



마이자몽



전체 방문자

206,741

어제

862

오늘

364

☆ 구독하기

N 네이버 이웃 맺기

RSS 피드

홈

방명록

개발

맛집 후기

자료구조 & 알고리즘 (6)



개발 (65)



Python (36)



맥OS (1)



[Spring] 스프링 AJAX 여러유형 사용 예시 :: 마이자몽

2018. 9. 6. 23:22 | 개발/Spring

AJAX(Asynchronous JavaScript and XML)은 여러 상황에서 사용되는데... 매번 사용할때마다 어떻게하는지 검색하고 애먹는경우가 많다.

그래서, 이 글만 보고 완벽하게 다시 사용할수 있도록 정리해볼려고 한다.

AJAX는 단순히 텍스트값만 넘길수도 있고, form을 넘길수도 있고, 파일 업로드와 같이 웹개발에서 중요한 역할을 해주기때문에....

AJAX만 잘 알아도 개발하는데 시간단축이 된다.

AJAX는 비동기식 방법으로 데이터에 접근하는 거라고한다. 쉽게 말하면 그냥 브라우저에서 새로고침하지 않고도 데이터값을 변경할수 있는 것이다.

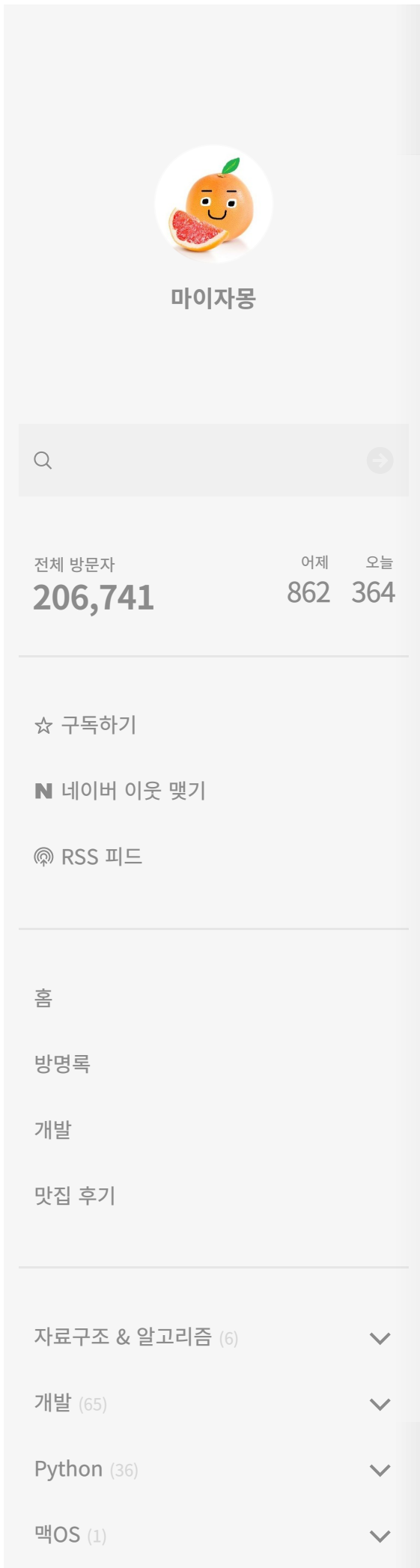
대표적으로 AJAX가 사용된 사례는 각종 포털사이트의 인기검색어 혹은 연관검색어이다. 그럼 유형별로 하나씩 써보자.

AJAX데이터를 담을 Jamong DTO 클래스

```

1 public class Jamong {
2     String name;
3     int age;
4     public String getName() {
5         return name;
6     }
7     public void setName(String name) {
8         this.name = name;
9     }
10    public int getAge() {

```



```

11     return age;
12 }
13 public void setAge(int age) {
14     this.age = age;
15 }
16
17 }

```

1. 기본

AJAX의 최소화 형태이다.

data를 전송해서 필요한 로직들을 처리하고 값을 return한다.

