게시판의 사용과 커스터마이징

게시판의 종류와 사용방법
게시판을 Pro Page에 적용하는 방법
문의형 게시판 커스터마이징하기
공지형 게시판 커스터마이징하기
소개형 게시판 커스터마이징하기
게시판 커스터마이징 방법

게시판의 종류와 사용방법

게시판의 종류

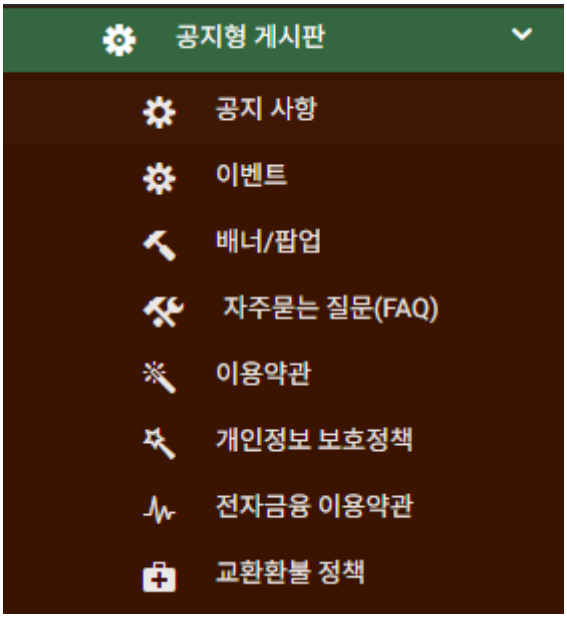
1. 표준 게시판

공지형 게시판
공지사항
이벤트
영업문의
영업문의
배너/팝업
FAQ(자주하는질문)
이용약관
개인정보 보호정책
마케팅 수신 동의
전자금융 이용약관
교환환불 정책
소개형 게시판
회사소개
제품소개
특허/인증허
포트폴리오
뉴스
문의형 게시판
티켓-1대1문의
공개질문
1대1문의

게시판 관리

- 게시판 구분(공지/소개/문의)





게시판/블로그 관리
게시판 구분(공지/소개/문의)
공지형 게시판
소개형 게시판
문의형 게시판
블로그 관리

모든 게시판에 대한 정보를 불러옵니다.

구분코드, PostCode, 상태, 종류설명, 카테고리 등이 포함되어 있습니다.

모든 게시판들은 psot_type_id를 통해 구분됩니다.

- 공지형 게시판

공지형 게시판
공지 사항
이벤트
배너/팝업
자주묻는 질문(FAQ)
이용약관
개인정보 보호정책
전자금융 이용약관
교환환불 정책

공지형 게시판의 자식 메뉴들이 노출됩니다.

각 메뉴에서는 해당되는 게시글들을 수정 /삭제 /복사 등 게시글 관리가 가능합니다.

공지형 게시판의 para_name은 다음과 같습니다.

분류	para_name(조회)	para_name(등록)
공지사항	notice-input	notice



이벤트	event-input	event
배너/팝업	banner-input	banner
자주묻는 질문(FAQ)	faq-input	faq
이용약관	policy-input	policy
개인정보 보호정책	privacy-input	privacy
전자금욱 이용약관	e-finance-input	e-finance
교환환불 정책	refund-input	refund

소개형 게시판

게시판/블로그 관리
게시판 구분(공지/소개/문의)
공지형 게시판
소개형 게시판
회사소개
제품소개
특허/인증서
포트폴리오
뉴스

소개형 게시판의 자식 메뉴들이 노출됩니다.

기능적인 부분은 위와 같습니다.

분류	para_name (조회)	para_name (등록)
회사소개	intro-input	intro
제품소개	production-input	production
특허/인증서	certificatate-input	certificatate

□□□□	portfolio-input	portfolio
□□	news-input	news

- □□□ □□□



para_name

□□	para_name (□□)	para_name (□□)
□□□□	item-inquiry-input	item-inquiry
□□ □□	contactus-input	contactus
1□1 □□	1to1-input	1to1
□□ □□	standard-input	standard

□ □□□□□ □□□ □□□□□ pro □□□□ □□□□ □□□□□.

