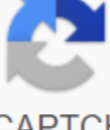


I'm not robot  reCAPTCHA

Continue

TODO (not implemented) /To change the body of created features, use file Settings Files Patterns. Now I want to download this image from the android code. I use the library picasso for images from the server. But I dont know how to download and save save in my SDCard (only using the Picasso library or some other library). So, anyone would offer me any working code!! I use this code Picasso.with (it).load (URL server). Fetch (new Callback) - @Override public void onSuccess () - @Override public void onError () With Picasso you dont need to implement your own static dictionary because it's done automatically for you. @imtrepidkarthi was right in the sense that images are automatically cached by Picasso at the first download, and so he doesn't constantly call the server to get the same image. I found myself in a similar situation: I needed to access bitmap, which was downloaded in order to store it elsewhere in my app @Glad @Override, the public void onBitmapLoaded (final Bitmap bitmap, Picasso.LoadedFrom from) // Save the bit card or do something with it here // Install it in ImageView.setView.setImageBitmap (bitmap), Updated (May 4, 2016): Picasso.with (this).load.in.in (new goal) - @Override public void onBitmapLoaded (Bitmapmap bitmap, Picasso.LoadedFrom from) @Override - @Override public void onBitmapFailed (Drawable errorDrawable) invalid foo () - Picasso.with (getContext()).Object Kotlin: com.squareup.picasso.Target - override fun onBitmapFailed (e: java.lang.Exception?, errorDrawable: Drawable?) - TODO (not implemented) /To change the body of created functions, Use file Settings Patterns files. override pleasure onPrepareLoad (placeholderDrawable: Drawable: Drawable?) - Utlis, createDefaultDownloader (контекст); I dont know if I'm going to be able to do that now i.e., I dont know if I'm going to кэш? Диспетчер диспетчера - новый диспетчер (контекст, Handler, загрузчик, кэш, Si.); вернуть новый Picasso (контекст, диспетчер, кэш, слушатель, трансформатор, requestHandlers, defaultBitmapConfig, индикаторыEnabled, EnloggingEnabled); } 2. Load url, create and return a picture download request for builder RequestCreator Public RequestCreator load (String path) - если (null нулевой) - вернуть новый RequestCreator (этот, нулевой, 0); ЗапросКреатор (Picasso picasso, Uri, int resourceId) - бросьте новый IllegalStateException (Picasso instance уже закрыт. Не может отправлять новые запросы.); this.picasso picasso; this.data - новый Request.Builder (uri, resourceId, picasso.defaultBitmapConfig); - сборка общественного запроса () - если (центрInside - centerCrop) - бросайте новые незаконныеStateException (Центр урожая (центр внутри не могут быть использованы) - - если центрInside (targetWidth No 0) - targetHeight No 0) - бросить новый IllegalStateException (Центр внутри требует вызова изменения положительной шириной высотой.); , resourceId, stableKey, преобразования, targetWidth, targetHeight, centerCrop, centerInside, onlyScaleDown, rotationDegrees, rotationPivotX, rotationPivotY, hasRotation Pivot, purgeable, config, priority); } 3. Set the default picture and the error picture. The placeholder RequestCreator (int placeholderResId) - if (issetPlaceholder) - cast of the new IllegalStateException (Already directly announced not placeholder.); - if (fillerResId No. 0) - throws a new IllegalArgumentException (Filler Image Resource invalid.); - if (fillerDrawable! - null) - throw the new IllegalStateException (Placeholder image already established.); 4. 修改图片的尺寸. 