

실전 예제



❑ 첫번째 예제

- 13장에서 작성해둔 연락처 서비스 이용
- 강력한 UI를 작성하기 위해 몇몇 플러그인 사용
 - 플러그인에 대해선 17장 참조

❑ 두번째 예제

- 부트스트랩 활용
- 라이트박스 플러그인 활용
- 간단한 이미지 뷰어

사용할 jQuery 플러그인(1)



■ waitme 플러그인

- AJAX 요청시 지연시간 발생할 때 화면에 처리중임을 나타낼 수 있는 플러그인
- <http://www.jqueryscript.net/loading/jQuery-Plugin-For-Creating-Loading-Overlay-with-CSS3-Animations-waitMe.html>
- 내려받은 파일 중 waitMe.js, waitMe.css, img.svg 파일을 waitme 디렉토리를 하나 만들어 저장

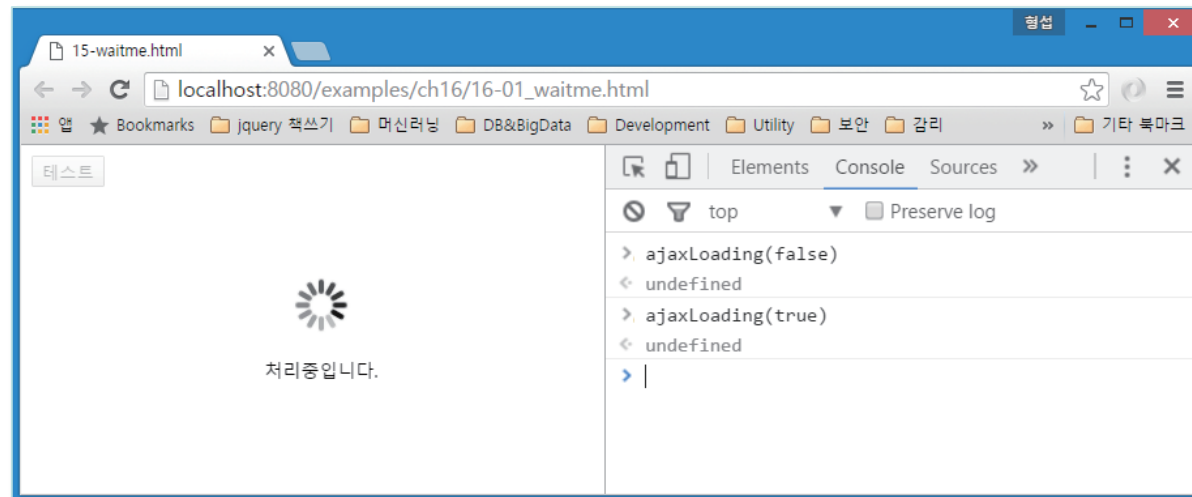
사용할 jQuery 플러그인(2)



■ 예제 16-01 : waitme 플러그인

```
07: <script src="http://code.jquery.com/jquery-3.1.0.js"></script>
08: <script src="waitme/waitMe.js"></script>
09: <script type="text/javascript">
10: var ajaxLoading = function(isLoading){
11:     if (isLoading == false) {
12:         $("body").waitMe("hide");
13:     } else {
14:         $("body").waitMe({
15:             effect: "ios",
16:             text: "처리중입니다.",
17:             bg: 'rgba(255,255,255,0.7)',
18:             color: '#000',
19:             source: 'waitme/img.svg',
20:         });
21:     }
22: };
23:
24: $(document).ready(function() {
25:     $("#test").click(function() {
26:         ajaxLoading(true);
27:     });
28: });
29: </script>
```

```
32: <div id="container" style="height:400px;">
33:     <button id="test">테스트</button>
34: </div>
```



사용할 jQuery 플러그인(3)



■ jPagate 플러그인

- 게시판과 같은 페이지 이동 기능 제공
- <http://tympanus.net/codrops/2009/11/17/jpaginate-a-fancy-jquery-pagination-plugin/>
- 내려받은 파일을 압축을 풀어서 jquery.paginate.js 파일과 css 디렉터리 아래의 style.css 파일을 복사
- 예제 16-02

```
33:      <h1>jPaginate 테스트</h1>
34:      <div>
35:          현재 페이지 : <span id="currentPage">1</span>
36:      </div>
37:      <div id="paging"></div>
```

사용할 jQuery 플러그인(4)



■ 예제 16-02(이어서)

```
11:    $("#paging").paginate({
12:        count : 100,
13:        start : 1,
14:        display : 10,
15:        border : true,
16:        border_color : '#333333',
17:        text_color : '#000000',
18:        background_color : 'aqua',
19:        border_hover_color : '#ccc',
20:        text_hover_color : '#000',
21:        background_hover_color : '#fff',
22:        images : false,
23:        mouse : 'press',
24:        onChange : function(page) {
25:            $("#currentPage").html(page);
26:        }
27:    });
```

jPaginate 테스트

현재 페이지 : 100

First << 94 95 96 97 98 99 100 >> Last

사용할 jQuery 플러그인(5)

