

# 📁-Frontend 📁 📁 📁

- 📁 📁 📁 📁 📁
  - `main_api` 📁 📁 📁
- 📁 📁 📁
  - 📁 📁 📁
  - 📁 Pro Page 📁 📁 📁
  - 📁 📁 📁
  - 📁 📁 📁
  - 📁 📁 📁
  - 📁 📁 📁
- 📁 📁 📁
- `cache-key-pair`?



# main\_api 部署

【部署】

1. [ussh](#) 部署

- 1) Front 部署 query 服务
- 2) 部署 weberp-go API 服务

部署前请确保环境已就绪。  
\*\* 1) 部署 query 服务

(1) 通过 git bash 部署 main\_api 服务 (ussh 部署)

./ussh dbr02 main\_api

(2) 通过 command mode 部署 shell script

main\_api-query-gitpp-weberp-go-run

部署前准备

- 部署 query 服务 main\_api/weberp-queries gitpp
- Redis Memory DB 部署 Flush
- port-grab-killer 18080 部署 18080 部署 weberp-go 部署
- weberp-go 部署 weberp-go API 部署

部署前请确保环境已就绪 query 服务 redis memory cache 部署 file 部署



□□□□ □□□ □□□□□□

□□□□□□ □□□□□ □□□□□□□

□□□□□ □□

1. □□ □□□

- □□□ □□□
  - □□□□
  - □□□
  - □□□□
  - □□□□
  - □□/□□
  - FAQ(□□□□□□)
  - □□□□
  - □□□□ □□□□
  - □□□ □□ □□
  - □□□□ □□□□
  - □□□□ □□
- □□□ □□□
  - □□□□
  - □□□□
  - □□/□□□
  - □□□□□
  - □□
- □□□ □□□
  - □□-1□1□□
  - □□□□
  - 1□1□□

□□□□ □□

- □□□ □□(□□/□□/□□)

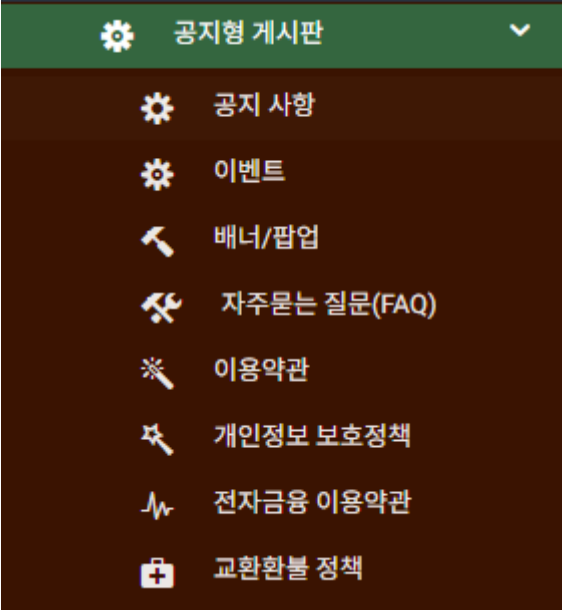


게시판 구분, PostCode, 게시판, 게시판, 게시판.

게시판, PostCode, 게시판, 게시판, 게시판, 게시판, 게시판.

게시판 psot\_type\_id 게시판.

- 게시판



게시판, 게시판, 게시판, 게시판.

게시판, 게시판, 게시판, 게시판, 게시판, 게시판, 게시판.

게시판 para\_name 게시판.

게시판	para_name(게시판)	para_name(게시판)
게시판	notice-input	notice

행사	event-input	event
배너/배너	banner-input	banner
자주 묻는 질문(FAQ)	faq-input	faq
정책	policy-input	policy
개인정보 보호	privacy-input	privacy
전자금융	e-finance-input	e-finance
환불	refund-input	refund

## - 메뉴 구성



게시판 메뉴는 공지, 소개, 문의로 구성됩니다.

소개형 게시판은 회사소개, 제품소개, 특허/인증서, 포트폴리오, 뉴스로 구성됩니다.

메뉴명	para_name (코드)	para_name (코드)
소개	intro-input	intro
제품	production-input	production
배너/배너	certificatate-input	certificatate

포트폴리오	portfolio-input	portfolio
뉴스	news-input	news

- 게시판 관리

게시판/블로그 관리

게시판 구분(공지/소개/문의)

공지형 게시판

소개형 게시판

문의형 게시판

상품문의

상담 신청

1대1 문의

공개 질문

para\_name

이름	para_name (이름)	para_name (이름)
상품문의	item-inquiry-input	item-inquiry
문의하기	contactus-input	contactus
1:1 문의	1to1-input	1to1
일반문의	standard-input	standard

이름, para\_name, para\_name pro, para\_name, para\_name.



