UNIVERSITÀ DEGLI STUDI DI NAPOLI
PARTHENOPE

Foundation
Program

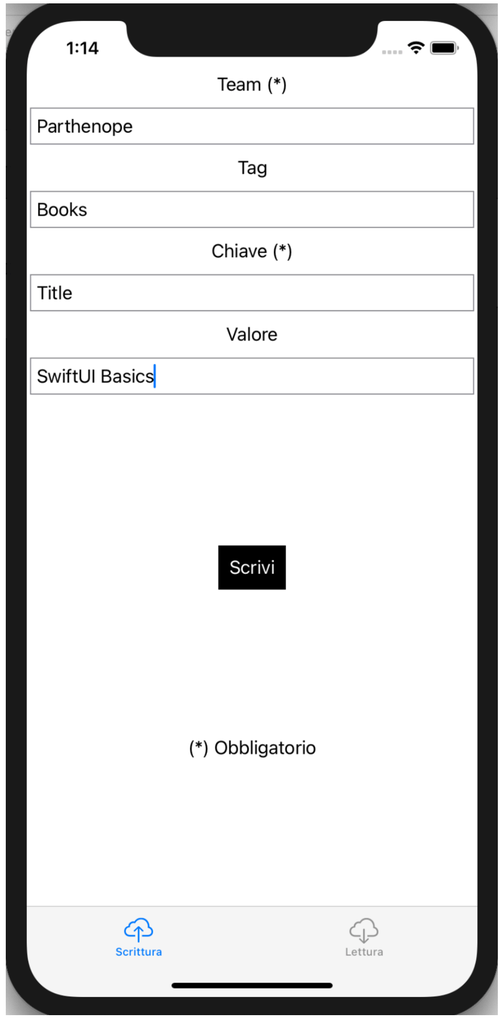
RESTful and Json

with ParthenoKit

Università degli Studi di Napoli
Parthenope

Foundation
Program

JSON(JavaScript Object Notation)




```
{
    "gruppo":"Parthenope",
    "chiave":"Title",
    "valore":"SwiftUI Basics",
    "sottogruppo":"Books"
 }
```

JSON formatted data is
human-readable and easier to parse

RESTful

Representational state transfer (**REST**) is a software architectural style that defines a set of constraints to be used for creating Web services.

Web services that conform to the REST architectural style, called **RESTful** Web services, provide interoperability between computer systems on the internet.

RESTful Web services allow the requesting systems to access and manipulate textual representations of Web resources by using a uniform and predefined set of stateless operations.

Other kinds of Web services, such as SOAP Web services, expose their own arbitrary sets of operations.

(from Wikipedia)

The `JSONSerialization` class

The `JSONSerialization` class it's provided by the Foundation Framework and it can convert a JSON into Swift data type like Dictionary, Array, String, Number and Bool

```swift
import Foundation


let data: Data // received from a network request, for example

let json = try? JSONSerialization.jsonObject(with: data, options: [])
```

The `JSONSerialization` class method `jsonObject(with:options:)` returns a value of type Any and throws an error if the data couldn't be parsed. Swift's built-in language features make it easy to safely extract and work with JSON data decoded with Foundation APIs — without the need for an external library or framework.

ParthenoKit

- A library written for educational purposes.
- It allows to save data to a remote server using RESTful calls
- Data are in JSON format
- Saved data are organized in a Dictionary

Note

- Since the iOS 9 release (2015) all the connections should ride on HTTPS otherwise the app will be not allowed to connect to the web service.
- To allow the connection to an HTTP service we have to add an NSAllowArbitraryLoads to the file info.plist
- Since January 2017 all the connections must ride on HTTPS otherwise the application will be not published on AppleStore

Class structure

```
import UIKit

public class ParthenoKit: NSObject {

    public func write(team: String, tag: String, key: String, value: Any, completion: @escaping (Bool)->Void)->Bool{
        let result = true

        //.......

        return result
    }

    public func read(team: String, tag: String, key: String, completion: @escaping (Any)->Void)->Bool{
        let ret = true

        //.......

        return ret
    }
}
```

