Android 101:

Build an app from start to finish

Saving and Displaying Data

Girl Develop It

GDI

don't be shy. develop it.

Image Model

Image Model

Class with getters and setters

Image Model

Class with getters and setters

For saving the images and retrieving them later

Image Model

Class with getters and setters

For saving the images and retrieving them later

What data do we want to know?

Image Model

Class with getters and setters

For saving the images and retrieving them later

What data do we want to know?
  The Image itself

Image Model

Class with getters and setters

For saving the images and retrieving them later

What data do we want to know?
  The Image itself
  Title

Image Model

Class with getters and setters

For saving the images and retrieving them later

What data do we want to know?

- The Image itself
- Title
- Description

Image Model

Class with getters and setters

For saving the images and retrieving them later

What data do we want to know?

- The Image itself
- Title
- Description
- Date Created

Image Model

Class with getters and setters

For saving the images and retrieving them later

What data do we want to know?

- The Image itself
- Title
- Description
- Date Created
- Username

ImageModel

ImageModel

```
private String title;
```

ImageModel

```
private String title;
private String username;
```

ImageModel

```
private String title;
private String username;
private String description;
```

ImageModel

```
private String title;
private String username;
private String description;
private long dateCreated;
```

ImageModel

```
private String title;
private String username;
private String description;
private long dateCreated;
private String pathToImage;
```

ImageModel

Build a getter and setter for each private variable

ImageModel

```
public void setTitle(String t){
```

Build a getter and setter for each private variable

ImageModel

```
public void setTitle(String t){
   title = t;
```

Build a getter and setter for each private variable

ImageModel

```
public void setTitle(String t){
   title = t;
}
```

Build a getter and setter for each private variable

ImageModel

```
public void setTitle(String t){
   title = t;
}
public String getTitle(){
```

Build a getter and setter for each private variable

ImageModel

```
public void setTitle(String t){
   title = t;
}
public String getTitle(){
   return title;
```

Build a getter and setter for each private variable

ImageModel

```
public void setTitle(String t){
   title = t;
}
public String getTitle(){
   return title;
}
```

Build a getter and setter for each private variable

ImageModel

```
public void setTitle(String t){
   title = t;
}
public String getTitle(){
   return title;
}
```

Build a getter and setter for each private variable

```
public void setDateCreated(long date){
```

ImageModel

```
public void setTitle(String t){
   title = t;
}
public String getTitle(){
   return title;
}
```

Build a getter and setter for each private variable

```
public void setDateCreated(long date){
   dateCreated = date;
```

ImageModel

```
public void setTitle(String t){
   title = t;
}
public String getTitle(){
   return title;
}
```

Build a getter and setter for each private variable

```
public void setDateCreated(long date){
   dateCreated = date;
}
```

ImageModel

```
public void setTitle(String t){
   title = t;
}
public String getTitle(){
   return title;
}
```

Build a getter and setter for each private variable

```
public void setDateCreated(long date){
   dateCreated = date;
}
public long getDateCreated(){
```

ImageModel

```
public void setTitle(String t){
   title = t;
}
public String getTitle(){
   return title;
}
```

Build a getter and setter for each private variable

```
public void setDateCreated(long date){
   dateCreated = date;
}
public long getDateCreated(){
   return dateCreated;
```

ImageModel

```
public void setTitle(String t){
   title = t;
}
public String getTitle(){
   return title;
}
```

Build a getter and setter for each private variable

```
public void setDateCreated(long date){
   dateCreated = date;
}
public long getDateCreated(){
   return dateCreated;
}
```

ImageModel

ImageModel

```
public ImageModel(String title, String username,
String description, String pathToImage, long
dateCreated){
```

ImageModel

```
public ImageModel(String title, String username,
String description, String pathToImage, long
dateCreated){

    setTitle(title);
```

ImageModel

```
public ImageModel(String title, String username,
String description, String pathToImage, long
dateCreated){

    setTitle(title);
    setUsername(username);
```

ImageModel

```
public ImageModel(String title, String username,
String description, String pathToImage, long
dateCreated){

    setTitle(title);
    setUsername(username);
    setDescription(description);
```

ImageModel

```
public ImageModel(String title, String username,
String description, String pathToImage, long
dateCreated){

    setTitle(title);
    setUsername(username);
    setDescription(description);
    setPathToImage(pathToImage);
```

ImageModel

```
public ImageModel(String title, String username,
String description, String pathToImage, long
dateCreated){

    setTitle(title);
    setUsername(username);
    setDescription(description);
    setPathToImage(pathToImage);
    setDateCreated(dateCreated);
```

ImageModel

```
public ImageModel(String title, String username,
String description, String pathToImage, long
dateCreated){

    setTitle(title);
    setUsername(username);
    setDescription(description);
    setPathToImage(pathToImage);
    setDateCreated(dateCreated);

}
```

ImageModel

Build the constructor.

```
public ImageModel(String title, String username,
String description, String pathToImage, long
dateCreated){

    setTitle(title);
    setUsername(username);
    setDescription(description);
    setPathToImage(pathToImage);
    setDateCreated(dateCreated);

}
```

ImageModel

Build the constructor.
Used when we create a new Image

```
public ImageModel(String title, String username,
String description, String pathToImage, long
dateCreated){

   setTitle(title);
   setUsername(username);
   setDescription(description);
   setPathToImage(pathToImage);
   setDateCreated(dateCreated);

}
```

