

## Scan, insert, delete item trong DynamoDB:

**Bước 1:** Install thư viện aws-sdk: `npm install aws-sdk`

**Bước 2:** Chạy server development: `npm run start`

**Bước 3:** Update trang `index.js` theo mẫu sau:

```
6 app.use(express.static(path.resolve(__dirname, '../views')));
7 app.use(express.static('../views'));
8 app.set('view engine', 'ejs');
9 app.set('views', './views');
10
11 // config aws dynamodb
12 const AWS = require('aws-sdk');
13 const config = new AWS.Config({
14   accessKeyId: 'AKIA5GY2TNE4NTATWY',
15   secretAccessKey: 'Vxj8URu29TafG0858nT0ufs1yUp/rqjJHNS5/k4+',
16   region: 'ap-southeast-1'
17 });
18 AWS.config = config;
19
20 const docClient = new AWS.DynamoDB.DocumentClient();
21
22 const tableName = 'SanPham';
23
24 app.get('/', (request, response) => {
25   const params = {
26     TableName: tableName,
27   };
28
29   docClient.scan(params, (err, data) => {
30     if (err) {
31       response.send('Internal Server Error');
32     } else {
33       return response.render('index', { sanPhams: data.Items });
34     }
35   });
36 });
```

`render(view: string, options?: object, callback?: (err: Error, html: string) => void): void`

Render view with the given options and optional callback fn. When a callback function is given a response will not be made automatically, otherwise a response of 200 and `text/html` is given.

Options:

- `cache` boolean hinting to the engine it should cache
- `filename` filename of the view being rendered

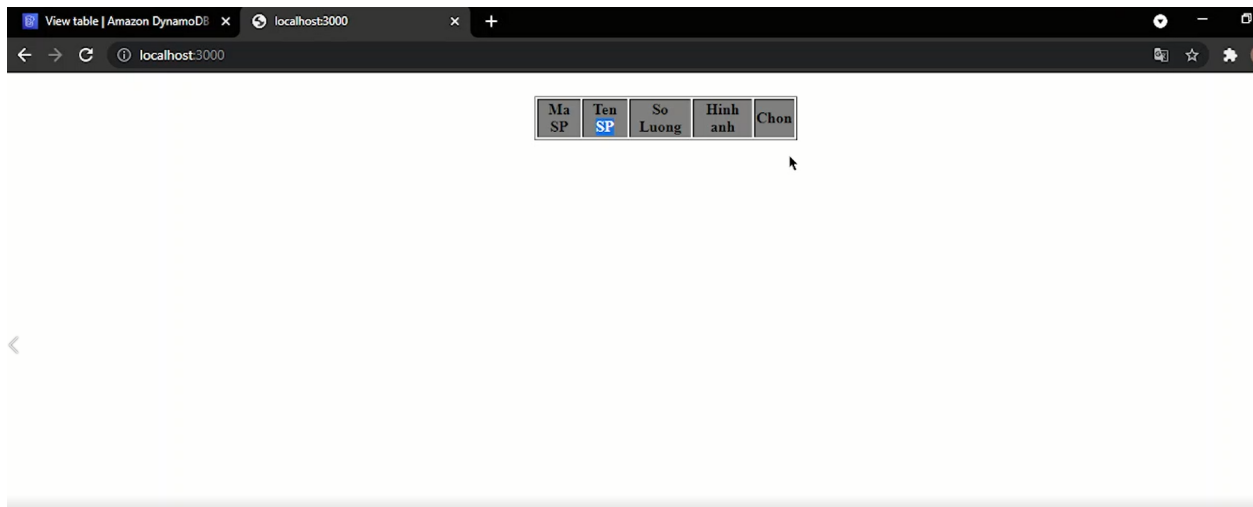
Lưu ý: `accessKeyId`, `secretAccessKey` lấy từ tài khoản IAM đã tạo phần trước

