
DROIDS ON ROIDS
TeMoMuHa
PRESENT
iOS FROM THE VERY BEGINNING
#5
NetWORKing – Part I

# O MNIE

Piotr Sochalewski

iOS Developer od 2 lat

let Swift, Android Toast
Informatyka Biznesowa

# AGENDA

1. Wstęp merytoryczny
2. Przydatne narzędzia
3. Networking w iOS
4. Live coding
5. Materiały

# WSTĘP MERYTORYCZNY

# POJĘCIA

- REST API
- Request
- Response
- JSON
- OAuth

# REST API


**RE**presentation **S**tate **T**ransfer

# HTTP METHODS

PATCH

PUT

POST

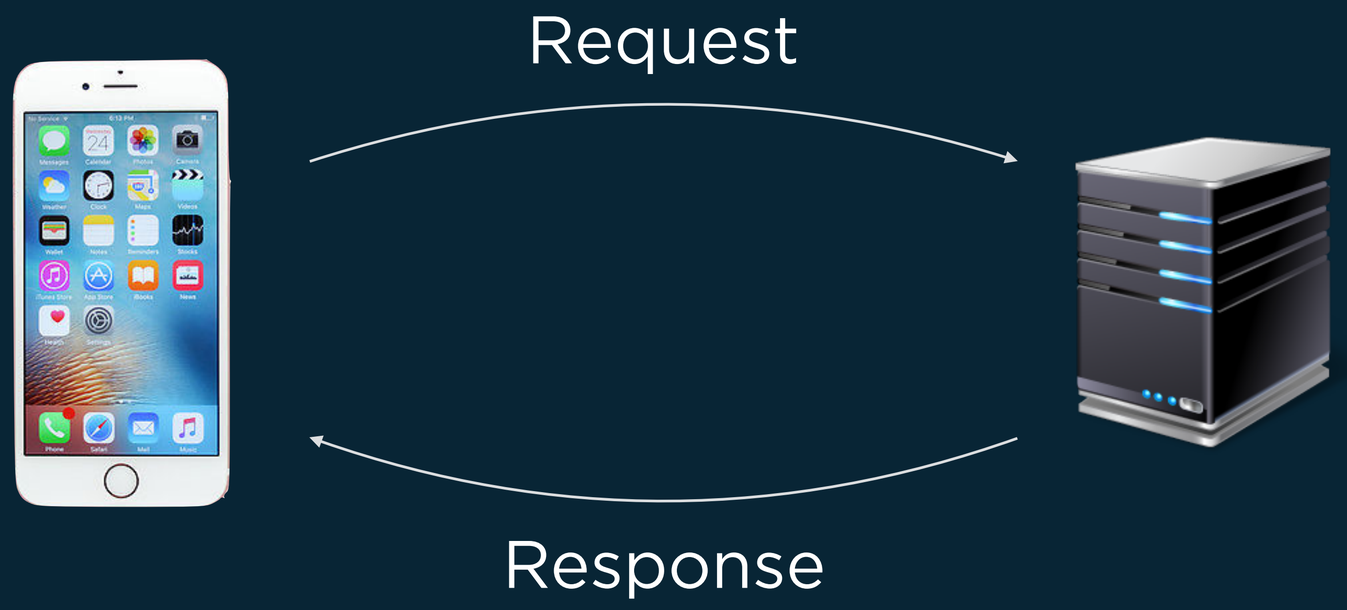
DELETE

GET

# HTTP METHODS

| zasób | metoda | wydarzenie |
|---|---|---|
| https://example.com/event | GET | pobranie wszystkich eventów |
| https://example.com/event/1 | GET | pobranie eventu o id 1 |
| https://example.com/event | POST | dodanie nowego eventu<br>(dane przesyłane w ciele żądania) |
| https://example.com/event | PUT | edycja istniejącego eventu<br>(dane przesyłane w ciele żądania) |
| https://example.com/event/1 | DELETE | usunięcie eventu o id 1 |