TakePictureActivity

save(View v)

TakePictureActivity

save(View v)

```
String imageTitle = titleField.getText().toString();
```

TakePictureActivity

save(View v)

```
String imageTitle = titleField.getText().toString();
String imageDescription = descriptionField.getText().toString();
```

TakePictureActivity

save(View v)

```
String imageTitle = titleField.getText().toString();
String imageDescription = descriptionField.getText().toString();

if(imageTitle.equals("") || imageDescription.equals("") ||
pathToImage.equals("")){
```

TakePictureActivity

save(View v)

```
String imageTitle = titleField.getText().toString();
String imageDescription = descriptionField.getText().toString();

if(imageTitle.equals("") || imageDescription.equals("") ||
pathToImage.equals("")){
   AlertDialog.Builder builder = new AlertDialog.Builder(this);
```

TakePictureActivity

save(View v)

```
String imageTitle = titleField.getText().toString();
String imageDescription = descriptionField.getText().toString();

if(imageTitle.equals("") || imageDescription.equals("") ||
pathToImage.equals("")){
   AlertDialog.Builder builder = new AlertDialog.Builder(this);
   builder.setMessage("Sorry, all fields are required")
```

TakePictureActivity

save(View v)

```
String imageTitle = titleField.getText().toString();
String imageDescription = descriptionField.getText().toString();

if(imageTitle.equals("") || imageDescription.equals("") ||
pathToImage.equals("")){
   AlertDialog.Builder builder = new AlertDialog.Builder(this);
   builder.setMessage("Sorry, all fields are required")
          .setCancelable(false)
```

TakePictureActivity

save(View v)

```
String imageTitle = titleField.getText().toString();
String imageDescription = descriptionField.getText().toString();

if(imageTitle.equals("") || imageDescription.equals("") ||
pathToImage.equals("")){
   AlertDialog.Builder builder = new AlertDialog.Builder(this);
   builder.setMessage("Sorry, all fields are required")
          .setCancelable(false)
          .setNegativeButton("OK", new
     DialogInterface.OnClickListener() {
```

TakePictureActivity

save(View v)

```
String imageTitle = titleField.getText().toString();
String imageDescription = descriptionField.getText().toString();

if(imageTitle.equals("") || imageDescription.equals("") ||
pathToImage.equals("")){
   AlertDialog.Builder builder = new AlertDialog.Builder(this);
   builder.setMessage("Sorry, all fields are required")
          .setCancelable(false)
          .setNegativeButton("OK", new
     DialogInterface.OnClickListener() {
          public void onClick(DialogInterface dialog, int id) {
```

TakePictureActivity

save(View v)

```
String imageTitle = titleField.getText().toString();
String imageDescription = descriptionField.getText().toString();

if(imageTitle.equals("") || imageDescription.equals("") ||
pathToImage.equals("")){
   AlertDialog.Builder builder = new AlertDialog.Builder(this);
   builder.setMessage("Sorry, all fields are required")
          .setCancelable(false)
          .setNegativeButton("OK", new
     DialogInterface.OnClickListener() {
          public void onClick(DialogInterface dialog, int id) {
               dialog.cancel();
```

TakePictureActivity

save(View v)

```
String imageTitle = titleField.getText().toString();
String imageDescription = descriptionField.getText().toString();

if(imageTitle.equals("") || imageDescription.equals("") ||
pathToImage.equals("")){
   AlertDialog.Builder builder = new AlertDialog.Builder(this);
   builder.setMessage("Sorry, all fields are required")
          .setCancelable(false)
          .setNegativeButton("OK", new
     DialogInterface.OnClickListener() {
          public void onClick(DialogInterface dialog, int id) {
               dialog.cancel();
          }
```

TakePictureActivity

save(View v)

```
String imageTitle = titleField.getText().toString();
String imageDescription = descriptionField.getText().toString();

if(imageTitle.equals("") || imageDescription.equals("") ||
pathToImage.equals("")){
   AlertDialog.Builder builder = new AlertDialog.Builder(this);
   builder.setMessage("Sorry, all fields are required")
          .setCancelable(false)
          .setNegativeButton("OK", new
     DialogInterface.OnClickListener() {
          public void onClick(DialogInterface dialog, int id) {
               dialog.cancel();
          }
   });
```

TakePictureActivity

save(View v)

```
String imageTitle = titleField.getText().toString();
String imageDescription = descriptionField.getText().toString();

if(imageTitle.equals("") || imageDescription.equals("") ||
pathToImage.equals("")){
   AlertDialog.Builder builder = new AlertDialog.Builder(this);
   builder.setMessage("Sorry, all fields are required")
          .setCancelable(false)
          .setNegativeButton("OK", new
     DialogInterface.OnClickListener() {
          public void onClick(DialogInterface dialog, int id) {
               dialog.cancel();
          }
   });
   AlertDialog alert = builder.create();
```

TakePictureActivity

save(View v)

```
String imageTitle = titleField.getText().toString();
String imageDescription = descriptionField.getText().toString();

if(imageTitle.equals("") || imageDescription.equals("") ||
pathToImage.equals("")){
   AlertDialog.Builder builder = new AlertDialog.Builder(this);
   builder.setMessage("Sorry, all fields are required")
          .setCancelable(false)
          .setNegativeButton("OK", new
     DialogInterface.OnClickListener() {
          public void onClick(DialogInterface dialog, int id) {
               dialog.cancel();
          }
   });
   AlertDialog alert = builder.create();
   alert.show();
```

