・・・スマートにプログラミング

Android 入門編 第3版 SDK4.x 対応

本書の補足サイト

ここでは、読者の皆様から寄せられた質問の中で、比較的多く見受けられたトラブル事例 とその解決策や書籍刊行後の更新情報等を集め掲載致しました。 お困りの際の解決のヒントとしてご利用下さい。

> 株式会社リックテレコム 書籍出版部 (情報更新日:2014.4.21)

index

●ソースコードのインポートについて

(関連箇所:書籍全体)^{情報提供:2014/4/21}

● Macintosh における JDK の取り扱いの変更について

(関連箇所:「2-4 Android 仮想デバイスの設定)^{情報提供: 2013/7/11}

●「Google APIs」が見あたらない。

(関連箇所: 「2-3 SDK コンポーネントの追加) ^{情報提供: 2013/7/10}

●掲載内容の誤記等について

下記の URL のサイトをご参照下さい。 http://www.ric.co.jp/book/error/error895.html

この他にも、本書の著者が所属する Tech Fun 社のサイトに、プログラミング環境を構築するにあたって、 初心者が陥りやすい様々なケースに対し丁寧にアドバイスをしたコーナーがあります。 http://techfun.cc/techinf/setup/android 環境構築にあたってわからない所が出てきたら、こちらの方も参照してみて下さい。

ソースコードのインポートについて

●正規のインポート手順

本書のサンプルソースコードは、ダウンロードの上ご利用することは可能です。そしてソ ースコードを Eclipse にインポートする際には、下記の手順で行うようにして下さい。なお、 誤った操作手順についてはこちらを参照下さい。

1. パッケージ・エクスプローラー上を右クリックし、「インポート」を選択します(図1)。 図1



インポート画面で、「Android」→「Existing Android Code Into Workspace」を選択し、「次
 ヘ」ボタンをクリックします(図 2)。

凶2

	- 0 X
選択	Ľ
インボート・ソースの違択(5): フィルター入力	
(P) (P) 第7(F) キャン	tıl

3. プロジェクトのインポート画面で、「ルート・ディレクトリー」の「参照」 ボタンでインポート対象のプロジェクトファイルを選択し、「プロジェクトをワークスペースにコピー」 のチェックボックスにチェックし、「完了」ボタンをクリックします(図3)。これでソー スコードは正常にインポートされます。

図3

0				
プロジェクトのインポート		·		
既存の Android フロジェクトを検索する	テイレクトリーを選択してくれ	201		
ルート・ディレクトリー: C:¥		¥He	elloAndroid	参照
プロジェクト:				
Project to Import			新規プロジェクト名	すべて選択
✓ C:¥		¥HelloAndroid	HelloAndroid	選択をすべて解除
☑ ブロジェクトをワークスペースにコピー □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	-			
	を追加(工)			
				2210(5)
				J畦17((ビ)
?	< 戻る(B)	次へ(N) >	完了(F)	キャンセル

●誤った操作手順

こちらの手順は、通常の(Android ではない)Java のプロジェクトをインポートするとき の操作となります。既に Java で Android 以外のアプリを作成したことがある人が陥りやす い手順です。ご注意下さい。なお、この方法でインポートしてしまった場合の対応策は、 こちらになります。

1. パッケージ・エクスプローラー上を右クリックし、「インポート」を選択します(ここ までは正規の手順と同じ)。

インポート画面で、「一般」→「既存プロジェクトをワークスペースへ」を選択し、「次
 ヘ」ボタンをクリックします(図 4)。

义	4
---	---

● インポート				
選択 アーカイブ・ファイルまたはディレクト	リーから新規プロジェクトを	作成します。		Ľ
インポート・ソースの選択(S):				
 ○ かん ○ アーカイブ・ファイル ○ ファイル・システム ○ 設存プロジェクトをワークス □ 設定 ○ Android ○ C/C++ ○ Git ○ XML ○ インストール ○ テーム ○ 雪気/デパッグ 	<u>~7</u> ^]			E
?	< 戻る(B)	次へ(N) >	完了(F)	キャンセル

 3. プロジェクトのインポート画面で、「ルート・ディレクトリー」の「参照」 ボタンで インポート対象のプロジェクトファイルを選択し、「プロジェクトをワークスペースにコピ ー」のチェックボックスにチェックし、「完了」ボタンをクリックします(図 5)。

図 5			
0			
プロジェクトのインポート 既存の Android ブロジェクトを検索するディレクトリ・	ーを選択してください		0
ルート・ディレクトリー: C:¥	¥He	lloAndroid	参照
プロジェクト:			
Project to Import		新規プロジェクト名	すべて選択
C:¥	i¥HelloAndroid	HelloAndroid	選択をすべて解除
			970974
図 ブロジェクトをワークスペースにコピー リーキング・セット			
□ ワーキング・セットにプロジェクトを追加(T)			
ワーキング・セット(0):			選択(E)
?	表る(B) 次へ(N) >	完了(F)	キャンセル

この手順でインポートをしてしまうと、src フォルダと gen フォルダの設定が外れてしまい、

エラーが発生してしまいます。この手順でインポートした場合の回避するためには、次の 追加操作が必要になります。

[追加操作]