# REQUEST + RESPONSE

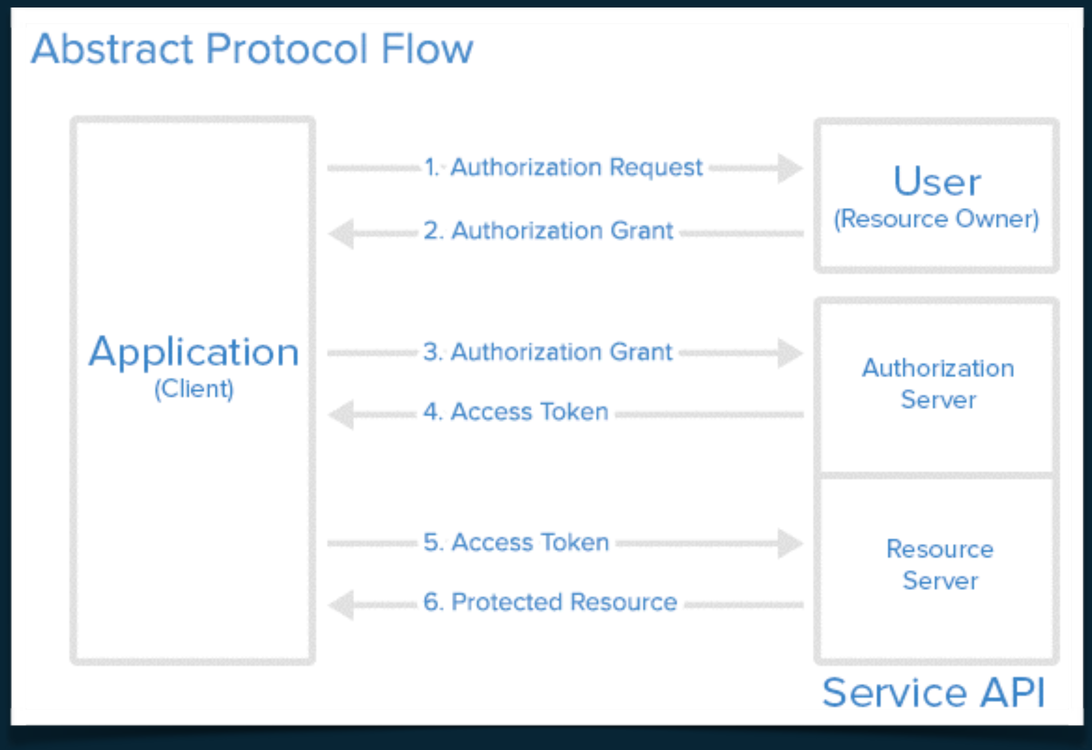


# JSON

```
GET https://iamrestapi.com/user?id=99
```

```
{
  "id": 99,
  "name": "Piotr",
  "last_name": "Sochalewski",
  "occupation": "iOS Developer",
  "year_of_birth": 1991,
  "photo": "https://iamrestapi.com/photo.jpg"
}
```

# OAUTH2



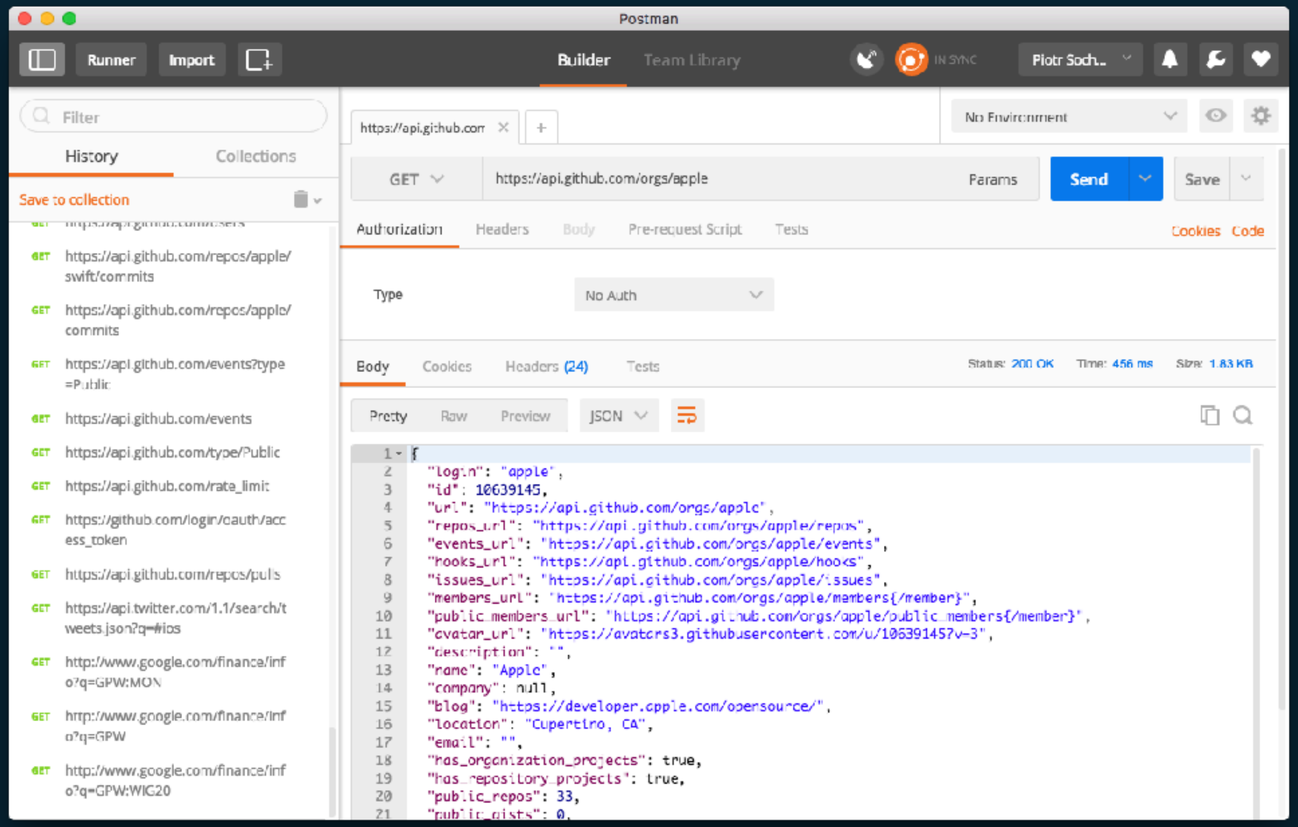

Źródło: Digital Ocean (https://www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2)