여기서 @ResponseBody의 역할을 알아야하는데, @RequestMapping으로 String type을 반환해주면, ViewResolver에서 정의한 prefix와 suffix가 return값에 추가되어 view로 이동이 된다.

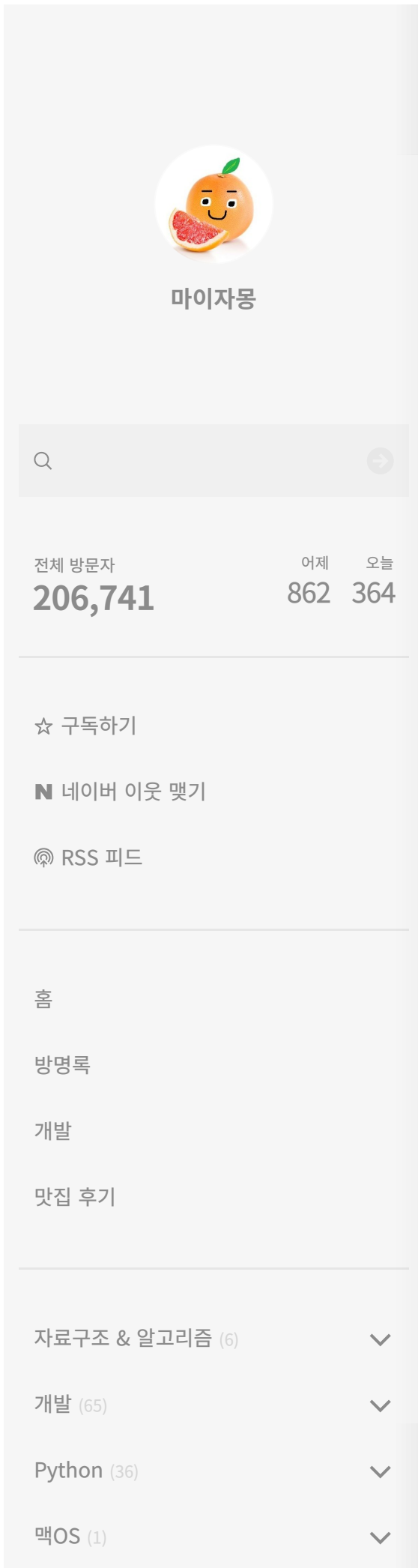
@ResponseBody를 사용해주면 view를 생성해주는것이 아니라, JSON 혹은 Object 형태로 데이터를 넘겨준다.

-Client

```

1  $('#btn1').on('click', function(){
2      var form = {
3          name: "jamong",
4          age: 23
5      }
6      $.ajax({
7          url: "requestObject",
8          type: "POST",
9          data: form,
10         success: function(data){
11             $('#result').text(data);
12         },
13         error: function(){
14             alert("simpleWithObject err");
15         }
16     });
17 });

```



-Server

```

1  @RequestMapping(value="/requestObject", method=RequestMethod.POST)
2  @ResponseBody
3  public String simpleWithObject(Jamong jamong) {
4      //필요한 로직 처리
5      return jamong.getName() + jamong.getAge();
6  }

```

-View

```

1  <body>
2      <button id="btn1">simpleAJAX</button>
3      <div id="result"></div>
4  </body>

```

simpleAJAX
jamong23

2. 폼 파라미터 넘기기

폼의 파라미터를 넘기기 위해 `serialize()` 함수를 사용한다. 필요한 로직 처리를 하고 마찬가지로 `@ResponseBody` Annotation을 사용하여 Object형태로 넘긴다.