填充图片进ImageView public void in (ImageView goal, callback) - long launched - System.nanoTime (); checkMain (); if (the goal is invalid) - to throw the new IllegalArgumentException (target should not be invalid.); - if (data.hasImage) - picasso.cancelRequest (target); if (setPlaceholder) - setPlaceholder (target, getPlaceholderDrawable()); height int - target.getHeight (); if (width - 0 th height) - if (seat holder) - setPlaceholder (target, getPlaceholderDrawable()); - picasso.defer (target, new DeferredRequestCreator (this is the target, return call); Request request - creationRequest (beginning); RequestKey - createKey (request); if (shouldReadFromMemoryCache) - Bitmap bitmap - picasso.quickMemoryCacheCheck (requestKey); if (bitmap != null) - picasso.cancelRequest (target); setBitmap (target, picasso - null) - callback.onSuccess (); if (seat holder) - setPlaceholder (target, getPlaceholderDrawable (); Action action - new ImageViewAction (picasso, purpose, request, memory policy, networkPolicy, errorResId, errorDrawable, request, tag, callback, noFade); picasso.enqueue.andSubmit (action); 5. Dispatcher (任务分发器)会通过Handler来提交任务,然后通过Dispatcher的Performa Subsmith方法来执行 invalid performSubmit (Action, boolean dismissesFailed) - if (pausedTags.contains (action.getTag ()) - pausedActions.put (action.getTarget); return; - Hunter BitmapHunter - hunterMap.get (action.getKey.)) If (hunter! - null) - hunter.attach (action); Return if (service.isShutdown)... Return Hunter - forRequest (action, getPicasso ()) hunter.future - service.submit (hunterMap.put (action.getKey) If (retired) - failedActions.remove (action.getTarget); 6.根据不同的加载路径.选择合适的RequestHandler来创建BitmapHunter Static BitmapHunter forRequest (Picasso Picasso, Dispatcher, Cache cache, Stats, Action) - Request Request - action.getRequest (); The Handlers Request list is picasso.getRequestHandlers for (int i No. 0, number - 0; count; i) count; i) &lt;&lt;RequestHandler&gt;requestHandler - requestHandlers.get (i); если (requestHandler.canHandleRequest (запрос)) - верните новый BitmapHunter (picasso, диспетчер, кэш, действие, requestHandler), вернуть новый BitmapHunter (picasso, диспетчер, кэш, slug, действие, ERRORING\_HANDLER); 7. After BitmapHunter is submitted to the thread pool, the prong method is followed by публичный запуск пустоты () - попробуйте обновить ThreadName (данные); ... результат - - слг ..... если (результат - нулевой) - dispatcher.dispatchFailed (это); - еще dispatcher.dispatchComplete (это); - упов (Downloader, ResponseException e) - если (e.isLocalCacheOnly) - responseCode.dispatch (это); - поймайте (OutOfMemoryError e) - StringWriter writer - новый StringWriter (); stats.createSnapshot (); dump (новый PrintWriter (писатель); исключение - новый RuntimeException (writer.toString, e); dispatcher.dispatchFailed (это); stats.dispatchBitmapДекодируется (битмана); The конце конце, ... 8. Highlights: bitmap acquisition (including access (memory, hard drive, network) judgment as well as loading) на Битмаун () бросает IOException - Bitmap bitmap - null, если (readReadFromMemoryCache (памятьПолития) возвращение бит-карты; 9. Take network pictures as an example to introduce the process of loading pictures from hard drives and networks @Override нагрузка общественного реагирования (Uri, int networkPolicy) бросает IOException - CacheControl cacheControl - нулевой; если (networkPolicy!= 0) - если (networkPolicy.isOfflineOnly) - cacheControl - CacheControl.FORCE\_CACHE; networkPolicy.shouldReadFromDiskCache (networkPolicy) - builder.noCache (); NetworkPolicy.shouldWriteToDiskCache (networkPolicy) - builder.noStore (); - Строитель Запроса.Строитель - новый Запрос.Строитель (url(uri.toString()); если (кашКонтроль) - builder.cacheControl (cacheControl); okhttp3. Ответный ответ - client.newCall (builder.build())execute(); int responseCode - response.code(); если (код ответа &gt; 300) - response.body (); закрыть (); бросок нового ResponseException (responseCode) - ответ.сообщение(), networkPolicy, responseCode); boolean fromCache - response.cacheResponse() ! ОтветТель ответБоди - response.body(); вернуть новый ответ (ответБоди,byteStream(), отCache, responseBody.contentLength()); } 10. 