Диалоговое окно Android с «иконифицированным» меню

Такой диалог присутствует в стандартном Андроиде, например, долгий тап на рабочем столе открывает диалог выбора добавляемого контента (виджет, обоины и т.д.). Итак, добро пожаловать под кат…  
  
  
Погуглив некоторое время и почитав официальные доки по платформе я так и не нашел, как реализовать подобный диалог. После некоторого времени копания я нашел результат, который быстрее всего будет очевидным для опытных Андроид-разработчиков.  
Ответ оказался прост и лежал он на поверхности. Суть его состоит в том, что для диалогового билдера нужно просто подставить соответствующий провайдер данных (есстественно, провайдер нужно самим написать).  
Итак, сейчас будет некоторое количество кода с комментариями о том, что происходит.

Список (провайдер) возможных типов аккаунтов

1. public final class AccountTypesProvider {
2. public static List<AccountType> accountTypes = Collections.unmodifiableList(Arrays.asList(
3. new AccountType(AccountType.TWITTER\_ACCOUNT, "Twitter", R.drawable.twitter\_icon\_big),
4. new AccountType(AccountType.FACEBOOK\_ACCOUNT, "Facebook", R.drawable.facebook\_icon\_big),
5. new AccountType(AccountType.BUZZ\_ACCOUNT, "Google Buzz", R.drawable.buzz\_icon\_big),
6. new AccountType(AccountType.LINKEDIN\_ACCOUNT, "LinkedIn", R.drawable.linkedin\_icon\_big),
7. new AccountType(AccountType.VKONTAKTE\_ACOUNT, "ВКонтакте", R.drawable.vkontakte\_icon\_big)
8. ));
9. }

\* This source code was highlighted with Source Code Highlighter.

Это простая обертка вокруг списка возможных типов аккаунтов. Тип аккаунта здесь простой POJO класс, который состоит из идентификатора (константа), названия сервиса и идентификатора ресурса иконки для данного типа сервиса.

ListAdapter для отображения списка типов аккаунтов в заданном layout'е

1. public final class AccountsTypesListAdapter extends ArrayAdapter<AccountType> {
2. private Activity context;
3. private List<AccountType> accountTypes;
5. public AccountsTypesListAdapter(Activity context, List<AccountType> accountTypes) {
6. super(context, R.layout.select\_account\_item, accountTypes);
8. this.context = context;
9. this.accountTypes = accountTypes;
10. }
12. @Override
13. public View getView(int position, View convertView, ViewGroup parent) {
14. LayoutInflater inflater = context.getLayoutInflater();
15. View row = inflater.inflate(R.layout.select\_account\_item, parent, false);
17. TextView label = (TextView) row.findViewById(R.id.text\_item);
18. label.setText(accountTypes.get(position).title);
20. ImageView icon = (ImageView) row.findViewById(R.id.icon\_item);
21. icon.setImageResource(accountTypes.get(position).bigIconId);
23. return row;
24. }
25. }

\* This source code was highlighted with Source Code Highlighter.

Для начала нужно передать в конструктор список типов, с которым будет работать адаптер списка. Привязка конкретного item'а к лэйауту происходит в переопределенном методе getView(). В нем загружается лэйаут из указанного ресурса, извлекаются виджеты и в них записываются данные об конкретном элементе списка. Кстати, индекс этого элемента автоматически доступен через параметр position.

Данный лист-адаптер работает с таким вот лэйаутом

1. <?xml version="1.0" encoding="utf-8"?>
2. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3. android:orientation="horizontal" android:layout\_width="fill\_parent"
4. android:layout\_height="fill\_parent" android:padding="10px">
6. <ImageView android:id="@+id/icon\_item" android:layout\_width="wrap\_content"
7. android:layout\_height="fill\_parent"/>
8. <TextView android:id="@+id/text\_item" android:layout\_width="wrap\_content"
9. android:layout\_height="fill\_parent" android:paddingLeft="10px"
10. android:paddingTop="5px" android:textStyle="bold"
11. android:textColor="#000000"/>
12. </LinearLayout>

\* This source code was highlighted with Source Code Highlighter.

Все, что теперь осталось — это привязать разработанный лист-адаптер к конкретному диалоговому окну

1. public static void showSelectAccountTypeDialog(Activity context, String title, OnClickListener dialogListener) {
2. AlertDialog.Builder builder = new AlertDialog.Builder(context);
3. builder.setTitle(title);
4. builder.setAdapter(new AccountsTypesListAdapter(context, AccountTypesProvider.accountTypes), dialogListener);
5. builder.create().show();
6. }

\* This source code was highlighted with Source Code Highlighter.

и вызвать в нужном месте activity диалог

1. private void displaySelectAccountTypeDialog() {
2. ApplicationDialogs.showSelectAccountTypeDialog(this, "Select network", new OnClickListener() {
3. @Override
4. public void onClick(DialogInterface dialogInterface, int selectedItemId) {
5. setupAccount(selectedItemId);
6. }
7. });
8. }

\* This source code was highlighted with Source Code Highlighter.

Диалог после своего завершения вернет индекс выбранного item'а параметром selectedItemId в листенер, который указан при вызове диалога. В данном простом случае этот индекс будет совпадать с ID типа аккаунта (item'а в списке), поэтому никаких дополнительных преобразований или извлечений не требуется. Для моей задачи этого индекса более чем достаточно.  
  
В итоге у меня получился такой вот симпатичный диалог  
  
  
Может я в данном топике просто описал прописные истины, может нет. Решать, как говорится, Вам, Хабраюзеры…  
  
Спасибо и удачи всем в увлекательном процессе разработки под Android-платформу ;)