



Bài 29

AJAX

Module: WBD

Mục tiêu

- Giải thích được cơ chế hoạt động của AJAX
- Trình bày được các tình huống sử dụng của AJAX
- Triển khai được AJAX
- Triển khai được AJAX sử dụng jQuery
- Triển khai được AJAX với các phương thức POST, GET



Ajax

Ajax là gì

Cơ chế hoạt động của Ajax

Ajax là gì

- AJAX – **A**synchronous **J**avaScript and **X**ML, là một nhóm các công nghệ phát triển web được sử dụng để tạo các ứng dụng web giàu tính tương tác, nhanh hơn và mượt mà hơn với sự giúp đỡ của XML, HTML, CSS và JavaScript.
- AJAX cho phép các trang web được cập nhật một cách không đồng bộ bằng cách trao đổi các lượng dữ liệu nhỏ với Server mà không cần phải tải lại.
- Với AJAX, người dùng có thể tiếp tục sử dụng ứng dụng trong khi các chương trình trên Client tạo Request để lấy thông tin từ Server.

Một số ví dụ sử dụng Ajax



- Google Map
- Google Suggest
- Gmail



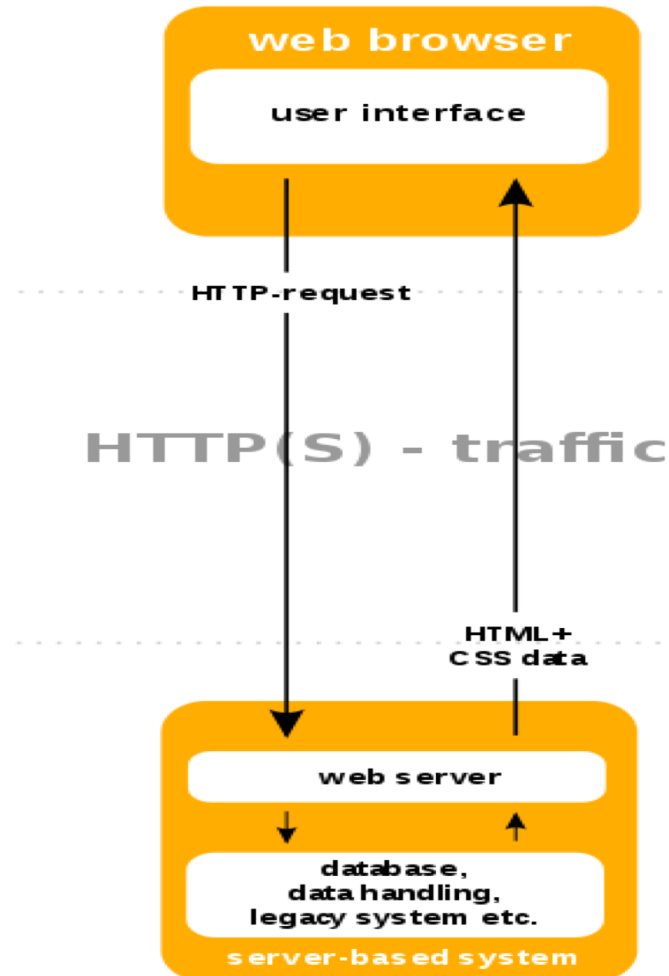
Các thành phần cơ bản trong Ajax

- HTML & CSS trong việc hiển thị thông tin
- JavaScript
- Document Object Model (DOM)
 - Tương tác với thông tin được hiển thị thông qua JavaScript
- Đối tượng XMLHttpRequest
 - Trao đổi dữ liệu một cách không đồng bộ với máy chủ.
- XML hoặc JSON
 - Định dạng cho dữ liệu truyền

