



Incredible Animated Ajax Based Solutins

**PREMIUM WEBSITE TEMPLATES
SCRIPTS & WORDPRESS PLUGINS**



Employees

Usage Documentation

Premium Facebook APP by iDangero.us

Thank you for purchasing the this item! If you have any questions that are beyond the scope of this help file, please please feel free to contact us via our support ticket form [here](#).

Table of Contents

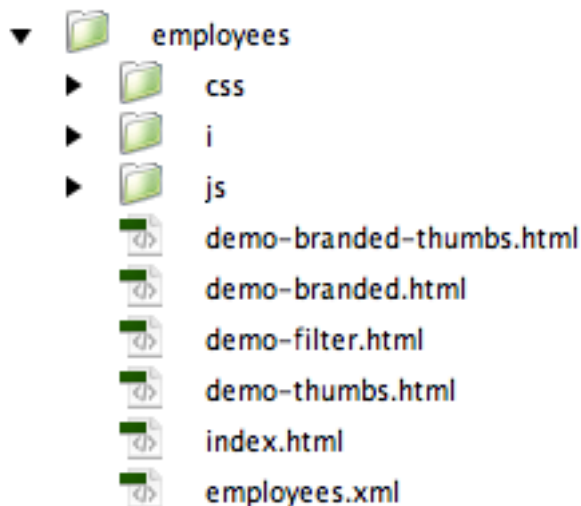
1. About Employees	3
2. Employees Files & Folders Structure	3
3. Installation.....	3
4. employees.xml	3
5. HTML Structure.....	4
5.1. Main Layout	4
5.2. Pages Structure	7
5.3. Filter Structure	8
5.4. CSS Files	9
5.5. JavaScripts	10
5.6. App Configuration.....	10
6. HTML Template System and XML Parser	11
6.1. Person Template	11
6.2. Popup Template	14
7. PSD Files & Structure	15
8. Set Up Facebook App	16
9. Licensing Terms	18

1. About Employees

Employees is the premium Ajax based HTML5 Facebook App developed by iDangero.us. It is designed for usage as Facebook page tab.

2. Employees Files & Folders Structure

Employees App files are located in the **employees/** folder from the downloaded archive:



- **css**: folder with a .css style files
- **i**: here you can find layouts images, icons, avatars used in this Template.
- **js**: JavaScripts scripts used in this template
- main **index.html** in the root folder with default HTML layout
- demo-*.html files with demos with different layouts and features
- employees.xml - xml formatted file with information about persons

3. Installation

To install Employees App (to server, not to FB tab) you have to copy all files & folders from the **employees/** folder to your web-site root folder on the server (It could be a real server or local testing server).

4. employees.xml

First of all, let's look at the structure of employees.xml file to learn how to configure and add new persons:

```
<!--Beginning of the XML file: -->
<?xml version="1.0" encoding="UTF-8"?>
<!--Container with all persons: -->
<employees>
  <!-- First person with id (identificator) "vk": -->
  <person id="vk">
    <!-- Name: -->
    <name>Vladimir Kharlampidi</name>
    <!-- Position: -->
    <position>CEO & Founder</position>
```

```
<!-- Path to avatar image. Should be 150x150 pixels: -->
<avatar>i/demo/avatar-1.jpg</avatar>
<!-- Other properties: -->
<department>Managment, IT</department>
<slogan>"Be yourself, everyone else is already taken!"</slogan>
<born>03.06.1986</born>
<worksince>May, 2010</worksince>
<likes>tasty food, good cars, girls, life, iDangero.us</likes>
<lives>Rostov-na-Donu, Russia</lives>
<website>http://www.idangero.us</website>
<!-- Links to social profiles:-->
<twitter>http://twitter.com/idangerous/</twitter>
<facebook>http://facebook.com/idangero.us/</facebook>
<linkedin>http://linkedin.com/</linkedin>
<!--Bio-->
<bio><![CDATA[
    <h1>About Vladimir</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras
non tortor et lacus lacinia condimentum. Nunc non turpis nulla. Pellentesque
habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.
Ut congue velit id ligula cursus facilisis luctus nisi dapibus.</p>
]]></bio>
</person>
<!-- Other persons goes here -->
<!-- .... --->
</employees>
```

As you can see the XML structure is pretty simple and self-explaining. Only one important point. As you can see the person's biography content (<bio>...</bio>) contains HTML code with heading (<h1>) and paragraphs(<p>). In this case you should wrap any HTML content inside of XML tags with CDATA comment like:

```
<bio><![CDATA[ here goes any HTML content ]]></bio>
```

It is very important and if you will not wrap HTML with CDATA, than XML parser just will not work!

