SCIENCES SORBONNE UNIVERSITÉ

NSXMLParser

Fabrice.Kordon@lip6.fr

LIP6

As an introduction…

- **Network access requires data parsing**
  - Useful to communicate structured information
    - **Exploit answers from servers**
- **XML = standard format for web services**
  - NSXMLParser is your friend

Principles

Parsing XML

- «À la SAX»
- Intensive use of delegation

Operating a parser

- Creation (à la Swift or à la Objective-C)
  - **Init with data or with an URL**

```
convenience init?(contentsOf url: URL)

init(data: Data)
```

- Setting up the delegate
  - **Answering to the NSXMLParserDelegate protocol**
- Start parsing

```
func parse() -> Bool
```

- The parser can provide informations about its state
  - **Properties lineNumber, columnNumber**

NSXMLParserDelegate

Main methods

- Start/end of parsing

```
func parserDidStartDocument(_ parser: XMLParser)

func parserDidEndDocument(_ parser: XMLParser)
```

- Begin/end of tag, inside a tag

```
func parser(_ parser: XMLParser,
            didStartElement elementName: String,
            namespaceURI: String?,
            qualifiedName qName: String?,
            attributes attributeDict: [String : String] = [:])

func parser(_ parser: XMLParser,
            didEndElement elementName: String,
            namespaceURI: String?,
            qualifiedName qName: String?)

func parser(_ parser: XMLParser,
            foundCharacters string: String)
```

- Error management

```
func parser(_ parser: XMLParser,
            parseErrorOccurred parseError: Error)

func parser(_ parser: XMLParser,
            validationErrorOccurred validationError: Error)
```

NSXMLParserDelegate

Main methods

- Start/end of parsing

```
func parserDidStartDocument(_ parser: XMLParser)

func parserDidEndDocument(_ parser: XMLParser)
```

- Begin/end of tag / inside a tag

```
func parser(_ parser: XMLParser,
            didStartElement elementName: String,
            namespaceURI: String?,
            qualifiedName qName: String?,
            attributes attributeDict: [String : String])

func parser(_ parser: XMLParser,
            didEndElement elementName: String,
            namespaceURI: String?,
            qualifiedName qName: String?)

func parser(_ parser: XMLParser,
            foundCharacters string: String)
```

- Error management

```
func parser(_ parser: XMLParser,
            parseErrorOccurred parseError: Error)

func parser(_ parser: XMLParser,
            validationErrorOccurred validationError: Error)
```

Management of DTD included too

Have a look at the FM

How it works


```
<book>
  <title isbn="0470879963">
    iPhone Application Development For Dummies
  </title>
</book>
```

didStartElement:**book**...

How it works


```
<book>
  <title isbn="0470879963">
    iPhone Application Development For Dummies
  </title>
</book>
```

didStartElement:title...attributes:{isbn= 0470879963}

How it works


```
<book>
  <title isbn="0470879963">
    iPhone Application Development For Dummies
  </title>
</book>
```

foundCharacters:"iPhone Application"

How it works


```
<book>
  <title isbn="0470879963">
    iPhone Application Development For Dummies
  </title>
</book>
```

foundCharacters:"Development For Dummies"

How it works


```
<book>
  <title isbn="0470879963">
    iPhone Application Development For Dummies
  </title>
</book>
```

didEndElement:**title**...

How it works


```
<book>
  <title isbn="0470879963">
    iPhone Application Development For Dummies
  </title>
</book>
```

didEndElement:book...

As a conclusion…

- **Very «classical» behaviour**
  - Including delegation

- **So?**
  - You may query web services
  - Update «MyMeteo»?
    - openweathermap.org also speak XML