



FPT POLYTECHNIC



THỰC HỌC – THỰC NGHIỆP



Conceive Design Implement Operate

NODEJS & RESFUL WEB SERVICE

RESTFUL API VÀ FRONTEND

- ⊙ Hiểu được cơ chế kết hợp giữa front end và backend
- ⊙ Sử dụng AJAX kết nối RESTFUL API
- ⊙ Sử dụng framework front-end với nodejs và sử dụng RestFul API



- 📖 Mô hình kết nối giữa frontend và backend
- 📖 Xây dựng FrontEnd với Ajax
- 📖 Xây dựng FontEnd bằng AngularJS



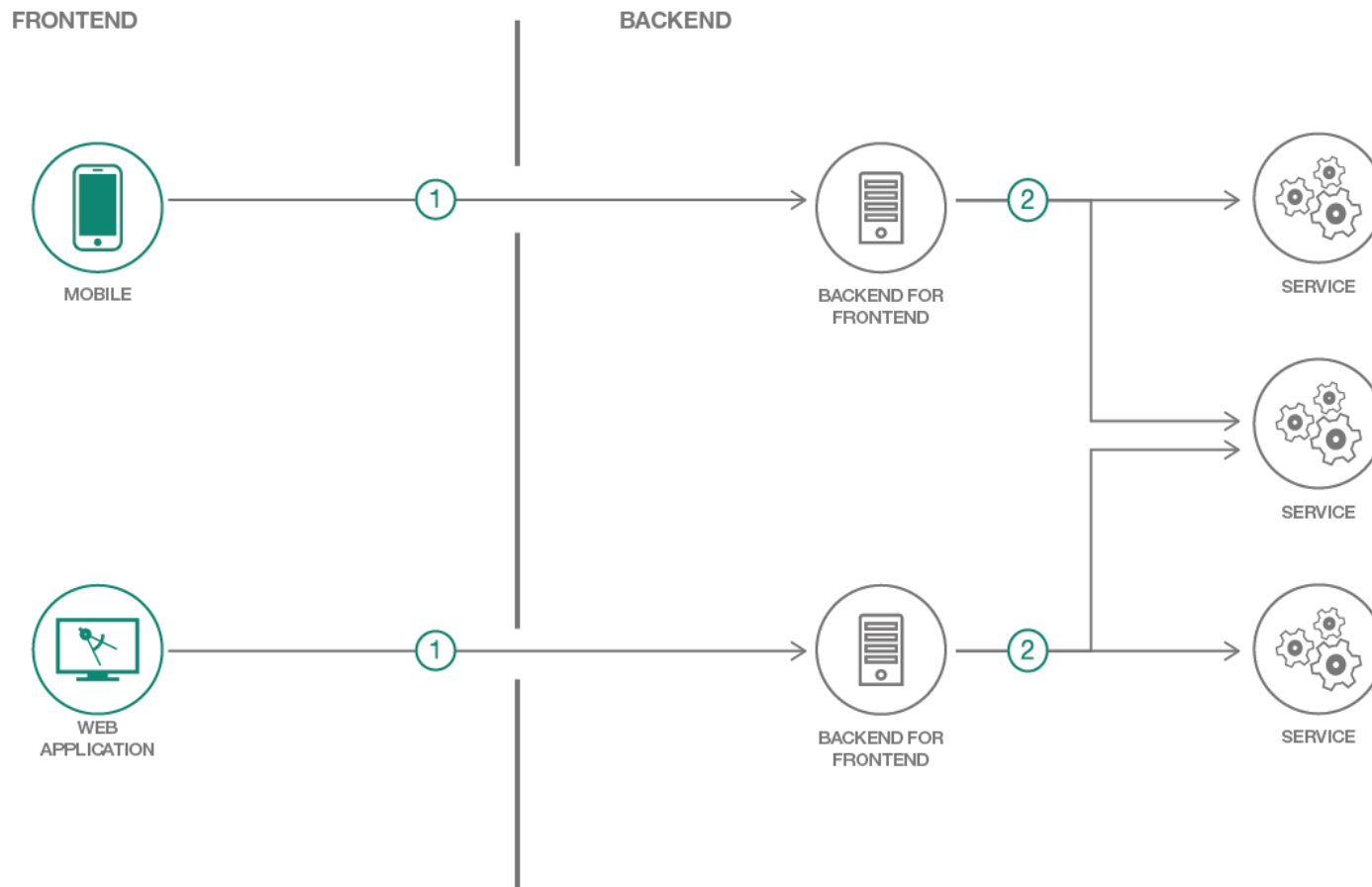


PHẦN 1: FRONT-END VỚI NODEJS

- ❑ Một ứng dụng web, mobile,... tương tác với người dùng.
- ❑ Xử lý tại client bằng javascript
- ❑ Gửi yêu cầu đến Backend với giao thức HTTP.

- ❑ Giúp phần front-end của một trang web có thể hoạt động được.
- ❑ Là Ứng dụng xử lý tại server như thao tác với database
- ❑ Phản hồi yêu cầu của front end qua giao thức HTTP.