5. HTML Structure

5.1. Main Layout

Let's look at the default HTML structure from index.html file:

```
<!doctype html>
<!--[if lt IE 7]> <html class="no-js lt-ie9 lt-ie8 lt-ie7" lang="en"> <![endif]-->
<!--[if IE 7]> <html class="no-js lt-ie9 lt-ie8" lang="en"> <![endif]-->
<!--[if IE 8]> <html class="no-js lt-ie9" lang="en"> <![endif]-->
<!--[if gt IE 8]><!--> <html class="no-js" lang="en"> <!--<![endif]-->
<head>
  <meta charset="utf-8">
```

```
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
<title>Employees</title>
<meta name="description" content="">
<link rel="stylesheet" href="css/employees.css">
<script src="js/libs/modernizr-2.5.3.min.js"></script>
</head>
<body>
  <div class="wrapper">
    <header>
      <!-- LOGO -->
      <div class="logo"></div>
    </header>

    <!-- Navigation Menu -->
    <nav>
      <span>Departments:</span>
      <a href="#" data-page="management" class="active">Management</a>
      <a href="#" data-page="acc">Accounting</a>
      <a href="#" data-page="advert">Advertising</a>
      <a href="#" data-page="it">IT</a>
      <a href="#" data-page="seo">SEO</a>
      <div class="clearfix"></div>
    </nav>

    <div class="content">
      <!-- Page with employees of Management Department -->
      <div id="management" class="page active-page">
        <!-- Page Title -->
        <h1 class="page-title">Management Department</h1>
        <div class="page-content">
          <!-- HTML description of Management Department -->
        </div>
        <!-- Employees of Management Department -->
        <div class="person" data-id="vk"></div>
```

```
<!--
```

```
<div class="person" data-id="vk"></div>
```

This means that inside of this div will be inserted HTML content with details of person which has id equal to "vk" in the employees.xml file. Look for chapter 6 to know about HTML output for each person

```
-->
```

```
<div class="person person-last" data-id="aa"></div>
<div class="clearfix"></div>
<div class="person" data-id="ee"></div>
<div class="person person-last" data-id="veekey"></div>
<div class="clearfix"></div>
</div>
<!-- Page with employees of Accounting Department -->
<div id="acc" class="page">
  <!-- ... -->
</div>
<!-- Page with employees of Advertising Department -->
<div id="advert" class="page">
```

```
<!-- ... -->
</div>
<!-- Page with employees of IT Department -->
<div id="it" class="page">
  <!-- ... -->
</div>

<!-- Page with employees of SEO Department -->
<div id="seo" class="page">
  <!-- ... -->
</div>
</div>

<!-- Footer With Copyrights -->
<footer>&copy; 2012 "Employees" Premium Facebook App by <a href="http://
www.idangero.us" target="_blank">iDangero.us</a>. All Rights Reserved.</footer>
</div>

<!-- Person template. Look For Chapter 6 -->
<div class="person-template" style="display: none;">
  <div class="person-mask"></div>
  <div class="person-avatar">
    <img data-attr="{ 'src': 'avatar' }">
    <div class="avatar-mask"></div>
  </div>
  <div class="person-info">
    <p class="person-name" data-insert="name"></p>
    <p class="person-position" data-insert="position"></p>
    <div class="person-slogan">
      <div class="person-slogan-angle"></div>
      <span data-replace="slogan"></span>
    </div>
  </div>
  <a class="open-popup learn-more" href="#"></a>
  <div class="clearfix"></div>
</div>
<!-- End Person Template -->

<!-- Popup. Template. Look For Chapter 6 -->
<div class="popup-overlay"></div>
<div class="popup-view">
  <div class="popup">
    <div class="popup-content">
      <a href="#" class="close-popup"></a>
      <div class="popup-left">
        <div class="popup-avatar">
          <img data-attr="{ 'src': 'avatar' }">
        </div>
        <div class="popup-links">
          <p data-check="twitter"><a class="external" data-
attr="{ 'href': 'twitter' }" target="_blank"><span data-replace="firstname"></span>
on Twitter</a></p>
          <p data-check="facebook"><a class="external" data-
attr="{ 'href': 'facebook' }" target="_blank">Facebook Page</a></p>
          <p data-check="linkedin"><a class="external" data-
attr="{ 'href': 'linkedin' }" target="_blank">LinkedIn Profile</a></p>
          <p><a href="#" class="open-popup-content bordered"><span data-
replace="firstname"></span>'s Bio</a></p>
        </div>
      </div>
    </div>
  </div>
</div>
```



```

        <div class="popup-info">
            <h2 class="popup-name" data-insert="name"></h2>
            <h3 class="popup-position" data-insert="position"></h3>
            <ul class="info-list">
                <li data-check="department"><i>Department:</i> <b data-
insert="department"></b></li>
                <li data-check="worksince"><i>Work since:</i> <b data-
insert="worksince"></b></li>
                <li data-check="slogan"><i>Slogan:</i> <b data-
insert="slogan"></b></li>
                <li data-check="likes"><i>Likes:</i> <b data-
insert="likes"></b></li>
                <li data-check="lives"><i>Lives in:</i> <b data-
insert="lives"></b></li>
                <li data-check="born"><i>Born date:</i> <b data-
insert="born"></b></li>
                <li data-check="website"><i>Website:</i> <b><a
class="external" data-insert="website" data-attr="{ 'href': 'website' }"
target="_blank"></a></b></li>
            </ul>
        </div>
        <div class="clearfix"></div>
        <div class="popup-loader" data-check="bio">
            <div class="popup-loader-content">
                <a class="close-popup-content" href="#"></a>
                <span data-replace="bio"></span>
            </div>
        </div>
    </div>
</div>
<!--End Popup -->

<!-- scripts -->
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js"></
script>
<script>window.jQuery || document.write('<script src="js/libs/
jquery-1.7.2.min.js"><\script>')</script>
<script src="js/plugins.js"></script>
<!-- File with configuration -->
<script src="js/config-default.js"></script>
<script src="js/employees.js"></script>
<!-- end scripts -->

</body>
</html>

