

**DrupalGap**  
**Mobile apps & Web apps**  
**for**  
**Drupal 7**

<http://miniblue.codestudiola.com>

Drupal  
DrupalGap  
Cordova  
Jquery  
HTML5  
CSS  
Javascript

# Vanilla Install D7

## Drupal Commerce

entity  
rules  
addressfield  
views  
c-tools

## DrupalGap

services  
views\_datasources

## Commerce\_stripe

libraries  
php stripe library

version: 3.0 Alpha

# Commerce\_drupalGap

<http://drupalgap.org/project/commerce>

# Commerce\_services

[https://www.drupal.org/project/commerce\\_services](https://www.drupal.org/project/commerce_services)  
services & REST server

# Commerce\_drupalGap\_stripe

[http://drupalgap.org/project/commerce\\_drupalgap\\_stripe](http://drupalgap.org/project/commerce_drupalgap_stripe)  
[https://www.drupal.org/project/commerce\\_drupalgap\\_stripe](https://www.drupal.org/project/commerce_drupalgap_stripe)

Follow ALL the steps detailed on these pages

# DrupalGap commerce

<http://drupalgap.org/project/commerce>

Follow ALL the steps detailed on this pages

# DrupalGap addressfield

<http://drupalgap.org/project/addressfield>

# Create Stripe Account

## Creating a Test Account

Just want to tinker with Stripe for an afternoon, you can skip registration, get your test keys, and play

Head to <https://stripe.com>.

**Click:** Sign In

On the Sign In page click Sign up, next to Don't have an account?

Click skip this step and you'll be able to access a temporary, unnamed account that can be used for testing purposes.

On the resulting page (what Stripe calls your dashboard) click Your Account

In the Your Account control panel, click API Keys to find usable keys.

Click Save Account for *permanent* test keys

<https://stripe.com/docs/checkout/tutorial>

Once modules are installed - Enable the modules/  
features needed

Give users permission to go through the Checkout  
process

go to: **admin/people/permissions**

In the **Access checkout** permission  
check the box for

**Anonymous** and **Authenticated** users

Click the **Save permissions** button

# Add a Product Type

go to: <admin/commerce/products/types/>

**Name:** Music

**Description:** Spun, mixed, and scratched recordings, Tracks. Also good for playing frisbee.

click **Save and add fields** button.



# Add a Recordings Field to the Product type

\*where ever you see 'commerce' this is the 'store' button on admin menu

go to: <admin/commerce/products/types/music/fields>

Under the **Add new field** section

**Label:** Recordings

**Field type:** List (text)

**Widget:** Select list

click **Save** button.

Next: **Allowed values list**

Single

CD

Vinyl

click **Save** button.

\* each item on a single line

Next:

**Label:** Recordings

**Required field:** Check the box

**Default value:** Vinyl

**Enable this field to function as an attribute field on Add to Cart forms:** Check this box

**Number of values:** 1

click **Save settings** button.



Created  
PRODUCT Type

Lets **ADD** some products

# Add some Products

go to: <admin/commerce/products/add/music>

Then enter the following values:

**Product SKU:** mu100-pp

**Title:** The Pink Poodles

**Price:** 5.00

**Recordings:** CD

Click [Save and add another](#) button

**Product SKU:** mu105-pp (each product or version, needs it's own sku)

**Title:** The Pink Poodles

**Price:** 3.00

**Recordings:** Single

Click [Save and add another](#) button

**Product SKU:** mu150-ss

**Title:** Silver Screen

**Price:** 5.00

**Recordings:** CD

Click **Save and add another** button

**Product SKU:** mu155-ss

**Title:** Silver Screen

**Price:** 8.00

**Recordings:** Vinyl

Click **Save and add another** button

**Product SKU:** mu200-pl

**Title:** The Platters

**Price:** 8.00

**Recordings:** Vinyl

Click **Save and add another** button

**Product SKU:** mu250-br

**Title:** Brylcreems

**Price:** 3.00

**Recordings:** Single

Click **Save and add another** button

**Product SKU:** mu300-kp

**Title:** King Pins

**Price:** 8.00

**Recordings:** Vinyl

Click **Save and add another** button

**Product SKU:** mu350-tc

**Title:** The Cars

**Price:** 5.00

**Recordings:** CD

Click **Save and add another** button

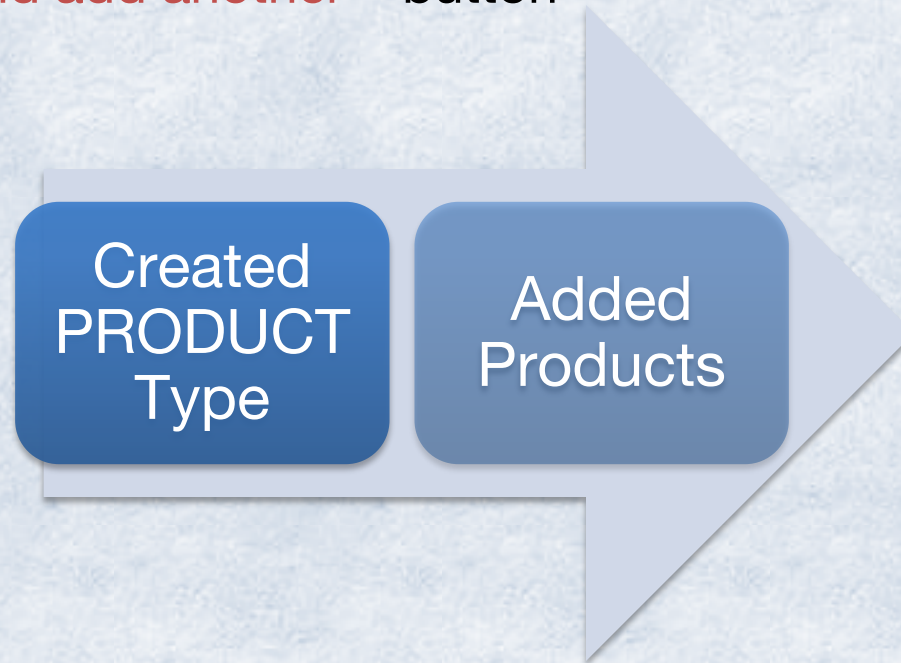
**Product SKU:** mu400-rw

**Title:** Radio Waves

**Price:** 8.00

**Recordings:** Vinyl

Click **Save and add another** button



At this point the products have been added to the data base.  
Now we need to add a content TYPE to display our products.

# Add a Content Type

go to: <admin/structure/types/add>

**Name:** Music Display

**Desc:** This will display music products.

**Publish:** Not Promoted

**Display Settings:** No Author

**Menu:** none

click [Save and add fields](#) button.

Continued on next slides-

# We will add 3 fields to the Content TYPE

Product  
Reference

- this will add the cart button in drupal commerce

Image

- show the product

Text

- a **short** field to add a tagline



**Add a Field:** to Reference the Products  
under the Add new field section

**Label:** Music Products

**Field type:** Product reference

**Widget:** Check boxes/radio buttons

click **Save** button

click **Save field settings** button

**Label:** Music Products

**Required field:** Check the box

**Products types that can be referenced:** Music

**Default value:** Vinyl

**Number of values:** Unlimited

click **Save settings** button.

# Add a Field Image Field to Show a Picture

If not there: [goto: admin/structure/types/manage/music/fields](#)

Under the Add new field section

**Label:** Cover Image

**Field type:** Image

**Widget:** Image

Click [Save](#) button

click [Save field settings](#) button

**Label:** Photo

**Required field:** Check the box

Leave all the other settings as the default values

click [Save settings](#) button.

# Add a Field short text field

Under the Add new field section

**Label:** Music Tagline

**Field Type:** Text

**Widget:** Text Field

Click **Save** button

We only want this for a SHORT description or tagline so lets keep it short for a reminder, also NO fancy text allowed - KISS

**Max Length:** 30 (255 is the maximum)

Click **Save** button

**Size of text field:** 30

**Text Processing:** Plain Text

click **Save settings** button

Now lets look at the LAYOUT of the page that you just created

If not there: goto: <admin/structure/types/manage/music/fields>

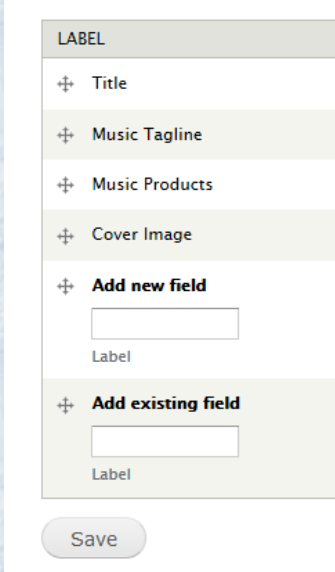
Manage Fields:

**Body:** Delete

**Tagline:** Move under the title

Click **Save**

Should look like this:



The screenshot shows the 'Manage Fields' interface for the 'Music' type. It features a list of fields with a plus icon to the left of each field name. The fields listed are: Title, Music Tagline, Music Products, and Cover Image. Below these is an 'Add new field' section with a text input field and a 'Label' label. Below that is an 'Add existing field' section with a text input field and a 'Label' label. At the bottom of the interface is a 'Save' button.

