To see exactly what is in a session object



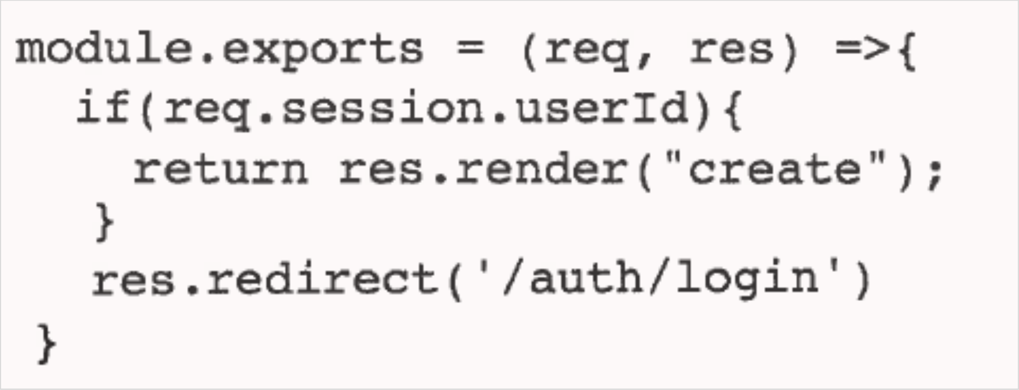
```
const BlogPost = require('../models/BlogPost.js')

module.exports = async (req, res) =>{
    const blogposts = await BlogPost.find({})
    console.log(req.session)
    res.render('index',{
        blogposts
    });
}
```

- When you log in to your app and get to the home page:



```
Session {
  cookie: { path: '/', _expires: null, originalMaxAge: null, httpOnly: true },
  userId: '60988b11369f556824075887' 로그인 안하면 Null
}
```

- **In newPost.js**
  - …

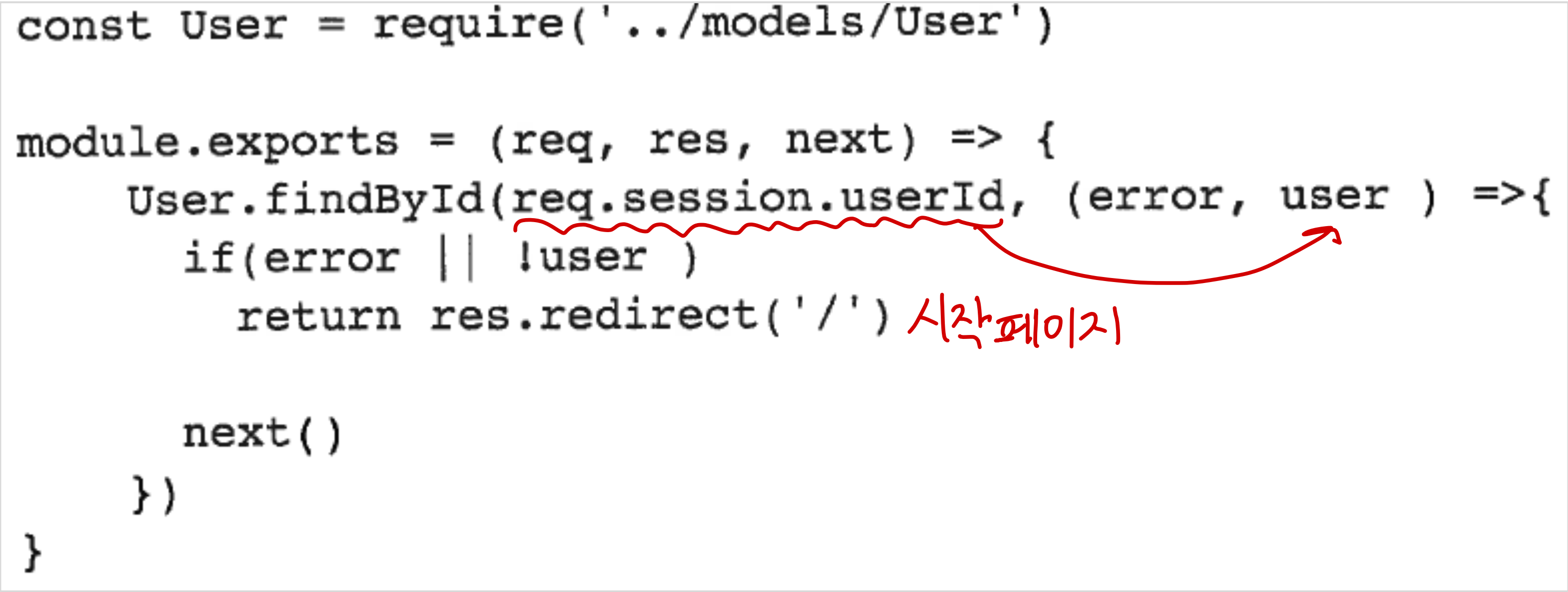


```
module.exports = (req, res) =>{
   if(req.session.userId){
      return res.render("create");
    }
    res.redirect('/auth/login')
 }
```