```

This was default layout with "Pages" structure. Let's look more closely how to use "Pages" or "Filter" structure.

5.2. Pages Structure

Let's look more closely at "Pages" structure and its menu:

```

<!-- Navigation -->
<nav>
    <span>Departments:</span>
    <!-- Links to appropriate pages -->

```

```

<a href="#" data-page="management" class="active">Management</a>
<a href="#" data-page="acc">Accounting</a>
<a href="#" data-page="advert">Advertising</a>
<a href="#" data-page="it">IT</a>
<a href="#" data-page="seo">SEO</a>
<div class="clearfix"></div>
</nav>
<div class="content">
  <!-- Pages -->
  <div id="management" class="page active-page">
    <!-- Page Title -->
    <h1 class="page-title">Management Department</h1>
    <!-- Page Content With Any HTML Content inside -->
    <div class="page-content">
      <p>This is a department description. You can put here any HTML content,
images, videos etc.</p>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris malesuada
suscipit nulla, nec mattis sapien dignissim non. Etiam eu risus eget purus iaculis
vestibulum in a purus.</p>
      <p>Aliquam mattis venenatis sem, ac tristique dolor volutpat eget. In felis
arcu, tempus eu fringilla dignissim, mollis vitae massa</p>
    </div>
    <!-- Persons on this page -->
    <div class="person" data-id="vk"></div>
    <div class="person person-last" data-id="aa"></div>
    <div class="clearfix"></div>
    <div class="person" data-id="ee"></div>
    <div class="person person-last" data-id="veekey"></div>
    <div class="clearfix"></div>
  </div>
  <div id="acc" class="page">
    <!-- The same page structure -->
  </div>
  <div id="advert" class="page">
    <!-- The same page structure -->
  </div>
  <div id="it" class="page">
    <!-- The same page structure -->
  </div>
  <div id="seo" class="page">
    <!-- The same page structure -->
  </div>
</div>

```

As you can see layout is pretty simple. The few points you should to know:

- first link (<a>) in menu should have the **"active"** class
- **"data-page"** attribute of every link should be the same as the ID attribute of appropriate page
- each page (<div>) should have the **"page"** class and be the child of "content" div
- first page (<div class="page">) should have additional **"active-page"** class