□□□□ Pro Page□ □□□□ □□

> 1□1□□ (□□□ □□□)□ pro page□ □□□□

[□□□ □□□ - 1□1 □□ □□□□ □□]

1. Controller □□

`pro` [`Controller` □ [`dabory/themes/pro/app/Http/Controller/etc` [`OneToOneController.php` □ □□□□ □□□□ □□□□□.

- □□ API □□ □□

□□□ API `App\Services\CallApiService` □ □□□□□.

```
class OneToOneController extends Controller
{
    private $callApiService;



    public function __construct(CallApiService $callApiService)
    {
        $this->callApiService = $callApiService;
    }



    public function list()
    {
        $limit = (int)request('limit', 12);
        $page = (int)request('page', 1);

        $oneToOnePage = $this->callApiService->callApi([
            'url' => 'list-type1-page',    // api□ □□□ url □□
            'data' => [                    // list-type1-page□ □□□ data
                'QueryVars' => [           // api □□□□ Query□ □□□ □ □□□ □□□ □□□
```

```
                'QueryName' => 'pro:my-page/post-list',   // pro/.../my-page/post-list/ □□□□ □□ □□ □□□ □□□.
                'SimpleFilter' => "post_code='1to1'",     // where □□ □□
                'SubSimpleFilter' => "image_type = 'thumb'",
                'IsntPagination' => false     // pagination □□
            ],
            'ListType1Vars' => [
                'OrderBy' => request('sort', 'mx.created_on desc')
            ],
            'PageVars' => [
                'Limit' => $limit,
                'Offset' => ($page - 1) * $limit
            ]
        ]
    ]);


//    dump($oneToOnePage);


    $oneToOnePage['Page'] = new LengthAwarePaginator($oneToOnePage['Page'],
$oneToOnePage['PageVars']['QueryCnt'],
        $limit, $page, ['path' => request()->url()]);
    // dump($oneToOnePage);
    return view('views.etc.1to1-list', compact('oneToOnePage'));
}


public function store()
{
    $validator = Validator::make(request()->all(), [
        'post_title' => 'required',
        'post_contents' => 'required',
        'pc1' => 'required',
        'pc2' => 'required|email'
    ]);


    if ($validator->fails()) {
        notify()->error(_e('Action failed'), 'Error', 'bottomRight');
        return redirect()->back()
```

```
            ->withErrors($validator)
            ->withInput();
        }


        $response =  $this->callApiService->callApi([
            'url' => 'post-act',
            'data' => [
                'Page' => [
                    [
                        'Id' => 0,
                        'PostTitle' => request('post_title'),
                        'PostContents' => request('post_contents'),
                        'PostTypeId' => 6,
                        'Pc1' => request('pc1'),
                        'Pc2' => request('pc2'),
                        'Status' => '2',
                        'MemberId' => session('member')['MemberId'],
                    ]
                ]
            ],
        ]);


        if ($this->callApiService->verifyApiError($response)) {
            notify()->error($response['body'], 'Error', 'bottomRight');
            return redirect()->back();
        }


//        notify()->success(_e('Action completed'), 'Success', 'bottomRight');
        return redirect()->route('1to1.list');
    }


    public function show($id)
    {
        $listType1Book = $this->callApiService->callApi([
            'url' => 'list-type1-book',
            'data' => [
```

```
        'Book' => [
            [
                'QueryVars' => [
                    'QueryName' => 'pro:my-page/post-details',
                    'SimpleFilter' => "post_code='1to1' and mx.id = $id",
                    'IsntPagination' => true,
                ],
                'PageVars' => [
                    'Limit' => 1
                ]
            ],
            [
                'QueryVars' => [
                    'QueryName' => 'pro:my-page/post-details-prenext',
                    'SimpleFilter' => "mx.id = (select max(id) from pro_post where id < ${id} and post_type_id =
6)",
                    'SubSimpleFilter' => "",
                    'IsntPagination' => true,
                ],
                'PageVars' => [
                    'Limit' => 1
                ]
            ],
            [
                'QueryVars' => [
                    'QueryName' => 'pro:my-page/post-details-prenext',
                    'SimpleFilter' => "mx.id = (select min(id) from pro_post where id > ${id} and post_type_id =
6)",
                    'SubSimpleFilter' => "",
                    'IsntPagination' => true,
                ],
                'PageVars' => [
                    'Limit' => 1
                ]
            ],
        ]
    ]
]);
```



```
    //    dd($listType1Book);
        if ($this->callApiService->verifyApiError($listType1Book)) {
            notify()->error($listType1Book['body'], 'Error', 'bottomRight');
            return redirect()->back();
        }



//        dump($listType1Book);
        return view('views.etc.1to1-details', compact('listType1Book'));
    }
}
```

