

# Utveckling av Facebook applikation

Samuel Backman

Examensarbete  
Informations- och medieteknik  
2013

Samuel Backman

EXAMENSARBETE	
Arcada	
Utbildningsprogram:	Informations- och medieteknik
Identifikationsnummer:	4172
Författare:	Samuel Backman
Arbetets namn:	Utveckling av Facebook applikation
Handledare (Arcada):	Johnny Biström
Uppdragsgivare:	-
<p>Sammandrag:</p> <p>Arbetet är en översikt över Facebooks plattform och hur man kan använda plattformen för att utveckla applikationer. Forskningsmetoden som använts är litteraturstudier inom utvecklande av Facebook applikationer och Facebooks plattform. Syftet med detta projekt är att studera hur Facebooks plattform fungerar och hur man kan använda den för att utveckla applikationer. Denna avhandling behandlar Facebook applikationsutveckling och Facebook-plattformens funktioner. De viktigaste delarna av plattformen och utvecklingsprocessen är täckt med information om allmänna problem och plattformspecifika metoder. Denna avhandling fokuserar bara på Facebook-plattformens applikationer, inte externa webbplatser eller mobila plattformar.</p>	
Nyckelord:	Facebook utveckling, Facebook applikationer, Facebook API, Facebook SDK, Facebooks plattform
Sidantal:	72
Språk:	Svenska
Datum för godkännande:	24.4.2013

DEGREE THESIS	
Arcada	
Degree Programme:	Information and Media Technology
Identification number:	4172
Author:	Samuel Backman
Title:	Facebook application development
Supervisor (Arcada):	Johnny Biström
Commissioned by:	-
<p>Abstract:</p> <p>This study is an overview of the Facebook platform and how to use the platform to develop applications. The research method used is a study of literature in the development of Facebook applications and Facebook's platform. The purpose of this project is to study how Facebook's platform works and how to use it to develop applications. This thesis deals with Facebook application development and Facebook-platform capabilities. The main elements of the platform and the development process are covered with information about general problems and platform specific methods. This thesis focuses only on the Facebook platform applications, not external websites or mobile platforms.</p>	
Keywords:	Facebook Development, Facebook API, Facebook SDK, Facebook Platform, Facebook Applications
Number of pages:	72
Language:	Swedish
Date of acceptance:	24.4.2013

# INNEHÅLL

<b>1</b>	<b>Inledning .....</b>	<b>5</b>
1.1	Bakgrund .....	5
1.2	Syfte och mål.....	5
1.3	Metoder .....	6
1.4	Terminologi.....	6
1.5	Avgränsning .....	7
<b>2</b>	<b>Facebook.....</b>	<b>8</b>
<b>3</b>	<b>Plattformen .....</b>	<b>10</b>
3.1	Komponenter.....	10
3.1.1	<i>Login</i> .....	10
3.1.2	<i>Facebook API</i> .....	11
3.1.3	<i>Sociala Plugins</i> .....	12
3.1.4	<i>Graph API</i> .....	13
3.1.5	<i>FQL</i> .....	14
3.1.6	<i>Open Graph</i> .....	15
3.2	Rättigheter.....	16
3.2.1	<i>Publicerings rättigheter</i> .....	17
3.2.2	<i>Begära om rättigheter</i> .....	18
3.3	Sekretess .....	18
<b>4</b>	<b>Facebook Applikationer.....</b>	<b>19</b>
4.1	Uppbyggnad .....	19
4.1.1	<i>Canvas</i> .....	19
4.1.2	<i>Profil</i> .....	19
4.1.3	<i>Notifikationer</i> .....	20
4.1.4	<i>Alarm</i> .....	20
4.1.5	<i>Inbjudningar</i> .....	20
4.1.6	<i>Nyhetsflöde</i> .....	20
4.2	Typ av applikation .....	21
4.2.1	<i>Canvas-applikation</i> .....	21
4.2.2	<i>Tab-applikation</i> .....	22
4.3	Förvaring av data .....	23
4.4	App center .....	23
<b>5</b>	<b>Utveckling .....</b>	<b>25</b>
5.1	Förberedning .....	25

5.2	Verktyg .....	25
5.2.1	<i>JavaScript SDK Konsol</i> .....	25
5.2.2	<i>Graph API Explorer konsol</i> .....	26
5.2.3	<i>Open Graph Debugger</i> .....	27
5.2.4	<i>Statistik</i> .....	27
5.3	Konfiguration av applikationen .....	27
5.3.1	<i>Obligatoriska inställningar</i> .....	28
5.4	Publicering .....	28
5.5	Exempelapplikation .....	28
<b>Källor</b> .....		<b>30</b>
<b>BILAGOR</b> .....		<b>34</b>
Bilaga 1. Källkoden för exempel applikationen .....		34

## Figurer

Figur 1. Facebooks marknadsförings fält. ....	9
Figur 2. Facebook inloggning med JavaScript [12]. ....	11
Figur 3. Facebook Open Graph berättelse [14]. ....	16
Figur 4. Facebook applikation ber om rättigheter[13].....	17
Figur 5. Facebook canvas applikation [15]. ....	21
Figur 6. Facebook sidofält [15.] ....	22
Figur 7. Facebook tab applikation [29]. ....	23
Figur 8. Graph API Explorer konsolen [18]. ....	26
Figur 9. Exempel applikation .....	29

## Tabeller

Tabell 1. Facebook Graph API stödda objekt [26.].....	13
---	----

# 1 INLEDNING

Sociala nätverk har blivit den mest dominerande delen av internet, vars användare spenderar mer tid på sociala nätverk än på någon annan typ av webbplats. Detta gör att sociala nätverk kan vara mycket spännande ur marknadsförares och utvecklares perspektiv. En stor mängd personuppgifter är tillgängliga för marknadsförare och utvecklare och därför kan mer riktade annonser och mångsidiga applikationer utvecklas för användarna.

Tidigare hade sociala nätverk bara enkla profiler, ingen möjlighet att utveckla applikationer och endast annonsutrymme som kunde köpas, men nu kan man utveckla applikationer som kan byggas och integreras i ett socialt nätverk. Dessa applikationer kan skapas för nästan vilka ändamål som helst. Facebook applikationer har blivit enormt populära och värdefulla.

Plattformen som applikationerna körs på är mycket intressant, och är den främsta orsaken bakom Facebooks popularitet. Plattformen erbjuder ett brett utbud av verktyg och massor av användardata som kan användas i applikationer.

## 1.1 Bakgrund

Idén kom från att jag på min fritid utvecklade en Facebook applikation och jag ville sprida min visdom om vad jag lärt mig. Jag hade inte före detta projekt utvecklat på Facebook eller använt mig av liknande plattformar.

## 1.2 Syfte och mål

Syftet med detta projekt är att studera hur Facebooks plattform fungerar och hur man kan använda den för att utveckla applikationer. Denna avhandling behandlar Facebook applikationsutveckling och Facebook-plattformens funktioner. De viktigaste delarna av plattformen och utvecklingsprocessen är täckt med information om allmänna problem och plattformspecifika metoder. Denna avhandling fokuserar bara på Facebook-plattformens applikationer, inte externa webbplatser eller mobila plattformar.

## 1.3 Metoder

Jag kommer att analysera Facebooks plattform ur en yttlig synvinkel och kommer inte att gå alltför djup in i hur plattformen fungerar. Jag kommer att berätta om hur man utvecklar applikationer på Facebook och i slutet av arbetet presenterar jag kort en applikation som jag utvecklat med hjälp av några olika bibliotek som Facebook bjuder på.

## 1.4 Terminologi

**AJAX**-Asynchronous JavaScript And XML, en teknik som används för att hämta data från server med JavaScript.

**API**-Application Programming Interface, ett gränssnitt för att kommunicera med en viss plattform.

**CSS**-Cascading Style Sheets, ett kodnings språk för att ge stil åt en webbsida.

**DOM**-Document Object Model, strukturen/arkitekturen för en nätsida.

**FBML**-Facebook Markup Language, kod språk som används för Facebook applikationer.

**FBJS**-Facebook JavaScript, kod språk som används för Facebook applikationer.

**FQL**-Facebook Query Language, ett kod språk för att kommunicera med Facebooks databas.

**HTML**-Hypertext Markup Language, ett kod språk som används för att bygga upp nätsidor.

**HTTP**-Hypertext Transfer Protocol, är ett protokoll för att ansluta sig till en webbadress.



**HTTPS**-Hypertext Transfer Protocol Secure, är ett säkrare protokoll för att ansluta sig till en webbadress.

**JavaScript**-Kod språk som körs i webbläsaren.

**JSON**-JavaScript Object Notation, ett kompakt, textbaserat format.

**PHP**-PHP Hypertext Preprocessor, kod språk som körs på en server.

**SQL**-Structured Query Language, ett databasspråk.

**UI**-User Interface, gränssnitt.

**URL**-Uniform Resource Locator, adressen till en nätsida.

**XML**-Extensible Markup Language, är ett enkodnings språk.

**Access\_token**-Är en nyckel som används i Facebooks autentisering

**API**-gräns snitt

**Responsiv design**-flytande design

## **1.5 Avgränsning**

Jag kommer inte att ta upp externa sidor som använder sig av Facebooks API utan bara interna applikationer på Facebook. Målgruppen för arbetet är personer som jobbar i webbranschen, speciellt nya webbutvecklare. Jag utgår ifrån att personer som läser detta arbete har en bas uppfattning om webbt teknologier.

## 2 FACEBOOK


Facebook är ett socialt nätverk som ägs och drivs av Facebook Inc [6]. Facebooks grundare Mark Zuckerberg började utveckla webbplatsen år 2004 i sitt rum på Harvard [34]. År 2006 släpptes sidan för alla personer som har en giltig e-postadress och är över 13 år [35]. Sedan starten har Facebook praktiskt taget över hela internet. Facebook är den mest använda sociala nätverket och den näst mest besökta webbplatsen på hela internet strax efter Google [5]. Idag har Facebook över en miljard användare, över en biljon ”likes” och över 219 miljarder bilder laddade upp [4].


Facebooks största konkurrenter är MySpace och Google+, de har liknande funktioner men kommer inte ens nära Facebook i användare eller trafik. Det finns även många andra företag som erbjuder på liknande enskilda funktioner som Facebook har, som t.ex. Google erbjuder på reklam platser och det finns många andra tjänster som erbjuder möjligheten att man kan ladda upp bilder eller kommunicera med andra användare [3].