5.3. Filter Structure

The "filter" structure is much more simpler, let's open **demo-filter.html** file:


```

<!-- Navigation -->
<nav>
  <span>Filter By Department:</span>
  <!-- Links to filter persons -->
  <a href="#" data-filter="*" class="active">All Departments</a>
  <a href="#" data-filter="management">Management</a>
  <a href="#" data-filter="acc">Accounting</a>
  <a href="#" data-filter="advert">Advertising</a>
  <a href="#" data-filter="it">IT</a>
  <a href="#" data-filter="seo">SEO</a>
  <div class="clearfix"></div>
</nav>
<div class="content">
  <div class="filter-content">
    <!-- Employees -->
    <div class="person" data-filter="management it" data-id="vk"></div>
    <div class="person" data-filter="management it" data-id="aa"></div>
    <div class="person" data-filter="management acc" data-id="ee"></div>
    <div class="person" data-filter="management it" data-id="veekey"></div>
    <div class="person" data-filter="advert" data-id="jb"></div>
    <div class="person" data-filter="advert" data-id="js"></div>
    <div class="person" data-filter="it" data-id="jd"></div>
    <div class="person" data-filter="it" data-id="bd"></div>
    <div class="person" data-filter="it" data-id="arnold"></div>
    <div class="person" data-filter="seo" data-id="mk"></div>
    <div class="person" data-filter="seo" data-id="mm"></div>
  </div>
</div>

```

As you can see the filter works on the relevance of "data-filter" attributes. If you'll click on the link with a "data-filter" attribute equal to "management", script will show persons with have "management" in their "data-filter" attributes. Also note that employee can belong to several departments, in this case you can specify few values in his data-filter attribute divided with space.

See the chapter 5.6. for filter configuration!

5.4. CSS Files

There is one CSS file used in this App. You can find css attachment in the HEAD section of html file. By default it is located in the **css/** folder. You can easily change the path or filename in the href attribute of LINK tag.

employees.css - this file contains all general layouts for the template. It is separated into different sections. If you would like to edit a specific section of the site, simply find the appropriate section in the CSS file, and then scroll down until you find the appropriate style that needs to be edited.

5.5. JavaScripts

Employees template uses 5 JavaScript files

Attachment of JavaScript files can be found in the HEAD section and in the end of html file:

```
<script src="js/libs/modernizr-2.5.3.min.js"></script>
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js"></script>
<script>window.jQuery || document.write('<script src="js/libs/
jquery-1.7.2.min.js"><\script>')</script>
<script src="js/plugins.js"></script>
<script src="js/config-default.js"></script>
<script src="js/employees.js"></script>
```

modernizr-2.5.3.min.js - Modernizr is an open-source JavaScript library that helps you build the next generation of HTML5 and CSS3-powered websites. More information can be found at <http://modernizr.com/>

jquery-1.7.2.min.js - jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. More information can be found at <http://jquery.com/>

plugins.js - additional JS pluigns used in Employees App

config-default.js - is the small configuration file with App settings. Look for chapter 5.6

employees.js - is the custom script used for animation, effects and Ajax requests. You can learn more about jQuery methods used here (fadeOut(), fadeIn(), fadeTo(), animate(), jQuery.get(), html(), delay(), hover() etc.) by visiting this [jQuery Documentation](#) Site.

5.6. App Configuration

Employees configuration is located in the of **js/config-default.js** file. You can configure few app options and filter:

```
emp = {

  /* Path to XML file with empliyees */
  xmlPath : 'employees.xml',

  /* Popup open effect - 'flip' or 'fade'
  popupEffect : 'flip', // or 'fade'

  /* Filter Configuration */
  filter : {
    cols : 2,
    cell : [410,162],
    use : false
```

```
}  
}
```

Filter configuration:

- cols - number of person thumbs/cards per row
- cell - [width, height] - of person thumb/cards
- use - set to "true" if you are going to use filter

6. HTML Template System and XML Parser

As we find out from previous chapter is that to insert some person from XML to HTML it is enough to add the div with a class="person" and data-id attribute equal to person ID from XML, for example:

```
<div class="person" data-id="vk"></div>
```

And XML parser will insert HTML from XML for person:

```
<person id="vk">....</person>
```

Let's find out how all those persons from XML are going to HTML and how to edit their HTML output. Employees App uses its own XML parser which is based on HTML Template.

So, let's open index.html file with default layout and will look at the end of the file for templates. There are two templates - for person and for popup with detailed information about person. Let's look at the "person template" to learn how does it work (popup template works in the same way):

6.1. Person Template

```
<!-- Person template -->  
<div class="person-template" style="display: none;">  
  <div class="person-mask"></div>  
  <div class="person-avatar">  
    <img data-attr="{ 'src': 'avatar' }">  
    <div class="avatar-mask"></div>  
  </div>  
  <div class="person-info">  
    <p class="person-name" data-insert="name"></p>  
    <p class="person-position" data-insert="position"></p>  
    <div class="person-slogan" data-check="slogan">  
      <div class="person-slogan-angle"></div>  
      <span data-replace="slogan"></span>  
    </div>  
  </div>  
  <a class="open-popup learn-more" href="#"></a>  
  <div class="clearfix"></div>  
</div>
```

```
<!-- End Person Template -->
```

Every person's HTML will be outputted according to this HTML template. Of course you can edit it like you need without limitations. Now let's look at special triggers (marked with green) which allow flexible configuration of output.

