



Elastic Deployment, Delivery, & Management Solution for Web Applications

Facebook Developer Starter Guide

October 2008

About this Guide

This Guide contains information about developing an application for Facebook, deploying the application on a Morph AppSpace, and adding the application to Facebook.

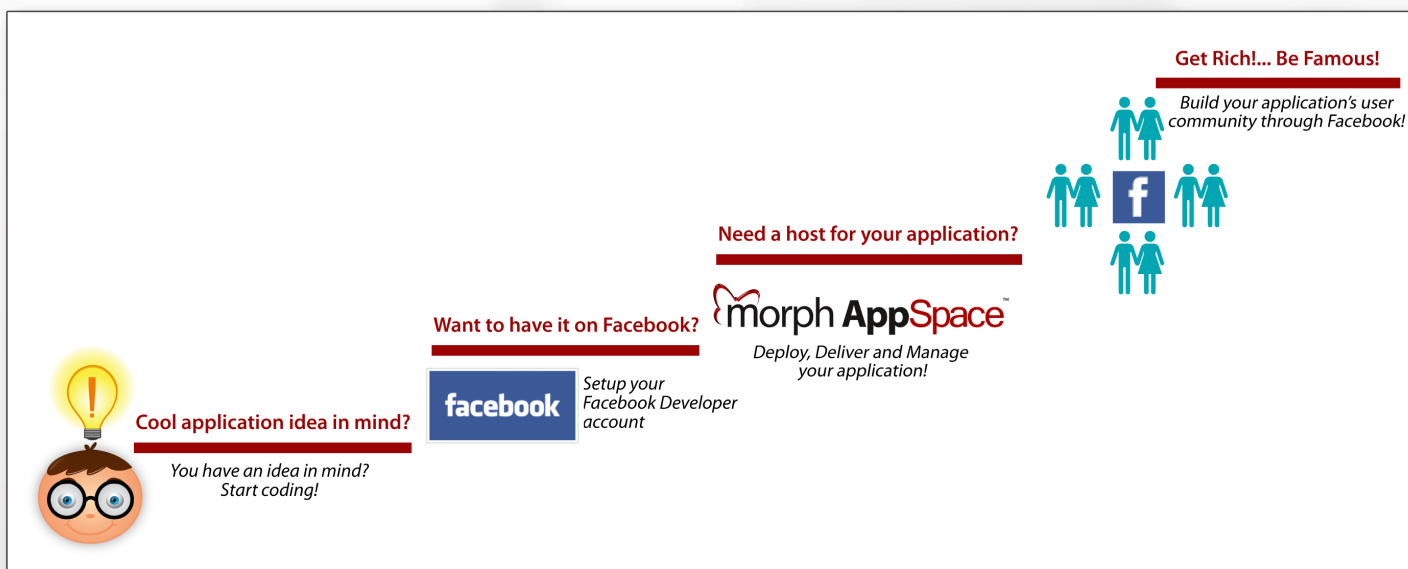
Overview

Facebook is a social networking Web site, which allows free access to users wanting to join one or more networking groups such as schools, companies, regions, or country of origin. Aside from being a social network, Facebook launched the Facebook platform that provides a framework for software developers to create applications that interact with core Facebook features.

A software developer who wants to deploy an application on the Facebook platform needs two things: an application and a host. Your application must run independently on a host. However, ordinary hosting schemes might not serve your application effectively. As your application requirements change, you have to think about load balancing, high availability, backup, security, redundancy, and a host of other related delivery and management things.

Morph AppSpace, an application delivery environment that runs within the Morph Application Platform¹, provides you with a complete deployment, delivery, and management system for your Facebook application.

If you're excited to create your first Facebook application, read the following sections. You will learn how to develop a Facebook application using Ruby on Rails, how to deploy the application on the Morph AppSpace, and how to add the application to Facebook.

