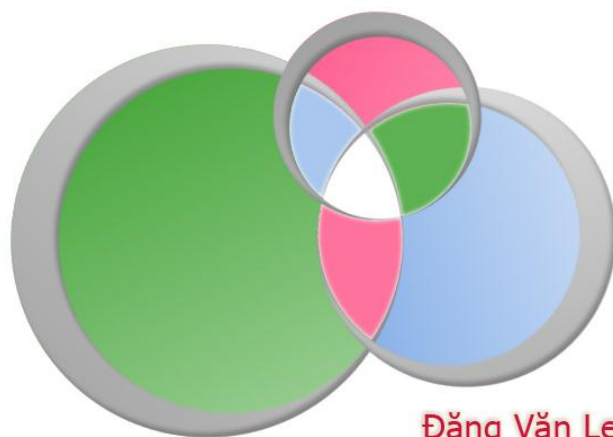


TÀI LIỆU CỦA THƯ VIỆN LẬP TRÌNH



Đặng Văn Lel

THƯ VIỆN LẬP TRÌNH²⁰¹²

Chinh phục đam mê - Chia sẻ tri thức

LẬP TRÌNH VÀ THIẾT KẾ
WEBSITE
TƯƠNG TÁC VỚI MẠNG XÃ HỘI
FACEBOOK

Dành Cho Người Mới Bắt Đầu

TẬP 2

Mục lục (nội tiếp tập 1)

PHẦN 3: GIỚI THIỆU FACEBOOK APIs	3
PHẦN 4: LÀM VIỆC VỚI GRAPH API.....	4
1. Nguyên lý hoạt động	4
1.1. Cổng giao tiếp HTTP	4
1.2. Dữ liệu gửi lên cổng giao tiếp	4
1.3. Kết quả trả về	4
1.4. Kết luận	5
2. Tạo ứng dụng đầu tiên.....	6
2.1. Tạo ứng dụng đại diện trên facebook Bước 1.....	6
2.2. Thiết lập các thông số cho ứng dụng đại diện Bước 2	10
2.3. Kết nối đến ứng dụng và tạo trang HTML để thử nghiệm Bước 3	13
2.4. Ứng dụng thực tế.....	23
3. Đọc dữ liệu từ Open Graph.....	24
3.1. Graph API Explorer là gì?	24
3.2. Truy cập và giao diện chính.....	24
3.3. Sử dụng Graph API Explorer.....	25
3.4. Tổng kết	28

PHẦN 3: GIỚI THIỆU FACEBOOK APIs

Tìm đọc tập 1 tại: <http://facebook.com/thuvienlaptrinh>

Trong tập này bạn sẽ được học nhiều hơn trong việc giao tiếp với facebook. Bạn sẽ có thể chủ động gửi bài viết, bình luận, chia sẻ, hoặc lấy và trình bày thông tin facebook ngay trên website của bạn. Tất cả những tiện ích đó được gom lại trong một nhóm được gọi là **Facebook APIs**. Sau đây là danh sách và định nghĩa các thành phần của Facebook APIs. Hãy đọc qua để có khái niệm ban đầu trước khi ta đi chi tiết từng phần một theo thứ tự (chỉ cần đọc qua khái niệm để biết, không cần suy nghĩ nhiều, khi đi vào từng phần chi tiết thì bạn sẽ hiểu rõ ràng hơn)

- ❖ **Graph API**: là API cơ bản nhất, cho phép bên ngoài có thể truy cập vào các đối tượng bên trong mạng lưới graph của facebook thông qua hình thức truy vấn HTTP. Hầu hết những API khác đều hoạt động dựa trên API này.
- ❖ **Facebook Query Language**: được gọi tắt là FQL, là API cung cấp một ngôn ngữ truy vấn cơ sở dữ liệu, giúp bên ngoài có thể trực tiếp truy vấn vào cơ sở dữ liệu của facebook và lấy ra những dữ liệu theo định dạng và số lượng mà họ mong muốn.
- ❖ **Open Graph**: tương tự như graph api, cũng giúp cho các ứng dụng có thể tạo ra được bài viết trên facebook, nhưng mang ý nghĩa "kể các hoạt động" của người dùng trên ứng dụng của bạn.
- ❖ **Dialogs**: facebook cung cấp sẵn một số hộp thoại dùng cho những việc như: đăng nhập bằng tài khoản facebook, gửi bài viết lên tường nhà của một người nào đó hoặc là gửi một yêu cầu cho bạn bè.
- ❖ **Chat**: có thể tích hợp facebook chat vào ứng dụng của bạn, mỗi sự tích hợp sẽ kết nối tới facebook thông qua dịch vụ Jabber/XMPP.
- ❖ **Localization and Translation**: công cụ giúp "nội địa hóa" các ứng dụng của bạn để thân thiện hơn về ngôn ngữ của từng vùng sử dụng.
- ❖ **Ads API**: giúp xây dựng các ứng dụng thay thế cho Facebook Ads Manager and Power Editor.

Bây giờ chúng ta sẽ bắt đầu học qua từng thành phần một.

PHẦN 4: LÀM VIỆC VỚI GRAPH API

1. Nguyên lý hoạt động

Nói về cách thức hoạt động của graph api, ta có thể chia ra làm 3 vấn đề chính: cổng giao tiếp HTTP, dữ liệu gửi lên cổng và kết quả trả về. Ta sẽ đi tìm hiểu các khái niệm này trước khi thực hành chi tiết về chúng.

1.1. Cổng giao tiếp HTTP

Graph API hoạt động dựa trên truy vấn HTTP. Mọi thao tác muốn thực hiện sẽ gửi một truy vấn HTTP đến cổng mà facebook cung cấp. Cú pháp cụ thể của một truy vấn lên cổng giao tiếp HTTP như sau:

Cú pháp

```
1 https://graph.facebook.com/<data>
```

Trong đó, "https://graph.facebook.com/" chính là cổng giao tiếp HTTP của graph api. Còn <data> sẽ là dữ liệu mà ta gửi đến cổng giao tiếp.

1.2. Dữ liệu gửi lên cổng giao tiếp

Cấu trúc của chuỗi <data> được gửi lên cổng giao tiếp sẽ có cú pháp tổng quát như thế này.

Cú pháp

```
1 https://graph.facebook.com/<action-data>&access_token=???
```

Thông tin này được chia làm 2 phần: action-data và access token

action-data	Là chuỗi dữ liệu mô tả một công việc cần thực hiện. Chia làm 2 mục đích sử dụng chính là " Lấy dữ liệu " và " Gửi dữ liệu ". Sẽ có cú pháp cụ thể sau.
access_token	Là khóa an ninh của truy vấn, khóa này đã được giới thiệu tập 1. Mọi truy vấn đến graph api đều phải có một khóa an ninh.

1.3. Kết quả trả về

Kết quả của mỗi truy vấn HTTP gửi đến graph api trả về là chuỗi dữ liệu dạng JSON^(?) hoặc là một thông báo lỗi.

Trường hợp thông báo lỗi chỉ xảy ra khi khai báo không đúng cú pháp của <data> hoặc do khóa an ninh không có quyền hạn để tiến hành công việc đang mong muốn.

(?) **JSON** là viết tắt của "**J**ava**S**cript **O**bject **N**otation", cấu trúc này dùng để lưu trữ thông tin, gần giống với XML, nhưng JSON thường được dùng khi cần lưu trữ những dữ liệu nhỏ và gọn. Vì vậy mà việc đọc và xử lý cũng nhanh hơn XML. Dưới đây là một ví dụ về dữ liệu JSON.

Ví dụ một chuỗi dữ liệu JSON

```
{
  "SinhVien":
  [
    { "HoVaTen": "Đặng Văn A", "NamSinh": 1989, "Lop": "DI0896A1" },
    { "HoVaTen": "Nguyễn Thị B", "NamSinh": 1990, "Lop": "DI0896A1" },
    { "HoVaTen": "Trần Văn C", "NamSinh": 1985, "Lop": "DI0896A1" }
  ]
}
```

Ý nghĩa các ký hiệu của JSON

{	Bắt đầu của một JSON
}	Kết thúc của một JSON
:	Đứng giữa tên thuộc tính và giá trị của thuộc tính
,	Là dấu ngăn cách giữa các thuộc tính trong cùng JSON, giữa các phần tử trong mảng hoặc là giữa hai JSON liên tiếp nhau.
[Ký hiệu bắt đầu của mảng dữ liệu, phần tử của mảng có thể là những JSON hoặc số hoặc chuỗi. Nhưng một mảng chỉ nên lưu trữ những phần tử cùng kiểu với nhau.
]	Ký hiệu kết thúc một mảng dữ liệu

Căn cứ vào ý nghĩa của những ký hiệu nêu trên, ta có thể phân tích dữ liệu trong ví dụ như sau: có 1 JSON gốc chứa mọi JSON còn lại (luôn luôn có một JSON gốc). JSON gốc chỉ có một thuộc tính tên "SinhVien", giá trị thuộc tính này là một mảng, trong mảng đó chứa 3 JSON. Mỗi JSON có 3 thuộc tính lần lượt là "HoVaTen", "NamSinh" và "Lop". Giá trị của 3 thuộc tính đó đi liền kề ngay sau tên thuộc tính.