1. "data-insert" Attribute

This special attribute will tell to XML parser to insert value of XML tag inside if element with this attribute.

So, for example, we have XML:

```
<person id="vk">
  <name>John</name>
  <secondname>Doe</secondname>
</person>
```

And the following HTML template:

```
<div>
  <p data-insert="name"></p>
  <p data-insert="secondname"></p>
</div>
```

The outputted HTML will be:

```
<div>
  <p data-insert="name">John</p>
  <p data-insert="secondname">Doe</p>
</div>
```

2. "data-replace" Attribute

This special attribute will tell to XML parser to replace the element with this attribute with a value of XML tag.

So, for example, we have XML:

```
<person id="vk">
  <name>John</name>
  <secondname>Doe</secondname>
</person>
```

And the following HTML template:

```
<p> <span data-replace="name"></span> </p>
<p> <span data-replace="secondname"></span> </p>
```

The outputted HTML will be:

```
<p> John </p>
<p> Doe </p>
```

3. "data-attr" Attribute

This special attribute will tell to XML parser to add HTML attributes to the element with this attribute.

So, for example, we have XML:

```
<person id="vk">
  <name>John Doe</name>
  <avatar>images/john_doe.jpg</avatar>
  <twitter>http://twitter.com/idangerous</twitter>
</person>
```

And the following HTML template:

```
<p> Name: <span data-replace="name"></span> </p>
<p> <img data-attr="{ 'src': 'avatar' }"> </p>
<p> <a data-attr="{ 'href': 'twitter', 'title': 'name' }">My Twitter Account</a> </p>
```

The outputted HTML will be:

```
<p> Name: John Doe </p>
<p>  </p>
<p> <a href="http://twitter.com/idangerous" title="John Doe">My Twitter Account</a> </p>
```

4. "data-check" Attribute

This special attribute will tell to XML parser to remove this HTML element if appropriate XML tag is missing.

So, for example, we have XML with 2 persons:

```
<person id="vk">
  <name>John Doe</name>
  <avatar>images/john_doe.jpg</avatar>
  <slogan>"Be yourself, everyone else is already taken!"</slogan>
</person>
<person id="bd">
  <name>Brian Doe</name>
  <avatar>images/brian_doe.jpg</avatar>
</person>
```

And the following HTML template:

```
<p> Name: <span data-replace="name"></span> </p>
<p> <img data-attr="{ 'src': 'avatar' }"> </p>
<p data-check="slogan"> Slogan: <span data-replace="slogan"></span> </p>
```

The outputted HTML for these 2 persons will be:

```
<p> Name: John Doe </p>
```

```
<p>  </p>
<p> Slogan: "Be yourself, everyone else is already taken!" </p>

<p> Name: Brian Doe </p>
<p>  </p>
```

5. "open-popup" Class

Clicking on any link with "open-popup" class which is inside of "person template" will open popup window with detailed information about person

6.2. Popup Template

Let's look at the default popup template in the index.html file:

```
<!-- Popup -->
<div class="popup-overlay"></div>
<div class="popup-view">
  <div class="popup">
    <div class="popup-content"> <a href="#" class="close-popup"></a>
    <div class="popup-left">
      <div class="popup-avatar"> <img data-attr="{ 'src': 'avatar' }"> </div>
      <div class="popup-links">
        <p data-check="twitter"><a class="external" data-
attr="{ 'href': 'twitter' }" target="_blank"><span data-replace="firstname"></span>
on Twitter</a></p>
        <p data-check="facebook"><a class="external" data-
attr="{ 'href': 'facebook' }" target="_blank">Facebook Page</a></p>
        <p data-check="linkedin"><a class="external" data-
attr="{ 'href': 'linkedin' }" target="_blank">LinkedIn Profile</a></p>
        <p><a href="#" class="open-popup-content bordered"><span data-
replace="firstname"></span>'s Bio</a></p>
      </div>
    </div>
    <div class="popup-info">
      <h2 class="popup-name" data-insert="name"></h2>
      <h3 class="popup-position" data-insert="position"></h3>
      <ul class="info-list">
        <li data-check="department"><i>Department:</i> <b data-
insert="department"></b></li>
        <li data-check="worksince"><i>Work since:</i> <b data-
insert="worksince"></b></li>
        <li data-check="slogan"><i>Slogan:</i> <b data-insert="slogan"></b></li>
        <li data-check="likes"><i>Likes:</i> <b data-insert="likes"></b></li>
        <li data-check="lives"><i>Lives in:</i> <b data-insert="lives"></b></li>
        <li data-check="born"><i>Born date:</i> <b data-insert="born"></b></li>
        <li data-check="website"><i>Website:</i> <b><a class="external" data-
insert="website" data-attr="{ 'href': 'website' }" target="_blank"></a></b></li>
      </ul>
    </div>
    <div class="clearfix"></div>
    <div class="popup-loader" data-check="bio">
      <div class="popup-loader-content"> <a class="close-popup-content"
href="#"></a> <span data-replace="bio"></span> </div>
```



```
        </div>
    </div>
</div>
</div>
<!--End Popup -->
```