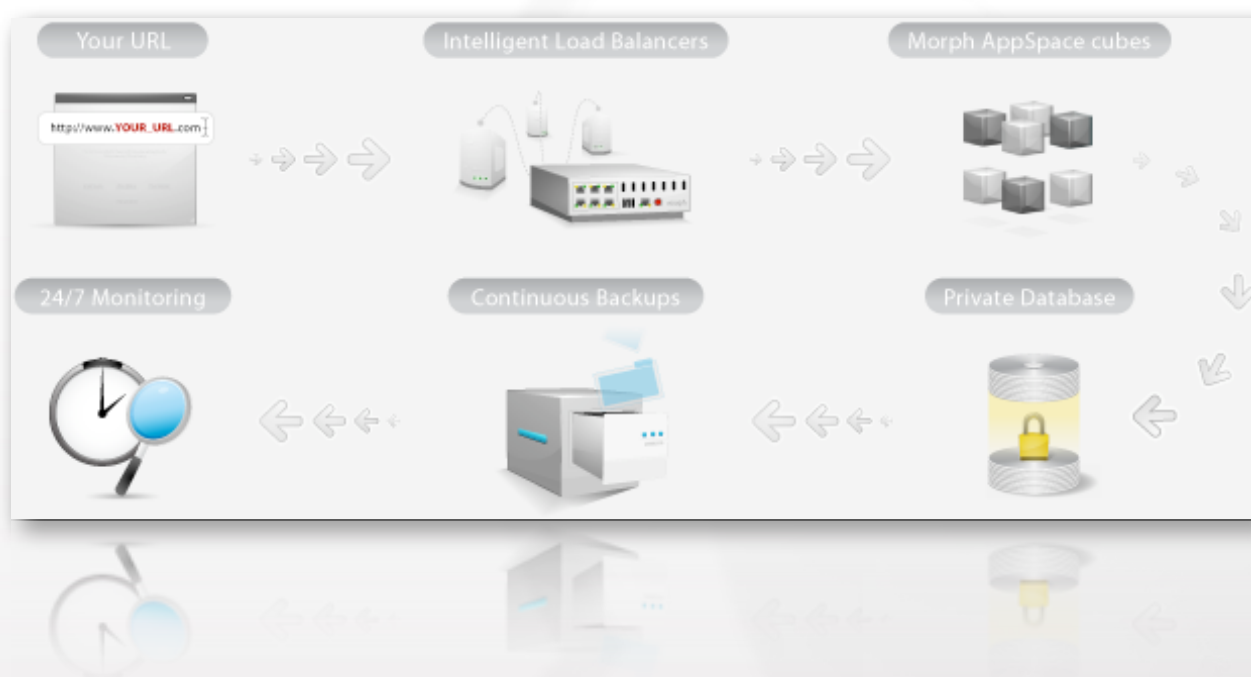


¹ The Morph Application Platform provides an end-to-end deployment and management environment for Web applications that includes load balancing, high availability infrastructure, distributed computing, virtualized web stacks, databases, managed backups, and monitoring. It also provides scalable resources and an environment that grow or shrink as demanded by utilization, allowing developers to support customers yet avoid over-provisioning.

Introducing the Morph AppSpace

The Morph AppSpace provides a complete deployment, delivery, and management system for Web applications. It runs within the Morph Application Platform, equipping you with a fault-tolerant environment that you can scale as your requirements change.

The Morph AppSpace is complete with load balancing, high-availability infrastructure, Morph AppSpace cube², integrated database, managed backups, monitoring, and self repair. All these free you from the burden of installing and configuring servers. You only need to focus on building your application and deploying it on a Morph AppSpace. Our well-experienced team will take care of the rest. Subscriptions for the Morph AppSpace start at USD 1 a day for an entry level Web application. For more information, visit www.mor.ph.



² A Morph AppSpace cube is a virtualized application compute environment that is managed by the Morph Application Platform and part of each Morph AppSpace subscription. Each Morph AppSpace cube is a self-contained environment for running a specific application. The Morph AppSpace cube for Rails includes Mongrel and Nginx.

Let's take a look at the technology behind the Morph AppSpace:

Deployment

When you deploy your application to the Morph AppSpace, your code is securely transferred to Amazon S3. Our system fetches the new upload and deploys it automatically to your Morph AppSpace subscription's application servers. If some migrations are not applied, we run them for you so that your database gets automatically updated. Within minutes, your application is up and running. We even keep multiple versions of your deployed application, just in case you need to revert to a previous version.

