Grails에서의 Ajax

Ajax

Table of Contents

grails에서는 ajax를 통신하는 가장 간단한 방법으로는 remoteLink 태그를 통해 ajax 통신을 할 수 있다.

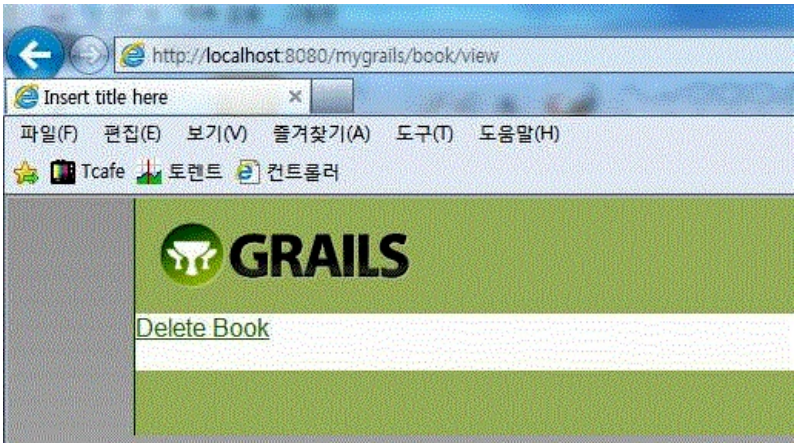
확인주소 : http://localhost:8080/mygrails/book/view
view.gsp의 내용을 아래와 같이 변경한다.

```
<%@ page contentType="text/html;charset=UTF-8" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR"/>
<meta name="layout" content="main"/>
<title>Insert title here</title>
<g:javascript library="jquery" />
</head>
<body>
  <div class="body">

 <div id="message"></div>
<g:remoteLink action="delete" id="1" update="message">
Delete Book
</g:remoteLink>

  </div>
</body>

</html>
```

BookController 의 내용을 아래와 같이 변경한다.

```
package net.grails.my

class BookController {

    def view = {

    }

def delete() {
    //def b = Book.get(params.id)
    //b.delete()
    def id = params.id
    render "Book ${id} was deleted"
} }
```

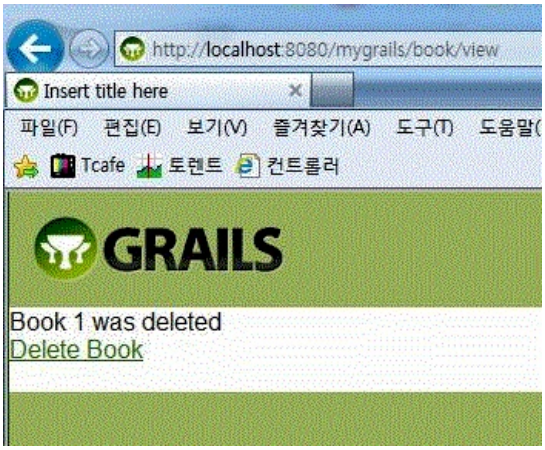
실행하면 아래와같이 확인할수있다



Delete Book A Tag가 실제로 렌더링 된 html이다.
클릭하면 ajax 로 요청을 보내고 성공하면 message div 에 html을 할당하는 모습을 볼 수 있다.

```
<div id="message"></div>
<a href="/mygrails/book/delete/1" onclick="jQuery.ajax({type:'POST', url:'/mygrails/book/delete/1',success:function(data,textStatus){jQu
Delete Book
</a>
```

Delete Book 클릭 하면 delete action이 실행된다.
delete action에서 리턴받은(response) 데이터를 message div 에 html을 할당 하는 걸 확인할 수 있다.



formRemote 태그를 이용하여 forn 전송을 할 수 도있다.

view.gsp의 내용을 아래와 같이 변경한다.

```
<%@ page contentType="text/html;charset=UTF-8" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR"/>
<meta name="layout" content="main"/>
<title>Insert title here</title>
<g:javascript library="jquery" />
</head>
<body>
  <div class="body">

  <div id="message"></div>
  <div id="error"></div>

<g:formRemote url="[controller: 'book', action: 'delete']"  update="[success: 'message', failure: 'error']" name="myForm">
    <input type="hidden" name="id" value="1" />
    <input type="submit" value="Delete Book!" />
</g:formRemote >

  </div>
</body>

</html>
```