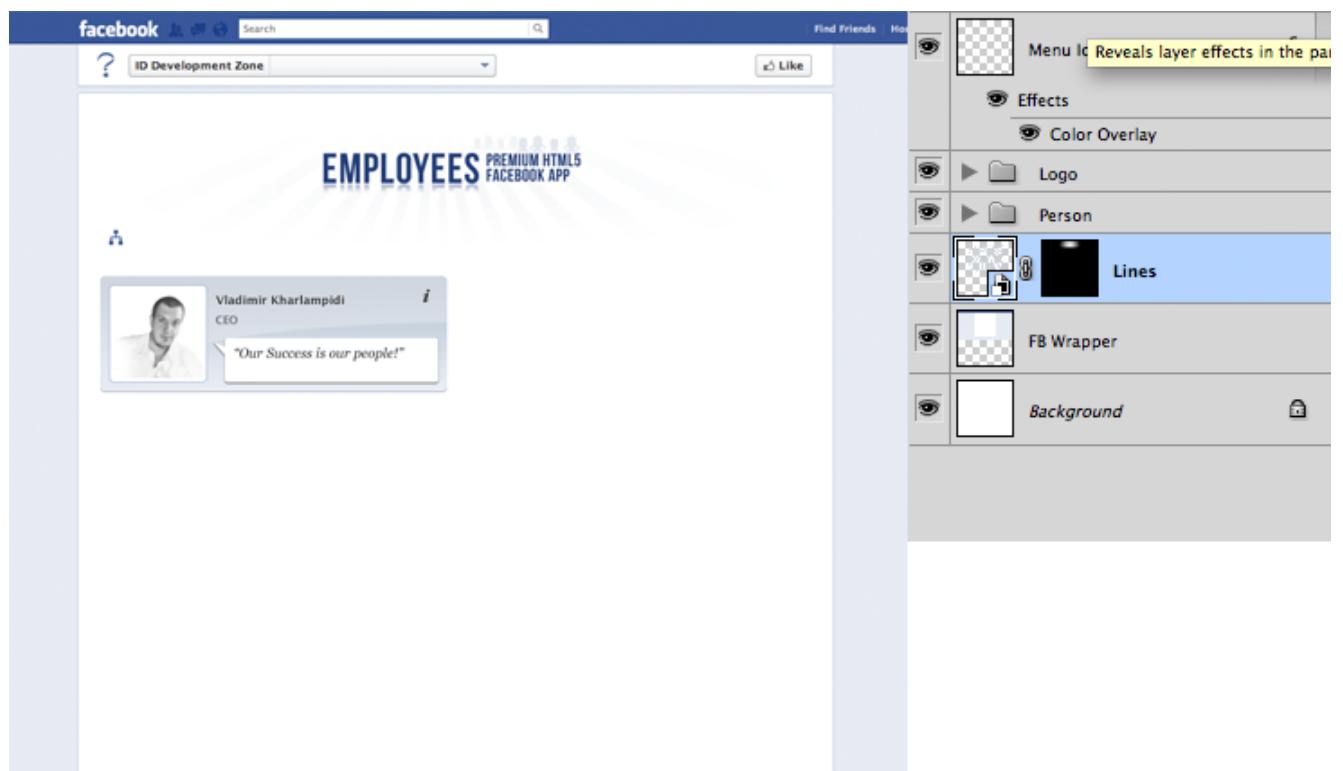
As you can see it also has "data-*" markers which work in the same way as in "person template". It also has some additional triggers:

- link(<a>) with an "open-popup-content" class. When you click on this link it will open div with "popup-loader-content" class, in this case it is a section with person's Biography.
- link(<a>) with an "close-popup-content" class. When you click on this link it will close div with "popup-loader-content" class, in this case it is a section with person's Biography.