Delivery

The Morph Application Platform's fault-tolerant environment provides an end-to-end enterprise-quality solution. Intelligent load balancers ensure that incoming traffic is properly routed to the application server with the least load. Firewalls protect the environment. Web delivery stacks and application servers are specifically tuned to optimize the performance of applications. Our system administrators constantly monitor, update, patch, and manage each component of the infrastructure. We also created a special library that allows you to leverage a fast cache system directly from Amazon S3. Our multi-tenant API allows your applications to support multiple themes and data segregation.

Management

The Morph Application Platform streamlines the way you manage your application. Our automated monitoring system keeps an eye on the servers, services, and applications, and alerts us to any potential issues. As a developer, you can view real-time statistics for your application, including CPU and memory utilization. In a click of a button, you can increase your application's resources to meet the needs of your visitors. You can directly manage your database, view your application's logs, and execute rake tasks. You totally have control of your application.

Developing a Facebook Application using Ruby on Rails

In this section, we will create a simple Rails application without any Facebook tools and references.



Note: The following steps assume that you have prior knowledge and experience in using Ruby on Rails. It is also assumed that you have installed Ruby on Rails on your machine.

As an example, let's develop the MakePoem application. The MakePoem application is a collaborative poem writing application. This application allows Facebook users to add the application, create a poem project that invites friends to write, invites other Facebook users to add the application, and so on.

The following lists the easy steps to develop the MakePoem application:

1. In the command console, type the following:

```
$ rails makepoem
```

```
$ cd makepoem
```

2. Create models for the application. Type `ruby script/generate model poem`.
3. Create the application's database. If you want to use PostgreSQL, type the following in the command console:

```
$ psql -U postgres
```

```
=# create database makepoem;
```

4. Configure `database.yml` to point to the database. If you use PostgreSQL, you can follow this example:

```
production:
```

```
  adapter: postgresql
```

```
  database: makepoem
```

```
  username: postgres
```

```
  host: localhost
```

5. Create migrations for the database. Execute migrations by typing the following:

```
$ rake db:migrate
```

6. Let's create the application's controller. Type this:

```
$ ruby script/generate controller home
```

7. We need to change the routes of the application. To do this, access `config/routes.rb` and add these lines:

```
map.connect '', :controller => 'home', :action => 'index'  
# Install the default routes as the lowest priority.  
map.connect ':controller/:action/:id'  
map.connect ':controller/:action/:id.:format'
```

8. Let's add the view for the application. In the `app/views/home` directory, create the `index.rhtml` file.

```
<html>  
  <head>  
    <title>MakePoem Home</title>  
  </head>  
  <body>  
    <h1>MakePoem</h1>  
    <h2>The MakePoem application is a collaborative poem writing  
    application. This application allows Facebook users to add the  
    application, create a poem project  
    that invites friends to write, and invites other  
    Facebook users to add the application.</h2>  
  </body>  
</html>
```

9. Delete the `index.html` file from the public directory.
10. In the command console, type `$ script/server`.
11. Access <http://0.0.0.0:3000/>. You will see an application such as this:

MakePoem

The MakePoem application is a collaborative poem writing application. This application allows Facebook users to add the application, create a poem project that invites friends to write and invites other Facebook users to add the application.

Adding the Facebook Gems on the Rails application

Since we now have a running Rails application, let's "Facebook" it. We will use the **facebooker (0.9.5)** gem and its pre-requisite, the **json (1.1.2)** gem.

Perform the following steps:

1. Let's install the gem needed for the development of the application. In the command console, type:

```
$ sudo gem install facebooker
$ cd makepoem
$ script/plugin install http://facebooker.rubyforge.org/svn/trunk/facebooker
```



Note: You might also need to install the json(1.1.2) gem.

2. Set up your Facebook Developer account. Go to <http://www.facebook.com/developers/>.



Note: This assumes that you have already signed up a Facebook account.

3. From your Facebook Developer application, click **Set Up New Application**. This displays:

Required Fields

Application Name (Limit: 50 characters)

Optional Fields

Base Options

Developer Contact E-mail (Limit: 100 characters)
For Facebook correspondence only. We will contact you at this address if there are any problems or important updates.