Orsaken till varför Facebook blev så populärt är att de kombinerade många olika tjänster till en. Man kan ladda upp bilder och dela dem, skriva inlägg, uppdatera statusar och kommunicera på många olika sätt med andra användare.

Det finns även möjlighet för företag att göra profiler och sidor på Facebook. Men för att man skall kunna göra företags sida så måste man ha ett personligt konto. Genom att användare gillar företaget sida så kan de ta emot uppdateringar från företaget.

Facebook har även blivit mycket populär bland utvecklare, eftersom det finns ett brett utbud av verktyg och data till förfogande. Efter att Facebook lanserade sin plattform för utvecklare har dess popularitet exploderat.


**Sponsored**  See All

 **Johan Lindeberg** likes Nordnet Suomi.

 **Nordnet Suomi**  
Like


---

**Netflix – 1 kk ILMAISEKSI**  
signup.netflix.com

 Katso elokuvia ja TV-sarjoja Netflixistä PS3:n, Xbox 360:n ja Wiin kautta!


---

**Nyt nettinäkyvyys kuntoon**  
oma.fonecta.fi

 Asiakkaasi ovat täällä, onko yrityksesi Facebook-sivu kunnossa? Lue lisää!


---

**Miesten tennarit!**  
zalando.fi

 Astele tyyllillä! Tennarit ja converseet miehille Zalando! ILMAINEN TOIMITUS JA PALAUTUS!

---

**iPhone 5**

 SIM-lukitsematon iPhone 5 nyt 699€. Osta vaikka osamaksulla!

Rasmus Johansson likes Hobby Hall.

Figur 1. Facebooks marknadsförings fält.

År 2012 förtjänade Facebook hela 82 % av sin inkomst genom skräddarsydda reklamer för sina användare (Figur 1). De förtjänar även en del på spel på Facebook, genom att man kan göra köp inne i spelen och för varje köp så får Facebook en liten andel.

Privatpersoner och företag kan båda göra reklam på Facebook. Som sagt så kan man skräddarsy reklamer på Facebook, man kan rikta dem på kön, plats, ålder, relationsstatus eller vad användaren har gillat på Facebook. Facebook uppskattades vara värd över 100 miljarder dollar år 2012 [34].

## 3 PLATTFORMEN

Facebook plattformen introducerades den 27 maj 2007 [8]. Plattformen erbjuder idag ett mycket brett sortiment av verktyg för programutvecklare att bygga applikationer som kan integreras med Facebook. Man kan även använda plattformen för att integrera redan existerande applikationer eller tjänster med Facebook.

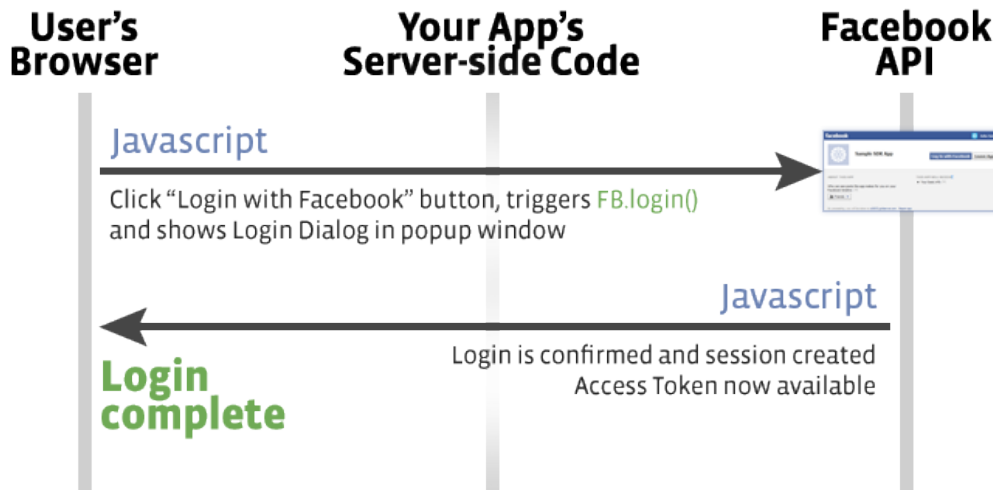
Det finns många orsaker varför Facebook lanserade plattformen. Genom att erbjuda ett brett sortiment av kraftfulla verktyg, kommer tjänsten att få mer innehåll och nya metoder att integrera sig. Dessa nya tjänster och funktioner kommer sedan att locka mer användare, som i sin tur ger mer data och statistik om användarna. Sociala plattformar värderas främst på hur många aktiva användare de har och mängden av data om dem. Detta beror på att sådan information lockar annonsörer, företag och utvecklare [1].

### 3.1 Komponenter

Facebook-plattformen kan delas in i sex huvuddelar; Login, Facebook API, Social Plugins, Graph API, FQL och Open Graph. Var och en av dessa delar har en egen roll i plattformen, även om största delen av dem kan kombineras [20].

#### 3.1.1 Login

Facebook erbjuder olika inloggningsmetoder för olika enheter och projekt. Var och en av dessa metoder använder sig av OAuth 2.0 standarden. När man loggar in genom en av dessa metoder genereras det en `access_token` som applikationen sedan kan använda för att göra API anrop för användarna. `Access_token` är en åtkomsttoken som Facebooks server genererar för att kunna hålla koll på varje session skilt. JavaScript SDK, iOS SDK och Android SDK sköter automatiskt om att `access_token` är giltig och att den skickas med anropen till Facebook API. Om man använder PHP SDK så måste man själv utveckla en funktion som sköter om detta. Man kan även kombinera JavaScript SDK med PHP SDK för att göra en bättre upplevelse för användarna. Man måste dock komma ihåg att även erbjuda en utloggningsmetod för användarna [12].



Figur 2. Facebook inloggning med JavaScript [12].

### 3.1.2 Facebook API

Hjärtat i Facebooks plattform är Facebook API. Den används för att autentisera användare som använder applikationer för att Facebook kräver detta så att applikationerna kan komma åt data och kunna modifiera den. Samtliga applikationer kan ha tillgång till användarnas data såsom vänner, nätverk, foton och meddelanden. Det är sådan information som gör att Facebook är världens största sociala nätverk. På grund av att en applikation kan ha tillgång till personlig data om användare som använder applikationen krävs det att man autentiserar användarna och applikationen. Genom denna API kan man även kommunicera med Facebooks andra API:n [22].

Facebook API är det ända sättet att använda sig av alla funktioner Facebook erbjuder på. Huvudfunktionerna är alarm, notifikationer och inbjudningar. Vissa av dessa funktioner kräver att applikationsspecifik data så som mallar eller frågor skall registreras innan de kan användas. Denna information är även registrerad med API:n. En del av dessa funktioner har även en daglig gräns hur många gånger man kan förfråga data eller skriva data. Orsaken till detta är att pröva undvika att någon missbrukar funktionerna.

Facebook API har en massa olika funktioner som kan användas i applikationer. Dessa funktioner är indelade i grupper baserade på typ och användning. Grupperna är administration, autentisering, data hämtning, publicering, foton och evenemang [23].

Man kommunicerar med Facebook API genom HTTP-förfrågningar. Nästan alla moderna programmeringsspråk kan skicka HTTP-förfrågningar och detta gör att plattformen kan användas på så många andra plattformar som möjligt [23].

Det finns fyra officiella API bibliotek som kan användas för att kommunicera med Facebooks API, dessa är PHP SDK, JS SDK, iOS SDK och Android SDK. Det finns även tredje parts bibliotek i andra språk för utvecklare som inte vill använda sig av de officiella biblioteken [21].

### **3.1.3 Sociala Plugins**

Sociala plugins är en funktion som Facebook erbjuder för sina användare. Detta är det lättaste sättet att integrera en extern webbplats med Facebook [25]. Sociala plugins består av många mindre delar:







- "Like button", tillåter användare att gilla innehåll på externa webbplatser.
- "Send button", tillåter användare att skicka innehåll åt en annan Facebook användare.
- "Follow button", tillåter användare att börja följa en Facebook sida från en extern webbplats.
- "Comments", tillåter användare att kommentera innehåll.
- "Activity feed", visar användare deras vänners aktiviteter.
- "Recommendation box", tillåter användare att rekommendera innehåll på externa webbplatser.
- "Like box", tillåter användare att gilla och se en Facebook sidas aktiviteter.
- "Registration", tillåter användare att registrerar sig på en extern webbplats med deras Facebook konton.
- "Facepile", tillåter användare att visa alla Facebook användare som har gillat eller börjat följa en Facebook sida.

### 3.1.4 Graph API

Graph API tillåter utvecklare att läsa från och skriva data till Facebook. Man får en enkel och konsekvent syn på Facebooks sociala graf genom att den presenterar t.ex. människor, foton, evenemang och sidor som objekt och visar kopplingar mellan dem [24].

Det finns många olika sätt att läsa data från Graph API:n och det beror mycket på vilken sort av data man vill ha. För att t.ex. hämta en användares profilbild så är det bara att göra en HTTP anrop på ”[https://graph.facebook.com/användarens namn/picture](https://graph.facebook.com/användarens_namn/picture)” [26]. Här är fler exempel på stödda objekt:

Tabell 1. Facebook Graph API stödda objekt [26.]

Object	Image	Example
People		<a href="https://graph.facebook.com/shaverm/picture">https://graph.facebook.com/shaverm/picture</a>
Events		<a href="https://graph.facebook.com/331218348435/picture">https://graph.facebook.com/331218348435/picture</a>
Groups		<a href="https://graph.facebook.com/203905803086416/picture">https://graph.facebook.com/203905803086416/picture</a>
Pages		<a href="https://graph.facebook.com/DoloresPark/picture">https://graph.facebook.com/DoloresPark/picture</a>
Applications		<a href="https://graph.facebook.com/2318966938/picture">https://graph.facebook.com/2318966938/picture</a>
Photo Albums		<a href="https://graph.facebook.com/376995711728/picture">https://graph.facebook.com/376995711728/picture</a>

Man kan även söka efter öppen data genom ”Graph API”. Här är några exempel på hur det fungerar [9]:

- Hämta alla öppna inlägg från användaren ”watermelon”:  
<https://graph.facebook.com/search?q=watermelon&type=post>
- Sök efter användaren ”Mark”:  
<https://graph.facebook.com/search?q=mark&type=user>
- Sök efter evenemang med ordet ”conference”:  
<https://graph.facebook.com/search?q=conference&type=event>
- Sök efter applikationen ”edomondo”:  
<https://graph.facebook.com/search?q=endomondo&type=application>
- Sök efter grupper med ordet ”programming”:  
<https://graph.facebook.com/search?q=programming&type=group>

Om man tar en närmare titt på adresserna så ser man att sökordet kommer efter ”q=” och efter ”type=” kommer vilken typ av objekt man söker efter.

