

# Spring Framework

## 26. Spring RESTful 웹서비스 어플리케이션 작성(2)

# CONTENTS

1

Ajax 개요

2

jQuery 개요

3

jQuery가 제공하는 기능

# 학습 목표

- Ajax 개요에 대하여 이해할 수 있습니다.
- jQuery 개요에 대하여 이해할 수 있습니다.
- jQuery가 제공하는 기능에 대해 이해할 수 있습니다.

A person's hands are shown holding a smartphone, with the screen glowing. The background is dark with out-of-focus, colorful bokeh lights in shades of yellow, orange, and blue. A semi-transparent dark banner is at the bottom, containing a yellow decorative shape and the title text.

AOP는

## 1. Ajax 개요

## ■ Ajax(Asynchronous Javascript and Xml)란?

웹2.0을 실현하는 핵심기능인 Ajax는 웹 사용자들에게 좀 더 수준 높은 인터페이스를 제공할 수 있도록 도움을 주는 기술의 묶음이다.

- Ajax는 그 자체가 특정한 기술이 아니라, HTML과 CSS, Javascript, XML, XMLHttpRequest객체를 비롯한 기존의 여러 기술들을 조합해서 사용하는 “웹의 새로운 접근법” 이라고 설명할 수 있다.
- “비동기적”이라는 의미는 서버로부터 데이터가 로드 되는 동안에도 계속해서 페이지를 사용할 수 있다는 뜻으로, 서버가 데이터를 전달해 주면 Ajax 이벤트가 발생하여 서버로부터 받은 데이터를 읽어 페이지의 일부를 수정하게 된다.

## ■ Ajax의 활용 예시

### 01 라이브 검색

- 라이브 검색(혹은 자동완성)이라고 불리는 기능은 주로 Ajax를 이용, 검색Site의 대부분이 사용하는 기술이고, 검색어를 입력하는 동시에 검색 결과가 나타나도록 한다.

### 02 사용자 정보 표시

- 회원가입 시 중복 아이디일 경우, “이미 사용 중인 아이디입니다”를 아이디 입력과 동시에 중복 메시지를 표시해주는 기능에서도 활용되고 있음
- 온라인 쇼핑몰에서 장바구니에 원하는 물품을 추가했을 때 페이지 이동이나 전체 페이지의 새로 고침 없이도 물품 정보만 추가되는 기능을 구현하고자 할 때도 Ajax를 활용한다.

## ■ XMLHttpRequest(XHR) 객체 사용법

- ① XMLHttpRequest 객체 생성 : Request를 보낼 준비

```
var xhr = new XMLHttpRequest();
```

- ② Callback 함수를 만든다 : 서버에서 Response가 왔을 때 실행되는 함수

```
xhr.onreadystatechange = function(){  
    if(xhr.readyState == 4){  
        var myDiv = document.getElementById('myDiv');  
        myDiv.innerHTML= xhr.responseText;  
    }  
}
```

## ■ XMLHttpRequest(XHR) 객체 사용법

③ Request를 Open 한다 : HTTP method와 호출할 Server의 url 정보를 전달

```
xhr.open("GET", "user.do");
```

④ Request를 Send 한다.

```
xhr.send();
```




## ■ jQuery를 이용하여 Ajax 통신을 하는 경우

일반 Javascript로 Ajax 코드를 작성하게 되면 코딩량도 많아지고, 브라우저 별로 구현을 다르게 해주어야 하는 단점이 있다.

jQuery를 이용하면 더 적은 코딩량과 동일한 코드로 대부분의 브라우저에서 같은 동작을 할 수 있도록 해준다.  
즉, 크로스 브라우징이 쉬워진다.

jQuery를 이용하면 100줄 정도의 Source 코드를 단 몇 줄 만으로 간단하게 서버와 데이터를 주고 받을 수 있다.

A person's hands are shown holding a smartphone, with the screen glowing. The background is dark with out-of-focus, warm-toned bokeh lights. A dark horizontal bar is at the bottom, containing a yellow decorative shape and the title text.

## 2. jQuery 개요

### ■ jQuery(<http://www.jquery.com>)란?

Javascript를 좀 더 쉽게 사용하도록 만들어진 Javascript Library로서, jQuery는 가볍고 빠르며 코드의 간결함을 제공하는 오픈 소스이다.

#### jQuery의 특징

- ◉ 가벼움 (jquery-1.9.1.min.js 의 경우 90.4KB)
- ◉ 크로스 브라우저 지원
- ◉ 강력한 CSS 셀렉터
- ◉ 메서드 체이닝
- ◉ Ajax 지원
- ◉ 풍부한 PlugIn 지원

### ■ jQuery 라이브러리 준비

#### 1. CDN(Content Delivery Network) 호스트를 사용하는 방법

- ◉ 구글과 마이크로소프트에서 CDN 호스트를 지원함
- ◉ jQuery도 CDN 호스트 지원  
→ <https://code.jquery.com/jquery-1.12.4.js>

#### 2. <http://www.jquery.com>에서 직접 다운로드 하여 로컬에 저장 후에 사용하는 방법

### jQuery를 사용하여 코드의 단순화

#### jQuery 없는 DOM 스크립팅

```
var external_links =  
    document.getElementById('external_links');  
var links =  
    external_links.getElementsByTagName('a');  
for(var i=0; i < links.length; i++){  
    var link = links.item(i);  
    link.onClick = function(){  
        return confirm('You are going to visit: '  
            + this.href);  
    }  
} //for
```