- ❖ Tên thuộc tính đặt trong cặp dấu ngoặc kép ""
- ❖ Giá trị thuộc tính có thể là chuỗi, số, mảng và cũng có thể là json khác.

1.4. Kết luận

Graph API giao tiếp với bên ngoài thông qua cổng <https://graph.facebook.com>. Mỗi truy vấn HTTP gửi đến cổng giao tiếp đều bao gồm 2 thông tin: dữ liệu mô

tả công việc và khóa an ninh. Kết quả trả về từ graph api là một chuỗi theo định dạng JSON hoặc là một thông báo lỗi. Thông báo lỗi chỉ xảy ra khi bạn khai báo sai cấu trúc dữ liệu gửi đến cổng hoặc do khóa an ninh không có quyền hạn để tiến hành công việc đó hoặc khóa đã hết hạn sử dụng.

2. Tạo ứng dụng đầu tiên

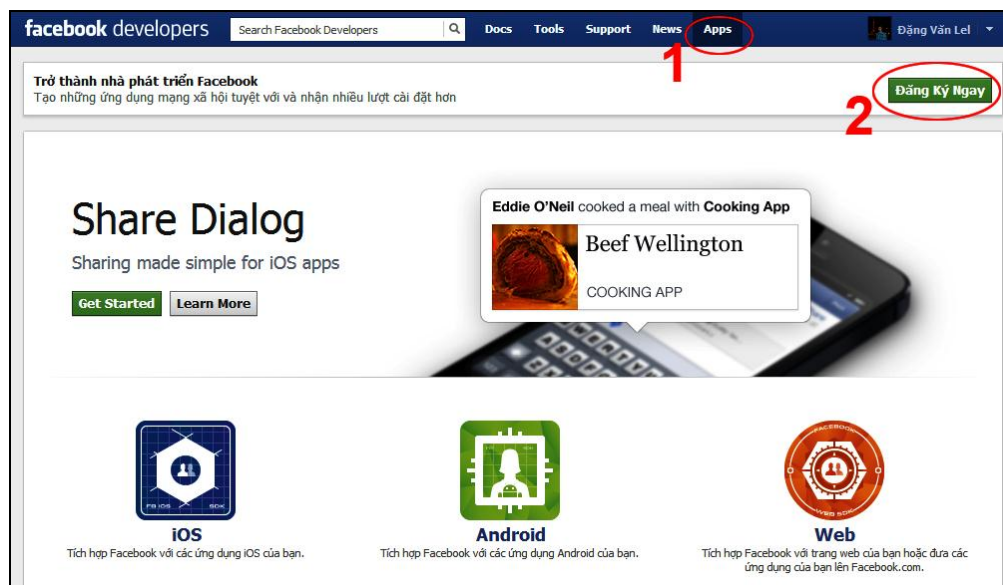
Bây giờ tôi sẽ hướng dẫn bạn tạo một trang HTML đơn giản, trong đó sẽ có phần kết nối đến facebook. Sau đó sẽ hiển thị một đoạn kết quả JSON mà graph api trả về khi truy cập thông tin cá nhân của người đang sử dụng.

2.1. Tạo ứng dụng đại diện trên facebook | Bước 1

Mọi website bên ngoài muốn sử dụng APIs của facebook đều phải tạo một ứng dụng trên facebook để làm đại diện cho website đó (xem lại tập 1). Đầu tiên, hãy truy cập vào:

<https://developers.facebook.com>

Nếu đây là lần đầu tiên bạn truy cập trang này, bạn sẽ thấy như hình dưới



Nếu bạn nhìn thấy **nút số 2**, nghĩa là tài khoản facebook của bạn chưa được đăng ký là một nhà phát triển. Bạn cần phải đăng ký để có thể sử dụng facebook apis. Còn nếu bạn đăng ký rồi thì hãy bấm vào **nút số 1** để đến trang quản lý ứng dụng trên facebook của bạn. (Do ngôn ngữ sử dụng của tài khoản facebook tôi dùng để chụp hình là Tiếng Việt nên có những nút và văn bản sẽ hiển thị Tiếng Việt).



ĐĂNG KÝ LÀ NHÀ PHÁT TRIỂN ỨNG DỤNG

Đầu tiên, bấm vào đăng ký (nút số 2 được khoanh đỏ như trên hình). Một popup sẽ hiển thị ra như sau:

1 Accept the terms. 2 Tell us about you. 3 Build your integration.

Growth
Use Facebook to grow your app and your audience

App Center
Distribute your app to millions of people

Tools and APIs
Start configuring your integration

Test Users
Create lightweight accounts to test your integration

☐ I accept the Facebook Platform Policy and the Facebook Privacy Policy

Tiếp Tục **Hủy bỏ**

Check vào "I accept the Facebook Platform Policy and the Facebook Privacy Policy" và bấm "Tiếp tục". Nội dung popup sẽ chuyển thành như hình sau:

1 Accept the terms. 2 Tell us about you. 3 Build your integration.

What's your role or title?
Lập trình viên

What's your experience level?
1 to 3 Years

Ngôn ngữ lập trình hoặc nền tảng nào bạn sử dụng?

- ☒ Web (HTML5, CSS3, JavaScript)
- ☐ iOS / Objective C
- ☒ Android
- ☒ Mobile web (HTML5, CSS3, JavaScript)
- ☐ PHP
- ☒ Java
- ☐ Ruby
- ☐ Python
- ☒ C#

Bạn lưu trữ và phân phối các ứng dụng của bạn ở đâu?

- ☐ Apple App Store
- ☐ Google Play
- ☐ Amazon Web Services
- ☐ Heroku
- ☐ Dedicated hosting
- ☒ Khác

Những loại ứng dụng nào bạn định xây dựng?

- ☒ Hình ảnh
- ☒ Tài chính
- ☒ Tham khảo
- ☒ Giáo dục
- ☒ Trò chơi
- ☒ Âm nhạc
- ☒ Thể thao
- ☒ Video
- ☒ Điều hướng
- ☒ Thời tiết
- ☒ Giải trí
- ☒ Y tế & Sức khỏe
- ☒ Tin tức
- ☒ Du lịch
- ☒ Ứng dụng cho trang
- ☒ Khác
- ☒ Sách
- ☒ Thời trang
- ☒ Phong cách sống
- ☒ Chương trình khuyến mãi và Cuộc thi
- ☒ Tiện ích
- ☒ Kinh doanh
- ☒ Năng suất
- ☒ Liên lạc
- ☒ Thức ăn & Đồ uống
- ☒ Địa phương
- ☒ Mua sắm

☒ Subscribe to weekly platform updates and platform status emails

☒ Like the Facebook Developers Page to stay up-to-date

Tiếp Tục **Bỏ Qua**

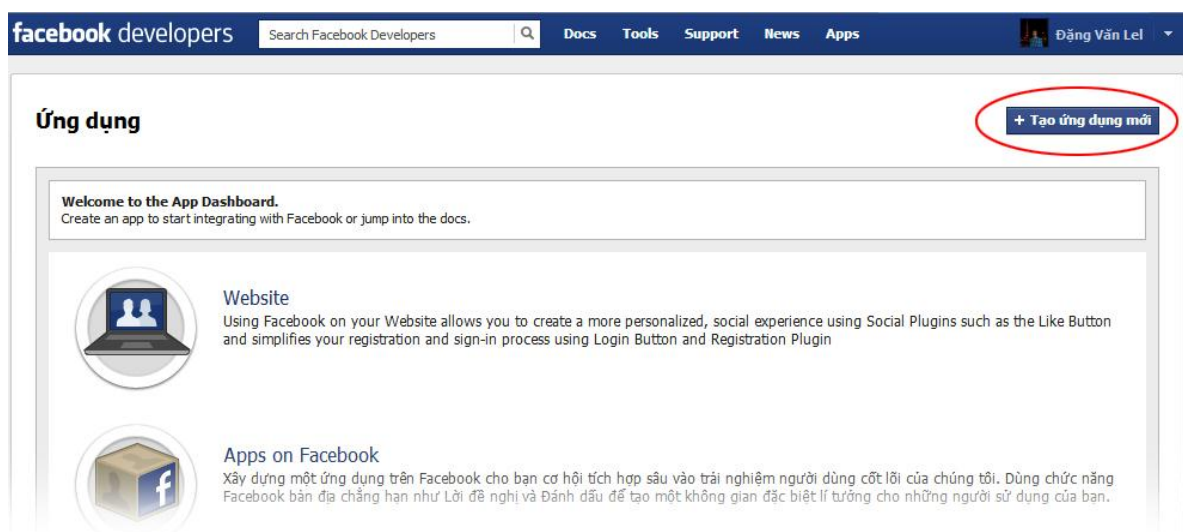
Lựa chọn và đánh dấu vào những thành phần mà bạn quan tâm, ở đây tôi chỉ lưu ý bạn 2 dấu check mà tôi dùng chỉ mũi tên đỏ ở hình trên. Nếu bạn check vào "Subscribe to weekly platform updates and platform status emails" nghĩa là bạn muốn facebook gửi email thông báo cho bạn mỗi khi facebook platform có thay đổi. Nếu bạn check vào "Like the Facebook Developers Page to stay up-to-date" nghĩa là bạn muốn thích trang Facebook Developers. Chọn xong, bấm "Tiếp tục", nếu màn hình popup hiển thị như hình dưới nghĩa là bạn đã đăng ký thành công và trở thành nhà phát triển trong danh sách của facebook.