För att kunna läsa eller skriva data genom ”Graph API” måste man ha rättigheter av användaren. Dessa rättigheter frågar man om när användaren börjar använda en tjänst som använder sig av Facebooks plattform [22]. Man kan läsa mera om ämnet i kapitel 3.2 Rättigheter.

### 3.1.5 FQL

Facebook Query Language ger ett lätt och användbart gränssnitt för att söka och hämta användardata. FQL är ett SQL liknande språk och alla som är bekanta med SQL kan lätt lära sig FQL. FQL har inte de komplexa metoder och funktioner som SQL har men grundfunktionerna är identiska. FQL är ett mycket effektivt sätt att söka och hämta användarnas personliga data [11].

FQL anrop kan skickas med hjälp av Facebooks API. Man kan skicka ett anrop i taget eller många anrop på en gång [11]. Anropen görs i en enkel SQL liknande form:



```
SELECT name FROM user WHERE uid = me()
```

I exemplet hämtas namnet av användaren som är inloggad i applikationen. Man kan även sortera data genom att lägga till ORDER BY eller avgränsa antalet resultat genom att lägga till LIMIT.

```
SELECT name FROM user WHERE uid = me() ORDER BY name LIMIT = 1
```

Man kan även skicka många anrop i FQL på engång, genom att man lägger till ”under anrop”. För att detta skall vara möjligt måste man använda WHERE [variabel] IN i anropet. När man ställer en WHERE [variabel] IN fråga måste anropet hänvisa till en tabell som har en unik identifikation:

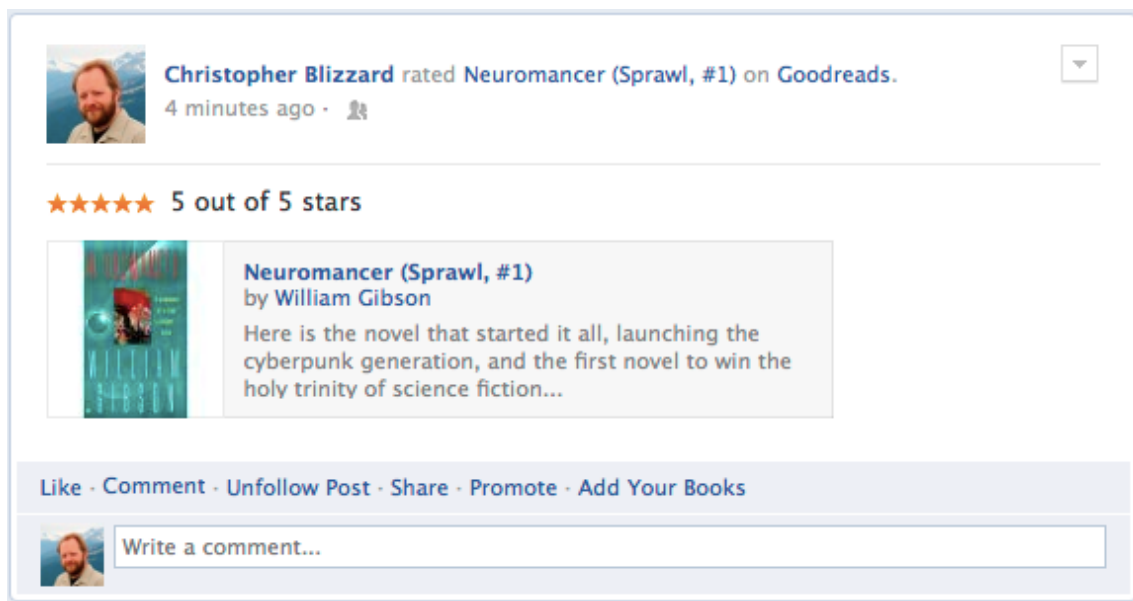
```
"anrop 1": "SELECT uid, rsvp_status FROM event_member WHERE eid=12345678"
```

```
"anrop 2": "SELECT name, url, pic FROM profile WHERE id IN (SELECT uid FROM #anrop 1)"
```

Med FQL kan man nå all data som Facebook har till förfogande men man kan bara läsa den inte modifiera. FQL stöder främst strängfunktioner. Dessa är mycket enkla funktioner som att kombinera två strängar, beräkna längden på en sträng och konvertera en sträng till versaler. Det finns även några icke-sträng funktioner som t.ex. få en tidpunkt och rand för att få ett slumpmässigt tal [11].

### 3.1.6 Open Graph

Open Graph tillåter att applikationer kan berätta berättelser på Facebook genom en strukturerad API. Berättelserna tillåter personer att dela med sig saker de gör, de människor de gör dem med och de platser där det hände (Figur 3). Open Graph erbjuder möjligheten för utvecklare att integrera sina applikationer djupt in i Facebook som ökar synligheten för applikationen [14].



Figur 3. Facebook Open Graph berättelse [14].

## 3.2 Rättigheter

När användaren loggar in i en applikation, får applikationen tillgång till deras offentliga profil och vänner. För att få mer information om en person, måste applikationen begära om ytterligare rättigheter av användaren [13][22].

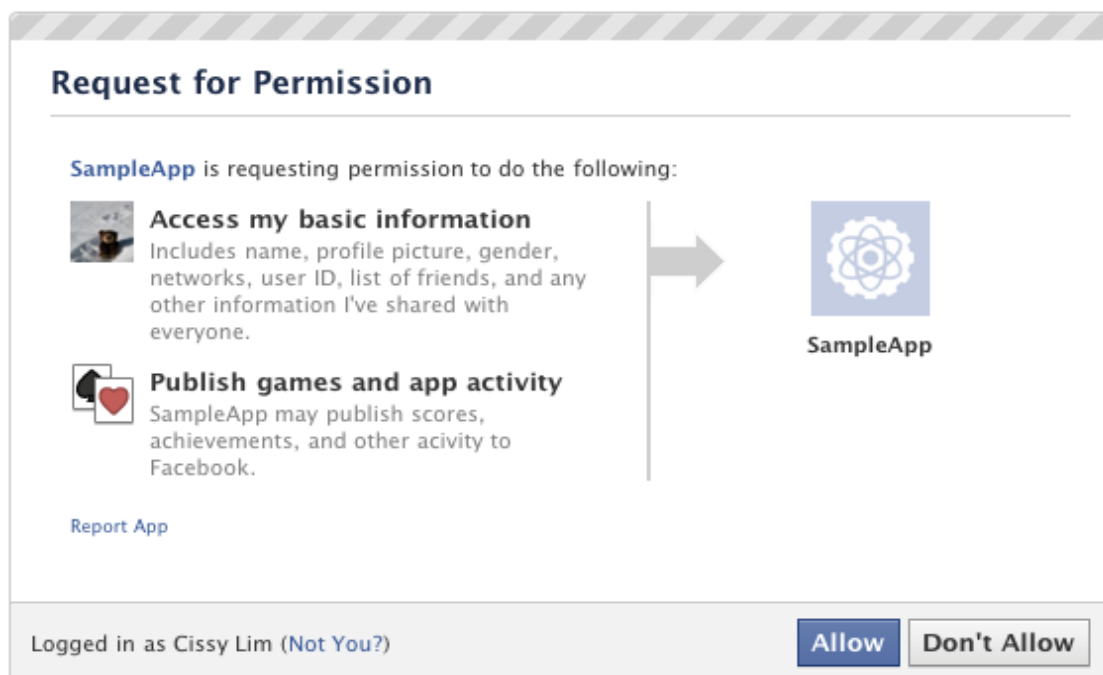
Rättigheterna delas in i följande grupper:

- Offentlig profil och vänlista: Hämta de offentliga delarna av användarens tidslinje, såsom deras namn, profilbild, åldersgrupp, kön, språk, land och annan offentlig information.
- E-post rättigheter: Ger tillgång till en användarens e-post.
- Utökade profilrättigheter: Ger tillgång till de flesta publiceringsrättigheter och känsligare information som användarens nyhetsflöde.
- Utökade rättigheter: Ger tillgång till mer känslig information och möjlighet att publicera och ta bort data.
- Open Graph rättigheter: Ger tillgång till Open Graph API, med undantag av publiceringsrättigheter.

- Sidrättigheter: Ger rättighet att fungera som användare på någon Facebook sida som användaren administrerar.

### 3.2.1 Publicerings rättigheter

För att kunna skriva data till en användares tidslinje, måste man begära skilt om skrivrättigheter av användaren [13][22]. Man måste be om ”Utökade rättigheter” (Figur 5). När en applikation har fått ”Utökade rättigheter”, så kan applikationen berätta historier via Open Graph.



Figur 4. Facebook applikation ber om rättigheter[13].

Graph API kan:

- Publicera på användarens tidslinje
- Ladda upp foto och video – inklusive skapa album
- Tagga bilder och videon
- Lägga till ”likes” och kommentarer till berättelser
- Publicera frågor
- Skapa anteckningar

- Publicera innehåll för evenemang eller grupper

Som en bästa praxis bör ”Utökade rättigheter” helt som alla andra rättigheter endast begäras när de på riktigt behövs [13][22].

### **3.2.2 Begära om rättigheter**

Som en tumregel, ju mer rättigheter en applikation ber om, desto mindre är chansen att användaren accepterar rättigheterna och fortsätter med inloggningen [13]. Be bara om rättigheter som är väsentliga för applikationen. Be om rättigheterna först sen när applikationen behöver dem, till exempel om man har utvecklat en applikation som visar var användarens vänner har checkat in, skall man be om rättigheten först när användaren vill se resultatet, på detta sätt ser användaren lättare varför applikationen vill ha rättigheten [13][22].

### **3.3 Sekretess**

När man integrerar applikationer med sociala nätverk kommer användarna att dela sig massor av information. Eftersom information i många fall kan vara mycket personlig har Facebook satt upp ett sekretessprotokoll [27].