LABEL	
+	Title
+	Music Tagline
+	Music Products
+	Cover Image
+	<b>Add new field</b>
	<input type="text"/>
	Label
+	<b>Add existing field</b>
	<input type="text"/>
	Label

Save

Created  
PRODUCT  
Type

Added  
Products

Created  
CONTENT  
Type

Lets ADD content

Content allows the **Add to cart** functionality to work in Drupal Commerce.

go to: [node/add/music-display](#)

**Title:** The Pink Poodles

**Tagline:** featuring: Shirley

**Music Products:**

The Pink Poodles CD  Check this box

The Pink Poodles Single  Check this box

**Cover Image:** Upload a picture to use

**Authoring:** Remove

**Save**

Now repeat this for all the products -

**Title: Silver Screen**

**Tagline:** All the Show Tunes

**Music Products:**

Silver Screen CD  Check this box

Silver Screen Vinyl  Check this box

**Cover Image:**

Upload a picture to use

**Authoring:** Remove

[Save](#)

**Title: The Platters**

**Tagline:** 1959 Golden Hits

**Music Products:**

The Platters Vinyl  Check this box

**Cover Image:**

Upload a picture to use

**Authoring:** Remove

[Save](#)

**Title: Brylcreems**

**Tagline:** Lend Me Your Comb

**Music Products:**

Brylcreems Single  Check this box

**Cover Image:**

Upload a picture to use

**Authoring:** Remove

[Save](#)

**Title: King Pins**

**Tagline:** Saturday Night Favorites

**Music Products:**

King Pins Vinyl  Check this box

**Cover Image:**

Upload a picture to use

**Authoring:** Remove

[Save](#)



**Title: The Cars**

**Tagline:** Crusin Tunes

**Music Products:**

The Cars CD  Check this box

**Cover Image:**

Upload a picture to use

**Authoring:** Remove

[Save](#)