**Bước 4:** Update file `index.ejs`, `index.css` theo mẫu sau:

```
JS index.js  index.js  index.css  package.json
views > index.js > html > body > div.content > table > ? > ?
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8" />
5   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6   <link rel="stylesheet" href="index.css" />
7 </head>
8 <body>
9   <div class="content">
10     <table border="1">
11       <tr>
12         <th>Ma SP</th>
13         <th>Ten SP</th>
14         <th>So Luong</th>
15         <th>Hinh anh</th>
16         <th>Chon</th>
17       </tr>
18       <% for (let i = 0; i < sanPhams.length; i++) { %>
19         <tr>
20           <td><%=sanPhams[i].ma_sp%></td>
21           <td><%=sanPhams[i].ten_sp%></td>
22           <td><%=sanPhams[i].so_luong%></td>
23         </tr>
24       <% } %>
25     </table>
26   </div>
27 </body>
28 </html>
```

```
JS index.js  index.js  index.css  package.json
views > index.css > h1
1 body {
2   width: 100%;
3   display: flex;
4   flex-direction: column;
5   justify-content: flex-start;
6   align-items: center;
7 }
8
9 .content {
10  width: 20%;
11  display: flex;
12  flex-direction: column;
13  justify-content: flex-start;
14  align-items: center;
15 }
16
17 .title {
18  width: 100%;
19  display: flex;
20  flex-direction: row;
21  justify-content: space-between;
22 }
23
24 table {
25  margin-top: 16px;
26  width: 100%;
27 }
28
29 th {
30  background-color: gray;
31 }
```

Kiểm tra lại giao diện tại localhost:3000



Bảng ở trên rỗng vì table *SanPham* trong *DynamoDB* chưa có dữ liệu.

**Bước 5:** Cài đặt thư viện *multer*: `npm install multer`

**Bước 6:** Update file *index.js*:

```
... JS index.js x index.js index.css package.json
JS index.js > ...
16 | region: 'ap-southeast-1'
17 | });
18 | AWS.config = config;
19 |
20 | const docClient = new AWS.DynamoDB.DocumentClient();
21 |
22 | const tableName = 'SanPham';
23 |
24 | const multer = require('multer');
25 |
26 | const upload = multer();
27 |
```

```

app.post('/', upload.fields([]), (req, res) => {
  const { ma_sp, ten_sp, so_luong } = req.body;

  const params = {
    TableName: tableName,
    Item: {
      "ma_sp": ma_sp,
      "ten_sp": ten_sp,
      "so_luong": so_luong
    }
  }

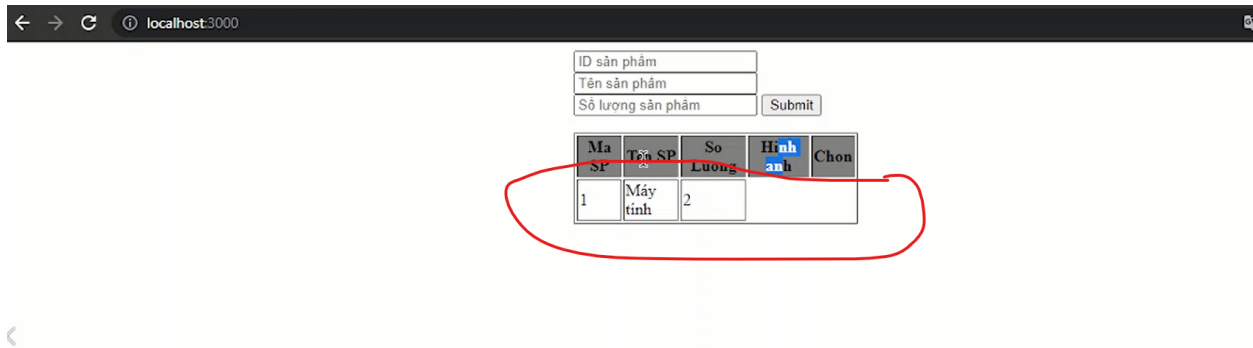
  docClient.put(params, (err, data) => {
    if (err) {
      return res.send('Internal Server Error');
    } else {
      return res.redirect("/");
    }
  })
});

```

**Bước 7:** Kiểm tra api bằng cách nhập thông tin sau vào form trên *localhost:3000* bấm *submit* và kiểm tra lại kết quả

The screenshot shows a web browser window with the address bar displaying 'localhost:3000'. The page contains a form with two input fields. The first field contains the number '1' and is labeled 'Máy tính' below it. The second field contains the number '2'. To the right of the second field is a 'Submit' button. Below the form is a table with 5 columns: 'Ma SP', 'Ten SP', 'So Luong', 'Hình ảnh', and 'Chọn'.