The “write” method

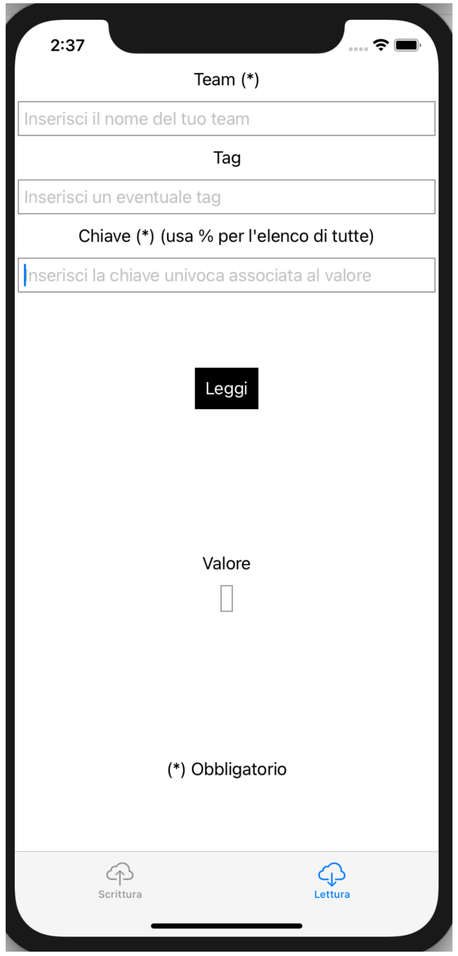
```
public func write(team: String, tag: String, key: String, value: Any, completion: @escaping (Bool)->Void)->Bool{
    var result = true

    let myUrl = "https://parthenope.testride.eu/wsfoundation.asmx/write"
    let parameters = ["gruppo" : team, "sottogruppo" : tag, "chiave" : key , "valore" : value] as [String : Any]
    guard let url = URL(string: myUrl) else {return false}
    var request = URLRequest(url: url)
    request.httpMethod = "POST"
    request.addValue("application/json", forHTTPHeaderField: "Content-Type")
    guard let httpBody = try? JSONSerialization.data(withJSONObject: parameters, options:[]) else {return false}
    request.httpBody = httpBody
    let session = URLSession.shared
    session.dataTask(with: request) {(data, response, error) in
        if let data = data {
            do {
                let json1: [String : Any]? = try JSONSerialization.jsonObject(with: data, options: []) as? [String : Any]

                result =  json1!["d"] as! Bool

                completion(result)

            } catch {
                result = false
            }
        }
    }.resume()
    return result
}
```

The “read” method

```
public func read(team: String, tag: String, key: String, completion: @escaping (Any)->Void)->Bool{
    var result = ""
    var ret = true

    let myUrl = "https://parthenope.testride.eu/wsfoundation.asmx/read"

    let parameters = ["gruppo" : team, "sottogruppo" : tag, "chiave" : key] as [String : Any]
    guard let url = URL(string: myUrl) else {return false}
    var request = URLRequest(url: url)
    request.httpMethod = "POST"
    request.addValue("application/json", forHTTPHeaderField: "Content-Type")
    guard let httpBody = try? JSONSerialization.data(withJSONObject: parameters, options:[]) else {return false}
    request.httpBody = httpBody
    let session = URLSession.shared
    session.dataTask(with: request) {(data, response, error) in
        if let data = data {
            do {
                let json1: [String : Any]? = try JSONSerialization.jsonObject(with: data, options: []) as? [String : Any]

                result =  json1!["d"] as! String
                if result.count>=2 {
                    if (result.first=="\"")&&(result.last == "\""){
                        result.removeFirst()
                        result.removeLast()
                    }
                }
                completion(result)

            } catch {
                ret=false
            }
        }
    }.resume()
    return ret
}
```