**Title: Radio Waves**

**Tagline:** Blues

**Music Products:**

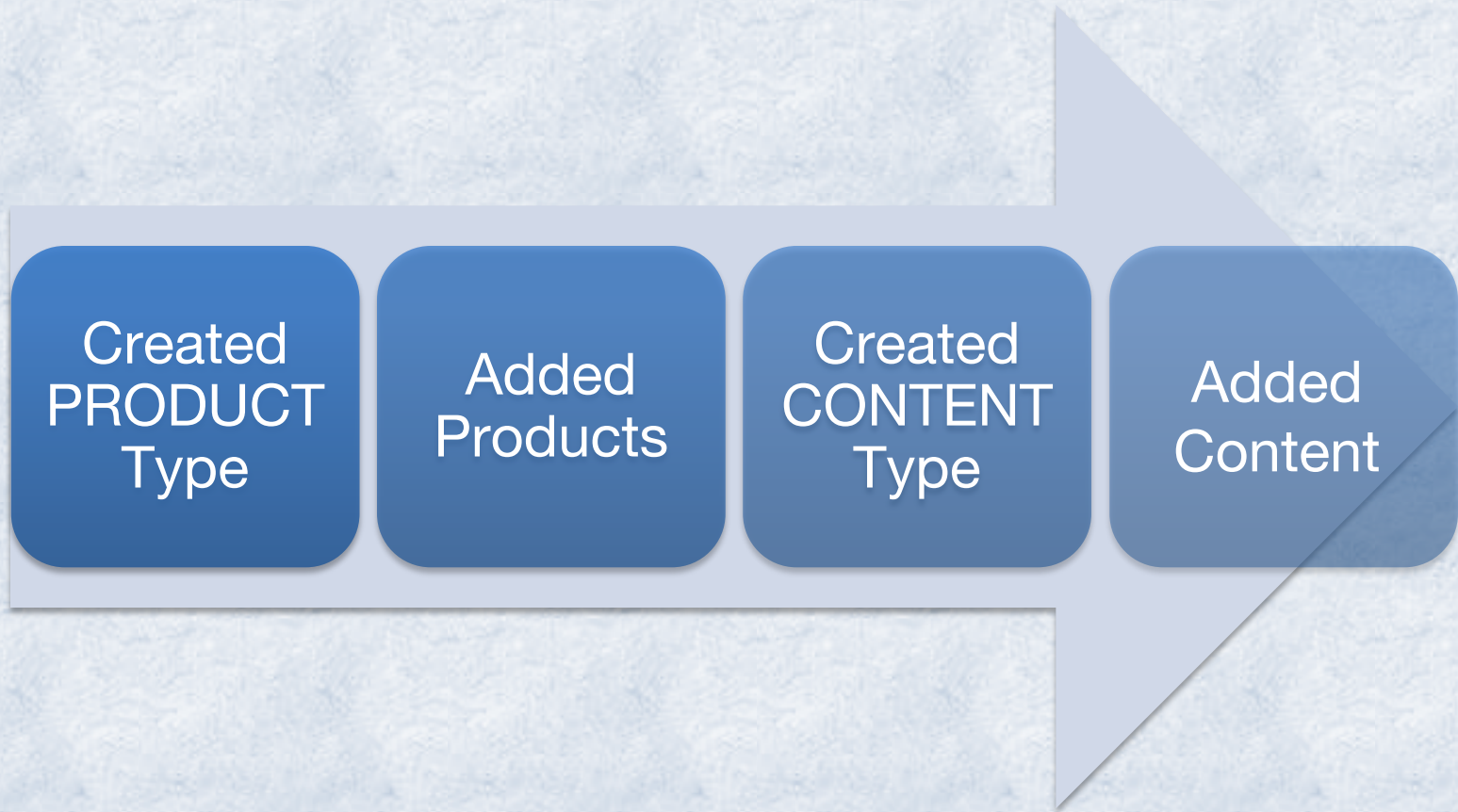
Radio Waves Vinyl  Check this box

**Cover Image:**

Upload a picture to use

**Authoring:** Remove

[Save](#)



Now we will use **Views** to create a page  
to display Product content

# Views Create a Page

goto: <admin/structure/views/add>

**View name:** Recordings

**Show:** Content

**of type:** Music Display

**sorted by:** Title

Next check the **Create a page**

**Page title:** Recordings

**Path:** recordings

**Display format:** Table

**Items to display:** 0

check **Create a menu link** checkbox

**Menu:** Main menu

**Link text:** Recordings

click **Continue & edit** button.

# Add an Image and Layout to the Product Page

## Fields section

Click **Add** button

**Search box:** cover

Check **Content: Photo**

Click **All Displays** button

**Change Label:** Photo

**Uncheck :** Place a colon after the label

**Image style:** Thumbnail (100 x 100)

**Link image to:** Content

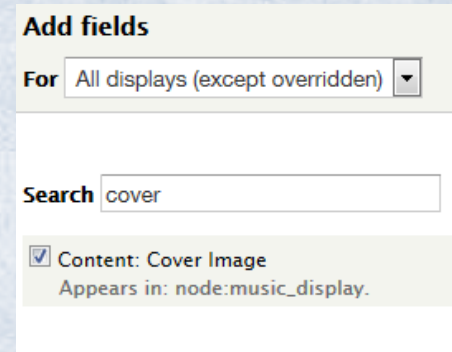
Click **Apply** button

Fields **Content:Title (title)** **Click**

**Uncheck :** Place a colon after the label

**Save**

Continue on next slide-



**Add fields**

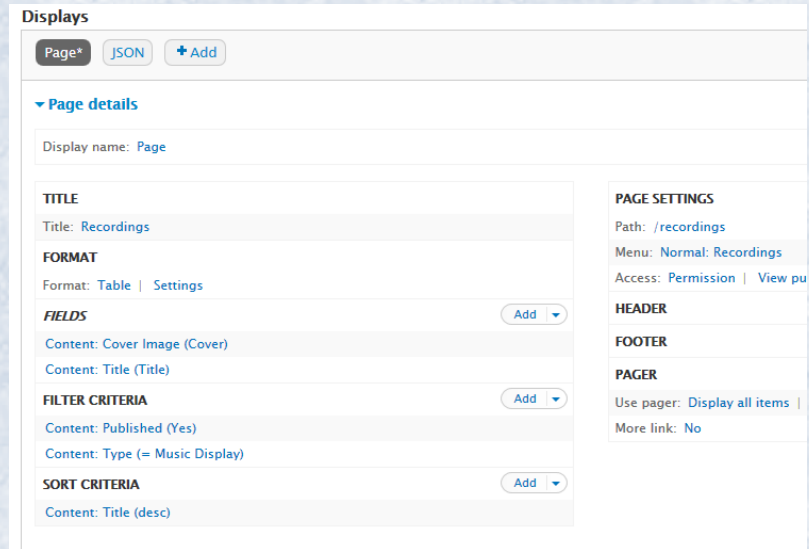
For

Search

Content: Cover Image  
Appears in: node:music\_display.