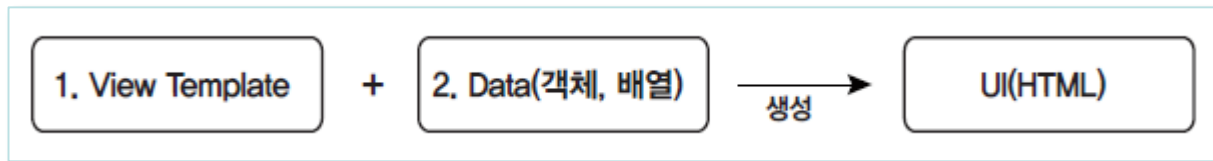


■ jsRender 플러그인

- 데이터를 이용해 UI로 나타내는 작업은 쉽지 않음
 - 문자열 이어붙이기 : 생성 속도는 빠르지만 유지 보수성과 가독성이 나쁨

```
var str = "<div style='width:"+ option.width +"px; height:" + option.height +  
    "px; color:" + option.color + "; background-color:" + option.bgColor + "'>" +  
    "이름 : " + data.name + "<br />" +  
    "나이 : " + data.age + "<br />" +  
    "이메일 : " + data.email + "<br />" +  
    "</div>";  
$("#result").html(str);
```

- jsRender는 모델과 뷰를 분리하여 개발할 수 있는 기능 제공



- <http://www.jsviews.com/#jsrender>

사용할 jQuery 플러그인(6)



■ 예제 16-03

```
11: <script src="https://code.jquery.com/jquery-3.1.0.js"></script>
12: <script src="https://www.jsviews.com/download/jsrender.js"></script>
13: <script type="text/javascript">
14: $(document).ready(function() {
15:     $.get("/contact/list.jsp", function(data) {
16:         var tpl = $.templates("#contact_template");
17:         var str = tpl.render(data.contacts);
18:         $("#container").html(str);
19:     });
20: });
21: </script>
```

```
22: <script id="contact_template" type="text/x-jsrender">
23:     <tr>
24:         <td>{{:no}}</td>
25:         <td>{{:name}}</td>
26:         <td>{{:tel}}</td>
27:         <td>{{:address}}</td>
28:     </tr>
29: </script>
```

```
36:         <tbody id="container">
37:
38:         </tbody>
```

- {{:no}} 와 같은 형식을 이용해 뷰템플릿 활용

사용할 jQuery 플러그인(7)



■ 예제 16-03 실행 결과

번호	이름	전화번호	주소
15	혜리	010-2121-3324	서울시
14	소진	010-2121-3323	서울시
13	하니	010-2121-3322	서울시
12	설현	010-2121-3321	서울시
11	진구	010-2121-3320	서울시
10	박해진	010-2121-3319	서울시
9	조진웅	010-2121-3318	서울시
8	김지원	010-2121-3317	서울시
7	송혜교	010-2121-3316	서울시
6	송중기	010-2121-3315	서울시
5	김수현	010-2121-3314	서울시
4	원재현	010-2121-3313	서울시
3	원채빈	010-2121-3312	서울시
2	이선영	010-2121-3311	서울시
1	원형섭	010-2121-3310	서울시

사용할 jQuery 플러그인(8)



■ jQuery UI Dialog 플러그인

- jQuery UI는 jQuery 측에서 제공하는 공식 플러그인
- jQuery UI는 다양한 플러그인의 집합체
 - 상호작용(Interaction)
 - 위젯(Widget)
 - 효과(Effect)
 - 유틸리티(Utility)
- 자세한 내용은 17장에서 살펴보도록하고 이 장에선 다이얼로그 플러그인만 사용
- CDN 사용

```
<link rel="stylesheet" href="https://code.jquery.com/ui/1.12.0/themes/humanity/jquery-ui.css" />
```

```
<script type="text/javascript" src="https://code.jquery.com/ui/1.12.0/jquery-ui.js"></script>
```

사용할 jQuery 플러그인(9)



■ 예제 16-04

```
06: <link rel="stylesheet" href="https://code.jquery.com/ui/1.12.0/themes/humanity/jquery-ui.css" />
07: <style>
08:     body { font-size: 75%; }
09:     input.text { margin-bottom:12px; width:95%; padding: .4em; }
10: </style>
11: <script src="https://code.jquery.com/jquery-3.1.0.js"></script>
12: <script src="https://code.jquery.com/ui/1.12.0/jquery-ui.js"></script>
13: <script type="text/javascript">
14: $(document).ready(function() {
15:     $("#dialog-new").dialog({
16:         modal:true,
17:         autoOpen : true,
18:         buttons : {
19:             "저장" : function() {
20:                 $(this).dialog("close");
21:                 alert("저장 완료");
22:             },
23:             "취소" : function() {
24:                 $(this).dialog("close");
25:             }
26:         }
27:     });
28: });
29: </script>
30: </head>
31: <body>
32:     <div id="dialog-new" title="새 연락처 추가하기">
33:         <form>
34:             <label for="name">이름</label>
35:             <input type="text" name="name" id="name"
36:                 class="text ui-widget-content ui-corner-all">
37:             <label for="tel">전화번호</label>
38:             <input type="text" name="tel" id="tel"
39:                 class="text ui-widget-content ui-corner-all">
40:             <label for="address">전화번호</label>
41:             <input type="text" name="address" id="address"
42:                 class="text ui-widget-content ui-corner-all">
43:         </form>
44:     </div>
45: </body>
```

```
27:     });
28: });
29: </script>
30: </head>
31: <body>
32:     <div id="dialog-new" title="새 연락처 추가하기">
33:         <form>
34:             <label for="name">이름</label>
35:             <input type="text" name="name" id="name"
36:                 class="text ui-widget-content ui-corner-all">
37:             <label for="tel">전화번호</label>
38:             <input type="text" name="tel" id="tel"
39:                 class="text ui-widget-content ui-corner-all">
40:             <label for="address">전화번호</label>
41:             <input type="text" name="address" id="address"
42:                 class="text ui-widget-content ui-corner-all">
43:         </form>
44:     </div>
45: </body>
```

사용할 jQuery 플러그인(10)



■ 예제 16-04 실행 결과

The screenshot displays a web browser window with the address bar showing `localhost:8080/examples/ch16/16-04_dialog.html`. The browser's bookmark bar includes folders like "jquery 책쓰기", "머신러닝", "DB&BigData", "Development", "Utility", "보안", "감리", and "E-book".