På Facebook har användarna mångsidiga inställningar för detta, där användarna kan hantera hur deras information delas inom Facebook och andra tjänster. Användarna kan ställa in synlighetsgränser på många olika nivåer som t.ex. vänner, grupper, nätverk och applikationer. Till exempel, för ett fotoalbum kan en användare ange att endast en grupp av användarens vänner, alla vännens vänner eller alla Facebook användare kan se albumet. Dessa inställningar kan ställas in nästan för allt innehåll på Facebook [27].

Användare kan även ställa in om det är tillåtet för tredje partens applikationer att komma åt information, men också välja specifikt vilken applikation kan komma åt vilken data. Många av dessa inställningar påverkar även hur bra vissa applikationer fungerar. Om t.ex. användaren har ställt in att ingen får komma åt ens personliga information kan detta leda till att många applikationer inte kommer att fungera för denna användare [27].

## **4 FACEBOOK APPLIKATIONER**

Sedan 2007 har det varit möjligt att utveckla egna applikationer på Facebook. Idag finns det redan över nio miljoner applikationer som använder sig av Facebooks applikationsplattform [2].

### **4.1 Uppbyggnad**

En Facebook applikation är en stor samling av olika objekt och funktioner som fungerar tillsammans. Eftersom grundidén med en social applikation är att kunna dela med sig till andra så är största delen av de olika funktionerna kommunikations relaterade. Genom att Facebook erbjuder så många olika sätt att integrera sig så kan man bygga en massa olika sorts applikationer [7][16][20][21].

#### **4.1.1 Canvas**

Canvasen är en sorts ram och den är den viktigaste delen i en applikation och är den delen som skiljer en applikation från en annan. Det är den delen av applikationen som körs inne i en Facebook sida, såsom om det vore en del av Facebook. Alla aktiviteter skall utföras på canvasen som är kopplade till resten av applikationen. De andra delarna av applikationen körs genom att användaren utför aktiviteter på canvasen [15].

#### **4.1.2 Profil**

Varje applikation har en profil. Applikationens profil är som en profil eller Om-sida om applikationen. Det är där var användarna går för att se vem som använder applikationen och för att läsa användarnas recensioner om applikationen. Helt som alla andra profilsidor på Facebook har denna också en vägg som användarna kan skriva kommentarer på. Profilsidan kan även användas för att ge mer detaljerad information om applikationen eller till exempel använda sidan som ett diskussionsforum. Det är också en bra plats att samla in feedback från användarna eller berätta om uppkommande uppdateringar [15].

### **4.1.3 Notifikationer**

Notifikationer är Facebooks sätt att meddela användarna om att det hänt något intressant inom användarnas nätverk på Facebook. En notifikation kan meddela användarna om nytt innehåll eller att någon av användarnas vänner har använt en applikation och delat med sig sin aktivitet. Notifikationer är ett bra sätt att få gamla användare att returnera till en applikation och även för att få nya användare att hitta applikationen. Det finns en daglig gräns för hur många notifikationer man kan skicka genom en applikation, för att undvika missbruk [28][15].

### **4.1.4 Alarm**

Ett alarm kan ha mer innehåll än en notifikation och de skickas till användarens personliga e-post konto. Detta är ett bra sätt att meddela användare som inte är för tillfället online på Facebook att något har hänt och de bör logga in. Dessa alarm har även en daglig gräns för hur många man kan skicka [15].

### **4.1.5 Inbjudningar**

I en applikation kan användarna endast kommunicera med andra användare som använder applikationen. Dock kan användare skicka inbjudningar till sina vänner för att få dem att använda programmet. Detta är en mycket viktig funktion för varje Facebook applikation. Inbjudningar är det bästa sättet att få nya användare till applikationer genom att utnyttja användarnas sociala relationer. Helt som notifikationer har även inbjudningar en daglig gräns [15].

### **4.1.6 Nyhetsflöde**

Nyhetsflöde är en mycket viktig del av en applikation. Användarna kan publicera berättelser om sina aktiviteter i sitt nyhetsflöde. Till skillnad från notifikationer och alarm som bara sänds till en viss grupp av användare är en användares nyhetsflöde synligt till alla som har tillgång till användarens profil. Nyhetsflöde berättelser har ingen daglig gräns men användaren måste godkänna varje berättelse före den publiceras. Dessa berättelser kan innehålla foton eller videoklipp, vilket gör dem mer mångsidiga

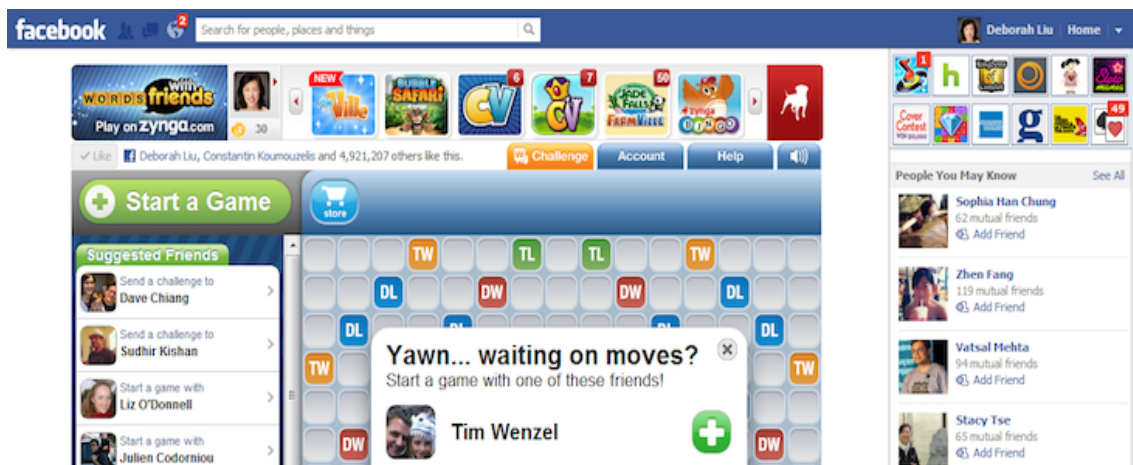
än notifikationer. På grund av att man kan göra mycket personliga berättelser och de har en mycket stor publik är detta också ett mycket bra sätt att marknadsföra en applikation och få nya användare [15].

## 4.2 Typ av applikation

Det finns två olika typer av applikationer att utveckla på Facebook. Man kan bygga dem någongdera som en canvas-applikation eller som en tab-applikation [7].

### 4.2.1 Canvas-applikation

En canvas applikation (Figur 6) är i stort sätt en vanlig nätsida som visas genom en iframe i Facebook. När man skapar en canvas applikation kan man välja en fördefinierad bredd på 760 pixlar eller låta applikationen vara responsiv, samma sak gäller även för höjden.

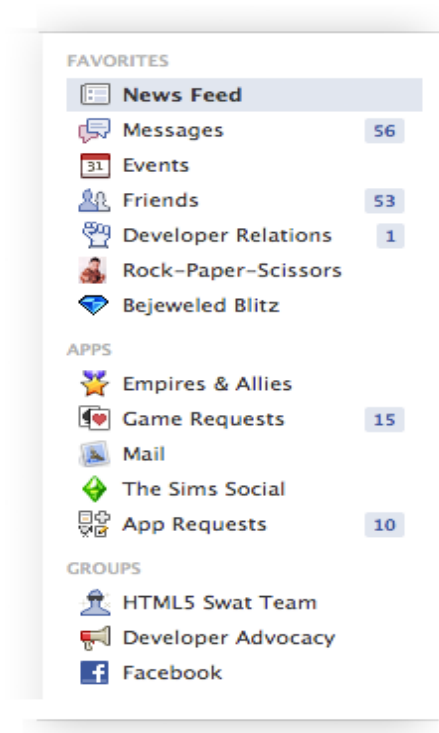


Figur 5. Facebook canvas applikation [15].

Eftersom applikationen är inne i en iframe så används inte Facebooks servrar för att köra applikationen och detta brukar ge en mycket bra prestation. Tack vare att det inte finns några begränsningar för vad för HTML-element och bibliotek som kan användas i en iframe är det mycket lätt att integrera en redan existerande webbapplikation med Facebook. De ändra ändringarna som möjligtvis kan behövas är i inloggning och

autentisering. Genom att använda en canvas applikation kan utvecklare snabbt integrera sina applikationer med Facebook [15].

När en användare besöker en canvas applikation kommer Facebooks server att returnera en blank sida med en iframe som respons. När användarens webbläsare har tagit emot responsen kommer iframen att hämta sitt innehåll från en förspecificerad adress. Detta är när applikationen egentligen startar och börjar göra sina API anrop. Eftersom Facebook erbjuder på så många olika bibliotek så underlättar det utvecklingen att använda sig av dem för att göra API anrop [15]. Canvas applikationer kan “installeras” i användarnas profiler så att man kan komma snabbt åt dem (Figur 7).



Figur 6. Facebook sidofält [15.]

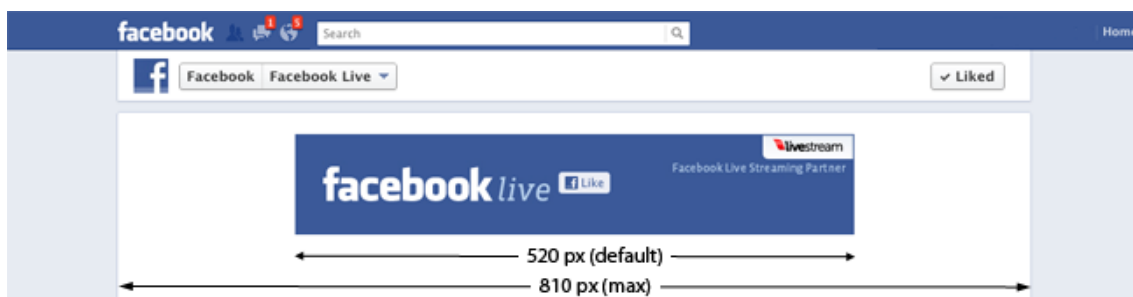
#### 4.2.2 Tab-applikation

Facebook Pages är en mycket använd egenskap på Facebook. Stora varumärken, kändisar osv. använder Facebook Pages som deras ”sociala hem” på webben. En av de



mest intressanta funktionerna som Facebook erbjuder utvecklare är möjligheten att köra applikationer inom en sådan sida. Dessa applikationer kallas för tab-applikationer och de liknar till utseendet mycket Facebooks egna sidor [29].