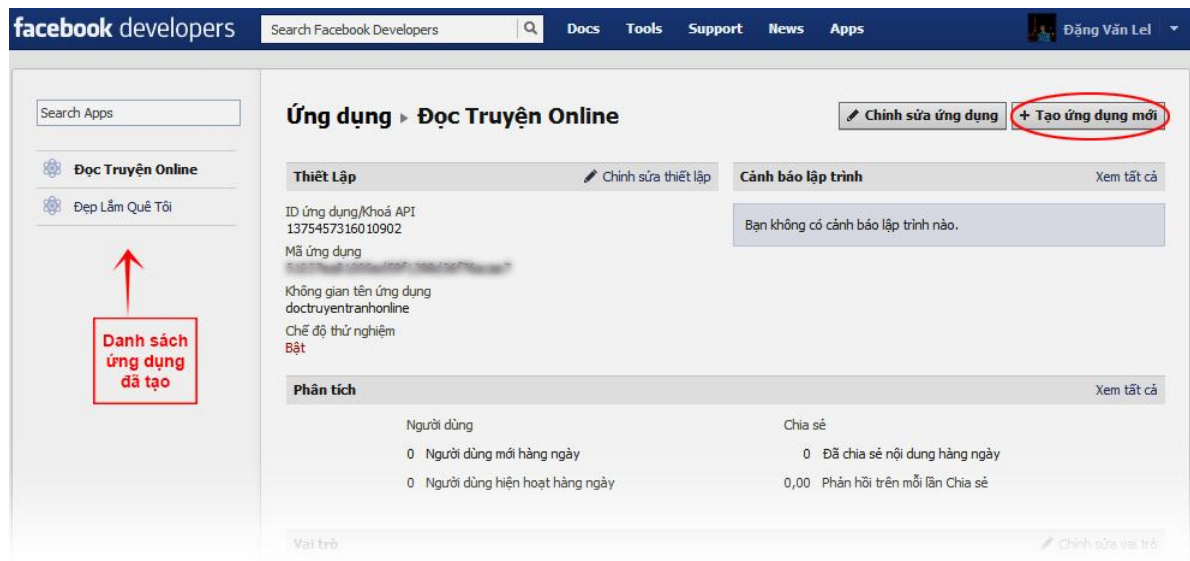
Bấm vào nút "Done" để tắt popup và bạn sẽ được chuyển tự động sang trang quản lý ứng dụng.

TẠO ỨNG DỤNG ĐẠI DIỆN

Giao diện của trang quản lý ứng dụng trong trường hợp bạn chưa có ứng dụng nào sẽ như hình bên dưới đây:



Và dưới đây là trường hợp bạn đã có ít nhất một ứng dụng được tạo trước đó.



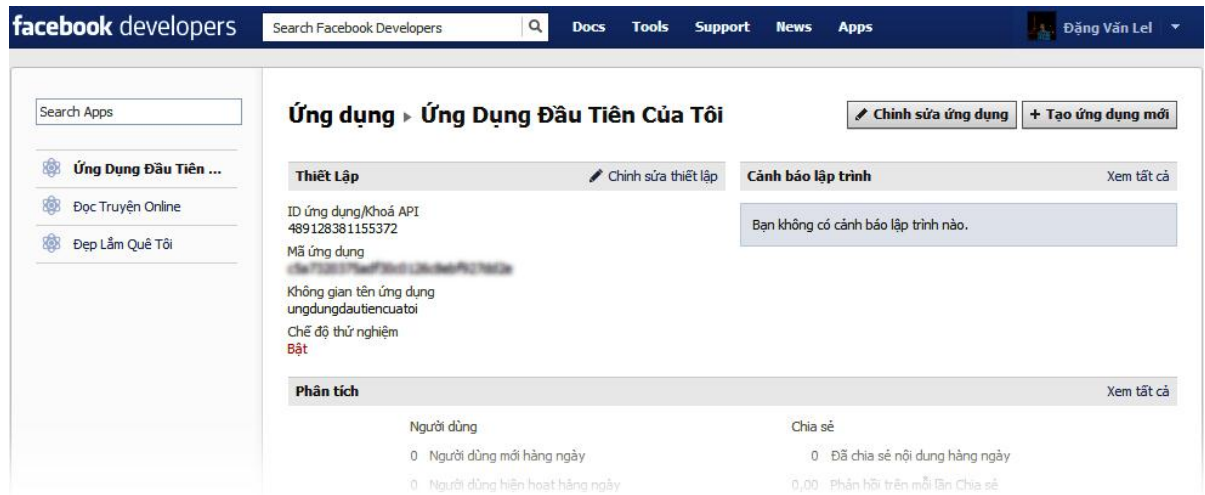
Bấm vào “Tạo mới ứng dụng”, một popup sẽ hiện ra như sau:

The screenshot shows the 'Tạo ứng dụng mới' (Create New App) form. The form contains several fields and a checkbox. The 'Tên ứng dụng' (App Name) field is highlighted with a red box and labeled '1'. The 'Không gian tên ứng dụng' (App Namespace) field is highlighted with a red box and labeled '2'. The 'App Category' dropdown is highlighted with a red box and labeled '3'. The 'Web Hosting' checkbox is highlighted with a red box. The form also includes a 'Hợp lệ' (Valid) status indicator and a 'Tiếp Tục' (Continue) button.

Số **1** là tên hiển thị của ứng dụng. Số **2** là tên namespace của ứng dụng (không dấu, không khoảng trắng). Ở số 3, lựa chọn một lĩnh vực của ứng dụng. Chú ý **không check** vào “Web Hosting”. Phần này sau này sẽ tìm hiểu sau. Bấm “Tiếp tục”, sau đó CAPCHA sẽ được yêu cầu nhập.

The screenshot shows the 'Require Captcha' page. The page displays a captcha image with the text '27sWmte' and a prompt to enter the text. The 'Tiếp Tục' (Continue) button is highlighted.

Nhập đúng CAPCHA rồi bấm “Tiếp tục” để xác nhận CAPCHA. Nếu nhập đúng, ứng dụng của bạn sẽ được tạo và màn hình sẽ hiển thị những thông tin cơ bản của ứng dụng đó như sau.



Tuy nhiên, để có thể sử dụng được ở website của bạn, thì bạn cần phải cấu hình một vài thông tin cần thiết như hướng dẫn ở bước 2 dưới đây.

2.2. Thiết lập các thông số cho ứng dụng đại diện | Bước 2

Bạn đã hoàn thành bước đầu tiên và đã tạo được một ứng dụng làm đại diện cho website của bạn trên facebook. Tuy nhiên, vẫn chưa sử dụng được. Bạn cần phải cấu hình vài thông tin nữa mới có thể sử dụng ứng dụng đó cho website của bạn. Bấm vào nút “Chỉnh sửa ứng dụng” để vào trang thay đổi các thông tin cho ứng dụng. Do giao diện phần này dài, nên tôi sẽ cắt ra từng phần một để hướng dẫn bạn.

PHẦN TIÊU ĐỀ CỦA ỨNG DỤNG



Phần này trình bày logo, tên hiển thị, ID, mã bí mật và trạng thái hoạt động của ứng dụng. Cụ thể như sau:

- Logo và tên là hai thành phần sẽ hiển thị ra bên ngoài. Bạn có thể thay đổi logo bằng cách rê chuột lên logo và bấm vào “chỉnh sửa”.

- ID là mã nhận dạng của ứng dụng trên facebook, mã này bạn sẽ sử dụng để kết nối với website của bạn.
- Mã bí mật của ứng dụng được sử dụng trong kết nối quản lý ứng dụng và không được công khai để đảm bảo chỉ có bạn mới có thể kết nối đến quản lý. Mã này có thể được tạo lại tự động bằng cách bấm vào "đặt lại".
- Trạng thái hoạt động của ứng dụng sẽ cho ta biết tình trạng mà ứng dụng đang chạy. Nếu bạn nhìn thấy "**Sandbox Mode**" nghĩa là ứng dụng đang chạy thử nghiệm, chế độ chạy thử nghiệm này sẽ làm ứng dụng chỉ có thể được sử dụng bởi chính người tạo ra hoặc những người được chỉ định là nhà thiết kế, kiểm thử của ứng dụng này.

PHẦN THÔNG TIN CƠ BẢN



Thông Tin Cơ Bản

Display Name: [?]

Namespace: [?]

Contact Email: [?]

App Domains: [?]