User Support E-mail (Limit: 100 characters)
Messages sent from users on your application's help page will be sent to this address.

Callback Url (Limit: 100 characters)
After logging into Facebook, users are redirected to the callback URL. See [authentication overview](#) for more details.

Canvas Page URL
☒ Use FBML ☐ Use iframe
 Your application will be viewable in the Facebook navigation at this URL - either as rendered FBML or loaded in an iframe. If you aren't sure what you should use here, choose FBML. You can use iframes within FBML on canvas pages with the <fb:iframe> tag, and most things you will want to do will be easier and faster with FBML.

Application Type ☒ Website ☐ Desktop

Mobile Integration ☒ My application uses the mobile platform
Checking this will enable some additional SMS functionality for applications.

IP Addresses of Servers Making Requests (comma-separated)
If you supply this information (e.g. 10.1.20.1, 10.1.20.3), requests from addresses other than those listed will be rejected.

- Type the **Application Name**. Base on the example, enter **MakePoem**.
- Type the **Callback URL**. This is the base URL of your server or host.

- Type the **Canvas Page URL**. Users of the application will see this path on their browser.



Note: After setting up your application, Facebook automatically generates the API key and the Secret key. Click on your application to see the generated keys.

4. Set up the facebooker environment via `RAILS_ROOT/config/facebooker.yml`. Enter the **Callback URL** and **Canvas Page Name** that you specified on your Facebook application settings.

```
# File: config/facebooker-mock.yml
# The api key, secret key, and canvas page name are required to
# get started
# Your callback url in Facebook should be set to http://
public\_host:public\_port
development:
  api_key: 1234567891011121314151617181920a
  secret_key: XXXXXXXXXXXXXXXX
  canvas_page_name: makepoem
  callback_url: http://makepoem.morphexchange.com
test:
  api_key: 1234567891011121314151617181920a
  secret_key: XXXXXXXXXXXXXXXX
  canvas_page_name: makepoem
production:
  api_key: 1234567891011121314151617181920a
  secret_key: XXXXXXXXXXXXXXXX
  canvas_page_name: makepoem
```


5. Open the `home_controller.rb` file. Add this line:

```
'ensure_authenticated_to_facebook'
```



Note: This line adds a `before_filter` to the controller.

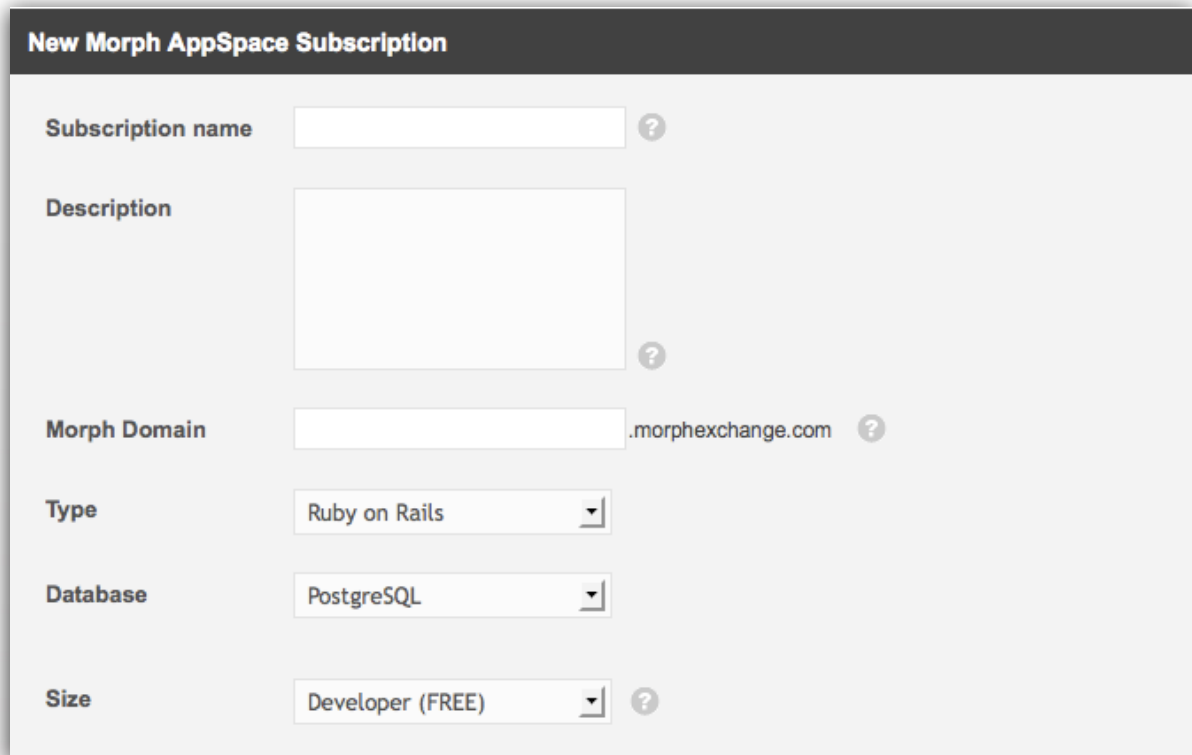
6. Check in the MakePoem application's code to a source code control repository such as SVN.