Till skillnad från canvas applikationer, när man skapar en tab-applikation kan man inte definiera bredden av innehållet, utan den är alltid 810 pixlar (Figur 3).



Figur 7. Facebook tab applikation [29].

Tab-applikationer integreras med Facebook helt på samma sätt som en canvas applikation, alltså via en iframe [29]. Man kan även använda sig av alla de bibliotek som Facebook erbjuder.

### 4.3 Förvaring av data

Eftersom Facebook inte upprätthåller applikationer, krävs det att man själv skaffar sig en webbplats var man kan köra applikationen. När användare använder applikationen kommer Facebooks server att skicka begäran till din webbplats för att hämta data till applikationen. Facebook hämtar helt enkelt data från din externa webbplats, bearbetar den och integrerar den i tjänsten på Facebook.

### 4.4 App center

Facebook har en plats var applikationsutvecklare kan visa upp sina applikationer, App center. Det är inte obligatoriskt att skicka in sin applikation, men det är ett mycket effektivt sätt att få nya användare. Man kan sortera applikationerna efter popularitet och

nyaste. Applikationerna är även indelade i olika kategorier enligt användningsområde och funktionalitet för att underlätta sökningen av en viss typ av applikation. Det finns sammanlagt 17 olika kategorier som man kan välja mellan när man skapar en applikation [30].

**Kategorierna:**

- böcker
- kommunikation
- utbildning
- mode
- mat och dryck
- spel
- hälsa och motion
- filmer
- musik
- nyheter
- fotografier
- shopping
- sporter
- TV
- resor och lokalisering
- verktyg
- filmklipp

App center är en mycket bra plats att söka efter inspiration genom att gå igenom applikationer som intresserar dig. App center innehåller information, en kort beskrivning och ett vitsord från användarna om varje applikation som blivit skickad dit. Från App center blir användarna skickade till applikationens profilsida, var de kan läsa mer om applikationerna [30].

## 5 UTVECKLING

Facebook applikationsutveckling är mycket likt traditionell webbutveckling. Dock krävs det att man förstår hur Facebooks API fungerar och att man klarar av att använda de bibliotek som Facebook bjuder på.

### 5.1 Förberedning

Innan utvecklingen kan börja, måste några nödvändiga saker utföras. Facebook kräver att alla utvecklare skall registrera sig. Detta görs i Facebook genom att man går till Facebooks utvecklingsida som är även den platsen var alla applikationers konfigurationer utförs. När man skapar en ny applikation genererar Facebook automatiskt ett id och en hemlig kod åt din applikation som används när din applikation gör API anrop. Eftersom Facebook inte upprätthåller applikationer måste användaren själv skaffa en webbplats var applikationen utvecklas. Det rekommenderas även att använda sig av ett av Facebooks bibliotek, dessa kan laddas ner från:

<http://developers.facebook.com/docs/sdks/>

### 5.2 Verktyg

Facebook erbjuder några mycket användbara verktyg som hjälper i utvecklingsprocessen och även efter att applikationen har blivit lanserad. Det finns bland annat testkonsoler för JavaScript SDK och Graph API. JavaScript SDK konsolen är ett mycket bra och effektivt sätt att lära sig hur man kan använda JavaScript SDK för applikationer. Det finns även möjlighet att göra test användare för att lättare kunna testa en applikation och när applikationen är klar så kan man släppa en beta version som bara ett begränsat antal användare kan använda. För att debugga en applikation kan man använda sig av Open Graph Debugger eller Access token Tool [20].

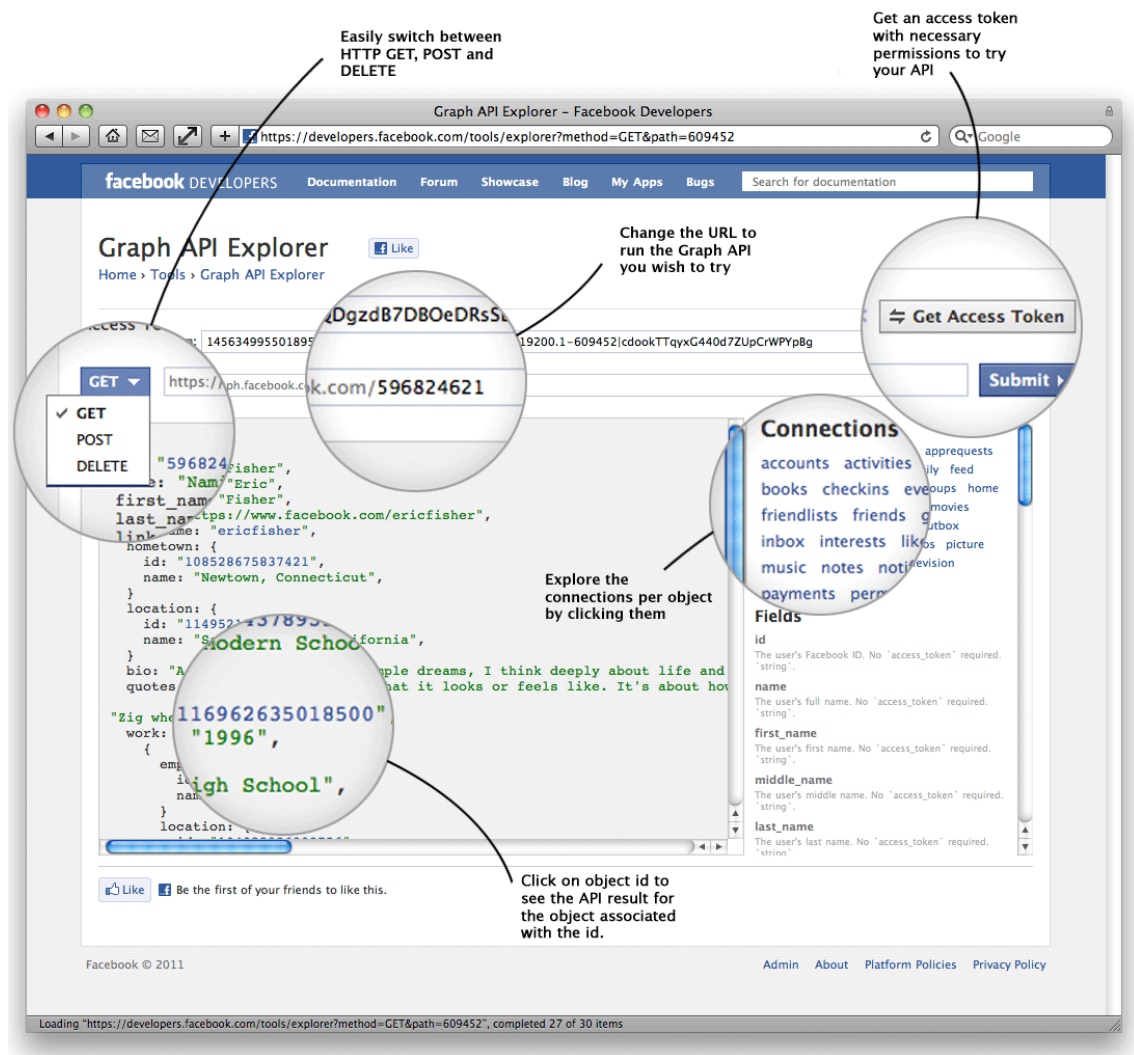
#### 5.2.1 JavaScript SDK Konsol

JavaScript SDK test konsolen kan användas för att testa olika Facebook API anrop. Konsolen har alla API-metoder och SDK funktioner som är tillgängliga för tillfället. Det

finns även färdiga exempel på olika funktioner och metoder för nybörjare som vill använda sig av JavaScript SDK biblioteket [21][31].

## 5.2.2 Graph API Explorer konsol

Graph API konsolen (Figur 8) kan användas för att lära sig hur Graph API anrop fungerar.



Figur 8. Graph API Explorer konsolen [18].

I Graph API konsolen kan man göra följande:

- Göra anrop till Graph API och få ett formaterat resultat tillbaka

- Gå igenom resultatet för att förstå hur de olika objekten och innehållet hänger ihop
- Begära lätt en åtkomstoken med olika rättigheter
- Byta lätt mellan http GET, POST och DELETE anrop för att hämta, skapa och radera data
- Hoppa mellan objekt genom att bara klicka på deras id i det formaterade resultatet.

Det är dock endast möjligt att visa information som användaren har rättigheter till. Sådan information är data från användaren, användarens vänner eller sådan data som har publicerats med rättigheten ”public” på Facebook [17][18][24].

### **5.2.3 Open Graph Debugger**

Open Graph Debugger kan användas för att debugga en sidas open graph taggar. Det är ett mycket bra verktyg för att underlätta felsökningen. Man kan gå igenom en applikations Open Graph berättelser, funktioner och objekt [32].

### **5.2.4 Statistik**

Facebook erbjuder även på ett statistikverktyg Insights för utvecklare som vill följa applikationens trafik och resursanvändning. Insights har många olika funktioner som kan hjälpa till att göra applikationen fungera bättre och snabbare eller så kan den användas för att samla in information om användningen av applikationen. En sektion i Insights visar till exempel hur många aktiva användare applikationen har [33].

## **5.3 Konfiguration av applikationen**

Att skapa en applikation och konfigurera den är verkligen lätt och okomplicerat. Det finns bara ett fåtal inställningar som behövs för att få en applikation och fungera. Även om det är enkelt att skapa en applikation så finns det massor inställningar för mer avancerade applikationer [34].

### **5.3.1 Obligatoriska inställningar**

Först och främst måste en applikation ha ett unikt namn. Namnet får inte innehålla ordet ”face”, eftersom alla namn som innehåller ordet är reserverade för Facebooks egna bruk. En annan inställning som är obligatorisk är applikationens webbadress. Den här adressen måste även bara unikt och helst innehålla applikationens namn. Dessa två inställningar är de enda som är synliga åt andra användare än utvecklaren själv [34].

## **5.4 Publicering**

Som tidigare nämnts, är det möjligt att publicera applikationer till Facebooks App center. För att säkra sig om att applikationen publiceras måste applikationen fungera korrekt och man måste uppfylla alla krav som ställts för applikationerna [30][34].