Hosting URL: [?] You have not generated a URL through one of our partners ([Get one](#))

Sandbox Mode: [?] ☒ **Mở** ☐ Vô hiệu hóa

- Display Name: là nơi bạn có thể đổi tên hiển thị của ứng dụng.
- Namespace: là nơi bạn có thể đổi namespace của ứng dụng.
- Contact Email: là nơi để bạn nhập và thay đổi địa chỉ email của bạn (chủ sở hữu ứng dụng).
- App Domains: phần này quan trọng nên tôi khoanh vùng đỏ. Dùng để khai báo tên miền của website mà ứng dụng này đại diện. Đa số các chức năng trên ứng dụng chỉ được áp dụng đối với website được khai báo ở đây. Chú ý là bạn chỉ cần khai báo tên miền, không bao gồm phần giao thức. Ví dụ như **congtvabc.com** chẳng hạn. Do tôi dùng localhost để làm ví dụ nên tôi chỉ khai báo là localhost (vì localhost không có phần .* ở cuối).
- Hosting URL: bỏ qua, tạm thời chưa dùng đến.
- Sandbox Mode: phần này cũng quan trọng, bạn có thể thay đổi chế độ chạy của ứng dụng này ở đây. Nếu chế độ Sandbox được chọn là "Mở" có

nghĩa là ứng dụng này đang chạy chế độ thử nghiệm và chỉ có tác dụng đối với những người dùng có liên quan như chủ sở hữu, nhà phát triển, người kiểm tra. Thường là tổng quá trình đang thử nghiệm thì nên để "Mở" còn khi cho vào sử dụng, lưu ý là phải "Vô hiệu quá" chế độ này và cho phép ứng dụng được sử dụng rộng rãi.

PHẦN THÔNG TIN CHỨC NĂNG

Khung tiêu đề ứng dụng và phần thông tin cơ bản là bắt buộc nên luôn luôn được mở ra cho phép bạn xem và chỉnh sửa thông tin. Tuy nhiên, tới những phần mang tính chức năng, khi bạn muốn sử dụng cái nào bạn mới kích hoạt nó. Cụ thể ta có danh sách các chức năng như sau:

Chọn cách ứng dụng của bạn tích hợp với Facebook	
<input checked="" type="checkbox"/> Trang web có Đăng nhập qua Facebook	Log in to my website using Facebook.
<input type="checkbox"/> Ứng dụng trên Facebook	Use my app inside Facebook.com.
<input type="checkbox"/> Website cho thiết bị di động	Bookmark my web app on Facebook mobile.
<input type="checkbox"/> Ứng dụng cho iOS	Publish from my iOS app to Facebook.
<input type="checkbox"/> Ứng dụng cho Android	Publish from my Android app to Facebook.
<input type="checkbox"/> Thẻ trang	Build a custom tab for Facebook Pages.

Mỗi một dòng trên là một chức năng mà Facebook Apis cung cấp cho ứng dụng của bạn. Tuy nhiên, mặc định thì chúng đều bị tắt. Để kích hoạt thì bạn chỉ đơn giản là click lên chức năng mà bạn muốn dùng. Ở đây, chúng ta tạm thời chỉ sử dụng chức năng ở hàng đầu tiên, chức năng "**Trang web có đăng nhập qua facebook**". Cái tên thì nó như vậy, nhưng đơn giản đây là chức năng mà người dùng có thể tương tác với dữ liệu Facebook của họ thông qua một website khác, và đó chính là cái ta sử dụng.

Khi click vào, nó sẽ xổ xuống thông tin như sau:

	Trang web có Đăng nhập qua Facebook	
	Site URL: [?]	<input type="text" value="http://localhost/FB_API"/>
<input checked="" type="checkbox"/>	Ứng dụng trên Facebook	Use my app inside Facebook.com.
<input type="checkbox"/>	Website cho thiết bị di động	Bookmark my web app on Facebook mobile.

Bạn chỉ cần nhập vào địa chỉ đầy đủ của trang web mà bạn muốn sử dụng và kết nối đến ứng dụng này. Ở đây, trên **localhost** của tôi, tôi tạo ra một website tên **FB_API** và sử dụng để kết nối đến ứng dụng này trong bước 3, nên tôi khai báo là `http://localhost/FB_API`, nếu bạn sử dụng website có tên miền `congtyabc.com` thì đơn giản chỉ cần khai báo `http://congtyabc.com`.

Nhắc nhở: khi bạn lỡ tay click lên một chức năng mà bạn không sử dụng, nếu bạn bỏ trống thông tin mà không nhập gì thì bạn sẽ không lưu thông tin lại được. Lúc này, đơn giản là bạn chỉ cần click lên dấu X ở phía bên phải của chức năng (ở vị trí mà tôi khoanh đỏ bên phải) để ẩn chức năng đó.

LƯU Ý QUAN TRỌNG

Địa chỉ website cần phải có tên miền giống với tên miền bạn khai báo ở phần thông tin cơ bản. Nếu không thì bạn sẽ không kết nối được.

A screenshot of the Facebook App configuration page. The 'App Domains' field is set to 'localhost' and is highlighted with a red box. A red arrow points from this field to the 'Site URL' field, which is also highlighted with a red box and contains the text 'http://localhost/FB_API'. The 'Contact Email' field is 'dangvanle@gmail.com'. The 'Hosting URL' field has a message: 'You have not generated a URL through one of our partners (Get one)'. The 'Sandbox Mode' is set to 'Mở' (Open). Below the configuration fields, there is a section titled 'Chọn cách ứng dụng của bạn tích hợp với Facebook' (Choose how your app integrates with Facebook). Under this section, there is a green checkmark icon and the text 'Trang web có Đăng nhập qua Facebook' (Website has Login with Facebook). The 'Site URL' field is also highlighted with a red box.

Đến đây, chúng ta đã xong việc tạo ứng dụng đại diện và cài đặt những thông tin cần thiết. Giờ ta sẽ đến phần thứ 3, phần tạo trang HTML để kết nối thử nghiệm.

2.3. Kết nối đến ứng dụng và tạo trang HTML để thử nghiệm | Bước 3

Trước khi bắt đầu, bạn cần phải hiểu cơ chế làm việc của quá trình kết nối. Mọi thao tác gọi đến Graph API chỉ đơn giản là một truy vấn HTTP từ bất cứ đâu, yêu cầu phải kèm theo Access Token. Vì vậy, khái niệm kết nối hiểu đơn giản là tạo ra Access Token và sử dụng nó khi gọi API.

Có 2 hình thức chủ yếu để ta có được Access Token là sử dụng "Javascript SDK" hoặc là "Truy vấn HTTP trực tiếp". Chúng ta sẽ lần lượt đi qua từng cách một, và việc của bạn là chọn ra một cách để sử dụng cho thích. Xin lưu ý, ở lần kết

nổi đầu tiên thì người dùng sẽ được yêu cầu cấp phép cho những quyền mà ứng dụng mong muốn, những lần sau thì ứng dụng sẽ ghi nhớ người dùng đó.

LẤY ACCESS TOKEN BẰNG TRUY VẤN HTTP TRỰC TIẾP

Theo cách này, ta phải gọi một truy vấn HTTP đến cổng giao tiếp theo một cú pháp định trước. Người sử dụng sẽ được yêu cầu đồng ý hoặc không đồng ý với các quyền mà bạn yêu cầu. Sau đó trình duyệt sẽ tự động chuyển tiếp đến một địa chỉ URL do bạn khai báo và gửi kèm theo Access Token đã tạo. Cụ thể từng chi tiết như sau:

Nội dung 1: cú pháp của một truy vấn HTTP để tạo Access Token.

Cú pháp	
1	<code>https://graph.facebook.com/oauth/authorize?type=user_agent</code>
2	<code>&client_id=YourAppID</code>
3	<code>&redirect_uri=YourSiteURL</code>
4	<code>&scope=PermissionList</code>

Trong đó,

YourAppID	<p>Là AppID của ứng dụng của bạn. Xem ở phần tiêu đề ứng dụng.</p> 
YourSiteURL	<p>Là địa chỉ của trang web sẽ nhận kết quả trả về. Trang web này phải thuộc website mà bạn đã khai báo với ứng dụng. Ở đây tôi khai báo là:</p> <p><code>http://localhost/FB_API/test.html</code></p>
PermissionList	<p>Danh sách quyền hạn mà bạn muốn người sử dụng cho phép. Mỗi quyền hạn cách nhau bởi một dấu phẩy “,”</p> <p>→ Bạn sẽ được học chi tiết hơn về quyền cũng như về khóa Access Token sau ví dụ minh họa này. Ở đây tôi sẽ sử dụng 2 quyền: read_stream, Quyền này sẽ cho tôi được phép đọc các bài viết trên tường nhà.</p>

Khi bạn thực hiện truy vấn này, người dùng đang đăng nhập Facebook sẽ được yêu cầu đồng ý các quyền mà bạn đã liệt kê. Nếu người dùng chưa đăng nhập Facebook thì họ sẽ được yêu cầu đăng nhập trước khi cho quyền. Sau khi các quyền đã được chấp nhận, trình duyệt sẽ chuyển hướng về trang web đã được khai báo và kèm theo Access Token dưới dạng một điểm neo trên địa chỉ:

`http://localhost/FB_API/test.html#access_token=...`

Nội dung 2: sau khi chuyển hướng về trang xử lý Access Token, việc của bạn giờ là viết code để đọc Access Token và lưu trữ hoặc làm gì đó phục vụ cho mục đích của bạn ở trang **test.html**. Để hiểu rõ hơn, vui lòng xem ví dụ sau đây:

A. Chuẩn bị URL truy vấn HTTP để lấy Access Token:

URL truy vấn sẽ dùng

```
1 https://graph.facebook.com/oauth/authorize?type=user_agent
2 &client_id=304746743004164
3 &redirect_uri=http://localhost/FB_API/test.html
4 &scope=read_stream
```

