

FSIOS - Workshop

# Networking und JSON-Parsing

Dennys Plettlinger

# Inhalte

- Ziel des Workshops
- Anwendungsbeispiele
- Theorie des Schwerpunktes
- Datenquelle
- Demo
- Übungen

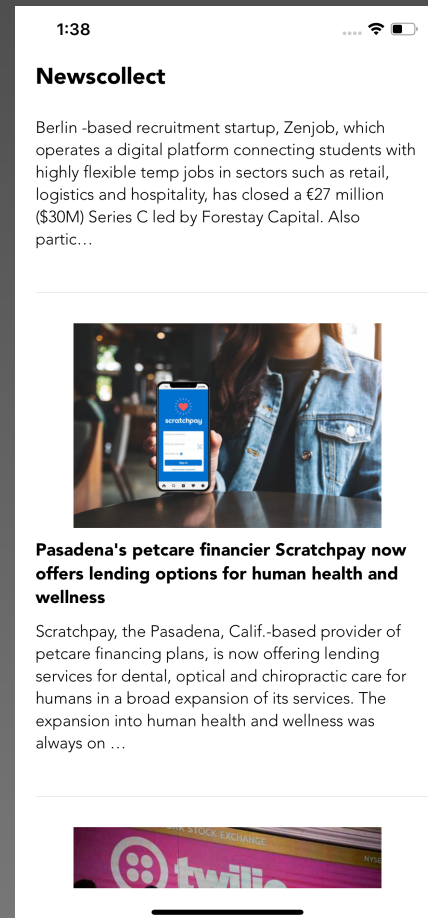
# Workshop Ressourcen

- Präsentationsfolien als PDF
  - <https://github.com/dennysplettlinger/swift-chucknorris-jokes>
- Anwendungsbeispiel „Newscollect“
  - <https://github.com/dennysplettlinger/swift-news-app>
- Vordefiniertes Projekt für die Übungen
  - <https://github.com/dennysplettlinger/swift-chucknorris-jokes>
- Cheatsheet für die Übungen
  - <https://github.com/dennysplettlinger/swift-chucknorris-jokes>

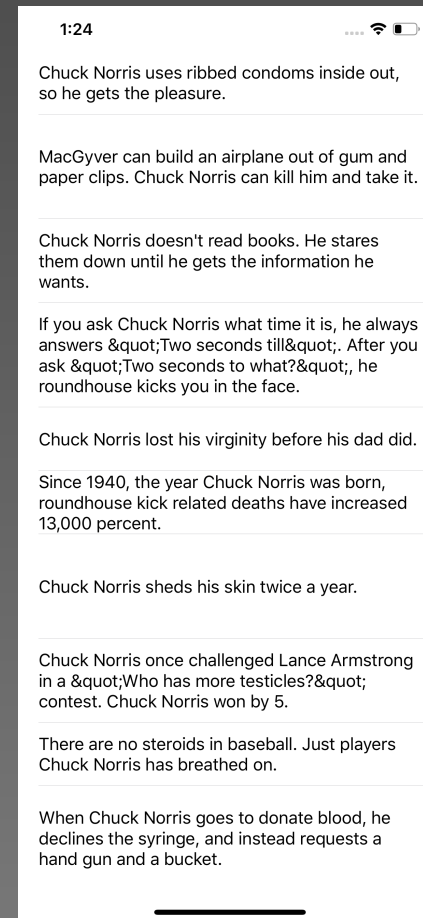
# Ziel des Workshops

Verstehen und Erlernen, wie Web APIs eingebunden  
und JSON-Daten verarbeitet werden

# Anwendungsbeispiele



Nachrichten App



Chuck Norris-Witze App

# Theorie des Schwerpunktes

## Multithreading

- Warum Multithreading? Verarbeitung von rechenintensiven Operationen im Hintergrund
- Multithreading mithilfe von GCD (Grand Central Dispatch)
- Gleichzeitige Verarbeitung von Vorgängen
- Beim Umgang mit GCD gibt es verschiedene Arten von Queues

# Theorie des Schwerpunktes

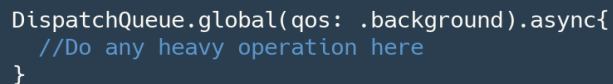
## Multithreading

1. Main Queue: wird im Hauptthread ausgeführt und hat die höchste Priorität

A code snippet in a dark-themed editor with three colored window control buttons (red, yellow, green) at the top left. The code is written in a light blue font on a dark blue background.

```
DispatchQueue.main.async{  
    //Do any UI updates here  
}
```

2. Global Queue: aufgeteilt in 4 verschiedene Typen
  - userInteractive: Vorgang passiert fast sofort, ähnlich wie beim Hauptthread
  - userInitiated: Vorgang Sekunden später
  - Utility: Vorgang dauert einige Sekunden bis einige Minuten
  - background: Vorgang nimmt viel Zeit in Anspruch (Minuten oder Stunden)