로그인되어있으면
post 가능

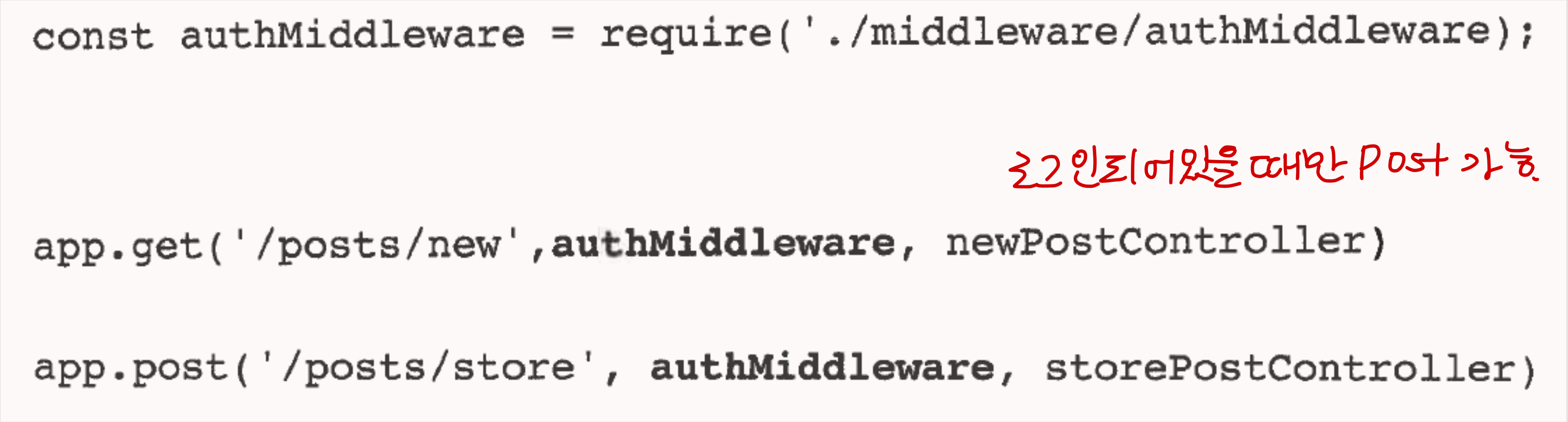
Protecting Pages with Middleware

- Create a custom middleware : **/middleware/authMiddleware.js**



```
const User = require('../models/User')

module.exports = (req, res, next) => {
    User.findById(req.session.userId, (error, user ) =>{
      if(error || !user )
        return res.redirect('/') 시작페이지

      next()
    })
}
```

- **In index.js :**



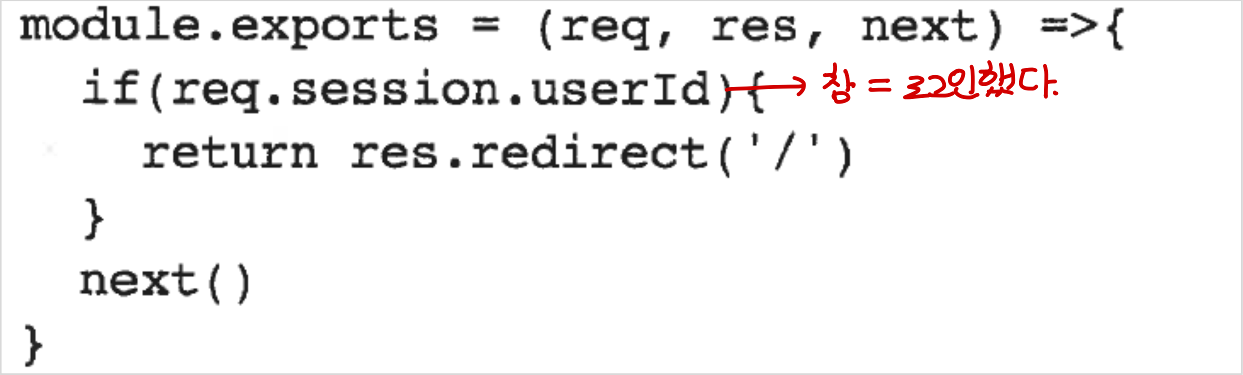
```
const authMiddleware = require('./middleware/authMiddleware);
```

로그인되어있을때만 Post 가능.

```
app.get('/posts/new',authMiddleware, newPostController)

app.post('/posts/store', authMiddleware, storePostController)
```

Prevent Login/Register if Logged In

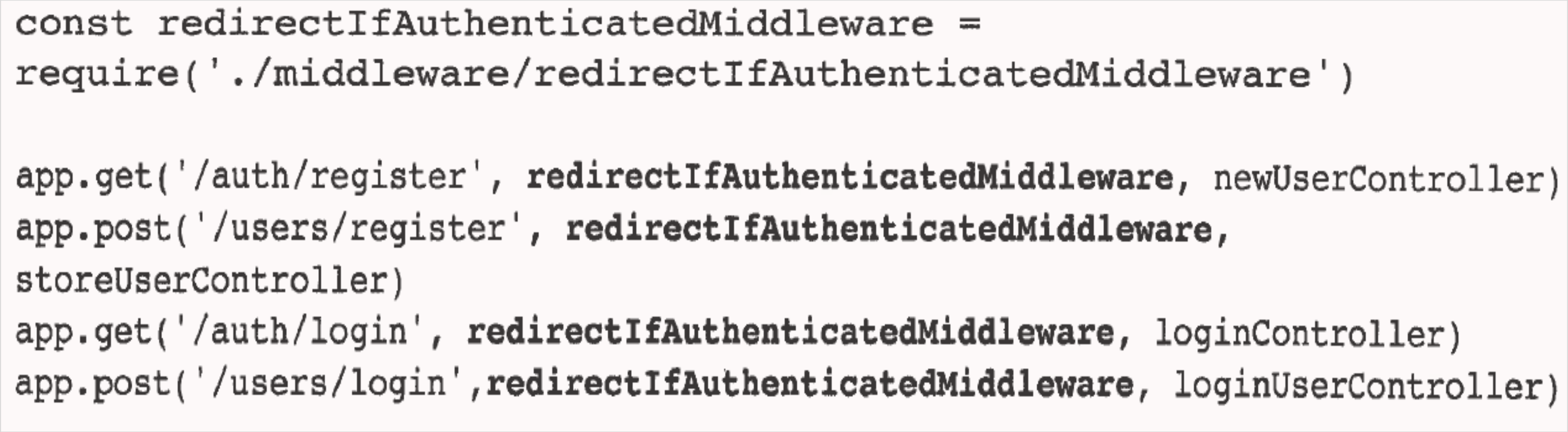
- In middleware folder, create **redirectIfAuthenticatedMiddleware.js**



```
module.exports = (req, res, next) =>{
  if(req.session.userId){
    return res.redirect('/')
  }
  next()
}
```

→ 참 = 로그인했다.