-Client



마이자몽



전체 방문자

206,741

어제

862

오늘

364

☆ 구독하기

N 네이버 이웃 맺기

RSS 피드

홈

방명록

개발

맛집 후기

자료구조 & 알고리즘 (6)



개발 (65)



Python (36)



맥OS (1)



```

1      $('#btn2').on('click', function(){
2          $.ajax({
3              url: "serialize",
4              type: "POST",
5              data: $("#frm").serialize(),
6              success: function(data){
7                  $('#result').text(data);
8              },
9              error: function(){
10                 alert("serialize err");
11             }
12         });
13     });

```

-Server

```

1      @RequestMapping(value="/serialize", method=RequestMethod.POST)
2      @ResponseBody
3      public String serialize(Jamong jamong) {
4          //필요한 로직 처리
5          return jamong.getName() + jamong.getAge();
6      }

```

-View

```

1      <body>
2      <form id="frm">
3          name : <input type="text" name="name" id="name">
4          age : <input type="text" name="age" id="age">
5      </form>
6      <button id="btn2">serialize</button>
7      <div id="result"></div>
8      </body>

```



마이자몽



전체 방문자

206,741

어제

862

오늘

364

☆ 구독하기

N 네이버 이웃 맺기

RSS 피드

홈

방명록

개발

맛집 후기

자료구조 & 알고리즘 (6)



개발 (65)



Python (36)



맥OS (1)



name : jamong

age : 23

serialize

jamong23

3. JSON.stringify()

AJAX에는 dataType이라는 옵션이 있다. 설정을 안해주면 MIME타입에 의해 자동으로 설정이된다.

그런데 나는 JSON형태로 보내고싶다.

AJAX를 사용하는 예시를 보면 JSON.stringify() 함수를 많이 사용한다.

이 함수는 JSON타입을 String 객체로 변환시켜주는 역할을 한다.

서버쪽으로 데이터를 넘길때, String객체로 변환시켜주기 않으면 URL의 get 파라미터 형식(?a=1&b=2)처럼 넘겨버린다.

자... 근데... stringify함수를 그냥 사용하게되면 415에러가 뜬다.

이부분에서 필자는 굉장히 삽질을 했는데.... json

라이브러리를 추가해줘야한다.

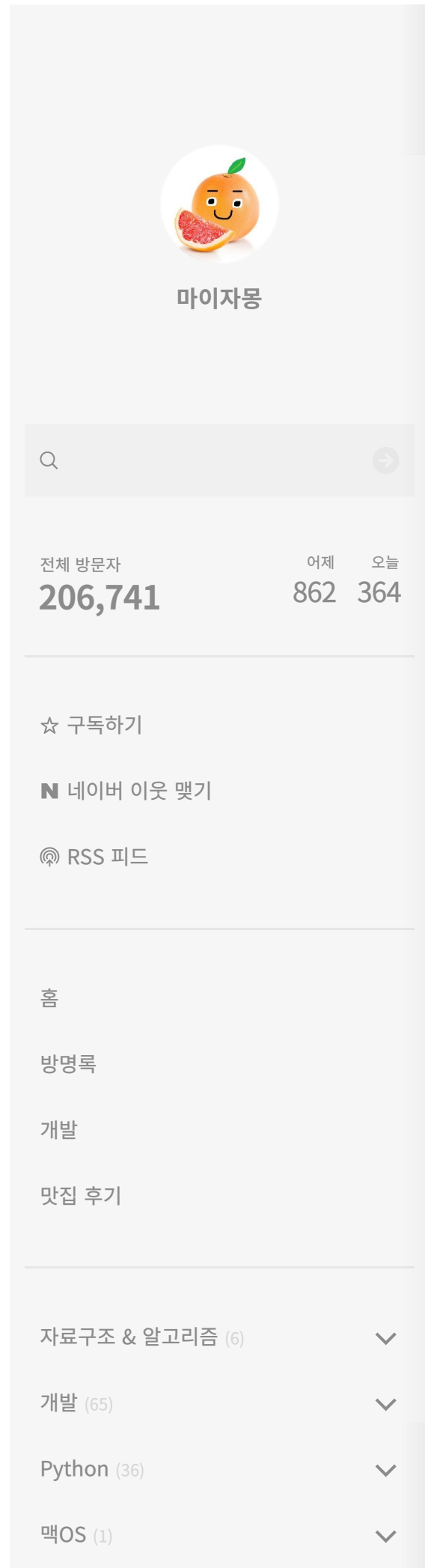
서버쪽으로 데이터를 넘기면서 무슨 변환작업을 해주는것 같은데, 추가안해주면 415에러가 뜬다.

pom.xml파일에 추가해준다.

```

1      <dependency>
2          <groupId>com.fasterxml.jackson.core</groupId>
3          <artifactId>jackson-databind</artifactId>
4          <version>2.9.5</version>
5      </dependency>

```



코드를 보기 이전에 이번에는 @RequestBody에 대해서 알아야한다.