Facebook har satt några krav för applikationer innan de kan publiceras till App center. För att Facebook skall kunna säkra sig om att det är en fungerande applikation måste applikationen ha en fungerande webbadress. Applikationen måste även ha en beskrivning och en logo, eftersom dessa visas i App center. Före en applikation kan publiceras måste applikationen ha åtminstone fem användare eller 10 oregistrerade användare. En oregistrerad användare har inte registrerat sig i applikationen utan använder den utan autentisering [30][34].

## **5.5 Exempelapplikation**

Jag har byggt ihop en exempel canvas applikation (Figur 10) som använder sig av PHP SDK och JavaScript SDK biblioteken, applikationen hittar man på:

<http://apps.facebook.com/sbthesis/>



Figur 9. Exempel applikation

I applikationen används Graph API för att hämta användarens vänner, senaste foton och saker användaren gillat. För att hämta vänner som använder applikationen använder applikationen sig av en FQL sökning.

JavaScript SDK biblioteket används för att logga in användaren och även för att dela med sig applikationen.

I själva koden finns det förklaring för de olika funktionerna och metoderna som jag har använt mig av. Källkoden hittar man i bilaga 1.

## KÄLLOR

Elektroniska källor:

1. BBC NEWS, 2007, *15 reasons Facebook may be worth \$15bn*, [WWW].

Tillgänglig: <http://news.bbc.co.uk/2/hi/7061398.stm>

Hämtad 23.3.2013

2. Brittany Darwell, 2012, *Facebook platform supports more than 42 million pages and 9 million apps*, [WWW].

Tillgänglig: <http://www.insidefacebook.com/2012/04/27/facebook-platform-supports-more-than-42-million-pages-and-9-million-apps/>

Hämtad 19.3.2013

3. Chloe Albanesius, 2013, *Facebook Names Google, Twitter, Microsoft as Major Rivals*, [WWW]. Tillgänglig:

<http://www.pcmag.com/article2/0,2817,2399738,00.asp>

Hämtad: 7.4.2013

4. Donna Tam, 2012, *Facebook by the numbers: 1.06 billion monthly active users*, [WWW].

Tillgänglig:

[http://news.cnet.com/8301-1023\\_3-57566550-93/facebook-by-the-numbers-1.06-billion-monthly-active-users/](http://news.cnet.com/8301-1023_3-57566550-93/facebook-by-the-numbers-1.06-billion-monthly-active-users/)

Hämtad 19.3.2013

5. ebizmba, 2013, *Top 15 Most Popular Websites*, [WWW].

Tillgänglig: <http://www.ebizmba.com/articles/most-popular-websites>

Hämtad 19.3.2013

6. Facebook, 2012, *Key Facts*, [WWW].

Tillgänglig: <http://newsroom.fb.com/Key-Facts>



Hämtad: 7.4.2013

7. Facebook, 2012, *Facebook Developers Guide*, [WWW].

Tillgänglig: <http://developers.facebook.com>.

Hämtad: 19.3.2013

8. Facebook, 2007, *Facebook Platform Launches*, [WWW].

Tillgänglig: <http://developers.facebook.com/blog/post/21>

Hämtad 23.3.2013

9. Facebook, 2013, *Search*, [WWW].

Tillgänglig: <https://developers.facebook.com/docs/reference/api/search/>

Hämtad 23.3.2013

10. Facebook, 2013, *Pictures*, [WWW].

Tillgänglig: <https://developers.facebook.com/docs/reference/api/using-pictures/>

Hämtad 23.3.2013

11. Facebook, 2013, *FQL*, [WWW].

Tillgänglig: <http://developers.facebook.com/docs/technical-guides/fql/>

Hämtad 25.3.2013

12. Facebook, 2013, *Login Architecture*, [WWW].

Tillgänglig: <http://developers.facebook.com/docs/concepts/login/login-architecture>

Hämtad 23.3.2013

13. Facebook, 2013, *Login Permissions*, [WWW].

Tillgänglig: <http://developers.facebook.com/docs/concepts/login/permissions-login-dialog/>

Hämtad 23.3.2013

14. Facebook, 2013, *Open Graph Overview*, [WWW].

Tillgänglig: <https://developers.facebook.com/docs/opengraph/overview/#how>

Hämtad 23.3.2013

15. Facebook, 2013, *Canvas*, [WWW].

Tillgänglig: <http://developers.facebook.com/docs/guides/canvas/>

Hämtad 26.3.2013

16. Facebook, 2013, *Facebook Tools*, [WWW].

Tillgänglig: <http://developers.facebook.com/tools/>

Hämtad 29.3.2013

17. Facebook, 2013, *Facebook Graph API Explorer*, [WWW].

Tillgänglig: <http://developers.facebook.com/tools/explorer>

Hämtad 29.3.2013

18. Facebook, 2011, *Facebook Graph API Explorer*, [WWW].

Tillgänglig: <http://developers.facebook.com/blog/post/517/>

Hämtad 29.3.2013

20. Facebook, 2013, *Facebook Tools*, [WWW].

Tillgänglig: <http://developers.facebook.com/web/>

Hämtad: 7.4.2013

21. Facebook, 2013, *Facebook SDK*, [WWW].

Tillgänglig: <https://developers.facebook.com/docs/sdks/>

Hämtad: 7.4.2013

22. Facebook, 2013, *Access Tokens and Types*, [WWW].

Tillgänglig: <http://developers.facebook.com/docs/concepts/login/access-tokens-and-types/>

Hämtad: 7.4.2013

24. Facebook, 2013, *Getting Started: The Graph API*, [WWW].

Tillgänglig: <http://developers.facebook.com/docs/getting-started/graphapi/>

Hämtad: 7.4.2013

25. Facebook, 2013, *Facebook for Webpages*, [WWW]

Tillgänglig: <http://developers.facebook.com/docs/guides/web/>

Hämtad: 7.4.2013

26. Facebook, 2013, *Facebook Pictures*, [WWW].

Tillgänglig: <http://developers.facebook.com/docs/reference/api/using-pictures/>

Hämtad: 7.4.2013

27. Facebook, 2013, *Facebook Platform Policy*, [WWW].

Tillgänglig: <http://developers.facebook.com/policy/>

Hämtad: 7.4.2013

28. Facebook, 2013, *App Notifications*, [WWW].

Tillgänglig: <https://developers.facebook.com/docs/concepts/notifications/>

Hämtad: 7.4.2013

29. Facebook, 2013 *Page Tab Tutorial*, [WWW].

Tillgänglig: <http://developers.facebook.com/docs/appsonfacebook/pagetabs/>

Hämtad: 7.4.2013

30. Facebook, 2013, *App Center*, [WWW].

Tillgänglig: <http://developers.facebook.com/docs/appcenter/>

Hämtad: 7.4.2013

31. Facebook, 2013, *Javascript Test Console*, [WWW].

Tillgänglig: <https://developers.facebook.com/tools/console/>

Hämtad: 7.4.2013

32. Facebook, 2013, *Debugger*, [WWW].

Tillgänglig: <https://developers.facebook.com/tools/debug>

Hämtad: 7.4.2013

33. Facebook, 2013, *Insights*, [WWW].

Tillgänglig: <https://developers.facebook.com/docs/insights/>

Hämtad: 7.4.2013

34. Facebook, 2013, *Apps*, [WWW].

Tillgänglig: <https://developers.facebook.com/apps>

Hämtad: 7.4.2013

35. Jeff Robertson, 2012, *How Does Facebook Make Money – The answer is simple*,

[WWW]. Tillgänglig: <http://www.empowernetwork.com/empowernlm/blog/how-does-facebook-make-money-the-answer-is-simple/>

Hämtad 19.3.2013

36. Nicholas Carlson, 2012, *At Last - The Full Story Of How Facebook Was Founded*,

[WWW]. Tillgänglig: <http://www.businessinsider.com/how-facebook-was-founded-2010-3#we-can-talk-about-that-after-i-get-all-the-basic-functionality-up-tomorrow-night-1>

Hämtad 19.3.2013

## BILAGOR

### Bilaga 1. Källkoden för exempel applikationen

Index.php:

```
<?php
```

```
// Detta är Samuel Backmans exempel applikation för examensarbetet.
```

```
// I denna applikation finns det några exempel på hur man kan använda,  
autentisering, Graph API och FQL.
```



```
*****  
**/  
  
require_once('sdk/src/facebook.php');  
  
$facebook = new Facebook(array(  
    'appId' => AppInfo::appID(),  
    'secret' => AppInfo::appSecret(),  
    'sharedSession' => true,  
    'trustForwarded' => true,  
));  
  
$user_id = $facebook->getUser();  
if ($user_id) {  
    try {  
        // Hämta användarens data  
        $basic = $facebook->api('/me');  
    } catch (FacebookApiException $e) {  
        if (!$facebook->getUser()) {  
            header('Location: '. AppInfo::getUrl($_SERVER['REQUEST_URI']));  
            exit();  
        }  
    }  
}  
  
//Graph API:
```

```

// Här hämtas fyra saker användaren gillat . 'limit=*" hämtar *
resultat.

$likes = idx($facebook->api('/me/likes?limit=4'), 'data', array());

// Här hämtas fyra av användarens vänner.

$friends = idx($facebook->api('/me/friends?limit=4'), 'data', array());

// Här hämtas 16 av användarens senaste foton.

$photos = idx($facebook->api('/me/photos?limit=16'), 'data', array());

//Här hämtas med FQL vänner som använder denna applikation.

$app_using_friends = $facebook->api(array(

    'method' => 'fql.query',

    'query' => 'SELECT uid, name FROM user WHERE uid IN(SELECT uid2 FROM
friend WHERE uid1 = me()) AND is_app_user = 1'

));
}

// Hämta information om applikationen; namn och id.

$app_info = $facebook->api('/'. AppInfo::appID());

$app_name = idx($app_info, 'name', '');

?>

<!DOCTYPE html>