Deploying the MakePoem application on the Morph AppSpace

The MakePoem application is ready for Facebook. Let's deploy and run the MakePoem application on the Morph AppSpace.

Follow these easy steps:

1. Login to your Morph Control Panel account. If you do not have an account, visit www.mor.ph.
2. From your Morph Control Panel **Subscription** page, create a new subscription by clicking **New Subscription**.



The screenshot shows a web form titled "New Morph AppSpace Subscription". The form contains the following fields and options:

- Subscription name**: A text input field with a question mark icon to its right.
- Description**: A large text area with a question mark icon to its right.
- Morph Domain**: A text input field followed by ".morphexchange.com" and a question mark icon.
- Type**: A dropdown menu with "Ruby on Rails" selected.
- Database**: A dropdown menu with "PostgreSQL" selected.
- Size**: A dropdown menu with "Developer (FREE)" selected and a question mark icon to its right.

3. Fill in the required fields. The **Subscription Name** must be unique.
4. Click **Save** to create your new Morph AppSpace subscription.

5. Select your newly added subscription widget. If your application needs a database, click **Create Database**.



6. Download the Capistrano file from the subscription widget. Save this file to your application's root directory.



Note: We recommend that you commit the Capistrano file to your SVN repository or to the version control system you are using.

7. Use a text editor to open the `morph_deploy.rb` file. Replace the `set :repository` line with your version control system.

Example: `set :repository, 'https://dev.myapp.com/repos/ruby/trunk/testing/makepoem'`

8. Upload and deploy your application on the Morph AppSpace. In the command console, type `cap -f morph_deploy.rb morph:deploy` (if you saved `morph_deploy.rb` in another directory, change this command accordingly).



Note: If you are using a version control system other than Subversion, perform the following:

* If you're using git:

```
cap -f morph_deploy.rb -s scm=git -s deploy_via=checkout
morph:deploy
```

