



Lập trình iOS

Bài 2. *Giao thức kết nối Web Service*

Ngành Mạng & Thiết bị di động





Nội dung

1. Giới thiệu Web Service

- Web Service là gì?
- Tại sao phải cần Web Service?

2. Các loại Web Service

3. Sử dụng Web Service trong iOS

4. Ví dụ



1.1. Web Service là gì?

- ❑ Web Service là một thành phần trên mạng có những đặc tả cho biết cách thức thực thi và nội dung trả về mà máy có thể hiểu được.
- ❑ Ví dụ: Thao tác thanh toán hóa đơn điện thông qua dịch vụ thẻ sử dụng ứng dụng trên điện thoại di động.



1.2. Tại sao phải cần Web Service

- ❑ Các trang web chủ yếu thể hiện cho người duyệt web, không phải để cho máy hiểu.
- ❑ Cần một giao thức chung để máy có thể hiểu được và độc lập với môi trường và ngôn ngữ lập trình.
- ❑ Cần một ngôn ngữ độc lập với môi trường và ngôn ngữ lập trình để chuyển tải nội dung.
- ❑ Cần một giao thức để đăng tải dịch vụ và phương pháp để truy vấn các dịch vụ này.



Nội dung

1. Giới thiệu Web Service
2. Các loại Web Service
 - Giao thức HTTP
 - SOAP
 - REST
 - RESTFUL Service
3. Sử dụng Web Service trong iOS
4. Ví dụ



