Echo에서 HTTP 응답 반환

1. 개요

Dabory□**Go** 언어에서는 **Echo** 프레임워크로 프론트엔드 단에 응답데이터로 **Json** [`c.String()` [`c.JSONBlob()` 을 사용하여 HTTP 응답을 Json으로 처리하는 [

---

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

□ □□

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

□ □□ □□

var c echo.Context

`return c.String(607, "")`

□ □□ □□

- `607` 이라는 커스텀 **HTTP**[응답으로 보낸다.
- 응 `""` [(빈 문자열)이다.
- [**`response.status`** [**`607`** 인지 확인 가능

---

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

□ □□

- `c.JSONBlob()` [**JSON** 형식의 응답을 반환 하는 함수이다
- `statusCode` 는 HTTP [`jsonData` 는 이미 변환된 J `[]byte` [데[.

□ □□ □□

```
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)` □ `ZbaksanSorderEyetestRes` □□□□ JSON □□□□ □□□□.
2. □□□ JS`c.JSONBlob(http.StatusOK, ret)` □ □□ □□□□□□□ □□□□.
3. [**`response.json()`** [**`response.data`** □ **JSON**□ □□ □ □□.

---

4. `c.String()` vs `c.JSONBlob()` □□

□□	**`c.String()`**	**`c.JSONBlob()`**
□□ □□	□□□**(String)**	**JSON (□□□ □□)**
□□ □□	□□ □□□, □□ □□ □□	JSON □□□ □□
□□ □□	`c.String(607, "")`	`c.JSONBlob(http.StatusOK, ret)`
□□□□□ □□	`response.text()`	`response.json()` [`response.data`

---

5. □□

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

- HTTP □ `607` □ □□□□ □□□□□□□□ □□ □□.
- [`response.status` □□□ □□ □ □□ □□ □□.

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

- JSON □□□□ □□□ □ □□.
- `json.Marshal()` □ □□ JSO`response.json()` □□ □□.

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

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