

Protocol22-API

- [\[\]](#)
- [Protocol22-API \[\]](#)
- [Protocol22-API \[\] \[\] \[\]](#)
- [\[JavaScript\] Protocol22-API \[\] \[\] \[\]](#)
- [\[WebApp\] Protocol22-API \[\] \[\] \[\] \(PHP\)](#)
- [\[WebApp\] Protocol22-API \[\] \[\] \[\] \(GoLang\)](#)
- [OwnerKey \[\] \[\] \[\] \[\] \[\]](#)



“

PROTOCOL22-API

PROTOCOL22 API SSO OwnerKey MAIN APP GUEST APP API API

OwnerKey?

PROTOCOL22-API OwnerKey

OwnerKey APP main APP API DB KEY

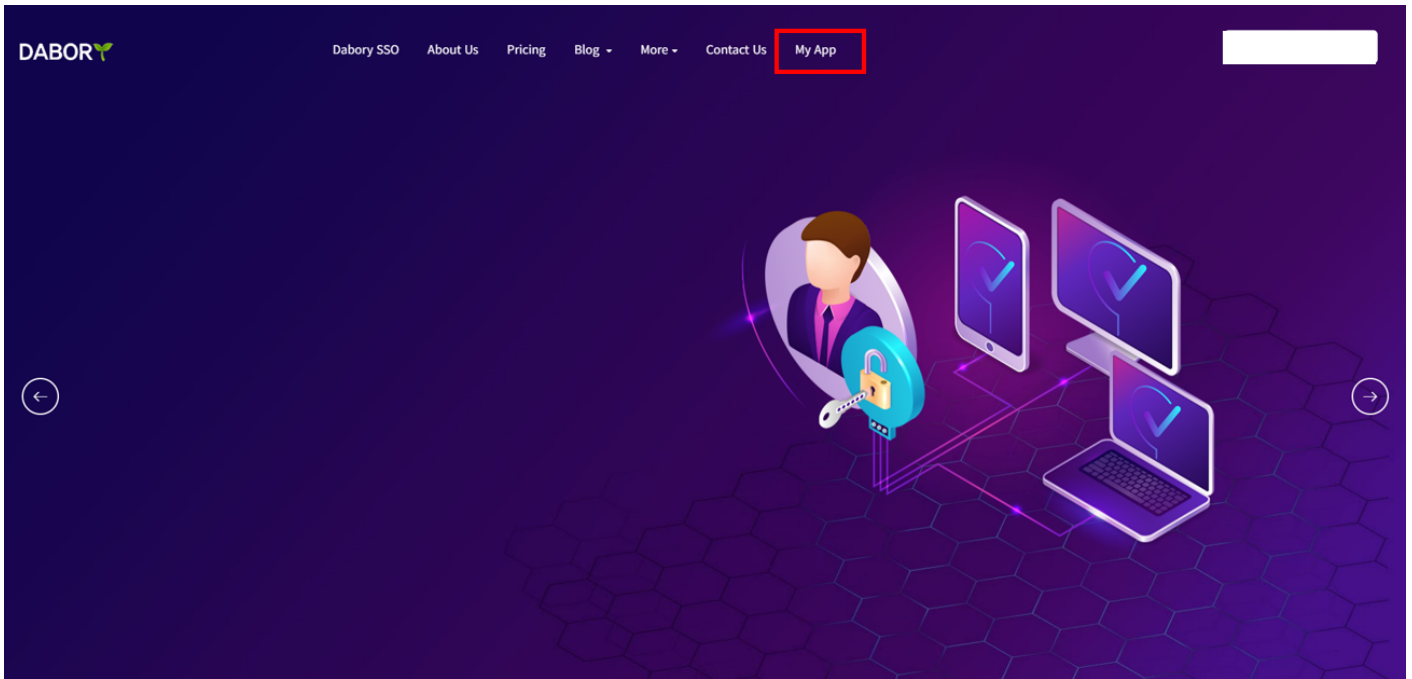
OwnerKey guest APP PROTOCOL22-API

“

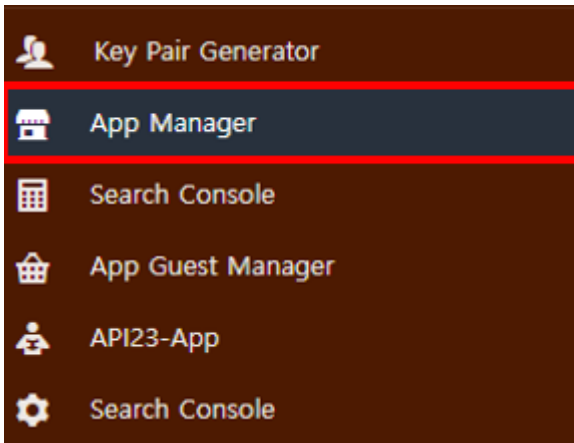
OwnerKey

1. main app dabory

2. guest app My App



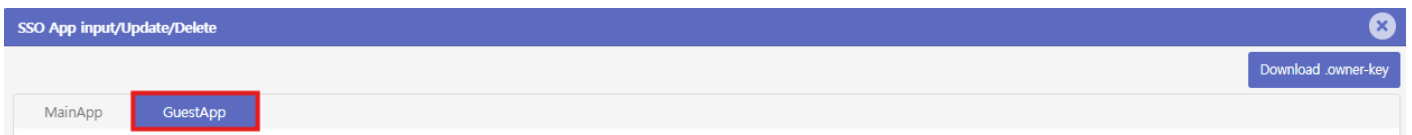
3. App Manager



4. app Owner key app

No	App Name*	App Type	Client Id	Redirect URI	Created	Updated
1	api_test			https://www.daborysso.com	24-04-11 16:42:18	70-01-01 09:00:00

5. GuestApp



6. a c

Generating GuestApp Ownery Key

Guest App Name* (ex:credit-strong-api-app)

OwnerCode*

AppType* (web, js, native-app)

□ Guest App Name : Owner Key□ □□

□ OwnerCode : Owner Key□ □□□□

- owner : □□ api □□
- keypair :
- dbupdate :
- limited : □□ api□ □□

□ AppType : guest app□ type

7. Download .owner-key □□□ □□□□ Owner key□ download □□□.

8. MainApp □□ □□ save □□□□□ □□□□□.

9. □□□ .owner-key□ □□□□ frontend □□□□□ open□□□.

10. .env □□□ MAIN_API_OWNER_KEY □□□ □□□□ □□□ owner key□ □□ □□□□□.

“

Protocol22-API □□

1. Javascript (js □□)

2. GO (webapp □□)

3. Laravel (webapp □□)

js app

	Javascript	WebApp
	ownerApp()	ownerApp(Laravel, Go)
	Guest APP API	api (Backend to Backend)
	\$.fn.ownerKeyLinker.ownerApp()	callOwnerApi()
OwnerKey	ownerKey	session
	OnwerKey API	GateToken API
	ownerKey (ownerKey)	ownerKey(ownerKey)
	<script> js ownerKey	Laravel, Go API

Protocol22-API

Protocol22-API의 API 목록입니다. URL, HTTP Method, Request Body, Response Body

URL	Method	Request Body	Response Body
/printing-curr-price	post	json	
/printing-curr-price	post	json	
/projects	get	json	
/projects	post	json	
/projects	post	json	
/projects/{id}	get	json	
/projects	delete	json	
/after-base64	post	json	
/send-images	post	json	
/text-send	post	json	
/gate-token-get	post	json	
/call-owner-api	post	json	
