

日本Androidの会 四国支部 第1回勉強会 - 2009/07/18

*Attender*ができるまで

有山 圭二

日本Androidの会 メンバー

有限会社シーリス 代表

講演及び資料の内容は、有山圭二及び有限会社シーリスの見解であり、
「日本Androidの会」全体の意見を代表するものではありません。

本日のお題 – Attenderができるまで

- Attenderとは
- 開発のきっかけ
- 開発の流れ

Attenderとは

- 受付にあったアレ
 - 自動受付システム



開発のきっかけ

- ABC (Android Bazaar and Conference) の前の、まいむぞう氏 (北海道支部) のメールでの発言、

どこの勉強会でもそうですが、入り口前の受付で、プリントアウトされた参加者名と突き合わせを行う作業って結構渋滞しますよね。

自分も勉強会など運営する立場のものとして、なんとかならんかなあと思っていたところです。

ちょっと今回は間に合わないでしょうけど、個人的にシステム作ってみますね

参加者へのリマインダメールに、参加セミナー毎のID付きリンクを含めて、

リンクをクリックするとQR画像が表示される。

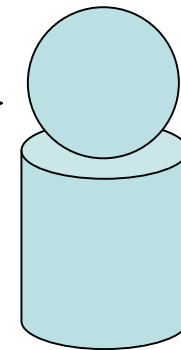
受付担当員はAndroid端末を使ってそのQRを読んで、

セミナー毎のIDと照合をかける、

みたいな感じでしょうか。

受付担当の人数は変わらないかも知れませんが、

ETCレーンみたいにすれば流れは良くなるでしょうね。



有山 『いいっすね』

ABC後、プロジェクトスタート

開発の流れ

- サーバー側のAPI
- データエンティティの作成
- サーバーインターフェースの設計
- Activityの設計 & 実装
- QRコードの読み取り

サーバー側のAPI(まいむぞう氏)

- APIの定義
 - 登録イベント一覧を取得
 - イベントの参加予定者一覧を取得
 - イベントの参加者をサーバーに登録
 - レスポンスで、最新の参加予定者一覧を返す
- レスポンスはXML

アプリを作ろう

データエンティティの定義

- イベント (EventData)
 - Serializable
- 参加者 (AttendeeData)
- 認証情報データ (AuthData)
 - Serializable

サーバインターフェースの設計

- イベント一覧を取得

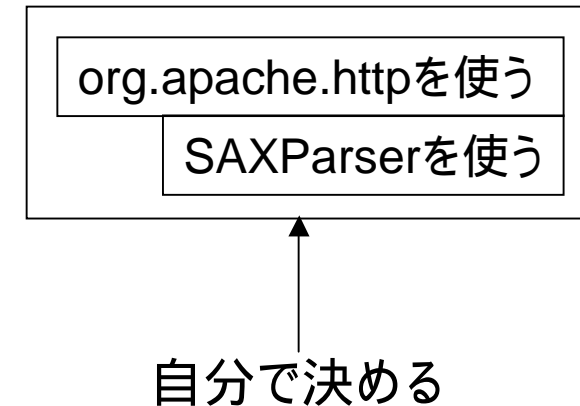
- List<EventData>
 getEventData(AuthData authData)

- イベントの参加予定者一覧を取得

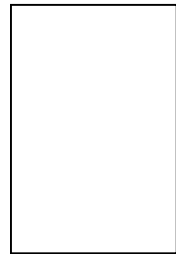
- HashMap<String, AttendeeData>
 getAttendeeData(
 AuthData authData,
 EventData eventData)

- イベントの参加者を登録

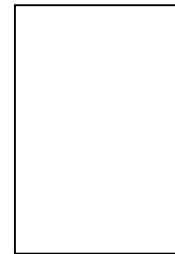
- HashMap<String,AttendeeData>
 uploadAttendeeData(
 AuthData authData,
 EventData eventData,
 HashMap<String,AttendeeData> attendeeDataList)



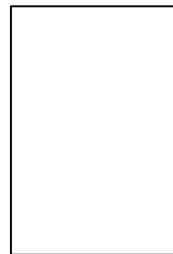
Activityの設計



EventListActivity



AttendeeListActivity



LoginActivity

EventListActivity

- アプリを起動して始めに表示される。
- 利用者に対応するイベントの出席予定者をサーバーから取得して表示する。
- イベントを選択すると、選択したイベントの出席予定者リストを表示する
(AttendeeListActivity起動)