B. Tạo trang test.html với nội dung chỉ có 1 đoạn javascript để đọc và in dữ liệu sau dấu # ở trên địa chỉ ra màn hình.

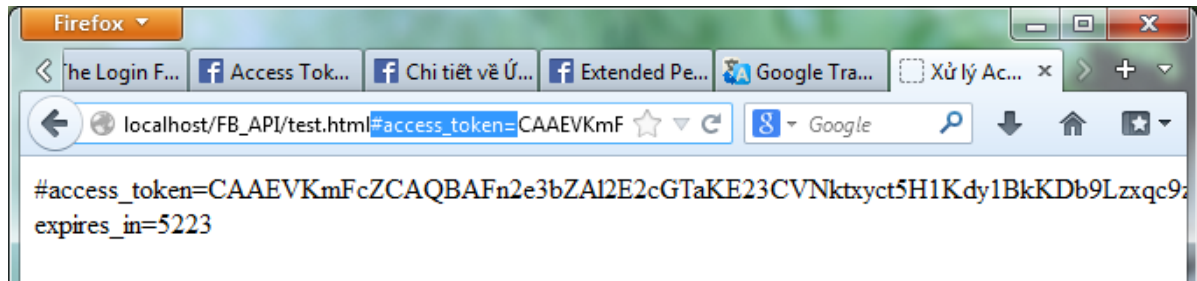
test.html

```
1 <html><head><title>Xử lý Access Token</title></head>
2 <body>
3 <script type="text/javascript">
4     if(window.location.hash!=undefined) {
5         document.write(window.location.hash);
6     }
7 </script></body></html>
```

C. Bắt đầu mở trình duyệt lên, truy vấn HTTP theo URL tạo ở bước 1:



Đây là màn hình mà Facebook yêu cầu người dùng cấp phép các quyền do ứng dụng yêu cầu. Ở đây, người dùng sẽ bấm đồng ý để duyệt từng quyền. Ngay sau đó, trình duyệt sẽ bị chuyển hướng về http://localhost/FB_API/test.html kèm theo Access Token sau dấu # ở Url như hình dưới:



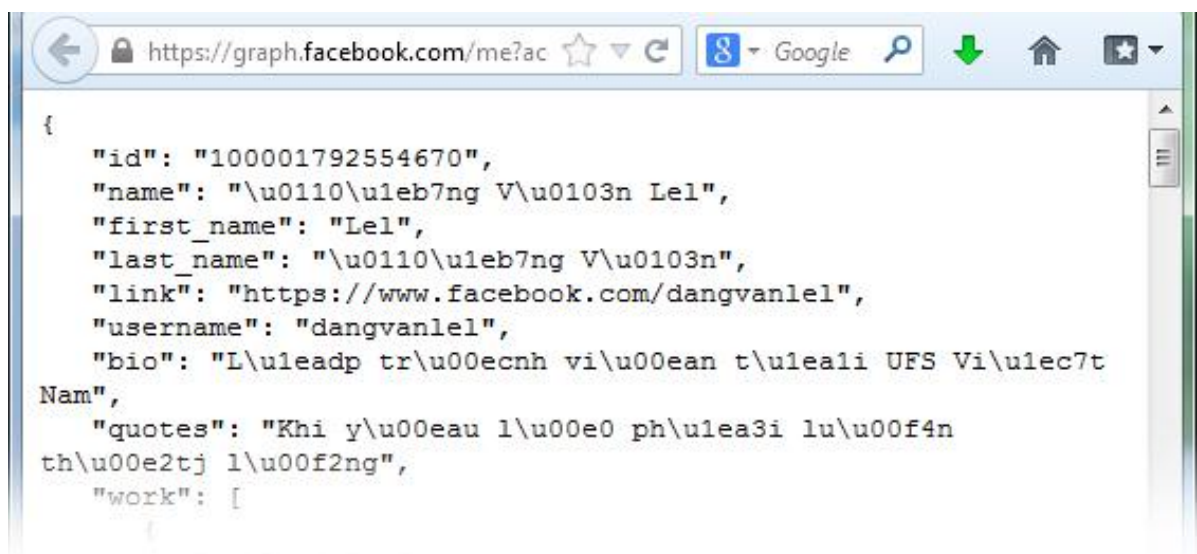
Và đoạn JAVASCRIPT chỉ việc đọc và in giá trị từ sau dấu # ra màn hình. Giá trị này có 2 thành phần:

access_token	Chứa chuỗi Access Token ta cần
expires_in	Thời gian sống của Access Token, tính bằng giây

D. Giờ đã có Access Token, ta sẽ thử truy cập dữ liệu cá nhân của người dùng. Tạo URL truy vấn HTTP đến cổng giao tiếp như sau:

URL truy vấn	
1	https://graph.facebook.com/me?access_token=ACT

Với **ACT** là chuỗi Access Token ta vừa có. Dán URL này vào trình duyệt, ta sẽ có kết quả trả về tương tự thể này, tùy theo tài khoản: (JSON)



Nếu bạn nhìn thấy được thông tin tương tự như trên, nghĩa là Access Token đã sử dụng được, việc lưu trữ và sử dụng khi gọi API là tùy vào mục đích của bạn.

LẤY ACCESS TOKEN BẰNG CÁCH SỬ DỤNG JAVASCRIPT SDK

JAVASCRIPT SDK LÀ GÌ?

Javascript SDK là một tập hợp của rất nhiều những chức năng API đóng gói sẵn dưới dạng các hàm chạy trên client (JAVASCRIPT). Giúp bạn có thể nhanh chóng và tiện lợi trong việc sử dụng Plugin, gọi các APIs và tạo kết nối giữa người dùng và website. Sự kết nối này còn được gọi là “Đăng nhập bằng Facebook”.

SỬ DỤNG

Để sử dụng, ta chỉ việc chèn thêm nội dung vào sau thẻ <body> như sử dụng facebook plugin vậy, nhưng với JSDK thì chèn đầy đủ hơn.

Chèn JAVASCRIPT SDK | Khai báo

```
1 <html>
2 <head></head>
3 <body>
4 <div id="fb-root"></div>
5 <script>
6 window.fbAsyncInit = function() {
7     /* khai báo sử dụng JSDK */
8     FB.init({
9         appId : 'AppID',
10        channelUrl : 'channel.html',
11        status : true,
12        cookie : true,
13        xfbml : true
14    });
15    /* các sự kiện về kết nối viết ở đây */
16 };
17 /* Khai báo đồng bộ hóa SDK – Cái này như sử dụng plugin (Xem lại tập 1) */
18 (function(d){ var js, id = 'facebook-jssdk', ref = d.getElementsByTagName('script')[0];
19     if (d.getElementById(id)) {return;} js = d.createElement('script'); js.id = id; js.async = true;
20     js.src = "//connect.facebook.net/vi_VN/all.js"; ref.parentNode.insertBefore(js, ref);
21 })(document);
22 </script>
23 ...
```

Trong đó,

appid	Là AppID của ứng dụng của website trên facebook (chuỗi)
channelUrl	Địa chỉ tập tin chanel, thường thì tập tin này chỉ có một dòng dữ liệu là: <code><script src="//connect.facebook.net/en_US/all.js"></script></code> Dòng dữ liệu này thay đổi ngôn ngữ sử dụng của plugin và các apis
Status	Có kiểm tra trạng thái kết nối không? Luôn để có nhé.
cookie	Cho phép sử dụng cookie hay không?
xfbml	TRUE nhé. Nó có nghĩa là parse XFBML hay không.

Tiếp theo, ta khai báo sự kiện nhận biết sự thay đổi của kết nối.

Chèn JAVASCRIPT SDK | Sự kiện

```

1  <html>
2  <head></head>
3  <body>
4  <div id="fb-root"></div>
5  <script>
6  window.fbAsyncInit = function() {
7      /* khai báo sử dụng JSDK */
8      FB.init({
9          appId : 'AppID',
10         channelUrl : 'channel.html',
11         status : true,
12         cookie : true,
13         xfbml : true
14     });
15     /* các sự kiện về thay đổi kết nối viết ở đây */
16     FB.Event.subscribe('auth.authResponseChange', function(rs) {
17         if (rs.status === 'connected') {
18             // đã kết nối
19         } else if (rs.status === 'not_authorized') {
20             // chưa kết nối
21         } else {
22             // chưa đăng nhập facebook
23         }
24     });
25 };
26 /* Khai báo đồng bộ hóa SDK – Cái này như sử dụng plugin (Xem lại tập 1) */
27 (function(d){ var js, id = 'facebook-jssdk', ref = d.getElementsByTagName('script')[0];
28 if (d.getElementById(id)) {return;} js = d.createElement('script'); js.id = id; js.async = true;
29 js.src = "//connect.facebook.net/vi_VN/all.js"; ref.parentNode.insertBefore(js, ref);
30 }(document));
31 </script>
32 ...

```

Giải thích: sự kiện **'auth.authResponseChange'** sẽ được kích hoạt mỗi khi có sự thay đổi trong quá trình kết nối giữa người dùng và ứng dụng. Có 3 trạng thái:

connected	Người dùng đã đăng nhập facebook, và cũng đã kết nối với ứng dụng.
not_authorized	Người dùng đã đăng nhập facebook nhưng chưa kết nối đến ứng dụng.
not_logged	Trường hợp còn lại là người dùng chưa đăng nhập facebook.