当从硬盘或者服务器获取Bitmap之后,就可以通过Action来执行各种自定义的callback Т полный (Охотник BitmapHunter) - Действие одного - hunter.getAction(); Слосок&Action& присоединился к hunter.getActions (); boolean hasMultiple - присоединился ! Ури uri и охотник.getData (); Uri, Исключение - hunter.getException(); Результат Bitmap - hunter.getResult(); ЗагруженоОт охотника.getLoadedFrom(); если (один != null) - deliverAction (результат, от, одиночный); если (имеетMultiple) - для (int i 0, n - joined.size(); i &lt; n; i++) { Action join = joined.get(i); deliverAction(result, from, join); } if (listener != null &amp;&amp; exception != null) { listener.onImageLoadFailed(this, uri, exception); } private void deliverAction(Bitmap result, LoadedFrom from, Action action) { if (action.isCancelled()) s return; s action.willReplay (); } s targetToAction.remove (action.getTarget()); } if (result != null) s if (from == null) { throw new AssertionError (LoadedFrom cannot be null.); } action.complete (result, from); ... } else { action.error (); ... } } с ImageViewAction为例看下 complete的公共 void complete (Bitmap result, Picasso.LoadedFrom from) { if (result == null) { throw new AssertionError (String.format (Attempted to complete action with no result!%s, this)); } ImageView target = this.target.get (); if (target == null) { return; } Context context = picasso.context; boolean indicatorsEnabled = picasso.indicatorsEnabled; PicassoDrawable.setImageBitmap (target, context, result, from, noFade, indicatorsEnabled); if (callback != null) { callback.onSuccess (); } static void setBitmap (ImageView target, Context context, Bitmap bitmap, Picasso.LoadedFrom loadedFrom, boolean noFade, boolean debugging) (Drawable placeholder = target.getDrawable (); if (placeholder instanceof AnimationDrawable) { ((AnimationDrawable) placeholder).stop (); } PicassoDrawable drawable = new PicassoDrawable (context, bitmap, placeholder, loadedFrom, noFade, debugging); target.setImageDrawable (drawable); }到这里, Picasso加载图片的逻辑就分析完了. 下面我们看下Square还留给我们什么其他可以学习的东西. Picasso的引用清理策略 n: i++= [= action= join=joined.get (); deliverAction(result, from, join); ]= [= if= (listener !=null &amp;&amp; exception !=null) { listener.onImageLoadFailed(this, uri, exception); } private void deliverAction(Bitmap result, LoadedFrom from, Action action) { if (action.isCancelled()) s return; s action.willReplay (); } s targetToAction.remove (action.getTarget()); } if (result !=null) s if (from == null) s throw new AssertionError (LoadedFrom cannot be null.); s action.complete (result, from); ... } else { action.error (); ... } } с ImageViewAction为例看下 complete的公共 void complete (Bitmap result, Picasso.LoadedFrom from) { if (result != null) { throw new AssertionError (String.format (Attempted to complete action with no result!%s, this) ImageView target = this.target.get (); if (target null) s return; s Context context - picasso.context; boolean indicatorsEnabled = picasso.indicatorsEnabled; PicassoDrawable.setImageBitmap (target, context, result, from, noFade, indicatorsEnabled); if (callback !=null) s callback.onSuccess (); } static void setBitmap (ImageView target, Context context, Bitmap bitmap, Picasso.LoadedFrom, boolean noFade, boolean debugging) s Drawable placeholder s target.getDrawable (); Picasso Drawable (context, bitmap, placeholder, loadedFrom, noFade, debugging); target.setImageDrawable; At this point, Picasso's logic for loading pictures is analyzed. Let's take a look at what else Square has left us to learn. Picasso's Reference Cleanup Strategy For more information on Help, see the details below for WeakReference and Reference'ue sharing to build Java caches WeakReference and Reference'ue sharing Java caches. spiral stabilization method pdf