TakePictureActivity

save(View v)

```
String imageTitle = titleField.getText().toString();
String imageDescription = descriptionField.getText().toString();

if(imageTitle.equals("") || imageDescription.equals("") ||
pathToImage.equals("")){
   AlertDialog.Builder builder = new AlertDialog.Builder(this);
   builder.setMessage("Sorry, all fields are required")
          .setCancelable(false)
          .setNegativeButton("OK", new
     DialogInterface.OnClickListener() {
          public void onClick(DialogInterface dialog, int id) {
               dialog.cancel();
          }
   });
   AlertDialog alert = builder.create();
   alert.show();
}
```

TakePictureActivity

save(View v)

TakePictureActivity

save(View v)

```
else{
```

TakePictureActivity

save(View v)

```
else{

   ImageModel imageModel = new ImageModel(imageTitle,
```

TakePictureActivity

save(View v)

```
else{

    ImageModel imageModel = new ImageModel(imageTitle,
app.getUsername(), imageDescription, pathToImage,
```

TakePictureActivity

save(View v)

```
else{

   ImageModel imageModel = new ImageModel(imageTitle,
app.getUsername(), imageDescription, pathToImage,
new Date().getTime());
```

TakePictureActivity

save(View v)

```
else{

   ImageModel imageModel = new ImageModel(imageTitle, 
app.getUsername(), imageDescription, pathToImage, 
new Date().getTime());
```

TakePictureActivity

save(View v)

```
else{

   ImageModel imageModel = new ImageModel(imageTitle,
app.getUsername(), imageDescription, pathToImage,
new Date().getTime());

  ArrayList<ImageModel> imageList = app.getImages();
```

TakePictureActivity

save(View v)

```
else{

   ImageModel imageModel = new ImageModel(imageTitle,
app.getUsername(), imageDescription, pathToImage,
new Date().getTime());

  ArrayList<ImageModel> imageList = app.getImages();
  imageList.add(0, imageModel);
```

TakePictureActivity

save(View v)

```
else{

   ImageModel imageModel = new ImageModel(imageTitle,
app.getUsername(), imageDescription, pathToImage,
new Date().getTime());

  ArrayList<ImageModel> imageList = app.getImages();
  imageList.add(0, imageModel);
  app.setImages(imageList);
```

TakePictureActivity

save(View v)

```
else{

   ImageModel imageModel = new ImageModel(imageTitle,
app.getUsername(), imageDescription, pathToImage,
new Date().getTime());

  ArrayList<ImageModel> imageList = app.getImages();
  imageList.add(0, imageModel);
  app.setImages(imageList);
```

TakePictureActivity

save(View v)

```
else{

   ImageModel imageModel = new ImageModel(imageTitle,
app.getUsername(), imageDescription, pathToImage,
new Date().getTime());

  ArrayList<ImageModel> imageList = app.getImages();
  imageList.add(0, imageModel);
  app.setImages(imageList);

  Intent intent = new Intent(TakePictureActivity.this,
  GalleryActivity.class);
```

TakePictureActivity

save(View v)

```
else{

   ImageModel imageModel = new ImageModel(imageTitle,
app.getUsername(), imageDescription, pathToImage,
new Date().getTime());

  ArrayList<ImageModel> imageList = app.getImages();
  imageList.add(0, imageModel);
  app.setImages(imageList);

  Intent intent = new Intent(TakePictureActivity.this,
  GalleryActivity.class);
  startActivity(intent);
```

TakePictureActivity

save(View v)

```
else{

   ImageModel imageModel = new ImageModel(imageTitle,
app.getUsername(), imageDescription, pathToImage,
new Date().getTime());

  ArrayList<ImageModel> imageList = app.getImages();
  imageList.add(0, imageModel);
  app.setImages(imageList);

  Intent intent = new Intent(TakePictureActivity.this,
  GalleryActivity.class);
  startActivity(intent);
  finish();
```

TakePictureActivity

save(View v)

```
else{

   ImageModel imageModel = new ImageModel(imageTitle,
app.getUsername(), imageDescription, pathToImage,
new Date().getTime());

  ArrayList<ImageModel> imageList = app.getImages();
  imageList.add(0, imageModel);
  app.setImages(imageList);

  Intent intent = new Intent(TakePictureActivity.this,
  GalleryActivity.class);
  startActivity(intent);
  finish();

}
```

AndroidManifest

Preparing for GalleryActivity

AndroidManifest

Preparing for GalleryActivity

```
<activity
```

AndroidManifest

Preparing for GalleryActivity

```
<activity
  android:name=".views.GalleryActivity"
```

AndroidManifest

Preparing for GalleryActivity

```
<activity
  android:name=".views.GalleryActivity"

  android:configChanges="keyboardHidden|orientation"
```