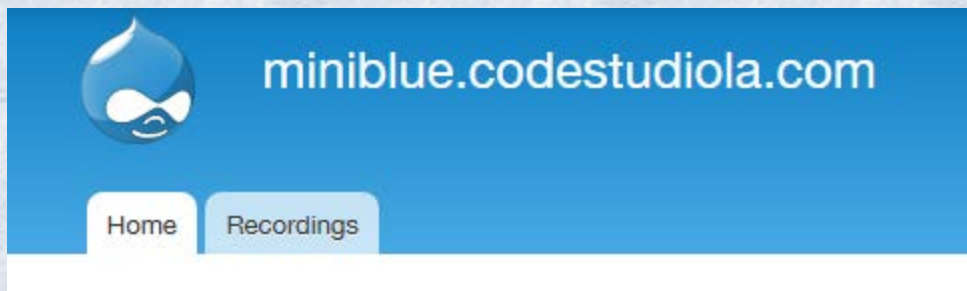
Fields click down arrow **Rearrange** button  
Drag Cover Image above title  
Click **Apply** button

**Save** your View










The screenshot shows the 'Displays' configuration interface for a 'Page' display. At the top, there are buttons for 'Page\*', 'JSON', and '+ Add'. Below this is a section for 'Page details' with a dropdown arrow. The 'Display name' is set to 'Page'. The main configuration area is divided into several sections: 'TITLE' (Title: Recordings), 'FORMAT' (Format: Table | Settings), 'FIELDS' (Content: Cover Image (Cover), Content: Title (Title)), 'FILTER CRITERIA' (Content: Published (Yes), Content: Type (= Music Display)), and 'SORT CRITERIA' (Content: Title (desc)). Each section has an 'Add' button with a dropdown arrow. On the right side, there are sections for 'PAGE SETTINGS' (Path: /recordings, Menu: Normal: Recordings, Access: Permission | View pu), 'HEADER', 'FOOTER', and 'PAGER' (Use pager: Display all items | More link: No).

Now if you look at the home page you will see a new Menu Item added.



# Recordings Page View

## Recordings

Cover	Title
	<a href="#">The Platters</a>
	<a href="#">The Pink Poodles</a>
	<a href="#">The Cars</a>
	<a href="#">Silver Screen</a>
	<a href="#">Radio Waves</a>
	<a href="#">King Pins</a>
	<a href="#">Brylcreems</a>

# Lets Enable a Payment Method

I choose to use Stripe

- It offers a test environment without a sandbox
- Super easy to change to LIVE once testing is complete
- DrupalGap works with Stripe

go to: **admin/commerce/config/payment-methods**

Click **enable** on the disabled Stripe payment method rule

Click **edit** next to the enabled Stripe payment method rule

Click **edit** next to the Stripe action

Grab your **Secret** and **Publishable** test keys from the account you setup earlier

Paste them into the corresponding text fields in the **Payment Settings** section

Click **Save**

Flush **ALL** Drupal cache



Your E-commerce Store is  
complete!  
Take it for a test drive.

ORDER NUMBER	UPDATED	NAME	USER	TOTAL	ORDER STATUS
2	Tue, 01/24/2017 - 12:05	george mack	g	\$8.00	Pending
1	Tue, 01/24/2017 - 06:42	sally	01dev	\$11.00	Pending