클라이언트쪽에서 데이터 형태를 JSON으로 보내주면, 서버쪽에서도 JSON으로 받아줘야한다.

JSON형태로 받게해주는 Annotation이 @RequestBody이다.

-Client

```

1  $('#btn3').on('click', function(){
2      var form = {
3          name: "jamong",
4          age: 23
5      }
6      $.ajax({
7          url: "stringify",
8          type: "POST",
9          data: JSON.stringify(form),
10         contentType: "application/json; charset=utf-8",
11         dataType: "json",
12         success: function(data){
13             var txt = data.name + data.age;
14             $('#result').text(txt);
15         },
16         error: function(){
17             alert("stringify err");
18         }
19     });
20 });

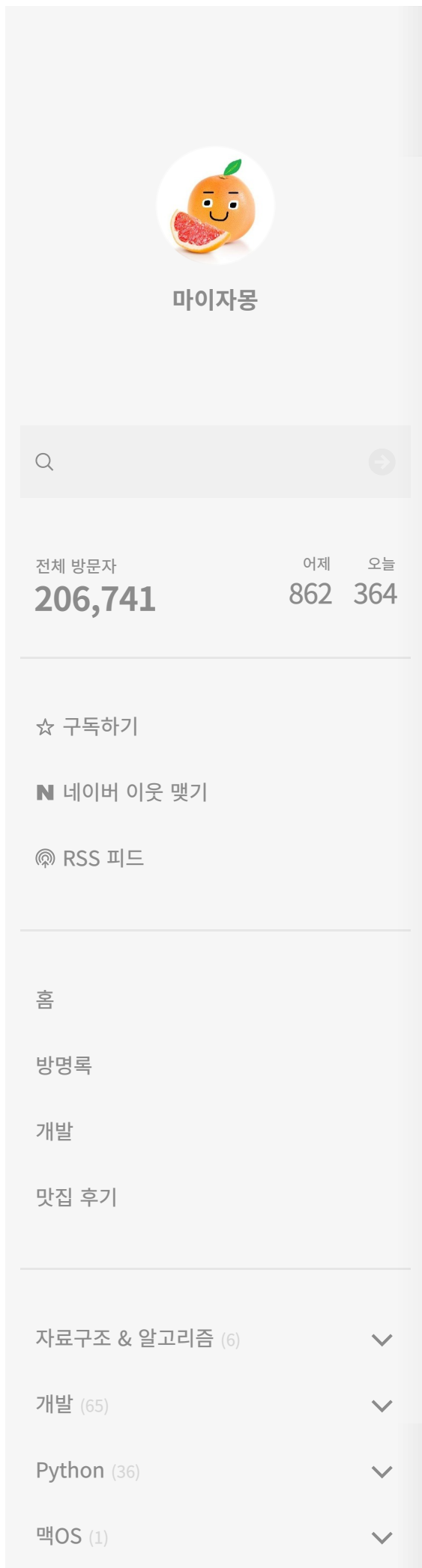
```

-Server

```

1  @RequestMapping(value="/stringify", method=RequestMethod.POST)
2  @ResponseBody
3  public Object stringify(@RequestBody Jamong jamong) {
4      HashMap<String, Object> map = new HashMap<>();
5      map.put("name", jamong.getName());
6      map.put("age", jamong.getAge());
7      return map;

```



```
8      return map;
```

-View

```
1 <body>
2   <button id="btn3">stringify</button>
3   <div id="result"></div>
4 </body>
```

stringify
jamong23

4. @RestController

다음은 @Controller가 아닌 @RestController에서 JSON타입으로 데이터를 주고받을거다.