Mô hình kết nối giữa FrontEnd và BackEnd



AJAX là gì?

- AJAX là chữ viết tắt của Asynchronous JavaScript and XML.
- Là một bộ các kỹ thuật thiết kế web giúp cho các ứng dụng web hoạt động bất đồng bộ – xử lý mọi yêu cầu tới server từ phía sau

Mô hình AJAX?

1. Trình duyệt tạo một lệnh gọi JavaScript để kích hoạt XMLHttpRequest.
2. Ở dưới nền, trình duyệt tạo một yêu cầu HTTP gửi lên server.
3. Server tiếp nhận, truy xuất và gửi lại dữ liệu cho trình duyệt.
4. Trình duyệt nhận dữ liệu từ server và ngay lập tức hiển thị lên trang. Không cần tải lại toàn bộ trang..

Gửi yêu cầu tới server

Để gửi yêu cầu tới server, chúng ta dùng hàm `open()` và `send()` của đối tượng `XMLHttpRequest`

```
xhttp.open(Method, url, true);
```

```
xhttp.send(); hoặc xhttp.send(string)
```

Ví dụ: Gửi yêu cầu nhận về tất cả các bài viết hiện hành

```
xhttp.open("GET", "localhost:3000/blog/posts", true);
```

```
xhttp.send();
```

Nhận phản hồi từ server

Để nhận phản hồi từ server trả về, chúng ta dùng 1 trong 2 thuộc tính sau

`xhttp.onreadystatechange`

`request.onload`

Ví dụ: Nhận về tất cả các bài viết hiện hành

```
request.onload=function(){  
    if (this.readyState == 4 && this.status == 200) {  
        var data=JSON.parse(request.responseText);  
        console.log(data);  
    }  
}
```

Để RESTFUL API NodeJS cho phép FrontEnd truy cập, ta bổ sung thêm quyền cho kết quả trả về (response)

```
app.use((req, res, next) => {  
  res.setHeader('Access-Control-Allow-Origin', '*');  
  res.setHeader('Access-Control-Allow-Methods', 'OPTIONS, GET, POST, PUT, PATCH,  
  DELETE');  
  res.setHeader('Access-Control-Allow-Headers', 'Content-Type, Authorization');  
  next();  
});
```

Gửi yêu cầu tới server với GET

```
xhttp.open("GET", url, true);
```

```
xhttp.send();
```

```
request.onreadystatechange = function() {  
    if (request.readyState == XMLHttpRequest.DONE) {  
        if (request.status == 200){  
            // success START  
            console.log(request.responseText);  
        }  
        else {  
            console.log("error");  
        }  
    }  
};
```

Nhận phản hồi bang thuộc tính responseText

Gửi yêu cầu tới server với POST

```
xhttp.open("POST", url, true);  
xhttp.send(data);
```

- Để POST dữ liệu như form, ta thêm HTTP header với hàm `setRequestHeader()`
- Cung cấp dữ liệu được post khi gọi hàm `send()`.

```
request.setRequestHeader("Content-Type","application/json; charset=UTF-8");
```

```
var data = JSON.stringify({"title": document.querySelector('#title').value,  
"content": document.querySelector('#content').value});
```

demo



PHẦN 2: FRONTEND VỚI ANGULARJS

- AngularJS là một framework viết bằng javascript
- AngularJS hoạt động bằng cách sử dụng các thuộc tính mở rộng cho thẻ html(bổ sung) và gắn kết dữ liệu bằng biểu thức
- Sử dụng bằng cách thêm vào trang web với thẻ <script>

- Các Directive thường dùng trong angular
 - ✓ Ng-app: chỉ định vị trí bắt đầu sử dụng angularjs trong ứng dụng
 - ✓ Ng-controller: chỉ định vị trí sử dụng dữ liệu trong ứng dụng
 - ✓ Ng-model: ánh xạ dữ liệu từ các control trên form tới biến trong ứng dụng

TỔNG QUAN VỀ SỬ DỤNG ANGULARJS

```

<!DOCTYPE html>
<html>
  <head>
    <script src="js/angular.min.js"></script>
    <title>FrontEnd&NodeJS </title>
  </head>
  <body [ng-app="myApp"]>
    <h1>Tasks</h1>
    <main[ng-controller="myController"]>
      <h2>welcome: {{name}}</h2>
    </main>
  </body>
  <script>
    [
      var app=angular.module("myApp", []);
      app.controller("myController", function($scope){
        $scope.name="FPT Poly Technic";
      });
    ]
  </script>
</html>
  
```

- Thực hiện Lặp một thẻ html
- Dữ liệu lặp thường là mảng json

```
<ul>
  <li ng-repeat="x in names">
    {{ x.name + ', ' + x.country }}
  </li>
</ul>
```

```
<script>
var app=angular.module("myApp",[]);
app.controller("myController",function($scope){
  $scope.name = [
    {name: 'Jani',country: 'Norway'},
    {name: 'Hege',country: 'Sweden'},
    {name: 'Kai',country: 'Denmark'}];
  });
</script>
```

