



KỸ THUẬT LẬP TRÌNH AJAX

ThS. Nguyễn Nghiệm
0913.745.789 - NghiemN@fpt.edu.vn



- Giới thiệu Ajax
- Kỹ thuật lập trình phía server
 - ✱ Trả về văn bản
 - ✱ Trả về chuỗi đối tượng json
 - Chuyển đổi map thành đối tượng json
 - Chuyển đổi object thành đối tượng json
 - ✱ Trả về mảng đối tượng json
 - Chuyển đổi list thành mảng đối tượng json
- Kỹ thuật lập trình phía client
 - ✱ Xử lý kết quả là văn bản
 - ✱ Xử lý kết quả là đối tượng json
 - ✱ Xử lý kết quả là mảng đối tượng json





AJAX LÀ GÌ ?

● AJAX (Asynchronous Javascript and XML)

- ✱ Sử dụng javascript truyền thông bất đồng bộ với server
- ✱ Dữ liệu nhận được có thể là
 - Text, HTML, XML, JSON

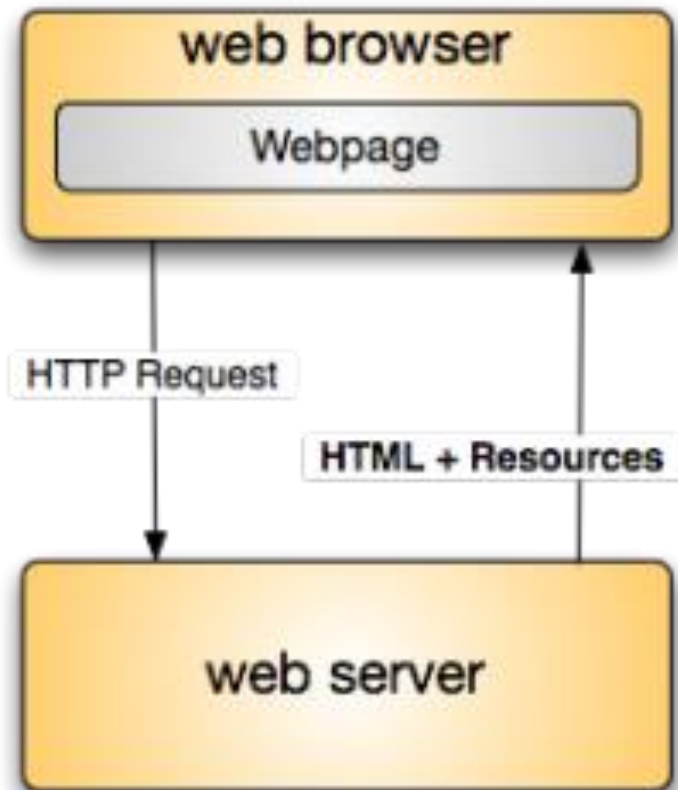
● Lợi ích

- ✱ Tương tác nhanh với server (không tải lại toàn trang)
- ✱ Dễ học (sử dụng công nghệ sẵn có)
- ✱ Được hỗ trợ nhiều trình duyệt
- ✱ Có nhiều framework mã nguồn mở
- ✱ Cộng đồng sử dụng rất lớn -> dễ tìm tài liệu và hỗ trợ

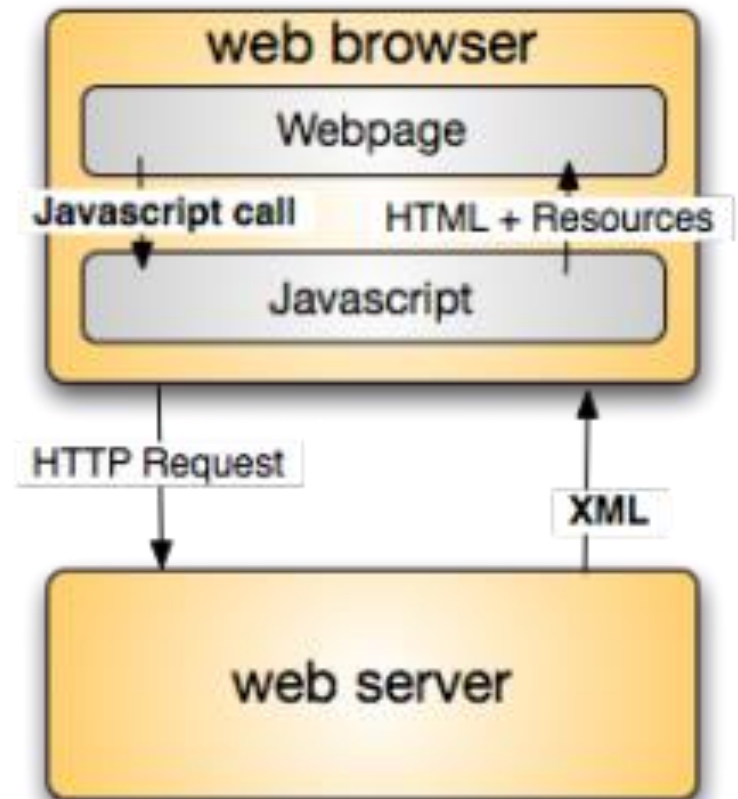


AJAX MODEL

Traditional web model



AJAX web model





KHỞI ĐỘNG AJAX VỚI JQUERY

```
$.ajax({  
  url: "ajax/hello.htm",  
  success: function(response) {  
    alert(response);  
  }  
});
```

```
@RequestMapping(value="ajax/hello")  
public String getView() {  
    return "view";  
}
```

