

Касаемо:

Первого:

Х	Заголовки	Предварительный просмотр	Ответ	Инициатор	Время
▼		{status: false, data: {sms: null, mms: null, voice_call: null, email: null,...},...}			
▼		{data: {sms: null, mms: null, voice_call: null, email: null,...}}			
▶		{billing: {CreateCustomer: false, Purchase: false, Payout: false, Recurring: false, FraudControl: false,...}}			
		email: null			
		incident: null			
		mms: null			
		sms: null			
		support: null			
		voice_call: null			
		error: "Error on collet data"			
		status: false			

Этап 12. API состояния систем и ответы в случае ошибок

7. Если сбор данных прошле без ошибок заполните ResultT.Data структурой

ResultSetT. Если ошибки были не заполняйте это свойство.

8. С помощью функции json.Marshal запишите в новую переменную содержимое

полученной структуры ResultT в виде json

Проводим проверку: в случае если симулятор не запущен (`go run simulator/main.go`), отсутствует графа `result.Data` и заполняются только `result.Error` и `result.Status`, и при отправке данных типа ResultT при результате проверки `false`, пустая графа `ResultT.Data` так декодируется функцией `json.Marshal`

```
data := getResultData()

check = getCheckServices(data)

if check {

result.Status = true

result.Data = data

result.Error = ""

} else {

result.Status = false

result.Error = "Error on collet data"

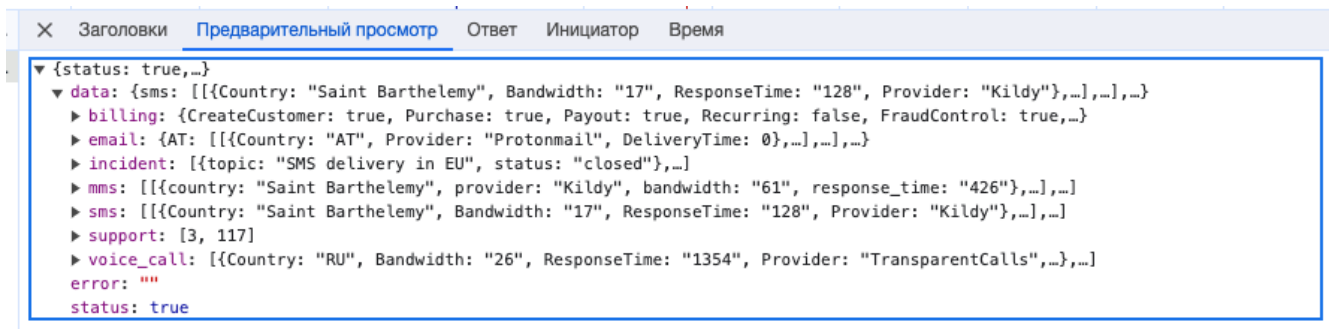
}
```

Второго:

1. Внес в структуру проекта перетягиваемый файл-симулятор `simulator`
2. Исправил пути к файлам на относительные
 - В самом файле `main.go` в папке `simulator` используется только функция `os.WriteFile` без предварительного создания `os.Create` и как результат файлы создаются в корне *проекта*, а не в директории `simulator` (перебирать код симулятора не стал)
3. Так же просматривая тестовую структуру <http://127.0.0.1:8383/test> которую генерирует симулятор, и при ссылке на который ***status_page.html*** выдает данные, заметил что есть расхождения, где `email`, отсутствует key при структуре `map[string] []model.EmailData` (видимо другая структура под `email` в тестовом выводе)
 - Так же перетягивал данный с тестового вывода запросом `Get` с <http://127.0.0.1:8383/test> и полученные байты кодировал под структуру `ResultSetT` (`json.Unmarshal`) и декодировал (`json.Marshal`) в байты обратно и отправил как результат, перенаправил ссылку на результат, и файл ***status_page.html*** уже выдал ошибку *TypeError: Load failed*

Запуск:

1. `go run simulator/main.go`
2. `go run main.go`
3. <http://localhost:8080/>
- 4.



```
{status: true,...}
  data: {sms: [{Country: "Saint Barthelemy", Bandwidth: "17", ResponseTime: "128", Provider: "Kildy"},...],...}
    billing: {CreateCustomer: true, Purchase: true, Payout: true, Recurring: false, FraudControl: true,...}
    email: {AT: [{Country: "AT", Provider: "Protonmail", DeliveryTime: 0},...],...}
    incident: [{topic: "SMS delivery in EU", status: "closed"},...]
    mms: [{country: "Saint Barthelemy", provider: "Kildy", bandwidth: "61", response_time: "426"},...]
    sms: [{Country: "Saint Barthelemy", Bandwidth: "17", ResponseTime: "128", Provider: "Kildy"},...]
    support: [3, 117]
    voice_call: [{Country: "RU", Bandwidth: "26", ResponseTime: "1354", Provider: "TransparentCalls"},...],...}
  error: ""
  status: true
```