/owner-token-get	post	json	
/auth/send-emailConfirm	post	json	
/auth/send-smsConfirm	post	json	
/auth/signup	post	json	

/item-url-scrap	post	json	
/gate-token-get-test	post	json	
/call-api	post	json	
/send/text	post	json	
/send-mail	post	json	
/test-send-mail	post	json	
/elasticsearch	post	json	
/elasticsearch-dsl	post	json	



Protocol22-API

Javascript

1. ownerApp

JavaScript API  ownerApp()  ajax  API .

```
$.fn.ownerKeyLinker.ownerApp = function (url, request, callback, async = true) {
  getParameter();

  $.ajax({
    // ${$.fn.ownerKeyLinker.serverUrl} : script  url  serverUrl
    url: `${$.fn.ownerKeyLinker.serverUrl}/dabory-app/${url}`, //ex. https://www.daborysso.com/dabory-
    app/auth/signup
    data: JSON.stringify(request),
    beforeSend: function (xhr) {
      xhr.setRequestHeader('OwnerKey', $.fn.ownerKeyLinker.ownerKey);
    },
    method: 'POST',
    dataType: 'json',
    async: async,
  })
  .done(function (json) {
    callback(json);
  })
  .fail(function (json, textStatus, errorThrown) {
    callback(json);
    iziToast.error({ title: 'Error', message: 'Dabory Owner App Error' });
  });
};
```

url : main app  /dabory-app/api url (ex. www.daborysso.com/dabory-app/auth/signup)

request :    (JSON 

callback :   

async : boolean (true -> async)

beforeSend: function(xhr) {
 xhr.setRequestHeader('OwnerKey', header);
}

2. ownerApi

JavaScript API ownerApi() function. ajax API call.

```
$.fn.ownerKeyLinker.ownerApi = function (url, request, callback, async = true) {  
  getParameter(); // $.fn.ownerKeyLinker.serverUrl $.fn.ownerKeyLinker.ownerKey  
  
  $.ajax({  
    url: `${$.fn.ownerKeyLinker.serverUrl}/dabory-app/call-owner-api`,  
    data: JSON.stringify(request),  
    beforeSend: function (xhr) {  
      xhr.setRequestHeader('OwnerKey', $.fn.ownerKeyLinker.ownerKey);  
      xhr.setRequestHeader('Url', url);  
    },  
    method: 'POST',  
    dataType: 'json',  
    async: async,  
  })  
  .done(function (json) {  
    callback(json);  
  })  
  .fail(function (json, textStatus, errorThrown) {  
    callback(json);  
    iziToast.error({ title: 'Error', message: 'Dabory Owner API Error' });  
  });  
};
```