AndroidManifest

Preparing for GalleryActivity

```
<activity
  android:name=".views.GalleryActivity"

  android:configChanges="keyboardHidden|orientation"

  android:screenOrientation="portrait"/>
```

gallery.xml

Add a ListView container

gallery.xml

Add a ListView container

```
<ListView android:layout_height="fill_parent"
```

gallery.xml

Add a ListView container

```
<ListView android:layout_height="fill_parent"
          android:layout_width="fill_parent"
```

gallery.xml

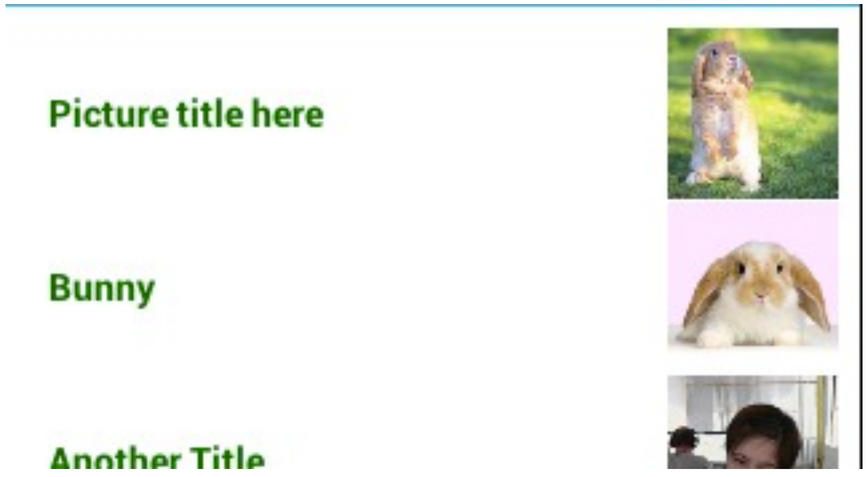
Add a ListView container

```
<ListView android:layout_height="fill_parent"
          android:layout_width="fill_parent"
          android:id="@+id/imageListView"/>
```

gallery.xml

Add a ListView container

```
<ListView android:layout_height="fill_parent"
          android:layout_width="fill_parent"
          android:id="@+id/imageListView"/>
```



Picture title here

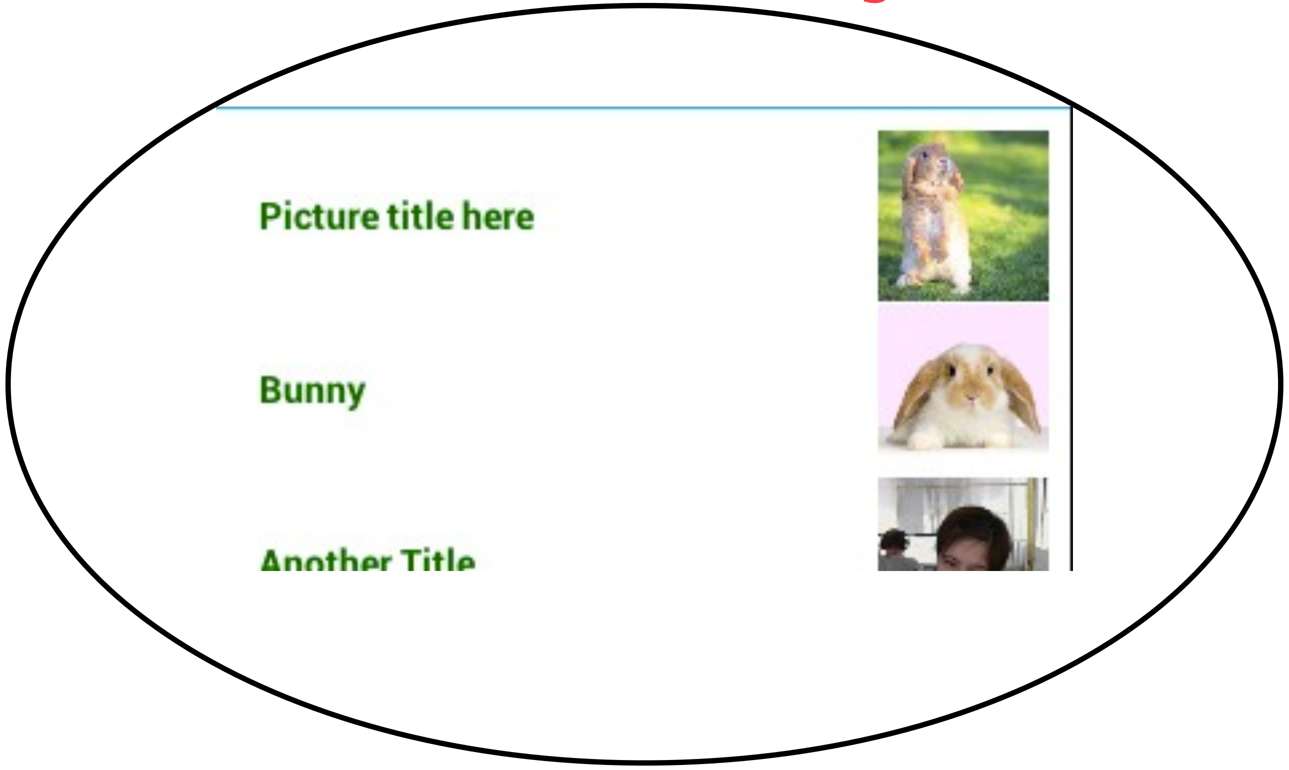
Bunny

Another Title

gallery.xml

Add a ListView container

```
<ListView android:layout_height="fill_parent"
          android:layout_width="fill_parent"
          android:id="@+id/imageListView"/>
```

