



Lập trình iOS

Bài 3. *Đọc dữ liệu XML - JSON*

Ngành Mạng & Thiết bị di động





Nội dung

1. XML

- Khái niệm
- XML dùng để làm gì?
- Cú pháp XML

2. XML trong iOS

3. JSON

4. JSON trong iOS

5. Mở rộng



1.1. Khái niệm

- ❑ XML (viết tắt từ tiếng Anh EXtensible Markup Language, "Ngôn ngữ Đánh dấu Mở rộng") là ngôn ngữ đánh dấu với mục đích chung do W3C đề nghị, để tạo ra các ngôn ngữ đánh dấu khác.
- ❑ Mục đích chính của XML là đơn giản hóa việc chia sẻ dữ liệu giữa các hệ thống khác nhau, đặc biệt là các hệ thống được kết nối với Internet.



1.2. XML dùng để làm gì

- ❑ Tạo ra một tiêu chuẩn thống nhất để truyền và trao đổi thông tin với nhau qua Internet bởi các web services.
- ❑ Tạo thuận lợi cho các trình ứng dụng đọc và xử lý thông tin dưới dạng XML.
- ❑ Các thông tin dưới dạng XML là human-legible và self-describing nghĩa là con người có thể đọc và hiểu được thông tin đó là về cái gì một cách dễ dàng.



1.3. Cú pháp XML

❑ Cú pháp XML cơ bản cho một phần tử:

- `<ten thuộc_tính="giá trị">nội dung</ten>`

❑ Ví dụ:

```
<?xml version="1.0" encoding="UTF-8"?>
<công_thức_nấu_ăn tên="bánh mì" thời_gian_chuẩn_bị="5 phút" thời_gian_nấu="3 tiếng">
  <title>Bánh mì cơ bản</title>
  <nguyên_liệu_lượng="3" đơn_vị="ca">Bột mì</nguyên_liệu>
  <nguyên_liệu_lượng="7" đơn_vị="gram">Men</nguyên_liệu>
  <nguyên_liệu_lượng="1.5" đơn_vị="ca" trạng_thái="ấm">Nước</nguyên_liệu>
  <nguyên_liệu_lượng="1" đơn_vị="thìa cà phê">Muối</nguyên_liệu>
  <chỉ_dẫn>
    <bước>Trộn tất cả các nguyên liệu với nhau và nhào kĩ</bước>
    <bước>Phủ một mảnh vải, ủ một tiếng đồng hồ trong phòng ấm.</bước>
    <bước>Nhào lại, đổ vào khuôn, cho vào lò nướng.</bước>
  </chỉ_dẫn>
</công_thức_nấu_ăn>
```



1.3. Cú pháp XML

- ❑ XML đòi hỏi rằng các phần tử phải được lồng nhau một cách đúng đắn — các phần tử không được có phần xen vào nhau.
- ❑ ***<!-- SAI! ĐỊNH DẠNG KHÔNG ĐÚNG! -->***

```
<p>Normal <em>emphasized <strong>strong emphasized</em>
strong</strong></p>
```



1.3. Cú pháp XML

- ❑ Mỗi tài liệu XML phải có đúng một phần tử gốc tại bậc trên cùng (còn gọi là *phần tử văn bản*), do đó đoạn sau cũng sẽ là một tài liệu XML định dạng sai:

- ❑ **<!-- SAI! ĐỊNH DẠNG KHÔNG ĐÚNG! -->**

```
<?xml version="1.0" encoding="UTF-8"?>  
<đồ vật>Đồ vật thứ nhất</đồ vật>  
<đồ vật>Đồ vật thứ hai</đồ vật>
```



1.3. Cú pháp XML

- ❑ XML cung cấp cú pháp đặc biệt để biểu diễn một phần tử với nội dung rỗng. Thay vì viết một thẻ bắt đầu và một thẻ kết thúc ngay sau đó, tài liệu có thể chứa thẻ phần tử rỗng mà trong đó dấu gạch chéo đứng ngay sau tên phần tử.

```
<foo></foo>  
<foo />
```




1.3. Cú pháp XML

- ❑ XML cung cấp hai phương pháp biểu diễn các ký tự đặc biệt: các tham chiếu thực thể (entity reference) và các tham chiếu ký tự số (numeric character reference).
- ❑ Trong XML, một thực thể (*entity*) là một phần dữ liệu được đặt tên với dữ liệu thường là text, chẳng hạn một ký tự đặc biệt.
- ❑ Một tham chiếu thực thể là một ký hiệu đại diện cho thực thể đó.