@RestController Annotation을 사용하면,
@ResponseBody Annotation을 빼고 사용하면된다.
@RestController가 알아서 JSON형태로 return해준다.

-Client

```
1  $('#btn4').on('click', function(){
2      var form = {
3          name: "jamong",
4          age: 23
5      }
6      $.ajax({
7          url: "restController",
8          type: "POST",
9          data: JSON.stringify(form),
10         contentType: "application/json; charset=utf-8",
11         dataType: "json",
12         success: function(data){
```



마이자몽



전체 방문자

206,741

어제

862

오늘

364

☆ 구독하기

N 네이버 이웃 맺기

RSS 피드

홈

방명록

개발

맛집 후기

자료구조 & 알고리즘 (6)



개발 (65)



Python (36)



맥OS (1)



```

13         for(var i=0; i<data.length; i++){
14             $('#result').append(data[i] + "<br>");
15         }
16     },
17     error: function(){
18         alert("restController err");
19     }
20 });
21 });

```

-Server

```

1 @RestController
2 public class AjaxRestController {
3     @PostMapping("/restController")
4     public Object restController(@RequestBody Jamong jamong,
5     ArrayList<String> arrList = new ArrayList<String>() {
6         for(int i=0; i<5; i++) {
7             arrList.add(jamong.getName() + i);
8         }
9         return arrList;
10    }
11 }

```


-View

```

1 <body>
2     <button id="btn4">restController</button>
3     <div id="result"></div>
4 </body>

```

CS



마이자몽

전체 방문자

206,741

어제

862

오늘

364

☆ 구독하기

N 네이버 이웃 맺기

RSS 피드

홈

방명록

개발

맛집 후기

자료구조 & 알고리즘 (6)

▼

개발 (65)

▼

Python (36)

▼

맥OS (1)

▼

restController

jamong0
jamong1
jamong2
jamong3
jamong4

5. 파일 업로드가 AJAX 처리

지금까지 AJAX의 정말 기본적인건 다 설명한것같다.

데이터 형태만 잘 맞춰서 넘겨주고, DB와 접속하든, API연동을 하여 데이터를 받아오든, 처리로직만 잘 처리해주면 된다.

이제 AJAX로 뭐든지 다 할 수 있을것 같다는 생각이 들지만....

폼에 input type이 file로 되어있는 파일들은 어떻게 처리하지? 라는 의문이 생길 것이다.

아래와 같이 잘 따라하면 된다.

필자는 파일 유효성검사나 업로드하는 작업은 하지않았다.

데이터가 넘어가고 받아져오는 것만 확인하기 위해 파일 이름만 출력하도록 코드를 작성했다.

파일을 받아줄 AjaxFile Dto 클래스

```

1 public class AjaxFile {
2
3     List<MultipartFile> images;
4
5     public List<MultipartFile> getImages() {
6         return images;
7     }
8     public void setImages(List<MultipartFile> images) {
9         this.images = images;
10    }
11 }
```

파일 업로드를 위해 maven 추가 pom.xml

```

1      <dependency>
2          <groupId>commons-fileupload</groupId>
3          <artifactId>commons-fileupload</artifactId>
4          <version>1.3.2</version>
5      </dependency>
6
7      <dependency>
8          <groupId>commons-io</groupId>
9          <artifactId>commons-io</artifactId>
10         <version>2.5</version>
11     </dependency>

```

servlet-context.xml

```

1      <!-- file Upload -->
2      <beans:bean id="multipartResolver" class="org.sp
3          <!-- 파일 최대 50MB -->
4          <beans:property name="maxUploadSize" value=
5      </beans:bean>


```

-Client

```

1      $('#btn5').on('click', function(){
2          var form = new FormData($('#frm')[0]);
3          $.ajax({
4              url: "file",
5              type: "POST",
6              data: form,
7              contentType: false,
8              processData: false,
9              success: function(data){
10                  for(var i=0; i<data.length; i++){
11                      $('#result').append(data[i] + '<br>')
12                  }
13              },