The main content area shows a dialog box titled "새 연락처 추가하기" (Add New Contact). It contains three input fields labeled "이름" (Name), "전화번호" (Phone Number), and "전화번호" (Phone Number). At the bottom of the dialog are two buttons: "저장" (Save) and "취소" (Cancel).

The right side of the browser window shows the "Console" tab. It contains two lines of jQuery code, each highlighted with a black box:

```
> $("#dialog-new").dialog("close")
```

```
> $("#dialog-new").dialog("open")
```

The console also shows the corresponding HTML structure for the dialog box, including the `<div id="dialog-new" class="ui-dialog-content ui-widget-content" style="width: auto; min-height: 36px; max-height: none; height: auto;">...</div>` for both the close and open actions.

첫번째 예제 - 연락처 정보 페이징 기능(1)



■ 연락처 정보 페이징 기능

- 기본적인 HTML 마크업 틀 작성
- 예제 16-05

[예제 16-05]

```
01: <!DOCTYPE html>
02: <html lang="">
03: <head>
04: <meta charset="utf-8">
05: <title>16-final</title>
06: <link rel="stylesheet" type="text/css" href="jpaginate/style.css" />
07: <link rel="stylesheet" type="text/css" href="waitme/waitMe.css" />
08: <link rel="stylesheet" href="https://code.jquery.com/ui/1.12.0/themes
    /humanity/jquery-ui.css" />
09: <style>
10:     body { font-size: 75%; }
11:     input.text { margin-bottom:12px; width:95%; padding: .4em; }
12: </style>
13: <script src="http://code.jquery.com/jquery-3.1.0.js"></script>
14: <script src="https://www.jsviews.com/download/jsrender.js"></script>
15: <script src="https://code.jquery.com/ui/1.12.0/jquery-ui.js"></script>
16: <script src="waitme/waitMe.js"></script>
17: <script src="jpaginate/jquery.paginate.js"></script>
```

```
18: <script type="text/javascript">
19: $(document).ready(function() {
20:     // 1. 공통 변수, 옵션
21:
22:     // 2. 공통 함수
23:
24:     // 3. 이벤트 함수
25:
26:     // 4. 화면 초기화 코드
27:
28: });
29: </script>
30: </head>
31: <body>
32:
33: </body>
34: </html>
```

첫번째 예제 - 연락처 정보 페이징 기능(2)



- 예제 16-06 : HTML 마크업 추가
 - <body> 태그 내부에 다음 코드 추가

[예제 16-06]

```
01: <h2 class="ui-widget ui-header">연락처 리스트</h2>
02: <button role="button" id="btnNewContact"
03:     class="add ui-button ui-button ui-state-default ui-button-text-only">
04:     <span class="ui-button-text">새로운 연락처 추가</span>
05: </button>
06: <div id="output" style="text-align:center;width:630px;">
07:     <table id="contact_table" style="border:solid 1px black;">
08:         <thead>
09:             <tr style="background-color: #cb842e; color:white;">
10:                 <th style="width:60px;">번호</th>
11:                 <th style="width:90px;">이름</th>
12:                 <th style="width:100px;">전화번호</th>
13:                 <th style="width:250px;">주소</th>
14:                 <th style="width:130px;">기능</th>
15:             </tr>
16:         </thead>
17:         <tbody id="contactlist">
18:
19:         </tbody>
20:     </table>
21:     <div id="paging" ></div>
22: </div>
```

첫번째 예제 - 연락처 정보 페이징 기능(3)



- jsRender에서 사용할 뷰템플릿 추가
 - <head> 태그 안쪽 마지막에 다음 코드 추가

[예제 16-07]

```
01: <script id="tmpl_contact" type="text/x-jsrender">
02:   <tr>
03:     <td>{{:no}}</td>
04:     <td>{{:name}}</td>
05:     <td>{{:tel}}</td>
06:     <td>{{:address}}</td>
07:     <td>
08:       <button data-no="{{:no}}" role="button"
09:         class="update ui-button ui-button ui-state-default
10:           ui-button-text-only">
11:         <span class="ui-button-text">수정</span>
12:       </button>
13:       <button data-no="{{:no}}" role="button"
14:         class="delete ui-button ui-button ui-state-default
15:           ui-button-text-only">
16:         <span class="ui-button-text">삭제</span>
17:       </button>
18:     </td>
19:   </tr>
20: </script>
```

첫번째 예제 - 연락처 정보 페이징 기능(4)