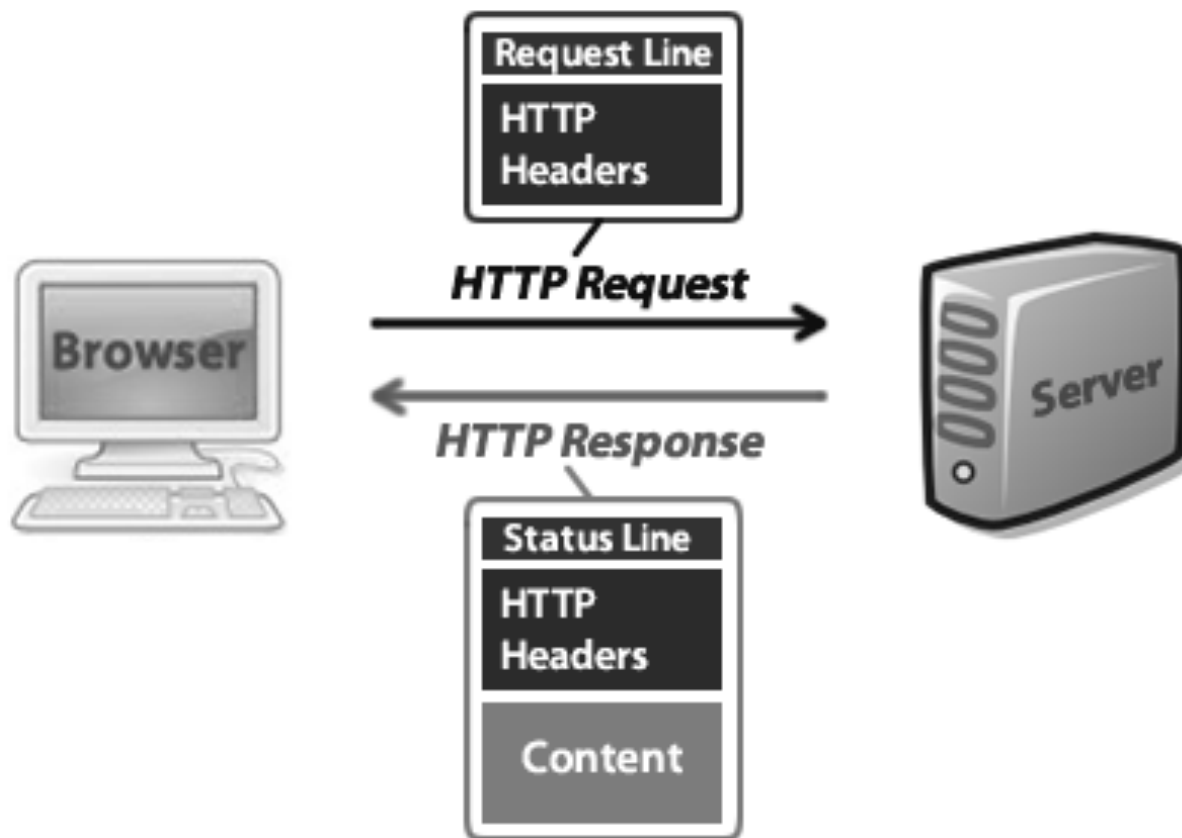
2.1. Giao thức HTTP

- ❑ **WWW (World wide web)** Là hệ thống các tài liệu liên kết với nhau thông qua internet.
- ❑ **HTTP (Hypertext Transfer Protocol)** Là giao thức mạng cho phép các hệ thống thông tin phân phối, và cộng tác với nhau. HTTP là nền tảng giao tiếp dữ liệu cho WWW.
- ❑ **HTTP hoạt động trên cơ chế giao thức request – response** trong mô hình điện toán client – server.



2.1. Giao thức HTTP

❑ Mô hình HTTP





2.1. Giao thức HTTP

❑ Ví dụ HTTP Request

```
GET http://www.fit.hcmus.edu.vn/vn/ HTTP/1.1

Accept: application/x-ms-application, image/jpeg, application/xaml+xml,
image/gif, image/pjpeg, application/x-ms-xbap, application/x-shockwave-flash,
application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword,

Accept-Language: vi-VN

User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1;
WOW64; Trident/4.0; GTB6.5; Mozilla/4.0 (compatible; MSIE 6.0; Windows
NT 5.1; SV1) ; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET
CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E;
OfficeLiveConnector.1.5; OfficeLivePatch.1.3; InfoPath.3;
AskTbGGSV5/5.8.0.12217) chrome/6.0.472.63

Accept-Encoding: gzip, deflate Connection: Keep-Alive

Host: www.fit.hcmus.edu.vn

Cookie:
.ASPXANONYMOUS=MG6LCiuSywEkAAAANTA2ZWNiYTAAtYThiNy00
MDA1LTkyNjUtYTllYzAxNTA3MTU10
```




2.1. Giao thức HTTP

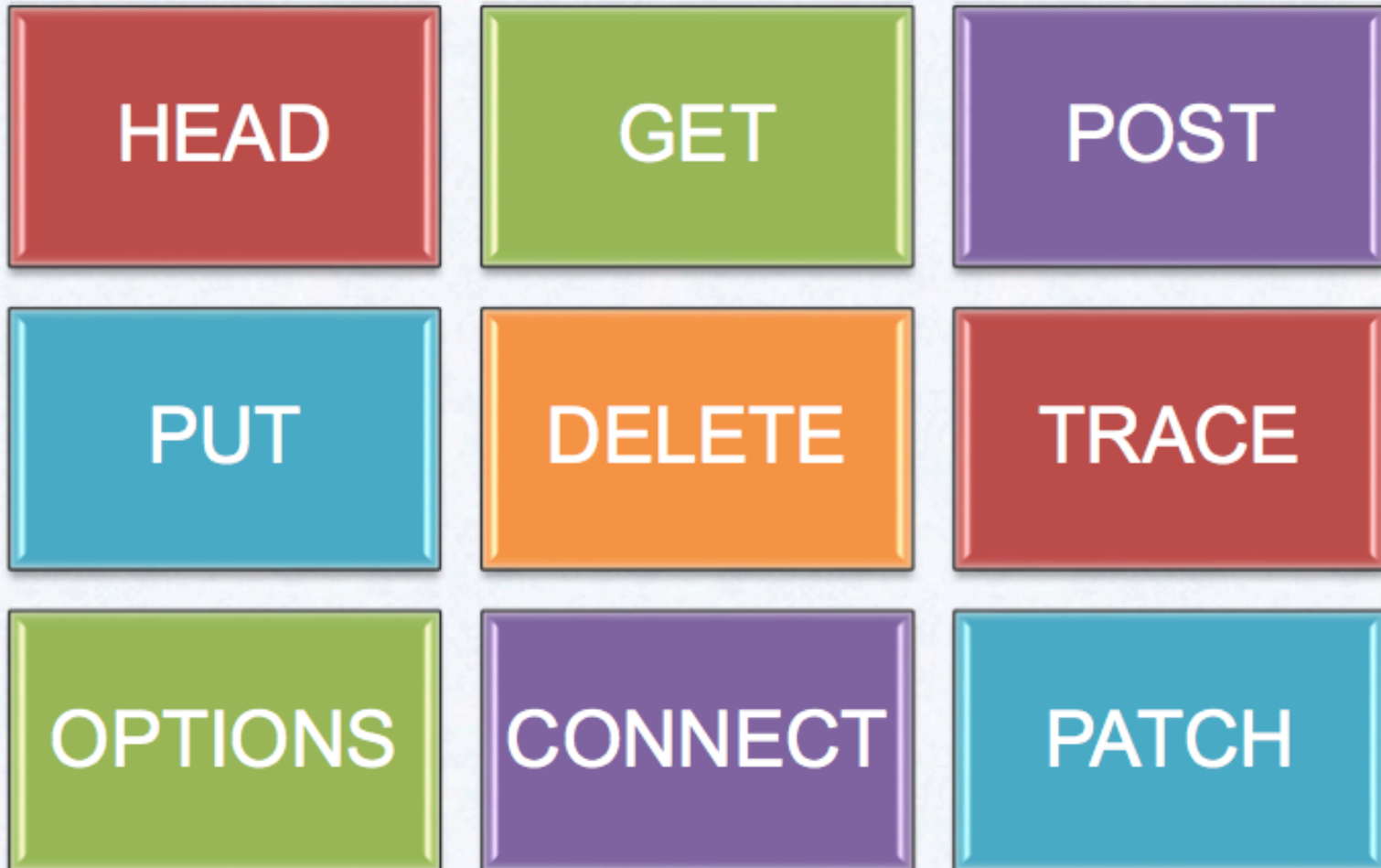
❑ Ví dụ HTTP Response

```
HTTP/1.0 200 OK
Cache-Control: private
Content-Type: text/html; charset=utf-8
Server: Microsoft-IIS/7.0
X-AspNet-Version: 2.0.50727
Set-Cookie: DotNetNukeAnonymous=7db3c001-c407-4adb-a60f-053b5dc76dc2; expires=Thu, 30-Sep-2010 03:12:33 GMT; path=/; HttpOnly
Set-Cookie: language=vi-VN; path=/; HttpOnly X-Powered-By: ASP.NET
Date: Thu, 30 Sep 2010 02:52:33 GMT Content-Length: 24404
X-Cache: MISS from vweb.hcmuns.edu.vn
Via: 1.0 vweb.hcmuns.edu.vn:80 (squid/2.6.STABLE16) Connection: keep-alive
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"> ...
// Nội dung trang web
```