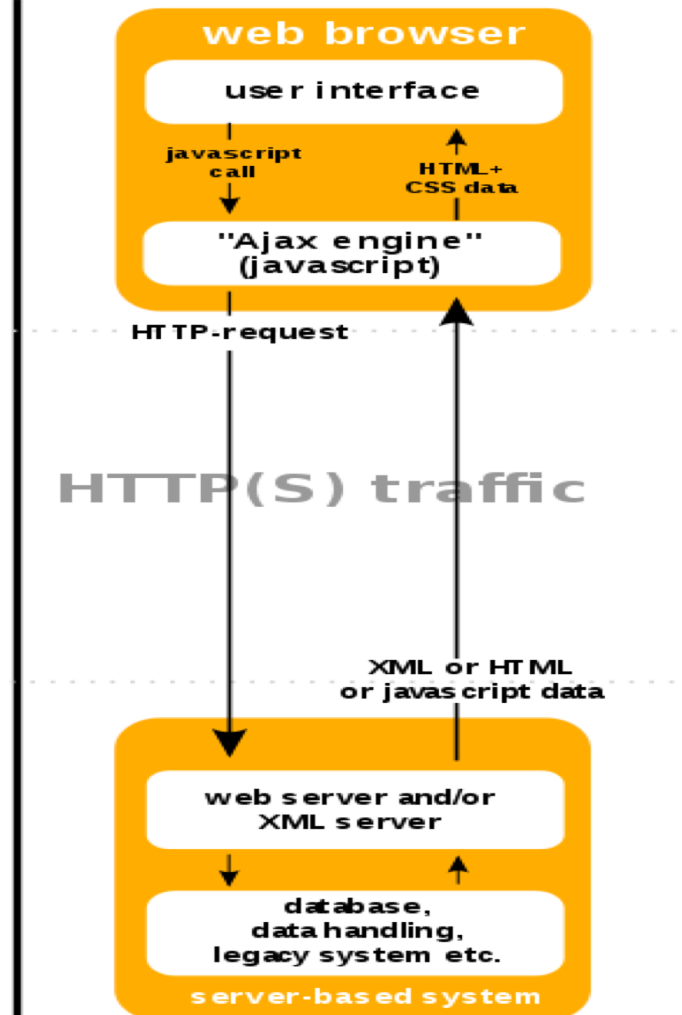
Cơ chế hoạt động



Conventional model of a web application



Ajax model of a web application



Nhược điểm Ajax



- Các trang web được tạo động không được ghi vào bộ lưu lịch sử lướt web của trình duyệt, do đó nút "back" của trình duyệt sẽ mất tác dụng.
- Việc cập nhật các trang web động cũng gây khó khăn cho người dùng trong việc bookmark một trạng thái nào đó của ứng dụng.
- Bất kỳ người dùng nào có trình duyệt không hỗ trợ Ajax hay JavaScript, hoặc đơn giản là đã bị vô hiệu hóa JavaScript, sẽ đương nhiên không thể sử dụng Ajax.
- Tấn công của các đoạn mã độc mà những nhà phát triển web có thể không kiểm thử hết được

Ưu điểm Ajax



- Một ứng dụng web có thể chỉ yêu cầu cho các nội dung cần thiết phải cập nhật, do đó giảm lượng lớn băng thông và thời gian nạp trang.
- Việc dùng các yêu cầu không đồng bộ (*asynchronous request*) cho phép giao diện người dùng của ứng dụng hiển thị trên trình duyệt giúp người dùng trải nghiệm sự tương tác cao, với nhiều phần riêng lẻ.
- Việc sử dụng Ajax có thể làm giảm các kết nối đến server, do các mã kịch bản (*script*) và các style sheet chỉ phải yêu cầu một lần.

Triển khai Ajax sử dụng Jquery trong Spring

Cấu hình thư viện



- Thêm dependency vào file gradle.build

compile **group: 'com.fasterxml.jackson.core', name: 'jackson-core', version: '2.9.5'**

compile **group: 'com.fasterxml.jackson.core', name: 'jackson-databind', version: '2.9.5'**

Cấu hình thư viện



- Thư viện sử dụng ajax vào file HTML

```
<script  
src="https://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js">  
</script>
```

- \$.ajax({}) là hàm được cung cấp bởi jquery để tương tác ajax với server. Hàm này nhận một đối tượng gồm các thuộc tính thường dung là url, success, dataType, type...