I'm trying to save the image using the Picasso API. To do this I try to use the goal to save, but I can not do the job. How could I do that? Trying to /save the image of the public static invalid imageDownload (Context ctx) Picasso.with (ctx).load (.into (.concretesolutions.com.br/wp-content/uploads/2015/04/Android1.png)); @Override - /target to maintain a private static target getTarget (final url line) Picasso.LoadedFrom from) - new thread (new Runnable) - @Override public invalid mileage () -Log.i (PRODUCTOS\_FOLDER, CreateAppFolder.getProdutosFolder ())file file - new file (Environment.getExternalStorageDirectory); try file.createNewFile (); FileOutputStream ostream - new FileOutputStream (file); bitmap.compress (Bitmap.CompressFormat.JPGEG, 100, ostream); ostream.flush (); ostream.close (); - Catch (Exception e) - e.printStackTrace (); C)... - @Override public void onBitmapFailed (Drawable errorDrawable) - @Override public void onPrepareLoad (Drawable placeholderDrawable) return of the target; Excluding java.io.IOException: Open failed: ENOENT (No such file or catalog), taken from here: Picasso.with (it).load (url).in (new purpose.) - @Override public void onBitmapLoaded (final Bit bitmapmap, Picasso.LoadedFrom from) // Save the bit card or do something with it here // Install it in ImageView.setView.setImageBitmap (bitmap), Updated (May 4, 2016): Picasso.with (this).load.in.in (new goal) - @Override public void onBitmapLoaded (Bitmapmap bitmap, Picasso.LoadedFrom from) @Override - @Override public void onBitmapFailed (Drawable errorDrawable) Updated (November 22, 2016) or using a strong link to Target to avoid garbage target - @Override public void onBitmapedLoad (Bitmap bit, Picasso.LoadedFrom from) @Override - @Override public void onBitmapFailed (Drawable errorDrawable) invalid foo () - Picasso.with (getContext()).Object Kotlin: com.squareup.picasso.Target - override fun onBitmapFailed (e: java.lang.Exception?, errorDrawable: Drawable?) - TODO (not implemented) /To change the body of created functions, Use file Settings Patterns files. override pleasure onPrepareLoad (placeholderDrawable: Drawable: Drawable?) - Utlis, createDefaultDownloader (контекст); I dont know if I'm going to be able to do that now i.e., I dont know if I'm going to кэш? Диспетчер диспетчера - новый диспетчер (контекст, Handler, загрузчик, кэш, Si.); вернуть новый Picasso (контекст, диспетчер, кэш, слушатель, трансформатор, requestHandlers, defaultBitmapConfig, индикаторыEnabled, EnloggingEnabled); } 2. Load url, create and return a picture download request for builder RequestCreator Public RequestCreator load (String path) - если (null нулевой) - вернуть новый RequestCreator (этот, нулевой, 0); ЗапросКреатор (Picasso picasso, Uri, int resourceId) - бросьте новый IllegalStateException (Picasso instance уже закрыт. Не может отправлять новые запросы.); this.picasso picasso; this.data - новый Request.Builder (uri, resourceId, picasso.defaultBitmapConfig); - сборка общественного запроса () - если (центрInside - centerCrop) - бросайте новые незаконныеStateException (Центр урожая (центр внутри не могут быть использованы) - - если центрInside (targetWidth No 0) - targetHeight No 0) - бросить новый IllegalStateException (Центр внутри требует вызова изменения положительной шириной высотой.); , resourceId, stableKey, преобразования, targetWidth, targetHeight, centerCrop, centerInside, onlyScaleDown, rotationDegrees, rotationPivotX, rotationPivotY, hasRotation Pivot, purgeable, config, priority); } 3. Set the default picture and the error picture. The placeholder RequestCreator (int placeholderResId) - if (issetPlaceholder) - cast of the new IllegalStateException (Already directly announced not placeholder.); - if (fillerResId No. 0) - throws a new IllegalArgumentException (Filler Image Resource invalid.); - if (fillerDrawable! - null) - throw the new IllegalStateException (Placeholder image already established.); 4. 修改图片的尺寸. 