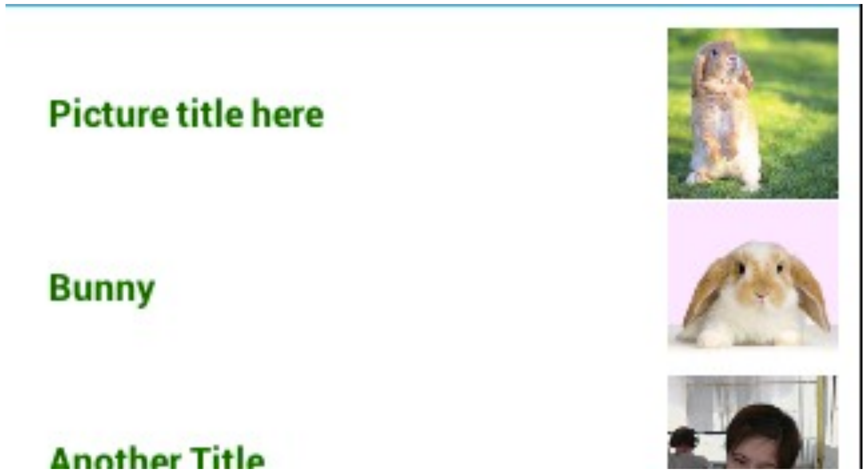


image_list_cell.xml

Build the layout for each item

image_list_cell.xml

Build the layout for each item



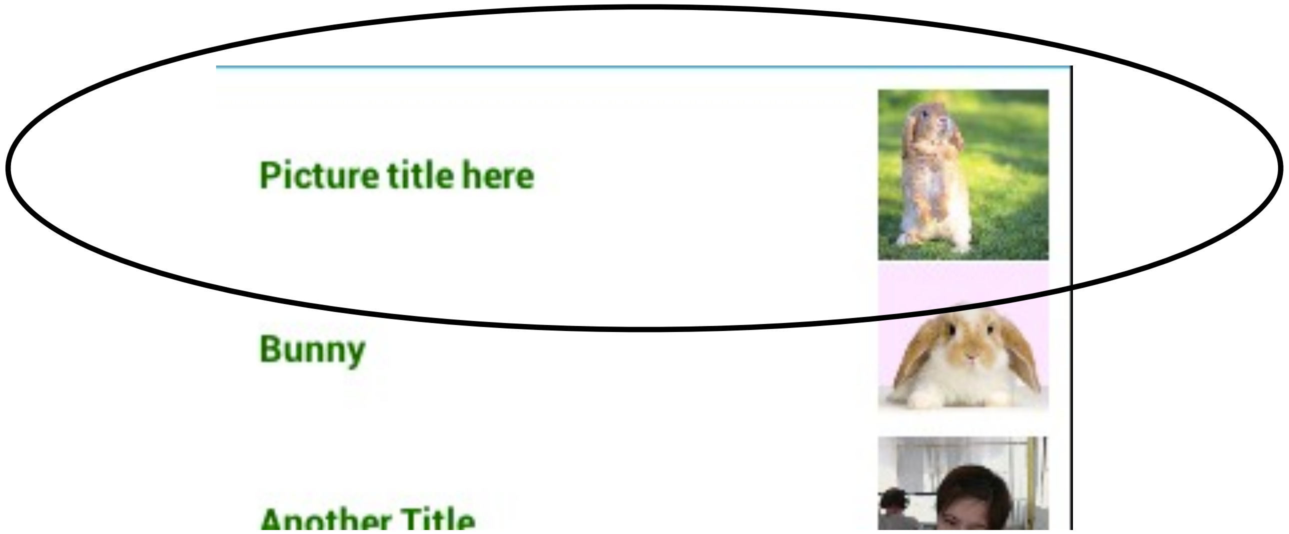
Picture title here

Bunny

Another Title

image_list_cell.xml

Build the layout for each item



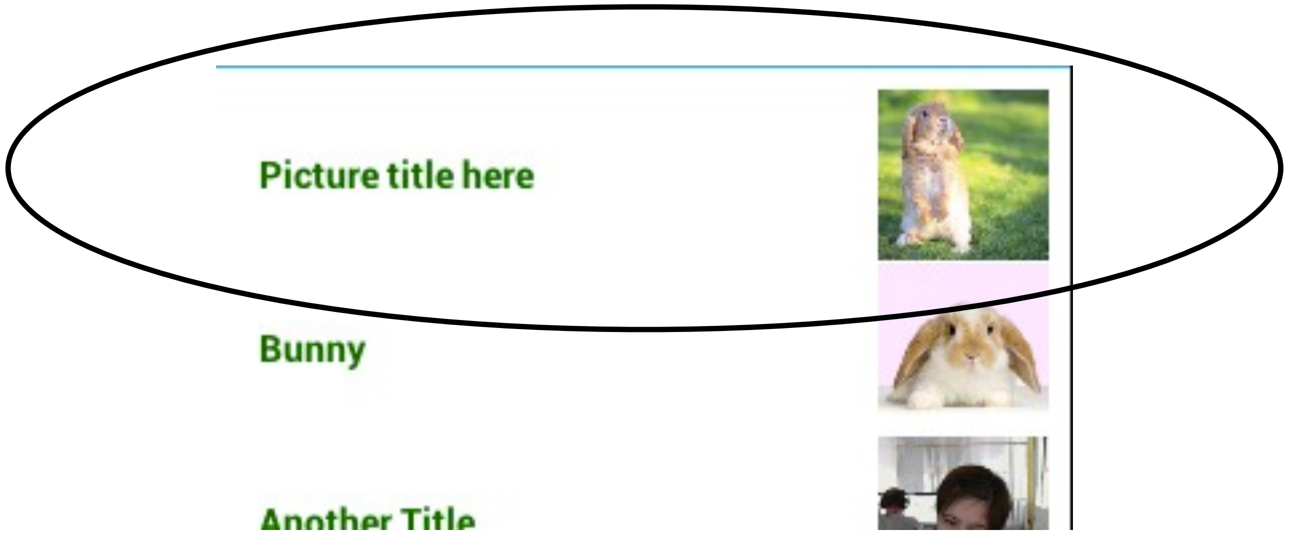
Picture title here

Bunny

Another Title

image_list_cell.xml

Build the layout for each item



Picture title here

Bunny

Another Title

TextView that is centered vertically

ImageView that shrinks and crops the image to size

image_list_cell.xml

Build the layout for each item

image_list_cell.xml

Build the layout for each item

```
<TextView
```

image_list_cell.xml

Build the layout for each item

```
<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:textSize="14dip"
```

image_list_cell.xml

Build the layout for each item

```
<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:textSize="14dip"
  android:text="Image Title Here"
  android:textStyle="bold"
```

image_list_cell.xml

Build the layout for each item

```
<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:textSize="14dip"
  android:text="Image Title Here"
  android:textStyle="bold"
  android:ellipsize="end"
```

image_list_cell.xml

Build the layout for each item

```
<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:textSize="14dip"
  android:text="Image Title Here"
  android:textStyle="bold"
  android:ellipsize="end"
  android:id="@+id/imageTitle"
  android:gravity="left|center_vertical"
  android:textColor="@color/GREEN"
```

image_list_cell.xml

Build the layout for each item

```
<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:textSize="14dip"
  android:text="Image Title Here"
  android:textStyle="bold"
  android:ellipsize="end"
  android:id="@+id/imageTitle"
  android:gravity="left|center_vertical"
  android:textColor="@color/GREEN"
  android:layout_weight="1" />
```

image_list_cell.xml

Build the layout for each item

image_list_cell.xml

Build the layout for each item

```
<ImageView
```

image_list_cell.xml

Build the layout for each item

```
<ImageView
   android:id="@+id/imageThumbnail"
```

image_list_cell.xml

Build the layout for each item

```
<ImageView
   android:id="@+id/imageThumbnail"
   android:layout_width="64dip"
```

image_list_cell.xml

Build the layout for each item

```
<ImageView
   android:id="@+id/imageThumbnail"
