

Tạo trang quản lý nhân viên sử dụng JQuery và JavaScript load dữ liệu API bằng Ajax có giao diện như hình dưới:



QLNV\_DNQ\_Script.js



QLNV AJAX.html

- Thực hiện các chức năng thêm, sửa xóa bằng cách lấy dữ liệu API từ Server Backend đẩy ra.

#### Quản lý nhân viên

ID:

1

Email:

daonq1@viettel.com.vn

Username:

daonq1

Fullname:

DaoNQ1

Select a Department:

Research and Development

Select a Position:

Scrum Master

Create Date:

05/29/2021

Save Update Reset

#### Thông tin nhân viên trên hệ thống

ID	Email	Username	Fullname	Department	Position	Create Date	Edit	Delete
1	daonq@viettel.com.vn	daonq	Full Name 1	Accounting	Developer	2021-05-27	Edit	Delete
2	daonq1@viettel.com.vn	daonq1	Full Name 2	Research and Development	Scrum Master	2021-05-29	Edit	Delete

- Dữ liệu API dạng:

```
[
  {
    "AccountID": "1",
    "Email": "daonq@viettel.com.vn",
    "Username": "daonq",
    "FullName": "NguyenDao",
    "Department": "Accounting",
    "Position": "Tester",
    "CreateDate": "2021-05-27"
  },
  {
    "AccountID": "2",
    "Email": "daonq1@viettel.com.vn",
    "Username": "daonq1",
    "FullName": "NguyenDao1",
    "Department": "Research and Development",
    "Position": "Developer",
    "CreateDate": "2021-05-29"
  }
]
```

#### 1. Tạo dữ liệu giả lập :

- Truy cập trang: <https://mockapi.io/>  
<https://60aef8dc5b8c300017deb66b.mockapi.io/accounts/Accounts>

- Tạo mới dữ liệu MockAPI giải lập: Dữ liệu này có các trường giống các trường trong bảng Account DB TestingSystem: **AccountID, Email, Username, FullName, Department, Position, CreateDate**

mock API

Docs

Projects → AccountAPI

API endpoint

https://60aef8dc5b8c300017deb66b.mockapi.io/accounts/:endpoint

NEW RESOURCE

GENERATE ALL

RESET ALL

Accounts

2

Schema (optional)

Define Resource schema, it will be used to generate mock data.

AccountID	Object ID
Email	String
Username	String
FullName	String
Department	String
Position	String
CreateDate	String
+	

Object template (optional)

In addition to resource schema you can define [Mustache template](#)

Template will only have access to one variable 'i' - index of current object

NOTE: template must be a valid JSON!

EXAMPLE:

```
{
  "name": "First{{i}} Last{{i}}",
  "email": "first{{i}}.last{{i}}@mail.com",
  "myArray": [100, 200, 300]
}
```

JSON template

## Endpoints

Enable/disable endpoints and customize response JSON. Custom response example:

```
{
  "items": "$mockData",
  "count": "$count",
  "anyKey": "anyValue"
}
```

**ON** GET → /Accounts

\$mockData

**ON** GET → /Accounts/:id

\$mockData

**ON** POST → /Accounts

\$mockData

**ON** PUT → /Accounts/:id

\$mockData

**ON** DELETE → /Accounts/:id

\$mockData

UPDATE

CANCEL

- Có thể sử dụng Fake-Data để Test dữ liệu trước.

Define Resource schema, it will be used to generate mock data.

AccountID

Object ID

Email

Faker.js

Email

Username

Faker.js

User name

FullName

Faker.js

First name

Department

Faker.js

Department

Position

Faker.js

City

CreateDate

String

+

	AccountID	Email	Username	FullName	DepartmentID	PositionID	CreateDate
▶	1	Email1@gmail.com	Username1	Fullname1	5	1	2020-03-05 00:00:00
	10	Email10@gmail.com	Username10	Fullname10	10	1	2020-04-09 00:00:00
	11	Email11@gmail.com	Username11	Fullname11	10	1	2021-05-27 08:42:37
	12	Email12@gmail.com	Username12	Fullname12	10	1	2021-05-27 08:42:37
	2	Email2@gmail.com	Username2	Fullname2	1	2	2020-03-05 00:00:00
	3	Email3@gmail.com	Username3	Fullname3	2	2	2020-03-07 00:00:00
	4	Email4@gmail.com	Username4	Fullname4	3	4	2020-03-08 00:00:00
	5	Email5@gmail.com	Username5	Fullname5	4	4	2020-03-10 00:00:00
	6	Email6@gmail.com	Username6	Fullname6	6	3	2020-04-05 00:00:00
	7	Email7@gmail.com	Username7	Fullname7	2	2	NULL
	8	Email8@gmail.com	Username8	Fullname8	8	1	2020-04-07 00:00:00
	9	Email9@gmail.com	Username9	Fullname9	2	2	2020-04-07 00:00:00
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## 2. Tạo trang QLVN AJAX.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Quản lý nhân viên With AJAX </title>
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.4.0/font/bootstrap-icons.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
</head>
<body>
  <div class="container">
    <div class="form">
      <h3>Quản lý nhân viên</h3>
      <form method="POST" id="Main_Form_ID">
        <div class="form-group">
          <label for="">ID: </label>
          <input type="number" required= "true" class="form-control" id="ID_ID" value="" placeholder="Enter Id">
        </div>
        <div class="form-group">
          <label for="">Email: </label>
          <input type="email" required= "true" class="form-control" id="Email_ID" value="" placeholder="Enter Email">
        </div>
        <div class="form-group">
          <label for="">Username: </label>
```