1. プロジェクトをインポートした後、プロジェクトの上を右クリックして、「Properties(プロパティ)」を選択します(図 6)。

図6

🚺 Java - ADT	CONTRACT OF A	Contraction (Contraction)		- 0 X
Eile Edit Refactor Source Navigate	Search Project Run y	<u>Window</u> <u>Help</u>		
11 - E & A & E = X =	☑ • 🔐 🎋 • 💽 • (Q₄▼(⊕♂▼(⊕⊗*))∃ × (□ × (□ × (□ × (□ × (□ × (□ × (□ × (Quick Access	📑 📑 Java
1 Package Explorer			BE Outline 🖾	
I Pactage Explorer S ● gill HelioAndroid ● gill gen exemple.ant ● gill res.evenule.ant ● gill res.evenule.ant	6 年 **** CCを右クリック	Problem: @ Javadac @ Declaration @ Censel 22 @ Intro Log 10 LogCat Andod Table 44-13 15:39:12 - HelloAndroid] Stereting activity example.android.helle.HelloAndroidActivity on device emulator-5554 [2014-04-13 IS:39:12 - HelloAndroid] Stereting activity example.android.helle.HelloAndroidActivity on device emulator-5554 [2014-04-13 IS:39:12 - HelloAndroid] Stereting activity example.android.helle.HelloAndroidActivity on device emulator-5554 [2014-04-13 IS:39:10 - HelloAndroid] Stereting activity example.android.helle.HelloAndroidActivity on device emulator-5554 [2014-04-13 IS:39:10 - HelloAndroid] Stereting activity example.android.helle.HelloAndroidActivity on device emulator-5554 [2014-04-13 IS:39:10 - HelloAndroid] Stereting activity example.android.helle.HelloAndroidActivity on device emulator-5554 [2014-04-13 IS:39:10 - HelloAndroid] Stereting activity example.android.helle.HelloAndroidActivity on device emulator-5554 [2014-04-13 IS:39:10 - HelloAndroid] Stereting activity example.android.helle.HelloAndroidActivity on device emulator-5554 [2014-04-13 IS:39:10 - HelloAndroid] Stereting activity example.android.helle.HelloAndroidActivity on device emulator-5554 [2014-04-13 IS:39:10 - HelloAndroid] Stereting activity example.android.helle.HelloAndroidActivity on device folder.convert to a source folder of trouble writing output: already prepared [2014-04-13 IS:39:10 - HelloAndroid] Stereting activity example.androidActivity on device folder.convert to a source folder of trouble writing output: already prepared [2014-04-13 IS:30:10 - HelloAndroid] Stereting activity example.androidActivity on device folder.convert to a source folder of trouble writing output: already prepared [2014-04-13 IS:30:10 - HelloAndroid] AndroidActivity on device androidActivity on device activity on device activ	Be Outline 13 An outline is not available.	• • • •
		<pre>[2014-013 33:0415 = Heiloadnoid] at Franking Annually. [2014-013 15:0415 = Heiloadnoid] Atomatic Target Mode using existing mainter "emalter-5554' running compatible AO 'Ar [2014-0413 15:0415 = Heiloadnoid] Installing Heiloadnoid.apk.ru. [2014-0413 15:0415 = Heiloadnoid] Installing activity example android.hello.Heiloadnoid.ctivity on device emulator-5554 [2014-0413 15:0415 = Heiloadnoid] Starting activity example android.hello.Heiloadnoid.ctivity on device emulator-5554 [2014-0413 15:0415 = Heiloadnoid] Starting activity example android.hello.Heiloadnoid.ctivity on device emulator-5554 [2014-0413 15:0415 = Heiloadnoid] Starting activity example.android.hello.Heiloadnoid.ctivity on device emulator-5554 [2014-0413 15:0415 = Heiloadnoid] Atomatic Target Mode: using emulator 'emulator-5554' running compatible AO 'Ar [2014-0413 15:0415 = Heiloadnoid] Atomatic Target Mode: using existing emulator 'emulator-5554' running compatible AO 'Ar [2014-0413 15:0415 = Heiloadnoid] Atomatic Target Mode: using existing emulator 'emulator-5554' running compatible AO 'Ar [2014-0413 15:0415 = Heiloadnoid] Atomatic Target Mode: using existing emulator 'emulator-5554' running compatible AO 'Ar [2014-0413 15:0415 = Heiloadnoid] Atomatic Target Mode: using existing emulator 'emulator-5554' running compatible AO 'Ar [2014-0413 15:0415 = Heiloadnoid] Atomatic Target Mode: using existing emulator 'emulator-5554' running compatible AO 'Ar [2014-0413 15:0415 = Heiloadnoid] Application already deployed. No med to reinstall [2014-0413 15:045 = Heiloadnoid] Application already deployed. No med to reinstall [2014-0413 15:045 = Heiloadnoid] Application already deployed. No med to reinstall [2014-0413 15:045 = Heiloadnoid] Application already deployed. No med to reinstall [2014-0413 15:045 = Heiloadnoid] Atomatic Target Mode: using existing ot tartext, is courset to a source folder. [2014-0413 15:045 = Heiloadnoid] Atomatic Target Mode: using existing ot tartext, is courset to a source folder. [2014-0413 15:045 = Heiloadnoid] At</pre>	droidTest' tegory.LAUNCHER] cmp-exam droidTest' tegory.LAUNCHER] cmp-exam front ir rename it.	ple.android.he ple.android.he
		143M of 199M		