```



마이자몽

전체 방문자

206,741

어제

862

오늘

364

☆ 구독하기

N 네이버 이웃 맺기

RSS 피드

홈

방명록

개발

맛집 후기

자료구조 & 알고리즘 (6)

개발 (65)

Python (36)

맥OS (1)



마이자몽



전체 방문자

206,741

어제

862

오늘

364

☆ 구독하기

N 네이버 이웃 맺기

RSS 피드

홈

방명록

개발

맛집 후기

자료구조 & 알고리즘 (6)



개발 (65)



Python (36)



맥OS (1)



```

14         error: function(){
15             alert("file err");
16         }
17     });
18 });

```

CS

-Server

```

1  @RequestMapping(value="/file", method=Request
2  @ResponseBody
3  public Object file(AjaxFile file) {
4      List<MultipartFile> list = file.getImages();
5      ArrayList<String> fileNameList = new ArrayList<
6      for(MultipartFile mf : list) {
7          fileNameList.add(mf.getOriginalFilename());
8      }
9      return fileNameList;
10 }

```

-View

```

1  <body>
2  <form id="frm" enctype="multipart/form-data">
3      <input multiple="multiple" type="file" id="images
4  </form>
5      <button id="btn5">file</button>
6      <div id="result"></div>
7  </body>

```



마이자몽



전체 방문자

206,741

어제

862

오늘

364

☆ 구독하기

N 네이버 이웃 맺기

RSS 피드

홈

방명록

개발

맛집 후기

자료구조 & 알고리즘 (6)



개발 (65)



Python (36)



맥OS (1)



Choose Files 4 files

file

Screen Shot 2018-09-06 at 9.39.05 PM.png
 Screen Shot 2018-09-06 at 9.53.03 PM.png
 Screen Shot 2018-09-06 at 10.26.56 PM.png
 Screen Shot 2018-09-06 at 10.40.57 PM.png

필자의 생각으로 이정도만 알고있다면, Ajax처리하는데 크게 문제가 없을거라고 생각한다.

AJAX의 기본 사용법을 알아봤고, DB 혹은 API를 연동하여 AJAX 실전편을 올릴 계획이다.

끝;

1



☆ 구독하기

N 이웃



'개발 > Spring' 카테고리의 다른 글



- [Spring] 스프링 엑셀파일 업로드 다운로드 기본(POI 라이브러... →
- [Spring] 몽고DB CRUD 데이터 연동 :: 마이자몽 →
- [Spring] background Thread 동적 작업 실행시키기 Threa... →
- [Spring] 스프링 Logback 로그 파일로 남기기 :: 마이자몽 →
- [Spring] 스프링 RestTemplate get방식 :: 마이자몽 →
- [Spring] 스프링 AJAX 여러유형 사용 예시 :: 마이자몽 ✓

AJAX

AJAX 실습

AJAX 예제

아작스

댓글

공유하기

다른 글

[Spring] 몽고DB CRUD 데이터 연동 :: 마이자몽

2019.04.14





마이자몽



전체 방문자

206,741

어제

862

오늘

364

☆ 구독하기

N 네이버 이웃 맺기

RSS 피드

홈

방명록

개발

맛집 후기

자료구조 & 알고리즘 (6)



개발 (65)



Python (36)



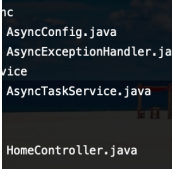
맥OS (1)



[Spring] background Thread 동적 작업 실행시키기

ThreadPoolTaskExecutor :: 마이자몽

2019.04.14



[Spring] 스프링 Logback 로그 파일로 남기기 :: 마이자몽

2018.12.28



[Spring] 스프링 RestTemplate get방식 :: 마이자몽

2018.09.10



다른 글 더 둘러보기 ➔

< 최신

122

다음 >



© myJamong. Designed by Fraccino.