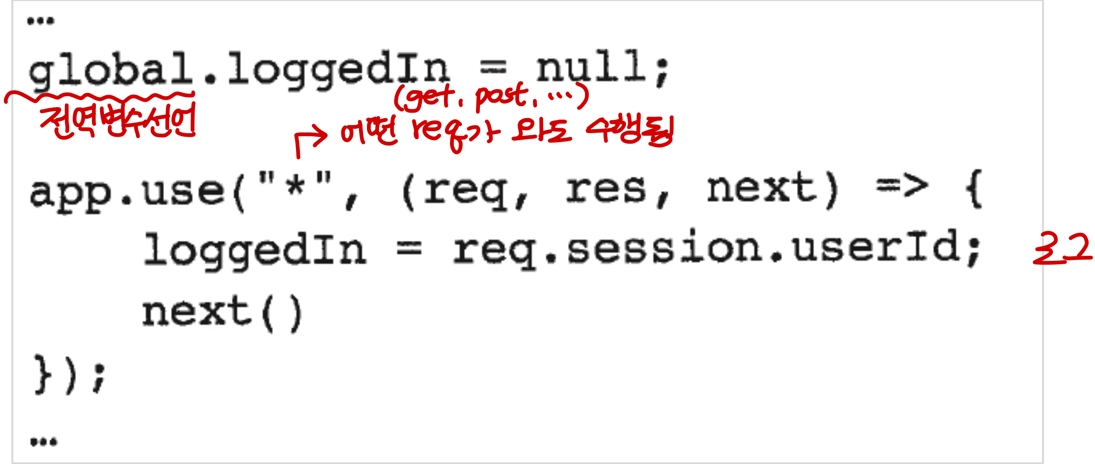
- **In index.js :**



```
const redirectIfAuthenticatedMiddleware =
require('./middleware/redirectIfAuthenticatedMiddleware')

app.get('/auth/register', redirectIfAuthenticatedMiddleware, newUserController)
app.post('/users/register', redirectIfAuthenticatedMiddleware,
storeUserController)
app.get('/auth/login', redirectIfAuthenticatedMiddleware, loginController)
app.post('/users/login',redirectIfAuthenticatedMiddleware, loginUserController)
```

Conditionally Display New Post, Login and New User links

- **In index.js** : add



```
…
global.loggedIn = null;

app.use("*", (req, res, next) => {
    loggedIn = req.session.userId;
    next()
});
…
```

전역변수선언

(get, post, …)

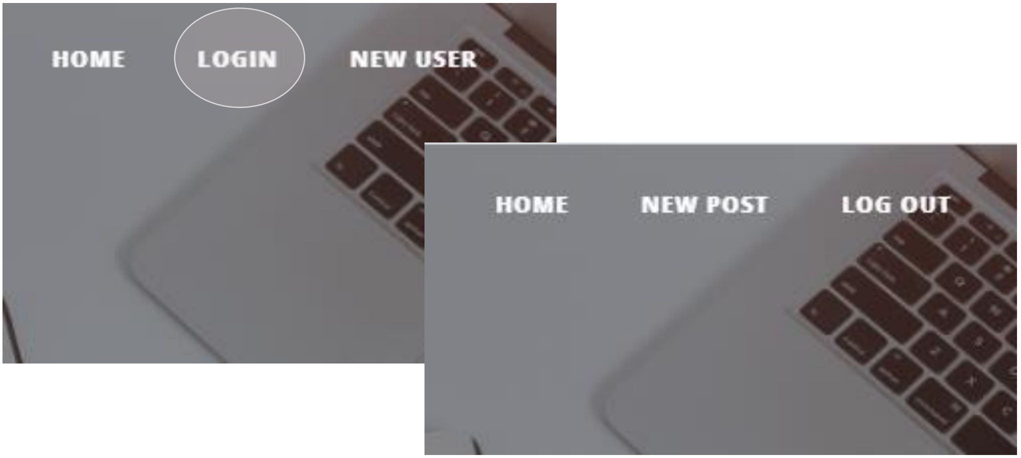
→ 어떤 req가 와도 수행됨

로그인 < O : 객체ID값
X : Null

들어감.