Now we can build the app

# Enable DrupalGap in your store

go to: <admin/config/services/drupalgap>

Click **Test Connection**

If module is installed and enabled **SUCCESS!**

It will then ask if you want to get SDK

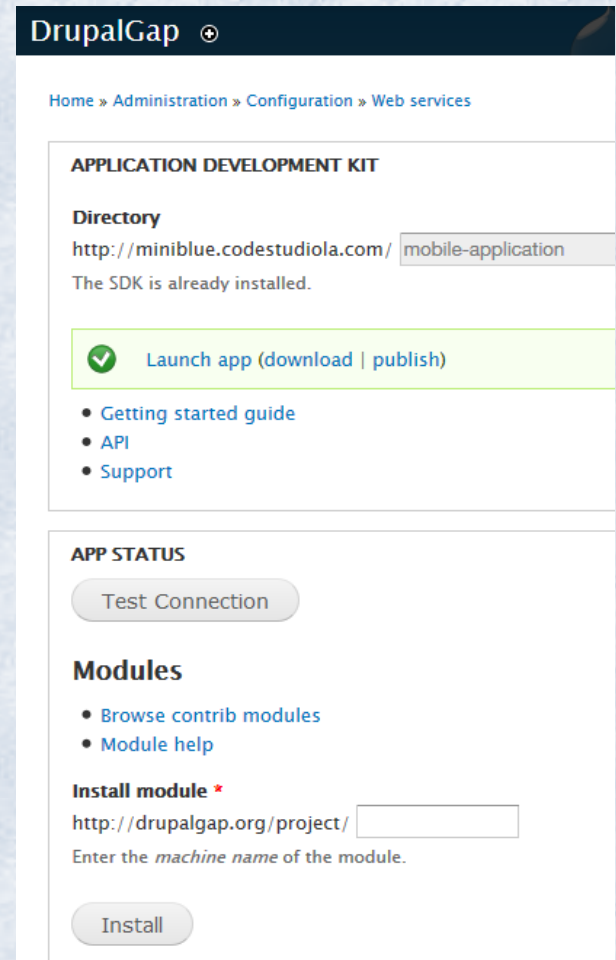
**click button**

Your interface will end up looking like this:

Follow the details on these pages to complete instillation:

<http://drupalgap.org/project/commerce>

[http://drupalgap.org/project/commerce\\_drupalgap\\_stripe](http://drupalgap.org/project/commerce_drupalgap_stripe)



The screenshot shows the DrupalGap configuration interface. At the top, there is a dark blue header with the text "DrupalGap" and a plus icon. Below the header, a breadcrumb trail reads "Home » Administration » Configuration » Web services". The main content area is divided into two sections. The first section, titled "APPLICATION DEVELOPMENT KIT", shows the "Directory" field with the value "http://miniblu.codestudiola.com/mobile-application" and a message stating "The SDK is already installed." Below this, there is a green button with a checkmark icon and the text "Launch app (download | publish)". Underneath the button, there are three links: "Getting started guide", "API", and "Support". The second section, titled "APP STATUS", contains a "Test Connection" button. Below this, there is a "Modules" section with two links: "Browse contrib modules" and "Module help". At the bottom of the "APP STATUS" section, there is an "Install module" section with a red asterisk, a text input field containing "http://drupalgap.org/project/", and a message that says "Enter the machine name of the module." Below this input field is an "Install" button.

# Create a JSON Page in Views to Retrieve the Recordings List

go to: <admin/structure/views/view/recordings/edit>

Click **+Add**

Click **Page**

**Change:** Display name to **JSON**

**Change:** Format to **JSON Data Document on This page (override)**

Click **Apply (this display)** button

Uncheck **Views API mode** checkbox

Click **Apply (this display)** button

Under **Page Settings** : change **Path** to **recordings.json**

Under **Fields**: change **Label** on both fields so the first character in the label is lower case on **this display** only

**Fields:** click **Add** button

**Search:** nid

Check box **Content: Nid**

**Change:** button to **Apply (this display)** Click

**Label:** nid

Click **Apply (this display)** button

[Continued ... Next slide](#)

**Save** your View

Should look like this:

**Displays**

Page JSON + Add

▼ JSON details

Display name: JSON

---

<b>TITLE</b> Title: Recordings	
<b>FORMAT</b> Format: <i>JSON data document</i>   <i>Settings</i>	
<b>FIELDS</b> <span>Add ▼</span>	
Content: Cover Image (cover)	
Content: Title (title)	
Content: Nid (nid)	
<b>FILTER CRITERIA</b> <span>Add ▼</span>	
Content: Published (Yes)	
Content: Type (= Music Display)	
<b>SORT CRITERIA</b> <span>Add ▼</span>	
Content: Title (desc)	

<b>PAGE SETTINGS</b>
Path: /recordings.json
Menu: No menu
Access: <a href="#">Permission</a>   <a href="#">View published</a>
<b>HEADER</b>
<b>FOOTER</b>
<b>PAGER</b>
Use pager: <a href="#">Display all items</a>   <a href="#">All it</a>
More link: No