1.3. Cú pháp XML

- ❑ XML có năm thực thể đã được khai báo trước

Ký tự đặc biệt	Tập hợp ký tự biểu diễn
&	&
<	<
>	>
'	'
"	"

- ❑ Ví dụ:

<tên-công-ty>AT&T</tên-công-ty>



Nội dung

1. XML

2. XML trong iOS

- Các Parser phổ biến
- So sánh các Parser
- Ví dụ với NSXMLParser

3. JSON

4. JSON trong iOS

5. Mở rộng



2.1. Các Parser phổ biến

- **NSXML:** là phương pháp được viết bằng Objective-C và là cách phân tích XML đơn giản nhất, tuy nhiên khó khăn trong việc tương tác như dạng DOM Parser.
- **libxml2:** là thư viện mã nguồn mở được tích hợp sẵn vào iPhone SDK và được cải tiến hơn so với NSXML.
- **TBXML:** là một phương pháp phân tích XML tương đối gọn. Nó chỉ cho phép “read-only”, không hỗ trợ XPath, không hỗ trợ validation nhằm tiêu thụ ít tài nguyên nhất có thể.



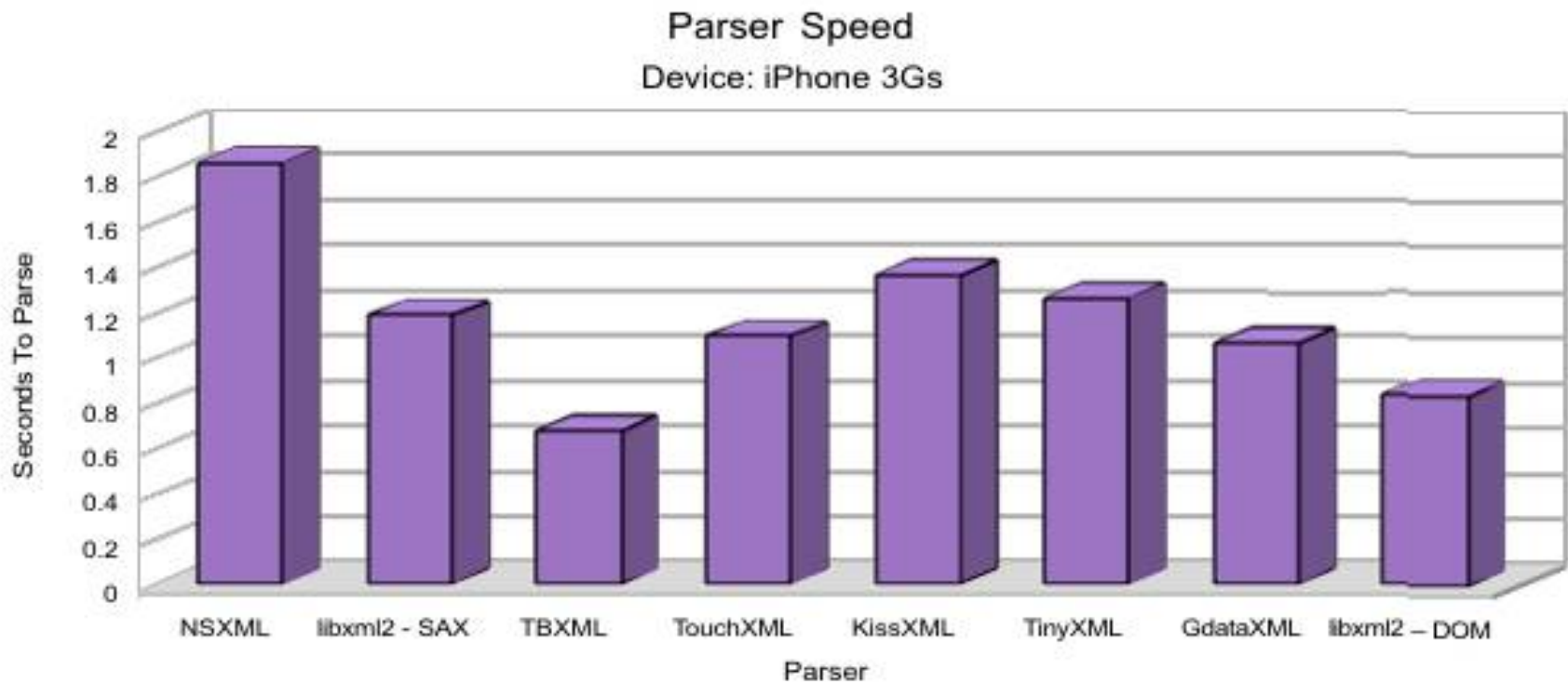
2.1. Các Parser phổ biến

- **TouchXML:** như là NSXML như tương tác được với DOM, chỉ cho phép “ read-only” nhưng có hỗ trợ Xpath.
- **KissXML:** phát triển dựa trên TouchXML tuy nhiên tích hợp thêm tính năng ghi và cập nhật cho XML
- **TinyXML:** là một thư viện nhỏ gọn viết bằng C chỉ gồm các tập tin .h và .m, hỗ trợ việc đọc, ghi XML tuy nhiên chưa hỗ trợ Xpath.
- **GDataXML:** được phát triển các Developer của Google, là thư viện tương tác DOM và hỗ trợ gần như đầy đủ các tính năng cần thiết