2.1. Giao thức HTTP

❑ Các hàm, phương thức chính của HTTP





2.2. SOAP

- ❑ **SOAP - Simple Object Accesss Protocol** : là một giao thức giao tiếp có cấu trúc như XML và mã hóa thành định dạng chung cho các ứng dụng trao đổi với nhau
- ❑ Khái niệm cơ bản nhất của mô hình SOAP là việc sử dụng các tài liệu XML như những thông điệp trao đổi. Điều này có nhiều ưu điểm hơn các giao thức truyền dữ liệu khác. Các thông điệp XML có thể được tổng hợp và đọc với một bộ soạn thảo text đơn giản, ta có thể làm việc với XML trên hầu hết mọi nền tảng



2.2. SOAP

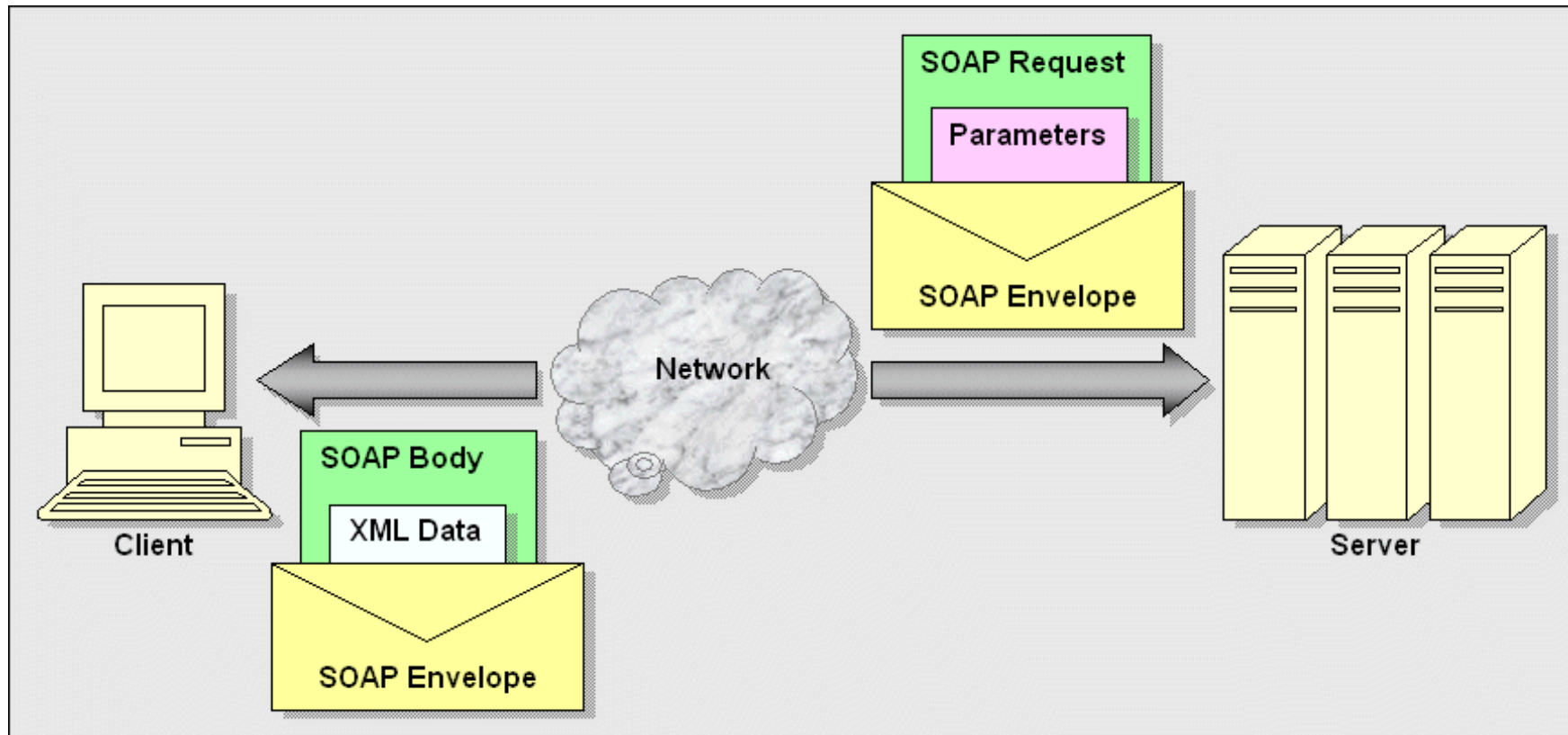
❑ Đặc trưng của SOAP

- SOAP được thiết kế đơn giản và dễ mở rộng.
- Tất cả các message SOAP đều được mã hóa sử dụng XML.
- SOAP sử dụng giao thức truyền dữ liệu riêng.
- Không có garbage collection phân tán, và cũng không có cơ chế tham chiếu. Vì thế SOAP client không giữ bất kỳ một tham chiếu đầy đủ nào về các đối tượng ở xa.
- SOAP không bị ràng buộc bởi bất kỳ ngôn ngữ lập trình nào hoặc công nghệ nào.