```

```

<html xmlns:fb="http://ogp.me/ns/fb#" lang="en">

<head>

  <meta charset="utf-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=2.0, user-scalable=yes" />

  <title><?php echo he($app_name); ?></title>

  <link rel="stylesheet" href="stylesheets/screen.css" media="Screen"
type="text/css" />

  <link rel="stylesheet" href="stylesheets/mobile.css" media="handheld,
only screen and (max-width: 480px), only screen and (max-device-width:
480px)" type="text/css" />

  <!--[if IEMobile]>

  <link rel="stylesheet" href="mobile.css" media="screen"
type="text/css" />

  <![endif]-->

  <!-- Här är Open Graph tags. För att läsa mer om Open Graph, gå till:
-->

  <!-- 'https://developers.facebook.com/docs/opengraph/' -->

  <meta property="og:title" content="<?php echo he($app_name); ?>" />

  <meta property="og:type" content="website" />

  <meta property="og:url" content="<?php echo AppInfo::getUrl(); ?>" />

  <meta property="og:image" content="<?php echo AppIn-
fo::getUrl('/logo.png'); ?>" />

  <meta property="og:site_name" content="<?php echo he($app_name); ?>"
/>

  <meta property="og:description" content="My first app" />

```



```
<meta property="fb:app_id" content="<?php echo AppInfo::appID(); ?>"
/>
```

```
<script type="text/javascript" src="/javascript/jquery-
1.7.1.min.js"></script>
```

```
<script type="text/javascript">
```

```
function logResponse(response) {
    if (console && console.log) {
        console.log('The response was', response);
    }
}
```

```
$(function(){
```

```
    // Kollar om knapparna klickas
```

```
    $('#postToWall').click(function() {
```

```
        FB.ui(
```

```
            {
```

```
                method : 'feed',
```

```
                link    : $(this).attr('data-url')
```

```
            },
```

```
            function (response) {
```

```
                // Om responsen är tom, ångrade användaren sig
```

```
                if (response != null) {
```

```
                    logResponse(response);
```

```
                }
```

```
    }  
  );  
});  
  
$('#sendToFriends').click(function() {  
  FB.ui(  
    {  
      method : 'send',  
      link   : $(this).attr('data-url')  
    },  
    function (response) {  
      // Om responsen är tom, ångrade användaren sig  
      if (response != null) {  
        logResponse(response);  
      }  
    }  
  );  
});  
  
$('#sendRequest').click(function() {  
  FB.ui(  
    {  
      method : 'apprequests',  
      message : $(this).attr('data-message')  
    },  
    function (response) {  
      // Om responsen är tom, ångrade användaren sig  
      if (response != null) {  
        logResponse(response);  
      }  
    }  
  );  
});
```

```

        function (response) {

            // Om responsen är tom, ångrade användaren sig

            if (response != null) {

                logResponse(response);

            }

        }

    );

});

});
</script>

<!--[if IE]>

    <script type="text/javascript">

        var tags = ['header', 'section'];

        while(tags.length)

            document.createElement(tags.pop());

    </script>

<![endif]-->

</head>

<body>

    <div id="fb-root"></div>

    <script type="text/javascript">

        window.fbAsyncInit = function() {

            FB.init({

                appId      : '<?php echo AppInfo::appID(); ?>', // App ID

```

```
        channelUrl : '///php echo $_SERVER["HTTP_HOST"];
?&gt;/channel.html', // Channel Filen

        status      : true, // kolla om användaren är in loggad

        cookie      : true, // tillåt cookies

        xfbml       : true // parse XFBML

    });

    // Kolla om användaren klickar på "log in" knappen
    FB.Event.subscribe('auth.login', function(response) {

        // Här laddas sidan på nytt så att php SDK kan läsa cookien som
        JS SDK sätter

        window.location = window.location;

    });

    FB.Canvas.setAutoGrow();

};

// Ladda JS SDK
(function(d, s, id) {
    var js, fjs = d.getElementsByTagName(s)[0];
    if (d.getElementById(id)) return;
    js = d.createElement(s); js.id = id;
    js.src = "//connect.facebook.net/en_US/all.js";
    fjs.parentNode.insertBefore(js, fjs);
}(document, 'script', 'facebook-jssdk'));
&lt;/script&gt;</pre
```

```

<header class="clearfix">

    <?php if (isset($basic)) { ?>

        <p id="picture" style="background-image:
url(https://graph.facebook.com/<?php echo he($user_id);
?>/picture?type=normal)"></p>

    <div>

        <h1>Välkommen, <strong><?php echo he(idx($basic, 'name'));
?></strong></h1>

        <p class="tagline">

            Det här är Facebook applikationen

            <a href="<?php echo he(idx($app_info, 'link'));?>" tar-
get="_top"><?php echo he($app_name); ?></a>

        </p>

    <div id="share-app">

        <p>Dela:</p>

        <ul>

            <li>

                <a href="#" class="facebook-button" id="postToWall" data-
url="<?php echo AppInfo::getUrl(); ?>">

                    <span class="plus">Publicera på din vägg</span>

                </a>

            </li>

            <li>

                <a href="#" class="facebook-button speech-bubble"
id="sendToFriends" data-url="<?php echo AppInfo::getUrl(); ?>">

```

```

        <span class="speech-bubble">Skicka meddelande</span>

    </a>

</li>

<li>

    <a href="#" class="facebook-button apprequests"
id="sendRequest" data-message="Test this awesome app">

        <span class="apprequests">Skicka förfrågan</span>

    </a>

</li>

</ul>

</div>

</div>

<?php } else { ?>

<div>

    <h1>Välkommen</h1>

    <div class="fb-login-button" data-
scope="user_likes,user_photos"></div>

</div>

<?php } ?>

</header>

<section id="get-started">

    <p>Välkommen till Samuel Backmans exempel applikation för
examensarbetet!</p>

    <a href="https://developers.facebook.com/docs/" target="_top"
class="button">Läs mer om hur man kan utveckla applikationer för
Facebook</a>

</section>

```

```

<?php
    if ($user_id) {
?>

<section id="samples" class="clearfix">

    <h1>Exempel av Facebook Graph API</h1>

    <div class="list">

        <h3>Några av dina vänner</h3>

        <ul class="friends">

            <?php

                foreach ($friends as $friend) {

                    // Extract the pieces of info we need from the requests
above

                    $id = idx($friend, 'id');

                    $name = idx($friend, 'name');

                ?>

            <li>

                <a href="https://www.facebook.com/<?php echo he($id); ?>" tar-
get="_top">

                    ">

                    <?php echo he($name); ?>

                </a>

            </li>

            <?php

```

```

    }

    ?>

</ul>

</div>

<div class="list inline">

    <h3>Dina senaste foton</h3>

    <ul class="photos">

        <?php

            $i = 0;

            foreach ($photos as $photo) {

                // Använd responsen från Graph API

                $id = idx($photo, 'id');

                $picture = idx($photo, 'picture');

                $link = idx($photo, 'link');

                $class = ($i++ % 4 === 0) ? 'first-column' : '';

            ?>

            <li style="background-image: url(<?php echo he($picture); ?>);"
class="<?php echo $class; ?>">

                <a href="<?php echo he($link); ?>" target="_top"></a>

            </li>

            <?php

                }

            ?>

        </ul>

```



```

</div>

<div class="list">

  <h3>Saker du gillat</h3>

  <ul class="things">

    <?php

      foreach ($likes as $like) {

        // Använd responsen från Graph API

        $id = idx($like, 'id');

        $item = idx($like, 'name');

        // Detta vill visa saker användaren gillat och visa en länk
till dem

        ?>

        <li>

          <a href="https://www.facebook.com/<?php echo he($id); ?>" tar-
get="_top">

            ">

            <?php echo he($item); ?>

          </a>

        </li>

        <?php

          }

        ?>

      </ul>

    </div>

```

```
<div class="list">

  <h3>Vänner som använder appen</h3>

  <ul class="friends">

    <?php

      foreach ($app_using_friends as $auf) {

        // Använd responsen från Graph API

        $id = idx($auf, 'uid');

        $name = idx($auf, 'name');

      ?>

    <li>

      <a href="https://www.facebook.com/<?php echo he($id); ?>" tar-
get="_top">

        ">

        <?php echo he($name); ?>

      </a>

    </li>

    <?php

      }

    ?>

  </ul>

</div>

</section>

<?php
```

```
}  
?>  
  
<section id="guides" class="clearfix">  
  <div style="margin: 0 auto; width: 540px;">  
    <h1>Läs mer om Facebook Apps</h1>  
    <ul>  
      <li>  
        <a href="https://developers.facebook.com/docs/guides/web/" target="_top" class="icon websites">Webbplatser</a>  
      </li>  
      <li>  
        <a href="https://developers.facebook.com/docs/guides/mobile/" target="_top" class="icon mobile-apps">Mobil applikationer</a>  
      </li>  
      <li>  
        <a href="https://developers.facebook.com/docs/guides/canvas/" target="_top" class="icon apps-on-facebook">Facebook applikationer</a>  
      </li>  
    </ul>  
  </div>  
</section>  
  
</body>  
</html>
```

Utils.php:

```
<?php

/**
 * @return
 */

function idx(array $array, $key, $default = null) {
    return array_key_exists($key, $array) ? $array[$key] : $default;
}

function he($str) {
    return htmlentities($str, ENT_QUOTES, "UTF-8");
}
```

Appinfo.php:

```
<?php

/**
 * Den här klassen har statiska metoder för att hämta information om
 * applikationen
 */

class AppInfo {

    /**
     * @return appID för applikationen
     */
}
```

```

*/

public static function appID() {

    return getenv('FACEBOOK_APP_ID');

}

/**

* @return appSecret för applikationen

*/

public static function appSecret() {

    return getenv('FACEBOOK_SECRET');

}

/**

* @return adressen

*/

public static function getUrl($path = '/') {

    if (isset($_SERVER['HTTPS']) && ($_SERVER['HTTPS'] == 'on' ||
$_SERVER['HTTPS'] == 1)

        || isset($_SERVER['HTTP_X_FORWARDED_PROTO']) &&
$_SERVER['HTTP_X_FORWARDED_PROTO'] == 'https'

    ) {

        $protocol = 'https://';

    }

    else {

        $protocol = 'http://';

    }

}

```

```
    return $protocol . $_SERVER['HTTP_HOST'] . $path;
}

}
```

Channel.html:

```
<!-- importerar Facebook JavaScript SDK biblioteket -->
<script src="//connect.facebook.net/en_US/all.js"></script>
```