■ 자바스크립트 코드 추가

- 페이지 번호, 페이지 사이즈는 클로저 활용
- 예제 16-08

```
04: // 1. 공통 변수, 옵션
05: var param = { pageno:1, pagesize:2 };
06: var totalPages = 0;
07:
08: var pageOption = {
09:     count : 1,
10:     start : 1,
11:     display : 10,
12:     border : true,
13:     border_color : "#333333",
14:     text_color : "#ffffff",
15:     background_color : "#cb842e",
16:     border_hover_color : "#333333",
17:     text_hover_color : "#000000",
18:     background_hover_color : "#cb842e",
19:     images : false,
20:     mouse : "press",
21:     onChange : function(page) {
22:         param.pageno = page;
23:         viewContactList();
24:     }
25: };
```

```
27: // 2. 공통 함수
28: var viewContactList = function() {
29:     $.get("/contact/list_long.jsp", param, function(data) {
30:
31:         $("#contactlist").empty();
32:         var tpl = $.templates("#tpl_contact");
33:         var str = tpl.render(data.contacts);
34:         $("#contactlist").html(str);
35:
36:         totalPages = Math.floor((data.totalCount-1)/ param.pagesize) + 1;
37:         pageOption.count = totalPages;
38:         pageOption.start = param.pageno;
39:         $("#paging").paginate(pageOption);
40:     });
41: };
42:
43: // 3. 이벤트 처리
44:
45: // 4. 화면 초기화 코드
46: viewContactList();
```

첫번째 예제 - 연락처 정보 페이징 기능(5)



■ 예제 16-08 작성후 실행 결과

연락처 리스트

새로운 연락처 추가

번호	이름	전화번호	주소	기능	
11	진구	010-2121-3320	서울시	수정	삭제
10	박해진	010-2121-3319	서울시	수정	삭제

First « 1 2 3 4 5 6 7 8 » Last

첫번째 예제 - waitMe 플러그인 적용(1)



■ waitMe 플러그인 적용

- 예제 16-08에서 의도적으로 list_long.jsp를 요청하도록 예제 작성
 - 의도적 지연 시간 발생 --> 이 시간동안 처리중임을 알릴 수 있도록 UI 작성
- 예제 16-09

[예제 16-09]

```
01: //다음 코드는 2. 공통 함수 영역에 배치한다.
02: var ajaxLoading = function(isLoading){
03:     if (isLoading == false) {
04:         $("body").waitMe("hide");
05:     } else {
06:         $("body").waitMe({
07:             effect: "ios",
08:             text: "처리중입니다.",
09:             bg: 'rgba(255,255,255,0.7)',
10:             color: '#000',
11:             source: 'waitme/img.svg',
12:         });
13:     }
```

```
14: };
15:
16: //다음 코드는 3. 이벤트 처리 영역에
17: $(document).ajaxStart(function() {
18:     ajaxLoading(true);
19: }).ajaxStop(function() {
20:     ajaxLoading(false);
21: });
```

첫번째 예제 - waitMe 플러그인 적용(2)



■ 예제 16-09 작성 후 실행 결과

연락처 리스트

새로운 연락처 추가

번호	이름	전화번호	주소	기능	
9	조진웅	010-2121-3318	서울시	수정	삭제
8	김지원	010-2121-3317	서울시	수정	삭제

First < 1 2 3 4 5 6 7 8 > Last

처리중입니다.

첫번째 예제 - 새로운 연락처 추가(1)



■ 새로운 연락처 추가

- jQuery UI Dialog 플러그인 사용
- 다음 코드를 <body> 태그 안쪽 마지막에 추가
- 예제 16-10

[예제 16-10]

```
01: <div id="input_dialog" title="연락처 추가/수정">
02:   <form id="f_contact">
03:     <label for="no">일련번호</label>
04:     <input type="text" name="no" id="no" class="text" disabled />
05:     <label for="name">이름</label>
06:     <input type="text" name="name" id="name" class="text" />
07:     <label for="tel">전화번호</label>
08:     <input type="text" name="tel" id="tel" class="text" />
09:     <label for="address">주소</label>
10:     <input type="text" name="address" id="address" class="text" />
11:   </form>
12: </div>
```

첫번째 예제 - 새로운 연락처 추가(2)



■ 자바스크립트 코드로 작성해야 할 기능

- 다이얼로그 플러그인의 기본 옵션을 적용하기
- #btnNewContact 버튼을 클릭했을 때 이벤트 처리
- 다이얼로그 화면에서 추가 버튼 클릭했을 때 처리
- 다이얼로그 화면에서 취소 버튼을 클릭했을 때 처리

■ 예제 16-11 작성

- 예제 16-05의 기본 틀에 작성할 코드의 영역을 표시해두었으므로 주석을 참조해 코드 추가

[예제 16-11]

```
01: //다음 코드는 1. 공통 변수, 옵션 영역에 배치
02: $("#input_dialog").dialog({
03:     autoOpen: false,
04:     modal : true
05: });
06:
```

```
07: //다음 코드는 2. 공통 함수 영역에 배치
08: var addContact = function() {
09:     var param = $("#form#f_contact").serialize();
10:     $.post("/contact/add.jsp", param, function(data) {
11:         if (data.status == "ok") {
12:             param.pageno = 1;
13:             viewContactList();
14:         } else {
15:             alert("연락처 추가 실패 : " + data.message);
16:         }
17:         $("#input_dialog").dialog("close");
18:     });
19: };
```

첫번째 예제 - 새로운 연락처 추가(3)



■ 예제 16-11 작성(이어서)

```
20:
21: var cancelInput = function() {
22:     $("#input_dialog").dialog("close");
23: };
24:
25: var initializeInputForm = function() {
26:     $("form#f_contact")[0].reset();
27: };
28:
29: //다음 코드는 3. 이벤트 처리 영역에 배치
30: $("#btnNewContact").click(function() {
31:     initializeInputForm();
32:     $("#input_dialog").dialog({
33:         modal : true,
34:         buttons : {
35:             "추가" : addContact,
36:             "취소" : cancelInput
37:         }
38:     });
39:     $("#input_dialog").dialog("open");
40: });
```

첫번째 예제 - 새로운 연락처 추가(4)