   android:layout_width="64dip"
   android:layout_height="64dip"
```

image_list_cell.xml

Build the layout for each item

```
<ImageView
   android:id="@+id/imageThumbnail"
   android:layout_width="64dip"
   android:layout_height="64dip"
   android:scaleType="centerCrop" />
```

ImageListAdapter

Adapter that builds each row in our list

ImageListAdapter

Adapter that builds each row in our list

```
private ArrayList<ImageModel> listItems;
```

ImageListAdapter

Adapter that builds each row in our list

```
private ArrayList<ImageModel> listItems;
private final Context context;
```

ImageListAdapter

Adapter that builds each row in our list

```
private ArrayList<ImageModel> listItems;
private final Context context;
private LayoutInflater inflater;
```

ImageListAdapter

Adapter that builds each row in our list

```
private ArrayList<ImageModel> listItems;
private final Context context;
private LayoutInflater inflater;


public ImageListAdapter(Context context,
ArrayList<ImageModel> listItems, LayoutInflater
inflater) {
```

ImageListAdapter

Adapter that builds each row in our list

```
private ArrayList<ImageModel> listItems;
private final Context context;
private LayoutInflater inflater;


public ImageListAdapter(Context context,
ArrayList<ImageModel> listItems, LayoutInflater
inflater) {
   this.context = context;
```

ImageListAdapter

Adapter that builds each row in our list

```
private ArrayList<ImageModel> listItems;
private final Context context;
private LayoutInflater inflater;


public ImageListAdapter(Context context,
ArrayList<ImageModel> listItems, LayoutInflater
inflater) {
   this.context = context;
   this.listItems = listItems;
```

ImageListAdapter

Adapter that builds each row in our list

```
private ArrayList<ImageModel> listItems;
private final Context context;
private LayoutInflater inflater;


public ImageListAdapter(Context context,
ArrayList<ImageModel> listItems, LayoutInflater
inflater) {
    this.context = context;
    this.listItems = listItems;
    this.inflater = inflater;
```

ImageListAdapter

Adapter that builds each row in our list

```
private ArrayList<ImageModel> listItems;
private final Context context;
private LayoutInflater inflater;


public ImageListAdapter(Context context,
ArrayList<ImageModel> listItems, LayoutInflater
inflater) {
   this.context = context;
   this.listItems = listItems;
   this.inflater = inflater;
}
```