1대1문의 구현을 위한 Controller 예시

이 `list-type1-page` URL로 data가 요청되[`weberp-queries` 에서 해당 URL의 QueryVars에 작성해준 조건에 해당하는 SQL 파일을 찾고, 데이터로 쿼리 [

응답 처리 및 **View** 반환

API로부터 응답을 받아 변수에 담고, View 페이지에서 필요한 데이터를 가공하거나 변환하여 반환합니다. View 페이지에서는 반환받은 배열 데이터를 통해 페이지 작업을 수행할 수 있습니다.

TEST

Controller와 view 페이지 작업이 완료되었다면 erp/pro 각각의 페이지에서 테스트를 진행합니다.

예제에서 진행했던 1대1문의 게시판을 확인해보겠습니다.

pro 페이지에 만들어준 문의하기 게시판에서 문의글을 작성한 뒤

게시판/블로그 관리
게시판 구분(공지/소개/문의)
공지형 게시판
소개형 게시판
문의형 게시판
상품문의
상담 신청
1대1 문의
공개 질문

erp 페이지로 이동하여 1대1문의 게시판을 확인합니다.



위와같이 문의내용이 잘 들어왔습니다.

참고사항

erp에 게시판 설치하기

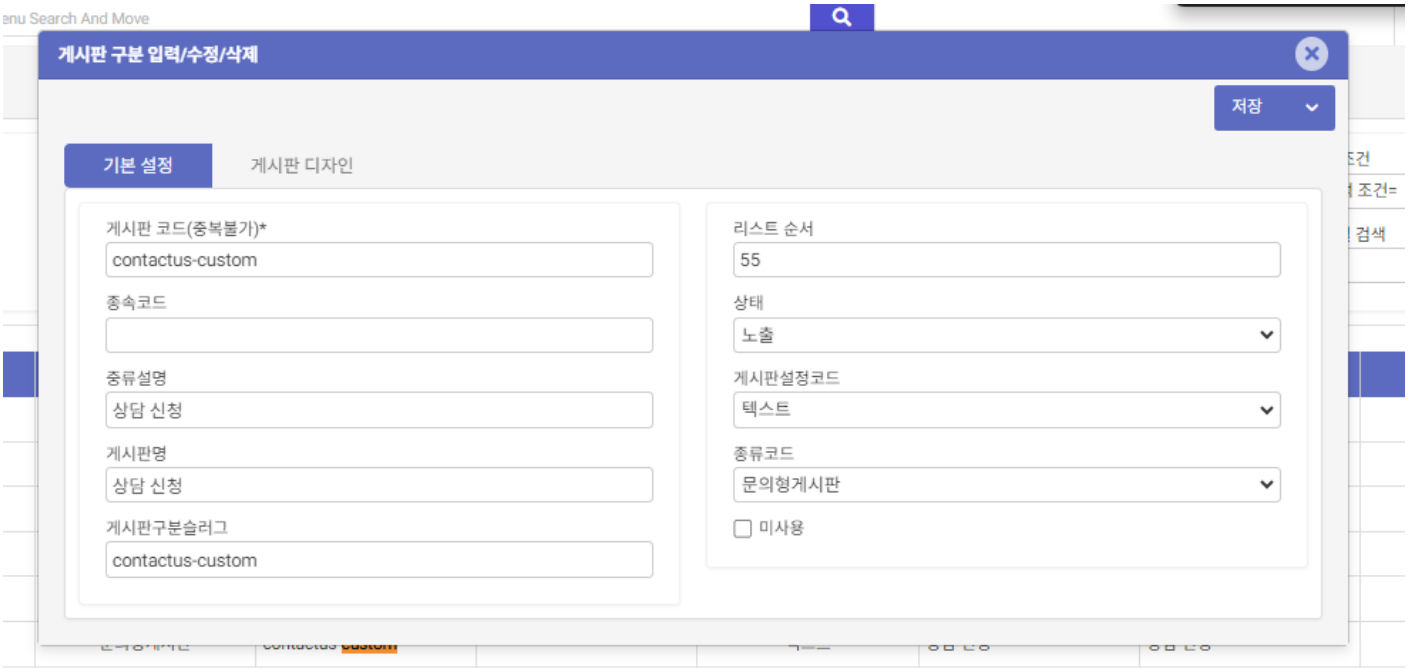