- Cú pháp:
\$.ajax({
 url: <địa chỉ trang web cần tương tác>,
 data:{<danh sách tham số gửi đến server>},
 type: <phương thức web yêu cầu>,
 dataType: <loại dữ liệu nhận được từ server>
 success: function(response){<xử lý kết quả trả về>}
});

Kiểu dữ liệu được trả về từ server

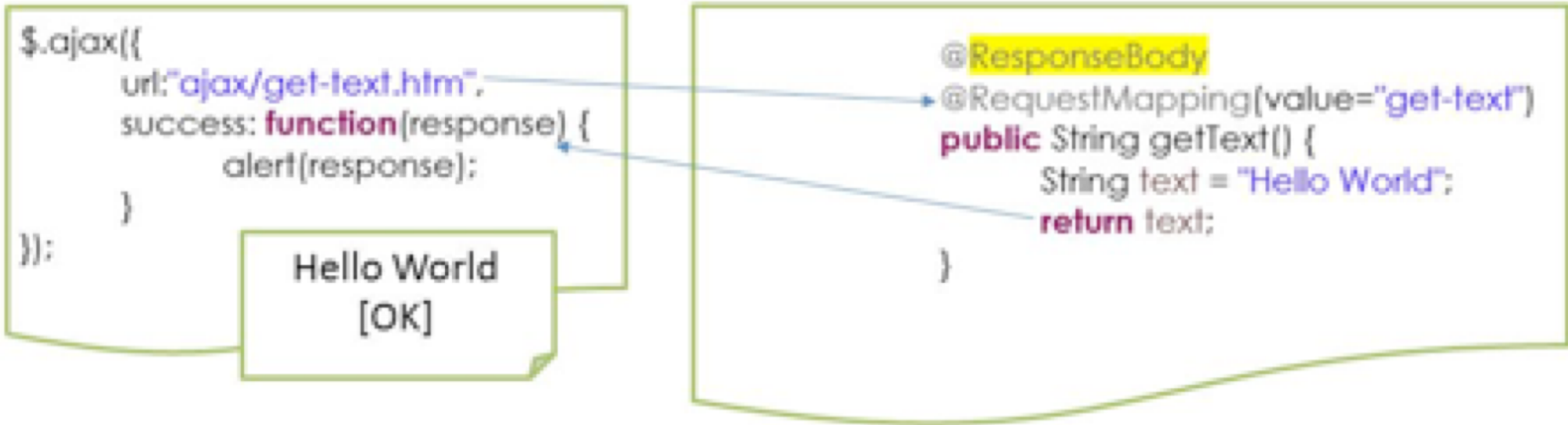


- Với kỹ thuật ajax client chỉ quan tâm đến loại dữ liệu trả về từ server mà không quan tâm đến server sẽ làm công việc gì.
- Các loại dữ liệu được trả về từ server gồm
 - Văn bản hoặc HTML
 - Đối tượng JSON

Xử lý text/html



- Yêu cầu ajax có thể gửi đến các action get-text.htm hoặc get-Html.htm để nhận được text hoặc html. Kết quả nhận được sẽ được thông báo bằng alert()



Xử lý đối tượng JSON



```
$(document).ready(function() {  
    $('#newSmartphoneForm').submit(function(event) {  
        var producer = $('#producer').val();  
        var model = $('#model').val();  
        var price = $('#price').val();  
        var json = { "producer" : producer, "model" : model, "price": price};  
        $.ajax({  
            headers: {  
                'Accept': 'application/json',  
                'Content-Type': 'application/json'  
            },  
            type : "POST",  
            data: JSON.stringify(json),  
            url: "/smartphones/createnewPhone",  
            success: function(smartphone) {  
                var respContent = "";  
                respContent += "<span class='success'>Smartphone was created: [";  
                respContent += smartphone.producer + " : ";  
                respContent += smartphone.model + " : " ;  
                respContent += smartphone.price + "]"</span>"  
                $('#sPhoneFromResponse').html(respContent);  
            }  
        });  
        event.preventDefault();  
    });  
});
```

```
@RequestMapping(value = "/create", method = RequestMethod.GET)  
public ModelAndView createSmartphonePage() {  
    ModelAndView mav = new ModelAndView("phones/new-phone");  
    mav.addObject("sPhone", new Smartphone());  
    return mav;  
}  
  
@RequestMapping(value = "/createnewPhone", method = RequestMethod.POST,  
    produces = MediaType.APPLICATION_JSON_VALUE, consumes = MediaType.APPLICATION_JSON_VALUE)  
@ResponseBody  
public Smartphone createSmartphone(@RequestBody Smartphone smartphone) {  
    return smartphoneService.save(smartphone);  
}
```

Demo: Ứng dụng thêm quản lý điện thoại



[Thực hành] Quản lý điện thoại



[Bài tập] Tích hợp AJAX cho ứng dụng Blog



Tổng kết

