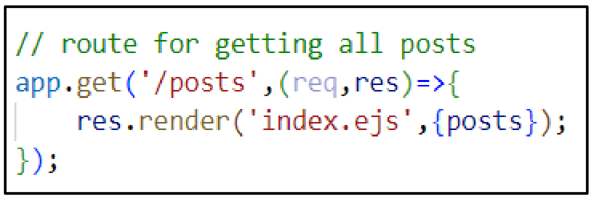
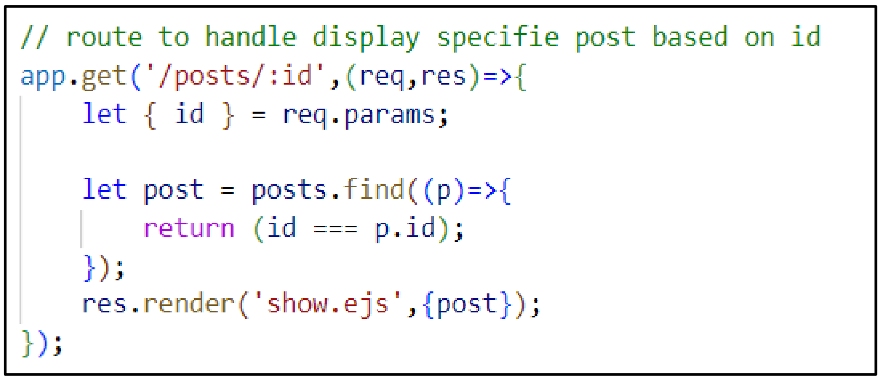
Getting all Posts using GET Request



```
// route for getting all posts
app.get('/posts',(req,res)=>{
    res.render('index.ejs',{posts});
});
```

Getting specific Post based on Id using GET Request

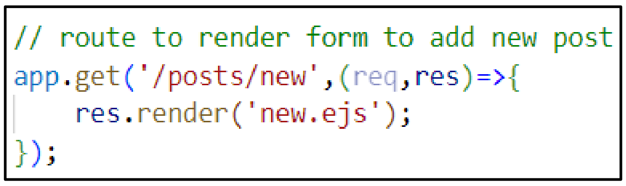


```
// route to handle display specifie post based on id
app.get('/posts/:id',(req,res)=>{
    let { id } = req.params;

    let post = posts.find((p)=>{
        return (id === p.id);
    });
    res.render('show.ejs',{post});
});
```

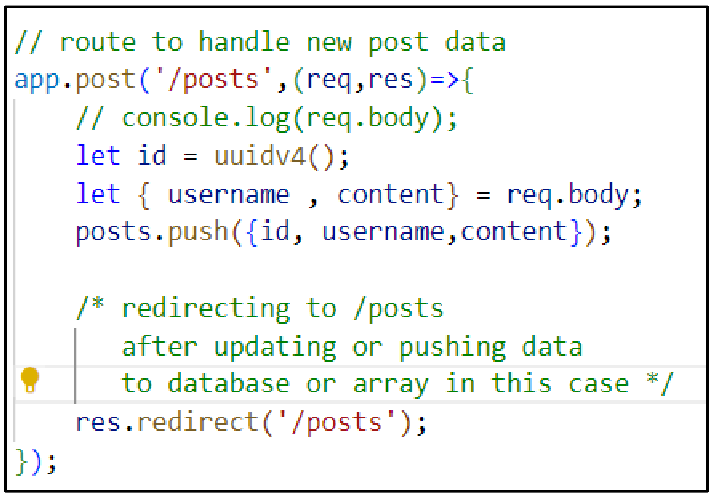
Requesting to create a new Post using POST Request

Step-1 :- requesting for a page containing form to enter new post detail



```
// route to render form to add new post
app.get('/posts/new',(req,res)=>{
    res.render('new.ejs');
});
```

Step-2 :- route handling form data and submit new post detail and redirect to homepage (/posts)



```
// route to handle new post data
app.post('/posts',(req,res)=>{
    // console.log(req.body);
    let id = uuidv4();
    let { username , content} = req.body;
    posts.push({id, username,content});

    /* redirecting to /posts
        after updating or pushing data
        to database or array in this case */
    res.redirect('/posts');
});
```