2.2. So sánh các Parser

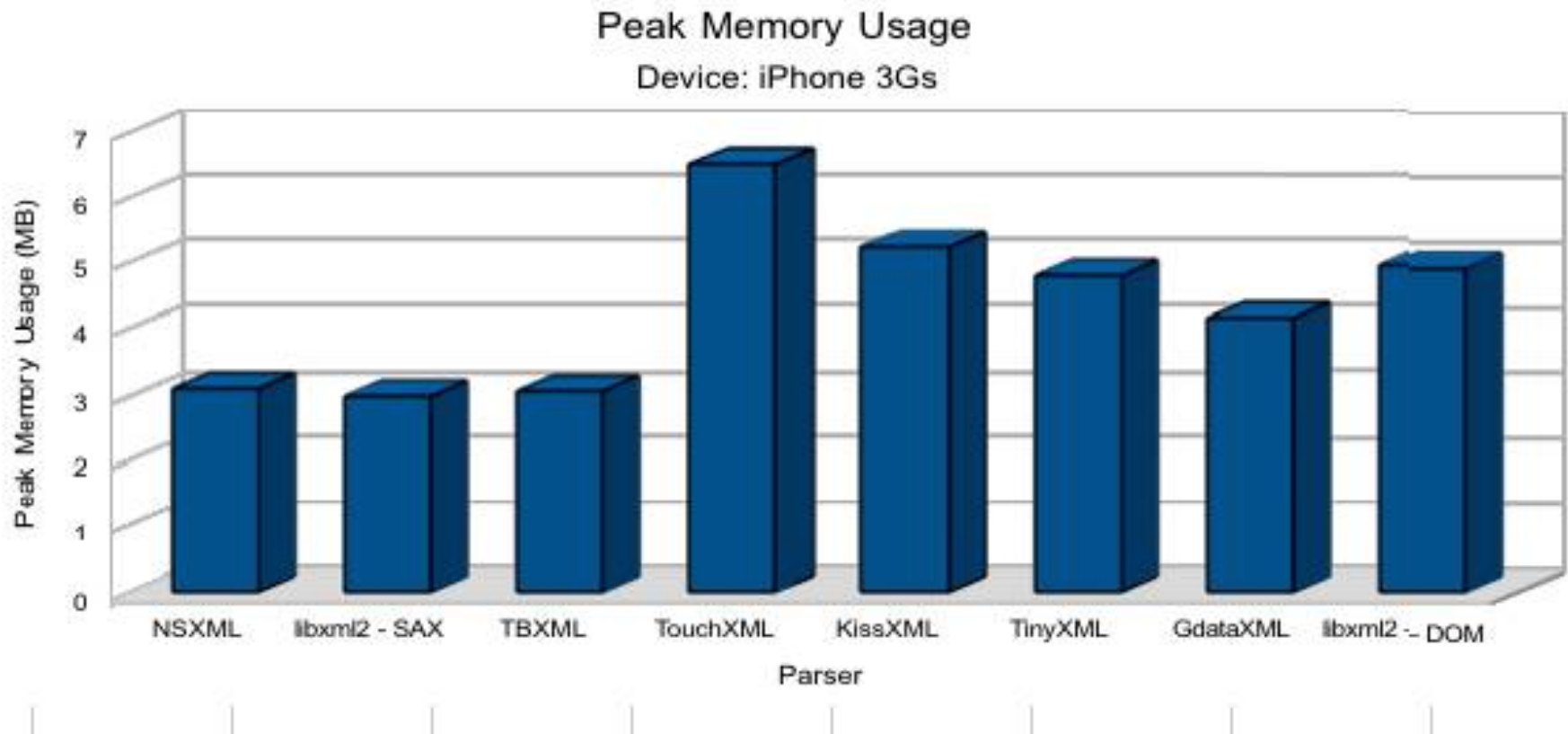
❑ Tốc độ





2.2. So sánh các Parser

□ Bộ nhớ





2.2. So sánh các Parser

□ Tổng quan

	NS	lib-SAX	TB	Touch	Kiss	Tiny	GData	lib-DOM
Tích hợp sẵn trong SDK	Y	Y	N	N	N	N	N	Y
Thời gian xử lý (giây)	1.87	1.19	0.68	1.1	1.37	1.27	1.07	0.84
Peak Memory Usage	3.11	3.01	3.07	6.5	5.25	4.8	4.15	4.97
Phân tích khi đang tải	N	Y	N	N	N	N	N	N
Ghi/sửa XML	N	N	N	N	Y	Y	Y	Y
Hỗ trợ XPath	N	N	N	Y	Y	N	Y	Y
C hay Obj-C	O	C	O	O	O	C	O	C
License	Apple	MIT	MIT	MIT	MIT	ZLib	Apache	MIT



2.3. Ví dụ với NSXMLParser

❑ Start of an element:

- - parser:didStartElement:namespaceURI:qualifiedName:attributes:

❑ Element Value

- - parser:foundCharacters:

❑ End of an element:

- - parser:didEndElement:namespaceURI:qualifiedName:



Nội dung

1. XML

2. XML trong iOS

3. JSON

- Các Parser phổ biến
- So sánh các Parser
- Ví dụ với NSXMLParser

4. JSON trong iOS

5. Mở rộng



3.1. Khái niệm JSON

- ❑ **JSON: viết tắt của JavaScript Object Notation. Nó là một chuẩn để định dạng dữ liệu, về mặt này, có thể so sánh JSON với XML, YAML**
- ❑ **JSON đi với JavaScript hoặc ActionScript thì nó có tính ưu việt hơn hẳn.**

```
{
  Nhóm1: [
    { HoTen: "Nguyen Van Giau", Tuoi: 18, Nam: true, NhómMau: "O" },
    { HoTen: "Tran Van Linh", Tuoi: 12, Nam: true, NhómMau: "A" },
    { HoTen: "Le Thi Ngoc", Tuoi: 25, Nam: false, NhómMau: "AB" }
  ],
  Nhóm2: [
    { HoTen: "Huynh Van Banh", Tuoi: 24, Nam: true, NhómMau: "B" },
    { HoTen: "Ngo Thi Em", Tuoi: 22, Nam: false, NhómMau: "A" },
  ]
}
```