AttendeeListActivity

- 利用者が選択したイベントの出席予定者を、サーバーから取得してリストで表示する。
- リストの行をタップすると、状態が「出席」に変更される。
- トラックボールをクリックすると、QRコードの読み取りモードに移行する。
 - QRコードを読み込み、得られたIDが出席予定者リストにあれば、当該出席予定者の状態を「出席」に変更する。
- メニューから「登録」を選択すると、その時点での出席者情報をサーバーへ送信、登録する。
- 間違えた人をタップした場合、長押しすれば取り消しをする事が出来る。
 - 但し、既にサーバーに送信してしまっている場合は、取り消しが出来ない。
 - サーバー側で取消の Protokol が追加されれば可能になる。
- 電話の着信やバックキーで戻った時に、出席登録者が居た場合、自動的にサーバーに送信する。

LoginActivity

- 他のActivityから必要に応じて呼び出される。
- 利用者が、サーバーにアクセスする際に必要となる認証情報を入力すると、情報をファイルに保存する。
- ログイン情報が設定(変更)された場合、呼び出し元に設定(変更)後の認証情報を返却する。

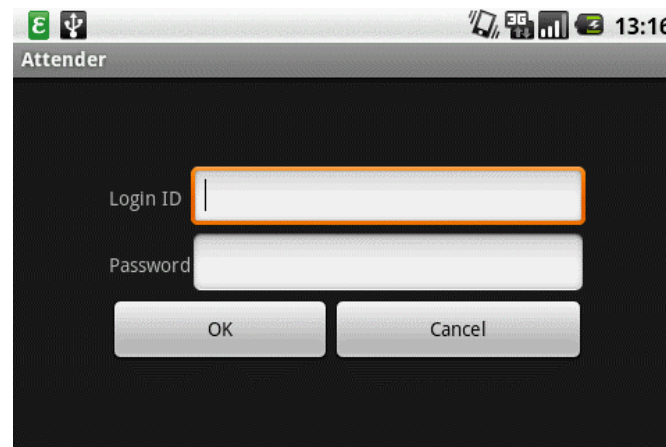
Activityの設計と実装



EventListActivity

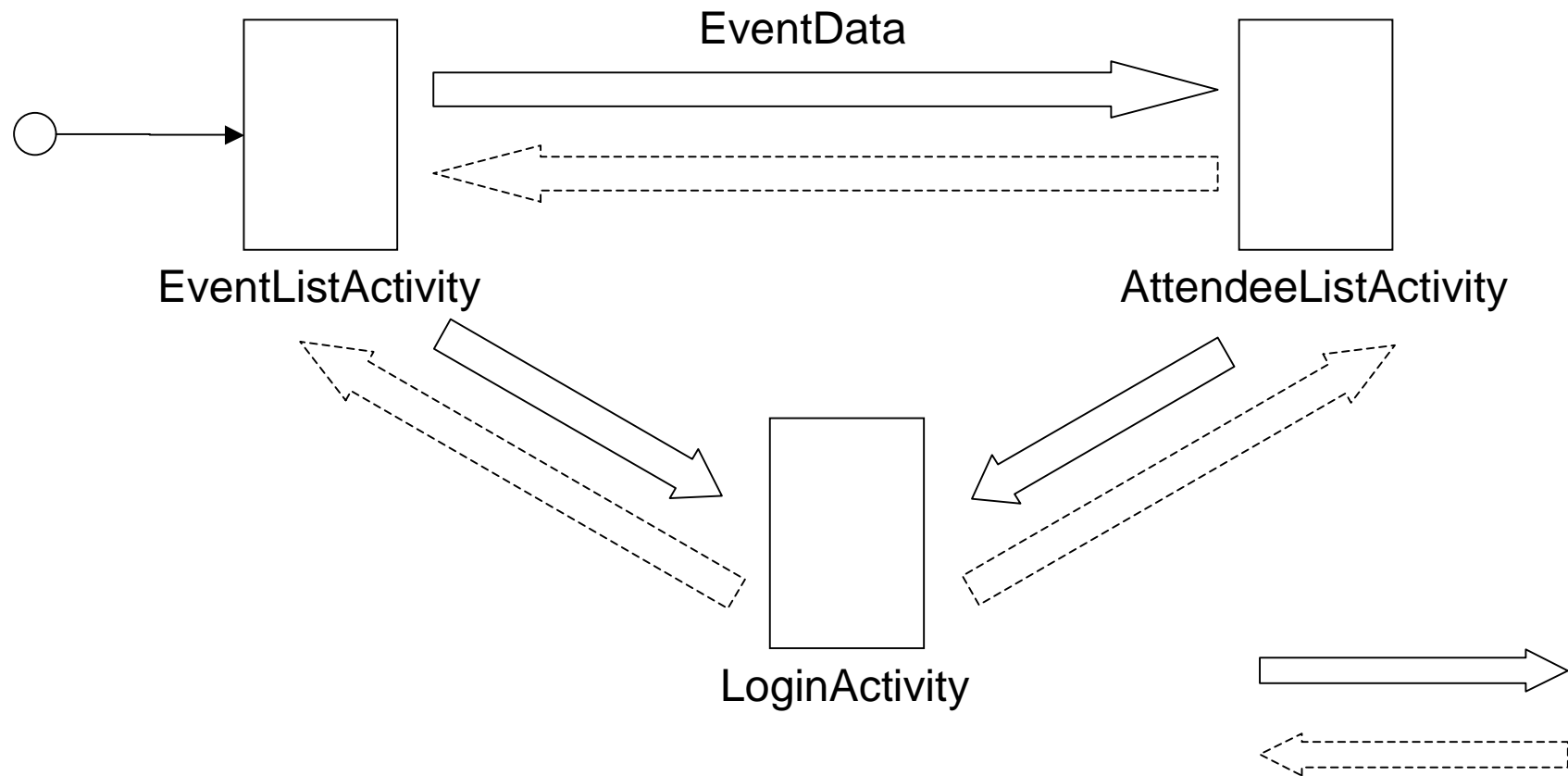


AttendeeListActivity