url : main app /dabory-app/call-owner-api (ex. www.daborysso.com/dabory-app/call-owner-api)

request : object (JSON)

callback : function

async : boolean (true -> async)

beforeSend: function(xhr) {
 xhr.setRequestHeader('OwnerKey', url);
}

OwnerApi vs OwnerApp

이 글에서는 OwnerApi와 OwnerApp의 차이점을 알아보겠습니다.

이 글에서는 OwnerApi와 OwnerApp의 차이점을 알아보겠습니다.

OwnerApp :

이 글에서는 OwnerApp의 특징을 알아보겠습니다. OwnerApp은 API를 호출하여 데이터를 가져오는 방식입니다.

OwnerApi :

이 글에서는 OwnerApi의 특징을 알아보겠습니다. OwnerApi는 API를 호출하여 데이터를 가져오는 방식입니다.

[JavaScript] Protocol22-API

■■■■ ■■

■■■■ Dabory Protocol22-API for JavaScript **Main App** **Guest App** **JavaScript** **API** ■■ ■■

■■■■ ■■ ■■■■■ ■■ ■■■■■ ■■■■.

script ■■■■

app ■■■■

JavaScript ■■■■ API ■■ ■■

1. ■■■■ ■■ JavaScript **OwnerKey** ■■ **MAIN APP** **URL** ■■■■ ■■

2. JavaScript ■■ ownerApp() ■■■■ ■■■■ API ■■■■ ■■■■.

3. API ■■■■ ■■ ■■■■■ ■■ ■■■■ ■■■■■.

“

JavaScript ■■■■ (■■■■ ■■)

EYERECORD 통합 서비스 회원가입



성명을 입력해주세요.

이메일을 입력해 주세요.

인증번호 발송

이메일 인증번호를 입력해 주세요.

인증확인

'-'없이 입력해 주세요.

인증번호 발송

휴대전화 인증번호를 입력해 주세요.

인증확인

비밀번호를 입력하세요.

비밀번호 한번 더 입력하세요.

비밀번호는 특수문자 포함 8자 이상 가능합니다.

약관에 동의합니다. [약관보기](#)

확인



EYERECORD 통합 싱글사인온 계정으로 등록되며 향후 EYERECORD 에서 제공하는 모든 서비스에서 동일한 아이디로 서비스가 가능합니다.

script

javascript script src url ServerUrl OwnerKey .

```
<script src="/js/dabory.owner-key-linker.js?serverUrl=https://www.daborysso.com&ownerKey=zEUhgZEpPbFpz/clghtYCsoFINi8JX3Lh45ank1s+w2P
```


callback : `function (response) { ... }`

async : `boolean` (default true -> `boolean`)

beforeSend: `function (xhr) { xhr.setRequestHeader('OwnerKey', ownerKey); }`

API `ownerApi()`

JavaScript API `ownerApi()` usage example. `ajax()` API usage example.

```
$.fn.ownerKeyLinker.ownerApi = function (url, request, callback, async = true) {
  getParameter(); // $.fn.ownerKeyLinker.serverUrl + $.fn.ownerKeyLinker.ownerKey + '...'

  $.ajax({
    url: `${$.fn.ownerKeyLinker.serverUrl}/dabory-app/call-owner-api`,
    data: JSON.stringify(request),
    beforeSend: function (xhr) {
      xhr.setRequestHeader('OwnerKey', $.fn.ownerKeyLinker.ownerKey);
      xhr.setRequestHeader('Url', url);
    },
    method: 'POST',
    dataType: 'json',
    async: async,
  })
  .done(function (json) {
    callback(json);
  })
  .fail(function (json, textStatus, errorThrown) {
    callback(json);
    iziToast.error({ title: 'Error', message: 'Dabory Owner API Error' });
  });
};
```

url : API endpoint (ex. `www.daborysso.com/dabory-app/call-owner-api`)

request : `object` (JSON)

callback : `function (response) { ... }`

async : `boolean` (default true -> `boolean`)

beforeSend: `headers.setRequestHeader('OwnerKey', header);`

OwnerApi `OwnerApp` `api url` `header`.

OwnerApi vs OwnerApp

API `OwnerApp` `api url` `header`.

API `OwnerApp` `api url` `header`.

