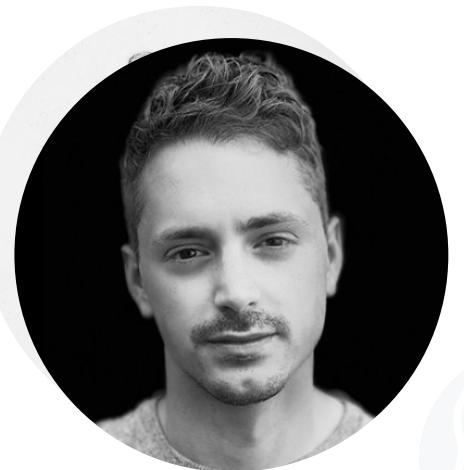


Working with JSON



Josh Duffney

Senior Cloud Advocate at Microsoft | Former Microsoft MVP

@joshduffney | www.duffney.io

Using JSON to Exchange Data



Client

Sends a request to the web service
with a JSON body (payload)

Web Service

Parses the request and sends a
JSON response to the client



Representing Data Formats in JSON

```
{  
  "id": 123456,  
  "title": "Journey to the Moon",  
  "year": 1925,  
  "pages": 250,  
  "genres": [  
    "Fiction",  
    "Drama",  
    "Sci-Fi"  
  ],  
  "rating": 4.5,  
  "version": 1  
}
```



Understanding the encoding/json package



Marshal

Transform a Go Struct to JSON

```
type Book struct {  
    ID      int64  
    Title   string  
    Published int  
    Pages   int  
    Genres  []string  
    Rating  float32  
    Version int32  
}
```

```
{  
    "ID": 123456,  
    "Title": "Beyond Reach",  
    "Year": 2010,  
    "Pages": "220",  
    "Genres": [  
        "Fiction",  
        "Sci-Fi"  
    ],  
    "Rating": 3.9  
}
```



Unmarshal

Transform JSON to a Go Struct

```
{  
    "id": 123456,  
    "title": "Beyond Reach",  
    "year": 2010,  
    "pages": "220",  
    "genres": [  
        "Fiction",  
        "Sci-Fi"  
    ],  
    "rating": 3.9  
}
```

```
> ID: 123456  
Title: Beyond Reach  
Published: 2010  
Pages: 220  
Genres: [Fiction Sci-Fi]  
Rating: 3.9  
Version: 1
```



Another Way to Transform JSON

Encode

Decode



Demo



Sending JSON requests

- Encode structs
- Write JSON helper functions
- Envelope responses

Parse JSON responses

- Decoding JSON
- Restrict request input



Overview/ Summary



Understanding the JSON Package

- Encode \ Marshal
- Decode \ Unmarshal

Sending JSON Responses

Using Struct Tags

Envelope JSON Responses

Parse JSON requests



Up Next:

Adding a Database