A code snippet in a dark-themed editor with three colored window control buttons (red, yellow, green) at the top left. The code is written in a light blue font on a dark blue background.

```
DispatchQueue.global(qos: .background).async{  
    //Do any heavy operation here  
}
```

# Theorie des Schwerpunktes

## Multithreading

- Verteilung von Aufgaben:



```
DispatchQueue.global(qos: .background).async{  
    //Do any heavy operation here  
    let image = downloadImageFromServer()  
    DispatchQueue.main.async {  
        self.imageView.image = image  
    }  
}
```



# Theorie des Schwerpunktes

## Netzwerkanfrage mit URLSession

- Warum eine Netzwerkanfrage? Um Daten über eine API anzufragen oder Daten senden zu können
- Netzwerkanfrage mithilfe der Klasse URLSession

# Theorie des Schwerpunktes

## Netzwerkanfrage mit URLSession

- Erstellung einer URL-Instanz aus dem angegebenen String (1)

```
let url = URL(string: "http://api.icndb.com/jokes")!
```

- Anwendungsweite geteilte Session (2)

```
let urlSession = URLSession.shared
```

- Erstellung eines Task, der Daten zurückliefert (3)

```
let task = urlSession.dataTask(with: url) { (data, response, error) in  
}
```

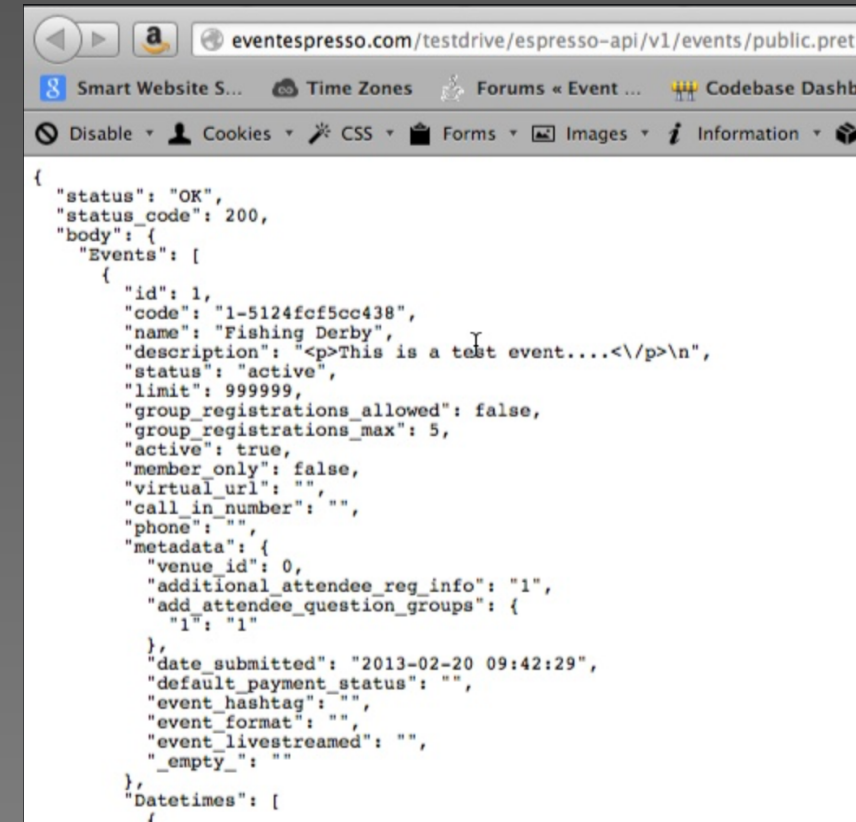
- Task starten (4)

```
task.resume()
```

# Theorie des Schwerpunktes

## JSON-Parsing mit dem Codable-Protocol

- Warum JSON-Parsing? Daten werden als gesamten String geliefert und müssen zu einem JavaScript-Objekt umgewandelt werden
- Umsetzung mit Klasse JSONDecoder und der Funktion decode()
- Veränderung von Eigenschaftsnamen mithilfe von CodingKeys



```
{
  "status": "OK",
  "status_code": 200,
  "body": {
    "Events": [
      {
        "id": 1,
        "code": "1-5124fef5cc438",
        "name": "Fishing Derby",
        "description": "<p>This is a test event...</p>\n",
        "status": "active",
        "limit": 999999,
        "group_registrations_allowed": false,
        "group_registrations_max": 5,
        "active": true,
        "member_only": false,
        "virtual_url": "",
        "call_in_number": "",
        "phone": "",
        "metadata": {
          "venue_id": 0,
          "additional_attendee_reg_info": "1",
          "add_attendee_question_groups": {
            "1": "1"
          },
          "date_submitted": "2013-02-20 09:42:29",
          "default_payment_status": "",
          "event_hashtag": "",
          "event_format": "",
          "event_livestreamed": "",
          "empty": ""
        },
        "Datetimes": [