Challenge

Using SwitUI, develop this interface:



ContentView

```
import SwiftUI

struct ContentView: View {

    @State var sTeam = ""
    @State var sTag = ""
    @State var sKey = ""
    @State var sVal = ""
    @State var p = ParthenoKit()

    var body: some View {
        TabView{
            WriteView(sTeam: $sTeam, sTag: $sTag, sKey: $sKey, sVal: $sVal, p: $p)
                .tabItem {
                    Text("Scrittura")
                    Image(systemName: "icloud.and.arrow.up")
                }
            ReadView(sTeam: $sTeam, sTag: $sTag, sKey: $sKey, sVal: $sVal, p: $p)
                .tabItem {
                    Text("Lettura")
                    Image(systemName: "icloud.and.arrow.down")
                }
        }

    }
}
struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}
```

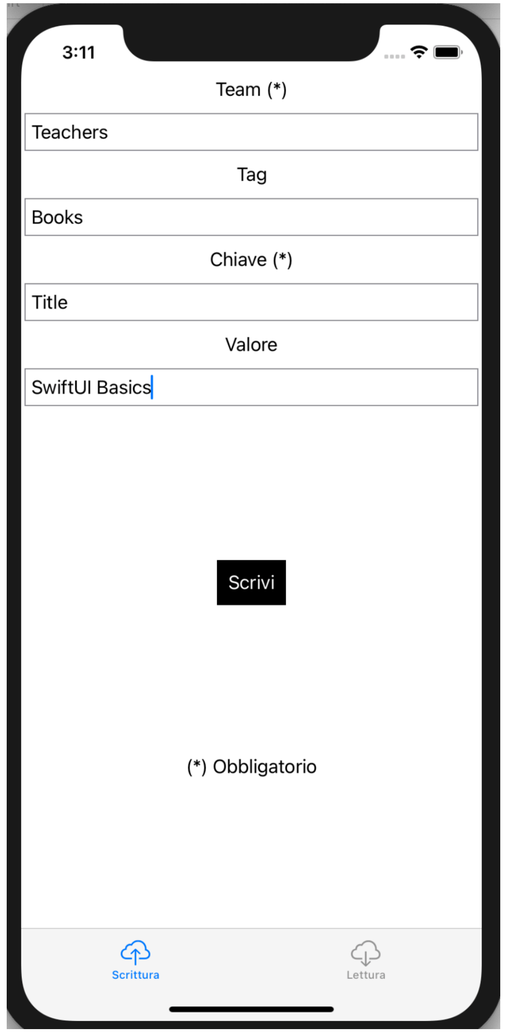
WriteView

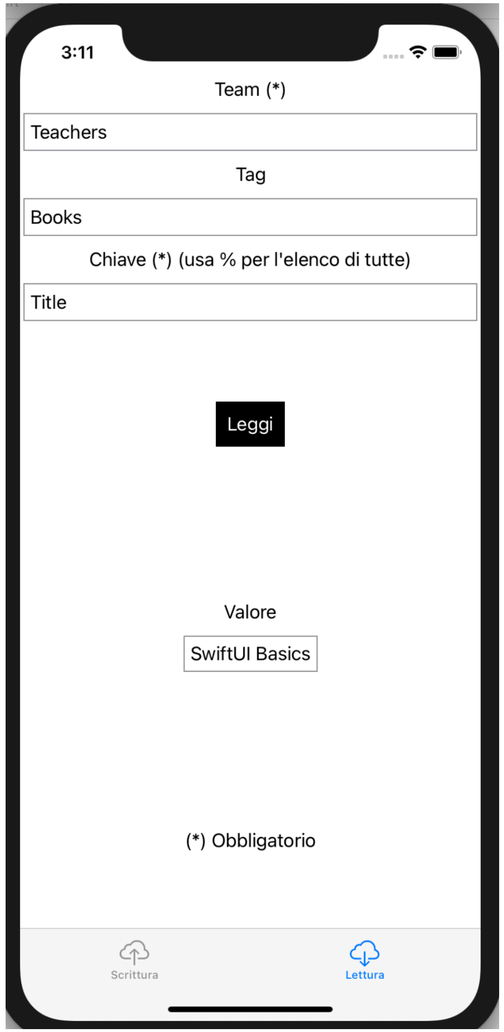
```swift
import SwiftUI


struct WriteView: View {
    @Binding var sTeam: String
    @Binding var sTag: String
    @Binding var sKey: String
    @Binding var sVal: String
    @Binding var p: ParthenoKit

    var body: some View {
        VStack{
            Group{
                Text("Team (*)")
                TextField("Inserisci il nome del tuo team", text: $sTeam)
                    .padding(6)
                    .border(Color.gray, width: 1)
                    .padding(3)

                Text("Tag")
                TextField("Inserisci un eventuale tag", text: $sTag)
                    .padding(6)
                    .border(Color.gray, width: 1)
                    .padding(3)

                Text("Chiave (*)")
                TextField("Inserisci la chiave univoca associata al valore", text: $sKey)
                    .padding(6)
                    .border(Color.gray, width: 1)
                    .padding(3)

                Text("Valore")
                TextField("Inserisci il valore da salvare", text: $sVal)
                    .padding(6)
                    .border(Color.gray, width: 1)
                    .padding(3)
            }

            Spacer()
            Button(action: {
                let result = p.write(team: sTeam, tag: sTag, key: sKey, value: sVal, completion: { ris in
                    if ris{
                        DispatchQueue.main.async {
                            sVal = ""
                        }
                    }else{
                        print("errore durante il salvataggio")
                    }
                })
                if result == false {