- 예제 16-11 작성 후 실행 결과

연락처 리스트

새로운 연락처 추가

번호	이름	전화번호
18	212	3
17	홍길동	010-1

First « 1 2 3 4 5 6

연락처 추가/수정

일련번호

이름

솔지

전화번호

010-8888-9999

주소

제주도

추가 취소

기능

수정 삭제

수정 삭제

첫번째 예제 - 연락처 수정,삭제 기능(1)



■ 연락처 수정,삭제 기능

- 이미 수정,삭제를 위해 예제 16-07을 통해 UI를 미리 작성해두었음

```
07:      <td>
08:          <button data-no="{{:no}}" role="button"
09:              class="update ui-button ui-button ui-state-default
10:                  ui-button-text-only">
11:              <span class="ui-button-text">수정</span>
12:          </button>
13:          <button data-no="{{:no}}" role="button"
14:              class="delete ui-button ui-button ui-state-default
15:                  ui-button-text-only">
16:              <span class="ui-button-text">삭제</span>
17:          </button>
18:      </td>
```

- 이벤트 위임 처리를 이용
 - 페이지를 이동할 때마다 버튼이 매번 만들어지기 때문
 - class가 update, delete 버튼을 클릭했을 때 부모 요소에서 이벤트 처리 위임

첫번째 예제 - 연락처 수정, 삭제 기능(2)



■ 예제 16-12

[예제 16-12]

```
01: //다음 코드는 2. 공통 함수 영역에 배치
02: var editContact = function(no) {
03:     $.get("/contact/contact.jsp", { no: no }, function(data) {
04:         if (data.errorMessage) {
05:             alert(data.errorMessage);
06:         } else {
07:             var dialog = $("#input_dialog");
08:             dialog.find("#no").val(data.no);
09:             dialog.find("#name").val(data.name);
10:             dialog.find("#tel").val(data.tel);
11:             dialog.find("#address").val(data.address);
12:             dialog.dialog({
13:                 autoOpen:true,
14:                 modal : true,
15:                 buttons : {
16:                     "수정" : updateContact,
17:                     "취소" : cancelInput
18:                 }
19:             });
20:         }
21:     });
22: };
23:
```

```
24: var updateContact = function() {
25:     var paramArray = $("#f_contact").serializeArray();
26:     paramArray.push({ name:"no", value:$("#no").val() });
27:     var param = $.param(paramArray);
28:     $.post("/contact/update.jsp", param, function(data) {
29:         if (data.status == "ok") {
30:             viewContactList();
31:         } else {
32:             alert("연락처 변경 실패 : " + data.message);
33:         }
34:         $("#input_dialog").dialog("close");
35:     });
36: };
37:
```


첫번째 예제 - 연락처 수정, 삭제 기능(3)



■ 예제 16-12(이어서)

```
38: var deleteContact = function(no) {
39:     if (confirm("정말로 삭제?") == true) {
40:         var param = {};
41:         param.no = no;
42:         $.post("/contact/delete.jsp", param, function(data) {
43:             if (data.status == "ok") {
44:                 viewContactList();
45:             } else {
46:                 alert("연락처 삭제 실패 : " + data.message);
47:             }
48:         });
49:     }
50: };
51:
52: //다음 코드는 3. 이벤트 처리 영역에 배치
53: $("#contactlist").on("click", "button.update", function() {
54:     var no = $(this).data("no");
55:     editContact(no);
56: });
57:
58: $("#contactlist").on("click", "button.delete", function() {
59:     var no = $(this).data("no");
60:     deleteContact(no);
61: });
```

첫번째 예제 - 연락처 수정, 삭제 기능(4)



- 예제 16-12(이어서)
 - 25행의 `serialize()`, `serializeArray()` 함수는 비활성화된 필드는 직렬화하지 않음
 - `serializeArray()` 메서드로 배열 객체를 생성한 다음 26 행에서 배열에 직접 추가 후 `$.param()` 메서드로 파라미터 문자열 생성
- 예제 16-12 작성후 실행 결과

연락처 리스트

새로운 연락처 추가

번호	이름	전화번호
14	소진	010-2121-3333
13	하니	010-2121-3333

First < 1 2 3 4 5 6 7

연락처 추가/수정

일련번호
14

이름
소진

전화번호
010-2121-7272

주소
서울시

수정 취소

기능
정삭제 정삭제

연락처 리스트

새로운 연락처 추가

번호	이름	전화번호	주소	기능
14	소진			
13	하니			

First < 1 2 3 4

localhost:8080 내용:

이 연락처를 정말로 삭제할까요?

☐ 이 페이지가 추가적인 대화를 생성하지 않도록 차단합니다.

확인 취소

기능
정삭제 정삭제

두번째 예제(1)



■ 다음 카카오의 이미지 검색 API를 이용한 이미지 뷰어

■ 부트스트랩과 jQuery 결합

■ 부트스트랩

- 트위터의 UI 디자이너 Mark Otto와 개발자 Jacob Thornton이 만들고 오픈소스화한 웹 개발용 프론트엔드(front-end) 프레임워크이다.
- 반응형 웹사이트를 위한 유동형 그리드 시스템(Grid System)과 HTML5/CSS3 기반의 다양한 UI 컴포넌트를 제공
- 보다 쉽고 빠르며, 현대적이면서도 일관된 웹 개발을
- 다양한 예제 모음(테마, 템플릿, 컴포넌트)이 갖추어져 있음
- 부트스트랩 참조 사이트
 - <http://getbootstrap.com/>
 - <http://bootstrapk.com/>
 - <http://www.w3schools.com/bootstrap/>