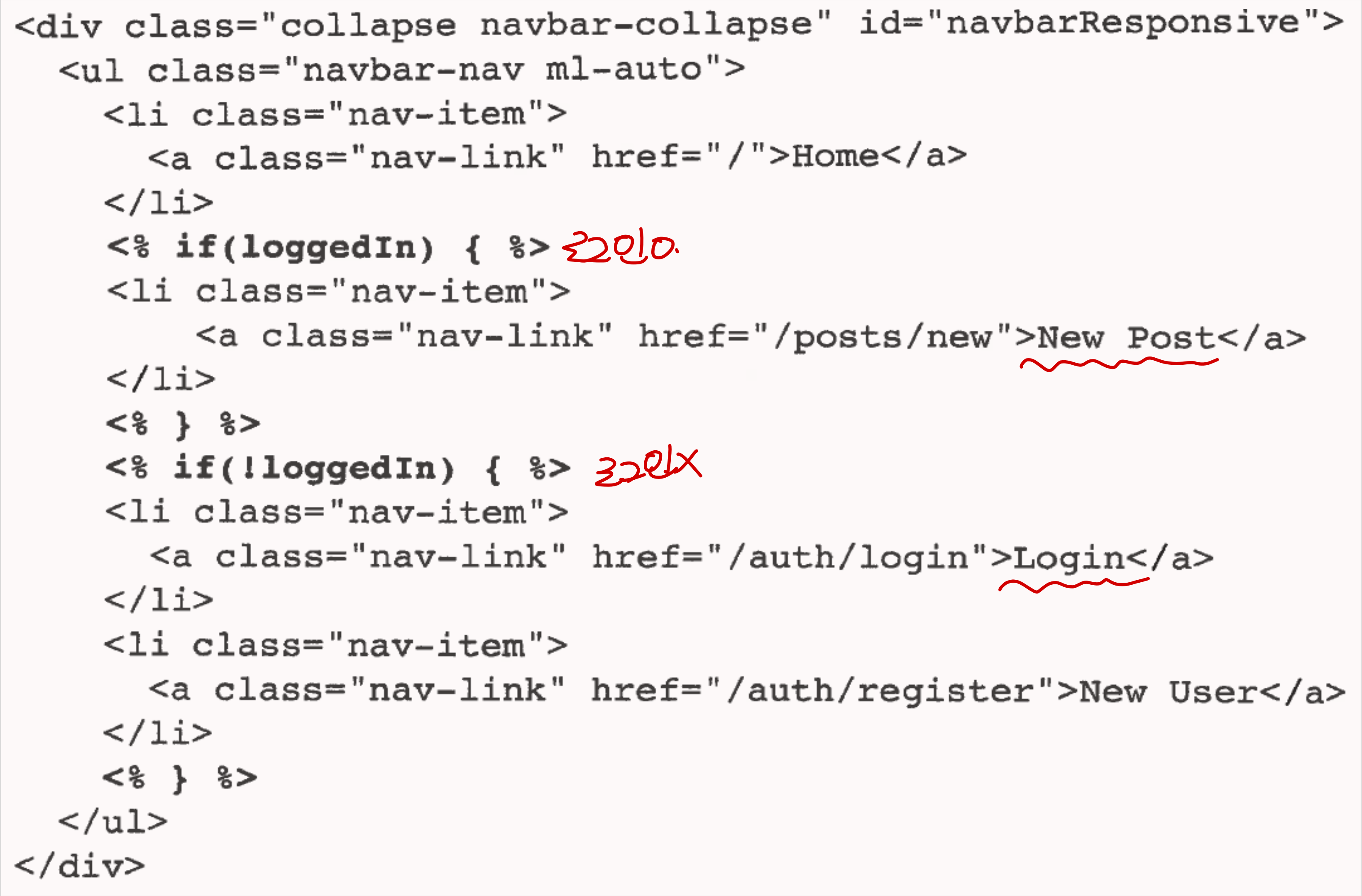
- loggedIn : **global variable** that will be **accessible from all our EJS files**

로그인 안한경우



로그인 한경우

- **In /layout/navbar.ejs :**

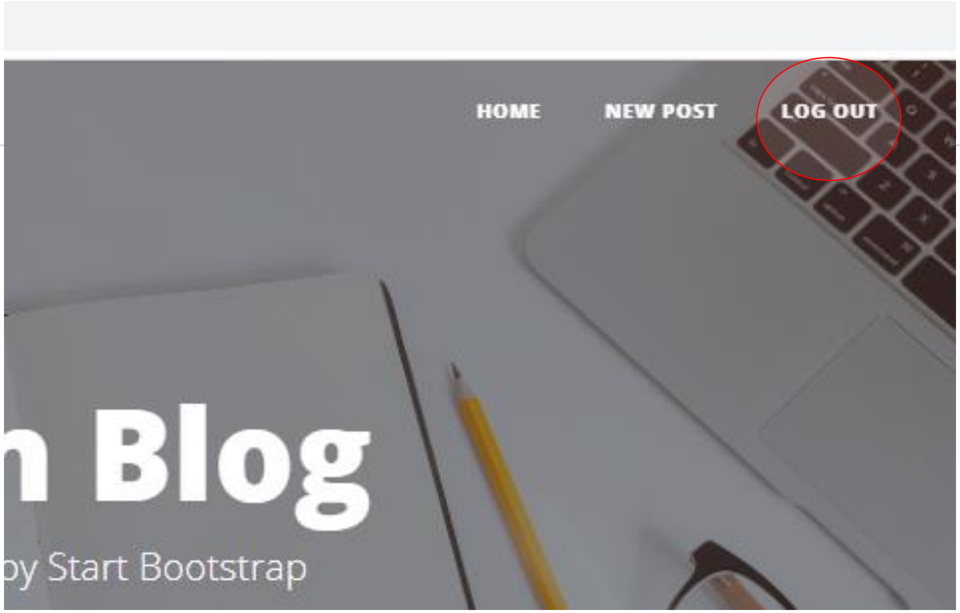


```
<div class="collapse navbar-collapse" id="navbarResponsive">
  <ul class="navbar-nav ml-auto">
    <li class="nav-item">
      <a class="nav-link" href="/">Home</a>
    </li>
    <% if(loggedIn) { %> 로그인O
    <li class="nav-item">
        <a class="nav-link" href="/posts/new">New Post</a>
    </li>
    <% } %>
    <% if(!loggedIn) { %> 로그인X
    <li class="nav-item">
      <a class="nav-link" href="/auth/login">Login</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="/auth/register">New User</a>
    </li>
    <% } %>
  </ul>
</div>
```