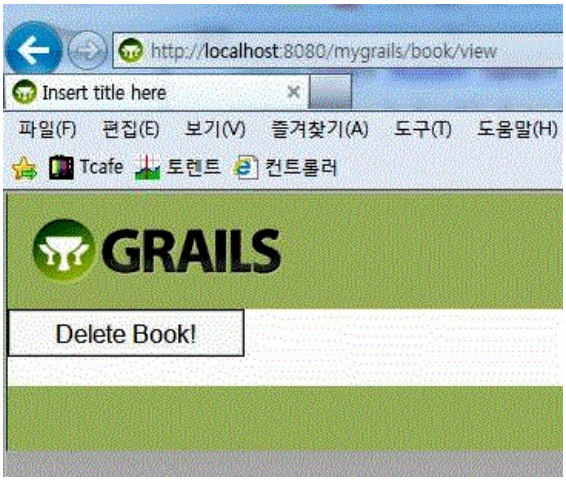
form 전송을 위해 submit 버튼과 hidden 으로 이루어져있다.

실제 렌더링 된 html은 아래와같다.

```
<div id="message"></div>
  <div id="error"></div>

<form onsubmit="jQuery.ajax({type:'POST',data:jQuery(this).serialize(), url:'/mygrails/book/delete',success:function(data,textStatus){jQuer
    <input type="hidden" name="id" value="1" />
    <input type="submit" value="Delete Book!" />
</form>
```

onsubmit 이벤트에 ajax 호출이 이루어져있다.

실행하면 아래와같은 모습이다.



Delete Book! 버튼을 클릭하면 결과는 formRemote 와 같다

Content Centric Ajax(컨텐트 중심 ajax)

template을 서버에서 렌더링 하여 html을 response 받을 수 있다.

테스트 시나리오
1. 페이지를 요청한다.
요청주소 : http://localhost:8080/mygrails/book/view/2
2. ajax 로 book의 상세정보가 담긴 template(html)을 요청하여 div에 셋팅한다.
ajax 요청주소 : action : showBook

_bookTemplate.gsp 을 추가하고 내용을 아래와 같이 변경한다

```
<%@ page contentType="text/html;charset=UTF-8" %>
<div style="background-color:#CCC">
<p>bookTemplate 입니다.</p>
<p>책 이름은  <span style="color:red">${book.title}</span> 입니다.</p>
</div>

```

BookController 의 내용을 아래와 같이 변경한다.

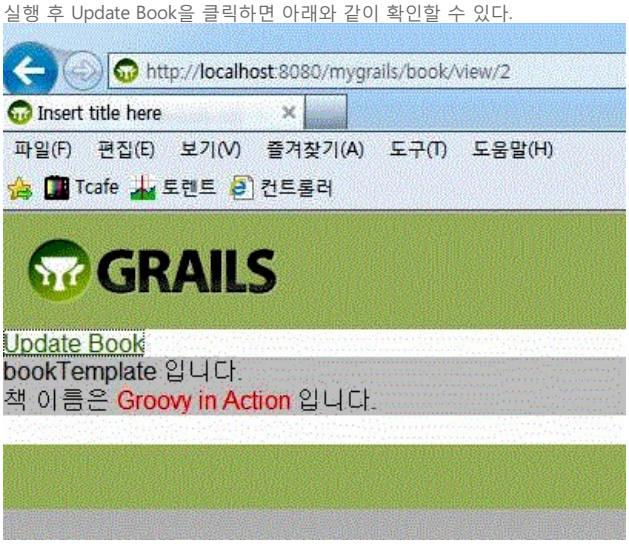
```
package net.grails.my

class BookController {

```

```
    def view = {
        Book book = new Book();
            println params.id
            book.id = params.id.toInteger()

            [book:book]
    }

def showBook() {
             Book book = new Book();
            book.id = params.id
            //조회로직
            book.title = "Groovy in Action"

     render( model: [book: book], template: "bookTemplate")
}
}
```

view.gsp의 내용을 아래와 같이 변경한다.

```
<%@ page contentType="text/html;charset=UTF-8" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR"/>
<meta name="layout" content="main"/>
<title>Insert title here</title>
<g:javascript library="jquery" />
</head>
<body>
  <div class="body">

  <div id="message"></div>
  <div id="error"></div>

<g:remoteLink action="showBook" id="${book.id}"
              update="book${book.id}">Update Book</g:remoteLink>
<div id="book${book.id}">
   <!--existing book mark-up -->
</div>
</div>
</body>

</html>
```

실행 후 Update Book을 클릭하면 아래와 같이 확인할 수 있다.



showBook action에서는 bookTemplate과 모델을 렌더링하여 reponse 한다

Data Centric Ajax with JSON

ajax 로 요청을 보내고 응답을 json 데이터로 받아서 처리한다.