	OwnerApp()	OwnerApi()
API	<code>OwnerApp</code> <code>api url</code> <code>header</code> (API <code>OwnerApp</code> <code>api url</code> <code>header</code>)	<code>OwnerApi</code> <code>api url</code> <code>header</code> <code>api url</code> <code>header</code>
API	guest (frontend)-> main (backend) -> guest(frontend)	guest (frontend)-> main (backend) -> api (backend) -> guest(frontend)

ownerApi `OwnerApp`

- API `OwnerApp` `api url` `header` `body` `request`.

```
$.fn.ownerKeyLinker.ownerApi(  
  'remote-signup-setup-get',  
  {  
    'SetupCode': 'passwd-policy',  
    'BrandCode': 'member'  
  },  
  function (response) {  
    window.passwdPolicy = response;  
    $('#policy-desc-em').text(window.passwdPolicy['PolicyDesc']);  
    console.log(window.passwdPolicy);  
    $('#join').modal('show');  
  }  
);
```

`api url` : `api url`

`body` : `api url` `body` `request`

[[[]]] : response [[[]]] callback [[]]

ownerApp [[]]

- API [[[]]] [[[]]] (auth/send-smsConfirm) [[[]]] [[[]]].

```
$.fn.ownerKeyLinker.ownerApp('auth/send-smsConfirm', {  
  "MobileNo": $('#mobile-no-txt').val(),  
}, function (response) {  
  if (response['error']) {  
    $($this).prop('disabled', false);  
    SmsNumber = null;  
    iziToast.error({ title: 'Error', message: response['message'] });  
  } else {  
    SmsNumber = response['data'];  
    iziToast.success({ title: 'Success', message: '[[ [ ] ] ]' });  
  }  
});
```

- API [[[]]] [[[]]] (auth/send-emailConfirm) [[[]]] [[[]]].

```
$.fn.ownerKeyLinker.ownerApp('auth/send-emailConfirm', {  
  "Email": $('#email-txt').val(),  
}, function (response) {  
  if (response['error']) {  
    $($this).prop('disabled', false);  
    EmailNumber = null;  
    iziToast.error({ title: 'Error', message: '[[ [ ] ] ]' });  
  } else {  
    EmailNumber = response['data'];  
    iziToast.success({ title: 'Success', message: '[[ [ ] ] ]' });  
  }  
});
```

API [[[]]] [[[]]] [[[]]].

[WebApp] Protocol22-API (PHP)

“

[WebApp] PROTOCOL22-API

0000 00

Labory OwnerKey API for WebApp Main App Guest App PHP Laravel WebApp PROTOCOL22-API

WebApp Owner Key API

1. .env
2. Laravel callOwnerApi() Owner APP API
3. GateToken frontUrl API
4. API

WebApp

- env

.env OwnerKey APP

MAIN APP 2

_

```
# [protocol22]
OWNER_APP_NAME=YOUR_OWNER_APP_NAME # Owner APP ( )
GUEST_APP_OWNER_KEY=YOUR_OWNER_KEY # Guest APP OwnerKey
GUEST_APP_OWNER_URL=YOUR_OWNER_URL # Owner APP URL (ex.www.daborysso.com)

OWNER_APP_NAME2=YOUR_OWNER_APP_NAME_2
GUEST_APP_OWNER_KEY2=YOUR_OWNER_KEY_2
GUEST_APP_OWNER_URL2=YOUR_OWNER_URL_2

OWNER_APP_NAME3=YOUR_OWNER_APP_NAME_3
GUEST_APP_OWNER_KEY3=YOUR_OWNER_KEY_3
GUEST_APP_OWNER_URL3=YOUR_OWNER_URL_3
```

OwnerKey?

- OwnerKey **GUEST APP MAIN APP API**
- WebApp OwnerKey **API**

- Laravel Owner API

callOwnerApi() MAIN APP API

```
public function callOwnerApi($request)
{
    // 'type' (guest[1] → 1)
    preg_match('/\d+/', $request['type'], $matches);
    $index = isset($matches[0]) ? (int) $matches[0] : null;

    // API URL GateToken
    if ($index !== null) {
        $apiUrl = session("OwnerGateToken.guest"][$index]['frontUrl'] ?? null;
        $gateToken = session("OwnerGateToken.guest"][$index]['gateToken'] ?? null;
    }

    // API URL
    $request['url'] = $apiUrl . '/' . $request['url'];
    $request['headers'] = ['GateToken' => $gateToken];
    return $this->callApi($request);
}
```

- API 调用

API 调用 remote-signup-setup-get api 调用 示例。

```
$officeInfo = $this->callApiService->callOwnerApi([
    'url' => 'remote-signup-setup-get', // api 调用 url
    'data' => [ // 调用 data
        'SetupCode' => 'passwd-policy',
        'BrandCode' => 'member',
    ],
    'type' => 'guest[1]' // env 调用 owner app 调用 session 调用 index 调用
]);
```

调用 Dabory Protocol22-API for WebApp 调用 API 调用。

[WebApp] Protocol22-API (GoLang)