# 1111 Pro Page 1111 11

1111 (111 111) 11 pro page 1111

[111 111 - 111 11 1111 11]

## 1. Controller 11

pro 11 Controller 11 dabory/themes/pro/app/Http/Controller/etc 11 OneToOneController.php 11 1111 1111 11111.

### - 11 API 11 11

111 API 11 App\Services\CallApiService 11 11111.

```
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 111 url 11
```

```

'data' => [ // list-type1-page 数据
  'QueryVars' => [ // api 接口 Query 参数
    'QueryName' => 'pro:my-page/post-list', // pro/.../my-page/post-list/ 接口 名称
    'SimpleFilter' => "post_code='1to1'", // where 条件
    'SubSimpleFilter' => "image_type = 'thumb'",
    'IsntPagination' => false // pagination 是否
  ],
  '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'));
}
}
```

1to1 Controller

list-type1-page URL data weberp-queries URL QueryVars SQL ,

## View

API, View . View .

## TEST

Controller view erp/pro .

1to1 .

pro .

게시판/블로그 관리

게시판 구분(공지/소개/문의)

공지형 게시판

소개형 게시판

문의형 게시판

상품문의

상담 신청

1대1 문의

공개 질문

erp ☐ ☐ ☐ 1:1 ☐ ☐ ☐.

일자범위

☒ 일 ☐ 주 ☐ 월 ☐ 분기 ☐ 반기 ☐ 년 ☐ 전체

날짜

2024-07-09

~

2024-07-09

검색 조건

=검색 조건=

상태별 검색

전체

<input type="checkbox"/>	번호	제목	생성일	요청파일	답변자	완료일	상태
<input type="checkbox"/>	1	문의 테스트	24-07-03 10:21:40	83ffdc2117634626548	kimhi65@Dc7d		미처리

☐ ☐ ☐ ☐ ☐.

☐

erp ☐ ☐ ☐

erp ☐ ☐ ☐

□□□□ □□□ □□□□□□

□□□ □□□ □□□□□□□□□□

□□□□ □□□ □□□□□□

□□□ □□□ □□□□□□□□□□



□□□□ □□□ □□□□□□

□□□ □□□ □□□□□□□□

0000 000 0000000

000000 000000000000 0000

게시판/블로그 관리

게시판 구분(공지/소개/문의)

공지형 게시판

소개형 게시판

문의형 게시판

블로그 관리

0000 00000 00 0000

erp 000 00 0000 000 00 000 00 0000 0000 00000000.

anu Search And Move

게시판 구분 입력/수정/삭제

저장

기본 설정

게시판 디자인

게시판 코드(중복불가)\*  
contactus-custom

종속코드

종류설명  
상담 신청

게시판명  
상담 신청

게시판구분슬러그  
contactus-custom

리스트 순서  
55

상태  
노출

게시판설정코드  
텍스트

종류코드  
문의형게시판

☐ 미사용

000 00 00 00





erp에 접속할 수 있습니다. 접속할 때 다음 정보를 입력할 수 있습니다.

다음 정보를 입력하여 contactus-custom-input.json 파일을 생성할 수 있습니다.

다음 정보를 입력합니다.

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

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

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

  "HeadSelectOptions": [
    {
      "Value": "xls-report",
      "Caption": "(엑셀) 연락처 목록"
    },
    {
      "Value": "custom-xls-report",
      "Caption": "커스텀 엑셀",
      "Component": "dummy",
      "Parameter": "/list/list-type1/download/contactus-custom",
      "ThemeDir": "/kgolf/erp"
    },
    {
      "Value": "new",
      "Caption": "신규 연락처"
    }
  ]
}
```

```
    },
    {
      "Value": "multi-delete",
      "Caption": "删除多条"
    }
  ],
```

```
"//FormVars": "FormVars[0]-caption, FormVars[1]-searchPopup",
```

```
"FormVars": [
  {
    "ListButton": "列表",
    "DateRange": "",
    "DateNavi": "日期 | (日期范围) | 日期",
    "Date": "日期",
    "From": "从",
    "To": "到",
    "FirstRange": "",
    "SecondRange": "",
    "ThirdRange": "",
    "FourthRange": "",
    "AddTotalLine": "",
    "SelectPopup": "选择(日期范围)",
    "MultiPopup": "",
    "ChartPopup": "",
    "DownloadList": "",
    "ShowOnlyClosed": "",
    "Balance": "",
    "OrderBy": "排序",
    "FilterOption": "过滤",
    "SimpleOption": "简单"
  }
],
```

```
"FilterSelectOptions": [
```

```
  {
    "Value": "",
    "Caption": "等于"
  },
```

```
{
  "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": "全部"  
    }  
],
```

```
"SelectPopupOptions": [  
    {  
        "Caption": "详细情况",  
        "Component": "kgolf::popup.popup-form1.form-a.post.contactus-custom-form",  
        "Parameter": "kgolf::/popup/popup-form1/form-a/post/contactus-custom"  
    }  
],
```