Requesting to Update a Post based on ID using PATCH Request

Step-1 :- Include method-override package to receive PUT, PATCH or
. DELETE Request

```
// including module to receive http requests like put, patch, or delete
const methodOverride = require('method-override');
```

Step-2 :- Include method-override middleware in **index.js**

```
// midleware TO OVERRIDE METHODS REQUEST
app.use(methodOverride('_method'));
```

Step-3 :- Changing form action method in **edit.ejs**

```
<form method="post" action="/posts/<%= post.id %>?_method=PATCH">
    <textarea name="content" rows="10" cols="35">
        <%= post.content %>
    </textarea>
```

Step-4 :- route handling form data and submit edit post detail and redirect to homepage (/posts)

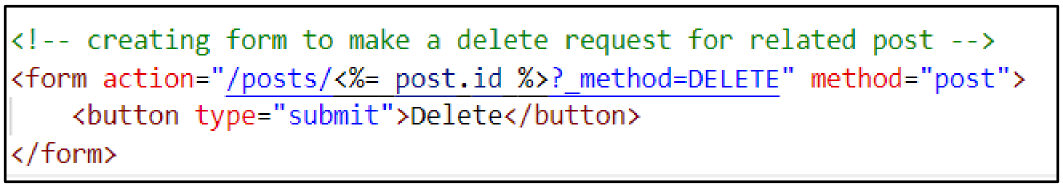
```
// route to handle edit request
app.get('/posts/:id/edit',(req,res)=>{
    let { id } = req.params;
    let post = posts.find((p)=>{
        return (id === p.id);
    });

    res.render('edit.ejs', { post });
});
```

```
// route to update our specific post
app.patch('/posts/:id',(req,res)=>{
    let { id } = req.params;
    // console.log(id);
    let post = posts.find((p)=>{
        return (id === p.id);
    });
    // update partial data of post
    post.content = req.body.content;
    // redirect to /posts after updating data
    res.redirect('/posts');
});
```

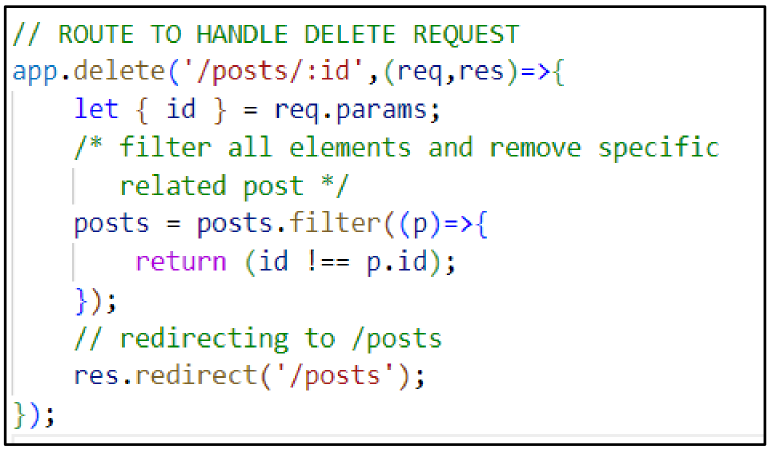
Requesting to Update a Post based on ID using DELETE Request

Step-1 :- adding delete button and changing form action method in **index.ejs**



```
<!-- creating form to make a delete request for related post -->
<form action="/posts/<%= post.id %>?_method=DELETE" method="post">
    <button type="submit">Delete</button>
</form>
```

Step-2 :- route handling form data and submit delete post detail and redirect to homepage (/posts)



```
// ROUTE TO HANDLE DELETE REQUEST
app.delete('/posts/:id',(req,res)=>{
    let { id } = req.params;
    /* filter all elements and remove specific
       related post */
    posts = posts.filter((p)=>{
        return (id !== p.id);
    });
    // redirecting to /posts
    res.redirect('/posts');
});
```

SCAN this QR code to get full notes and source code