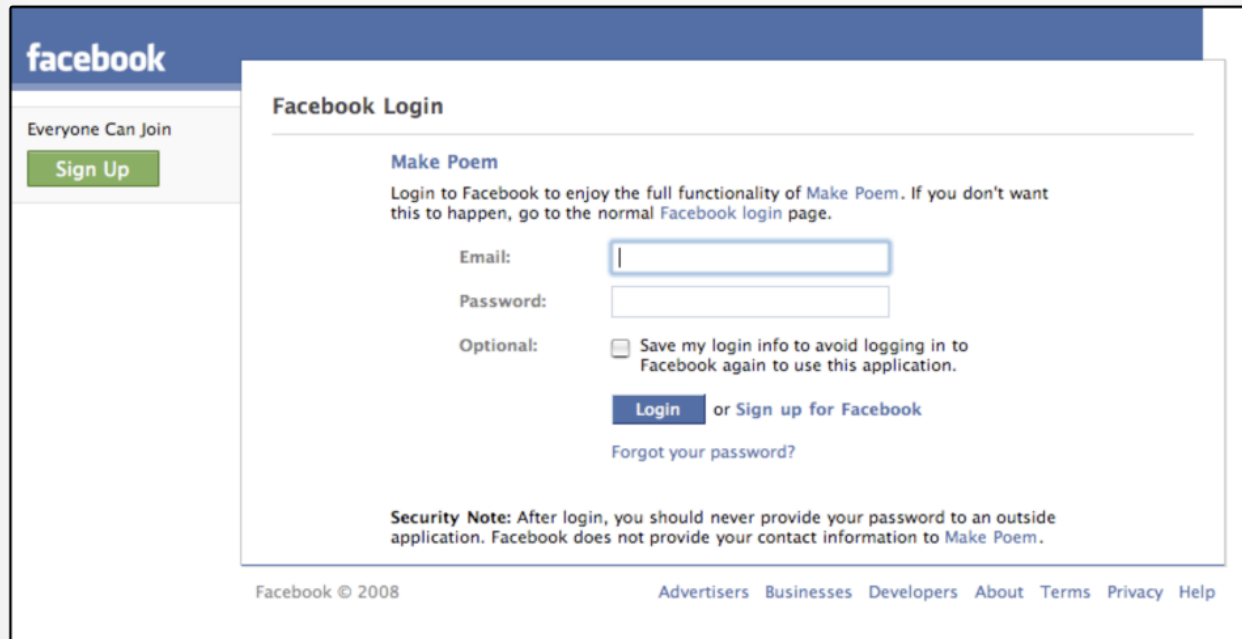
* If you're using Mercurial with a local repository:

```
cap -f morph_deploy.rb -s scm=mercurial -s deploy_via=checkout
-s repository=. morph:deploy
```

* If you're using a local Bazaar repository:

```
cap -f morph_deploy.rb -s scm=bzr -s repository='' morph:deploy
```

9. Once the command has finished, wait for a few minutes (while your application is being initialized) and then in a browser go to <your_app>.morphexchange.com. The window displays this:



The screenshot shows the Facebook Login interface for an application named 'Make Poem'. On the left, there is a sidebar with the Facebook logo and a 'Sign Up' button. The main content area is titled 'Facebook Login' and includes a sub-header 'Make Poem'. Below this, a message states: 'Login to Facebook to enjoy the full functionality of Make Poem. If you don't want this to happen, go to the normal Facebook login page.' The login form contains fields for 'Email:' and 'Password:', followed by an 'Optional:' section with a checkbox for 'Save my login info to avoid logging in to Facebook again to use this application.' Below the form are buttons for 'Login' and 'or Sign up for Facebook', along with a link for 'Forgot your password?'. A 'Security Note' at the bottom advises users not to provide their password to outside applications. The footer includes 'Facebook © 2008' and a list of links: 'Advertisers', 'Businesses', 'Developers', 'About', 'Terms', 'Privacy', and 'Help'.

For more information about the Morph AppSpace, refer to the Rails Developers' Cookbook.



Morph Labs is the leading provider of Platform as a Service (PaaS) that virtualizes the application environment through the use of open source technologies to simplify the deployment, delivery, and management of web based applications.

Morph Labs uses virtual infrastructures including Amazon Web Services to provide a truly elastic environment for web applications that can be instantly provisioned and seamlessly scaled.

Morph Labs is a global company with headquarters in Cebu City, Philippines with additional in-country operations in Manila along with Los Angeles, California and Portland, Oregon in the U.S.A.

www.mor.ph

World Headquarters

Unit 1A, Asiatown IT Park,
Lahug, Cebu City 6000
Philippines

Worldwide Inquiries

info@mor.ph
www.mor.ph

Copyright 2008, Morphlabs, Inc. All rights reserved.
This document is provided for information purposes only
and the contents herein are subject to change without notice.
This document is not warranted to be error-free, nor subject
to any other warranties or conditions, whether expressed orally
or implied in law, including implied warranties and conditions of
merchantability or fitness for a particular purpose. We disclaim
any liability with respect to this document and no contractual
obligations are formed either directly or indirectly by this document.
Morph, Morph Application Platform, Morph AppSpace, and
Simplify Innovation are registered trademarks of Morphlabs, Inc.
Other names may be trademarks of their respective owners.