```

        <input type="text" required= "true" class="form-
control" id="Username_ID" value="" placeholder="Enter Username">
    </div>
    <div class="form-group">
        <label for="">Fullname: </label>
        <input type="text" required= "true" class="form-
control" id="Fullname_ID" value="" placeholder="Enter Fullname">
    </div>
    <div class="form-group">
        <label for="sel1">Select a Department:</label>
        <select class="form-control" id="Department_ID">
            <option>--Select a Department--</option>
            <option>Accounting</option>
            <option>Business Development</option>
            <option>Engineering</option>
            <option>Human Resources</option>
            <option>Research and Development</option>
        </select>
    </div>
    <div class="form-group">
        <label for="sel1">Select a Position:</label>
        <select class="form-control" id="Position_ID">
            <option>--Select a Position--</option>
            <option>Developer</option>
            <option>Tester</option>
            <option>Scrum Master</option>
            <option>PM</option>
        </select>
    </div>
    <div class="form-group">
        <label for="">Cretate Date: </label>
        <input type="date" required= "true" class="form-
control" id="Cretate_Date_ID" value="">
    </div>
    <div class="form-group">
        <input type="submit" class="btn btn-success" value="Save">
        <input type="button" class="btn btn-success" id="update_btn" value="Update">
        <input type="button" class="btn btn-success" id="reset_btn" value="Reset">
    </div>
</form>
</div>

<div class="container">
    <div class="form">
        <h3>Thông tin nhân viên trên hệ thống</h3>
        <table class="table table-hover" id="Detail_Staff_ID">
            <thead>
                <tr>

```

```

        <th>ID</th>
        <th>Email</th>
        <th>Username</th>
        <th>Fullname</th>
        <th>Department</th>
        <th>Position</th>
        <th>Cretate Date</th>
        <th>Edit</th>
        <th>Delete</th>
    </tr>
</thead>
<tbody id="Result_TB">

</tbody>
</table>
</div>
</div>

<script type="text/javascript" src="QLNV_DNQ_Script.js"></script>
</body>
</html>

```

### 3. Tạo trang QLVN\_DNQ\_Script.js để viết JQuery, AJAX

```

// Trong bài tập này sẽ sử dụng JQuery và JavaScript, JavaScript để thực hiện chức năng quản lý nhân
viên, tương ứng với bảng Account trong DB TestingSystem, Các chức năng: Thêm, Sửa, Xóa. Dữ liệu này
sẽ tương tác với server Backend API.
// Khai báo 1 array để lưu thông tin tất cả các Account trên hệ thống.
var listAccount = []
$(function () {
    // Gọi hàm load dữ liệu từ Server API khi load trang
    getListEmployees();

    // Hàm thực thi khi load đầy đủ các thành phần html
    $('#reset_btn').click(function () {
        // Xử lý sự kiện cho nút reset, Sử dụng JQuery để lấy các giá trị các trường theo ID, sau đó
        Set về rỗng
        $('#ID_ID').val("")
        $('#Email_ID').val("")
        $('#Username_ID').val("")
        $('#Fullname_ID').val("")
        $('#Department_ID').val("")
        $('#Position_ID').val("")
        $('#Cretate_Date_ID').val("")
    })

    // Xử lý sự kiện khi nhấn nút Save
    $('#Main_Form_ID').submit(function () {

```

```

        // Xử lý sự kiện khi nhấn nút Submit của Form(Save), ở đây phải sử dụng Return False ở cuối
sự kiện này để không redirect sang trang mới.
        // Lấy các giá trị người dùng nhập vào
        var v_ID_ID = $('#ID_ID').val()
        var v_Email_ID = $('#Email_ID').val()
        var v_Username_ID = $('#Username_ID').val()
        var v_Fullname_ID = $('#Fullname_ID').val()
        var v_Department_ID = $('#Department_ID').val()
        var v_Position_ID = $('#Position_ID').val()
        var v_Cretate_Date_ID = $('#Cretate_Date_ID').val()