“

[WebApp] Protocol22-API

API

API Dabory OwnerKey API for WebApp main app guest app

WebApp

2. WebApp

GateToken

```
package main
func main() {
  // (Goroutine)
  go func() {
    for {
      // 1. UpdateGateTokenIfNeeded() (GateToken)
      err := controllers_func.UpdateGateTokenIfNeeded()
```

```

}

// 2. 更新 GateToken 如果过期
if err != nil {
    e.ErrLog("UpdateGateTokenIfNeeded Error", err)
}

// 3. 2秒后再次检查 (2 * 1秒)
time.Sleep(2 * time.Second)
}
}()

// 主函数: main 函数 启动 服务
select {} // main 函数 启动 服务
}

```

添加 "OwnerKey" 字段 GateToken, BackUrl

Host: dabory-app/gate-token-get -> Host -> main_api (GateToken)

```

// GateToken 更新
func UpdateGateTokenIfNeeded() error {
    myOwnerKey := "My_OwnerKey" //OwnerKey (GateToken 更新)

    // GateToken 更新 (OwnerKey 更新)
    err := locals.GuestGateTokenGets(myOwnerKey)
    if err != nil {
        fmt.Println("GateToken 更新:", err)
        return err
    }

    return nil
}

type AppApi struct {
    ApiUrl string
    GateToken string
}

```

```
var GAppApis [2]AppApi // GateToken, ApiUrl [] []
```

```
func GuestGateTokenGets(myOnwerKey string) error {
```

```
    []appType := 0 //Dbupdate
```

```
    []// 0:keypair, 1:Dbu // "SsoConnString" Must be Deprecated
```

```
    []var err error
```

```
    []if myOnwerKey != "" {
```

```
        []// e.LogStr("Iskdjqfals", e.LogFuncName()+"; GuestGateTokenOwner with OwnerKey: "+myOnwerKey)
```

```
        []_, _, err = GuestGateTokenOwner(appType, "https://host_domain_URL.com", myOnwerKey) // app type ->
```

```
        OwnerCode
```

```
            // Host Domain URL, OwnerKey[] [] [] [] []
```

```
        []if err != nil {
```

```
            []return e.ErrLog(e.FuncRun("23rfsr3qrse", e.CurrFuncName()), err)
```

```
        []}
```

```
    []}
```

```
    []return nil
```

```
    }
```

```
func GuestGateTokenOwner(appType int, pivotUrl string, ownerKey string) (string, string, error) { // HOST
```

```
URL/dabory-app -> pivot URL
```

```
    []var appTypeCode string
```

```
    []if appType == 0 {
```

```
        []appTypeCode = "keypair"
```

```
    []} else if appType == 1 {
```

```
        []appTypeCode = "dbupdate"
```

```
    []}
```

```
    []if GAppApis[appType].GateToken != "" { // [] [] [] [] ([] GateToken[] [] [])
```

```
        []// fmt.Println("34092ujfa : "+"Using GateToken in ARRAY to access : ", GAppApis[0].GateToken)
```

```
    []} else { // (GateToken[] [] [] [] [])
```

```
        []fmt.Println("34092ujfa : " + "isn't exist GateToken in ARRAY to access ")
```

```
        []vGt := &GateTokenGetReq{
```

```
            []OwnerKey: ownerKey,
```

```
        []}
```

```
        []bodyBytes, _ := json.Marshal(vGt)
```

```

frontUrl := pivotUrl + "/dabory-app/gate-token-get"
msgBytes, stalnt, err := HttpResponseSimplePost("POST", frontUrl, bodyBytes)
    // Strong -> Host Lalavel Http -> Lalavel -> main_api GateToken
e.LogStr("23rfsr3qr-GateTokenGetReq", e.LogFuncName()+"; Raravel frontUrl : "+frontUrl)
if err != nil {
return "", "", e.ErrLog(e.FuncRun("45425fd34sd-The HTTP request "+frontUrl, e.CurrFuncName()), err)
}

if stalnt != 200 {
TrackFailure() //
fmt.Println("GateToken ", err, "(1 ", len(failTimestamps), ")", " Code : ", stalnt)

// 1 maxFailures(3)
if len(failTimestamps) >= maxFailures {
fmt.Println("1 GateToken ", maxFailures, " -> ")
os.Exit(1) //
}

return "", "", errors.New(e.FuncRun("87ty344ra3-Request Fail "+string(msgBytes), e.CurrFuncName()))
}

failTimestamps = []int64{} //

ret := &struct {
ApiUrl string
GateToken string
}{}

if err := json.Unmarshal(msgBytes, ret); err != nil { // GateToken BackendUrl ret
return "", "", e.ErrLog(e.FuncRun("45425fd34sd-Json Format "+frontUrl, e.CurrFuncName()), err)
}

// GateToken
GAppApis[appType].ApiUrl = ret.ApiUrl
GAppApis[appType].GateToken = ret.GateToken
e.OkLog("Just Added GateToken in ARRAY to access AppType: " + appTypeCode)
e.OkLog("Just Added GateToken in ARRAY to access AppType: " + ret.GateToken)
e.OkLog("Just Added GateToken in ARRAY to access AppType: " + ret.ApiUrl) // main_api URL
}