2. プロパティの設定画面で、「Java Build Path(Java ビルドパス)」を選択し、「Source(ソ ース)」タブを選択し、「Add Folder(フォルダ追加)ボタンをクリックします(図 7)。

図7

ype filter text	Java Build Path	 ⇔ • ⇔ • •
 Resource Android Android Lint Preferences 	Source of Projects Libraries Source folders on build path:	
Builders	a 😂 HelloAndroid	Add Folder
Java Build Path	المالية المالية المالية المالية	Link Source
 Java Compiler Java Editor 	Rative library location: (None) Figure optional compile problems: No	Edit
Javadoc Location Project References		Remove
Run/Debug Settings		
Validation		
	Allow output folders for source folders	
	Default output folder:	
	HelloAndroid/bin	Browse
4 III		
(?)		OK Cancel

3. ソースフォルダ選択画面で、「gen」と「src」のチェックボックスにチェックをつけて、「OK」ボタンをクリックします(図8)。

図 8		
O Source Folder Selection		
Select the source folder:		
🔺 🔳 😂 HelloAndroid		
> 📄 🗁 .settings		
🔲 🗁 assets		
bin 🕞 🕞 bin		
s s s s s s s s s s s s s s s s s s s		
Create New Folder		
	ОК	Cancel

4. 確認画面が表示されるため「Yes」をクリックします(図9)。

図9



以上の手順を追加することにより、本書のサンプルソースは問題なく動作します。

なお、「Chapter 6 インテント」のサンプルソースの「ImplicitIntentSample2」については、地 図を動作するためのターゲット API が「Google APIs」ではなくなっているので、プロパテ ィ画面で「Android」を選択し、「Project Build Target」を「Google APIs」に変更する必要が あります(図10)。

図10

🚯 Properties for ImplicitIntentSample2				
type filter text	Android	↓ ↓ ↓ ↓		
 Resource Android 	Project Build Target			
Android Lint Preferences Builders	Target Name Vendor	Platf A 4.2.2 17		
Java Build Path	Google APIs Google Inc.	4.2.2 17		
 Java Coue Style Java Compiler Java Editor Javadoc Location 	 Android 4.4.2 Android Open Source Pr Google APIs Google Inc. Google API Google Inc. 	4.4.2 19 4.4.2 19 4.4.2 19 4.4.2 19		
Project References Run/Debug Settings Task Tags ▷ Validation	Library			
	Reference Project	Add Remove Up Down		
4	Restore Defaults	Apply		
OK Cancel				

[目次] に戻る。

Macintosh における JDK の取り扱いの変更について (※本情報については、第1刷の読者の方のみご覧下さい。)

44 頁の「Note: Mac と Linux での開発環境の構築」に

なお Mac の場合、Mac OS X v10.6 Snow Leopard 以降のものであれば、JDK があらかじめイ ンストールされています。「Finder」から「アプリケーション」→「ユーティリティ」→「Java」 →「Java Preferences.app」で「Java Preferences」というソフトウェアを起動し、(後略)

という表記がありますが、2012 年 10 月 Apple より、Java Preferences アプリケーションが OS より削除されることが発表されました。

http://support.apple.com/kb/HT5493?viewlocale=ja JP&locale=ja JP

[システム環境設定]→[java]と操作することにより、自らの Mac OS に Java が入っているか どうかを確認した上で、必要に応じてインストールすることになります。 特に Java 7 以降は 下記 Oracle のサイトから Java をインストールする必要があります。

http://www.oracle.com/technetwork/java/javase/downloads/index.html

[目次] に戻る。

「Google APIs」が見あたらない。

[トラブル事例]

Chapter2 の SDK コンポーネントの追加について。SDK Manager.exe から「Android SDK Manager」のコンポーネントー覧が表示されたが本書で記載されている「Google APIs」 が SDK コンポーネントー覧に表示されない。

Preferences の Android を確認しても「Google APIs」はなく、表示されてい

るのは「Android4.2.2」のみです。

このような場合「Google APIs」を追加するにはどうすれば良いか?

(関連箇所:「2-3 SDK コンポーネントの追加)

[回答]

Android 4.2.2 以外選択できないのは、マシン側の問題ではなく、ネットワーク環境に問題がある可能性があります。

セキュリティチェックが厳しい企業のイントラネット等のネットワーク環境でセットア ップしている場合、プロキシサーバを経由している場合などは、インストールできない ことがあります。

プロキシ経由の場合は、SDK Manager に設定を追加する必要があります。下記のサイト にその方法が詳しく掲載されておりますので、ご参照下さい。

(ブログ「いいね!Android アプリ」より)

http://papakingyo-android.blogspot.jp/2011/11/android-sdkavd.html

[目次] に戻る。