확인주소 : http://localhost:8080/mygrails/book/view/2
BookController의 내용을 아래와 같이 변경한다.

```
package net.grails.my
import grails.converters.JSON
class BookController {
```

```
    def view = {
        Book book = new Book();
            println params.id
            book.id = params.id.toInteger()

            [book:book]
    }

def showBook() {
             Book book = new Book();
            book.id = params.id.toInteger()
            //조회로직
            book.title = "Groovy in Action".toString()
            render book as JSON
}
}
```

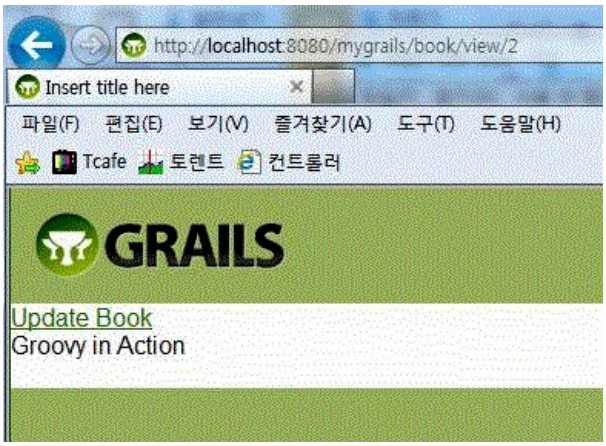
Book.groovy의 내용을 아래와 같이 변경한다.

```
package net.grails.my

class Book {
    String title
    static constraints = {
    }
}
```

view.gsp의 내용을 아래와 같이 변경한다.

```
<%@ page contentType="text/html;charset=UTF-8" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR"/>
<meta name="layout" content="main"/>
<title>Insert title here</title>
<g:javascript library="jquery" />
</head>
<body>
  <div class="body">

  <g:javascript>
function updateBook(data) {
    $("#book" + data.id + "_title").html( data.title );
}
</g:javascript>
<g:remoteLink action="showBook" id="${book.id}" onSuccess="updateBook(data)">
    Update Book
</g:remoteLink>
<g:set var="bookId">book${book.id}</g:set>
<div id="${bookId}">
    <div id="book${book.id}_title">The Stand</div>
</div>
</div>
</body>

</html>
```

실행 후 Upadate Book을 클릭하면 아래와 같이 확인할 수 있다



Data Centric Ajax with XML

XML 로 요청을 보내고 응답을 XML 데이터로 받아서 처리한다.
BookController에 임포트를 추가한다.

```
import grails.converters.XML
```

showBook의 내용을 아래와 같이 변경한다.

```
def showBook() {
        Book book = new Book();
        book.id = params.id.toInteger()
        //조회로직
        book.title = "Groovy in Action".toString()
        render book as XML
}
```

view.gsp
updateBook 의 함수를 아래와 같이 변경한다.
json과는 달리 xml을 파싱하는 부분이 들어간다.

정보 : fiddler같은 웹 디버깅 툴을 이용하여 xml이 어떤 형태로 response 되는지 확인 하면 파싱하기가 수월 할 것이다.

```
function updateBook(data) {
    var id = $(data).find("book").attr("id");
    $("#book" + id + "_title").html( $(data).find("title").text() );
}
```

Responding to both Ajax and non-Ajax requests

ajax 요청일때와 ajax 요청이 아닐때 모두 각각 다르게 응답을 줄 수 있다.

```
def listBooks() {
    def books = Book.list(params)
    if (request.xhr) {
        //ajax 일 경우 처리
    } else {
       //non-ajax 일 경우 처리
    }
}
```

관련 키워드

Grails Ajax