```

# Theorie des Schwerpunktes

## JSON-Parsing mit dem Codable-Protocol

### - JSON-Struktur (1)

```
{  
  "name": "Luke Freaking Skywalker",  
  "age": 22,  
  "address": "Moisture Farm, Tatooine"  
}
```

### - JSON-Daten in Model konvertieren (3)

```
do {  
    let model = try JSONDecoder().decode(Person.self, from: jsonData)  
    print(model)  
} catch {  
    print("Error parsing JSON")  
}
```

### - Model (2)

```
struct Person: Codable {  
    var name: String  
    var age: Int  
    var address: String  
}
```

# Theorie des Schwerpunktes

## JSON-Parsing mit dem Codable-Protocol

### - JSON-Struktur (1)

```
{  
  "name": "Luke Freaking Skywalker",  
  "age": 22,  
  "address": "Moisture Farm, Tatooine"  
}
```

### - Model (2)

```
struct Person : Codable {  
    var name: String  
    var age: Int  
    var addressPlace: String  
  
    enum CodingKeys: String, CodingKey {  
        case name  
        case age  
        case addressPlace= "address"  
    }  
}
```

# Datenquelle

**News API**[Get started](#)[Documentation](#)[News sources](#)[Pricing](#)[Login](#)[Get API key](#)

## Search worldwide news with code

Get breaking news headlines, and search for articles from over 50,000 news sources and blogs with our news API

[Get API key](#)

All articles about Bitcoin from the last month, sorted by recent first

GET

http://newsapi.org/v2/everything?q=bitcoin&from=2020-04-04&sortBy=publishedAt&apiKey=API\_KEY

```
{
  status: "ok",
  totalResults: 3137,
  - articles: [
    - {
      - source: {
        id: null,
        name: "Apkmirror.com"
      },
      author: null,
      title: "Square Cash App 3.13.1 by Square, Inc.",
      description: "The post Square Cash App 3.13.1 by Square, Inc. appeared first on APKMirror.",
      url: https://www.apkmirror.com/apk/square-inc/square-cash/square-cash-3-13-1/
```

Top business headlines

GET

http://newsapi.org/v2/top-headlines?country=us&apiKey=API\_KEY

```
{
  status: "ok",
  totalResults: 70,
  - articles: [
    - {
      - source: {
        id: "the-washington-post",
        name: "The Washington Post"
      },
      author: "Laurie McGinley",
      title: "FDA steps up scrutiny of food and drug companies",
      description: "The Food and Drug Administration has stepped up its scrutiny of food and drug companies, announcing that it has taken action against more than 100 commercial corporations in the past year."
    }
  ]
}
```

# Datenquelle

```
Top headlines from TechCrunch right now

GET http://newsapi.org/v2/top-headlines?sources=techcrunch&apiKey=API_KEY
Y

{
  status: "ok",
  totalResults: 10,
  articles: [
    - {
      - source: {
        id: "techcrunch",
        name: "TechCrunch"
      },
      author: "Danny Crichton",
      title: "Competing with Disney+ and Quibi, Supreme Court launches streaming service",
      description: "It's a tough media world out there, and few institutions have been spared the transformations underway thanks to the internet. With Disney+ and Netflix posting records streaming numbers, and Quibi posting numbers that are perhaps a wee bit more bite-sized tha...",
      url: https://techcrunch.com/2020/05/04/competing-with-disney-and-quibi-supreme-court-launches-streaming-service/,
      urlToImage: https://techcrunch.com/wp-content/uploads/2020/05/GettyImages-521094906.jpg?w=541,
      publishedAt: "2020-05-04T14:31:01Z",
      content: "Its a tough media world out there, and few institutions have been spared the transformations underway thanks to the internet.\r\nWith Disney+ and Netflix posting records streaming numbers, and Quibi posting numbers that are perhaps a wee bit more bite-sized tha... [+1631 chars]"
    },
    - {
      - source: {
        id: "techcrunch",
        name: "TechCrunch"
      },
    },
  ],
}
```

```
ApiResponse {
  status: String
  totalResults: Int
  articles: [Article]
}
```

```
Article {
  title: String
  description: String
  urlToImage: String
}
```

# Demo



# Übungen

## Übung 1:

- Teil 1: Definiere die Struktur bzw. das Model des JSON-Objektes in einer eigenen Swift-Datei
- Teil 2: Nutze die Klassen URLSession und JSONDecoder um ein Array mit JSON-Objekten zu bekommen, in einer Schleife zu durchlaufen und die Jokes auszugeben.
- API (Daten): <http://api.icndb.com/jokes>

## Übung 2 (Optional):

- Lasse die Daten in Tabellenzellen ausgeben. Die Daten sollen dabei im Hintergrund geladen und der TableView angezeigt werden. (Die TableView soll dazu aktualisiert werden)