

MERN stack - Shopping Online

Lab 03

Contents

1 MERN stack	2
1.1 Functionals	2
1.1.1 Admin - listcategory	2
1.1.2 Admin - addcategory	5
1.1.3 Admin - updatecategory	6
1.1.4 Admin - deletecategory	8

1 MERN stack

1.1 Functionals

1.1.1 Admin - listcategory

1.1.1.1 Server

- Create 'server/models/CategoryDAO.js' file

```

1 require('../utils/MongooseUtil');
2 const Models = require('../Models');
3
4 const CategoryDAO = {
5   async selectAll() {
6     const query = {};
7     const categories = await Models.Category.find(query).exec();
8     return categories;
9   }
10 };
11 module.exports = CategoryDAO;

```

- Update 'server/api/admin.js' file

```

1 ...
2 // daos
3 const CategoryDAO = require('../models/CategoryDAO');
4 ...
5 // category
6 router.get('/categories', JwtUtil.checkToken, async function (req, res) {
7   const categories = await CategoryDAO.selectAll();
8   res.json(categories);
9 });
10 ...

```

- Postman: (GET) http://localhost:3000/api/admin/categories

- Headers: "x-access-token": <token>

1.1.1.2 Client-admin

- Update 'client-admin/src/components/MainComponent.js' file

```

1 ...
2 import Category from './CategoryComponent';
3
4 class Main extends Component {
5   ...
6   render() {
7     ...
8     return (
9       ...
10      <Routes>
11        ...
12          <Route path='/admin/category' element={<Category />} />
13        </Routes>
14      );
15    ...
16  }
17}
18 ...

```

- Update 'client-admin/src/components/MenuComponent.js' file

```

1 ...
2 class Menu extends Component {
3   ...
4   render() {
5     return (

```

```

6     ...
7     <ul className="menu">
8         ...
9             <li className="menu"><Link to='/admin/category'>Category</Link></li>
10            ...
11        </ul>
12        ...
13    );
14 }
15 ...
16 }
17 ...

```

- Create 'client-admin/src/components/CategoryComponent.js' file

```

1 import axios from 'axios';
2 import React, { Component } from 'react';
3 import MyContext from '../contexts/MyContext';
4 import CategoryDetail from './CategoryDetailComponent';
5
6 class Category extends Component {
7     static contextType = MyContext; // using this.context to access global state
8     constructor(props) {
9         super(props);
10        this.state = {
11            categories: [],
12            itemSelected: null
13        };
14    }
15    render() {
16        const cates = this.state.categories.map((item) => {
17            return (
18                <tr key={item._id} className="datatable" onClick={() => this.trItemClick(item)}>
19                    <td>{item._id}</td>
20                    <td>{item.name}</td>
21                </tr>
22            );
23        });
24        return (
25            <div>
26                <div className="float-left">
27                    <h2 className="text-center">CATEGORY LIST</h2>
28                    <table className="datatable" border="1">
29                        <tbody>
30                            <tr className="datatable">
31                                <th>ID</th>
32                                <th>Name</th>
33                            </tr>
34                            {cates}
35                        </tbody>
36                    </table>
37                </div>
38                <div className="inline" />
39                <CategoryDetail item={this.state.itemSelected} />
40                <div className="float-clear" />
41            </div>
42        );
43    }
44    componentDidMount() {
45        this.apiGetCategories();
46    }
47    // event-handlers
48    trItemClick(item) {
49        this.setState({ itemSelected: item });
50    }
51    // apis
52    apiGetCategories() {
53        const config = { headers: { 'x-access-token': this.context.token } };
54        axios.get('/api/admin/categories', config).then((res) => {

```

```

55     const result = res.data;
56     this.setState({ categories: result });
57   );
58 }
59
60 export default Category;