## Sample of JSON view

Title ⚙️▼

Recordings

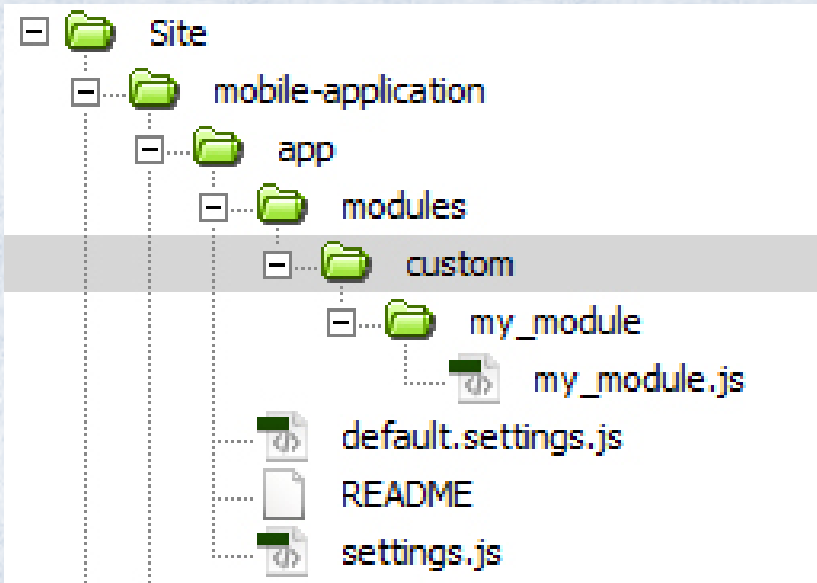
Content ⚙️▼

```
{
  "nodes" : [
    {
      "node" : {
        "cover" : {
          "src" : "http://your website path",
          "alt" : ""
        },
        "title" : "The Platters",
        "nid" : "3"
      }
    },
    {
      "node" : {
        "cover" : {
          "src" : "http://your website path",
          "alt" : ""
        },
        "title" : "The Pink Poodles",
        "nid" : "1"
      }
    },
    {
      "node" : {
        "cover" : {
```

# Create a Page to Display the Recordings List in App

```
1  /**
2   * Implements hook_menu().
3   */
4  function my_module_menu() {
5      try {
6          var items = {};
7          items['recordings-list'] = {
8              title: 'Recordings List',
9              page_callback: 'my_module_recordings_list_page'
10         };
11         return items;
12     }
13     catch (error) {
14         console.log('my_module_menu - ' + error);
15     }
16 }
17
18 function my_module_recordings_list_page() {
19     try {
20         var content = {};
21         content['my_recordings_list'] = {
22             theme: 'view',
23             format: 'ul',
24             path: 'recordings.json',
25             row_callback: 'my_module_recordings_list_page_row',
26             empty_callback: 'my_module_recordings_list_page_empty'
27         };
28         return content;
29     }
30     catch (error) { console.log('my_module_recordings_list_page - ' + error); }
31 }
32
33 function my_module_recordings_list_page_row(view, row) {
34     try {
35         var image = theme('image', { path: row.cover.src });
36         var title = '<h2>' + row.title + '</h2>';
37         var html = l(image + title, 'node/' + row.nid);
38         return html;
39     }
40     catch (error) { console.log('my_module_recordings_list_page_row - ' + error); }
41 }
42
43 function my_module_recordings_list_page_empty(view) {
44     try {
45         return "Sorry, we are out of recordings.";
46     }
47     catch (error) { console.log('my_module_recordings_list_page_empty - ' + error); }
48 }
```

# Create a Custom DrupalGap Module to Customize our App



Create: **custom** folder  
Inside that folder

Create: **my\_module** folder  
Inside my\_module folder

Create: **my\_module.js**

Save the code from previous slide  
in the my\_module.js file

**SAVE**

Congratulations! You just made a custom module. The code you added creates a custom page using hook\_menu()  
add this to the **settings.js** file: (last file in image above)

```
// App Front Page  
drupalgap.settings.front = 'recordings-list';
```