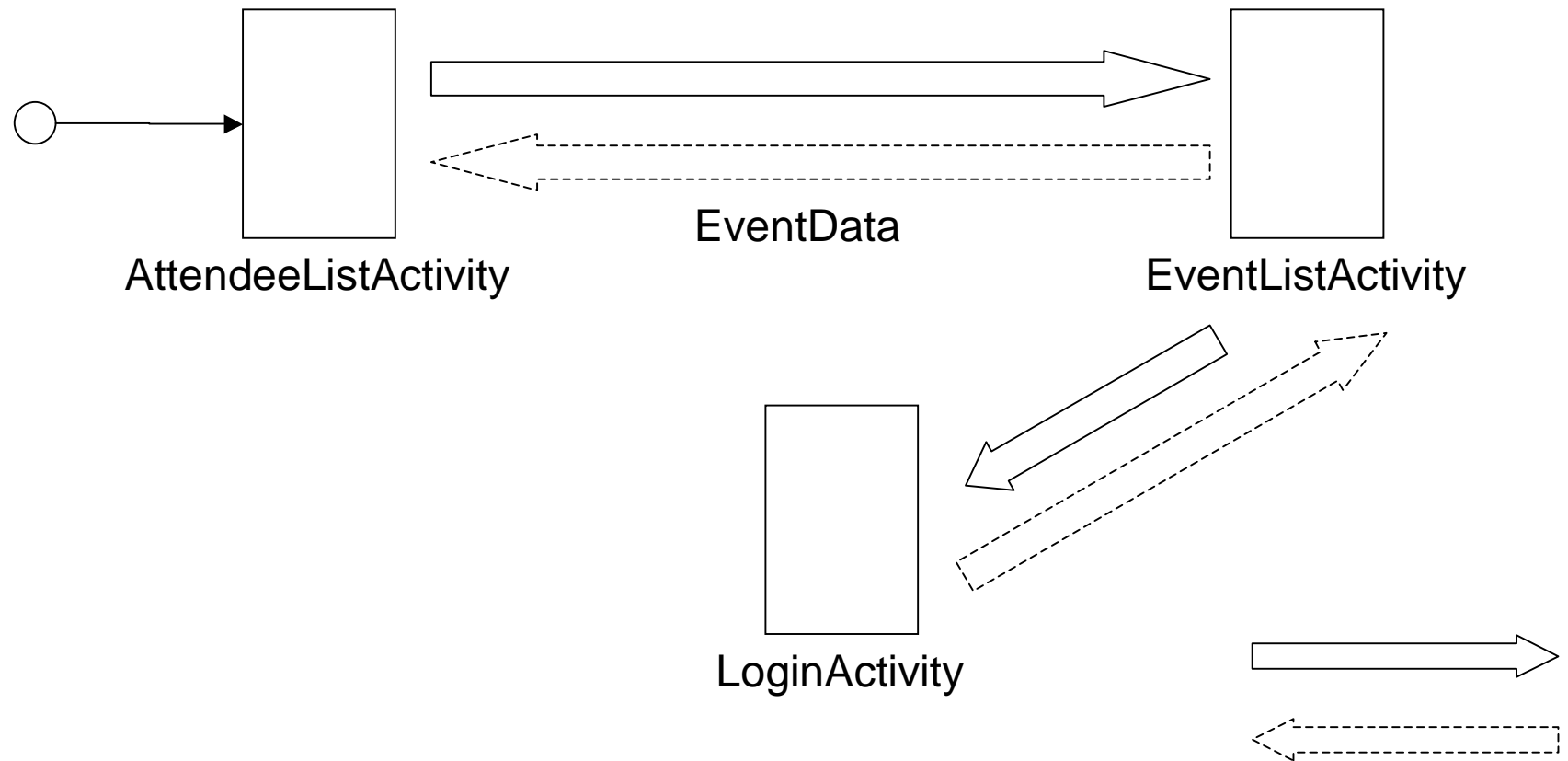


LoginActivity

Activity遷移の設計



直接呼ばれる場合を考える



画面の設計

- リスト1行分のレイアウトリソースを作成
 - 変更すべきウィジェットにはIDをつける
- リストアダプタを実装
 - BaseAdapterを継承したクラス
 - 内部的には、Listでデータを持つ
(AttendeeListActivityは、ArrayListで持つ)
 - View getView(
int position,
View convertView,
ViewGroup parent)
を、実装する。
- 出席予定者一覧は並び替えたい
 - Comparatorの切り替えで実現

