Echo에서 HTTP 응답 반환

1. 개요

Dabory는 **Go** 언어에서는 **Echo** 프레임워크로 프론트엔드 단에 응답데이터로 **Json** 을 `c.String()` 과 `c.JSONBlob()` 을 사용하여 HTTP 응답을 Json으로 처리하는 법

---

2. `c.String(statusCode int, response string)` 사용법

기본 설명

- `c.String()` 은 단순 문자열 응답을 반환 하는 함수이다
- `statusCode` 는 HTTP 상태 코드, `response` 는 프론트엔드로 보낼 응답 본문이다.

기본 예제

var c echo.Context
`return c.String(607, "")`

주의 사항

- `607` 이라는 커스텀 **HTTP** 상태 코드로 응답을 보낸다.
- 응답 `""` 은 (빈 문자열)이다.
- 프론트 `response.status` 에서 `607` 인지 확인 가능

---

3. `c.JSONBlob(statusCode int, jsonData []byte)` 사용법

기본 설명

- `c.JSONBlob()` 은 **JSON** 형식의 응답을 반환 하는 함수이다
- `statusCode` 는 HTTP 상태 코드, `jsonData` 는 이미 변환된 JSON `[]byte` 데이터.

□ □□ □□

```
ret, err := json.Marshal(vRet)
if err != nil {
    return c.String(500, "Failed to encode JSON: "+err.Error())
}
return c.JSONBlob(http.StatusOK, ret)
```

□ □□ □□

1. `json.Marshal(vRet)`을 `ZbaksanSorderEyetestRes` 구조체를 JSON 데이터로 변환한다.
2. 변환된 JS`c.JSONBlob(http.StatusOK, ret)`을 통해 프론트엔드로 전송한다.
3. [**`response.json()`** [**`response.data`**로 **JSON**을 받을 수 있다.

---

4. `c.String()` vs `c.JSONBlob()` 비교

기능	**`c.String()`**	**`c.JSONBlob()`**
응답 형식	문자열**(String)**	**JSON (**바이트 배열**)**
사용 목적	단순 메시지, 상태 코드 반환	JSON 데이터 응답
예제 코드	`c.String(607, "")`	`c.JSONBlob(http.StatusOK, ret)`
프론트에서 처리	`response.text()`	`response.json()` [`response.data`

---

5. 요약

[**`c.String(607, "")`**

- HTTP 상`607`을 반환하여 프론트엔드에서 감지 가능.
- [`response.status` 값으로 확인 후 특정 로직 실행.

[**`c.JSONBlob(http.StatusOK, ret)`**

- JSON 데이터를 반환할 때 사용.
- `json.Marshal()`을 통해 JSO`response.json()`으로 처리.

Revision #4
Created 7 February 2025 03:02:27 by 하늘

Updated 7 February 2025 05:55:17 by □□