erp 게시판 커스터마이징

게시판 커스터마이징 방법



게시판을 커스터마징 하기 위해서는

erp 게시판 구분 메뉴에서 게시판 구분 등록을 통해 새로운 커스터마징 게시판을 등록해야합니다.



게시판 구분 등록 예시

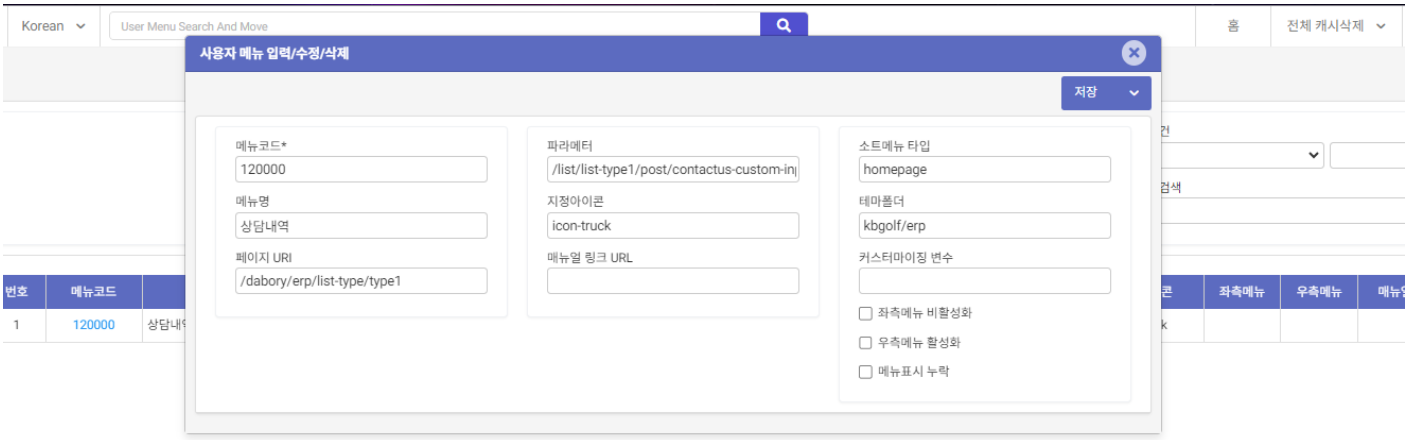
번...	일련번호	구분코드	PostCode	BrandCode	SetupCode	종류설명	게시판(블로그)카테고리명	상태	사용중지	세팅 Json
6	34	소개형게시판	news		분류 필요	뉴스	뉴스	노출		
5	35	소개형게시판	sise		텍스트	커스터마이징	회원권 시세	노출		{"Design":{"PcTitleLeng":0,"...
4	2	문의형게시판	ticket		분류 필요	티켓-1대1문의	티켓-1대1문의	숨김		
3	4	문의형게시판	qna		분류 필요	공개질문	공개질문	숨김		
2	6	문의형게시판	1to1		분류 필요	1대1문의	1대1문의	숨김		
1	55	문의형게시판	contactus-custom		텍스트	상담 신청	상담 신청	노출		{"Design":{"PcTitleLeng":0,"...

□□□□ □□□ □□ □□□□□.