- Thực hiện ánh xạ 2 chiều giữa control trên form tới biến

```
<div ng-app="myApp" ng-controller="myCtrl">  
  Full name: <input ng-model="fullName">  
</div>
```

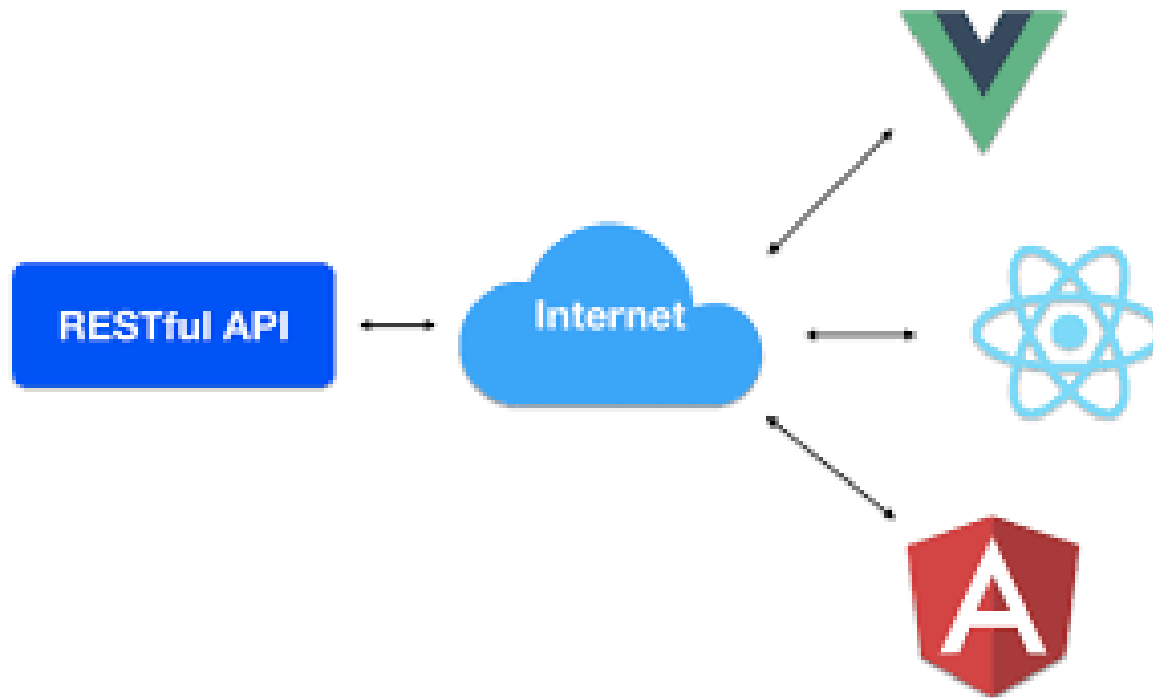
```
<script>  
var app = angular.module('myApp', []);  
app.controller('myCtrl', function($scope) {  
  $scope.fullName = "Nguyễn Văn Tào";  
});  
</script>
```

- Thực hiện ánh xạ 2 chiều giữa control trên form tới biến

```
<div ng-app="myApp" ng-controller="myCtrl">  
  Full name: <input ng-model="fullName">  
</div>
```

```
<script>  
var app = angular.module('myApp', []);  
app.controller('myCtrl', function($scope) {  
  $scope.fullName = "Nguyễn Văn Tào";  
});  
</script>
```

- **\$http** là một dịch vụ của AngularJS để đọc dữ liệu từ servers.



- Cách dùng http của angularjs.

```
<div ng-app="myApp" ng-controller="myCtrl">
  Full name: <input ng-model="fullName">
</div>

<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope, $http) {
  $http.method("URL")
  .then(function(response) {
    $scope.biến = response.data;
  });
});
</script>
```


- Các phương thức của http service.

.delete()

.get()

.head()

.jsonp()

.patch()

.post()

.put()

- Phương thức GET

```
<div ng-app="myApp" ng-controller="myCtrl">
<ul>
  <li ng-repeat="c in data">
    {{ c.Name + ', ' + c.Country }}
  </li>
</ul>

</div>

<script>

var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope, $http) {
  $http.get("localhost:3000/customers").then(function(response) {
    $scope.data = response.data.records;
  });
});

</script>
```

- Phương thức POST

```
<div ng-app="myApp" ng-controller="myCtrl">
<ul>
  <li ng-repeat="c in data">
    {{ c.Name + ', ' + c.Country }}
  </li>
</ul>
</div>
<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope, $http) {
  $http.post("localhost:3000/customer", JSON.stringify(data))
    .then(function(response) {
      $scope.data = response.data;
      $scope.statusval = response.status;
      $scope.statustext = response.statusText;
    });
});
</script>
```

demo

- ☑ Mô hình kết nối giữa frontend và backend
- ☑ Xây dựng FrontEnd với Ajax
- ☑ Xây dựng FontEnd bằng AngularJS



thank
you!