2.2. SOAP

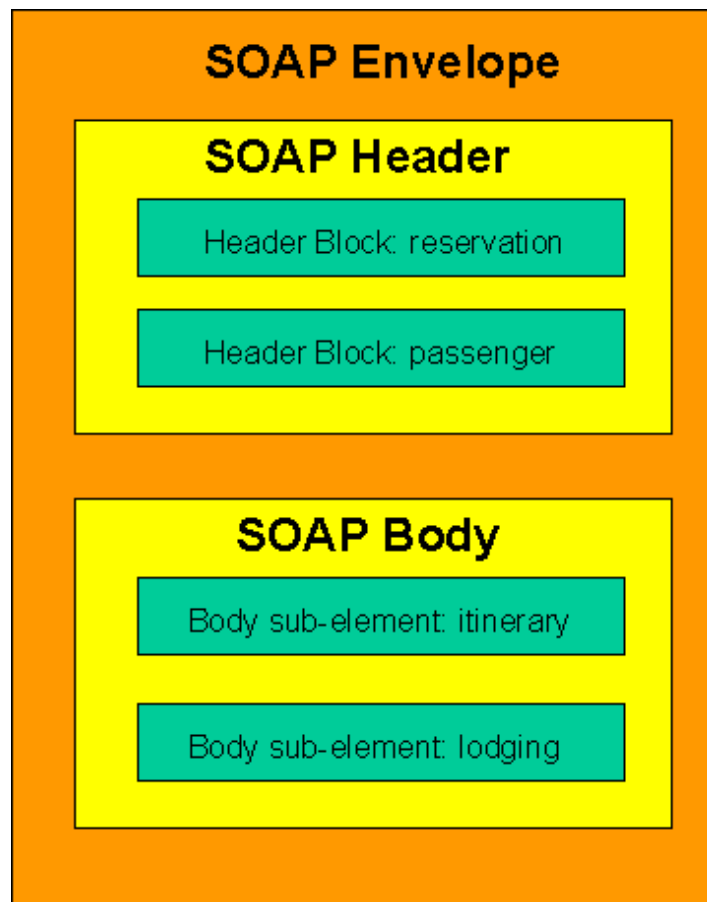
❑ Hệ thống SOAP đơn giản





2.2. SOAP

❑ Cấu trúc thông điệp SOAP





2.3. REST

- ❑ **REST (Representational State Transfer) là một kiểu kiến trúc phần mềm cho các hệ thống phân tán siêu truyền thông như là WWW**
- ❑ **Đặc trưng của REST**
 - Cho phép mỗi thành phần phát triển độc lập
 - Hỗ trợ đa nền tảng
 - Mỗi yêu cầu từ client phải có đủ thông tin cần thiết để server có thể hiểu được mà không cần phải lưu trữ thêm thông tin nào trước đó
 - Tất cả tài nguyên được truy cập thông qua một interface thống nhất (HTTP GET, PUT, POST, DELETE, ...)



2.4. RESTFULL SERVICE

- ❑ Là một web service đơn giản sử dụng HTTP và tính chất của REST.

- ❑ Là một tập tài nguyên các thành phần được định nghĩa:
 - URI gốc cho web service
 - MIME type hỗ trợ bởi web service
 - Tập hành động hỗ trợ bởi web service sử dụng HTTP method (GET, POST, PUT, DELETE)



Nội dung

1. Giới thiệu Web Service
2. Các loại Web Service
3. Sử dụng Web Service trong iOS
 - NSURLRequest
 - NSMutableURLRequest
 - NSURLConnection
4. Ví dụ



3. Sử dụng Web Service trong iOS

- ❑ Để truy xuất Web Service, iOS hỗ trợ các lớp để truy xuất và quản lý việc lấy dữ liệu từ một đường dẫn URL với các phương thức giống như **restful service**.
- ❑ Để cho thuận tiện việc truyền tải dữ liệu Web Service viết cho các thiết bị iOS thường được viết theo cấu trúc **JSON** hoặc **XML**.



3.1. NSURLRequest

❑ Đối tượng **NSURLRequest** đại diện cho việc tải một nội dung thông qua một **URL**.

❑ Các phương thức khởi tạo request:

- + **requestWithURL**: Tạo và trả về đối tượng tương ứng với URL với cache và timeout mặc định.
- - **initWithURL**: Khởi tạo và trả về đối tượng **NSURLRequest** với cache và timeout mặc định.
- + **requestWithURL:cachePolicy:timeoutInterval**: Khởi tạo và trả về đối tượng **NSURLRequest** với URL cache và timeout chỉ định.
- - **initWithURL:cachePolicy:timeoutInterval**: Khởi tạo và trả về đối tượng **NSURLRequest** với URL cache và timeout chỉ định.



3.1. NSURLRequest

❑ Các thuộc tính liên quan đến request:

- **NSURLRequestCachePolicy**: Trả về kiểu cache đã thiết lập.
- **HTTPShouldUsePipelining**: Trả về giá trị yêu cầu tiếp tục truyền dữ liệu hay không.
- **mainDocumentURL**: Trả về URL của document kết hợp với yêu cầu request.
- **timeoutInterval**: Trả về giá trị timeout của một request.
- **networkServiceType**: Trả về loại dịch vụ mạng được yêu cầu (wifi, 3G,...)
- **URL**: Trả về địa chỉ URL được request



3.1. NSURLRequest

❑ Các thuộc tính liên quan đến HTTP Request:

- **allHTTPHeaderFields**: Trả về một Dictionary các dữ liệu có trong header của request.
- **HTTPBody**: Trả về Data dữ liệu nội dung chính của request.
- **HTTPBodyStream**: Trả về dòng stream nội dung chính của request.
- **HTTPMethod**: Quy định giao thức của request. Mặc định là “GET”.
- **HTTPShouldHandleCookies**: Kết quả trả về cho biết có sử dụng cache cho request hay không
- **-valueForHTTPHeaderField**: Trả về giá trị đặc biệt của header gói request



3.1. NSURLRequest

❑ Chấp nhận request với mạng 3G:

- **allowsCellularAccess**: Trả về YES nếu cho phép sử dụng mạng 3G để request.
- **supportsSecureCoding**: Trả về YES nếu request thực hiện giao thức



3.2. NSMutableURLRequest

- ❑ NSMutableURLRequest là một lớp con của NSURLRequest hỗ trợ các nhà phát triển trong việc đóng gói các gói request đơn giản.
- ❑ Lớp NSMutableURLRequest chỉ phục vụ cho việc đóng gói một thông tin của request muốn gửi các request này phải các lớp khác như NSURLSession hoặc NSURLConnection.