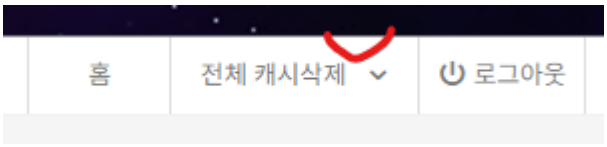


□□□□ □□□□ □□□ □□□ □□ □□□□□ □□□

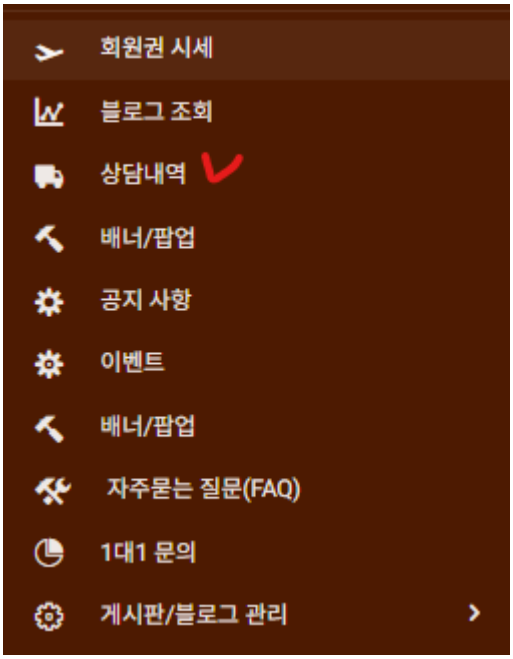
□□ □□□ API □□ - □□□ □□□ □□□□□.


Korean
User Menu Search And Move
홈
전체 캐시삭제
사용자 메뉴 입력/수정/삭제
저장
메뉴코드*
120000
메뉴명
상담내역
페이지 URI
/dabory/erp/list-type/type1
파라메터
/list/list-type1/post/contactus-custom-in
지정아이콘
icon-truck
매뉴얼 링크 URL
소트메뉴 타입
homepage
테마폴더
kbgolf/erp
커스터마이징 변수
좌측메뉴 비활성화
우측메뉴 활성화
메뉴표시 누락
번호
메뉴코드
1
120000
좌측메뉴
우측메뉴

□□ □□□ □□ □□□ □□□ □□□□□□.


홈
전체 캐시삭제
로그아웃

□□ □□□□□ □□□ □□□ □□□ □□□ □□□.


회원권 시세
블로그 조회
상담내역
배너/팝업
공지 사항
이벤트
배너/팝업
자주묻는 질문(FAQ)
1대1 문의
게시판/블로그 관리

□□□ □□□ □□□ □□□□□□.

□□ □□□□ □□□□ crud□ □□□□□ □□□

erp□ □□□□□ □□□□□□. □□□□□ □□□ □□□ □□□□ □□□□ □□□ □□□□□ □□□.

□□ □□□ □□□□ □□□□□ □□□□ contatctus-custom-input.json □□□ □□□□ □□□ □□□□□□.