```
"// OrderByOptions": "OrderBy Options//全部 Value 默认",  
"OrderByOptions": [  
    {  
        "Value": "mx.created_on desc",  
        "Caption": "全部 降序"  
    },  
    {  
        "Value": "mx.created_on asc",  
        "Caption": "全部 升序"  
    }  
],
```

```
"//BalanceOptions": "Show OnlyBalaceRemained or All",  
"BalanceOptions": [  
    {  
        "Value": "",  
        "Caption": "全部 显示"  
    },  
    {  
        "Value": "c10 > 0",  
        "Caption": "全部 隐藏"  
    }  
]
```

```
],
```

```
"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"
}
]
}
```

pro 1000 1000 1000 1000 1000 1000.

pro 1000 1000 1000 1000 1000 1000.

1000

pro 1000 1000

□□□ □□ □□

“

□□□ □□□□

- □□ Git Server [□□□□□□□□](#)

“

□□□□ □□□(main\_api) □□□□

1. □□□ □□□□□.
2. home □□□□ □□□ □□□ □□□□ □□□□.

□□□□ main\_api□ □□□□□.


```
cd /home  
mkdir main_api
```

3. □□□□ main\_api□ □□□□ □□□□.

```
cd main_api
```

4. git server에 mybin을 mybin으로 올린다. (

5. HTTP url을 얻는다.

 dabory / mybin

Watch 2Star 0포크 0

파일이슈 0풀 리퀘스트 0위키

설명 없음

437 커밋1 브랜치0 릴리즈

브랜치: mastermybin

HTTPSSHhttp://git.daboryhost.com:10880/dabory/mybin.git

6. git clone을 mybin으로 실행한다.

```
git clone http://git.daboryhost.com:10880/dabory/mybin.git
```

7. mybin에 init-weberp-go-install-in-main\_api를 추가한다.

8. init-weberp-go-install-in-main\_api 파일을 main\_api 디렉토리로 복사한다.

```
./init-weberp-go-install-in-main_api
```

이제 main\_api 디렉토리에 init-weberp-go-install-in-main\_api 파일이 생성된다.

\* init-weberp-go-install-in-main\_api 파일을 main\_api 디렉토리로 복사한다.

```
cd mybin
chmod +x init-weberp-go-install-in-main_api
```

이제 main\_api 디렉토리에 init-weberp-go-install-in-main\_api 파일이 생성된다.

```
cd ..
./init-weberp-go-install-in-main_api
```

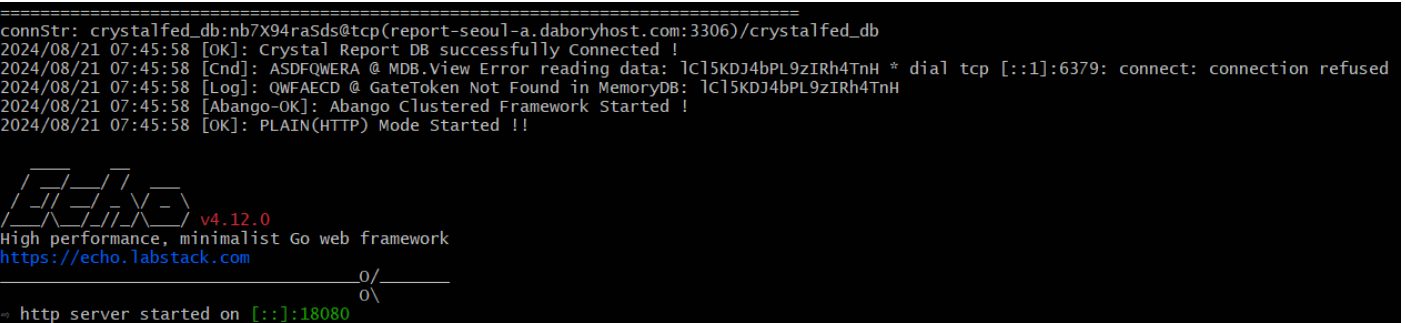
9. init-weberp-go-install-in-main\_api 파일을 main\_api 디렉토리로 복사한다.

- (1). cache-key-pair
- (2). weberp-go
- (3). weberp-queries

10. [ ] [ ]. [ ] [ ].

[ ] [ ] [ ] [ ] [ ] [ ].

```
./mybin/without-update-weberp-go-run
```



11. [ ] [ ] [ ] [ ].

[ ] [ ]

- cache-key-pair
- wheberp-go
- weberp-queries
- mybin

[ ] [ ]

- cache-key-pair[ ]?

# cache-key-pair?

“

cache-key-pair

이제 이 앱은 app에 있는 key pair를 사용합니다.

이 key-pair는 앱에 있는 key pair를 사용합니다.

cache-key-pair는 key-pair를 사용합니다. 이 앱은 key-pair를 사용합니다.



- [이 rest api](#)를