```

```
return GAppApis[appType].ApiUrl, GAppApis[appType].GateToken, nil
}
```

GateToken main_api

```
// API URL
func process() {
    // 1. GateToken API URL
    apiUrl := locals.GAppApis[0].ApiUrl // API URL
    gateToken := locals.GAppApis[0].GateToken // API GateToken

    // 2. API URL (credit-page)
    mainApiURL := apiUrl + "/credit-page"

    // 3. Request Body (credit-page)
    requestBody := map[string]interface{}{
        "PageVars": map[string]interface{}{
            "Fields": "credit_no",
            "Limit": 1,
            "Desc": "id",
            "MyFilter": "",
            "QueryCnt": 0,
            "Query": "",
            "Asc": "",
            "Offset": 0,
            "ReturnJson": "",
        },
    }

    // 4. API Request (GateToken)
    body, err := RequestWithGateToken(mainApiURL, gateToken, requestBody)
    if err != nil {
        // Error handling
        e.ErrLog("API Error: ", err)
    }
    return
}
```

```
// 5. API 호출
```

```
fmt.Println("API 호출:", string(body))  
}
```

Gate Token을 받아서 (Gate Token을 받아서)

```
// Gate Token을 받아서 API 호출  
func RequestWithGateToken(mainApiURL string, gateToken string, requestBody map[string]interface{}) ([]byte, error) {  
    // 1. HTTP 호출 (Gate Token을 받아서)  
    resp, body, err := SendHttpRequest(mainApiURL, gateToken, requestBody)  
    if err != nil {  
        fmt.Println("HTTP 호출:", err)  
        return nil, err  
    }  
    defer resp.Body.Close()  
  
    fmt.Println("출력:", string(body)) // API 호출  
  
    // 2. 500 응답 -> Gate Token을 받아서 (Gate Token을 받아서)  
    if resp.StatusCode == 500 {  
        fmt.Println("Gate Token을 받아서! Gate Token을 받아서.")  
  
        // Gate Token을 받아서  
        locals.GAppApis[0] = locals.AppApi{ }  
  
        // Gate Token을 받아서 (5초 대기)  
        for locals.GAppApis[0].GateToken == "" {  
            fmt.Println("Gate Token을 받아서.")  
            time.Sleep(5 * time.Second)  
        }  
  
        fmt.Println("Gate Token을 받아서!")  
  
        // Gate Token을 받아서 API 호출  
        newGateToken := locals.GAppApis[0].GateToken  
        resp, body, err := SendHttpRequest(mainApiURL, newGateToken, requestBody)
```

```

    if err != nil {
        fmt.Println("HTTP ", err)
        return nil, err
    }
    defer resp.Body.Close()

    // 获取 token
    if resp.StatusCode == 200 {
        e.OkLog("GateToken HTTP ")
        return body, nil
    } else {
        return nil, fmt.Errorf("GateToken , : %d", resp.StatusCode)
    }
}

// GateToken
e.OkLog("GateToken HTTP ")
return body, nil
}

// HTTP
func SendHttpRequest(mainApiURL string, gateToken string, requestBody map[string]interface{})
(*http.Response, []byte, error) {
    // 1. JSON
    jsonData, err := json.Marshal(requestBody)
    if err != nil {
        fmt.Println("JSON ", err)
        return nil, nil, err
    }

    // 2. HTTP (POST )
    req, err := http.NewRequest("POST", mainApiURL, bytes.NewBuffer(jsonData))
    if err != nil {
        fmt.Println("HTTP ", err)
        return nil, nil, err
    }

    // 3. HTTP (GateToken )
    req.Header.Set("Content-Type", "application/json")
    req.Header.Set("GateToken", gateToken)
}

```

```
// 4. HTTP 请求
```

```
client := &http.Client{}
```

```
resp, err := client.Do(req)
```

```
if err != nil {
```

```
    fmt.Println("HTTP 请求:", err)
```

```
    return nil, nil, err
```

```
}
```

```
// 5. 读取响应
```

```
body, err := io.ReadAll(resp.Body)
```

```
if err != nil {
```

```
    fmt.Println("读取响应:", err)
```

```
    return nil, nil, err
```

```
}
```

```
// 6. 打印结果
```

```
fmt.Println("状态码:", resp.StatusCode)
```

```
fmt.Println("响应体:", string(body))
```

```
return resp, body, nil
```

```
}
```







OwnerKey







OwnerKey Guest App Owner App 

OwnerKey SSO




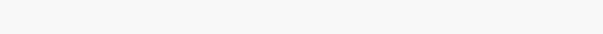
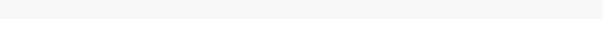