```

- Create 'client-admin/src/components/CategoryDetailComponent.js' file

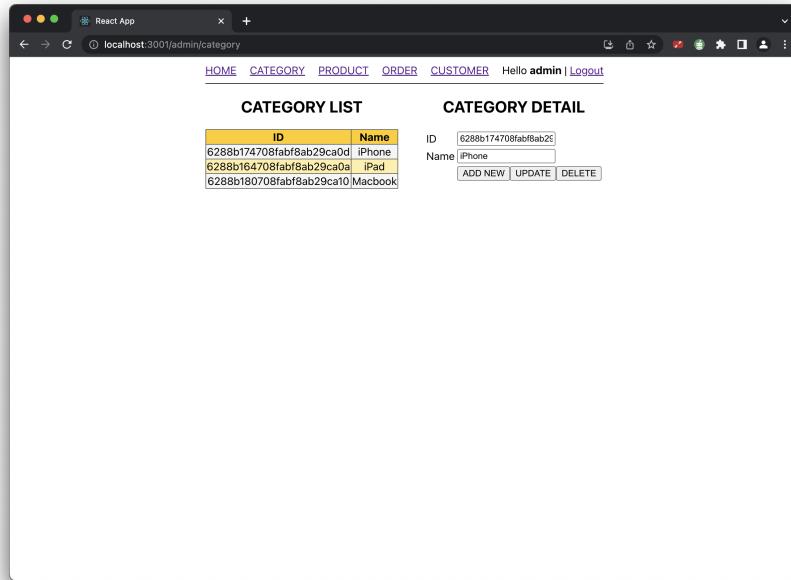
```

1 import axios from 'axios';
2 import React, { Component } from 'react';
3 import MyContext from '../contexts/MyContext';
4
5 class CategoryDetail extends Component {
6   static contextType = MyContext; // using this.context to access global state
7   constructor(props) {
8     super(props);
9     this.state = {
10       txtID: '',
11       txtName: ''
12     };
13   }
14   render() {
15     return (
16       <div className="float-right">
17         <h2 className="text-center">CATEGORY DETAIL</h2>
18         <form>
19           <table>
20             <tbody>
21               <tr>
22                 <td>ID</td>
23                 <td><input type="text" value={this.state.txtID} onChange={(e) => { this.
24                   setState({ txtID: e.target.value }) }} readOnly={true} /></td>
25               </tr>
26               <tr>
27                 <td>Name</td>
28                 <td><input type="text" value={this.state.txtName} onChange={(e) => { this.
29                   setState({ txtName: e.target.value }) }} /></td>
30               </tr>
31               <tr>
32                 <td></td>
33                 <td>
34                   <input type="submit" value="ADD NEW" />
35                   <input type="submit" value="UPDATE" />
36                   <input type="submit" value="DELETE" />
37                 </td>
38               </tr>
39             </tbody>
40           </table>
41         </form>
42       </div>
43     );
44   }
45   componentDidUpdate(prevProps) {
46     if (this.props.item !== prevProps.item) {
47       this.setState({ txtID: this.props.item._id, txtName: this.props.item.name });
48     }
49 }
50
51 export default CategoryDetail;

```

- Browser: http://localhost:3001/admin/category

1.1.1.3 Screenshots



1.1.2 Admin - addcategory

1.1.2.1 Server

- Update 'server/models/CategoryDAO.js' file

```

1 ...
2 const CategoryDAO = {
3   ...
4   async insert(category) {
5     const mongoose = require('mongoose');
6     category._id = new mongoose.Types.ObjectId();
7     const result = await Models.Category.create(category);
8     return result;
9   }
10 };
11 ...

```

- Update 'server/api/admin.js' file

```

1 ...
2 // category
3 router.post('/categories', JwtUtil.checkToken, async function (req, res) {
4   const name = req.body.name;
5   const category = { name: name };
6   const result = await CategoryDAO.insert(category);
7   res.json(result);
8 });
9 ...

```

- Postman: (POST) http://localhost:3000/api/admin/categories

- Headers: "x-access-token": <token>
- Body (raw+JSON): { "name": "Watch" }

1.1.2.2 Client-admin

- Update 'client-admin/src/components/CategoryComponent.js' file

```

1 ...
2 class Category extends Component {
3   ...
4   render() {
5     ...

```

```

6   return (
7     <div>
8       ...
9         <CategoryDetail item={this.state.item} updateCategories={this.updateCategories} />
10      ...
11    );
12  }
13  updateCategories = (categories) => { // arrow-function
14    this.setState({ categories: categories });
15  }
16  ...
17}
18...
19

```

- Update 'client-admin/src/components/CategoryDetailComponent.js' file

```

1 ...
2 class CategoryDetail extends Component {
3   ...
4   render() {
5     ...
6     return (
7       ...
8         <input type="submit" value="ADD NEW" onClick={(e) => this.btnAddClick(e)} />
9       ...
10    );
11  ...
12 }
13 // event-handlers
14 btnAddClick(e) {
15   e.preventDefault();
16   const name = this.state.txtName;
17   if (name) {
18     const cate = { name: name };
19     this.apiPostCategory(cate);
20   } else {
21     alert('Please input name');
22   }
23 }
24 // apis
25 apiPostCategory(cate) {
26   const config = { headers: { 'x-access-token': this.context.token } };
27   axios.post('/api/admin/categories', cate, config).then((res) => {
28     const result = res.data;
29     if (result) {
30       alert('OK BABY!');
31       this.apiGetCategories();
32     } else {
33       alert('SORRY BABY!');
34     }
35   });
36 }
37 apiGetCategories() {
38   const config = { headers: { 'x-access-token': this.context.token } };
39   axios.get('/api/admin/categories', config).then((res) => {
40     const result = res.data;
41     this.props.updateCategories(result);
42   });
43 }
44 ...
45 }
46 ...