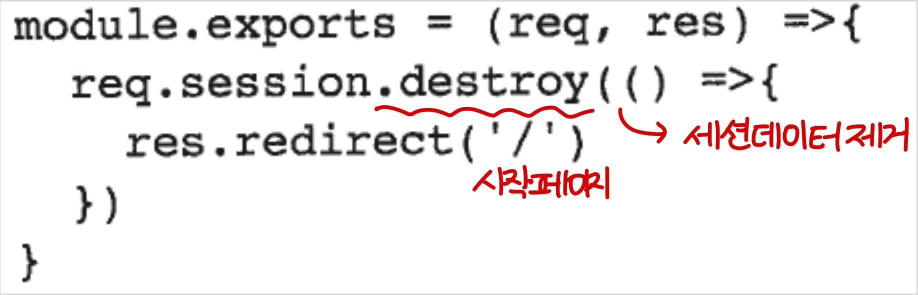
User Logout

- **In /layout/navbar.ejs :**

```
…
    <% if(loggedIn) { %>
    <li class="nav-item">
        <a class="nav-link" href="/posts/new">New Post</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="/auth/logout">Log out</a>
    </li>
    <% } %>
…
```



- Create the log out controller : **/controllers/logout.js:**



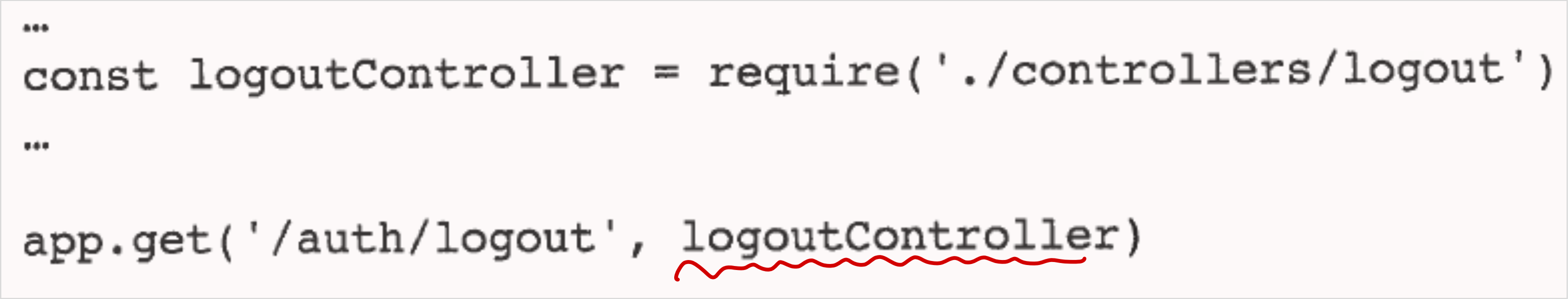
```
module.exports = (req, res) =>{
  req.session.destroy(() =>{
    res.redirect('/')
  })
}
```

세션데이터제거

시작페이지

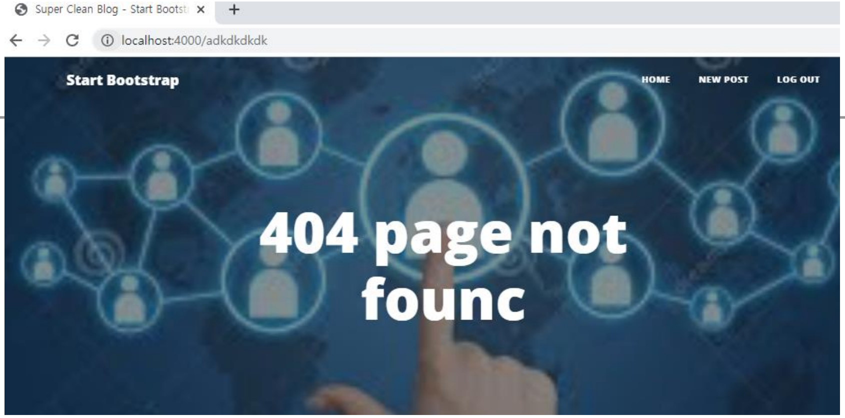
  - req.session.destroy() : destroy all session data

- **In index.js :**



```
…
const logoutController = require('./controllers/logout')
…

app.get('/auth/logout', logoutController)
```