The page at localhost:8080 says:

<h1>Hello Ajax World</h1>

OK

```
<%@ page pageEncoding="utf-8"%>  
<h1>Hello Ajax World</h1>
```



CÁC THUỘC TÍNH CỦA \$.AJAX({})

- **url** đưa ra địa chỉ trang web cần tương tác
 - ✱ **url:** "ajax/hello.htm"
- **success** định nghĩa hàm xử lý kết quả trả về từ server
 - ✱ **success:** function(response){}
- **data** cung cấp các tham số gửi đến server
 - ✱ **data:** {id: "nghiemn", password: "iloveyou"}
- **type** chỉ ra method
 - ✱ **type:** "POST"
- **dataType** đưa ra kiểu dữ liệu nhận từ server
 - ✱ **dataType:** "json"

Kỹ thuật lập trình phía server
với

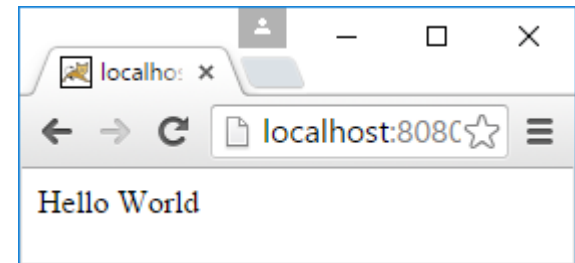
Spring MVC



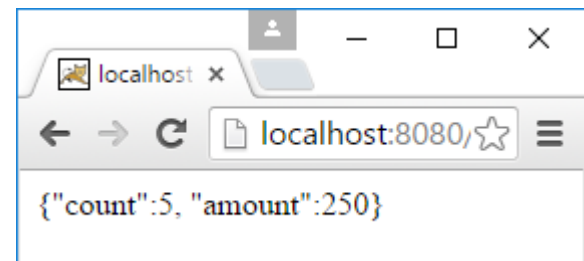
@ResponseBody

- Action được chú thích với **@ResponseBody** sẽ trả về kết quả là nội dung của **return** thay vì kết quả của **view**

```
@ResponseBody
@RequestMapping(value="get-text")
public String getText() {
    String text = "Hello World";
    return text;
}
```



```
@ResponseBody
@RequestMapping(value="get-json")
public String getJson() {
    String text = "{\"count\":5, \"amount\":250}";
    return text;
}
```





OBJECTMAPPER

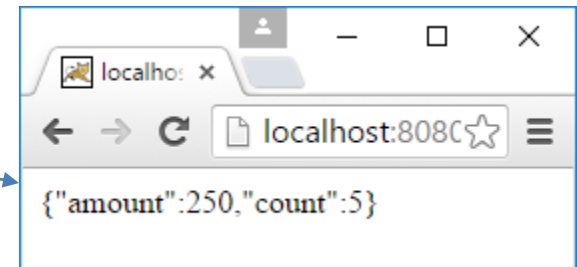
- ObjectMapper là lớp cho phép chuyển đổi giữa đối tượng java và đối tượng json
 - ✱ Map => đối tượng json
 - {"count":5, "amount":250}
 - ✱ Bean => đối tượng json
 - {"count":5, "amount":250}
 - ✱ List<Object> => mảng đối tượng json
 - [{"count":5, "amount":250}, {"count":5, "amount":250}]



MAP => JSON OBJECT

```
@ResponseBody
@RequestMapping(value="get-map")
public String getMap() {
    Map<String, Object> map = new HashMap<String, Object>();
    map.put("count", 5);
    map.put("amount", 250);

    try {
        ObjectMapper mapper = new ObjectMapper();
        String text = mapper.writeValueAsString(map);
        return text;
    }
    catch (Exception e) {
        return "error";
    }
}
```



Chuyển Map thành đối tượng json
+ Key thành tên thuộc tính
+ Value thành giá trị thuộc tính



JAVABEAN => JSON OBJECT

@ResponseBody

@RequestMapping(value="get-object")

```
public String getObject() {  
    CartInfo cart = new CartInfo();  
    cart.setCount(5);  
    cart.setAmount(250.0);
```

```
    try {
```

```
        ObjectMapper mapper = new ObjectMapper();  
        String text = mapper.writeValueAsString(cart);  
        return text;
```