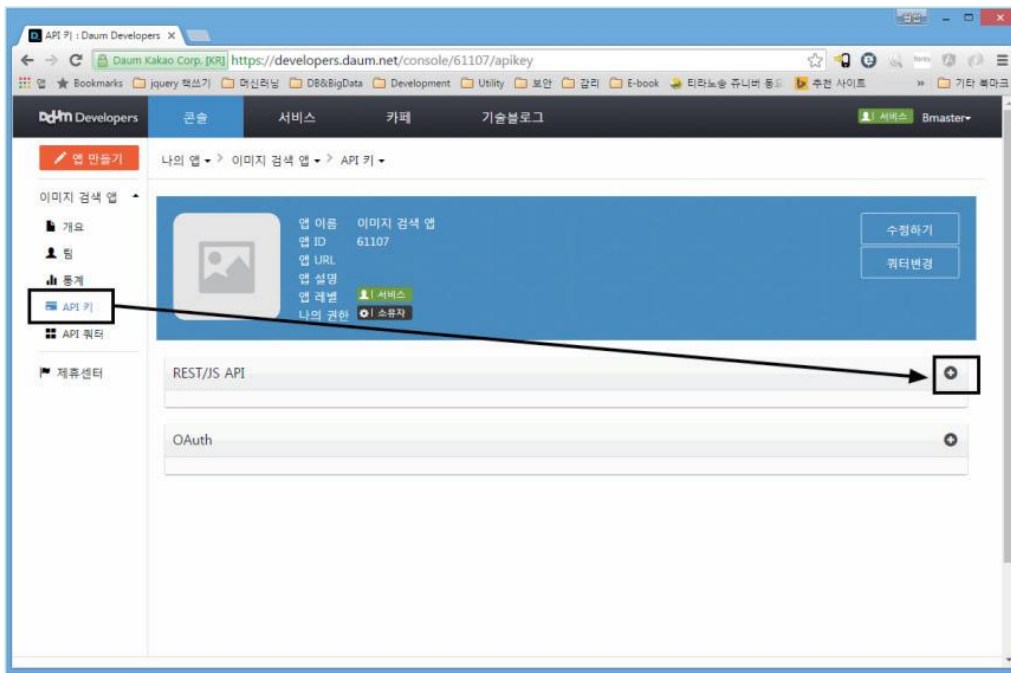
■ JSONP 기법 사용

두번째 예제(2)



■ 다음 개발자 센터 앱 등록

■ <https://developers.daum.net/>



REST/JS APIKey 발급 및 Whitelist 입력

Daum Developers 에서는 API키 발급시, 허용된 시스템에서만 호출가능하도록 하고 있습니다. apikey를 보호하는 차원에서 API가 호출될 플랫폼을 한정짓고, 해당 플랫폼의 정보를 입력받고 있습니다.

플랫폼 선택

☒ 웹브라우저 ☐ 서버 ☐ 안드로이드 ☐ iOS
☐ 모든 플랫폼

WhiteList 입력

Referer 입력

선택한 플랫폼과 1:n 매칭으로 사용할 수 있습니다.
줄바꿈으로 구분하여 추가 입력할 수 있습니다.

완료

취소


두번째 예제(3)



■ 다음 개발자 센터 앱 등록(이어서)

■ 발급된 API KEY

나의 앱 > 이미지 검색 앱 > API 키



앱 이름 이미지 검색 앱
앱 ID 61107
앱 URL
앱 설명
앱 레벨 서비스
나의 권한 소유자

수정하기
쿼터변경

REST/JS API

API 키	69265a1de3f2d73ea17f3fa647209f4d
플랫폼	웹브라우저 http://localhost:8080
생성일	2016/06/30 12:53:11
수정일	2016/06/30 12:53:11
마지막 수정자	수정한 사람 없음

편집 삭제

두번째 예제(4)



❏ 이미지 검색 API 문서 살펴보기

- <https://developers.daum.net/services/apis/search/image>

The screenshot shows the Daum Developers website with the '서비스' (Services) tab selected. The left sidebar lists various API categories, and the main content area displays the '이미지 검색' (Image Search) API documentation. The documentation includes a description of the API, the request URL, and a table of request parameters.

이미지 검색
#api #reference

이미지 검색 API는 모든 이미지 서비스에 대한 검색 결과를 외부 개발자 및 사용자에게 XML, JSON(P) 형식으로 전달하는 API 서비스입니다.
이미지의 유용한 정보를 자신의 블로그나 자사 웹 서비스에 제공할 수 있게 해줍니다.

요청 주소 (Request URL)

APIKey GET `https://apis.daum.net/search/image`

요청 변수 (Request Parameters)

파라미터	타입	설명	값의 범위	예제
q	string	검색을 원하는 질의어		daum

두번째 예제(5)



■ API 문서 참조

- 요청 파라미터 정리
- 결과 형식 파악

[요청 URL 예]

`https://apis.daum.net/search/image?apikey={{apikey}}&q=세종대왕&output=json`

[표 16-01 : 주요 요청 파라미터]