Creating a 404 page



404 page not founc

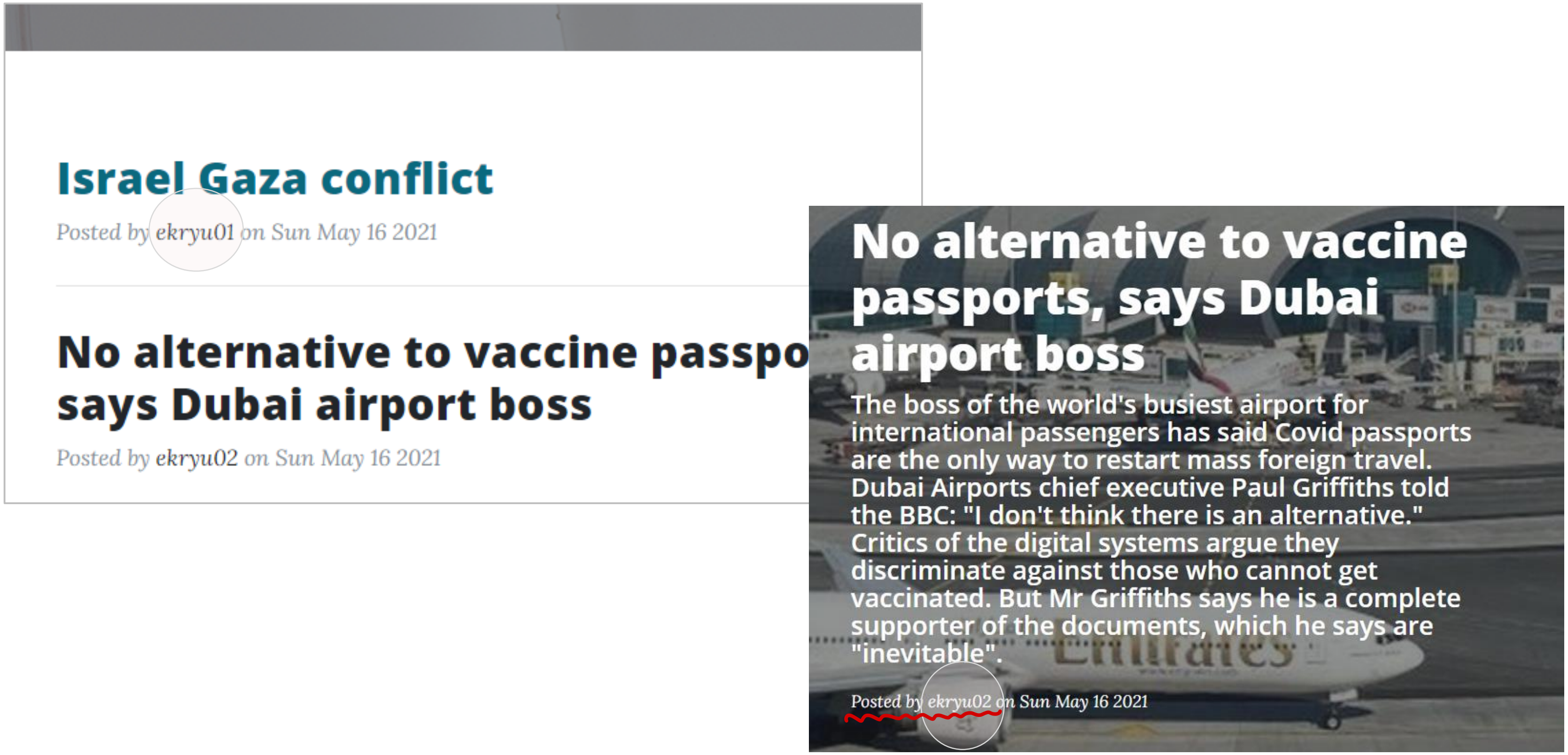
- **Ex) localhost:4000/sdfsfs**
  - "Cannot GET / sdfsfs" error

- In views folder, create **notfound.ejs** (copying from login.ejs)
  - Change its **h1** header to "404 page not found".

- **In index.js**
  - At the end of the file, add:

```
app.get('/auth/login', redirectIfAuthenticatedMiddleware,
loginController)
app.post('/users/login',redirectIfAuthenticatedMiddleware,
loginUserController)
app.get('/auth/logout', logoutController)
app.use((req, res) => res.render('notfound'));
```

마지막줄
에 추가



Israel Gaza conflict

Posted by ekryu01 on Sun May 16 2021

No alternative to vaccine passpo says Dubai airport boss

Posted by ekryu02 on Sun May 16 2021

No alternative to vaccine passports, says Dubai airport boss

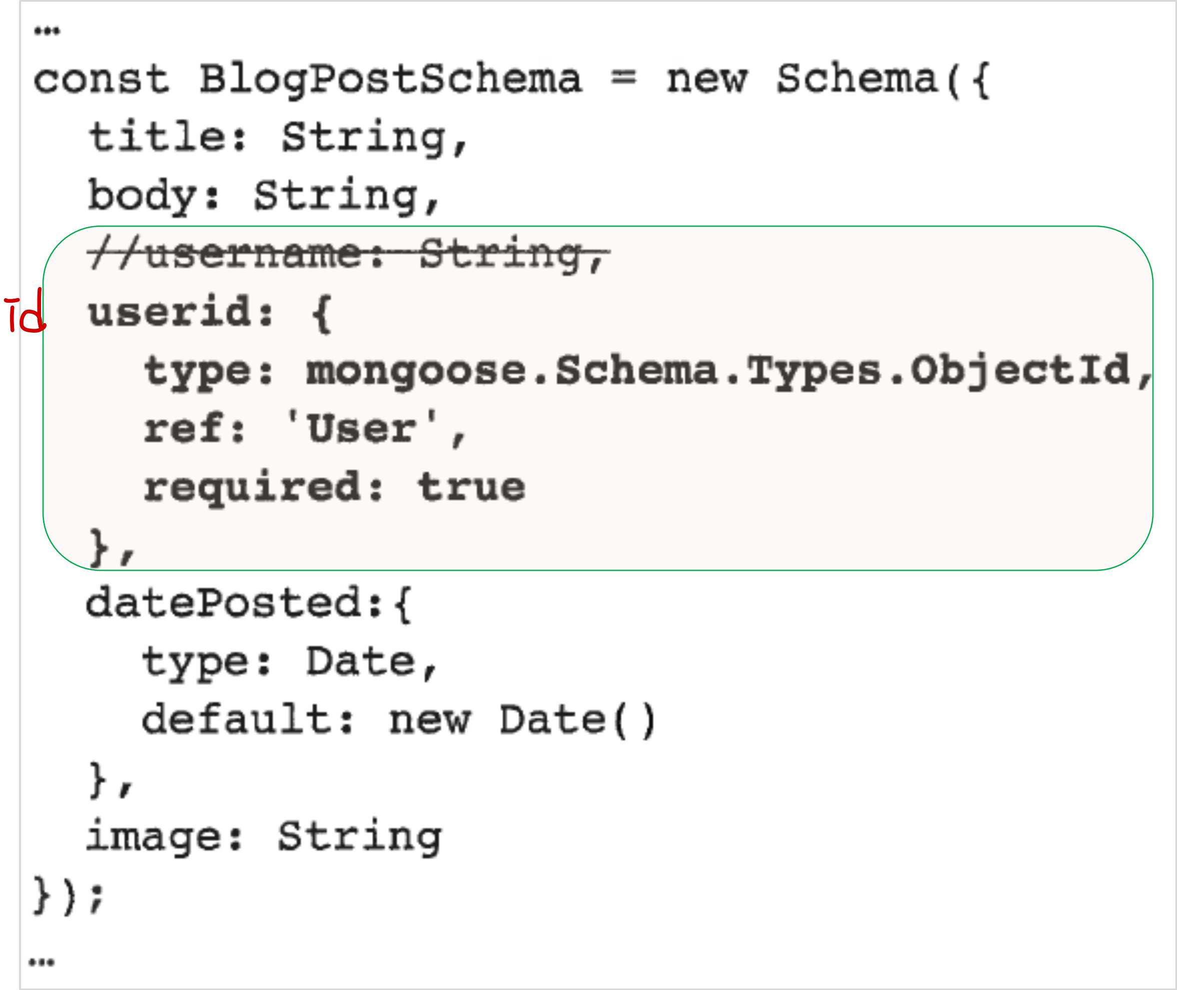
The boss of the world's busiest airport for international passengers has said Covid passports are the only way to restart mass foreign travel. Dubai Airports chief executive Paul Griffiths told the BBC: "I don't think there is an alternative." Critics of the digital systems argue they discriminate against those who cannot get vaccinated. But Mr Griffiths says he is a complete supporter of the documents, which he says are "inevitable".