□□□ □□□ □□□□.

```
{
   "//list-type-1": "simple and plain list by query of table join",



   "General": {
      "Title": "□□□□ □□",
      "PageApi": "list-type1-page",
      "ActApi": "post-act"
   },



   "QueryVars": {
      "QueryName": "kbgolf::post/contactus-custom-input",
      "FilterName": "",
      "FilterValue": "",
      "FilterDate": "mx.official_date",
      "SimpleFilter": "pt.post_code = 'contactus-custom'"
   },



   "HeadSelectOptions": [
      {
         "Value": "xls-report",
         "Caption": "(□□)□□ □□□□"
      },
      {
         "Value": "custom-xls-report",
         "Caption": "□□ □□□□",
         "Component": "dummy",
         "Parameter": "/list/list-type1/download/contactus-custom",
         "ThemeDir": "/kbgolf/erp"
      },
      {
         "Value": "new",
         "Caption": "□□□□ □□"
```

```
    },
    {
        "Value": "multi-delete",
        "Caption": "□□□□ □□ □□"
    }
],


"//FormVars": "FormVars[0]-caption, FormVars[1]-searchPopup",
"FormVars": [
    {
        "ListButton": "□□",
        "DateRange": "",
        "DateNavi": "□□ | (□□□□) | □□",
        "Date": "□□ □□□",
        "From": "□□",
        "To": "□□",
        "FirstRange": "",
        "SecondRange": "",
        "ThirdRange": "",
        "FourthRange": "",
        "AddTotalLine": "",
        "SelectPopup": "□□ □□(□□□ □□)",
        "MultiPopup": "",
        "ChartPopup": "",
        "DownloadList": "",
        "ShowOnlyClosed": "",
        "Balance": "",
        "OrderBy": "□□ □□",
        "FilterOption": "□□ □□",
        "SimpleOption": "□□□ □□"
    }
],


"FilterSelectOptions": [
    {
        "Value": "",
        "Caption": "=□□ □□="
    },
```

```
    {
      "Value": "mx.pc1",
      "Caption": "□□□"
    },
    {
      "Value": "mx.pc5",
      "Caption": "□□□□"
    },
    {
      "Value": "mx.pc3",
      "Caption": "□□ □□"
    },
    {
      "Value": "mx.pc4",
      "Caption": "□□□ □□"
    }
],


"SimpleSelectOptions": [
    {
      "Value": "",
      "Caption": "□□"
    },
    {
      "Value": "mx.pc7='□□□□'",
      "Caption": "□□□□"
    },
    {
      "Value": "mx.pc7='□□□□'",
      "Caption": "□□□□"
    }
],
"//ListType1RangeVars": "[0]-filter, [1]-component, [2]-parameter",
"ListType1RangeVars": [
    {
      "FirstRange": "",
      "SecondRange": "",
      "ThirdRange": "",
      "FourthRange": ""
```

```
    },
    {
      "FirstRange": "",
      "SecondRange": "",
      "ThirdRange": "",
      "FourthRange": ""
    },
    {
      "FirstRange": "",
      "SecondRange": "",
      "ThirdRange": "",
      "FourthRange": ""
    }
],


"DateNaviOptions": [
    {
      "Value": "day",
      "Caption": "□"
    },
    {
      "Value": "week",
      "Caption": "□"
    },
    {
      "Value": "month",
      "Caption": "□"
    },
    {
      "Value": "quarterly",
      "Caption": "□□"
    },
    {
      "Value": "semiannual",
      "Caption": "□□"
    },
    {
      "Value": "year",
      "Caption": "□"
```

```
    },
    {
        "Value": "all",
        "Caption": "□□"
    }
],


"SelectPopupOptions": [
    {
        "Caption": "□□□□ □□",
        "Component": "kbgolf::popup.popup-form1.form-a.post.contactus-custom-form",
        "Parameter": "kbgolf::/popup/popup-form1/form-a/post/contactus-custom"
    }
],


"// OrderByOptions": "OrderBy Options//□□□ Value□ Default",
"OrderByOptions": [
    {
        "Value": "mx.created_on desc",
        "Caption": "□□□□ □□"
    },
    {
        "Value": "mx.created_on asc",
        "Caption": "□□□□ □□"
    }
],


"//BalanceOptions": "Show OnlyBalaceRemained or All",
"BalanceOptions": [
    {
        "Value": "",
        "Caption": "□□ □□"
    },
    {
        "Value": "c10 > 0",
        "Caption": "□□□□ □□"
    }
```

```
],


"DisplayVars": {
    "IsListFirst": true,
    "IsExcelColumn": true,
    "IsSelectPopupHidden": true,
    "IsC1Popup": "1",
    "InitLines": 10,
    "HeadHeight": "140",
    "BodyHeight": "570"
},


"ThumbContainerVars": {
    "ListWidth": 60,
    "ListHeight": 50
},


"// ListVars": "ListVars[0]-caption, ListVars[1]-size(px), ListVars[2]-align->right(number) left(left)",
"ListVars": [
    {
        "$Radio": "$Radio",
        "$Check": "$Check",
        "No": "□□",
        "C1": "□□□",
        "C2": "□□□",
        "C3": "□□□□",
        "C4": "□□□ □□",
        "C5": "□□□□",
        "C6": "□□□□",
        "C7": "□□□□",
        "C8": "□□"
    },
    {
        "$Radio": "2",
        "$Check": "2",
        "No": "2",
        "C1": "5",
```

```
            "C2": "4",
            "C3": "4",
            "C4": "5",
            "C5": "5",
            "C6": "5",
            "C7": "7",
            "C8": "4"
        },
        {
            "$Radio": "center",
            "$Check": "center",
            "No": "center",
            "C1": "center",
            "C2": "center",
            "C3": "center",
            "C4": "center",
            "C5": "center",
            "C6": "center",
            "C7": "center",
            "C8": "center"
        }
    ]
}
```

이제 pro 페이지와 연동하여 상담내역을 관리할 수 있는 게시판이 구성됐습니다.
pro 페이지 적용은 아래 링크를 통해 체크할 수 있습니다.

참고사항

pro 페이지 적용방법