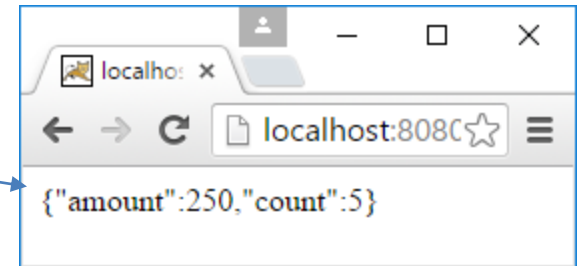
```
    }
```

```
    catch (Exception e) {  
        return "error";
```

```
    }
```

```
}
```

```
public class CartInfo {  
    Integer count;  
    Double amount;  
    getters/setters  
}
```



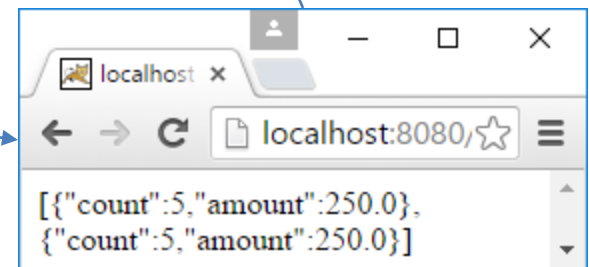
Chuyển đổi tượng java bean
thành đối tượng json với
+ Giữ nguyên tên thuộc tính
+ Giữ nguyên giá trị thuộc tính



LIST<OBJECT> ==> JSON ARRAY

```
@ResponseBody
@RequestMapping(value="get-list")
public String getList() {
    CartInfo cart1 = new CartInfo();
    cart1.setCount(5);
    cart1.setAmount(250.0);
    CartInfo cart2 = new CartInfo();
    cart2.setCount(5);
    cart2.setAmount(250.0);
    List<CartInfo> list = new ArrayList<CartInfo>();
    list.add(cart1);
    list.add(cart2);
    try {
        ObjectMapper mapper = new ObjectMapper();
        String text = mapper.writeValueAsString(list);
        return text;
    }
    catch (Exception e) {
        return "error";
    }
}
```

Chuyển đổi tập tượng java bean
thành mảng đối tượng json với



Kỹ thuật lập trình phía client với jQuery



AJAX

```
$.ajax({  
  url:"ajax/get-text.htm",  
  success: function(response) {  
    alert(response);  
  }  
});
```

Hello World
[OK]

```
@ResponseBody  
@RequestMapping(value="get-text")  
public String getText() {  
  String text = "Hello World";  
  return text;  
}
```

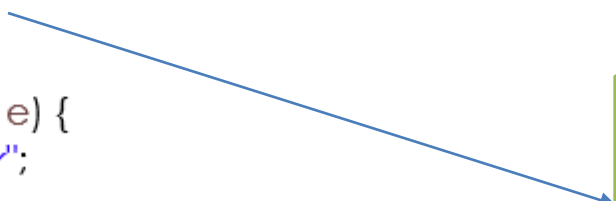
```
$.ajax({  
  url:"ajax/get-json.htm",  
  dataType:"json",  
  success: function(response) {  
    alert(response.count);  
  }  
});
```

5
[OK]

```
@ResponseBody  
@RequestMapping(value="get-json")  
public String getJson() {  
  String text = "{\\"count\\":5, \\"amount\\":250}";  
  return text;  
}
```



```
@ResponseBody
@RequestMapping(value="get-list")
public String getList() {
    CartInfo cart1 = new CartInfo();
    cart1.setCount(5);
    cart1.setAmount(250.0);
    CartInfo cart2 = new CartInfo();
    cart2.setCount(5);
    cart2.setAmount(250.0);
    List<CartInfo> list = new ArrayList<CartInfo>();
    list.add(cart1);
    list.add(cart2);
    try {
        ObjectMapper mapper = new ObjectMapper();
        String text = mapper.writeValueAsString(list);
        return text;
    }
    catch (Exception e) {
        return "error";
    }
}
```



```
[{"count":5, "amount":250},
{"count":5, "amount":250}]
```



XỬ LÝ MẢNG ĐỐI TƯỢNG

```
$.ajax({  
  url:"ajax/get-list.htm",  
  dataType:"json",  
  success: function(response) {  
    $(response).each(function(index, item){  
      alert(index + "=" + item.count)  
    });  
  }  
});
```

[{"count":5, "amount":250},
{"count":5, "amount":250}]

5
[OK]

10
[OK]

- `$(response).each(function(index, item))` sẽ duyệt mảng response.

- ✱ **Index**: vị trí phần tử hiện tại

- ✱ **Item**: phần tử hiện tại



- Giới thiệu Ajax
- Kỹ thuật lập trình phía server
 - ✱ Trả về văn bản
 - ✱ Trả về chuỗi đối tượng json
 - Chuyển đổi map thành đối tượng json
 - Chuyển đổi object thành đối tượng json
 - ✱ Trả về mảng đối tượng json
 - Chuyển đổi list thành mảng đối tượng json
- Kỹ thuật lập trình phía client
 - ✱ Xử lý kết quả là văn bản
 - ✱ Xử lý kết quả là đối tượng json
 - ✱ Xử lý kết quả là mảng đối tượng json