파라미터	값	설명
q	문자열	검색어
result	기본값: 10, 1~20까지 지정 가능	한 페이지에 출력될 결과 수, <code>pagesize</code> 의 의미
pageno	기본값: 1	페이지 번호
sort	date:날짜순 accu:추천순	정렬 기준
output	json, xml	결과 형식
callback		JSONP를 지원하기 위한 요청 파라미터

```
{
  - channel: {
    result: "21",
    pageCount: "3995",
    title: "Search Daum Open API",
    totalCount: "254193",
    description: "Daum Open API search result",
  - item: [
    - {
      pubDate: "20140414061442",
      title: "조선왕 수명 47세, 영조<b>대왕</b> 83세...비결은 왕성한 신장의 정기",
      thumbnail: "https://search2.kakaocdn.net/argon/130x130_85_c/J20Y65e9gp5",
      cp: "269393",
      height: "402",
      link: "http://blog.daum.net/jinikang/17205297",
      width: "400",
      image: "http://image.chosun.com/sitedata/image/201404/02/2014040202823_1.jpg"
    },
    + {...},
    + {...},
    + {...},
    + {...},
    + {...},
    + {...},
    + {...},
    + {...},
    + {...}
  ],
  lastBuildDate: "Thu, 30 Jun 2016 14:47:30 +0900",
  link: "http://dna.daum.net/apis",
  generator: "Daum Open API"
}
```

두번째 예제(6)



■ 부트스트랩을 이용한 UI 디자인

■ 부트스트랩 템플릿

- <http://startbootstrap.com/template-overviews/4-col-portfolio/> 템플릿
- 템플릿을 내려받은 후 압축을 풀어 css, font, js 폴더를 복사하여 프로젝트 폴더에 붙여넣는다.

■ 라이트박스 플러그인

- <https://github.com/ashleydw/lightbox>
- 다운로드 후 압축 풀고 dist 폴더의 css 파일과 js 파일을 구분하여 각 프로젝트의 css, js 폴더에 붙여넣는다.

■ jsRender 플러그인

- 첫번째 예제에서 이미 사용해본바 있음.

두번째 예제(7)



■ 예제 16-13 : 기본틀 작성

[예제 16-13]

```
01: <!DOCTYPE html>
02: <html lang="">
03: <head>
04: <meta charset="utf-8">
05: <meta http-equiv="X-UA-Compatible" content="IE=edge">
06: <meta name="viewport" content="width=device-width, initial-scale=1">
07: <title>이미지 뷰어 예제</title>
08: <link rel="stylesheet" href="css/bootstrap.min.css">
09: <link rel="stylesheet" href="css/ekko-lightbox.min.css">
10: <style>
11:     body { padding-top: 70px; }
12:     .portfolio-item { margin-bottom: 25px; }
13:     footer { margin: 50px 0; }
14: </style>
15: <script src="js/jquery.js"></script>
16: <script src="js/bootstrap.min.js"></script>
17: <script src="js/ekko-lightbox.min.js"></script>
18: <script src="https://www.jsviews.com/download/jsrender.js"></script>
```

```
19: <script type="text/javascript">
20: $(document).ready(function() {
21:
22: });
23: </script>
24: </head>
25: <body>
26:
27: </body>
28: </html>
```

두번째 예제(8)



- 예제 16-14 : HTML 마크업 추가
 - <body> 태그 안쪽에 작성

[예제 16-14]

```
01: <nav class="navbar navbar-inverse navbar-fixed-top" role="navigation">
02:   <div class="container">
03:     <div class="navbar-header">
04:       <button type="button" class="navbar-toggle"
05:         data-toggle="collapse"
06:         data-target="#bs-example-navbar-collapse-1">
07:         <span class="sr-only">Toggle navigation</span>
08:         <span class="icon-bar"></span>
09:         <span class="icon-bar"></span>
10:         <span class="icon-bar"></span>
11:       </button>
12:       <a class="navbar-brand" href="#">이미지 검색</a>
13:     </div>
14:     <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
15:       <ul class="nav navbar-nav">
16:         <li>
17:           <a href="https://developers.daum.net/services/apis/search/image">
18:             DAUM 이미지 검색 페이지</a>
19:         </li>
20:         <li>
21:           <a href="https://developers.daum.net/">
22:             DAUM Developers Center</a>
23:         </li>
24:         <li>
25:           <a href="https://en.wikipedia.org/wiki/JSONP">JSONP Wiki</a>
26:         </li>
27:       </ul>
28:     </div>
29:   </div>
30: </nav>
```

```
31: <!-- Page Content -->
32: <div class="container">
33:   <div class="row">
34:     <div id="navbar" class="navbar-collapse navbar-form navbar-left">
35:       <div class="form-group col-lg-12">
36:         <input id="keyword" type="text" placeholder="검색어"
37:           class="form-control">
38:         <button id="inquiry" class="btn btn-success form-control">
39:           검색</button>
40:       </div>
41:     </div>
42:     <div id="imagelist" class="row">
43:
44:     </div>
45:
46:   <footer>
47:     <div class="row">
48:       <div class="navbar-collapse navbar-form navbar-left">
49:         <div class="form-group col-lg-12">
50:           <button type="submit" id="getNext"
51:             class="btn btn-primary">
52:             다음 20개 가져오기</button>
53:         </div>
54:       </div>
55:     </footer>
56:   </div>
```

두번째 예제(9)



■ 예제 16-15 : 뷰템플릿 작성

[예제 16-15]

```
<script id="image_template" type="text/x-jsrender">
  <div class="col-md-2 portfolio-item">
    <a data-toggle="lightbox" data-gallery="multis"
      data-title="{{:title}}" href="{{:image}}">
      
    </a>
  </div>
</script>
```

두번째 예제(10)



■ 예제 16-16 : 자바스크립트 코드 작성

[예제 16-16]