```

- Browser: <http://localhost:3001/admin/category>

1.1.3 Admin - updatecategory

1.1.3.1 Server

- Update 'server/models/CategoryDAO.js' file

```

1 ...
2 const CategoryDAO = {
3   ...,
4   async update(category) {
5     const newvalues = { name: category.name }
6     const result = await Models.Category.findByIdAndUpdate(category._id, newvalues, { new: true })
7     return result;
8   }
9 };
10 ...

```

- Update 'server/api/admin.js' file

```

1 ...
2 // category
3 router.put('/categories/:id', JwtUtil.checkToken, async function (req, res) {
4   const _id = req.params.id;
5   const name = req.body.name;
6   const category = { _id: _id, name: name };
7   const result = await CategoryDAO.update(category);
8   res.json(result);
9 });
10 ...

```

- Postman: (PUT) http://localhost:3000/api/admin/categories/<id>

- Headers: "x-access-token": <token>
- Body (raw+JSON): { "name": "Watch2" }

1.1.3.2 Client-admin

- Update 'client-admin/src/components/CategoryDetailComponent.js' file

```

1 ...
2 class CategoryDetail extends Component {
3   ...
4   render() {
5     ...
6     return (
7       ...
8       <input type="submit" value="UPDATE" onClick={(e) => this.btnUpdateClick(e)} />
9       ...
10    );
11   ...
12 }
13 // event-handlers
14 btnUpdateClick(e) {
15   e.preventDefault();
16   const id = this.state.txtID;
17   const name = this.state.txtName;
18   if (id && name) {
19     const cate = { name: name };
20     this.apiPutCategory(id, cate);
21   } else {
22     alert('Please input id and name');
23   }
24 }
25 // apis
26 apiPutCategory(id, cate) {
27   const config = { headers: { 'x-access-token': this.context.token } };
28   axios.put('/api/admin/categories/' + id, cate, config).then((res) => {
29     const result = res.data;
30     if (result) {
31       alert('OK BABY!');
32       this.apiGetCategories();
33     }
34   });
35 }
36 
```

```

33     } else {
34         alert('SORRY BABY!');
35     }
36 });
37 }
38 ...
39 }
...

```

- Browser: http://localhost:3001/admin/category

1.1.4 Admin - deletecategory

1.1.4.1 Server

- Update 'server/models/CategoryDAO.js' file

```

1 ...
2 const CategoryDAO = {
3     ...,
4     async delete(_id) {
5         const result = await Models.Category.findByIdAndRemove(_id);
6         return result;
7     }
8 };
9 ...

```

mongoose 6tr lên ph i dùng
findOneAndDelete(_id), không dùng
findByIdAndRemove

- Update 'server/api/admin.js' file

```

1 ...
2 // category
3 router.delete('/categories/:id', JwtUtil.checkToken, async function (req, res) {
4     const _id = req.params.id;
5     const result = await CategoryDAO.delete(_id);
6     res.json(result);
7 });
8 ...

```

- Postman: (DELETE) http://localhost:3000/api/admin/categories/<id>

- Headers: "x-access-token": <token>

1.1.4.2 Client-admin

- Update 'client-admin/src/components/CategoryDetailComponent.js' file

```

1 ...
2 class CategoryDetail extends Component {
3     ...
4     render() {
5         ...
6         return (
7             ...
8                 <input type="submit" value="DELETE" onClick={(e) => this.btnDeleteClick(e)} />
9                 ...
10            );
11        ...
12    }
13    // event-handlers
14    btnDeleteClick(e) {
15        e.preventDefault();
16        if (window.confirm('ARE YOU SURE?')) {
17            const id = this.state.txtID;
18            if (id) {
19                this.apiDeleteCategory(id);
20            } else {
21                alert('Please input id');
22            }

```

```
23     }
24 }
25 // apis
26 apiDeleteCategory(id) {
27   const config = { headers: { 'x-access-token': this.context.token } };
28   axios.delete('/api/admin/categories/' + id, config).then((res) => {
29     const result = res.data;
30     if (result) {
31       alert('OK BABY!');
32       this.apiGetCategories();
33     } else {
34       alert('SORRY BABY!');
35     }
36   });
37 }
38 ...
39 }
40 ...
```

- Browser: http://localhost:3001/admin/category