Posted by ekryu02 on Sun May 16 2021

RELATING POSTS COLLECTION WITH USERS COLLECTION

Link the Users and Posts collection

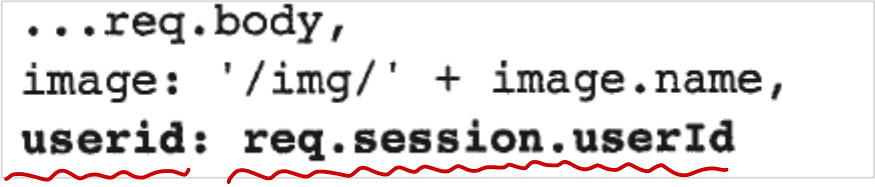
- **In models/BlogPost.js :**



```
…
const BlogPostSchema = new Schema({
   title: String,
   body: String,
   ~~//username: String,~~
   userid: {
      type: mongoose.Schema.Types.ObjectId,
      ref: 'User',
      required: true
   },
   datePosted:{
      type: Date,
      default: new Date()
   },
   image: String
});
…
```

객체id

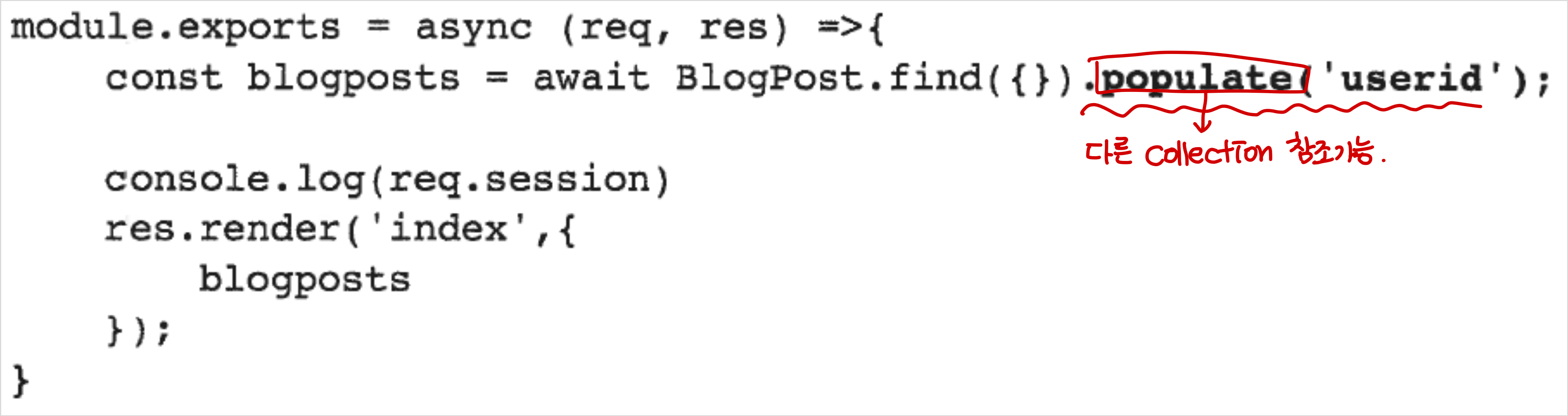
- **In storePost.js :**



```
...req.body,
image: '/img/' + image.name,
userid: req.session.userId
```

Display Post with User Data

- In controllers/home.js :