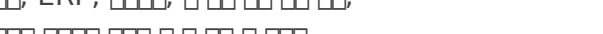






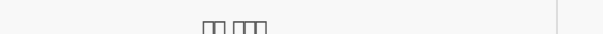
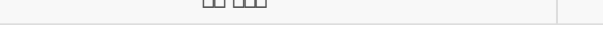




1. OwnerKey = ()

-  
-  "  " 
→ 












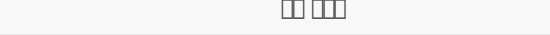






2. SSO (Single Sign-On) =

-  
-  



“Dabory                             



- 
-  Dabory   
-   ERP,           





Dabory Composable

SEO, Content	SEO, Content
SEO, Content	SEO, Content
SEO, Content	SEO, Content
SEO, Content	SEO, Content

DC(Dabory Composable) OwnerKey SEO SEO SEO SEO SEO SEO SEO SEO SEO:

1. SEO Strategy

1.1 Content Strategy

- OwnerKey SEO Content Strategy
- SEO Content Strategy
- A B Content Strategy, SEO Content Strategy → SEO Content Strategy → SEO Content Strategy

1.2 Authority Sharing

- Content Strategy
- DC Content Strategy
- Content Strategy

1.3 Content Strategy

- Content Strategy OwnerKey
- Content Strategy (Link diversity) → Content Strategy

2. SEO Strategy

2.1 Canonical Content Strategy

- Content Strategy canonical Content Strategy
- Content Strategy
- Content Strategy → SEO Content Strategy

2.2 Topic Cluster Content Strategy

- Content Strategy Content Strategy (E, E, A, T)
- Content Strategy E-E-A-T (Experience, Expertise, Authoritativeness, Trustworthiness)
- Content Strategy

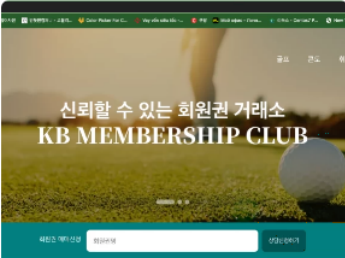
2.3 Content Strategy

USE CASES

Home > Use Cases

12

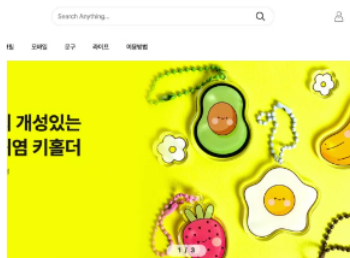
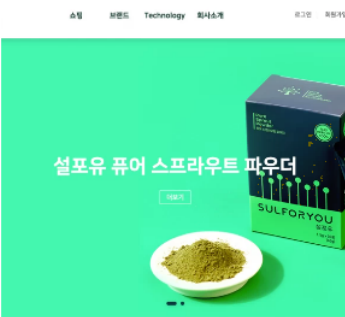
블로그 검색



KIMHI65@OC7D 2024.12.14 23:20:52

골프/콘도 회원권거래소-KB Golf

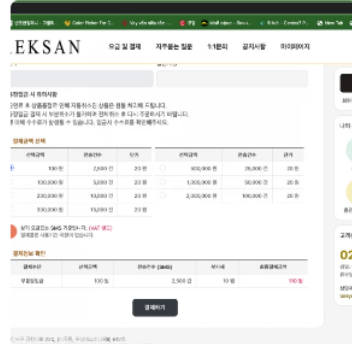
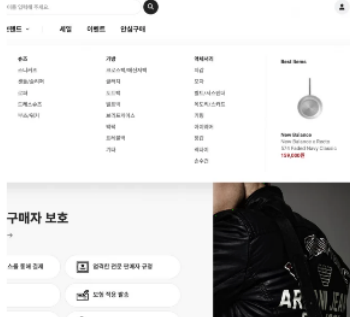
K-B 멤버십 쇼핑을 개발 계획 1. 프로젝트 개요 목적: 골프, 콘도, 피트니스 회원권과 같은 프리미엄 멤버십 서비스의...



KIMHI65@OC7D 2024.12.14 23:17:31

인쇄물 다품종소량 주문 쇼핑물-블랭커

블랭커 온라인 스토어 IT 솔루션: 맞춤형 인쇄 제품을 위한 혁신적인 플랫폼 우리는 맞춤형 인쇄 제품을 제공하는 온...



KIMHI65@OC7D 2024.12.14 23:16:00

포인트 충전 방식 문자 전송 관리 웹사이트-백산광학

대용량 문자처리 쇼핑물 개발 계획 1. 프로젝트 개요 목적: 대용량 문자 메시징 기능이 통합된 고성능 쇼핑물 시스템...



composable USE CASES factory

- env

.env OwnerKey. Frier

```
# [Friend Keys]
OWNER_APP_NAME_0=YOUR_OWNER_APP_NAME # Owner APP ( )
GUEST_APP_OWNER_KEY_0=YOUR_OWNER_KEY # Guest APP OwnerKey
GUEST_APP_OWNER_URL_0=YOUR_OWNER_URL # Owner APP URL (ex.https://composable.daboryhost.com)

OWNER_APP_NAME_1=YOUR_OWNER_APP_NAME_2
GUEST_APP_OWNER_KEY_1=YOUR_OWNER_KEY_2
GUEST_APP_OWNER_URL_1=YOUR_OWNER_URL_2
```

OWNER_APP_NAME_2=YOUR_OWNER_APP_NAME_3

GUEST_APP_OWNER_KEY_2=YOUR_OWNER_KEY_3

GUEST_APP_OWNER_URL_2=YOUR_OWNER_URL_3

OwnerKey?