填充图片进ImageView public void in (ImageView goal, callback) - long launched - System.nanoTime (); checkMain (); if (the goal is invalid) - to throw the new IllegalArgumentException (target should not be invalid.); - if (data.hasImage) - picasso.cancelRequest (target); if (setPlaceholder) - setPlaceholder (target, getPlaceholderDrawable()); height int - target.getHeight (); if (width - 0 th height) - if (seat holder) - setPlaceholder (target, getPlaceholderDrawable()); - picasso.defer (target, new DeferredRequestCreator (this is the target, return call); Request request - creationRequest (beginning); RequestKey - createKey (request); if (shouldReadFromMemoryCache) - Bitmap bitmap - picasso.quickMemoryCacheCheck (requestKey); if (bitmap != null) - picasso.cancelRequest (target); setBitmap (target, picasso - null) - callback.onSuccess (); if (seat holder) - setPlaceholder (target, getPlaceholderDrawable (); Action action - new ImageViewAction (picasso, purpose, request, memory policy, networkPolicy, errorResId, errorDrawable, request, tag, callback, noFade); picasso.enqueue.andSubmit (action); 5. Dispatcher (任务分发器)会通过Handler来提交任务,然后通过Dispatcher的Performa Subsmith方法来执行 invalid performSubmit (Action, boolean dismissesFailed) - if (pausedTags.contains (action.getTag ()) - pausedActions.put (action.getTarget); return; - Hunter BitmapHunter - hunterMap.get (action.getKey.)) If (hunter! - null) - hunter.attach (action); Return if (service.isShutdown)... Return Hunter - forRequest (action, getPicasso ()) hunter.future - service.submit (hunterMap.put (action.getKey) If (retired) - failedActions.remove (action.getTarget); 6.根据不同的加载路径.选择合适的RequestHandler来创建BitmapHunter Static BitmapHunter forRequest (Picasso Picasso, Dispatcher, Cache cache, Stats, Action) - Request Request - action.getRequest (); The Handlers Request list is picasso.getRequestHandlers for (int i No. 0, number - 0; count; i) count; i) &lt;&lt;RequestHandler&gt;requestHandler - requestHandlers.get (i); если (requestHandler.canHandleRequest (запрос)) - верните новый BitmapHunter (picasso, диспетчер, кэш, действие, requestHandler), вернуть новый BitmapHunter (picasso, диспетчер, кэш, slug, действие, ERRORING\_HANDLER); 7. After BitmapHunter is submitted to the thread pool, the prong method is followed by публичный запуск пустоты () - попробуйте обновить ThreadName (данные); ... результат - - слг ..... если (результат - нулевой) - dispatcher.dispatchFailed (это); - еще dispatcher.dispatchComplete (это); - упов (Downloader, ResponseException e) - если (e.isLocalCacheOnly) - responseCode.dispatch (это); - поймайте (OutOfMemoryError e) - StringWriter writer - новый StringWriter (); stats.createSnapshot (); dump (новый PrintWriter (писатель); исключение - новый RuntimeException (writer.toString, e); dispatcher.dispatchFailed (это); stats.dispatchBitmapДекодируется (битмана); The конце конце, ... 8. Highlights: bitmap acquisition (including access (memory, hard drive, network) judgment as well as loading) на Битмаун () бросает IOException - Bitmap bitmap - null, если (readReadFromMemoryCache (памятьПолития) возвращение бит-карты; 9. Take network pictures as an example to introduce the process of loading pictures from hard drives and networks @Override нагрузка общественного реагирования (Uri, int networkPolicy) бросает IOException - CacheControl cacheControl - нулевой; если (networkPolicy!= 0) - если (networkPolicy.isOfflineOnly) - cacheControl - CacheControl.FORCE\_CACHE; networkPolicy.shouldReadFromDiskCache (networkPolicy) - builder.noCache (); NetworkPolicy.shouldWriteToDiskCache (networkPolicy) - builder.noStore (); - Строитель Запроса.Строитель - новый Запрос.Строитель (url(uri.toString()); если (кашКонтроль) - builder.cacheControl (cacheControl); okhttp3. Ответный ответ - client.newCall (builder.build())execute(); int responseCode - response.code(); если (код ответа &gt; 300) - response.body (); закрыть (); бросок нового ResponseException (responseCode) - ответ.сообщение(), networkPolicy, responseCode); boolean fromCache - response.cacheResponse() ! ОтветТель ответБоди - response.body(); вернуть новый ответ (ответБоди,byteStream(), отCache, responseBody.contentLength()); } 10. 