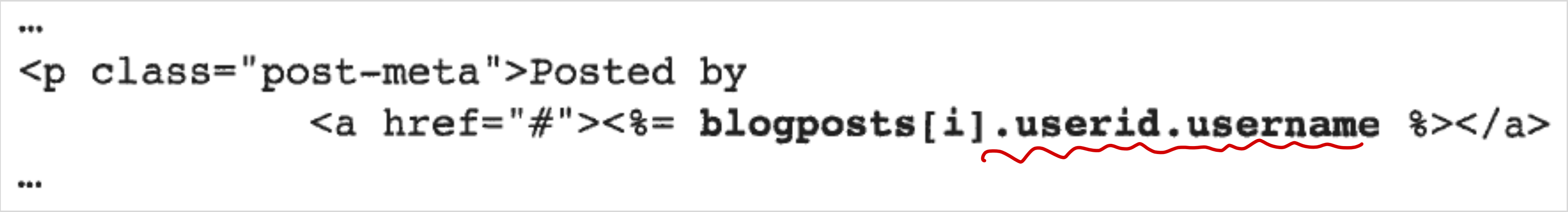


```
module.exports = async (req, res) =>{
    const blogposts = await BlogPost.find({}).populate('userid');

    console.log(req.session)
    res.render('index',{
        blogposts
    });
}
```

다른 collection 참조가능.

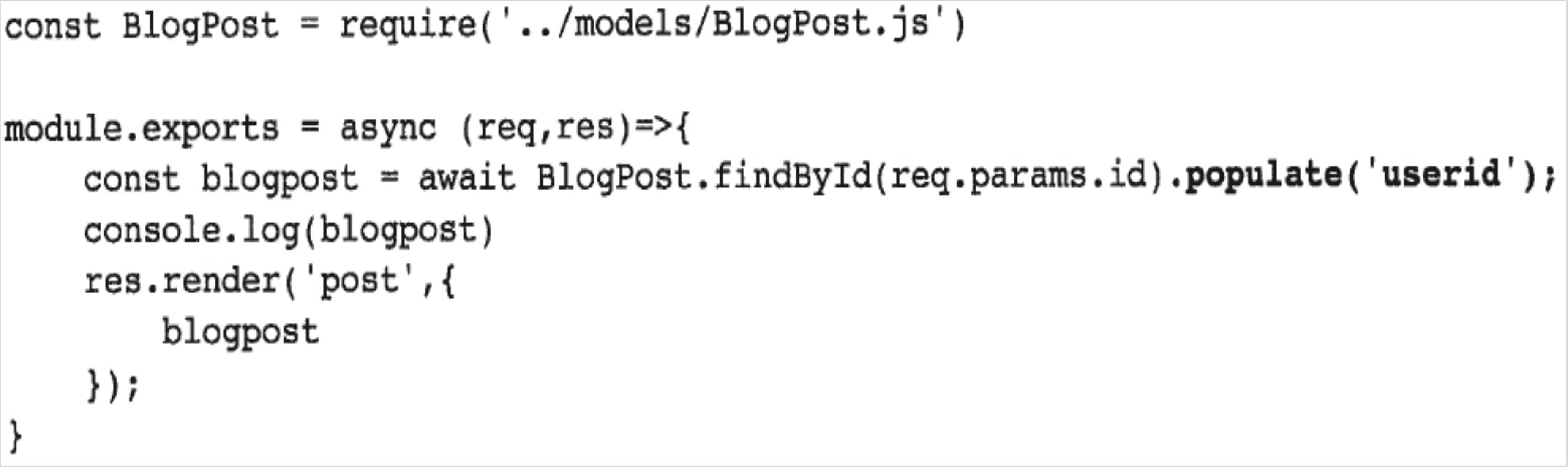
- **populate**() lets you reference documents in other collections.

- **In views/index.ejs :**



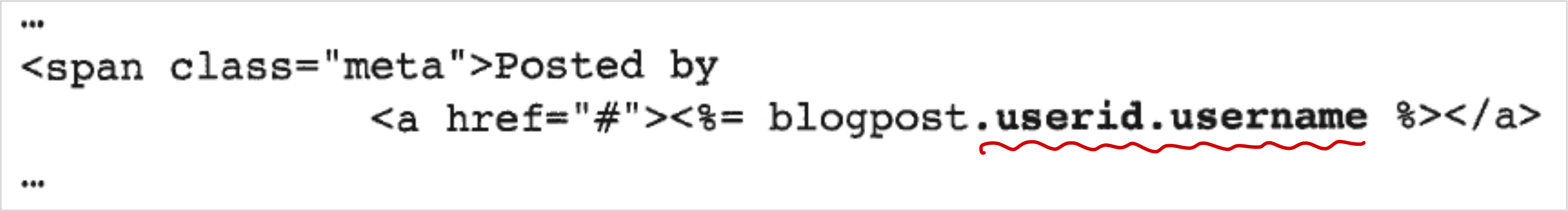
```
…
<p class="post-meta">Posted by
            <a href="#"><%= blogposts[i].userid.username %></a>
…
```

- **In controllers/getPost.js :**



```
const BlogPost = require('../models/BlogPost.js')

module.exports = async (req,res)=>{
    const blogpost = await BlogPost.findById(req.params.id).populate('userid');
    console.log(blogpost)
    res.render('post',{
        blogpost
    });
}
```

- **In views/Post.ejs :**



```
...
<span class="meta">Posted by
              <a href="#"><%= blogpost.userid.username %></a>
...
```

Summary

- APPLYING **MONGODB** TO Blog PROJECT
- **UPLOADING IMAGE** WITH EXPRESS
- INTRODUCTION TO EXPRESS **MIDDLEWARE**
- REFACTORING TO **MVC**
- **USER REGISTRATION**
- USER AUTHENTICATION WITH **EXPRESS SESSIONS**