# Then enable the modules add this to: `settings.js`

```
/** Contributed Modules - www/app/modules */  
  
//Drupal.modules.contrib['example'] = {};  
Drupal.modules.contrib['addressfield'] = { minified: true };  
Drupal.modules.contrib['commerce'] = { minified: true };  
drupalgap.settings.commerce = {  
  bundles: {  
    music_display: {  
      product_reference_field_name: 'field_music_products'  
    }  
  }  
};  
Drupal.modules.contrib['commerce_drupalgap_stripe'] = {};  
  
/** Custom Modules - www/app/modules/custom */  
Drupal.modules.custom['my_module'] = {};
```



# Build App and Test on Device

go to: <admin/config/services/drupalgap>

Here you can:

**Launch app:** to test in browser

**Download app:** to build in cordova or phoneGap

\* clear **ALL** cache to view in browser

To Build with PhoneGap follow these steps:

[http://docs.drupalgap.org/7/Compiling\\_a\\_Mobile\\_Application/Compiling\\_an\\_App\\_with\\_PhoneGap\\_Build](http://docs.drupalgap.org/7/Compiling_a_Mobile_Application/Compiling_an_App_with_PhoneGap_Build)

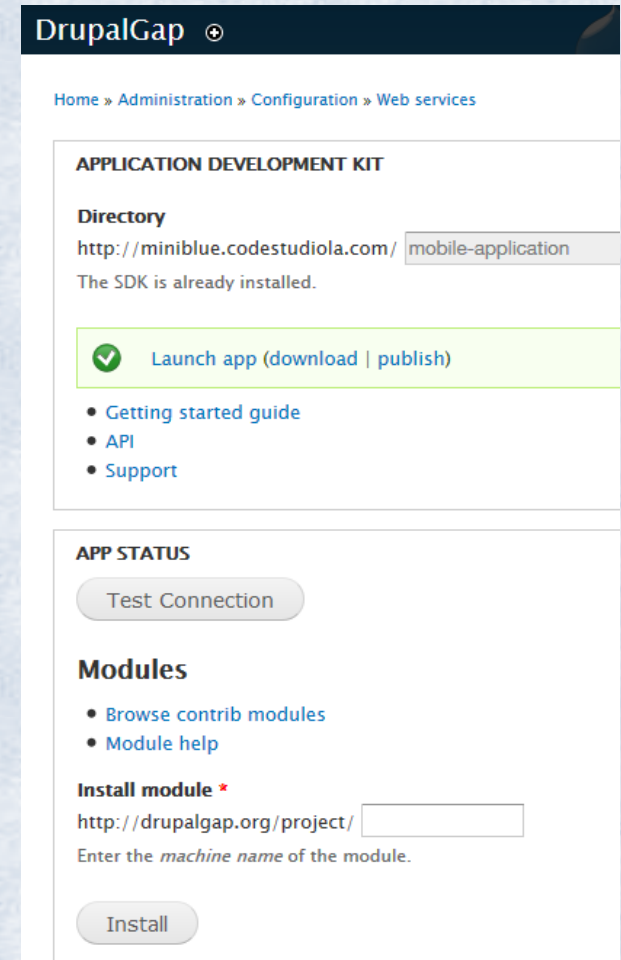
\*this is Adobe tool, no coding needed

To build with Cordova follow these steps:

[http://docs.drupalgap.org/7/Compiling\\_a\\_Mobile\\_Application](http://docs.drupalgap.org/7/Compiling_a_Mobile_Application)

\* this is the commandLine tool

Walk thru for this method now



The screenshot shows the DrupalGap administration interface. At the top, there is a dark blue header with the text "DrupalGap" and a plus icon. Below the header, a breadcrumb trail reads "Home » Administration » Configuration » Web services". The main content area is titled "APPLICATION DEVELOPMENT KIT". Under this heading, there is a "Directory" section with the URL "http://miniblue.codestudiola.com/" and a text input field containing "mobile-application". Below this, it states "The SDK is already installed." A prominent green button with a checkmark icon and the text "Launch app (download | publish)" is visible. Below the button, there are three links: "Getting started guide", "API", and "Support". The "APP STATUS" section contains a "Test Connection" button. The "Modules" section has two links: "Browse contrib modules" and "Module help". The "Install module" section features a red asterisk, the URL "http://drupalgap.org/project/" followed by an empty text input field, and the instruction "Enter the machine name of the module." Below this is an "Install" button.

Thank you 4 coming 2 the LA PhoneGap user group Meetup!

I ve enjoyed sharing the info with you  
Now create **YOUR** e-Commerce site  
add **YOUR** apps to the app store  
;)

<http://miniblue.codestudiola.com>



shaRon sachse  
<http://sharon-folio.com>  
@PCartiste