3.2. NSMutableURLRequest

- ❑ Ngoài một số phương thức như của `NSURLRequest`, `NSMutableURLRequest` còn có một số phương thức như sau:
 - **-addValue: forHTTPHeaderField:** Thêm thông tin cho một trường dữ liệu trong gói header.
 - **-setAllHTTPHeaderFields:** Thay thế các giá trị của header bằng giá trị thông qua cho phép.
 - **-setValue: forHTTPHeaderField:** Thiết lập giá trị của các thuộc tính header có sẵn trong gói request. Ví dụ như user-agent,...



3.3. **NSURLConnection**

- ❑ **NSURLConnection** cung cấp các phương pháp để lấy nội dung từ một URL thông qua một **NSURLRequest** hoặc **NSMutableURLRequest**.
- ❑ Lớp này cung cấp các phương thức đơn giản để quản lý việc kết nối, nhận gửi dữ liệu thông qua delegate **NSURLConnectionDelegate** và **NSURLConnectionDataDelegate** và quản lý việc download thông qua **NSURLConnectionDownloadDelegate**



3.3. NSURLConnection

❑ Thông tin URL kết nối.

- **originalRequest** : Trả về bản sao của NSURLRequest yêu cầu kết nối ban đầu.
- **currentRequest** : Trả về yêu cầu kết nối hiện tại

❑ Xử lý dữ liệu đồng bộ

- **+sendSynchronousRequest:returningResponse:error:** Thực hiện tải nội dung theo NSURLRequest một cách đồng bộ chạy trong main thread



3.3. NSURLConnection

❑ Xử lý dữ liệu bất đồng bộ

- **+ connectionWithRequest:delegate:** Tạo và trả về một kết nối URL và bắt đầu tải dữ liệu từ request yêu cầu.
- **- initWithRequest:delegate:** Tạo và trả về một kết nối URL và bắt đầu tải dữ liệu từ request truyền vào.
- **+ sendAsynchronousRequest:queue:completionHandler:** Thực hiện tải dữ liệu từ Request thông qua hàng đợi và block hoàn tất.
- **-start:** Bắt đầu tải dữ liệu nếu chưa sẵn sàng
- **-cancel:** Hủy kết nối trong trường hợp bất đồng bộ



3.3. NSURLConnection

❑ NSURLConnectionDelegate - Quản lý việc kết nối của request:

- - **connection:willSendRequestForAuthenticationChallenge:**
Thông báo yêu cầu xác thực kết nối.
- -**connection:canAuthenticateAgainstProtectionSpace:** Gửi xác nhận xem delegate có đáp ứng được yêu cầu xác thực hay không.
- -**connection:didCancelAuthenticationChallenge:** Gửi yêu cầu hủy kết nối xác thực.
- -**connection:didReceiveAuthenticationChallenge:** Được gửi khi một yêu cầu xác thực là để tải nội dung.
- -**connection:didFailWithError:** Thông báo khi quá trình kết nối và tải dữ liệu không thành công kèm theo lỗi



3.3. NSURLConnection

❑ **NSURLConnectionDataDelegate** - Quản lý việc tải dữ liệu khi kết nối.

- **-connection:didReceiveResponse:** Thông báo server đã nhận được yêu cầu request và phản hồi lại thông báo response
- **-connection:didReceiveData:** Nhận dữ liệu từ server từng phần một mỗi phần quy định là NSData.
- **-connectionDidFinishLoading:** Thông báo kết nối đã hoàn tất.
- **-connection:willSendRequest:redirectResponse:** Thông báo URL phải chuyển hướng để tải theo yêu cầu request.



Nội dung

1. Giới thiệu Web Service
2. Các loại Web Service
3. Sử dụng Web Service trong iOS
4. Ví dụ



4. Ví dụ

- ☐ Tạo kết nối không đồng bộ.
- ☐ Tạo kết nối đồng bộ
- ☐ Sử dụng POST data
- ☐ Thiết lập timeout
- ☐ Sử dụng cache

Thảo luận