#### jQuery를 사용한 DOM 스크립팅

```
$('#external_links a').click(function(){  
    return confirm('You are going to visit: '  
        + this.href);  
});
```

### ■ jQuery가 제공하는 기능

1. HTML 엘리먼트 선택(Selector)
2. HTML 엘리먼트의 attribute 값 읽기와 쓰기
3. HTML 엘리먼트 동적으로 조작(Manipulation)
4. Loop 기능
5. CSS 조작
6. Event 처리
7. Ajax 처리

### ■ jQuery 시작하기

- ◉ jQuery를 사용하는 모든 웹 페이지는 `$(document).ready()` 로 시작해야 함.

```
<script>  
    $(document).ready(function(){  
        //페이지 로드시에 해야 할일  
    });  
//또는 ready() 함수의 축약버전으로  
$(function() {  
    //페이지 로드시에 해야 할일  
});  
</script>
```

A person's hands are shown holding a smartphone, with the screen glowing. The background is dark with out-of-focus, colorful bokeh lights in shades of yellow, orange, and blue. A semi-transparent dark banner is at the bottom, containing a yellow decorative element and the section title.

### 3. jQuery가 제공하는 기능



## ■ Selector

//태그명 직접입력 선택

```
$("div");
```

// 클래스명으로 선택

```
$(".class");
```

//ID 명으로 선택

```
$("#id");
```

// 모든 Element 선택 하여 CSS적용

```
$("*").css("border","1px");
```

//다중 셀렉터 기준으로 해당 엘리먼트들 모두 선택

```
$("div,span,p.myClass");
```

//DIV요소중 3번째 이후 부터 ((index) 보다 큰(이후) 요소)

```
$("div:gt(2)");
```

// <input type="checkbox">

```
$(":checkbox");
```

## ■ Attribute 값 읽기와 쓰기

- jQuery를 `attr(attrName)`, `attr(attrName, value)`, `attr(object)` 함수  
: 선택된 element의 속성(attribute)의 값을 리턴, 설정한다.

적용 전

```
<img id="greatphoto">
```

attr()

```
$("#greatphoto").attr("src","brush-seller.jpg");  
$("#greatphoto").attr("src");  
$("#greatphoto").attr({  
  alt: "Brush Seller",  
  title: "photo by Kelly Clark"  
});
```

적용 후

```

```

## ■ Manipulation

- ◉ **append()** 함수

: 선택된 element의 content 맨 끝에 인자로 넘어온 내용을 추가한다.

- ◉ **appendTo()** 함수

: 선택된 element를 target에 해당하는 Element의 content의 끝에 추가한다.

```
$(".inner").append("<p>Test</p>");  
$("<p>Test</p>").appendTo(".inner");
```

- ◉ **html(), html(htmlString)** 함수

: 선택된 Element의 html을 리턴, 설정한다.

```
$("#div.demo-container").html();  
$("#div").html("<h1>제목</h1>");
```

#### ■ Loop 기능

- ◉ `.each(function(index, element))` 함수

: jQuery 객체들을 위해 반복 로직을 처리하는 함수

```
<ul>
  <li>foo</li>
  <li>bar</li>
</ul>
```

```
$('.li').each(function(index,element) {
    alert(index + ' : ' + $(element).text());
});
```

## ■ CSS 조작

- ◉ `css(propertyName)`, `css(propertyName, value)`, `css(object)` 함수

: 선택된 element의 propertyName에 해당되는 CSS정보를 리턴, 설정한다.

```
$("#span").css("width");  
$("#span").css("width", "100px");  
$("#span").css({"width":"100px",  
"height":"50px"});
```

- ◉ `addClass(className)` 함수

: 선택된 element에 해당되는 클래스 추가

```
$("#p").addClass("myClass");
```

#### ■ Event 처리

- ◉ **on() 함수**

: 선택된 element에 이벤트 핸들러를 묶어준다.

```
$("#p").on("click", function(){  
    alert( $(this).text() );  
});
```

```
$("#div.test").on({  
    mouseenter: function(){  
        $(this).addClass("inside");  
    },  
    mouseleave: function(){  
        $(this).removeClass("inside");  
    }  
});
```

#### ■ Ajax 처리

- ◉ \$.ajax() 함수

```
$('#btnSelect').on('click',function(){
    $.ajax({
        url:'users',
        type:'GET',
        contentType:'application/json;charset=utf-8',
        dataType:'json',
        error:function(xhr,status,msg){
            alert("상태값 : " + status + " Http에러메시지 : "+msg);
        },
        success:userSelectResult
    });
});
```

A person's hands are shown holding a smartphone with a white screen. The background is dark with out-of-focus, glowing yellow and blue circular lights, creating a bokeh effect. A semi-transparent dark horizontal bar is at the bottom, containing a yellow graphic element and the text '학습정리'.

학습정리



지금까지 [Spring RESTful 웹서비스 어플리케이션 작성(2)]에 대해서 살펴보았습니다.

## Ajax개요

- ◉ Ajax 개념 및 활용예시
- ◉ XMLHttpRequest (XHR) 객체

## jQuery 개요

- ◉ jQuery 개념 및 특징
- ◉ jQuery 라이브러리 준비, `$(document).ready()`

## jQuery가 제공하는 기능

- ◉ Selector, attribute 조작, Manipulation, CSS 조작, Event, Ajax