ImageListAdapter

Adapter that builds each row in our list

ImageListAdapter

Adapter that builds each row in our list

```
    @Override
```

ImageListAdapter

Adapter that builds each row in our list

```java
@Override
public int getCount() {
```

ImageListAdapter

Adapter that builds each row in our list

```
@Override
public int getCount() {
  return listItems.size();
```

ImageListAdapter

Adapter that builds each row in our list

```
@Override
public int getCount() {
  return listItems.size();
}
```

ImageListAdapter

Adapter that builds each row in our list

```
@Override
public int getCount() {
  return listItems.size();
}

@Override
```

ImageListAdapter

Adapter that builds each row in our list

```
@Override
public int getCount() {
  return listItems.size();
}

@Override
public Object getItem(int position) {
```

ImageListAdapter

Adapter that builds each row in our list

```
@Override
public int getCount() {
  return listItems.size();
}

@Override
public Object getItem(int position) {
   return listItems.get(position);
```

ImageListAdapter

Adapter that builds each row in our list

```
@Override
public int getCount() {
  return listItems.size();
}

@Override
public Object getItem(int position) {
   return listItems.get(position);
}
```

ImageListAdapter

Adapter that builds each row in our list

```
@Override
public int getCount() {
  return listItems.size();
}

@Override
public Object getItem(int position) {
   return listItems.get(position);
}

@Override
```

ImageListAdapter

Adapter that builds each row in our list

```
@Override
public int getCount() {
  return listItems.size();
}

@Override
public Object getItem(int position) {
   return listItems.get(position);
}

@Override
public long getItemId(int position) {
```

ImageListAdapter

Adapter that builds each row in our list

```
@Override
public int getCount() {
  return listItems.size();
}

@Override
public Object getItem(int position) {
   return listItems.get(position);
}

@Override
public long getItemId(int position) {
   return position;
```

ImageListAdapter

Adapter that builds each row in our list

```
@Override
public int getCount() {
  return listItems.size();
}

@Override
public Object getItem(int position) {
   return listItems.get(position);
}

@Override
public long getItemId(int position) {
   return position;
}
```

ImageListAdapter

Adapter that builds each row in our list

ImageListAdapter

Adapter that builds each row in our list

```
@Override
```

ImageListAdapter

Adapter that builds each row in our list

```
@Override
public View getView(int position, View convertView, ViewGroup
parent) {
```

ImageListAdapter

Adapter that builds each row in our list

```
@Override
public View getView(int position, View convertView, ViewGroup
parent) {
```

ImageListAdapter

Adapter that builds each row in our list

```
@Override
public View getView(int position, View convertView, ViewGroup
parent) {

  ImageModel imageModel = listItems.get(position);
```

ImageListAdapter

Adapter that builds each row in our list

```
@Override
public View getView(int position, View convertView, ViewGroup
parent) {

  ImageModel imageModel = listItems.get(position);
```

ImageListAdapter

Adapter that builds each row in our list

```
@Override
public View getView(int position, View convertView, ViewGroup
parent) {

  ImageModel imageModel = listItems.get(position);

  View newView = inflater.inflate(R.layout.image_list_cell,
  null);
```

ImageListAdapter

Adapter that builds each row in our list

```
@Override
public View getView(int position, View convertView, ViewGroup
parent) {

  ImageModel imageModel = listItems.get(position);

  View newView = inflater.inflate(R.layout.image_list_cell,
  null);

  ImageView imageThumbnail = (ImageView)
```

ImageListAdapter

Adapter that builds each row in our list

```
@Override
public View getView(int position, View convertView, ViewGroup
parent) {

  ImageModel imageModel = listItems.get(position);

  View newView = inflater.inflate(R.layout.image_list_cell,
  null);

  ImageView imageThumbnail = (ImageView)
       newView.findViewById(R.id.imageThumbnail);
```

ImageListAdapter

Adapter that builds each row in our list

```
@Override
public View getView(int position, View convertView, ViewGroup
parent) {

  ImageModel imageModel = listItems.get(position);

  View newView = inflater.inflate(R.layout.image_list_cell,
  null);

  ImageView imageThumbnail = (ImageView)
       newView.findViewById(R.id.imageThumbnail);

  TextView imageTitle = (TextView)
```

ImageListAdapter

Adapter that builds each row in our list

```
@Override
public View getView(int position, View convertView, ViewGroup
parent) {

  ImageModel imageModel = listItems.get(position);

  View newView = inflater.inflate(R.layout.image_list_cell,
  null);

  ImageView imageThumbnail = (ImageView)
       newView.findViewById(R.id.imageThumbnail);

  TextView imageTitle = (TextView)
     newView.findViewById(R.id.imageTitle);
```

ImageListAdapter

Adapter that builds each row in our list

ImageListAdapter

Adapter that builds each row in our list

```
String pathToImage = imageModel.getPathToImage();
```

ImageListAdapter

Adapter that builds each row in our list

```
String pathToImage = imageModel.getPathToImage();
File selFile=new File(pathToImage);
```

ImageListAdapter

Adapter that builds each row in our list

```
String pathToImage = imageModel.getPathToImage();
File selFile=new File(pathToImage);
Bitmap thumbnailBmp = Utils.decodeFile(selFile);
```