3.2. Kiểu dữ liệu trong JSON

❑ Có 5 kiểu dữ liệu chính

- Number: kiểu số bao gồm số nguyên và số thực.
- String: kiểu chuỗi, nội dung bao bởi cặp dấu nháy kép “, những ký tự đặc biệt được escape bởi dấu \.
- Boolean: kiểu luận lý bao gồm 2 giá trị là true và false
- Array: kiểu mảng, gồm các phần tử phân cách nhau bởi dấu phẩy ‘,’ và mảng được bao bởi cặp dấu [và].
- Object: kiểu đối tượng, gồm những cặp giá trị đi cùng nhau, mỗi cặp phân cách bởi dấu phẩy ‘,’ , đối tượng được bao bởi cặp dấu { và }, cặp giá trị bao gồm tên và giá trị được phân cách bởi dấu hai chấm ‘:’ .



3.3. Ưu điểm của JSON

- Là kiểu dữ liệu trên nền cơ sở Javascript nên dễ dàng tiếp cận.
- Dữ liệu truyền tải ngắn gọn so với những định dạng dữ liệu khác như: XML, HTML...
- Dễ dàng chuyển đổi(parse) dữ liệu từ dạng chuỗi (nhận từ server) sang dữ liệu có thể sử dụng được (thành Object, Number, Array).
- Dễ truy cập nội dung.



Nội dung

1. XML

2. XML trong iOS

3. JSON

4. JSON trong iOS

- NSJSON Serialization
- Ví dụ Parsing JSON

5. Mở rộng



4.1. NSJSON Serialization

❑ Json Object

- **+ JSONObjectWithData:options:error:** Phương thức gọi trả về object từ JSON data.
- **+ JSONObjectWithStream:options:error:** Phương thức gọi trả về object từ đối tượng `NSInputStream`.

❑ Json Data

- **+ dataWithJSONObject:options:error:** Trả về dữ liệu JSON từ đối tượng
- **+ isValidJSONObject:** Phương thức gọi trả về true/false nhằm kiểm tra obj có phải là `JSONObject` hay không



4.2. Ví dụ Parsing JSON



Nội dung

1. XML

2. XML trong iOS

3. JSON

4. JSON trong iOS

5. Mở rộng



5. Mở rộng

- ❑ Giới thiệu về Google API
- ❑ Kích hoạt và sử dụng Google API



5.1 Giới thiệu về Google API

- ❑ Google APIs (Application Program Interfaces) là một tập các giao diện lập trình ứng dụng được phát triển bởi Google cho phép kết nối và sử dụng các dịch vụ của Google cũng như là các tích hợp vào các dịch vụ khác.
- ❑ Để sử dụng được các APIs này, ta cần phải chứng thực và ủy quyền sử dụng giao thức OAuth 2.0. Và ứng dụng ở phía người dùng sẽ yêu cầu một token để truy xuất từ máy chủ chứng thực của Google




5.1 Kích hoạt và sử dụng Google API

Getting started

Take a VM quickstart

Spin up virtual machines using Google Compute Engine, Node.js, and MongoDB to create a guestbook app




 [Get Started](#)

Enable Google APIs for use in your apps

Enable your app to take advantage of Google's products and services by using an API


API [Enable APIs](#)

Documentation

-  [Google Cloud Platform Documentation](#)
-  [Google Cloud Solutions](#)
-  [Google Cloud Tutorials](#)


Deploy a Hello World app

Learn how to use App Engine to create a Hello World app and deploy it in the cloud. App Engine lets you build and deploy an app without worrying about the underlying infrastructure.

 [Get Started](#)


Create a Cloud Storage bucket


Store your unstructured data safely and with high availability using Google Cloud Storage

 [Get Started](#)

Start with a prebuilt solution

Launch your favorite software packages on Google Cloud Platform with just a few clicks

 [Browse packages](#)

 [Create an empty project](#)