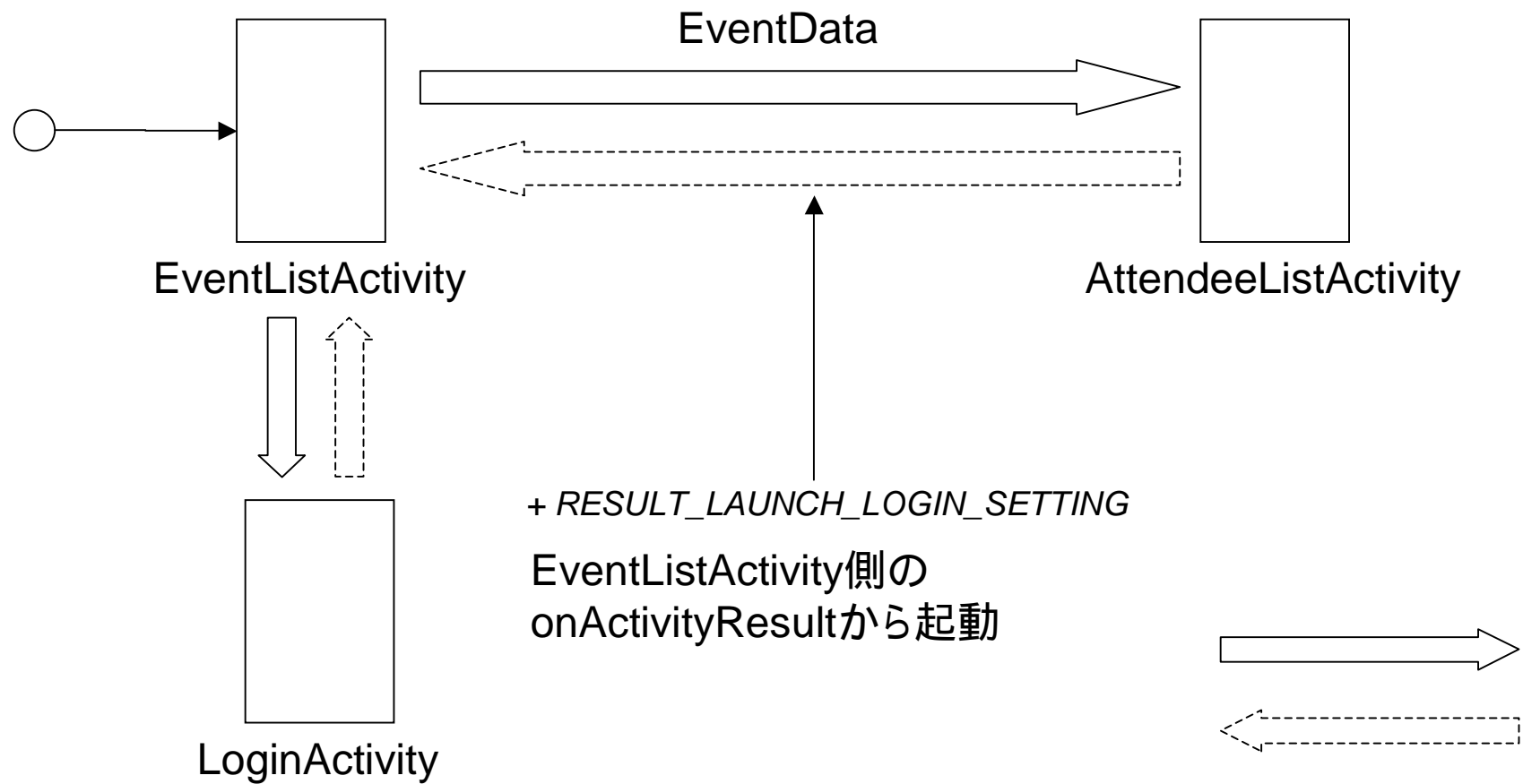
QRコードの読み取り

- Attenderでは実装しない、する必要がない
 - 面倒くさい
 - 外部アプリにお願いする事に決定
- Intentに外部アプリのActionを設定
 - “com.google.zxing.client.android.SCAN”
 - startActivityForResultで起動
- 結果をonActivityResultで受ける
 - RESULT_OKか、RESULT_CANCELEDか。
 - String getStringExtra("SCAN_RESULT");

Activityの遷移で迷った所

- AttendeeListActivityから、ログイン設定を変更した場合
 - LoginActivityは、AttendeeListActivityから起動される
 - LoginActivityが終了後、AttendeeListActivity側で反映しても、どのみちイベント情報一覧を再取得する必要がある。

結果として



Watch out!

- QRコードスキャナが、インストールされていないと、
 - ActivityNotFoundExceptionが発生
 - 必要なアプリは予め明示する必要がある
 - 可能であればアプリ側で、依存アプリの有無を検知して、Android Marketへ誘導する
- 通信関係のExceptionをどこで受け取る？
 - IOException
 - HttpResponseException
- 結果的にはActivityで処理した
 - Toastで表示

感想

- この規模のアプリなら、じっくり設計しなくても、形に出来るので楽しい
 - 総開発時間は、18時間程
- QRコードスキャナをIntentで手軽に呼び出す事が出来るのは嬉しい
 - かなりの工数削減

正に、「Fast & easy application development」

今後の課題

- QRコードスキャナがインストールされていない場合の処理を追加
- QRコード読み取り時、正常 / エラー発生に対応するサウンドを再生
- オフライン対応
 - 端末がネットに繋がっていなくても使えるよう、事前ダウンロード型のオフラインモードを搭載する。
- ユーザビリティの向上
 - サーバーへのアップロード処理の案内
- テストの継続
 - エラー修正

ご静聴ありがとうございました

講演及び資料の内容は、有山圭二及び有限会社シーリスの見解であり、「日本Androidの会」の意見を代表するものではありません。

このページの内容の一部は、[Google が作成、提供している](#)コンテンツを複製したもので、[クリエイティブ コモンズの表示 2.5 ライセンス](#)に記載の条件に従って使用しています。

最後に

- Attenderは、オープンソースで公開予定
– 修正BSDライセンス(予定)