Ma SP	Ten SP	So Luong	Hình ảnh	Chọn



**Bước 8:** Update api xóa một item:

- Chỉnh sửa file *index.ejs* theo mẫu sau:

```
JS index.js • index.ejs x index.css package.json
views > index.ejs > html > body > div.content > form#delete
4 <meta charset="UTF-8" />
5 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6 <link rel="stylesheet" href="index.css" />
7 </head>
8 <body>
9 <div class="content">
10 <form id="create" method="POST" action="/" enctype="multipart/form-data">
11 <input type="number" name="ma_sp" placeholder="ID sản phẩm" />
12 <input type="text" name="ten_sp" placeholder="Tên sản phẩm" />
13 <input type="number" name="so_luong" placeholder="Số lượng sản phẩm" />
14 <input type="submit" value="Submit" />
15 </form>
16 <form id="delete" method="POST" action="/delete" enctype="multipart/form-data">
17 <div class="title"><h5>Danh sách sản phẩm</h5><input type="submit" value="Xóa" /></div>
18 <table border="1">
19 <tr>
20 <th>Ma SP</th>
21 <th>Ten SP</th>
22 <th>So Luong</th>
23 <th>Chon</th>
24 </tr>
25 <% for (let i = 0; i < sanPhams.length; i++) { %>
26 <tr>
27 <td>%=sanPhams[i].ma_sp%</td>
28 <td>%=sanPhams[i].ten_sp%</td>
29 <td>%=sanPhams[i].so_luong%</td>
30 <td><input type="checkbox" name="%=sanPhams[i].ma_sp%" /></td>
31 </tr>
32 <% } %>
33 </table>
34 </form>
35 </div>
36 </body>
37 </html>
```

- Cập nhật api xóa item trong file *index.js*:

```

62
63 app.post('/delete', upload.fields({}), (req, res) => {
64   const listItems = Object.keys(req.body);
65
66   if (listItems.length === 0) {
67     return res.redirect("/");
68   }
69
70   function onDeleteItem(index) {
71     const params = {
72       TableName: tableName,
73       Key: {
74         "ma_sp": listItems[index]
75       }
76     }
77
78     docClient.delete(params, (err, data) => {
79       if (err) {
80         return res.send('Internal Server Error');
81       } else {
82         if (index > 0) {
83           onDeleteItem(index - 1);
84         } else {
85           return res.redirect("/");
86         }
87       }
88     })
89   }
90
91   onDeleteItem(listItems.length - 1);
92 });
93

```

**Bước 9:** Kiểm tra lại api xóa bằng cách sau:

- Truy cập localhost:3000, check vào checkbox của 1 hoặc nhiều item trong bảng danh sách và bấm nút xóa.

ID sản phẩm  
 Tên sản phẩm  
 Số lượng sản phẩm

Danh sách sản phẩm

Mã SP	Tên SP	Số Lượng	Chon
2	Điện thoại	3	<input type="checkbox"/>
1	Máy tính	2	<input checked="" type="checkbox"/>

← → ↻ ⓘ localhost:3000

ID sản phẩm

Tên sản phẩm

Số lượng sản phẩm

Submit

Danh sách sản phẩm

Xóa

Mã SP	Tên SP	Số Lượng	Chon
2	Điện thoại	3	<input type="checkbox"/>

< >

- Nếu thấy item được chọn bị xóa khỏi danh sách tức là thành công.
- Có thể truy cập vào *DynamoDB* để kiểm tra item trong database.