# PRZYDATNE NARZĘDZIA

# CURL

```
curl -i
  -X POST
  -H "Content-Type: application/json"
  -d '{"mail": "email@address.com"}'
  https://httpbin.org/ip
```

# POSTMAN


Postman
Runner
Import
Builder
Team Library
IN SYNC
Piotr Soch...
Filter
History
Collections
Save to collection
GET https://api.github.com/repos/apple/swift/commits
GET https://api.github.com/repos/apple/commits
GET https://api.github.com/events?type=Public
GET https://api.github.com/events
GET https://api.github.com/type/Public
GET https://api.github.com/rate_limit
GET https://github.com/login/oauth/access_token
GET https://api.github.com/repos/pulls
GET https://api.twitter.com/1.1/search/tweets.json?q=#ios
GET http://www.google.com/finance/info?q=GPW:MON
GET http://www.google.com/finance/info?q=GPW
GET http://www.google.com/finance/info?q=GPW:WIG20
https://api.github.com
No Environment
GET
https://api.github.com/orgs/apple
Params
Send
Save
Authorization
Headers
Body
Pre-request Script
Tests
Cookies
Code
Type
No Auth
Body
Cookies
Headers (24)
Tests
Status: 200 OK
Time: 456 ms
Size: 1.83 KB
Pretty
Raw
Preview
JSON
{
"login": "apple",
"id": 10639145,
"url": "https://api.github.com/orgs/apple",
"repos_url": "https://api.github.com/orgs/apple/repos",
"events_url": "https://api.github.com/orgs/apple/events",
"hooks_url": "https://api.github.com/orgs/apple/hooks",
"issues_url": "https://api.github.com/orgs/apple/issues",
"members_url": "https://api.github.com/orgs/apple/members{/member}",
"public_members_url": "https://api.github.com/orgs/apple/public_members{/member}",
"avatar_url": "https://avatars3.githubusercontent.com/u/10639145?v=3",
"description": "",
"name": "Apple",
"company": null,
"blog": "https://developer.apple.com/opensource/",
"location": "Cupertino, CA",
"email": "",
"has_organization_projects": true,
"has_repository_projects": true,
"public_repos": 33,
"public_gists": 0,

# NETWORKING W iOS

# NETWORKING W iOS

URLSession

Alamofire

Moya

# URLSession

The *NSURLSession* class and related classes provide an API for downloading content. This API provides a rich set of delegate methods for supporting authentication and gives your app the ability to perform background downloads when your app is not running or, in iOS, while your app is suspended.

# URLSession

```
let session = URLSession.shared

let url = URL(string: "https://httpbin.org/get")!
var request = URLRequest(url: url)
request.httpMethod = "GET"

let dataTask = session.dataTask(with: request) { data,
response, error in
    print("\(data), \(response), \(error)")
}

dataTask.resume()
```

# ALAMOFIRE

```
Alamofire.request("https://httpbin.org/get")
```

# ALAMOFIRE

```
Alamofire.request("https://httpbin.org/get", method: .get)
```

# ALAMOFIRE

```
Alamofire.request("https://httpbin.org/get", method: .get)
    .responseJSON { response in }
```

# ALAMOFIRE

```
Alamofire.request("https://httpbin.org/get", method: .get)
    .responseJSON { response in }
```

```
Alamofire.upload(
    multipartFormData: (MultipartFormData) -> Void,
    to: URLConvertible,
    encodingCompletion:
        ((SessionManager.MultipartFormDataEncodingResult) -> Void)?
)
```

# COCOAPODS

```
$ sudo gem install cocoapods

$ cd Snappy
$ pod init
$ open Podfile -a Atom
```

```
target 'Snappy' do
  use_frameworks!

  pod 'Alamofire'
end
```

```
$ pod install
$ open Snappy.xcworkspace
```

# LIVE CODING

github.com/DroidsOnRoids/Backend

# CLOSURES

*Closures* are self-contained blocks of functionality that can be passed around and used in your code.

Closures in Swift are similar to blocks in C and Objective-C and to lambdas in other programming languages.

```
{ (parameters) -> return type in
    statements
}
```

# CLOSURES

If a closure is passed as an argument to a function and it is invoked after the function returns, the closure is escaping.

```
func request(completion: ((Bool) -> Void)) {
    …
    completion(true)
}

func request(completion: @escaping ((Bool) -> Void)) {
    …
    Class.anotherCompletion { success in
        completion(success)
    }
}
```

# MATERIAŁY

# MATERIAŁY

- https://developer.apple.com/reference/foundation/urlsession
- https://developer.apple.com/reference/foundation/jsonserialization
- https://github.com/Alamofire/Alamofire
- https://www.raywenderlich.com/category/ios/networking