ImageListAdapter

Adapter that builds each row in our list

```
String pathToImage = imageModel.getPathToImage();
File selFile=new File(pathToImage);
Bitmap thumbnailBmp = Utils.decodeFile(selFile);
imageThumbnail.setImageBitmap(thumbnailBmp);
```

ImageListAdapter

Adapter that builds each row in our list

```
String pathToImage = imageModel.getPathToImage();
File selFile=new File(pathToImage);
Bitmap thumbnailBmp = Utils.decodeFile(selFile);
imageThumbnail.setImageBitmap(thumbnailBmp);
 imageThumbnail.setScaleType(ImageView.ScaleType.
```

ImageListAdapter

Adapter that builds each row in our list

```
String pathToImage = imageModel.getPathToImage();
File selFile=new File(pathToImage);
Bitmap thumbnailBmp = Utils.decodeFile(selFile);
imageThumbnail.setImageBitmap(thumbnailBmp);
 imageThumbnail.setScaleType(ImageView.ScaleType.
 CENTER_CROP);
```

ImageListAdapter

Adapter that builds each row in our list

```
String pathToImage = imageModel.getPathToImage();
File selFile=new File(pathToImage);
Bitmap thumbnailBmp = Utils.decodeFile(selFile);
imageThumbnail.setImageBitmap(thumbnailBmp);
 imageThumbnail.setScaleType(ImageView.ScaleType.
 CENTER_CROP);

 imageTitle.setText(imageModel.getTitle());
```

ImageListAdapter

Adapter that builds each row in our list

```
String pathToImage = imageModel.getPathToImage();
File selFile=new File(pathToImage);
Bitmap thumbnailBmp = Utils.decodeFile(selFile);
imageThumbnail.setImageBitmap(thumbnailBmp);
imageThumbnail.setScaleType(ImageView.ScaleType.
CENTER_CROP);

imageTitle.setText(imageModel.getTitle());

return newView;
```

ImageListAdapter

Adapter that builds each row in our list

```
    String pathToImage = imageModel.getPathToImage();
    File selFile=new File(pathToImage);
    Bitmap thumbnailBmp = Utils.decodeFile(selFile);
    imageThumbnail.setImageBitmap(thumbnailBmp);
     imageThumbnail.setScaleType(ImageView.ScaleType.
     CENTER_CROP);

     imageTitle.setText(imageModel.getTitle());

    return newView;
}
```

GalleryActivity

onCreate()

GalleryActivity

onCreate()

```
super.onCreate(savedInstanceState);
```

GalleryActivity

onCreate()

```
super.onCreate(savedInstanceState);

setContentView(R.layout.gallery);
```

GalleryActivity

onCreate()

```
super.onCreate(savedInstanceState);

setContentView(R.layout.gallery);

this.app = (GirlDevelopIt)getApplicationContext();
```

GalleryActivity

onCreate()

```
super.onCreate(savedInstanceState);

setContentView(R.layout.gallery);

this.app = (GirlDevelopIt)getApplicationContext();

initElements();
```

GalleryActivity

onCreate()

```
super.onCreate(savedInstanceState);

setContentView(R.layout.gallery);

this.app = (GirlDevelopIt)getApplicationContext();

initElements();

populateImagesList();
```

GalleryActivity

initElements()

GalleryActivity

initElements()

```
imageListView = (ListView)
```

GalleryActivity

initElements()

```
imageListView = (ListView)
this.findViewById(R.id.imageListView);
```

GalleryActivity

populateImagesList()

GalleryActivity

populateImagesList()

```
LayoutInflater mLInflater = (LayoutInflater)
this.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
```

GalleryActivity

populateImagesList()

```
LayoutInflater mLInflater = (LayoutInflater)
this.getSystemService(Context.LAYOUT_INFLATER_SERVICE);

ImageListAdapter listAdapter = new ImageListAdapter(this,
app.getImages(), mLInflater);
```

GalleryActivity

populateImagesList()

```
LayoutInflater mLInflater = (LayoutInflater)
this.getSystemService(Context.LAYOUT_INFLATER_SERVICE);

ImageListAdapter listAdapter = new ImageListAdapter(this,
app.getImages(), mLInflater);

imageListView.setAdapter(listAdapter);
```

GalleryActivity

populateImagesList()

```
LayoutInflater mLInflater = (LayoutInflater)
this.getSystemService(Context.LAYOUT_INFLATER_SERVICE);

ImageListAdapter listAdapter = new ImageListAdapter(this,
app.getImages(), mLInflater);

imageListView.setAdapter(listAdapter);

imageListView.invalidateViews();
```

GalleryActivity

populateImagesList()

```
LayoutInflater mLInflater = (LayoutInflater)
this.getSystemService(Context.LAYOUT_INFLATER_SERVICE);

ImageListAdapter listAdapter = new ImageListAdapter(this,
app.getImages(), mLInflater);

imageListView.setAdapter(listAdapter);

imageListView.invalidateViews();

((BaseAdapter)imageListView.getAdapter()).notifyDataSetChange
d();
```

Next Week!

Next Week!

Adding the last Activity

Next Week!

Adding the last Activity

Explaining the mystery of the app variable

Next Week!

- Adding the last Activity
- Explaining the mystery of the app variable
- Next steps

Questions?