Base.css:

```
/*
    title: base css
*/

/* =global elements
----- */

body {
    color: #999;
    font: 14px/1.5em "Lucida Grande", Arial, sans-serif;
    -webkit-font-smoothing: antialiased;
    width: 720px;
    margin: 20px;
}
```

```
a:link, a:visited {
    color: #3b5998;
    font-weight: bold;
    text-decoration: none;
}

a:hover, a:active {
    text-decoration: underline;
}

h1 {
    color: #656565;
    font-family: "Helvetica Neue", Arial, sans-serif;
    font-size: 18px;
    font-weight: normal;
}

/* ----- */

/* =header
----- */

header {
    border-left: 1px solid #b3b3b3;
    border-right: 1px solid #b3b3b3;
```

```
border-top: 1px solid #b3b3b3;

font-size: 11px;

padding: 24px 18px;
}

header h1 {

border-bottom: 2px solid #e9e9e9;

margin: 0 0 9px;

padding: 2px 0 8px;
}

header #picture + div {

margin-left: 82px;
}

header #picture {

background-position: center 25%;

background-repeat: no-repeat;

background-size: 64px;

float: left;

margin-right: 18px;

width: 64px;

height: 64px;

overflow: hidden;
}
```



```
header .tagline {  
    float: left;  
}
```

```
#share-app {  
    float: right;  
    width: 344px;  
}
```

```
#share-app ul {  
    float: right;  
}
```

```
#share-app p {  
    float: left;  
    margin-left: 8px;  
}
```

```
#share-app li {  
    float: left;  
    list-style: none;  
    margin-left: 8px;  
}
```

```
/* ----- */
```

```
/* =buttons
----- */

.facebook-button:link, .facebook-button:visited {

    background-color: #eee;

    background-image: -moz-linear-gradient(top, #fdfdfd, #eaebeb);

    background-image: -ms-linear-gradient(top, #fdfdfd, #eaebeb);

    background-image: -o-linear-gradient(top, #fdfdfd, #eaebeb);

    background-image: -webkit-gradient(linear, left top, left bottom,
from(#fdfdfd), to(#eaebeb));

    background-image: -webkit-linear-gradient(top, #fdfdfd, #eaebeb);

    background-image: linear-gradient(top, #fdfdfd, #eaebeb);

    border: 1px solid #a9a9a9;

    -moz-box-shadow: 0 1px 0 rgba(0, 0, 0, .1);

    -webkit-box-shadow: 0 1px 0 rgba(0, 0, 0, .1);

    box-shadow: 0 1px 0 rgba(0, 0, 0, .1);

    color: #4f4f4f;

    display: -moz-inline-box;

    display: inline-block;

    font-family: 'Lucida Grande', Tahoma, Verdana, Arial, sans-serif;

    font-size: 11px;

    font-weight: bold;

    line-height: 20px;

    margin: 0;

    outline: none;

    text-decoration: none;
```

```
text-shadow: 0 1px 1px rgba(255, 255, 255, .33);
}

.facebook-button span {
  background: url("../images/sprites.png") no-repeat;
  padding: 0 7px 0 21px;
}

.facebook-button .speech-bubble {
  background-position: 6px 3px;
}

.facebook-button .plus {
  background-position: 6px -17px;
}

.facebook-button .apprequests {
  background-position: 6px -37px;
}

.button:link, .button:visited {
  background-color: #7a90c0;
  background-image: -moz-linear-gradient(top, #97afe2, #6073a1);
  background-image: -ms-linear-gradient(top, #97afe2, #6073a1);
  background-image: -o-linear-gradient(top, #97afe2, #6073a1);
```

```
background-image: -webkit-gradient(linear, left top, left bottom,
from(#97afe2), to(#6073a1));

background-image: -webkit-linear-gradient(top, #97afe2, #6073a1);

background-image: linear-gradient(top, #97afe2, #6073a1);

border: none;

-moz-border-radius: 30px;

-webkit-border-radius: 30px;

border-radius: 30px;

-moz-box-shadow: 0 1px 1px rgba(0, 0, 0, .5), 0 1px 1px rgba(255, 255,
255, .33) inset;

-webkit-box-shadow: 0 1px 1px rgba(0, 0, 0, .5), 0 1px 1px rgba(255,
255, 255, .33) inset;

box-shadow: 0 1px 1px rgba(0, 0, 0, .5), 0 1px 1px rgba(255, 255, 255,
.33) inset;

color: #1d222c;

display: inline-block;

font: bold 14px "Helvetica Neue", Helvetica, Arial, Geneva, sans-serif;

line-height: 32px;

padding: 0 25px;

text-decoration: none;

text-shadow: 0 1px 1px rgba(255, 255, 255, .2);

}

/* ----- */

/* =get started

----- */

#get-started {
```

```
background: url("../images/get-started-bg.jpg");

border: 1px solid #353d4f;

height: 105px;

text-align: center;

text-shadow: 0 1px 0 rgba(0, 0, 0, .29);

padding: 32px 0 0;

}

#get-started p {

    color: #fff;

    font-family: "CorbelRegular";

    font-size: 24px;

    height: 23px;

    margin-bottom: 24px;

}

#get-started p span {

    font-family: "HybreaRegular";

    font-size: 22px;

}

/* ----- */

/* =guides/samples spacing

----- */

#guides h1,
```

```
#samples h1 {
    margin-bottom: 15px;
    text-align: center;
}

#guides, #samples {
    padding: 20px 5px 20px 10px;
}

#guides ul li,
#samples .list {
    float: left;
    margin-right: 3px;
    width: 172px;
}

#guides ul li p,
#samples ul {
    padding: 0 4px;
}

/* ----- */

/* =guides
----- */

#guides {
```

```
border-bottom: 1px solid #b3b3b3;

border-left: 1px solid #b3b3b3;

border-right: 1px solid #b3b3b3;

}

#guides ul li {

    font-size: 11px;

    list-style: none;

}

#guides ul li p {

    -moz-hyphens: auto;

    -webkit-hyphens: auto;

    line-height: 20px;

    text-align: justify;

}

#guides ul li a.icon {

    background: url("../images/sprites.png") no-repeat;

    display: block;

    font-family: "Helvetica Neue", Arial, sans-serif;

    font-size: 14px;

    margin-bottom: 10px;

    padding-top: 124px;

    text-align: center;
```

```
}

#guides ul li a.websites {
    background-position: 35px -87px;
}

#guides ul li a.mobile-apps {
    background-position: 35px -231px;
}

#guides ul li a.apps-on-facebook {
    background-position: 35px -375px;
}

#guides ul li a.heroku {
    background-position: 35px -519px;
}

/* ----- */

/* =samples
----- */

#samples {
    background-image: url("../images/samples-bg.png");
    border-bottom: 1px solid #dadada;
    border-left: 1px solid #b3b3b3;
```



```
border-right: 1px solid #b3b3b3;

border-top: 1px solid #fff;

}

#samples h3 {

background: url("../images/header-sketch.png") no-repeat;

color: #646464;

font-weight: bold;

font-size: 11px;

line-height: 29px;

text-align: center;

}

#samples ul {

background: url("../images/picture-sketch.png") 4px 8px;

background-repeat: repeat-y;

height: 161px;

overflow: hidden;

padding-top: 8px;

}

#samples li {

background-repeat: no-repeat;

background-position: 0 0;

color: #7c7c7c;
```

```
display: block;

font-size: 11px;

font-weight: bold;

height: 32px;

margin-bottom: 11px;

overflow: hidden;

position: relative;

}
```

```
#samples li a:link,
#samples li a:visited {

color: #7c7c7c;

display: block;

line-height: 32px;

padding-left: 36px;

}
```

```
#samples .list img {

height: auto;

left: 0;

position: absolute;

top: 0;

width: 32px;

}
```

```
#samples .inline ul {  
    background-repeat: repeat;  
}  
  
#samples .inline li {  
    float: left;  
    margin-left: 12px;  
    overflow: hidden;  
    padding: 0;  
    width: 32px;  
    background-position: center 25%;  
    background-repeat: no-repeat;  
    -moz-background-size: cover;  
    background-size: cover;  
}  
  
#samples .inline li.first-column {  
    margin-left: 0;  
}  
  
#samples .inline li a:link,  
#samples .inline li a:visited {  
    display: block;  
    width: 32px;  
    height: 32px;
```

```
overflow: hidden;

text-indent: -9999px;

}

/* ----- */

/* =clearfix
----- */

.clearfix:after {

clear: both;

content: ".";

display: block;

font-size: 0;

height: 0;

line-height: 0;

visibility: hidden;

}

.clearfix {

zoom: 1;

}
```

Mobile.css:

```
body {  
    margin: 10px;  
    width: auto;  
}  
  
#guides ul li,  
#samples .list {  
    float: none;  
    margin: 0 auto 10px;  
    width: 172px;  
}  
  
#share-app li {  
    margin: 0 8px 4px 0;  
}  
  
#share-app {  
    width: auto;  
}
```

Reset.css:

```
/*  
    title: reset css  
*/
```

```
html, body, div, span, applet, object, iframe,  
h1, h2, h3, h4, h5, h6, p, blockquote, pre,  
a, abbr, acronym, address, big, cite, code,  
del, dfn, em, img, ins, kbd, q, s, samp,  
small, strike, strong, sub, sup, tt, var,  
b, u, i, center,  
dl, dt, dd, ol, ul, li,  
fieldset, form, label, legend,  
table, caption, tbody, tfoot, thead, tr, th, td,  
article, aside, canvas, details, embed,  
figure, figcaption, footer, header, hgroup,  
menu, nav, output, ruby, section, summary,  
time, mark, audio, video {  
  
    border: 0;  
  
    margin: 0;  
  
    padding: 0;  
  
    font: inherit;  
  
    font-size: 100%;  
  
    vertical-align: baseline;  
  
}  
  
/* HTML5 display-role reset for older browsers */  
  
article, aside, details, figcaption, figure,
```

```
footer, header, hgroup, menu, nav, section {  
    display: block;  
}  
  
body {  
    line-height: 1;  
}  
  
ol, ul {  
    list-style: none;  
}  
  
blockquote, q {  
    quotes: none;  
}  
  
blockquote:before, blockquote:after, q:before, q:after {  
    content: '';  
    content: none;  
}  
  
table {  
    border-collapse: collapse;  
    border-spacing: 0;  
}
```

Screen.css:

```
@import url("https://statics.herokuapp.com/fonts/fonts.css");  
  
@import url("reset.css");  
  
@import url("base.css");
```