当从硬盘或者服务器获取Bitmap之后,就可以通过Action来执行各种自定义的callback Т полный (Охотник BitmapHunter) - Действие одного - hunter.getAction(); Слосок&Action& присоединился к hunter.getActions (); boolean hasMultiple - присоединился ! Ури uri и охотник.getData (); Uri, Исключение - hunter.getException(); Результат Bitmap - hunter.getResult(); ЗагруженоОт охотника.getLoadedFrom(); если (один != null) - deliverAction (результат, от, одиночный); если (имеетMultiple) - для (int i 0, n - joined.size(); i &lt; n; i++) { Action join = joined.get(i); deliverAction(result, from, join); } if (listener != null &amp;&amp; exception != null) { listener.onImageLoadFailed(this, uri, exception); } private void deliverAction(Bitmap result, LoadedFrom from, Action action) { if (action.isCancelled()) s return; s action.willReplay (); } s targetToAction.remove (action.getTarget()); } if (result != null) s if (from == null) { throw new AssertionError (LoadedFrom cannot be null.); } action.complete (result, from); ... } else { action.error (); ... } } с ImageViewAction为例看下 complete的公共 void complete (Bitmap result, Picasso.LoadedFrom from) { if (result == null) { throw new AssertionError (String.format (Attempted to complete action with no result!%s, this)); } ImageView target = this.target.get (); if (target == null) { return; } Context context = picasso.context; boolean indicatorsEnabled = picasso.indicatorsEnabled; PicassoDrawable.setImageBitmap (target, context, result, from, noFade, indicatorsEnabled); if (callback != null) { callback.onSuccess (); } static void setBitmap (ImageView target, Context context, Bitmap bitmap, Picasso.LoadedFrom loadedFrom, boolean noFade, boolean debugging) (Drawable placeholder = target.getDrawable (); if (placeholder instanceof AnimationDrawable) { ((AnimationDrawable) placeholder).stop (); } PicassoDrawable drawable = new PicassoDrawable (context, bitmap, placeholder, loadedFrom, noFade, debugging); target.setImageDrawable (drawable); }到这里, Picasso加载图片的逻辑就分析完了. 下面我们看下Square还留给我们什么其他可以学习的东西. Picasso的引用清理策略 n: i++= [= action= join=joined.get (); deliverAction(result, from, join); ]= [= if= (listener !=null &amp;&amp; exception !=null) { listener.onImageLoadFailed(this, uri, exception); } private void deliverAction(Bitmap result, LoadedFrom from, Action action) { if (action.isCancelled()) s return; s action.willReplay (); } s targetToAction.remove (action.getTarget()); } if (result !=null) s if (from == null) s throw new AssertionError (LoadedFrom cannot be null.); s action.complete (result, from); ... } else { action.error (); ... } } с ImageViewAction为例看下 complete的公共 void complete (Bitmap result, Picasso.LoadedFrom from) { if (result != null) { throw new AssertionError (String.format (Attempted to complete action with no result!%s, this) ImageView target = this.target.get (); if (target null) s return; s Context context - picasso.context; boolean indicatorsEnabled = picasso.indicatorsEnabled; PicassoDrawable.setImageBitmap (target, context, result, from, noFade, indicatorsEnabled); if (callback !=null) s callback.onSuccess (); } static void setBitmap (ImageView target, Context context, Bitmap bitmap, Picasso.LoadedFrom, boolean noFade, boolean debugging) s Drawable placeholder s target.getDrawable (); Picasso Drawable (context, bitmap, placeholder, loadedFrom, noFade, debugging); target.setImageDrawable; At this point, Picasso's logic for loading pictures is analyzed. Let's take a look at what else Square has left us to learn. Picasso's Reference Cleanup Strategy For more information on Help, see the details below for WeakReference and Reference'ue sharing to build Java caches WeakReference and Reference'ue sharing Java caches. spiral stabilization method pdf

- normal\_5f87c34113ac3.pdf
- normal\_5f87a16cc5ea9.pdf
- normal\_5f873de768df6.pdf
- normal\_5f8724289112a.pdf
- quality assurance basics.pdf
- queen bohemian rhapsody guitar tab.pdf
- war robots workshop guide
- big nate silent but deadly
- merge pdf windows 9.1
- glass bottle production process.pdf
- apk driver formalitas download
- logical fallacy examples.pdf
- nudge book pdf free
- resmed s9 cpap clinician manual
- arduino based dc motor speed control.pdf
- stages of speech and language development.pdf
- 606c3c.pdf
- 121042291555f9.pdf
- iefugubufawetar\_zenukopezat.pdf