Bạn phải lưu ý rằng sự kiện này chỉ được kích hoạt khi **có sự thay đổi của kết nối**. Nghĩa là phải có sự chuyển từ trạng thái này sang trạng thái kia. **"FB"** là đối tượng đại diện cho JAVASCRIPT SDK mà từ đó ta sẽ gọi đến những hàm định nghĩa sẵn để tiến hành công việc mà mình mong muốn. Dưới đây là khai báo các hàm được gọi trong trường hợp có sự thay đổi kết nối.

Chèn JAVASCRIPT SDK | Sự kiện | Xử lý kết nối

```

1  <html>
2  <head></head>
3  <body>
4  <div id="fb-root"></div>
5  <script>
6  window.fbAsyncInit = function() {
7      /* khai báo sử dụng JSDK */
8      FB.init({
9          appId : 'AppID',
10         channelUrl : 'channel.html',
11         status : true, cookie : true, xfbml : true
12     });
13     /* các sự kiện về kết nối viết ở đây */
14     FB.Event.subscribe('auth.authResponseChange', function(rs) {
15         if (rs.status === 'connected') {
16             Display_Name(rs.authResponse.accessToken);
17         } else if (rs.status === 'not_authorized') {
18             FB.login();
19         } else {
20             FB.login();
21         }
22     });
23 };
24 /* Khai báo đồng bộ hóa SDK – Cái này như sử dụng plugin (Xem lại tập 1) */
25 (function(d){ var js, id = 'facebook-jssdk', ref = d.getElementsByTagName('script')[0];
26 if (d.getElementById(id)) {return;} js = d.createElement('script'); js.id = id; js.async = true;
27 js.src = "//connect.facebook.net/vi_VN/all.js"; ref.parentNode.insertBefore(js, ref);
28 }(document));
29 /* phần viết hàm xử lý | hiển thị khóa, tên người kết nối và ẩn liên kết kết nối */
30 </script>
31 ...

```

Giải thích:

→ Khi trạng thái kết nối từ một trạng thái khác đổi thành "connected" sự kiện này sẽ gọi hàm **Display_Name(acckey)** , hàm này do chúng ta tự định nghĩa.

→ Khi trạng thái kết nối từ một trạng thái khác đổi thành "not_authorized" hoặc thành "not_logged" thì sự kiện này sẽ gọi **FB.Login()** để thực hiện kết nối, hàm này thuộc JSDK.

Tiếp theo, ta sẽ khai báo hàm Display_Name() để lấy tên của người vừa kết nối và hiển thị lên câu xin chào, đồng thời ẩn LINK kết nối.

Chèn JAVASCRIPT SDK | Sự kiện | Xử lý kết nối

```

1  <html><head></head><body> <div id="fb-root"></div>
2  <script>
3    window.fbAsyncInit = function() {
4      /* khai báo sử dụng JSDK */
5      FB.init({
6        appId : 'AppID', channelUrl : 'channel.html',
7        status : true, cookie : true, xfbml : true
8      });
9      /* các sự kiện về kết nối viết ở đây */
10     FB.Event.subscribe('auth.authResponseChange', function(rs) {
11       if (rs.status === 'connected') {
12         Display_Name(rs.authResponse.accessToken);
13       } else if (rs.status === 'not_authorized') {
14         FB.login();
15       } else {
16         FB.login();
17       }
18     });
19   };
20   /* Khai báo đồng bộ hóa SDK – Cái này như sử dụng plugin (Xem lại tập 1) */
21   (function(d){ var js, id = 'facebook-jssdk', ref = d.getElementsByTagName('script')[0];
22   if (d.getElementById(id)) {return;} js = d.createElement('script'); js.id = id; js.async = true;
23   js.src = "//connect.facebook.net/vi_VN/all.js"; ref.parentNode.insertBefore(js, ref);
24   }(document))
25   /* phần viết hàm xử lý – hiển thị khóa, tên người kết nối và ẩn liên kết kết nối */
26   function Display_Name(acckey){
27     $('#LoginLink').css('display', 'none');
28     $('#Key').html('AccessToken của bạn: ' + acckey);
29     FB.api('/me', function(ketqua){
30       $('#WelcomeText').html('Xin chào <b>'+ketqua.name+'</b>');
31     });
32   }
33 </script>
34 ...

```

Tiếp theo, là phần nội dung. Ta sẽ thiết kế nội dung chỉ đơn thuần như sau:

Chèn JAVASCRIPT SDK | Sự kiện | Xử lý kết nối | Nội dung

```

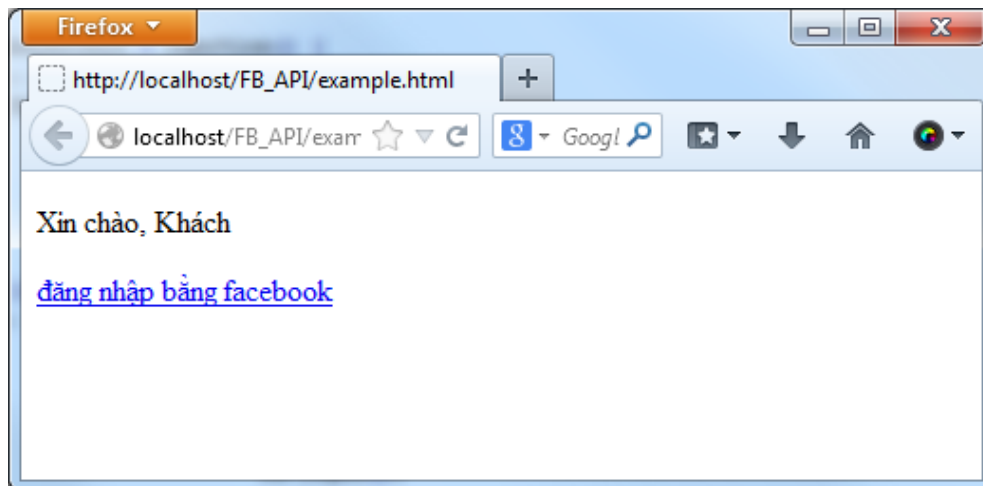
1  <html>
2  <head></head>
3  <body>
4  <div id="fb-root"></div>
5  <script>
6  window.fbAsyncInit = function() {
7      /* khai báo sử dụng JDK */
8      FB.init({
9          appId : 'AppID',
10         channelUrl : 'channel.html',
11         status : true,
12         cookie : true,
13         xfbml : true
14     });
15     /* các sự kiện về kết nối viết ở đây */
16     FB.Event.subscribe('auth.authResponseChange', function(rs) {
17         if (rs.status === 'connected') {
18             Display_Name(rs.authResponse.accessToken);
19         } else if (rs.status === 'not_authorized') {
20             FB.login();
21         } else {
22             FB.login();
23         }
24     });
25 };
26 /* Khai báo đồng bộ hóa SDK – Cái này như sử dụng plugin (Xem lại tập 1) */
27 (function(d){ var js, id = 'facebook-jssdk', ref = d.getElementsByTagName('script')[0];
28 if (d.getElementById(id)) {return;} js = d.createElement('script'); js.id = id; js.async = true;
29 js.src = "//connect.facebook.net/vi_VN/all.js"; ref.parentNode.insertBefore(js, ref);
30 }(document))
31 /* phần viết hàm xử lý – hiển thị khóa, tên người kết nối và ẩn liên kết kết nối */
32 function Display_Name(){
33     $('#LoginLink').css('display', 'none');
34     $('#Key').html('AccessToken của bạn: ' + acckey);
35     FB.api('/me', function(ketqua){
36         $('#WelcomeText').html('Xin chào <b>'+ketqua.name+'</b>');
37     });
38 }
39 </script>
40 <!-- Nội dung HTML | bao gồm cả chèn jQuery để sử dụng cho ví dụ -->
41 <script src="http://code.jquery.com/jquery-1.9.1.js"></script>
42 <p id='WelcomeText'> Xin chào, Khách </p><p id='Key'> </p>
43 <a id='LoginLink' href='javascript:FB.Login()'>đăng nhập bằng facebook</a>
44 </body>

```

```
</html>
```

Kịch bản chạy thử ví dụ:

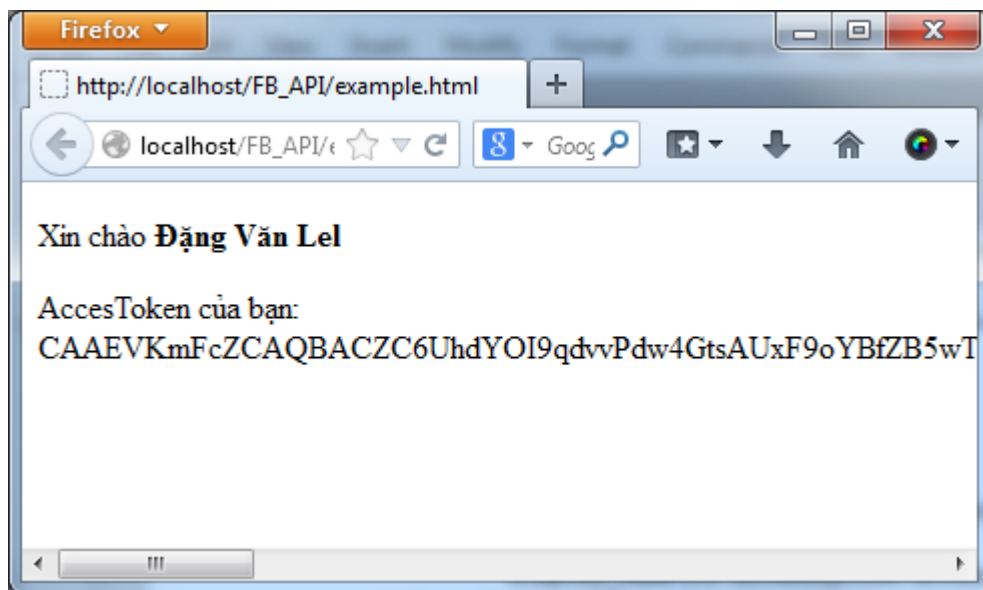
- Khi truy cập http://localhost/FB_API/example.html ta có nội dung:



- Khi click vào liên kết “đăng nhập bằng facebook”, popup kết nối đến facebook hiện ra:



- Người dùng “Đồng ý” các quyền mà ứng dụng yêu cầu. Sau khi xử lý xong, popup sẽ bị tắt và nội dung của example sẽ bị đổi thành:



- Trạng thái kết nối này được duy trì mãi cho đến khi người dùng chủ động vào setting của họ và xóa ứng dụng ra khỏi danh sách sử dụng của họ. Có nghĩa là sau này bạn mở example lên thì nó sẽ tự động kiểm tra kết nối và sẽ hiện thị ngay nội dung kết quả trên.

Bạn có thể viết hàm xử lý riêng với mục đích riêng, có thể lưu trữ Access Token nếu cần. Nhưng trong trường hợp của JSDK, khi dùng JSDK gọi API thì bạn không cần cung cấp Access Token vì nó đã có trong kết nối.

Hàm `FB.api('/me', function(ketqua){});` ở ví dụ trên là một lời gọi đến Graph APIs, nó tương ứng với câu truy vấn `https://graph.facebook.com/me?acc...` và `ketqua` chính là biến JSON chứa kết quả của lời gọi. Bạn có thể truy cập các thành phần của kết quả theo kiểu hướng đối tượng, tức là dùng dấu chấm (.) đi kèm với tên thuộc tính cần lấy, ví dụ như `ketqua.name`. Nếu bạn gọi một thuộc tính không tồn tại thì nó trả về `undefined`.

2.4. Ứng dụng thực tế

Tổng kết lại, ta thấy rằng, để làm việc với Open Graph ta có 2 cách : gửi truy vấn HTTP lên trực tiếp thông qua cổng <https://graph.facebook.com> hoặc dùng Javascript SDK. Vậy ta nên sử dụng cách nào cho tiện nhất. Điều này ăn thua vào mục đích sử dụng của bạn.

Nếu bạn muốn tạo một Access Token để lưu trữ và sử dụng trong các lời gọi POST bài tự động thì truy vấn trực tiếp là lựa chọn hoàn hảo. Còn nếu bạn chỉ cần sử dụng Plugin và hiển thị vài thông tin của người dùng hoặc là chức năng đăng nhập bằng facebook thì JSDK là thứ bạn nên dùng.

Chúng ta đã biết cách Open Graph hoạt động, chúng ta cũng biết cách lấy một khóa truy cập. Giờ ta sẽ đi học về cấu trúc dữ liệu khi gửi lên Open Graph.

3. Đọc dữ liệu từ Open Graph

Open Graph cho người dùng 2 cách thức truy vấn là đọc dữ liệu và gửi dữ liệu, phần này sẽ trình bày với các bạn về đọc dữ liệu. Nhắc lại giới thiệu trước đó, cấu trúc của một lời gọi đến cổng Graph có dạng như sau:

Cú pháp

1 `https://graph.facebook.com/<action-data>&access_token=???`

Cách lấy access_token thì bạn đã biết. Giờ chúng ta sẽ đi học về cách khai báo action-data. Cấu pháp của action-data tương đối nhiều và phức tạp, độ tùy chỉnh cao, nên facebook đã cung cấp sẵn một công cụ hữu ích cho lập trình viên có thể dễ dàng tạo ra chuỗi dữ liệu đúng theo ý muốn một cách nhanh nhất và đảm bảo độ chính xác cao nhất, công cụ này được gọi là Graph API Explorer.

3.1. Graph API Explorer là gì?

Đây là công cụ giúp cho bạn dễ dàng khám phá và hiểu APIs nhất. Bạn sẽ làm quen nhanh chóng với chuỗi action-data. Bạn có thể thử nghiệm việc đọc, thêm dữ liệu bằng các chuỗi action-data do bạn tự tạo thông qua hỗ trợ của công cụ này. Bằng cách đó, bạn sẽ nhanh chóng nắm bắt và kiểm soát được cấu trúc của những chuỗi action-data.

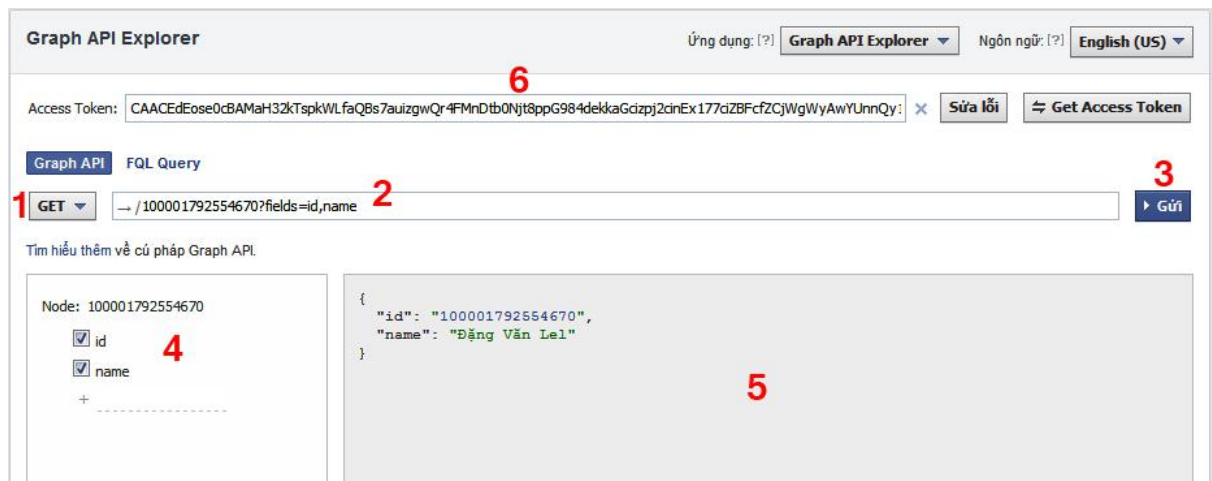
3.2. Truy cập và giao diện chính

Công cụ này bạn được truy cập thông qua Url:

Graph API Explorer

1 `https://developers.facebook.com/tools/explorer`

Giao diện chính của công cụ này khi mới truy cập sẽ bao gồm những thành phần như sau:



- ▣ Phần **1**: để chọn hình thức sử dụng, có 2 hình thức GET, POST (POST chia làm POST và DELETE). Get tương ứng với lấy dữ liệu, Post tương ứng với gửi dữ liệu (viết) và Delete tương ứng với xóa dữ liệu.
- ▣ Phần **2**: là chuỗi action-data sẽ dùng, chuỗi này có thể tự đánh vào hoặc dùng tool ở phần 4 tạo ra.
- ▣ Phần **3**: là nút bấm để thực hiện truy vấn và để xem kết quả của action-data đã tạo. kết quả được hiển thị ở phần 5.
- ▣ Phần **4**: là công cụ giúp bạn tạo action-data bằng cách click chọn. (sẽ hướng dẫn cụ thể sau).
- ▣ Phần **5**: là khung trình bày kết quả của truy vấn.

3.3. Sử dụng Graph API Explorer

Bạn đã được giới thiệu về API Explorer, giờ tôi sẽ hướng dẫn bạn sử dụng công cụ này để tạo ra chuỗi action-data cho mục đích sử dụng của mình.

Đầu tiên bạn cần hiểu rằng, chuỗi action-data sẽ bao gồm 2 thành phần chính: "đối tượng bị tác động" và "hành động".

- ▣ Đối tượng bị tác động: là thành phần xuất hiện đầu tiên trong chuỗi action-data, đối tượng có thể là con người, bài viết, hình ảnh, ... và được đại diện bởi id hoặc username.
- ▣ Hành động: là ám chỉ dữ liệu để mô tả mình sẽ làm gì với đối tượng đó. Như là mình muốn lấy id, tên, đường dẫn đến đối tượng, ...

