8장. 커스텀 테이블 사용방법

- 강한 커스터마이징(Strong Cutomizing)

강한 커스터마이징(Strong Cutomizing)

다보리 컴포저블의 표준화된 테이블 구조는 통합 관리, 데이터의 일관성, 유지보수 효율성 등

여러 가지 면에서 큰 장점을 제공합니다. 만약 특정 애플리케이션에서 구조 변경이 발생했을

`db-update` 를 실행하면 메인 API 서버에서 돌고 있는 모든 애플리케이션의 테이블 구조를

동기화할 수 있습니다. 하지만 모든 애플리케이션이 동일한 데이터 구조를 사용함에 따라 유

연성의 제한이 생길 수 있습니다.

이 [illegible] 이징(Strong Custo[illegible]을 통해 표준 테이블 구조에서 벗어

나 다양하게 커스터마이징할 수 있습니다. 커스터마이징 테이블에는 테이블명에 특정 prefix를

붙여 기존 표준 테이블과 구분합니다.

예 `dbr_item` 이라 `zzz_dbr_item` 이 같이 테이블명

을 변경할 수 있습니다. 이렇게 테이블을 커스터마이징하여 각 애플리케이션 별로 개성 있는

데이터 구조를 구축할 수 있습니다.

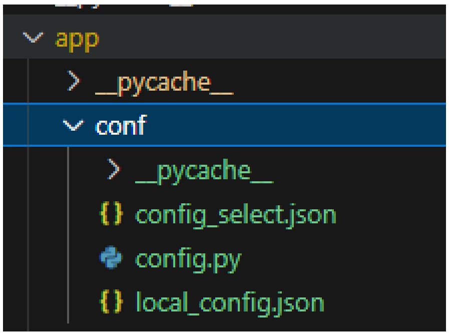
또한, Strong Cutomizing은 Java, Python, Golang으로 API 서버가 구축되어 있습니다. 이

API 서버를 통해 해당 어플리케이션의 개발 언어와 무관하게 다보리 컴포저블의 게이트웨이

서버를 이용할 수 있습니다. 이를 통해 각 애플리케이션은 고유한 요구사항을 반영한 맞춤형

데이터 구조를 가질 수 있으면서도, 다보리 컴포저블의 통합 관리 및 일관성 유지의 장점을 활

용할 수 있습니다.



강한커스터마이징 방법

강한 커스터마이징 api를 이용하기 위해서는 main-api server에서 발급받은 key-pair 값과 각 api 서버의 설정 파일[

필요합니다. 설정에 대한 설명은 다음과 같습니다.

파이썬을 예시로 들면 app/conf/config_select.json 파일과 local_config.json 파일을 확인해야 합니다. app/conf/config_select.json 파일에는 다음과 같이 설정파일을 찾을 수 있도록 지정할 수 있습니다.

```
{
   "ConfSelect": "local",
   "ConfPostFix": "_config.json"
}
```

ConfSelect와 ConfPostFix를 통해 서버에서 설정파일을 찾을 수 있도록 작성합니다.서버에서는

app/conf/config_select.json 파일의 ConfSelect와 ConfPostFix를 통해 설정 파일을 찾고

config_select.json파일을 읽어 각 설정값들을 확인합니다. config_select.json 파일에서는

{"key" : "", "value" : "" } 형식으로 설정값들이 정의되어 있습니다.

{ "Key": "IsYDBFixed", "Value": "No" } 여기서 IsYDBFixed의 value를 yes로 설정해놓

gate-token 없이도 게이트웨이 서버를 이용할 수 있습니다.

□□ □□□□□□□.