7. PSD Files & Structure

There is one layered **psd** files comes with the **Employees** layout. You can locate it into **psd/** folder from the downloaded archive. Using this file you can easily redesign layout and every its elements, or to create your own color theme or layout. To edit it file you'll need an Adobe™ Photoshop™ (Recommended is CS5 or higher version)

Here is **employees.psd** file.

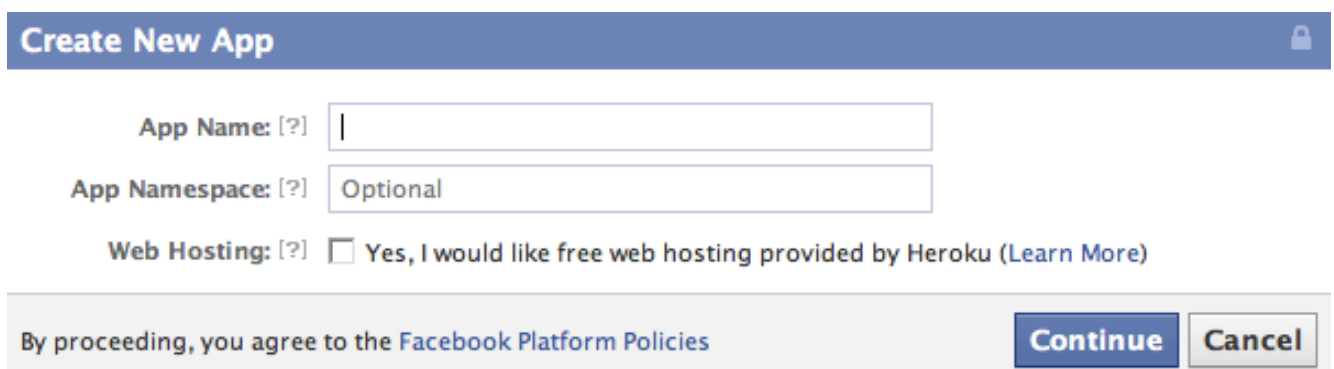


All layers in this file are well organized and easy to understand and redesign.

8. Set Up Facebook App

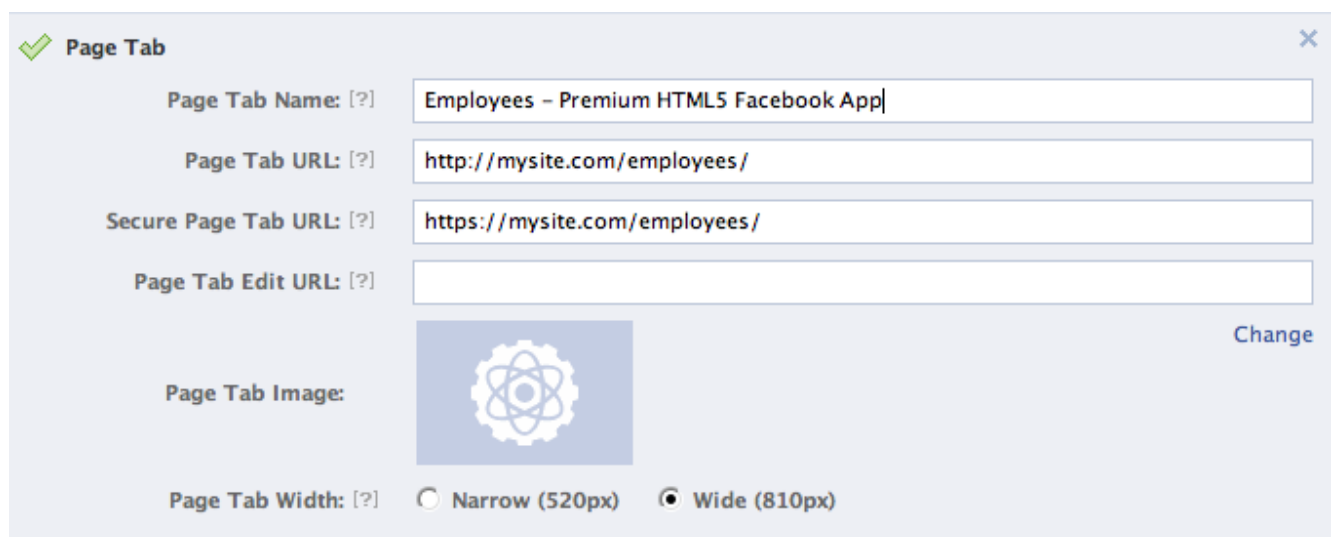
To add Facebook tab with Employees App you need:

1. Upload your Employees folder to real server.
2. Go to <http://developers.facebook.com>
3. Click "Apps" menu link:
4. Click "Create New App":
5. Fill up the following form:



The screenshot shows the 'Create New App' form on the Facebook Developers website. The form has a blue header with the title 'Create New App' and a lock icon. Below the header, there are three input fields: 'App Name' with a placeholder '[?]' and an empty text box, 'App Namespace' with a placeholder '[?]' and the value 'Optional', and 'Web Hosting' with a placeholder '[?]', an unchecked checkbox, and the text 'Yes, I would like free web hosting provided by Heroku (Learn More)'. At the bottom, there is a grey bar with the text 'By proceeding, you agree to the Facebook Platform Policies' and two buttons: 'Continue' and 'Cancel'.