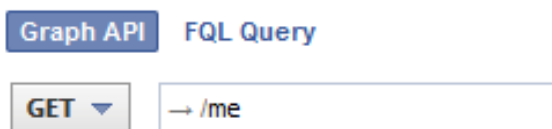
Bây giờ ta sẽ đi thực hành lấy dữ liệu của chính mình, dữ liệu ta sẽ lấy sẽ bao gồm những thông tin như sau:

Tập 2	Dành cho người mới bắt đầu	Đặng Văn Lel
-------	----------------------------	--------------

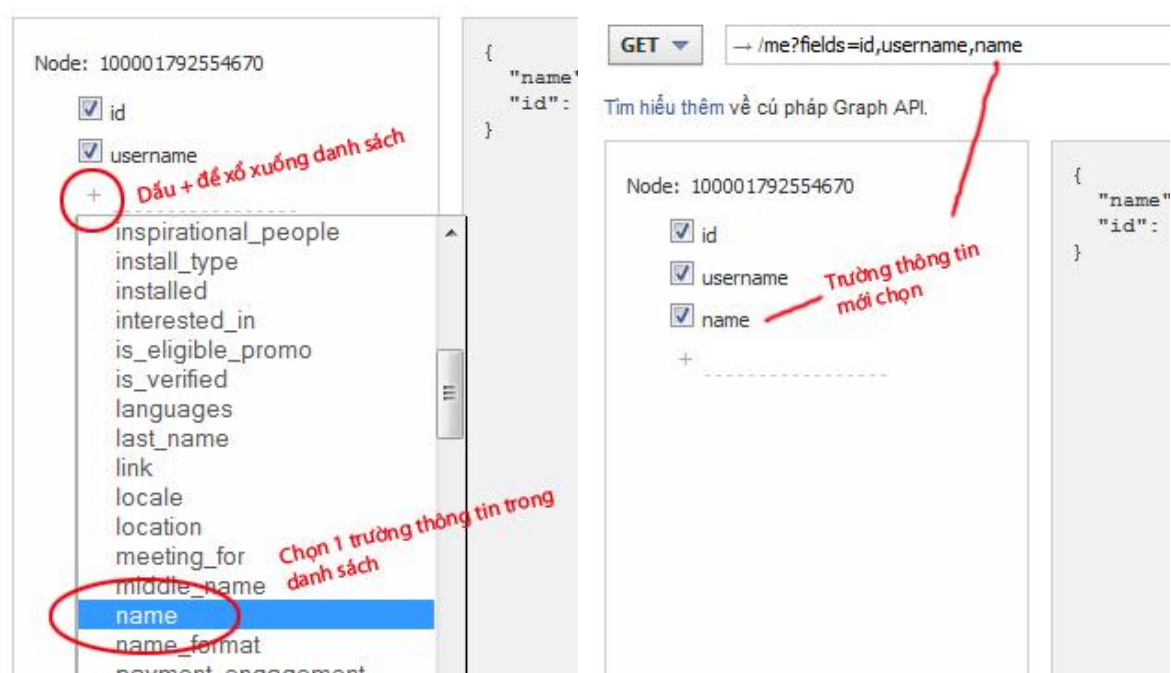
- ▣ Mã số nhận dạng (id)
- ▣ Tên định danh nhận dạng (username)
- ▣ Tên hiển thị đầy đủ (name)
- ▣ Giới tính (gender)
- ▣ Địa chỉ trang cá nhân (link)
- ▣ Hình đại diện loại vuông (picture – type: square)

3 BƯỚC TIỀN HÀNH

Bước 1: tại vị trí (2) bạn xóa trắng phần action-data do công cụ tự sinh ra sẵn và thay bằng "me". Vì mình muốn truy cập dữ liệu của chính mình (người đang kết nối) nên có thể dùng "me" để thay thế id và username.



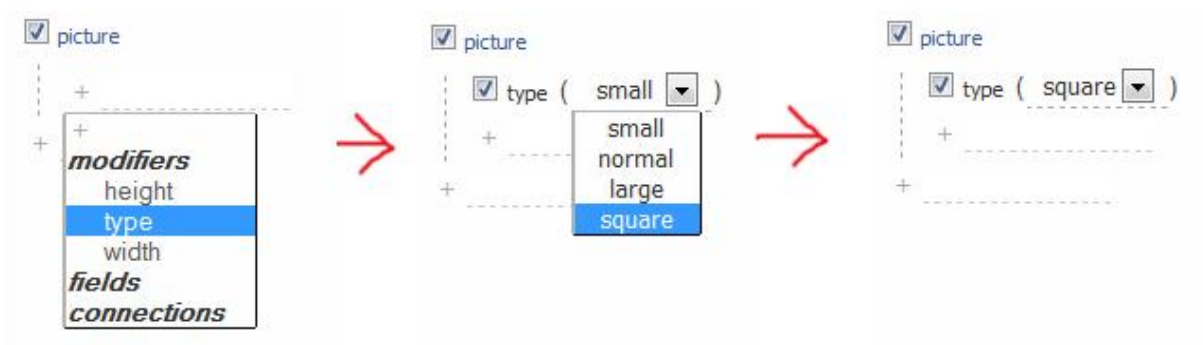
Bước 2: quan sát tại vị trí (4) và thấy nó có 2 trường dữ liệu mà đã được chọn trước là **id** và **username**. Để xóa 1 trường thông tin ta chỉ việc click lên tên của trường thông tin đó. Còn để thêm 1 trường thông tin thì ta click vào dấu **+** ở bên dưới trường thông tin cuối cùng, và sẽ được hiện ra 1 danh sách các trường thông tin để chọn.



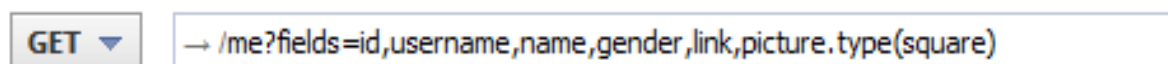
Cứ tiếp tục chọn cho tới khi đầy đủ các trường thông tin ta cần. Đặc biệt ngoại lệ có trường thông tin sẽ có những thông tin nhỏ hơn (được gọi là các tùy chọn) như trường thông tin picture. Khi ta chọn tới trường picture thì kết quả ta tạm được thời này.



Theo yêu cầu thì ta phải chọn đúng picture – type: square, ta click vào dấu + thêm thuộc tính của picture và chọn “type”, sau đó chọn giá trị “square” cho type, như hình sau.



Đến đây đã đủ các thông tin ta cần và nếu bạn nhìn lại khung số (2), bạn sẽ thấy ta có giá trị chuỗi action-data trông như sau:



Bước 3: Để xem thử kết quả trả về khi truy vấn chuỗi action-data đó, ta click lên nút “Gửi” ở vị trí số (2). Và kết quả (theo định dạng JSON) sẽ được hiển thị ở khung số (5) như hình sau:

GET

→ /me?fields=id,username,name,gender,link,picture.type(square)

Gửi

Tìm hiểu thêm về cú pháp Graph API

Node: 100001792554670

☒ id

☒ username

☒ name

☒ gender

☒ link

☒ picture

☒ type (square)

+

+

```
{
  "id": "100001792554670",
  "gender": "male",
  "link": "https://www.facebook.com/dangvanlel",
  "name": "Đặng Văn Lel",
  "username": "dangvanlel",
  "picture": {
    "data": {
      "url": "https://fbcdn-profile-a.akamaihd.net/hprofile-ak-ash2/1086846_100001792554670_
      "is_silhouette": false
    }
  }
}
```

Như vậy, chỉ sau vài thao tác đơn giản, bạn đã có thể tạo ra được chuỗi action-data, bạn chỉ còn việc copy nó và sử dụng vào mục đích của mình trong ứng dụng mà bạn đang thiết kế.

3.4. Tổng kết

Đến lúc này bạn đã có thể hoàn toàn nắm được cơ bản cấu trúc để thao tác với Graph APIs. Ta có thể tổng hợp lại kiến thức theo tóm tắt như sau:

- ✓ Graph APIs được cung cấp thông qua cổng giao tiếp:
<https://graph.facebook.com>
- ✓ Dữ liệu gửi lên cổng Graph có 2 phần: Action Data và Access Token Key.
- ✓ Action Data bao gồm 2 phần chính là: đối tượng bị tác động và hành động. Chuỗi này có thể được tự tạo (nếu bạn đã quen) hoặc nhờ trợ giúp của công cụ Graph API Explorer được cung cấp tại:
<https://developers.facebook.com/tools/explorer>
- ✓ Access Token Key có thể lấy dựa vào 2 hình thức: truy vấn trực tiếp và sử dụng javascript sdk.
- ✓ Khi có đủ action-data và access token key, ta chỉ việc thực hiện 1 truy vấn http đến cổng graph để lấy dữ liệu ta cần theo cấu trúc:

Cú pháp

```
1 <cong graph>/<action-data>& access_token=<access-token-key>
```

Trong đó, phần màu đỏ là dữ liệu cố định.

Dưới đây là chuỗi truy vấn HTTP đầy đủ gửi lên cổng Graph theo ví dụ bên trên:

Ví dụ

```
https://graph.facebook.com/me?fields=id,username,name,gender,link,picture.type(square)&
access_token=CAACEdEose0cBAM46dK4MgCmVLyELU7EDjOCfYaY605yMi2QQZCs76DmD0S
mqXyGUzMWQSGqSpZAGUL0vwsZC9M0TjFs6szoPb4zl4YvHLyEFIsIMqvXj7mvaZCZCBLxIZBWA
7OaZAue7iZCAiyZCINDQZCmpeM4ahgv5ADH68OZCzMYcqlfouqqRh5XPfAAPQWmi1NcPtlov
MQuHwZDZD
```

--- Hết Tập 2 ---

Tập tiếp theo sẽ nói về

VIẾT BÀI LÊN FACEBOOK QUA GRAPH APIs

<http://facebook.com/thuvienlaptrinh>

Cảm ơn các bạn đã ủng hộ