        // Tạo 1 đối tượng account để lưu giữ thông tin nhận được
        var account = {
            'Email': v_Email_ID,
            'Username': v_Username_ID,
            'FullName': v_Fullname_ID,
            'Department': v_Department_ID,
            'Position': v_Position_ID,
            'CreateDate': v_Cretate_Date_ID
        }
        // Add account tới MOCK API
        $.post("https://60aef8dc5b8c300017deb66b.mockapi.io/accounts/Accounts", account,
            function (data, status) {
                // error
                if (status == "error") {
                    alert("Error when loading data");
                    return;
                }
                // success
                // Hàm này để hiển thị thông tin account ở table
                getListEmployees();
            });
        return false;
        // Sử dụng return false để không redirect tới 1 trang khác.
    })
})

// Đoạn bên trên khi load hoàn chỉnh trang web sẽ kết thúc ở đây, đoạn dưới sẽ viết các hàm hỗ trợ k
hác
// Viết hàm showAccount()
function showAccount() {
    // Xóa hết kết quả đang hiển thị ở bảng kết quả
    $('#Result_TB').empty()
    // Lặp trong listAccount để in thông tin từng phần tử
    // Hiển thị thêm 2 nút để sửa và xóa các Account
    for (var index = 0; index < listAccount.length; index++) {
        $('#Result_TB').append(`
            <tr>
                <th>${listAccount[index].AccountID}</th>

```

```

        <th>${listAccount[index].Email}</th>
        <th>${listAccount[index].Username}</th>
        <th>${listAccount[index].FullName}</th>
        <th>${listAccount[index].Department}</th>
        <th>${listAccount[index].Position}</th>
        <th>${listAccount[index].CreateDate}</th>
        <th><button class="btn btn-warning" onclick="editAccount(${index})">Edit</button></th>
        <th><button class="btn btn-
warning" onclick="deleteAccount(${index})">Delete</button></th>
    </tr>
    `)
}
}
// Viết hàm xóa account
function deleteAccount(Index) {
    var v_del_ID = listAccount[Index].AccountID;
    // Hiển thị 1 Confirm Popup, chọn Có = True
    var confirm_del = confirm('Bạn có chắc chắn muốn xóa Account này không')
    if (confirm_del) {
        $.ajax({
            url: 'https://60aef8dc5b8c300017deb66b.mockapi.io/accounts/Accounts/' + v_del_ID,
            type: 'DELETE',
            success: function (result) {
                // error
                if (result == undefined || result == null) {
                    alert("Error when loading data");
                    return;
                }
                // success
                getListEmployees();
            }
        });
    } else {
        return
    }
}
// Viết hàm để Edit các account
function editAccount(Index) {
    $('#ID_ID').val(listAccount[Index].AccountID)
    $('#Email_ID').val(listAccount[Index].Email)
    $('#Username_ID').val(listAccount[Index].Username)
    $('#Fullname_ID').val(listAccount[Index].FullName)
    $('#Department_ID').val(listAccount[Index].Department)
    $('#Position_ID').val(listAccount[Index].Position)
    $('#Cretate_Date_ID').val(listAccount[Index].CreateDate)

    // Xử lý sự kiện khi click vào nút Update

```



```

$('#update_btn').click(function () {
    var v_ID_ID = $('#ID_ID').val()
    var v_Email_ID = $('#Email_ID').val()
    var v_Username_ID = $('#Username_ID').val()
    var v_Fullname_ID = $('#Fullname_ID').val()
    var v_Department_ID = $('#Department_ID').val()
    var v_Position_ID = $('#Position_ID').val()
    var v_Cretate_Date_ID = $('#Cretate_Date_ID').val()

    var account = {
        'AccountID': v_ID_ID,
        'Email': v_Email_ID,
        'Username': v_Username_ID,
        'FullName': v_Fullname_ID,
        'Department': v_Department_ID,
        'Position': v_Position_ID,
        'CreateDate': v_Cretate_Date_ID,
    }

    $.ajax({
        url: 'https://60aef8dc5b8c300017deb66b.mockapi.io/accounts/Accounts/' + v_ID_ID,
        type: 'PUT',
        data: account,
        success: function (result) {
            // error
            if (result == undefined || result == null) {
                alert("Error when loading data");
                return;
            }
            getListEmployees();
        }
    });
})
}

// Viết hàm gọi dữ liệu server API bằng Ajax khi load trang Web
function getListEmployees() {
    // call API from server
    $.get("https://60aef8dc5b8c300017deb66b.mockapi.io/accounts/Accounts", function (data, status) {
        // reset list employees
        listAccount = [];
        // error
        if (status == "error") {
            // TODO
            alert("Error when loading data");
            return;
        }
        // success
        parseData(data);
    });
}

```

```
        showAccount();
    });
}
// hàm chuyển dữ liệu API nhận được bằng Ajax chuyển về dạng List để sử dụng trong chương trình
function parseData(data) {
    // employees = data;
    data.forEach(function (item) {
        var account = {
            'AccountID': item.AccountID,
            'Email': item.Email,
            'Username': item.Username,
            'FullName': item.FullName,
            'Department': item.Department,
            'Position': item.Position,
            'CreateDate': item.CreateDate,
        }
        listAccount.push(account)
    });
}
```