6. After that you will see App's page where you can edit it's info, choose its picture, icon and etc. You need to edit "Page Tab" tab in the following order:



The screenshot shows the 'Page Tab' configuration page for a Facebook app. The page has a light blue background and a title bar with a green checkmark and the text 'Page Tab'. Below the title bar, there are four input fields: 'Page Tab Name' with a placeholder '[?]' and the value 'Employees - Premium HTML5 Facebook App', 'Page Tab URL' with a placeholder '[?]' and the value 'http://mysite.com/employees/', 'Secure Page Tab URL' with a placeholder '[?]' and the value 'https://mysite.com/employees/', and 'Page Tab Edit URL' with a placeholder '[?]' and an empty text box. Below the input fields, there is a 'Page Tab Image' section with a placeholder image of a gear icon and a 'Change' button. At the bottom, there is a 'Page Tab Width' section with a placeholder '[?]' and two radio buttons: 'Narrow (520px)' and 'Wide (810px)', with 'Wide (810px)' selected.

Page Tab URL - URL to "employees" folder on your server, it can be root folder of your site, not exactly "employees" folder like on screenshot above.

Secure Page Tab URL - the same URL but with HTTPS prefix. Your server should support SSL protocol. If its not you can leave this field empty, but some of facebook visitors will not be able to see your app

Page Tab Image: choose your page tab image

Page Tab Width : choose Wide (810px) !

7. After that click on the "Save Changes" button bellow:

<input checked="" type="checkbox"/> Native iOS App	Publish from my iOS app to Facebook.
<input checked="" type="checkbox"/> Native Android App	Publish from my Android app to Facebook.
<div>Save Changes</div>	

That is done, now your "app" is ready and it is ready to be integrated to your facebook page.

First of all you need and App ID. You can found it on the top of App Edit page:



And final step. Go to the page where you want to add Tab with Employees and type the following URL in your browser:

http://www.facebook.com/dialog/pagetag?app_id=YOUR_APP_ID&next=YOUR_URL

Where

YOUR_APP_ID - App ID of your APP.

YOUR_URL - URL specified in the Page Tab URL

According to the screenshot examples above the URL will look like:

http://www.facebook.com/dialog/pagetag?app_id=381400835250647&next=http://mysite.com/employees/

That is all. You will see screen where you will need to choose to which page you want to add this App.

You can learn more about configuring Page tabs on official Facebook developers website: <http://developers.facebook.com/docs/appsonfacebook/pagetabs/>

9. Licensing Terms

Regular License (RL)

RL gives you as a customer non-exclusive & non-transferable right to use the product you've bought, in this case is the **Employees** (further "**Item**")

1. We do not limit the number of **Item**'s copies you are going to use. Using one **Item** you can create, for example, even 100 web-sites.
2. You can use the **Item** by itself or it's also possible to apply it in other project you work at.
3. You can use the **Item** for your own purpose as well as for your clients.
4. You can use the **Item** in commercial projects.
5. You can reproduce the **Item**:
 - on a web-site or as a web-site
 - as a part of software you create not for sale
 - as a printed variant
 - in digital(electronic) format (as a presentation or an e-book)
 - in video products
6. All photos used on iDangero.us for demonstration photo gallery functioning are our private property (if only there is nothing said about contrary). It's prohibited to use these photos wherever.
7. You are not allowed to sell, resell, license or give the **Item** free (any way) without our written consent. Please, do not offer to do it to any person.
8. You also do not have the right to use the **Item** in a project for selling (for example, for other templates, scripts, graphics and so on).
9. It's prohibited to rework / redesign / reproduce the **Item** (i.e.to rename it or change graphics & so on) and after this to sell it as your own.
10. In spite of reselling limitation you could claim money for the **Item** from your client.
11. If the **Item** (the whole **Item** or its parts) is created with materials used by GNU General Public License (GPL) (or some other license) it means you should follow all the terms of the license using the **Item**.