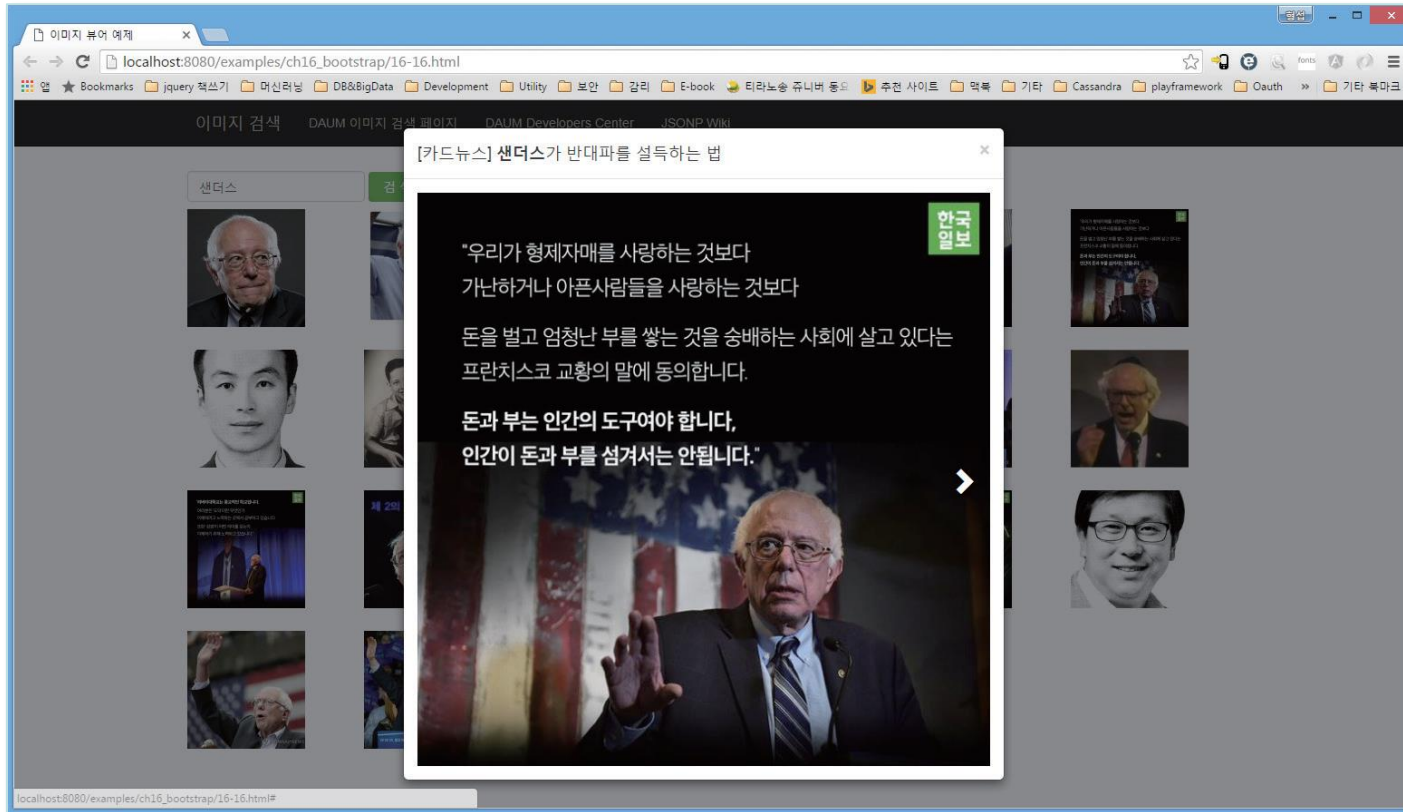
```
01: <script type="text/javascript">
02: $(document).ready(function() {
03:     var url = "https://apis.daum.net/search/image?callback=?";
04:     var param = {
05:         pageno:1, result:20, output:"json", sort : "accu",
06:         apikey : "[발급받은 API 키]"
07:     };
08:     //검색용 함수: JSONP 기법 이용
09:     var searchImages= function() {
10:         $.getJSON(url, param, function(data) {
11:             var tpl = $.templates("#image_template");
12:             var str = tpl.render(data.channel.item);
13:             $("#imagelist").append(str);
14:         });
15:     };
16:     //다음 20개 가져오기를 클릭했을 때의 이벤트 처리
17:     $("#getNext").on("click", function() {
18:         param.pageno++;
19:         searchImages();
20:     });
```

```
21:     //'검색' 버튼을 클릭했을 때의 이벤트 처리
22:     $("#inquiry").on("click", function() {
23:         $("#imagelist").empty();
24:         param.pageno = 1;
25:         param.q = $("#keyword").val();
26:         searchImages();
27:     });
28:     //검색어 입력을 위한 텍스트박스에서 엔터키를 눌렀을 때도 검색이 가능하도록 함.
29:     $("#keyword").on("keyup", function(e) {
30:         if (e.keyCode == 13) {
31:             $("#inquiry").trigger("click");
32:         }
33:     });
34:     //이벤트 위임 처리를 통해 검색된 썸네일 이미지를 클릭했을 때 라이트박스 처리
35:     $("#imagelist").on("click", "div.portfolio-item a", function(e) {
36:         e.preventDefault();
37:         $(this).ekkoLightbox();
38:     });
39: });
40: </script>
```

두번째 예제(11)



■ 예제 16-16 작성후 실행 결과



두번째 예제(12)



- 35행의 코드
 - 이벤트 위임 처리를 통해 라이트박스 기능 제공
- 원본 이미지가 사라지거나 접근을 허용하지 않는 경우가 있음.
- 다음 이미지 검색 API는 3번째 페이지까지만 조회 가능함.
 - 다음 20개 가져오기 버튼은 두번만 가능하다는 의미