                    print("errore durante il salvataggio")
                }


            }) {
                Text("Scrivi")
                    .padding(10)
                    .foregroundColor(.white)
            }.background(Color.black)
            Spacer()
            Text("(*) Obbligatorio")
            Spacer()
        }
    }
}
```

ReadView

```swift
import SwiftUI


struct ReadView: View {

    @Binding var sTeam: String
    @Binding var sTag: String
    @Binding var sKey: String
    @Binding var sVal: String
    @Binding var p: ParthenoKit

    var body: some View {

        VStack{
            Group{
                Text("Team (*)")
                TextField("Inserisci il nome del tuo team", text: $sTeam)
                    .padding(6)
                    .border(Color.gray, width: 1)
                    .padding(3)

                Text("Tag")
                TextField("Inserisci un eventuale tag", text: $sTag)
                    .padding(6)
                    .border(Color.gray, width: 1)
                    .padding(3)

                Text("Chiave (*) (usa % per l'elenco di tutte)")
                TextField("Inserisci la chiave univoca associata al valore", text: $sKey)
                    .padding(6)
                    .border(Color.gray, width: 1)
                    .padding(3)
            }

            Spacer()

            Button(action: {
                let _ = p.read(team: sTeam, tag: sTag, key: sKey, completion: {ris in
                    DispatchQueue.main.async {
                        sVal =  ris as! String
                    }
                })
            })
            {
                Text("Leggi")
                    .padding(10)
                    .foregroundColor(.white)
            }.background(Color.black)
            Spacer()

            Group{
                Spacer()
                Text("Valore")
                if let valore = sVal {
                    Text(valore)
                        .padding(6)
                        .border(Color.gray, width: 1)
                        .padding(3)
                }

                Spacer()
                Spacer()
                Text("(*) Obbligatorio")
                Spacer()
            }
        }
    }
}
```

The app


3:11
Team (*)
Teachers
Tag
Books
Chiave (*)
Title
Valore
SwiftUI Basics
Scrivi
(*) Obbligatorio
Scrittura
Lettura


3:11
Team (*)
Teachers
Tag
Books
Chiave (*) (usa % per l'elenco di tutte)
Title
Leggi
Valore
SwiftUI Basics
(*) Obbligatorio
Scrittura
Lettura

And now…

…document, reflect and share!