

# A bit about JSON

Fabrice.Kordon@lip6.fr



# As an introduction...

## JSON?

- The other web service format
  - ▶ Simpler than XML
  - ▶ Smaller than XML
- Many use in web services too

## A bit of decoding in Swift/Objective-C

- (NS)JSONSerialization
  - ▶ Also used to encode in JSON some existing class (not presented)
  - ▶ Other mechanisms preferred for serialization

## Goal of this video

- providing a «minimum survival guide»

# JSONSerialization

## Conversion between JSON and objects

 Data (JSON flow) into Any?


▶ Do not forget the cast then

```
class func jsonObject(with data: Data,  
                      options opt: JSONSerialization.ReadingOptions = []) throws  
    -> Any
```

▶ Any? into a JSON stream

```
class func data(withJSONObject obj: Any,  
               options opt: JSONSerialization.WritingOptions = []) throws -> Data
```

▶ Also available from/to stream (files)

 Errors handled by means of exceptions

# A small example (1/2)

```
func JSON2Dictionnaire(_ s: String) -> [String: Any?] {
    let d = s.data(using: .utf8)! // converion into Data (for jsonObject)
    do {
        let dico = try JSONSerialization.jsonObject(with: d,
                                                    options: []) as? [String: Any]

        if dico == nil {
            return [:]
        } else {
            return dico!
        }
    } catch let error {
        print("Error \(error.localizedDescription)")
        return [:]
    }
}

func displayDic(_ d: [String:Any?]) {
    print ("-----")
    print("d = \(d)")
    print ("-----")
    print("name    = \(d["name"])")
    print("num     = \(d["address_num"])")
    print("street  = \(d["address_street"])")
}
```

# A small example (2/2)

```
let s1 = """"
{"name" : "Sorbonne Université" , "address_num" : 4, "address_street" : "Place Jussieu"}
""""

let d1 = JSON2Dictionnaire(s1)
displayDic(d1)

print ("-----")
let s2 = """"
{"name" : "Sorbonne Université" , "address_street" : "Place Jussieu"}
""""

let d2 = JSON2Dictionnaire(s2)
displayDic(d2)

print ("-----")
let s3 = """"
{"name" : "Sorbonne Université", address_street : "Place Jussieu"}
""""

let d3 = JSON2Dictionnaire(s3)
displayDic(d3)
```

# Execution

```
-----  
d = ["name": Optional(Sorbonne Université), "address_num": Optional(4), "address_street": Optional(Place Jussieu)]  
-----
```

```
name    = Optional(Optional(Sorbonne Université))  
num     = Optional(Optional(4))  
street  = Optional(Optional(Place Jussieu))  
-----
```

```
-----  
d = ["name": Optional(Sorbonne Université), "address_street": Optional(Place Jussieu)]  
-----
```

```
name    = Optional(Optional(Sorbonne Université))  
num     = nil  
street  = Optional(Optional(Place Jussieu))  
-----
```

```
Error The data couldn't be read because it isn't in the correct format.  
-----
```

```
d = [:]  
-----
```

```
name    = nil  
num     = nil  
street  = nil
```

# As a conclusion...

## JSON revealed....

- Very basic example
- More sophisticated components
  - ▶ JSONDecoder
  - ▶ JSONEncoder



## Serialization

- Possibility to have a better mapping...
  - ▶ Regular serialisation
  - ▶ NSCodering protocol