- OwnerKey **GUEST APP** **MAIN APP** **API**
- WebApp OwnerKey **API**

- URL

url

(1). composable Use Cases.

Organizations using Golang as well as Dabory Composable

Organizations in every industry use Go to power their software and services

(2). /bbs/list/use-cases

- **□□(Guest)** □□□□ □□ □□

(1) □□ □□□□ □□□□ □□□□ □□□□.

(2) □□□□ □□(□□□□□□) - □□□□□□(□□) - □□□□□□□□(□□) - □□□□□□ □□ □□□□ □□□□ □□□□□□.



슈퍼 유저 메뉴



관리자 계정



구분설정 관리



잔액/재고/적립금 설정



메뉴 확장과 API 관리



시스템 관리(주의)



바둑판 메뉴 불러오기



사용자 메뉴 불러오기



홈페이지 메뉴 불러오기



회원(마이앱) 메뉴 불러오기



통합 설정 불러오기



거래구분 불러오기



품목구분 불러오기



게시판 구분 불러오기



표준 품목옵션 불러오기



회계계정 불러오기

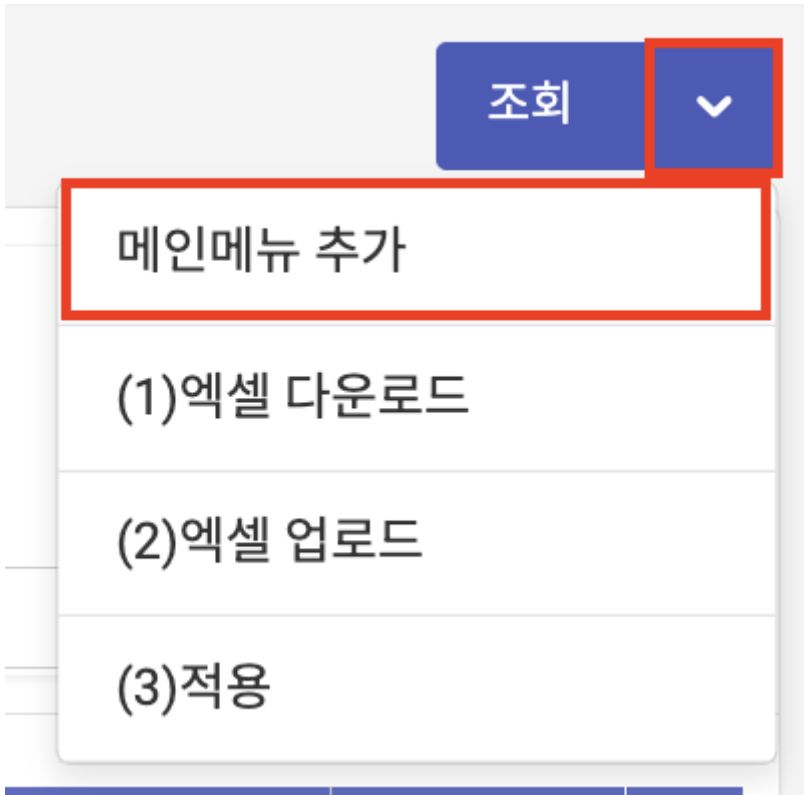


DB Update



Encryption Keys

(2). □□ □□ □□□ □ □□ □□□□.



(3). □□□□ □ □□□ □ □□□□ □□□□ □□□□.

□□□□URI □ □□□ □□□□ □□□□URI□ .env□ OWNER_APP_NAME □ □□□ □□ □□□□ □□□□.



저장

메뉴코드*

400000

메뉴명

유즈 케이스

페이지 URI

/bbs/list/use-cases/composable

탭메뉴해쉬



언어구분

한국어

링크 타입

일반페이지

종류

primary

상태

all

지정아이콘

커스터마이징 변수

- 메뉴표시 누락
- target=_blank 인가
- PC에서누락
- Mobile에서누락
- 태블릿에서누락
- 로그인경우만
- 로그아웃경우만

메뉴코드: 100000~900000

메뉴명: ()

URI: URL env OWNER_APP_NAME

URI: /bbs/list/use-cases env OWNER_APP_NAME composable /bbs/

종류: primary

상태: all

타입: 일반페이지

지정아이콘:

target=_blank:

PC:

Mobile:

태블릿:

