


বাংলায় অ্যান্ড্রয়েড সহায়িকা



National
Mobile Application
Awareness Development &
Capacity Building Program

 **GDG** Sonargaon

বাংলায় অ্যান্ড্রয়েড সহায়িকা

অ্যান্ড্রয়েডকে বাংলা ভাষাভাষি মানুষের কাছে আনোঁর সহজভাবে উপস্থাপন করার জন্য আমাঁদের “বাংলায় অ্যান্ড্রয়েড সহায়িকা” পেট্রর োলটি সবাঁর জন্য উন্মুক্কত করাঁ হল।

জাতীয় মোবাইল অ্যাঁপ পর্শিক্ষণ কর্মর্সূচির মাধ্যমে ইতিমধ্যে বাংলাদেশের ৬৪টি জেলায় অ্যান্ড্রয়েড অ্যাঁপ পর্শিক্ষণ দেয়া হয়েছে। এরই ধাঁরাবাঁহকতায় এই বাঁই/পেট্রর োলটি পর্কাশ করাঁ হল। গুগল নিয়মিন্ তর্ত অ্যান্ড্রয়েড এর অফিসিয়াল ডেভলপাঁর সাইটে ট্রকন্টেন্ট এর বাংলা সঙ্করণ হল এই “বাংলায় অ্যান্ড্রয়েড সহায়িকা”।

এই পেট্রর োলটি এখনও অসম্পূর্ন এবং এতে পর্তিনিয়ত আপডেটের কাজ চলেবে। আপনাঁর যে কোন মন্ তবয্/সংশোধন/অবদান এর জন্য আমাঁদের জানান। আমাঁদের টিম বিসয়টি পযর্ োলাচনা করে উপযুক্কত বয্ বস্ থা গর্হণ করবে।



মন তবয / স ংশ াধন/ অবদ ান

আপনার যে কোন মন তবয / স ংশ াধন/ অবদ ান এর জন্য আমাদে র জান ান।
আমাদে র টিম বিষয়টি পয র ্ ালে চ াচনা ক রে উপযুক্ত তবয বস ্ থা গ র ্ হন
ক রে ব।

বা ংল ায় অয ্ ান ্ ডর ্ য়ে ড প র ্ ি শ ক ্ ষ ণ প র া ম শ র ্ ফ ম র ্

* Required

আপনার নাম *

আপনার ইমে ল *

আপনার ফোন

ঐ চ ্ ছ ক

প র া ম শ র ্ র ধ র ণ

ঐ চ ্ ছ ক

☐ ব া ন া ন / এ ডি ট ং ভূ ল

☐ কে া ড / ছ বি ভূ ল

☐ ন তু ন স ংযে া জ ন

☐ ন তু ন বি শ য়ে র

প র ্ ি শ ক ্ ষ ণ টি সমব ্ নে ্ ধ আপনার মত া মত / ইনপু ট

আপনি কে া ন অধয ্ ায় পু রে া / অধি ক া ং শ প রি ব ত র ্ ন ক রে ত চ া ই লে
তা এ থ া নে দি ন

ফাই লের ল ং ক (ঐ চ ্ ছ ক)

আপনি কে া ন অধয ্ ায় পু রে া / অধি ক া ং শ প রি ব ত র ্ ন ক রে ত চ া ই লে
তা গু ল ড ক আ ক া রে আপ লে া ড ক রে সে ই ল ং ক টি এ থ া নে দি ন

Submit

Never submit passwords through Google Forms.

Powered by

This content is neither created nor endorsed by Go

অয্যাপিলে কশন তৈরীর কাজ শুরুর
করুন

(<http://developer.android.com/training/index.html>)

অন্য হাম্‌ডৰ্‌য়েড ডেভেলপাৰ দেৱ পৰীক্ষক যোগে আপনাৰ দেৱ সৰ্বাগতম। এই ক্লাসে এমন সৰ্বশিক্ষা পাবেম যেনে খালে আলোচনা কৰা হবে কীভাবে কোড সম্‌হাম্‌পলি দিয়ে একটি টসুমিনিদিদৰ্‌ষ্‌টকা জাপিম আপনাৰ অন্য হাপ এবয়ব্‌হাৰ কৰতে পাৰবেম। ক্লাসগ্ৰন্থে কয়েকটি গৱ্‌হুপে শেৰ্‌গব্‌দৰ্‌ধ কৰা হৈছে যি উপৰেৰিল্লেকৰ বাম পাশেৰে নিভিগে শৰ্মেৰ উপৰেৰি দিকে দেদৰ্‌তে পাৰবেম।

পর্যবেক্ষণের সুপ, অর্থের যা এই অর্থ যা যা আপনাকে অর্থ যা যা উন্নয়ন দেয় অর্থ যা
 ডেভেলপ এর মূল বিষয়গুলি লোকে শেখাবে। আপনি যদি দ্রুত অর্থ যা উন্নয়ন দেয়
 অর্থ যা ডেভেলপার হোয়া, তাহলে আপনার উচিত এই কল্যাণগুলি
 ধারাবাহিকভাবে সম্প্রদায় করা।

১. পর্থম অয্প তৈরী করা

অয্য ান ্ ডর্ য়েড সফটওয়্যার ডেভেলপ িক টইমস্ টল কর্ণার পর অয্য ান ্ ডর্ য়েড অয্য াপ ডেভেলপ এর মূল বিষয়গুলো লে াশে থ ার জনয্য এই ক্ ল াস িদে য়ে শুরুর কর্ণ।

1. এ য় ত য ি ম য় ত য অ য় ঙ ক তৈ ত য ঐ ক য় া
2. ব্র য় প া য় ব্র য় া প ি ত্র ঙে শ য় য় া য় ক য় া
3. স য় জ ঙ গ ঙ জ া য় ণ য় ট া য় ঙে স য় তৈ ত য ঐ ক য় া
4. ব্র য় ম য় া য় ম য় ক া য় ম য় ক য় া ম য় শ য় ঙ ক য় া

২. একশান বার সংযোজন

অযোধ্যাপকায়ন ককর মবাস তবায়ি মর সবচে চয়ে গরু তব প মর ডি জা ঠি ম
উপাদান ম চৈ চৈ এক শ ম বায় । যদি ও ১১ লে ভে ভে লর এ পি আই চ া ল ক রায় সময়
অযোধ্যা ম ডর য়ে ড ২.১ বা এ র উপর মর স া স ক রি ম র ডি জ ত া ঠি সে এক শ ম বায়
অযো তত্ত ক তর ক রায় জ ম য় আপ নি স া প া ট র ল া ঠি বের র া ব য় ব শ া য ক রি ত
প া ঠি ম ।

1. একশান বার সেট আপ করা
2. একশান বার স্কেয়াজম করা
3. একশান বার স্টাইল করা
4. একশান বার ওভারলেন করা

৩. ভিন ভিন ন ডিভাইসে ক সাথে পাটর করা

কীভাবে বিবকল্প উপায়ে অ্যাপটাইজেশন করা যায় তা বিবর্তন সংস্করণের
 ডিডাইনস একটি একক অচক বয় বহার করে বয় বহারকারীদের একটি নতুন
 অভিজ্ঞতা দেওয়া যায়।

১. ভিত্তম্‌ ন ভিত্তম্‌ ন ভাষাংকৈ সপাটৈর্‌ কর্‌
২. ভিত্তম্‌ ন ভিত্তম্‌ ন সিস্‌ কর্‌ ন সপাটৈর্‌ কর্‌
৩. ভিত্তম্‌ ন ভিত্তম্‌ ন সস্‌ কর্‌ ন সপাটৈর্‌ কর্‌

৪. একটি টি ভি টি লাইফসাইকেল বয় বস্‌থাপনা করা

কীভাবে অর্থানুভবিত একটি ভিত্তিকাজ শুরু করে এবং শেষ করে এবং লাইফসাইকেল কলব্যবাপক পদ্ধতি ব্যবহার করে ব্যবহারকারীদের একটি

নিম্নলিখিত সূচকগুলির অভিজ্ঞতা দেওয়া যায়।

১. একটি ভিডিও (কমন্স ক্যান্ডি) শূন্য করা
২. একটি ভিডিওতে (কমন্স ক্যান্ডি) বিবর্তিত এবং বিবর্তিতের পর পুনরায় শূন্য
৩. কমন্স ক্যান্ডি থামাটোনা (স্টপ) এবং পুনরায় (রিস্টার্ট) শূন্য করা
৪. কমন্স ক্যান্ডি পুনর্নিময়ন

৫. ফর্ম্যাগমেন্ট সহকারে একটি ডায়নামিক ইউজার ইন্টারফেস তৈরি করা

অ্যাপ এর জনস্বার্থ কীভাবে ইউজার ইন্টারফেস তৈরি করা হয় যা বড়িস্ক্রিনে বহুবিধ ইউজার ইন্টারফেস কায়রু কর্ম পরিবেশনের জনস্বার্থে থ্রুট নমনীয় এবং ছোট স্ক্রিনে কাজ করার মতো উপযোগী করা-ফোন এবং ট্যাবলেট এর জনস্বার্থ একক অ্যাপলিকেশন প্যাকেজ এর মূল উপাদানগুলো তৈরি করা।

১. একটি ফর্ম্যাগমেন্ট তৈরি করা
২. নমনীয় (ফ্লেক্সিবল) ইউজার ইন্টারফেস তৈরি
৩. অন্যান্য ফর্ম্যাগমেন্ট সাথে যোগাযোগ করা

৬. তথ্যে সন্ধান করা

কীভাবে তথ্য ডিভাইসে সন্ধান করতে হয়, সেটাহতে প্যারামিটারে এটা অস্থায়ী ফাইল, ডাউনলোড করা অ্যাপ তথ্য, বয়বহারকারীর মিডিয়া, কাঠামোবদ্ধ তথ্য অথবা অন্য কিছু।

১. কী-ভ্যালু সেট সন্ধান করা
২. ফাইলে সন্ধান করা
৩. SQL ডাটাবেজে তথ্যে সন্ধান করা

৭. অন্য এ্যাপের সাথে পারস্পরিক ক্রিয়া (ইন্টারেকশন) করা

উন্নতমানের কমন্স সম্পাদনের লক্ষ্যে ডিভাইসে বিদ্যমান অন্যান্য অন্যান্য অ্যাপ এর সর্বোচ্চ সুবিধা নিয়ে বয়বহারকারীকে একটা ভালে অভিজ্ঞতার বয়বস্থা করা।

১. বয়বহারকারীকে (ইউজার) অন্য অ্যাপ এ পাঠানো
২. একটি ভিডিও (কমন্স ক্যান্ডি) থেকে ফলাফল পরীক্ষা
৩. একটি ভিডিও শূন্য করতে অন্য অ্যাপকে অনুমোদন

আপনার প্রথম অ্যাপ তৈরী করুন

(<http://developer.android.com/training/basics/firstapp/index.html>)

অ্যান্ড্রয়েড অ্যাপ লিকেশন ডেভেলপমেন্ট এ
আপনাকে স্বাগতম!

এই ক্লাসে আপনি শিখবেন কীভাবে আপনার প্রথম অ্যান্ড্রয়েড অ্যাপ
তৈরী করবেন। আপনি শিখতে পারবেন কীভাবে অ্যান্ড্রয়েড প্রজেক্ট
করা যায় এবং অ্যাপ এর ডিবাগ এবং স্ক্রন পরিচালনা করা যায়। আপনি
আরও শিখতে পারেন অ্যান্ড্রয়েড অ্যাপ ডিজাইন করার কিছু মৌলিক
ধারণা যা যাঁর মধ্যে রয়েছে কীভাবে একটি সরল ইউজার ইন্টারফেস তৈরী করা
যায় এবং ইউজার ইন্পুট পরিচালনা করা যায়। এই ক্লাস শুরুর করার পূর্বে
আপনি নিম্নলিখিত হয়ে নিম্নলিখিত ডেভেলপার পূরণ পরিবেশ তৈরী করা আছে,
যেখানে দরকার:

১. অ্যান্ড্রয়েড এসডিকে (SDK) ডাউনলোড করা

২. ইক্লিপস এর জন্য অ্যান্ড্রয়েড ডেভেলপার টুল (ADT) প্লাগিন
ইনস্টল করা (আপনি যদি ইক্লিপস IDE ব্যবহার করেন থাকেন)

৩. এসডিকে ম্যানুয়াল জার ব্যবহার করে সর্বশেষ এসডিকে টুলস এবং
প্লাটফর্ম ডাউনলোড করুন

আপনি যদি ইতিমধ্যে এই কাজগুলো শেষ করেন না থাকেন, তাহলে
অ্যান্ড্রয়েড এসডিকে (Android SDK) (এই <http://developer.android.com/sdk/index.html> লিংক
থেকে) ডাউনলোড করুন, নিম্নলিখিত ইনস্টল এর ধাপগুলো অনুসরণ
করুন। আর আপনি যদি এই কাজ শেষ করেন না থাকেন তাহলে আপনি এই কাজ শুরুর
করার জন্য পরস্তুত। এই ক্লাসে একটা টিউটোরিয়াল ব্যবহার করা হবে
যাতে ধীরে ধীরে একটা অ্যান্ড্রয়েড অ্যাপ তৈরীর করে দেখানো হবে
যেখানে অ্যান্ড্রয়েড অ্যাপ ডেভেলপমেন্ট এর মৌলিক ধারণাগুলি
সম্পর্কে জানা যাবে, সুতরাং পরিত্যাগ অনুসরণ করা
গুরুত্বপূর্ণ।

প্রথম পাঠ শুরুর করা

অ্যান্ড্রয়েড পর্জেক্ট তৈরী করা

(<http://developer.android.com/training/basics/firstapp/creating-project.html>)

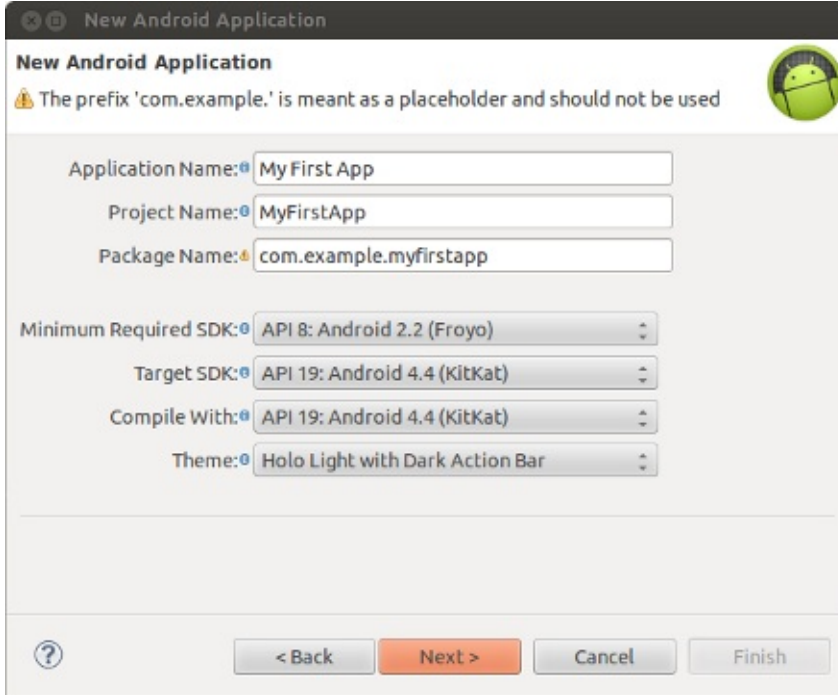
একটি অ্যান্ড্রয়েড পর্জেক্ট সকল ফাইল ধারণ করে, যা আপনার অ্যান্ড্রয়েড অ্যাপ এর জন্য সোর্স কোড গঠন করে। ডিফল্ট পর্জেক্ট ডিরেক্টরির এবং ফাইলের একটি সেটের সাথে একটি নতুন অ্যান্ড্রয়েড পর্জেক্ট শুরু করাটা অ্যান্ড্রয়েড এসডিকে টুলস সহজ করে দিয়েছে। এই অনুশীলনী আপনাকে দেখাবে একটি কমান্ড লাইন থেকে ইক্লিপস (এডিটিং প্লাগিন সহ) বা এসডিকে টুলস ব্যবহার করে নতুন পর্জেক্ট শুরু করা যায়।

নোট: আপনার অ্যান্ড্রয়েড এসডিকে ইনস্টলড করে ফেলা উচিত, এবং আপনি যদি ইক্লিপস ব্যবহার করে থাকেন তাহলে আপনার ADT Plugin (21.0.0 সংস্করণ বা এর পরবর্তী সংস্করণ) ইনস্টল করে ফেলা উচিত। আপনি যদি না করে থাকেন তাহলে এই অনুশীলনী শুরু করার পূর্বে Installing the Android SDK এই নিবন্ধে শিক্ষা অনুসরণ করা উচিত (লিংক: <http://developer.android.com/sdk/installing/index.html>)।

ইক্লিপস দিয়ে পর্জেক্ট তৈরী করা

১. টুলবারের New  ক্লিক করুন

২. উইন্ডোতে যা দেখা যাবে, সেখানে **Android** ফোল্ডার ওপেন করুন, **Android Application Project** নিবর্চন করুন, **Next** এ ক্লিক করুন



ফিগার ১. ইক্লিপস এ

নতুন অ্যান্ড্রয়েড অ্যাপ পর্জেক্ট উইজাডর্

৩. যে ফর্মটি আসবে তা পূরণ করুন:

- **Application Name** হচ্ছে অ্যাপ এর নাম যেটা হারকারী দেয় নিকট দৃশ্যমান হবে. এই পর্জেক্টের জন্য এটার নাম দিন
- **Package Name** হচ্ছে আপনার অ্যাপের প্যাকেজ নাম (জাভা পেরাগরামি অ্যেলগুয়েজ প্যাকেজগুলোর মতো একই নিয়মে মেনে চলতে হবে)! আপনার প্যাকেজের নাম অবশ্যই সম্ব্পূর্ণ অ্যান্ড্রয়েড সিস্টেমে যে সকল প্যাকেজ ইনস্টলড হয়েছে তার থেকে সর্বতম হবে! এই কারণে সবচেয়ে ভালো হয় আপনি যদি আপনার পর্তিষ্ঠানের নাম বা আপনার পর্কাশগী (পাবলিকেশন) পরিচয়ের ডোমেইন নাম আছে তা উল্লেখ করে ব্যবহার করেন! এই পর্জেক্টের জন্য আপনি "com.example.myfirstapp" এই ধরনের কিছু একটা ব্যবহার করতে পারেন! কিন্ত এই "com.example" নামে সম্ব্পসি দিয়ে আপনার অ্যাপ গুলে পর্কাশ করতে পারবেন না!
- **Minimum Required SDK** হচ্ছে অ্যান্ড্রয়েডের সর্বনিম্ন সংস্করণ যা আপনার অ্যাপ সাপোর্ট করবে, এপিআই লেভেল বহুবার দিই দর্শ করে। যত বেশী ডিভাইসকে সাপোর্ট করা সম্ভব তা করতে এটাতে সর্বনিম্ন ভার্সন বা সংস্করণ সেট করা উচিত যা আপনার অ্যাপ অনুমোদন করে এর মূল (কোর) বৈশিষ্ট্যগুলো দিতে পারে। যদি আপনার অ্যাপের কোন ফিচার শুধুমাত্র অ্যান্ড্রয়েডের নতুন সংস্করণে সম্ভব হয় এবং এটা অ্যাপের কোর ফিচার সেটের জন্য ক্রিটিক্যাল না হয়, আপনি তখনই ফিচারটি চালু করতে পারবেন, যখন এই ভার্সনে এটা চালানো হবে যা এটাকে সাপোর্ট করবে। (যেভাবে ভিন্ন ভিন্ন প্ল্যাটফর্ম সংস্করণকে সাপোর্ট করা অধ্যায়ে আলোচনা করা হয়েছে)। এই পর্জেক্টের জন্য এই সেটিংস ফল্ট ভ্যালুতে ছেড়ে দিন।

- Target SDK অ্যান্ড্রয়েডের সর্বোচ্চ ভাসরন কে নিবন্ধিত করে (API Level ও ব্যবহার করে থাকে) যা দিয়ে আপনি আপনার অ্যাপ লিকশন পরীক্ষা করিয়ে নিতে পারেন।

যখন এন্ড্রয়েডের নতুন কেকান ভাসরন (সংস্করণ) আসবে, আপনার উচিত হবে নতুন ভাসরনে আপনার অ্যাপ পরীক্ষা করে নেয়া এবং নতুন প্ল্যাটফর্ম বৈশিষ্ট্যের সুবিধা নিয়ে এই সর্বশেষ এপিআই লেভেলের সাথে ম্যাচ করতে এই ভ্যালু আপডেট করুন।

- Compile With হচ্ছে প্ল্যাটফর্ম ভাসরন যার বিপরীতে আপনি আপনার অ্যাপ কমপাইল করতে পারবেন। সর্বশেষ অ্যান্ড্রয়েড ভাসরনে এটাবাইডিফল্‌ট সেট করা আছে যা আপনার এসডি কেতে পাবেন (এটা অ্যান্ড্রয়েড ৪.১ অথবা এর চেয়ে বেশী হওয়া উচিত; আপনি যদি এই ভাসরন না থাকে তাহলে অবশ্যই SDK Manager ব্যবহার করে এটাইনস্টল করতে হবে) লিংক: <http://developer.android.com/sdk/installing/adding-packages.html>। আপনি এখনও আপনার অ্যাপ পুরাতন ভাসরনের জন্য তৈরি করতে পারেন, কিন্ত এই সেটিং এর লক্ষ্য হওয়া উচিত যাতে এটানতুন ভাসানের ও কাজ করে এবং সর্বশেষ ভাসরনের ডিভাইসেও এই অ্যাপ ব্যবহারকারীদের জন্য অন্তর্ভুক্ত করে দেয়া উচিত।
- Theme নিবন্ধিত করে আপনার অ্যাপ কাজ করানোর জন্য অ্যান্ড্রয়েড ইউজার ইন্টারফেস স্টাইল। আপনি এটাকে নিজেই মতো করে কাজ করতে দিতে পারেন।

Next বাটনে ক্লিক করুন

৪. পেরিস্করেন পরেজেক্ট এর আকৃতি দিতে, ডিফল্‌ট সিলেকশন কে সেভাবেই রেখে **Next** বাটনে ক্লিক করুন*

৫. পরবর্তী স্করন আপনার অ্যাপের জন্য লঞ্চার আইকন তৈরী করতে সাহায্য করবে।

আপনি বিভিন্ন উপায়ে একটি আইকন এর ধরন নিবন্ধিত করতে পারেন এবং টুলগুয়েলা স্করনের যনতব (ডেভিজিটি) অনুসারে একটি আইকন তৈরী করতে পারেন। আপনার অ্যাপ পাবলিশ করার আগে আপনার নিশ্চিত হওয়া উচিত যে Iconography ডিজাইন গাইড এনিমিডরস্‌টভাবে যে সংজ্ঞা দেয়া আছে তার সাথে আপনার আইকন মেলিকনা। (লিংক: <http://developer.android.com/design/style/iconography.html>)

Next বাটনে ক্লিক করুন

৬. আপনার অ্যাপ তৈরী করা শুরু করতে আপনি একটা এয়ারিক্‌টিভিটি টেমপ্লেট তৈরী করতে পারেন

এই পরেজেক্ট এর জন্য BlankActivity নিবন্ধিত করুন এবং Next বাটনে ক্লিক করুন এখানে যা যাতে আছে সেভাবেই রাখুন এবং Finish বাটনে ক্লিক করুন

কিছু ডিফল্‌ট ফাইলের সাথে আপনার অ্যান্ড্রয়েড পরেজেক্টের সেট আপ করা শেষ হয়েছে এখন আপনি আপনার অ্যাপ তৈরীর কাজ শুরু করতে পার্স্‌তুত. পরবর্তী অনুষীলনীতে চলে যান:

কমান্ড লাইন টুলস দিয়ে পর্জেক্ট তৈরী করুন

আপনি যদি ADT দিয়ে Eclipse IDE ব্যবহার না করেন থাকেন, তাহলে আপনি এর পরিবর্তে কমান্ড লাইন থেকে SDK tools ব্যবহার করে আপনার পর্জেক্ট তৈরী করতে পারেন:

১. অ্যান্ড্রয়েড SDK's tools এর অভ্যন্তরে রিডেক্টরিস পরিবর্তন করুন

২. যা করতে হবে:

Android list targets

* এটা অ্যান্ড্রয়েড প্ল্যাটফর্মের একটি তালিকা তৈরী করে দেবে যা আপনি আপনার সফটওয়্যারে ডেভেলপমেন্ট টুল (SDK) এর জন্য ডাউনলোড করেছিলেন। প্ল্যাটফর্মটি খুঁজে নিন যার বিপরীতে আপনি আপনার অ্যাপ কম্পাইল করতে চান। টাগের্ট আইডি এর একটা নোট তৈরী করুন। আমাদের পরামর্শ হচ্ছে আপনি যথাসম্ভব সবচেয়ে আপডেট ভার্সনকে নিবর্তাচিত করুন। আপনি এখনও পুরাতন ভার্সনকে সাপোর্ট করতে পারেন এমন অ্যাপ তৈরী করতে পারেন, কিন্ত এই সেটিং এর লক্ষ্য হওয়া উচিত যাতে এটা নতুন ভার্সানের ও কাজ করে এবং সবরূপে শেষ ভার্সানের ডিভাইসেও এই অ্যাপ ব্যবহারকারীকে দরজন্য অন্তর্ভুক্ত করে দেয়া উচিত।

আপনি যদি দেখেন যে কোন টাগের্ট লিস্টড না, তাহলে আপনাকে অ্যান্ড্রয়েড এসডিউকে ম্যানুয়ালি জার টুল ব্যবহার করে এটাইনস্টল করতে হবে। দেখুন Adding Platforms and Packages: (<http://developer.android.com/sdk/installing/adding-packages.html>)।

৩. যা করতে হবে:

android create project --target --name MyFirstApp \
--path /MyFirstApp --activity MainActivity \
--package com.example.myfirstapp

টাগের্ট তালিকা থেকে একটা আইডি এর সাথে এর পরিবর্তে পর্জিতস্থাপন করে নিন (পূর্ববর্তী ধাপ থেকে) এবং যে লোকশনে আপনার অ্যান্ড্রয়েড পর্জেক্ট সেভ করতে চান সেই সেই লোকশন এখানে পর্জিতস্থাপন করুন।

আপনার অ্যান্ড্রয়েড পর্জেক্ট এখন কিছু ডিফল্ট কনফিগারেশন দিয়ে সেট আপ হয়ে ছে এবং আপনি এখন আপনার অ্যাপ তৈরীর জন্য কাজ শুরু করতে পারেন। পরবর্তী অনুষঙ্গীলন তৈরী যান।

পরামর্শ *: আপনার PATH এনভায়রনমেন্ট টেভি রেইবল এ platform-tools/ এবং একই সাথে tools/ যোগ করুন

অ্যাপ রান করা

(<http://developer.android.com/training/basics/firstapp/running-app.html>)

যদি একটি অ্যান্ড্রয়েড পর্জেক্ট তৈরী করতে পূর্ববর্তরী অনুশীলনী অনুসরণ করে থাকেন, এরমধ্যে অ্যাপ তৈরী করে আছে "Hello World" সোর্স ফাইল এর একটি ডিফল্ট স্ট্যা অ্যাপ রান করতে তা ক্রমিক অনুমোদন দিয়ে থাকে।

আপনি কীভাবে আপনার অ্যাপ রান করতে বর্তমান দুটো বিষয়ের উপর নির্ভর করে:

আপনার সত্যিকার অ্যান্ড্রয়েড পাওয়ার ডিভাইস আছে অথবা আপনি ইমুলেটর ব্যবহার করে থাকেন। এই অনুশীলনী আপনাকে শেখাবে আপনার অ্যাপ কীভাবে সত্যিকার ডিভাইসে অথবা অ্যান্ড্রয়েড ইমুলেটরে ইনস্টল ও রান করতে হয়, এবং উভয়ক্ষেত্রেই ইমুলেটর অথবা কম্যান্ড লাইন টুল দিয়ে ইনস্টল ও রান করতে হয়।

আপনার অ্যাপ রান করার পূর্ববর্ত অ্যান্ড্রয়েড পর্জেক্টের কিছু ডিরেক্টরি এবং ফাইল সম্বন্ধে সতর্ক থাকে উচিত:

AndroidManifest.xml

Manifest file অ্যাপের মৌলিক বৈশিষ্ট্য টিনিয়ে আলাচনাকরে এবং এর প্রতিটা উপাদান নিয়েও আলাচনাকরে। এই ফাইলে আপনি নানা ধরনের আলাচনা শিখতে পারবেন যদি আপনি আরও টের ইনিংক লাসগুলা পড়েন।

আপনার মেনিফেস্টের উপাদানের অন্যতম উপাদান হিসাবে উপাদান অন্তর্ভুক্ত হওয়া উচিত। এটা বিভিন্ন অ্যান্ড্রয়েড ভার্সনে android:minSdkVersion এবং android:targetSdkVersion এটির বিউট ব্যবহার করে আপনার অ্যাপ এর কমপেটিবিলিটি কে যাচনা করে। আপনার পরথম অ্যাপের ক্ষেত্রে এটা এমন দেখানো উচিত:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android" ... >
<uses-sdk android:minSdkVersion="8" android:targetSdkVersion="17" />
...
</manifest>
```

আপনার সব সময়ই উচিত সেবর্চ পযর্য়ে android:targetSdkVersion সেট করা এবং অনুরূপ প্ল্যাটফর্ম এ আপনার অ্যাপ সেট করে নেওয়া। আরও তথ্যের জন্যে দেখুন (Supporting Different Platform Versions) ভিন্ন ভিন্ন প্ল্যাটফর্ম সংস্করণকে সাপোর্ট করা অধ্যায়টি দেখুন।

src/

আপনার অ্যাপের প্রধান সোর্স ফাইল এর জন্য ডিরেক্টরি। বাই ডিফল্ট এটা একটি Activity ক্লাসে অন্তর্ভুক্ত যতখন রান করে যখন অ্যাপ আইকন ব্যবহার করে এটা তার যাত্রা শুরু করে।

src/

app resources এর জন্য কিছু সাব-ডিরেক্টরি ধারণ করে। এখানে অল্প কিছু আছে:

drawable-hdpi/

ড্রয়েবল অবজেক্টের জন্য ডিরেক্টরী (যেমন বিটম্যাপ) যা উচ্চ

যনতব্ মান সম্ পনয্ িস্ কর্ নের (hdpi) জনয্ ডিজাইন করা হয়েছে। অন্য্ ডর্য়েবল ডিরেক্টরি সেই বৈশিষ্ট্য ধারণ করে যা অন্য্ ানয্ িস্ কর্ ন এর জনয্ পর্ যোজয্ ।

layout/

অয্ াপে ইউজার ইন্টারফেসের আলাচনাকরে এমন ফাইলের জনয্ ডিরেক্টরী

values/

অন্য্ ানয্ বহুবিধ XML ফাইলের জনয্ ডিরেক্টরী যারিসে সাসর্ এর সংগর্হকে ধারণ করে যেমন িস্টর্ ং এব ং কালার এর বনর্ না।

যখন আপনি ডিফল্ট অয্ ান্ ডর্য়েড অয্ াপ তৈরী এব ং রান করবেন, তখন ডিফল্ট Activity ক্লাস একটা লেআউট ফাইল শুরুর করবে এব ং লোড করা শুরুর করে যা বলবে "Hello World"। এর ফলাফল চমকপর্দা কিছ্ ইনয়কিন্ত্ এটা গুরুর তব্ পূর্ণর্ য়ে ডেভলপিং শুরুর পূর্বেই আপনি জানেন আপনার অয্ াপ কীভাবে রান করে।

একটা সত্যিকার ডিভাইসে সরান করুন

আপনার যদি এবটা সত্যিকার অ্যান্ডরয়েড প্যাপার ডিভাইস থাকে, তাহলে আপনি নিম্নে বর্ণিত পদ্ধতিতে আপনার অ্যাপ ইনস্টল এবং রান করতে পারবেন:

1. আপনার ডিভাইসটা ইউএসবি ক্যাবলের মাধ্যমে আপনার ডেভেলপমেন্ট মেশিনে প্লাগ ইন করুন, আপনার ডিভাইসের সাথে সামগ্রিক সম্পর্ক ইউএসবি ড্রাইভার ইনস্টল করা লাগতে পারে। ড্রাইভার ইনস্টল করতে সাহায্যের জন্য OEM USB Drivers (<http://developer.android.com/tools/extras/oem-usb.html>) তথ্যটি দেখুন।


2. আপনার ডিভাইসে USB debugging চালু করুন

* অ্যান্ডরয়েড ৩.২ অথবা এর চেয়ে পুরাতন এর ক্ষেত্রে আপনি অপশনটি খুঁজে পাবেন Settings > Applications > Development এখানে।

* অ্যান্ডরয়েড ৪.০ এবং এর চেয়ে উন্নত সংস্করণের ক্ষেত্রে Settings > Development options এ খুঁজে পাবেন

নোট: অ্যান্ডরয়েড ৪.২ এবং এর চেয়ে উন্নত সংস্করণের ক্ষেত্রে Development options বাই ডিফল্ট হিডেন থাকে। এটাকে পেতে হলে Setting > About phone > যান Build number সাতবার টেকা দিন। পূর্ববর্তের সিস্টেম ফিরে এসে Development options পাবেন।

ইক্লিপস থেকে অ্যাপ রান করতে:

1. আপনার একটি প্রজেক্ট ফাইল ওপেন করুন এবং টুলবার থেকে **Run**  ক্লিক করুন
2. **Run as** নামে যে উইন্ডো আসবে তার **Android Application** কনবর্তিত করে **OK** তে ক্লিক করুন

ইক্লিপস আপনার কানেক্টেড ডিভাইসে অ্যাপ ইনস্টল করে দিবে এবং চালু করে দিবে,

অথবা কমান্ড লাইন থেকে আপনার অ্যাপ রান করা:

1. আপনার অ্যান্ডরয়েড পর্জেক্টের দিকে ডিরেক্টরির পরিবর্তন করুন এবং করুন:

```
ant debug
```

2. নিশ্চিত করুন আপনার PATH এনভায়রনমেন্টে ডিরেক্টরি অ্যান্ডরয়েড এসিডিক platform-tools/ ডিরেক্টরির অন্তর্ভুক্ত আছে। তারপর এটা করুন:

```
adb install bin/MyfirstApp-debug.apk
```

3. আপনার ডিভাইসে **MyFirstActivity** খুঁজে বের করুন এবং এটা ওপেন করুন

এভাবে আপনি আপনার অ্যান্ডরয়েড অ্যাপ, একটি ডিভাইসে তৈরী ও রান করতে পারবেন। ডেভেলপিং শুরু করতে পরবর্তী অনুষঙ্গীলনীতে চলে যান।


ইমু য়্লেটের রান করুন

আপনি ইন্সটলপস বাকমান্ডলাইন যেটাই ব্যবহার করে থাকুন না কেন, ইমু য়্লেটের অ্যাপ রান করতে আপনার পরে থেমেই একটি Android Virtual Device (AVD) তৈরী করতে হবে। AVD হচ্ছে অ্যান্ডরয়েড ইমু য়্লেটের রিডভাইস কনফিগারেশন যা আপনাকে বিবিধ রিডভাইসকে মডেল করতে অনুমোদন করে।



ফিগার ১. AVD ম্যানেজার কিছুর ভাচুর য়্যাল রিডভাইস দেখাচ্ছে।

১. অ্যান্ডরয়েড ভাচুর য়্যাল রিডভাইস ম্যানেজার:

- ইন্সটলপস এটুলবার থেকে অ্যান্ডরয়েড ভাচুর য়্যাল রিডভাইস ম্যানেজার এ  ক্লিক করুন

* কমান্ডলাইন থেকে, ডিরেক্টরিতে `/tools/` তে পরিবর্তন করুন এবং এটা করুন:

```
android avd
```

f

২. অ্যান্ডরয়েড ভাচুর য়্যাল রিডভাইস ম্যানেজার প্যানেলে, New ক্লিক করুন।

৩. AVD এর জন্য বিস্তারিত বিবরণগুলো পূরণ করুন। এর একটি নাম, একটা প্ল্যাটফর্ম ট্যাগের্ট, একটি SD কার্ড সাইজ এবং একটি সিস্কন দিন (HVGA হচ্ছে ডিফল্ট),

৪. Create AVD তে ক্লিক করুন।

৫. অ্যান্ডরয়েড ভাচুর য়্যাল রিডভাইস ম্যানেজার থেকে নতুন AVD নিবরোচিত করুন এবং Start এ ক্লিক করুন

৬. ইমু য়্লেটের ব্রুস্ট আপ হওয়ার পর ইমু য়্লেটের সিস্করন আনলক করুন।

ইক্লিপস থেকে অ্যাপ রান করতে:

1. আপনার যে কোন একটি পর্জেক্ট ফাইল ওপেন করুন এবং টুলবার থেকে Run ক্লিক করুন

1. Run as নামে যে উইন্ডো আসবে তার Android Application কে নির্বাচিত করে OK তে ক্লিক করুন

ইক্লিপস আপনার AVD তে অ্যাপ ইনস্টল করে দিবে এবং চালু করে দিবে

অথবা কমান্ড লাইন থেকে আপনার অ্যাপ রান করুন

1. আপনার অ্যান্ড্রয়েড পর্জেক্ট এর দিকে ডিরেক্টরি পরিবর্তন করুন এবং নিচের কাজটি করুন:

```
ant debug
```

2. নিশ্চিত করুন আপনার PATH এনভায়রনমেন্টে ভেরিয়েবল

অ্যান্ড্রয়েড এসডিকে platform-tools/ ডিরেক্টরি অন্তর্ভুক্ত আছে। তারপর করুন:

```
adb install bin/MyFirstApp-debug.apk
```

3. ইমুয়েলারে **MyFirstActivity** খুঁজে বের করুন এবং এটা ওপেন করুন

এভাবে আপনি আপনার অ্যান্ড্রয়েড অ্যাপ ইমুয়েলারে তৈরী ও রান

করতে পারবেন। ডেভেলপিং শুরু করতে পরবর্তী অনুশীলনীতে চলে যান।

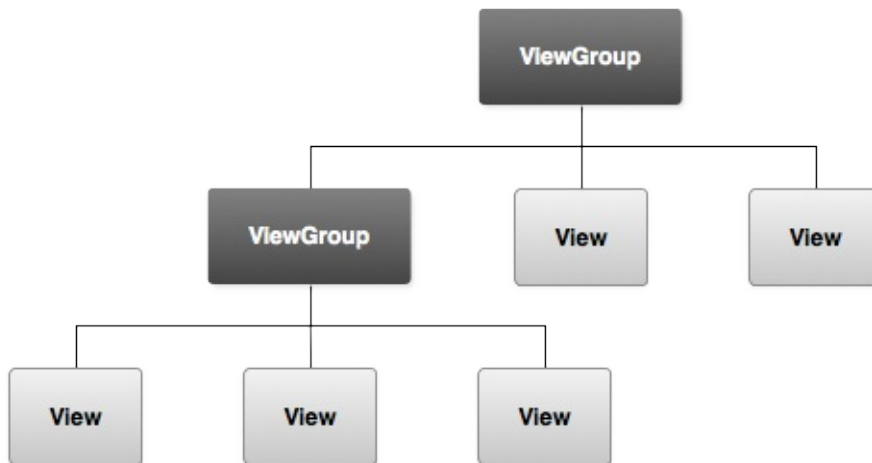
একটি সরল ইউজার ইন্টারফেস তৈরী করা

(<http://developer.android.com/training/basics/firstapp/building-ui.html>)

অ্যান্ড্রয়েড অ্যাপ এর গুরুত্বপূর্ণ অংশ হল ইউজার ইন্টারফেস, View এবং ViewGroup এর বিষয়বস্তু তৈরি করার কৌশল। View অবজেক্ট আসলে ইউজার ইন্টারফেস উপাদান যেমন Buttons বা text fields এবং ViewGroup হচ্ছে অদৃশ্য ডিভাইসের ইন্টারফেস ডিভাইসের ইন্টারফেসের পুরো পরিমাপ, যেমন কোন গির্ড বা কোন ডিভাইসের ইন্টারফেস।

অ্যান্ড্রয়েড একটি XML ডেভাইসের ইন্টারফেসের সরবরাহ করে থাকে যা View এবং ViewGroup এর সাবক্লাসগুলিকে কল করে, সুতরাং ইউজার ইন্টারফেসের ইন্টারফেসের কৌশল আপনাকে আপনার ইউজার ইন্টারফেসকে XML এ আলাদা করে তৈরি করে দেবে।

বিবর্তনশীল আউট: আপনার ইউজার ইন্টারফেস লেআউটের আনন্ডাইম কোড এর বদলে XML এ আলাদা করে জানা নাটকীয়ভাবে পরিবর্তন করা, কিন্তু এটা বিশেষ করে জরুরী এইজন্য যে আপনি ডিভাইসের সিস্টেমের স্ট্রাকচারের ডিভাইসের লেআউট তৈরী করতে পারবেন। উদাহরণ হিসাবে বলতে পারি আপনি দুইটা ভাস্করীর লেআউট তৈরী করতে পারেন এবং বিদ্যমান সিস্টেমকে জানা নাটকীয়ভাবে একটি ডিভাইসের সিস্টেমের জন্য এবং আরেকটি বড় সিস্টেমের জন্য। আরও জানতে ডিভাইসের ডিভাইসের সিস্টেমের স্ট্রাকচারের ক্লাসটি দেখুন।



ফিগার ১. লেআউটের মধ্যবর্তী থেকে ViewGroup অবজেক্টস এবং অন্য View অবজেক্ট ধারণ।

এই অনুশীলনে, আপনি XML ল্যাঙ্গুয়েজে একটি লেআউট তৈরী করতে পারবেন যা একটি টেক্সটফিল্ড এবং একটি বাটন অন্তর্ভুক্ত করে। নিম্নোক্ত অনুশীলনে, বাটন চেপে টেক্সটফিল্ড এর কন্টেন্ট যখন অন্য একটি ডিভাইসে পাঠানো হবে তখন আপনি একে সবার জন্য জানাবেন।

লিনিয়ার লেআউট তৈরী করুন

res/layout/ ডিরেক্টরী থেকে activity_main.xml ফাইল ওপেন করুন।

নোট: ইন্টেলক্লিপস এ যখন লেআউট ফাইল ওপেন করবেন, পরে যাই আপনি পর্ফরম্যান্স লেল আউট এডিটর দেখতে পারবেন। এই এডিটর দিয়ে আপনি WYSIWYG টুল ব্যবহার করে লেআউট তৈরী করতে পারবেন। এই অনুশীলনীতে আপনাকে সরাসরি XML এর সাথে কাজ করতে হবে, সুতরাং XML এডিটর ওপেন করতে সিস্করেনের একদম নীচে Activity_main.xml ট্যাগে ক্লিক করুন। যখন আপনি RelativeLayout রুটিভিউ এবং TextView চাইল্ডভিউ এর সাথে activity_main.xml ফাইল সহ এই পর্জেক্ট তৈরী করবেন, আপনাকে একটি বাল্ডিংক এয়ারকিটিভিটি টেমপ্লেট তৈরী করতে হবে। পরে যাই, িউপাদান ডিলিট করুন এবং একে পরিবর্তন করুন। তারপর android:orientation এটির বিউট এয়ারকিটিভিটি এবং "horizontal" এ এটা সেট করুন। ফলাফলটা এরকম দেখাবে:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal" >
</LinearLayout>
```

LinearLayout হচ্ছে ভিউগরুপ (ভিউগরুপের সাবক্লাস) যা চাইল্ডভিউ লেআউট ভিউগরুপের ক্লাস বা হারাইজন্টাল ওরিয়েন্টেশনে তৈরী করে, যেভাবে এটির বিউট দ্বারা সুনির্দিষ্ট করা হয়েছে। LinearLayout এর পর্তিটা চাইল্ডকর্মানুসারে সিস্করেনে দেখা যাবে যেখানে XML এর মধ্যে দৃশ্যমান হবে।

অন্য দুটি বিশেষ ট্যাগ, তাদের আকার সুনির্দিষ্ট করার জন্য android:layout width এবং android:layout height সকল ধরনের ভিউ এর জন্য দরকার।

কারণ LinearLayout হচ্ছে লেআউট এর মধ্যে রুটিভিউ, এটার সিস্করেনের যে আয়তন সম্পূর্ণভাবে পূর্ণ করানো উচিত যা অয়ারপাওয়া যাবে যা "সম্পূর্ণত্ব বহন" এ উচ্চতা এবং পর্স্থ এর সেটিং এর মাধ্যমে করা হয়েছে। এই ভ্যালু এটাই নিশ্চিত করে যে পয়ারেন্ট ভিউ এর উচ্চতা ও পর্স্থ এর সাথে ম্যাচ করার জন্য ভিউটির উচ্চতা এর দৈর্ঘ্য এবং পর্স্থের আয়তন বৃদ্ধি করানো।

লেআউট পর্পারটিস বিষয়ক আরও তথ্যের জন্য Layout গাইড দেখুন (<http://developer.android.com/guide/topics/ui/declaring-layout.html>)।

টেক্সট ফিল্ড এ্যাড (সংযোজন) করুন

ইউজার কতর্ক এডিট টেক্সট ফিল্ড তৈরী করতে, এর মধ্যে একটি উপাদান যোগ করুন।

পরিচিত `View` অবজেক্ট এর মতোই, `EditText` অবজেক্ট এর পরিপাটিসকে সুনির্দিষ্ট করতে নির্দিষ্ট `XML` এন্ট্রি বিউটিনিং ধারণা করা উচিত। কীভাবে উপাদানের ভিতরে এটাকে আপনার নিশ্চিত করা উচিত তা এখানে আছে:

```
<EditText android:id="@+id/edit_message"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:hint="@string/edit_message" />
```

রিসোর্স অবজেক্টস সম্পর্কে কিছু কথা

সহজ ভাষায় রিসোর্স অবজেক্ট হচ্ছে একটি সর্বজনীন ইন্টজার (`integer`) নাম যা একটি অ্যাপ রিসোর্স এর সেট যুক্ত, যেমন বিটম্যাপ, লেআউট ফাইল, বা সিস্টেম। পরিচিত `রিসোর্স` আপনার পরিবেশের `gen/r.java` ফাইলের মধ্যে অনুরূপ রিসোর্সকে নির্ধারণ করে দেয়। আপনার রিসোর্স এ উল্লেখ করতে জক্লে অবজেক্ট নাম ব্যবহার করতে পারেন, যেমন আপনার যখন `android:hint` এন্ট্রি বিউটিনিং এর জন্য সিস্টেম ভাষা লু সুনির্দিষ্ট করে পরিয়ে যান হবে। আপনি একটি নিয়মবহিঃ তিরে সোর্স আইডি তৈরী করতে `android.id` এন্ট্রি বিউটিনিং করে একটি ভিউ এর সাথে সংযুক্ত করতে পারেন যা আপনাকে অন্য কোড থেকে ওই ভিউকে রেফার করতে অনুমোদন করে।

আপনি যখন আপনার অ্যাপ কমপাইল করেন তখন পরিবর্তিত `AndroidManifest.xml` এর `R.java` তৈরী করে। আপনাকে কখনই নিজ হাতে এই ফাইলকে পরিবর্তন করতে হবে না।

আরও তথ্যের জন্য `Providing Resources` গাইড পড়ুন (লিংক: <http://developer.android.com/guide/topics/resources/providing-resources.html>)

এই বিবেশ গুন সমূহ সম্পর্কে:

`android.id`

এটা ভিউ এর জন্য একটি ইউনিক আইডি সিস্টেম ফায়ার সর্ববরাহ করে, যা আপনি আপনার অ্যাপ কোড থেকে অবজেক্টকে রেফারেন্স সিহসাবে ব্যবহার করতে পারেন, যেমন অবজেক্ট পড়ে বা সংশোধন করতে (আপনি এটা পরিবর্তিত অনুশীলনীতে পারেন)

যখন আপনি `XML` থেকে যে কোন রিসোর্স অবজেক্ট উল্লেখ করবেন তখন (`@`) চিহ্নটি দরকার হবে। এটা রিসোর্স আইডি (`id`), সংলগ্ন শব্দ এবং তারপর রিসোর্স নাম (`edit_message`) কে অনুসরণ করে।

(+) চিহ্নটি রিসোর্স আইডি এর পূর্বে থাকে যা পরিবেশে শূন্য মাত্র যখন আপনি পরিবর্তিত একটি রিসোর্স আইডি নির্ধারণ করেন। আপনি যখন অ্যাপ কমপাইল করেন, `AndroidManifest.xml` এর `R.java` ফাইল এর মধ্যে নতুন রিসোর্স আইডি তৈরী করতে আইডি নাম ব্যবহার করে যা `EditText` উপাদানকে রেফার করে। একবার যদি এভাবে রিসোর্স আইডি কে ঘোষণা করা হয়, তাহলে আইডির অন্য রেফারেন্স এর জন্য প্লাস সাইন ব্যবহার করতে হবে না। প্লাস সাইন শূন্য মাত্র তখনই দরকার শূন্য মাত্র যখন একটি নতুন রিসোর্স আইডি কে নির্দিষ্ট করে করা হয় এবং ক্রমিক রিসোর্স যেমন সিস্টেম বা লেআউট এর জন্য এর

দরকার নেই।

android:layout width এবং android:layout height

উচ্চতা ও প্রস্থের জন্য সুনির্দিষ্ট সাইজ ব্যবহার করার পরিবর্তে, "wrap_content" মান সুনির্দিষ্ট করে ডিউএর কন্টেন্টের সাথে খাপ খেতে যতটুকু বড় হওয়া প্রয়োজন ডিউএর ঠিক ততটুকু বড় হওয়া উচিত। আপনি যদি এর পরিবর্তে "match_parent", ব্যবহার করে থাকেন, তখন EditText উপাদান স্ক্রিনের নীচ পূরণ করবে, কারণ এটি পয়ন্টের নীচ LinearLayout এর সাইজের সাথে ম্যাচ করতে পারে। আরও জানতে দেখুন Layouts গাইড (<http://developer.android.com/guide/topics/ui/declaring-layout.html>)।

android:hint

প্রদর্শন করার জন্য এটি একটি ডিফল্ট স্ট্রিং যখন টেক্সট ফিল্ড খালি থাকে। ভ্যালু হিসাবে হাডর্কে ডিউএর ব্যবহারের পরিবর্তে, "@string/edit_message" ভ্যালু একটি আলাদা ফাইলে নির্ধারিত স্ট্রিং রিসোর্সের রেফার করে। কারণ এটি একটি কনস্ট্রাক্টর রিসোর্সের রেফার করে (শুধুমাত্র আইডি নং ট্রায়াস নয়), এর জন্য কোন প্লাস দরকার হবে না। যাইহোক, যেহেতু আপনি এখন পর্যন্ত স্ট্রিং রিসোর্সের নির্ধারণ করেননি, আপনি পরে থেমেই একটা কম্পাইল এরর দেখতে পারবেন।

নোট: এই স্ট্রিং রিসোর্স এর এলিমেন্ট আইডি এর মেতা আইএনাম: edit_message। যাইহোক রিসোর্সের রেফারেন্স সবসময়ই রিসোর্স টাইপ দ্বারা সুবিধা প্রাপ্ত হয়(যেমন id অথবা string), সুতরাং একই নামে কোন সংঘাত তৈরী করেন না।

সিস্টেমের রিসোর্স এ্যাড(সংযোজন)করুন

যখন আপনার ইউজার ইন্টারফেসে স্টেক্সট এ্যাড করার দরকার হবে তখন সবসময় পরিত্যাগিত সিস্টেমের রিসোর্স এর মতো করে নিধর্যারন করা উচিত। সিস্টেমের রিসোর্স একটি সিস্টেম গুলে লোকেশনে সকল ইউজার ইন্টারফেসকে পরিচালনা করতে অনুমোদন করে। সিস্টেমের এক্সটানশ্যনাল আইজিএ আপনার অ্যাপকে পরিত্যাগিত সিস্টেমের রিসোর্স এর জন্য বিকল্প সংজ্ঞাপর্দান করে বিভিন্ন লয়ালিটি গুয়েজে লোকালাইজ করতেও অনুমোদন করে।

বাইন্ডিং ফল্ট, আপনার অ্যান্ডরয়েড পর্জেক্ট res/values/strings.xml এ একটি সিস্টেমের রিসোর্স ফাইল অন্তর্ভুক্ত করে। "edit_message" নামে একটি নতুন সিস্টেমের এ্যাড করুন এবং "Enter a message" এ ভয়ালুটি সেট করুন ("hello_world" সিস্টেমের টিআপিনি ডিলিট করে দিতে পারেন)।

আপনি যখন এই ফাইলে অবস্থান করবেন, বাটনের জন্য একটি "Send" সিস্টেমের এ্যাড করতে পারেন যা আপনার কথুবিশয়রই এ্যাড করতে হবে, যার নাম "button_send"। strings.xml এর জন্য ফলাফলটি এমন হবে:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="app_name">My First App</string>
  <string name="edit_message">Enter a message</string>
  <string name="button_send">Send</string>
  <string name="action_settings">Settings</string>
  <string name="title_activity_main">MainActivity</string>
</resources>
```

সিস্টেমের রিসোর্স বয়বহার করে আপনার অ্যাপ অন্যান্য লয়ালিটি গুয়েজে লোকালাইজ করা সম্পর্কিত আরও তথ্য জানতে, বিভিন্ন ডিভাইস সাপোর্ট করা ক্লাসিটি দেখুন।

ব াটন এ য ্ াড (স ংযে য াজন) কর ুন

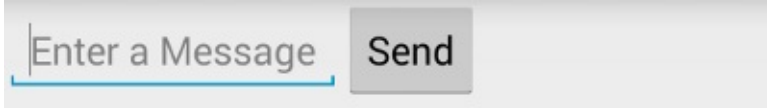
এ িল ে মন ্ ট ে ক ত া ক ্ষ ণ ি ক ভ া ে ব অন ু স র ন ক ে র এ খ ন ে ল আউ ে ট <Button> এ য ্ াড কর ুন

```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/button_send" />
```

উ চ ্ চ ত া এ ব ্ প র ্ স ্ থ "wrap_content" ত ে স ট ক র া হ য় য া ত ে ব াটন ি ট ব াটন ে ট ক ্ স ট এ র স া থে খ া প থে তে য ত ত ট ু ক ু ব ড় হ ও য় া প র ্ য া জন ি ঠ ক ত ত ট ু ক ু ব ড় হ য়। এ ই ব াটন ে র android:id এ ি ট র ্ বি ব উ ে ট র দ র ক া র ন া ই, ক া র ন এ ক ি ট ি ভ ি ট ে ক া ড থে ক ে র ফ া ে র ন ্ স ড হ ে ব ন া।

সিস্করন পরস্ থ ভরাট করার মেতা কের ইনপুট বক্স তৈরী করুন

লেআউট ইদানিং কালে ডিজাইন করা হয়েছে যাতে EditText এবং Button উভয় উইজিট তার উপাদানগুলো সাথে খাপ খাওয়ায় মেতা যথেষ্ট বড় হয়. ফিগার ২ তে যেভাবে দেখানো হয়েছে।



ফিগার ২. EditText এবং Button উইজিট এর "wrap_content" এ তাদের পরস্ থ সেট আছে।

এটা বাটনের জন্য ভালা কাজ করে কিন্ত টেক্সট ফিল্ড এর জন্য ততটা ভালা নয়, কারণ বয় বহারকারীরা বড় আকারের কিস্ক টাইপ করতে পারে। সুতরাং অবয় বহুত সিস্করন পরস্ থ কে টেক্সট ফিল্ড দিয়ে পূর্ণ করটা ভালা হবে। আপনি এটা করতে পারবেন এর LinearLayout মধ্যে **weight** পরপাটির দি়ে, যা android:layout_weight এটির বিউট বয় বহার করে এটাকে সুনিদিষ্ট করতে পারবেন।

ওয়েট ভয়ালু হচ্ছে একটা সংখ্যা যা পরতিটা ভিউ এর জন্য যে পরিমাণ সেপস আছে তার কতটুকু বয় করা উচিত তার পরিমাণ সুনিদিষ্ট করে দেয়, অন্যরূপভাবে বিবিলি ভিউ এর কেষ্টের কতটা সেপস বয় করা হবে সেটাও। এই কাজ এভাবে বেঝা যায় যে একটা পানীয়র মিশরনে কোনটাকি পরিমাণ আছে: দুই অংশ ফলের রস এবং এক অংশ পানি, এর অর্থ পানীয়র দুই-তৃতীয়াংশ ফলের রস আছে। উদাহরন হিসাবে, আপনি যদি একটা ভিউ এর ওয়েট ২ ধরেন এবং আরেকটির জন্য ১ ধরেন তহলে এর যোগফল হবে ৩, সুতরাং পরথম ভিউ বিদয়মান সেপসের দুই-তৃতীয়াংশ পূর্ণ করে আছে এবং দিবৃতীয়া ভিউ বাকী অংশ দখল করে। আপনি যদি তৃতীয়া ভিউ যোগ করেন এবং এটার ওয়েট ১ নিধরারন করেন তখন পরথম ভিউ (যার ওয়েট ২) বিদয়মান সেপসের অধরক দখল করবে, বাকী দুইটা ভিউ এক চতুর্থ অংশ করে স্থান দখল করবে।

সকল ভিউ এর ডিফল্ট ওয়েট হচ্ছে ০, সুতরাং আপনি যদি কোন ওয়েট ভয়ালু ০ এর চেয়ে বেশী করে নিদিষ্ট করে দেন, তখন ওই ভিউ তার যেটুকু দরকার তা দেয়া সতে তবুও যতটুকু সেপস বাকী আছে তার সবটুকুই পূরণ করে নেয়। সুতরাং EditText এ লিমেম্ টি দিয়ে আপনার লেআউটে ট্রিবিদয়মান সেপস পূরণ করার জন্য এটার ওয়েট ১ দিন এবং বাটনটি টেকে কোন ওয়েট না দি়েই ছেড়ে দিন।

```
<EditText
    android:layout_weight="1"
    ... />
```

ওয়েট সুনিদিষ্ট করার সময় আপনার লেআউটকে কায়র কর করতে, আপনার উচিত EditText এর পরস্ থর আকার শূন্যতে (Zero) পরিবর্তন করা (0dp)। পরস্ থর সেটিং জেরা (0dp) লেআউটে ট্রি পারফরমেন্স কে উন্নত করে করারন "wrap_content" কে পরস্ থ হিসাবে বয় বহার করতে সিস্ টমিটিকে পরস্ থর হিসাবিনকাশ করার দরকার হয়, যা শেষ পরন্তত অপরাসিঙ্ গক কারন বিদয়মান সেপস কে পূরন করতে ওয়েট ভয়ালু কে আরেকটি পরস্ থ হিসাব করতে হয়।

```
<EditText
    android:layout_weight="1"
    android:layout_width="0dp"
    ... />
```

ফিগারও দেখাচ্ছে যে যখন আপনি সকল ওয়েট EditText এ লেমনে টিনিয়াজিত করেন তার ফলাফলিক।

Send

ফিগারও. EditText উইজিট সকলে লআউট ওয়েট পর্দান করে, LinearLayout এর মধ্যে বাকী স্পস পূর্ণ করুন।

এখানে দেখা যাচ্ছে একটি সম্পূর্ণ লআউট ফাইল দেখতে কেমন হবে:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">
<EditText android:id="@+id/edit_message"
    android:layout_weight="1"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:hint="@string/edit_message" />
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_send" />
</LinearLayout>
```

এই লআউট ডিফল্ট Activity ক্লাস দ্বারা পরে যাগ করা হয় যা এসিডকে টুলস তৈরী করেছিল, যখন আপনি এই পরে জেক্ট তৈরী করেছিলেন, সুতরাং আপনি এখন অ্যাপ রান করে এর ফলাফল দেখতে পারেন।

* ইন্টেলপসএ, টুলবার থেকে Run  এ ক্লিক করুন।

* অথবা কমান্ড লাইন থেকে আপনার অ্যান্ডরয়েড পরে জেক্ট রুট ডিরেক্টরী পরিবর্তন করুন এবং এরপর করুন:

```
ant debug
adb install bin/MyFirstApp-debug.apk
```

পরবর্তী অনুশীলনীতে, বাটন পের্স করতে কীভাবে বেসপন্ড করবেন, টেক্সট ফিল্ড থেকে কীভাবে কনটেন্ট পড়বেন, অন্য কমর্কান্ড শুরুর করবেন সেই কাজগুলো এবং অন্যান্য কাজে শেখানো হবে।

অন্য একটি টিভি টি শুরুর করা

(<http://developer.android.com/training/basics/firstapp/starting-activity.html#RespondToButton>)

পূর্ববর্তরী অনুশীলনী শেষ করার পর আপনি যে অর্থাৎ পাবেন যা একটি টেক্সট ফিল্ড এবং একটি বটাম সহ একটি এ্যক্টিভিটি দেখাবে (একটি স্কেলিং লিস্ট করুন)। এই অনুশীলনীতে MainActivity তে ক্লিক ছুঁতে কেঁদে এঁদে করতে পারবেন যানতুন কাজ শুরুর করে যখন বয় বহারকারী সেন্ড বটাম এ ক্লিক করবে।

সেন্ড বাটনকে রেসপন্স করা

বাটনের অনিক্লক ইভেন্ট এর রেসপন্স করতে, activity_main.xml লেআউট ফাইল ওপেন করুন এবং <Button> এলিমেন্ট এ android:onClick এটিট্রিবিউট এ য়াড করুন:

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_send"
    android:onClick="sendMessage" />
```

android:onClick এটিট্রিবিউট এর ভ্যালু "sendMessage" হচ্ছে আপনার একটি ভিউ এর একটা পদ্ধতির নাম, ব্যবহারকারী যখন বাটন ক্লিক করে সিস্টেম তখন এটাকে কল করে।

MainActivity ক্লাস ওপেন করুন (পর্জেক্টের src/ ডিরেক্টরীতে আছে) এবং সমরূপ মেথড এ য়াড করুন।

```
/** Called when the user clicks the Send button */
public void sendMessage(View view) {
    // Do something in response to button
}
```

এর জন্য দরকার View ক্লাস নিয়ে আসা দরকার।

```
import android.view.View;
```

টিপ: ইন্টেলিজেট এ Ctrl + Shift + O পের্স কেরিমিস ক্লাস আনতে হয় (Mac এ Cmd + Shift + O)

android:onClick এ যে পদ্ধতির নাম দেওয়া হয়েছে তার সাথে সিস্টেম এর ম্যাচ করার জন্য ক্লিক হলে বিশেষ টাইপে মেথডে বৈধতা দেওয়া হয়েছে ঠিক সেভাবে হতে হবে। সুনির্দিষ্টভাবে পদ্ধতিটি অবশ্যই -

- Be public/ উন্মুক্ত হতে হবে
- Have a void return value ঐ ধাবধাডরফ বাং হাধষব/ ভয়েড রিটার্ন ভ্যালু থাকতে হবে
- একমাত্র প্যারামিটার হিসাবে একটা View থাকতে হবে (এটা হবে সেই ভিউ যেটাকে ক্লিক করা হবে)

পরবর্তীতে, আপনি টেক্সট ফিল্ড এর কন্টেন্ট পড়তে এবং ওই টেক্সট অন্য কোর্সে পর্দান করে এই পদ্ধতি পছন্দ করবেন।

ইন্টেন্ট তৈরী করা

একটি Intent হচ্ছে একটা অবজেক্ট যা দুটো পৃথক বিষয়ের (যেমন দুটো একটি টিভি) মধ্যে রানটাইম বাইন্ডিং পর্দান করে। Intent একটা অ্যাপের “কোন কিছুর করার পরিকল্পনা”কে চিত্রায়িত করে। আপনি সুবিধস্বত্ব বহু বিধকাজে ইন্টেন্ট ব্যবহার করতে পারেন তবে বেশীরভাগ সময় এটা অন্য কর্মরকান্ড শুরু করার কাজে ব্যবহার করা হয়ে থাকে।

DisplayMessageActivity নামের কর্মরকান্ড শুরু করতে sendMessage() পদ্ধতির মধ্যে একটি ইন্টেন্ট তৈরী করুন:

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

কনস্ট্রাক্টর এখানে দুইটা পর্যায়ক্রমিক বস্তু ব্যবহার করেছেন:

- একটা Context হচ্ছে এটার পরথম পর্যায়ক্রমিক (এটা ব্যবহার করা হয় কারন Activity ক্লাস হচ্ছে Context এর একটি সাবক্লাস)
- অ্যাপ উপাদানের Class যেখানে সিস্টেমের উচিত Intent ডিলিভারি দেওয়া (এক্ষেত্রে একটি টিভি টিউন শুরু হওয়া উচিত)

একটি ইন্টেন্ট অন্য অ্যাপে পাঠানো

এই অনুশীলনীতে যে ইন্টেন্ট তৈরী করা হয়েছে যা এক্সপ্লিসিট ইন্টেন্ট (**explicit intent**) হিসাবে বিবেচিত, কারন Intent টি যথার্থ অ্যাপকে ম্যাপার্টেনেন্টকে সুনির্দিষ্ট করে দেয় যেখানে ইন্টেন্ট দেয়া উচিত। যাহোক ইন্টেন্ট আবার ইমপ্লিসিট (**implicit**) ও হতে পারে, এখানে ক্যাঙ্কিত কর্ম্যে ম্যাপার্টেনেন্টকে সুনির্দিষ্ট করে দেয় না, কিন্ত ইন্টেন্টের সম্পর্ক করে ডিভাইসে যেকোন অ্যাপ ইনস্টল করতে অনুমোদন করে যতক্ষণ না এটা এই কাজের জন্য মেটা-ডেটা সেম্পসিফিকেশনকে সমস্ত করে যা বিভিন্ন ইন্টেন্ট এ সুনির্দিষ্ট। আরও জানতে অন্য অ্যাপের সাথে পারস্পরিক যোগাযোগ করা সম্ভবিকরত ক্লাস দেখুন।

নোট: যদি আপনি আইডি যেমন ইক্লিপস ব্যবহার করে থাকেন তাহলে isplayMessageActivity রেফারেন্স একটি এরর তৈরী করতে পারে, কারন ক্লাসটি এখন পর্যন্ত অসুতব্ধ শীল নয়। আপাতত এররটি উপেক্ষা করুন; আপনি শিঘরই ক্লাসটি তৈরী করতে পারবেন।

একটি ইন্টেন্ট শুধুমাত্র অন্য আরেকটি একটি টিভি শুরু করাকে অনুমোদনই করেনা, একটি টিভিতে একগুচ্ছ ডাটাতিনিয়ে আসে। sendMessage() পদ্ধতির ভিতরে EditText এ লিখেন টিপেতে findViewById() ব্যবহার করুন এবং ইন্টেন্ট এ এর টেক্সট ভয়ালু এয়াড করুন:

```
Intent intent = new Intent(this, DisplayMessageActivity.class);
EditText editText = (EditText) findViewById(R.id.edit_message);
String message = editText.getText().toString();
intent.putExtra(EXTRA_MESSAGE, message);
```

নোট: android.content.Intent এবং android.widget.EditText এর জেনে অ্যাপনার স্টেটমেন্ট আনার দরকার। আপনি একমুহুর্তে EXTRA_MESSAGE কনস্ট্যান্টকে নিখরহে আরন করতে পারবেন।

একটি Intent কি (Key) ভয়ালু পেয়ার হিসাবে বিভিন্ন ডাটাতাইপসিনিয়ে আসতে পারে যাকে **extras** বলে। putExtra() পদ্ধতি পরথম পর্যায়ক্রমিক (Key) নেম নেয় এবং দ্বিতীয় পর্যায়ক্রমিক (Key) ভয়ালু সিনিয়ে থাকে।

পরবর্তী এ্যাক্টিভিটি এর জন্য এক্সট্রা ডাটাতা অনুসন্ধান করতে

পাবলিক কনস্ট্যান্ট ব্যবহার করে আপনার ইন্টেন্ট এর এক্সট্রার
জন্যিক (Key) নিধারণ করে দেওয়া উচিত। সুতরাং MainActivity ক্লাসের
সবচেয়ে উপরে Extra_Message ডিফিনিশন এ্যাড করে দিন:

```
public class MainActivity extends Activity {  
    public final static String EXTRA_MESSAGE = "com.example.myfirstapp.MESSAGE";  
    ...  
}
```

পির্ফেক্ট হিসাবে আপনার অ্যাপ এর প্যাঁকেজ নাম ব্যবহার করে
ইন্টেন্ট এক্সট্রার জন্যিক (Keys) নিধারণ করা সাধারণত একটি ভুল
চর্চা। এটো নিশ্চিত করে যে অন্য অন্য অ্যাপের সাথে পারস্পরিক
কিরয়োর কেসে তরা সব তর্ক।

দ্বিতীয় কায়রকর্ম (একটি ভিডিও) শুরু করুন

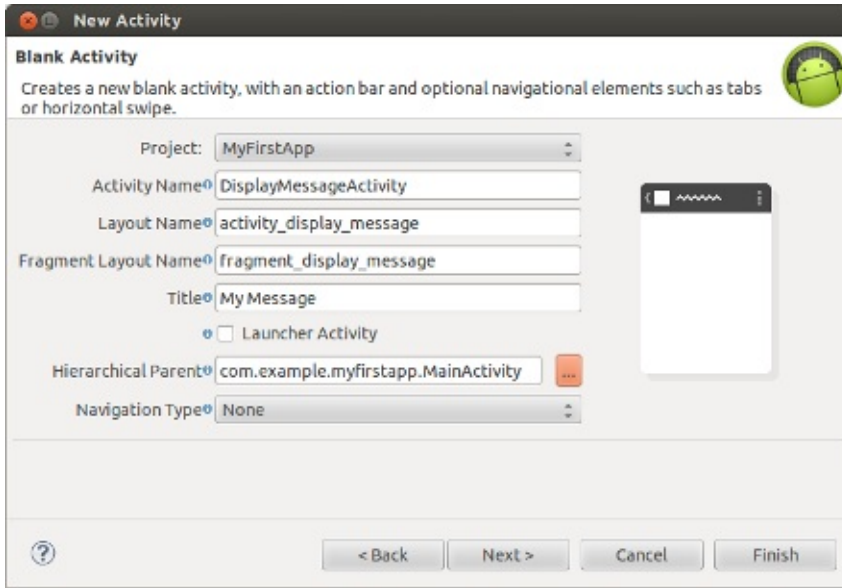
একটি ভিডিও শুরু করতে startActivity() কে আহবান করুন এবং আপনার Intent এ
পরিবেশ করান। সিস্টেম এই আহবান গ্রহণ করবে এবং Intent দ্বারা
নির্ধারিত Activity এর একটি ইনস্ট্যান্স শুরু করে।

সেন্ড বাটন ক্লিকে ডেকে আনানতুন কে ডের সাথে সম্বন্ধে sendMessage()
টি এখন দেখতে নিম্নরূপ:

```
/** Called when the user clicks the Send button */  
public void sendMessage(View view) {  
    Intent intent = new Intent(this, DisplayMessageActivity.class);  
    EditText editText = (EditText) findViewById(R.id.edit_message);  
    String message = editText.getText().toString();  
    intent.putExtra(EXTRA_MESSAGE, message);  
    startActivity(intent);  
}
```

এখন আপনার এই কাজ করতে DisplayMessageActivity ক্লাস তৈরি করা প্রয়োজন

দিব্ তীয় একটি টি ভি টি তৈরী করুন



ফিগার ১. ইক্লিপসে নতুন একটি টি ভি টি উইজাডর্।

ইক্লিপস ব্যবহার করে নতুন একটি টি ভি টি তৈরী করতে:

1. টুলবার থেকে **New**  তে ক্লিক করুন
2. উইন্ডোতে যাে দেখা যাবে তার থেকে **Android** ফোল্ডার ওপেন করুন এবং **Android Activity** নিবর্াচিত করুন। তারপর **Next** ক্লিক করুন।
3. **Blank Activity** নিবর্াচিত করুন এবং **Next** ক্লিক করুন।
4. একটি টি ভি টি রি বস্ তারিত পূরন করুন:
 - **Project:** MyFirstApp
 - **Activity Name:** DisplayMessageActivity
 - **Layout Name:** activity_display_message
 - **Title:** My Message
 - **Hierarchical Parent:** com.example.myfirstapp.MainActivity
 - **Navigation Type:** None

Finish ক্লিক করুন.

আপনি যদি ভিন্ ন আইডি বা কমান্ ড লাইন টুলস ব্যবহার করে থাকেন তাহলে পর্েজক্ ট এর src/ ডিরেক্ টরীতে DisplayMessageActivity.java নামে একটি নতুন ফাইল তৈরী করুন, যেটা MainActivity.java আসল ফাইলের কাছাকাছি

এই displayMessageActivity.java ফাইলটি ওপেন করুন। আপনি যদি এই একটি টি ভি টি তৈরী করতে ইক্লিপস ব্যবহার করে থাকেন:

* ক্লাস ইতিমধ্যে য্ ক ািঙ্ খত onCreate() পদ্ধতির একটি ট বাস্ ত বায়ন অন্তর্ভুক্ত করে ছে

* এখানে onCreateOptionsMenu() পদ্ধতির একটি ট বাস্ ত বায়নও অন্তর্ভুক্ত করে ছে। কিন্তু এটা এই অ্যাপের জন্য আপনার দরকার হবে না সুতরাং

আপনি এটামুে ছে ফলেত পোঁরেন

* এখন `onOptionsItemSelected()` পদ্ধতির একটি বাস্তুবায়নও অন্তর্ভুক্ত করে ছে যা একশনবারের আপ (টু) বিবেহিতয়ার এর জনয আচরন নিয়ন্ত্রণ করে। এটাকে ধের রাখার এটা একটা উপায়।

কারণ ActionBar APIs শুধুমাত্র HONEYCOMB (API level 11) এবং এর চেয়ে উন্নত সংস্করণে আছে, বর্তমান প্ল্যাটফর্ম ভার্সন চেক করার জন্য আপনাকে `getActionBar()` পদ্ধতির চারপাশে অবশ্যই একটি পরিস্থিতি এ্যাড করতে হবে। `lint` এর এড়াতে আপনাকে অবশ্যই `onCreate()` পদ্ধতিতে `@SuppressWarnings("NewApi")` ট্যাগ এ্যাড করতে হবে। `DisplayMessageActivity` ক্লাসটি এখন দেখতে এই রকম হওয়া উচিত:

```
public class DisplayMessageActivity extends Activity {

    @SuppressWarnings("NewApi")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_display_message);

        // Make sure we're running on Honeycomb or higher to use ActionBar APIs
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
            // Show the Up button in the action bar.
            getActionBar().setDisplayHomeAsUpEnabled(true);
        }
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case android.R.id.home:
                NavUtils.navigateUpFromSameTask(this);
                return true;
        }
        return super.onOptionsItemSelected(item);
    }
}
```

আপনি যদি ইন্টেলপস ছাড়া অন্য আইডিই ব্যবহার করে থাকেন তাহলে উপরের কোড দ্বারা আপনার `DisplayMessageActivity` ক্লাস আপডেট করেন।

সকল `Activity` এর সাবক্লাসের `onCreate()` পদ্ধতিটি অবশ্যই বাস্তুবায়ন করা উচিত। সিস্টেম এটাকে কল করে যখন একটি ভিউটি এর নতুন একটা উদাহরণ তৈরী করে। এই মেথডে আপনাকে অবশ্যই পদ্ধতি দিয়ে একটি ভিউটেলআউট নিধরারন করতে হবে এবং যেখানে আপনার একটি ভিউটি কমেপোনেন্ট এর জনয পরারিম্ভক সেটআপ দিতে হবে।

নোট: আপনি যদি ইন্টেলপস ছাড়া অন্য আইডিই ব্যবহার করে থাকেন তাহলে আপনার পরে জেক্ট `activity_display_message` তেলআউট থাকবে না যা `setContentView()` খুঁজে থাকে। এটা ঠিক আছে কারণ আপনি এই মেথড/পদ্ধতি পরবর্তীতে আপডেট করে বন এবং আপনি ওই তেলআউট ব্যবহার করেছেন না।

টাইটেল সিস্টার্ম্ এয াড করুন

আপনি যদি ইক্লিপস বয্ বহার করে থাকেন, পরবর্তী সেকশন এড়িয়ে চলে যেতে পারেন, কারণে টমপ্ লটিট নতুন একটি টিভি টর জনয্ টাইটেল সিস্টার্ম্ সর্ববরাহ করে থাকে।

আপনি যদি ইক্লিপস ছাড়া অন্য আইডি বয্ বহার করে থাকেন তাহলে strings.xml ফাইলে নতুন একটি টিভি টর টাইটেল এয াড করে দিন

```
<resources>
...
    <string name="title_activity_display_message">My Message</string>
</resources>
```


এটা কেমিনিফেস্ট এয্যাড করুন

একটি টিভিটি এলিমেন্ট বয়বহার করে অবশ্যই আপনার মিনিফেস্ট ফাইল `AndroidManifest.xml`, সকল ফাইল ডিকে লয়ার হওয়া উচিত।

একটি টিভিটি তৈরী করতে আপনি যখন ইক্সপস্টুলস বয়বহার করেন তখন এটা একটা ডিফল্ট এনিম্টর তৈরী করে। আপনি যদি ভিন্ন আইডি বয়বহার করে থাকেন, আপনার মিনিফেস্ট এনিম্টর এয্যাড করতে হবে। এটা দেখতে এরকম হবে:

```
<application ... >
...
<activity
    android:name="com.example.myfirstapp.DisplayMessageActivity"
    android:label="@string/title_activity_display_message"
    android:parentActivityName="com.example.myfirstapp.MainActivity" >
    <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value="com.example.myfirstapp.MainActivity" />
</activity>
</application>
```

`android:parentActivityName` এটি ট্রিবিউটিটি অ্যাপের লজিক্যাল হায়ারার্কি এর মধ্যে যেথেকে এই এক্টিভিটির প্যাঠোরেন্ট এক্টিভিটির নামে ঘোষণা করে। সিস্টেম ডিফল্ট এনিম্ভগেশন আচরন বাস্তুবায়ন করতে এই ভ্যালু বয়বহার করে থাকে, যেমন অ্যান্ডরয়েড ৪.১ (API level 16) বা এর পরবর্তী ভার্সনে `Up navigation`। আপনি একই এনিম্ভগেশন বিহীন ভিত্তিয়ার অ্যান্ডরয়েডের পুরোনা সঙ্করেনি দিতে পারেন, `Support Library` বয়বহার করে এবং এলিমেন্ট এয্যাড করে, যেভাবে এখানে দেখানো হয়েছে।

নোট: আপনার অ্যান্ডরয়েড এসডিকে ইতিমধ্যে সবরংশে অ্যান্ডরয়েড সাপোর্টার লাইব্রেরীতে অন্তর্ভুক্ত হওয়ার কথা। এটা এডিটিবান্ডল এর সাথে অন্তর্ভুক্ত কিকিন্তু আপনি যদি ভিন্ন আইডি বয়বহার করে থাকেন, তাহলে `adding Platforms and Packages` ধাপেই আপনাকে এটাইনস্টল করতে হতে। যখন ইক্সপস্টুলস এটেমপ্লেট বয়বহার করা হয় তখন সাপোর্টার লাইব্রেরী সয়ংকির্যভাবে আপনার অ্যাপ পরজেক্ট এয্যাড হয়ে যায় (আপনি ***android Dependencies*** তালিকায় লাইব্রেরীর JAR ফাইল দেখতে পারেন)। আপনি যদি ইক্সপস্টুলস বয়বহার না করেন, আপনাকে ম্যানুয়ালী আপনার পরজেক্ট লাইব্রেরী এয্যাড করতে হবে - `setting up the Support Library` (<http://developer.android.com/tools/support-library/setup.html>) গাইড অনুসরণ করুন তারপর আবার এখানে ফিরে আসুন।

আপনি যদি ইক্সপস্টুলস দিয়ে ডেভলপ করেন থাকেন তাহলে আপনি এখন অ্যাপ রান করতে পারেন, কিকিন্তু বেশী কিছু হবে না। সেন্ডবাটনে ক্লিক করে দ্বিতীয় এক্টিভিটি শুরু করুন কিকিন্তু এটাটেমপ্লেট কতরক পর্দা ডিফল্ট "Hello World" লেআউট বয়বহার করে। আপনি শিঘ্রই ডিসপ্লে কেস্টিমটেটক্ সটিভিউ এপিরবর্তন করে এক্টিভিটিকে আপডেট করবেন, সুতরাং আপনি যদি ভিন্ন আইডি বয়বহার করে থাকেন তাহলে চিহ্নিত হবে নতুন অ্যাপ এখনো কম্পাইল হয় নাই।

ইন্টেন্ট গ্রহণ করুন

পর্টিটিটা Activity কেই Intent আহবান করে থাকে, ব্যবহারকারী এটাকে কীভাবে পরিচালনা করে সেটাকে জানি বিষয় নয়। আপনি Intent টা পেতে পারেন যা getIntent() কে কল করে আপনার একটি ভিউটিকে শুরুর করতে পারেন এবং তথ্যকে পূরণরূদ্ধার করে এটার মধ্যে রেখে দেয়।

DisplayMessageActivity ক্লাসের onCreate() পদ্ধতির মধ্যে ইন্টেন্ট খুঁজেন এবং MainActivity কতরকপর্দত তেমেসজটা সিরিয়েনন।

```
Intent intent = getIntent();  
String message = intent.getStringExtra(MainActivity.EXTRA_MESSAGE);
```

যে যে সজ্জা দিয়ে পল্ল করুন

সিদ্ধ করলে যেমতে সজ্জা করতে একটা TextView উইডজিট (widget) তৈরী করুন এবং `setText()` ব্যবহার করে টেক্সটটি সেট করুন। `setContentView()` দিকে পাস করে দিয়ে একি টিভিভিউ লেআউট এর রুট ভিউ হিসাবে TextView এ্যাড করুন।

DisplayMessageActivity এর জন্য সম্পূর্ণ onCreate() পদ্ধতিটি দেখতে এরকম হবে:

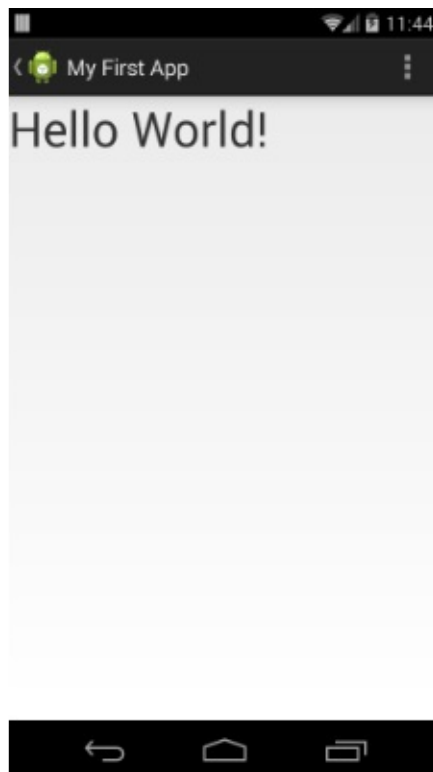
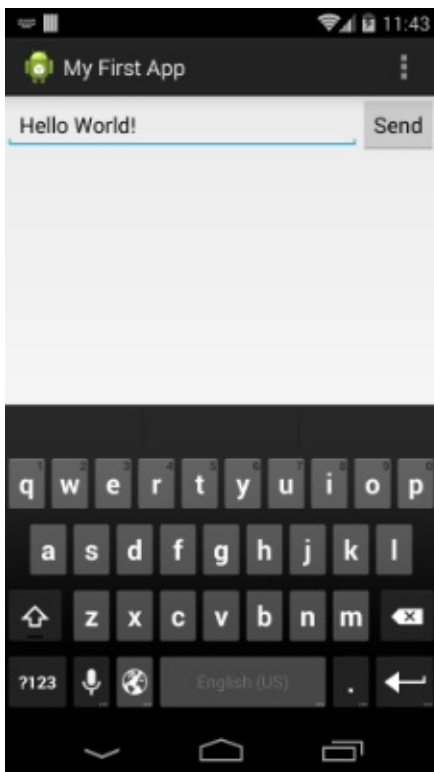
```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    // Get the message from the intent
    Intent intent = getIntent();
    String message = intent.getStringExtra(MainActivity.EXTRA_MESSAGE);

    // Create the text view
    TextView textView = new TextView(this);
    textView.setTextSize(40);
    textView.setText(message);

    // Set the text view as the activity layout
    setContentView(textView);
}
```

আপনি এখন অ্যাপ রাখতে পারবেন। যখন এটা ওপেন হবে, টেক্সট ফিল্ড
একটামেয়ে সজটাইপ করুন, সেমন্ড বাটনে ক্লিক করুন এবং মেয়ে সজটি
দিব্বতীয় একটি ভিডিওতে দৃশ্যমান হবে।



ক্ষিপণের ২. চুড়ানত অবস্থাপ্রমণে ধর্ম উভয় একটি টিভি টি অর্ধান্ডর ৪.০
 তেরান করে।

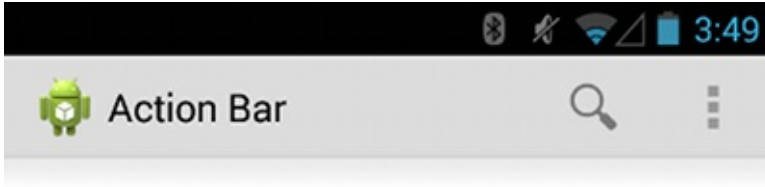
আপনি আপনার পর্থম যে অ্যান্ডারয়েড অ্যাপটি তৈরী করলেন এটাই সেট!

একশন বার এ য় াড (স ংযে াজন) কর া

(<http://developer.android.com/training/basics/actionbar/index.html>)

একশন বার হচেছ আপনার অ্যাপ এর একটি টিভি টি বাস্ তবায়নের জন্য
অন্যতম গুরুত্বপূর্ণ র্ ডিজাইন উপাদান। এটাে বশিকছু ইউজার
ইন্টারফেসে বৈশিষ্ট্য পর্দান করে যা অন্য ানয় অ্য ান্ডর়েড
অ্যাপস এর মধ্যে সমন্বয় করে আপনার অ্যাপকে বয় বহারকারীে দরকাছে
তা ক্ ষনিকভাবে জনপিপর্য় করে তুলবে। এর মূল কাজ সমূহ:

- আপনার অ্যাপকে একটা পরিচয় দেওয়ার জন্য একটি ডেডিকেটেড সেপস
এবং অ্যাপের মধ্যে ইউজার এর লোকেশন নিদর্শ করে।
- অনুমেয় উপায়ে গুরুত্বপূর্ণ কন্ট্রোল অনুষর্বে বশ করতে পারা
(যেমন সাচর্)
- নেভিগেশন এবং ভিউ পরিবর্তন করতে সহায়তা করে (ট্যাব বার্ডর্প-
ডাউন লিস্ট সহ)



এই পর্শিক্ষণ ক্ ল়ে স একশন বারের মেীলকি বিষয় দর্ুততার সাথে
শেখার গ াইড এর পরামর্ দেয়। একশন বারের বিভিন্ ন বৈশিষ্ট
সম্পর্কে র্ আরও জ ানতে দেদখুন Action Bar (<http://developer.android.com/guide/topics/ui/actionbar.html>)
গ াইড।

এই অধ্যায়ের অনুশীলনীয় সমূহ

একশনবারে সটআপ করা

শিখুন আপনার একটি টিভি টেবল কীভাবে বৈশিষ্ট্যবাহক একশনবার এয়ড করতে হয়, হতে পারে আপনার অয়প শুম্মাতর্ অয়ান্ডর্য়েড ৩.০ এবং এর চে উন্নত স্ক্রনকে সাপোর্ট করে অথবা অয়ান্ডর্য়েড ২.১ এর মতো নীচের স্ক্রনকেও সাপোর্ট করতে পারে (অয়ান্ডর্য়েড সাপোর্ট লাইব্রেরী ব্যবহার করে)

একশনবার এয়ড (সংযোজন) করা

শিখুন, কীভাবে একশনবারের মধ্যে ইউজার একশন এয়ড এবং রেসপন্স করতে হয়

একশনবার এর স্টাইল ঠিক করা

শিখুন কীভাবে একশনবার এর দৃশ্যমানতাকে কাস্টমাইজ করতে হয়

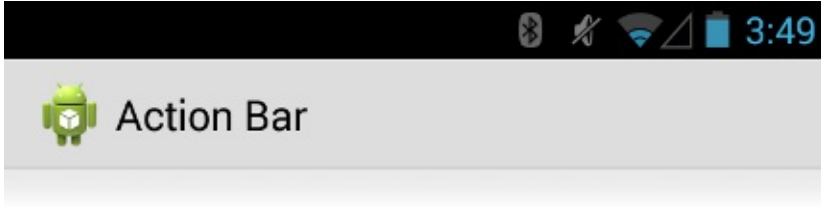
একশনবার ওভারলে করা

শিখুন কীভাবে আপনার লেআউটের সামনে একশনবারকে ওভারলে করতে তরুটি টহীন পরিবর্তনকে অনুমোদন করবে যখন একশনবার হাইড করা থাকে।

একশন বার সেট আপ করা

(<http://developer.android.com/training/basics/actionbar/setting-up.html>)

এর সবচেয়ে মৌলিক ধরন, একশন বার একটি টিভি টির জন্য টাইটেল এবং অ্যাপ আইকন বাম পাশে ডিসপ্লে করে। এমনকি এই সরল চেহারা, একশন বার সকল একটি টিভি টির জন্য উপকারী কেননা এটা ব্যবহারকারীকে জানায় যে তারা এখন কোন অবস্থায় আছে এবং নিয়মিত আপনার অ্যাপের পরিচয় বহন করার জন্য।



ফিগার ১. অ্যাপ আইকন এবং একটি টিভি টি টাইটেল সহকারে একটি একশন বার।

একটামৌলিক একশন বার সেটআপ করতে দরকার কারন, আপনার অ্যাপ একটা একটি টিভি টি থিম ব্যবহার করে যা একশন বারকে সক্ষম করে। কীভাবে এই ধরনের থিম চাওয়া হয় তা নিম্নের ক্রে আপনার অ্যাপ কতদূরক অ্যান্ডরয়েডের সবার্নিন্ডেকোন ভাসরনকে সাপোর্ট করে। সুতরাং এই অনুশীলনী দুইটা অধ্যায়ে ভাগ করা হয়েছে যে অ্যান্ডরয়েডের সবার্নিন্ডেকোন সংস্করণ পর্যন্ত সাপোর্ট করে।

শুধুমাত্র অ্যান্ড্রয়েড ৩.০ এবং এর চেয়ে উন্নত সংস্করণকে সাপোর্ট করা

অ্যান্ড্রয়েড ৩.০ (API level 11) দিয়ে শুরু করলে, একশনবার সকল একটি ভিউটির মধ্যে অন্তর্ভুক্ত থাকে যা Theme.Holo থিম (অথবা এর যেকোন একটি শাখা) ব্যবহার করে, যেটা একটি ডফল্ট থিম, যখন targetSdkVersion অথবা minSdkVersion এটি বিউটর যেকোন একটি "১১" অথবা এর চেয়ে বেশীতে সেট করা হয়।

সুতরাং আপনার একটি ভিউটিতে একশনবার এ্যাড করতে, সেজাসুজি ১১ বা এর চেয়ে বেশীতে এটির বিউট সেট করুন। উদাহরন:

```
<manifest ... >
  <uses-sdk android:minSdkVersion="11" ... />
  ...
</manifest>
```

নোট: আপনি যদি কাস্টম থিম সেট করে থাকেন, নিশ্চিত হোন যে এটা যেকোন একটি Theme.Holo থিম এর পর্যায়ের নট হিসাবে ব্যবহার করে।

এখন Theme.Holo থিম আপনার অ্যাপে ব্যবহৃত হচ্ছে এবং সকল একটি ভিউটি একশনবার দেখাচ্ছে।

অ্যান্ড্রয়েড ২.১ এবং এরচেয়ে উন্নত সংস্করণকে সাপোর্ট করে

যখন অ্যান্ড্রয়েড ৩.০ এরচেয়ে পুরাতন সংস্করণে (অ্যান্ড্রয়েড ২.১ পর্যন্ত) এটারান করে তখন একশনবার এ্যাড করতে আপনার অ্যাপলিকেশনে অ্যান্ড্রয়েড সাপোর্ট লাইব্রেরী অন্তর্ভুক্ত করা দরকার।

শুরু করার সময় Support Library Setup তথ্য পড়ুন এবং **v7 appcompat** লাইব্রেরী সেটআপ করুন (কখনও যদি আপনি লাইব্রেরী প্যাকেজ ডাউনলোড করে থাকেন, তাহলে Adding libraries with resources নিবন্ধে শিক্ষা অনুসরণ করুন)

কখনও আপনার অ্যাপ পরে জেক্টর সাথে যদি অন্তর্ভুক্ত সাপোর্ট লাইব্রেরী থেকে থাকে:

1. আপনার একটি ভিডিও আপডেট করুন যাতে এটি `ActionBarActivity` কে বিস্তৃত করে।
উদাহরণ:

```
public class MainActivity extends ActionBarActivity { ... }
```

2. আপনার মেনিফেস্ট ফাইলে যেকোন একটা `Theme.AppCompat` থিম ব্যবহার করার জন্য হয় উপাদান বা সবতর উপাদান আপডেট করুন। উদাহরণ:

```
<activity android:theme="@style/Theme.AppCompat.Light" ... >
```

নোট: আপনি যদি কাস্টম থিম তৈরি করে থাকেন, নিশ্চিত হোন যে এটি যে কোন একটা `Theme.AppCompat` থিম এর প্যারেন্ট হিসাবে ব্যবহার করে। একশনবার স্টাইল করা অধ্যায়টি দেখুন।

এখন আপনার একটি ভিডিও একশনবারকে অন্তর্ভুক্ত করে নিয়েছে যখন অ্যান্ড্রয়েড ২.১ (API level 7) এবং এর পরবর্তী সংস্করণে রান করে।

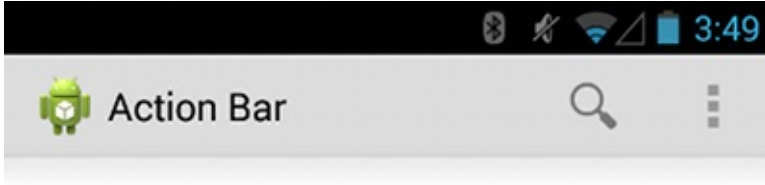
মেনিফেস্ট আপনাদের অ্যাপের এপিআই লেভেল সঠিকভাবে সেট করার কথা স্মরণ রাখবেন:

```
<manifest ... >
  <uses-sdk android:minSdkVersion="7" android:targetSdkVersion="18" />
  ...
</manifest>
```


একশন বারটন এয াড (স ংয াজন) করা

(<http://developer.android.com/training/basics/actionbar/adding-buttons.html>)

একশন বার আপন াকে অয াপে সর বতর্ ম ান পে র ঙ্ ক ষ াপে ট সব চে চ য়ে
গুরুর তব পূ নর্ একশন আইটেম এর জনয া বারটন এয াড করতে দেয়। ওইগু লে া
একশন বারের সর াসির আইকন বা/এব ং টেক্ সট এর সাথে দৃ শয ান হয় য াকে
একশন বারটন (**action buttons**) ব ল া হয়। যে একশন, একশন বার এর সাথে থ াপ থ ায় না বা
তেমন গুরুর তব পূ ণর্ নয় সেট া একশন ওভারফে ল াতে তি হে ডন থ াকে।



ফিগ ার ১, সাচর্ এব ং একশন ছি ড়ে য়ে দে য় ার জনয া একশন বারটন সহক ারে একশন
বার, য া অতি তিরক্ ত একশন উ নে ম াচন করে।

তগথ এ একশনকে সুনির্দিষ্ট করুন

একশন ওভারফ্লোর মধ্য সকল একশন বাটন এবং অন্য যেকোনও আইটেম একটি এক্সএমএল menu resource কতৃৎক নিধর্যারিত হয়ে থাকে। একশনবারে একশন এয়াদ করতে আপনার পর্ত্তেজেক্টর res/menu ডিরেক্টরীতে একটানতুন এক্সএমএল ফাইল তৈরী করুন।

আপনি একশনবারে যে আইটেমগুলো অন্তর্ভুক্ত করতে চান তার পর্ত্তিটার জন্য একটি এলিমেন্ট এয়াদ করুন। উদাহরন:

res/menu/main_activity_actions.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
<!-- Search, should appear as action button -->
<item android:id="@+id/action_search"
    android:icon="@drawable/ic_action_search"
    android:title="@string/action_search"
    android:showAsAction="ifRoom" />
<!-- Settings, should always be in the overflow -->
<item android:id="@+id/action_settings"
    android:title="@string/action_settings"
    android:showAsAction="never" />
</menu>
```

একশনবার আইকন ডাউনলোড করুন

অয়ান্ডরয়েড iconography গাইডলাইনকে ভালেভাভে ময়াদ করতে, আপনার Action Bar Icon Pack কতৃৎক পর্ত্তেজেক্টর আইকন ব্যবহার করা উচিত। (<http://developer.android.com/design/downloads/index.html#action-bar-icon-pack>)

এটা জানায় যে সাচর একশনকে একশন বাটন হিসাবে দৃশ্যমান হওয়া উচিত, যখন একশনবারে যেথেষ্ট জায়গা থাকে, কিন্তু সেটি একশন সবসময় ওভারফ্লোতে দৃশ্যমান হওয়া উচিত। (বাইডিফল্ট, সকল একশন ওভারফ্লোতে দৃশ্যমান হয়, কিন্তু পর্ত্তিটা একশনের জন্য আপনার ডিজাইনের উদ্দেশ্য স্পষ্টত জানানো হচ্ছে ছভালে চচর্যা)

একটি ইমেজের জন্য icon (আইকন) এটির বিউটি এরিরেসাসর্ আইডি দরকার। আপনার পর্ত্তেজেক্টর res/rawable/ ডিরেক্টরীতে @drawable/ নামে একটা বিটময়্যাপ ইমেজ সেভ করেছেন। উদাহরণ হিসাবে, "@drawable/ic_action_search" G+K ic_action_search.png. কিনিদের্শ করে। একইভাবে Title এটির বিউটি স্টর্যেরিরেসাসর্ ব্যবহার করে যা আপনার পর্ত্তেজেক্টর res/values/ ডিরেক্টরীর একটা এক্সএমএল কতৃৎক নিধর্যারিত, যা একটি সহজ ইউজার ইন্টারফেস তৈরী করা অধ্যয়্যে আলেচিত হয়েছে।

নোট: যখন আপনার অয়্যেপের জন্য আইকন এবং অন্য যেকোনও বিটময়্যাপ ইমেজ তৈরী করবেন, তখন এটা গুরুত্বপূর্ণ যে আপনি মাল্টিপল ভাসর্ন (ভিন্ন্নিভিন্ন্নি স্যস্করণ) পর্ত্তান করবেন যা বিভিন্ন স্কর্ন যনতব্ অনুষারে তৈরী করা। এটা ভিন্ন্নিভিন্ন্নি স্কর্নকে সাপোর্ট করা অনুষাীলনীতে বিস্তারিত আলেচনা করা হয়েছে।

আপনার অয়্যাপ যদি অয়ান্ডরয়েড ২.১ এর মতো কম ভাসর্নে কাজ করার মতো উপযুক্ত হতে সাপোর্ট লাইব্রেরী ব্যবহার করে থাকে, তাহলে android:নমসে স্পসেথেক showAsAction এটির বিউটি পায়্যাবে না। এর পরিবর্তে এই এটির বিউটি সাপোর্ট লাইব্রেরী কতৃৎক পর্ত্তদত্ত হয় এবং আপনাকে অবশ্যই আপনার নিজসব নমসে স্পসেকে নিধর্যারন করতে হবে এবং ওই নমসে স্পসেকে এটির বিউটি পর্ফিক্টি হিসাবে ব্যবহার করতে হবে। (একটি কাস্টম এক্সএমএল নমসে স্পসে আপনার অয়্যাপ নেমের উপর ভিত্তিত করে হওয়া উচিত, কিন্তু এটা যে কোন নাম হতে পারে যেটা আপনি চান এবং শুধুমাত্র ফাইলের সূবিধার মধ্য পর্ত্তেবশ্যেয়গয়্যে থাকেন

আপনি এটাে যাষণা করবেন)। উদাহরন:

res/menu/main_activity_actions.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:yourapp="http://schemas.android.com/apk/res-auto" >
<!-- Search, should appear as action button -->
<item android:id="@+id/action_search"
      android:icon="@drawable/ic_action_search"
      android:title="@string/action_search"
      yourapp:showAsAction="ifRoom" />
...
</menu>
```

একশন বারের একশন এন্ড করুন

একশন বারের মধ্যে মেনু আইটেম রাখতে, নিচের স্টেপ অনুসরণ করে `Menu` অবজেক্টের মধ্যে মেনু রিসোর্সটি বসে তৈরি করতে আপনার একটি অ্যাক্টিভিটির মধ্যে `onCreateOptionsMenu()` কল ব্যবহার করে মেনুটি বসে তৈরি করুন। উদাহরণ:

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu items for use in the action bar
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.main_activity_actions, menu);
    return super.onCreateOptionsMenu(menu);
}
```

একশন বাটনের রেসপন্স করা

যখন একজন ব্যবহারকারী একশন ওভারফোল্ডার মধ্যে কোন একশন বাটনে বা অন্য কোন আইটেমে চাপ দিবে, সিস্টেম আপনার একক্টিভিটির `onOptionsItemSelected()` কল ব্যবহার করে দেখতে কল করবে। আপনার এই পদ্ধতি বাস্তবায়নে কোন আইটেমটিতে চাপ দেওয়া হয়েছে তা ঠিক করতে নিম্নের `MenuItem` এর `getItemId()` কল করে- ফিরতি আইডি ভ্যালু যা আপনি ক্রেসপন্সিং ডেভেলপমেন্ট এর `android:id` এটি রিবউট করে যাষণা করেছিলেন তার সাথে ম্যাচ করে।

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle presses on the action bar items
    switch (item.getItemId()) {
        case R.id.action_search:
            openSearch();
            return true;
        case R.id.action_settings:
            openSettings();
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
```

লে-লে-ভল একটি টিভি টির জন্য আপ বাটন এ্যাড (সংযোজন) করা



ফিগার ৪. Gmail এ আপ বাটন

আপনার অ্যাপের মধ্যে যে সকল স্ক্রিন আছে তা আপনার অ্যাপের প্রধান
পর্বে বশপথনয় (একটি টিভি টি যা হোম স্ক্রিন নয়) একশনবারের **Up** বাটন
পের্সকরে, বয়বহারকারীকে দর অ্যাপ এর হায়ারারকিতে লিজিক্যাল
পয়ারেন্টি স্ক্রিনে নিভিগেট করার জন্য একটি উপায় দেওয়া উচিত।

যখন অ্যান্ড্রয়েড ৪.১ (API level 16) বা এর চেয়ে উন্নত সংস্করণের রান করে
অথবা সাপোর্ট লাইব্রেরী থেকে ActionBarActivity বয়বহার করা হয়, পারফর্ম
করা **Up** নিভিগেশন শুধুমাত্র চায় যে আপনি মিনিফেস্ট ফাইলে
পয়ারেন্টি একটি টিভি ডিকে লয়ার করে বন এবং একশনবারের জন্য **Up** বাটন
সিকর্য করবেন।

উদাহরন: এখানে আপনি মিনিফেস্ট একটি টিভি টির পয়ারেন্টি
ডিকে লয়ার করতে পারেন:

```
For example, here's how you can declare an activity's parent in the manifest:<application ... >
...
<!-- The main/home activity (it has no parent activity) -->
<activity
    android:name="com.example.myfirstapp.MainActivity" ...>
...
</activity>
<!-- A child of the main activity -->
<activity
    android:name="com.example.myfirstapp.DisplayMessageActivity"
    android:label="@string/title_activity_display_message"
    android:parentActivityName="com.example.myfirstapp.MainActivity" >
    <!-- Parent activity meta-data to support 4.0 and lower -->
    <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value="com.example.myfirstapp.MainActivity" />
</activity>
</application>
```

তারপর `setDisplayHomeAsUpEnabled()` কল করার মাধ্যমে **Up** বাটন হিসাবে অ্যাপ
আইকনকে সিকর্য করতে পারেন:

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_displaymessage);

    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    // If your minSdkVersion is 11 or higher, instead use:
    // getActionBar().setDisplayHomeAsUpEnabled(true);
}
```

কারণ সিস্টেমটি এখন জানে যে `DisplayMessageActivity` জন্য `MainActivity` হচ্ছে
পয়ারেন্টি একটি টিভি, যখন বয়বহারকারী **Up** বাটন পের্সকরে,
সিস্টেমটি পয়ারেন্টি একটি টিভি টিতে সঠিকভাবে নিভিগেশন করে-আপনার
Up বাটনের ইভেন্টে কোন হস্তক্ষেপ করার দরকার নেই।

আপ েনিভেগেশন সম্পকেৰ্ আরও তথয্ জানতে, েনিভেগেশন পর্দান করা
অধয্ ায়ি টে দেখুন।

একশন বারি টেকে স্টাইল করা

(<http://developer.android.com/training/basics/actionbar/styling.html>)

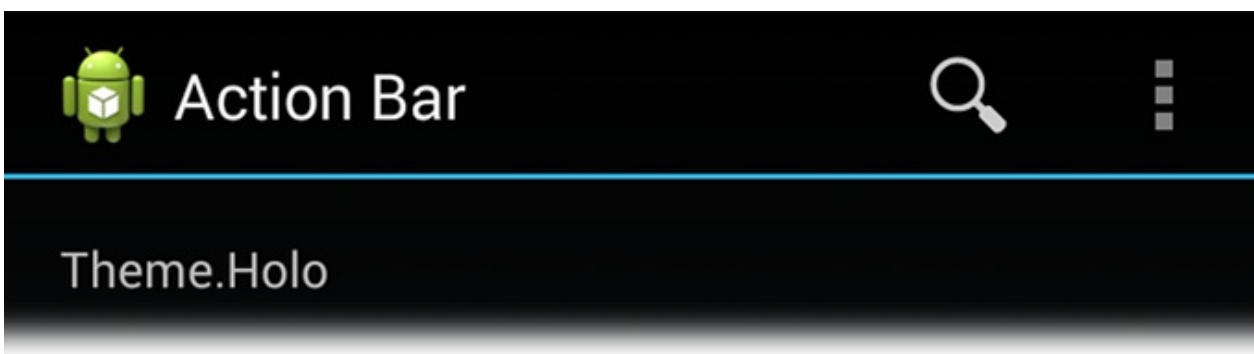
একশন বার আপনার বয় বহারকারীকে দর্শিত দেছে আপনার অয় াপে কাজ করতে এবং নেতিভে গট করার পরিচি ত এক উপায়, কিন্তু এর মানে এই নয় যে অনয় ানয় অয় াপে যেভ াবে থ াকে আপনার অয় াপে ও িঠকে সেভ াবে থ াকে ব। আপনি যদি একশন বারকে স্টাইল করে আপনার বয় ান ডপনেয় র সাথে ভেলে াভ াবে ফিট করতে চান তাহলে আপনি খুব সহজেই অয় ান ডর য়ে ডের style and theme রিেস াসর্ বয় বহার করে তা করতে পারেন (লিঙ্ক:

<http://developer.android.com/guide/topics/ui/themes.html>)।

অয় ান ডর য়ে ডে বশি কছু িবল্ টইন একি টি ভি টি থিম অন্তভূর ক্ত করে ছে যার মধ্যে রয়েছে “ডাকর্” বা “লাইট” একশন বার স্টাইল। আপনি আপনার একশন বারের দৃশ্যমানত াকে পরিবর্তন করতে চাইলে এই থিমগুলো াকে আরও পর্ স ারি ত/ি বস্ ত ত করতে পারবেন।

নেট: আপনি যদি একশন বারের জেনয় সাপে াটর্ লাইবেবর্ রী এপিআই (অচ ও) বয় বহার করে থ াকেন, অবশয় ই আপন াকে স্টাইলের Theme.AppCompat ফয় ামিলি বয় বহার (অথবা ওভাররাইড করতে হবে) করতে হবে (এপিআই ১১ বা এর উপের লেভেলে যে Thme.Helo ফয় ামিলি আছে তার পরিবর্তে বয় বহার করতে হবে)। আপনি যদি তাই করে থ াকেন, তাহলে পর্ তি ট া স্টাইল পর্ পারি টি যা আপনি ডিকে লয়ার করে ছেন অবশয় ই দুইবার ডিকে লয়ার করতে হবে, একবার প্ ল াটফর্ মের স্টাইল পর্ পারি টি (android:পর্ পারি টি) বয় বহার করে এবং একবার সাপে াটর্ লাইবেবর্ রীতে অন্তভূর ক্ত স্টাইল পর্ পারি টি বয় বহার করে (appcompat.R.attr পর্ পারি টি)। িবস্ ত ারি ত জানতে নিচের উদাহরণ দেখুন।

অ্যান্ড্রয়েড থীম ব্যবহার করুন



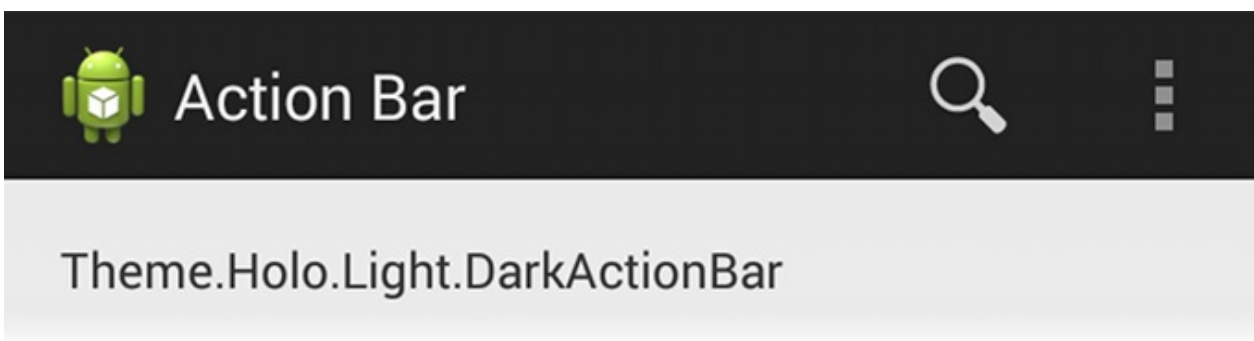
অ্যান্ড্রয়েড দুইটা ভেজলাইন একটি ডিভিটি থিম অন্তর্ভুক্ত করে যা একশনবারের জন্য কালার নির্দেশ করে:

- “ডার্ক” থিম এর জন্য Theme.Holo
- “লাইট” থিমের জন্য Theme.Holo.Light

আপনি এই থিম আপনার পূর্ণাঙ্গ অ্যাপে পর্যালোচনা করতে পারেন অথবা কোন সর্বতনতর একটি ডিভিটিতে এলিমেন্ট বা একক এলিমেন্ট এর জন্য ধ্রুৱক্ষয়বসব এটির বিউট এর সাথে আপনার মেনিফেস্ট ফাইল এ তাদের ডিকে ক্লয়ার করার মাধ্যমে।

উদাহরণের জন্য:

```
<application android:theme="@android:style/Theme.Holo.Light" ... />
```



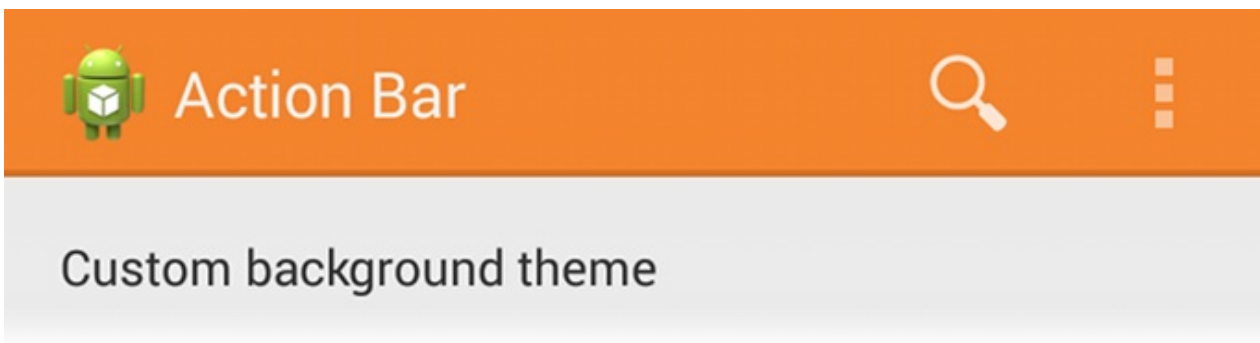
আপনি ডার্ক একশনবারও ব্যবহার করতে পারেন যেখানে একটি ডিভিটির বাকী অংশ Theme.Holo.Light.DarkActionBar থিম ডিকে ক্লয়ার করার মাধ্যমে লাইট কালার স্কিম ব্যবহার করে।

যখন সাপোর্ট লাইব্রেরী ব্যবহার করেন, আপনি অবশ্যই পরিবর্তন Theme.AppCompat থিম ব্যবহার করেন

- Theme.AppCompat: “ডার্ক” থিম এর জন্য
- Theme.AppCompat.Light: “লাইট” থিম এর জন্য
- Theme.AppCompat.Light.DarkActionBar “লাইট” থিম সাথে ডার্ক একশন বার এর জন্য

নিশ্চিত হোন যে আপনি যে একশন বার আইকন ব্যবহার করেছেন তা যেন আপনার একশন বারের কালার থেকে সম্পূর্ণ ভাবে বিভিন্ন হয়। আপনার সাহায্যের জন্য, Action Bar Icon Pack হোলে লাইট এবং হোলে ডার্ক একশন বার উভয় এর সাথেই ব্যবহার করার জন্য স্ট্যান্ডার্ড একশন আইকন অন্তর্ভুক্ত করে।

বয়স্ক গরু াউন্ড কাস্টমাইজ করুন



একশনবার বয়স্ক গরু াউন্ড পরিবর্তন করতে, আপনার একটি টিভিটির জন্য একটি কাস্টম থিম তৈরি করুন যা actionBarStyle প্যারামিটারে উভাররাইড করে। এই প্যারামিটারটি অনস্টাইল নির্দেশ করে যার মধ্যে আপনি একশনবার বয়স্ক গরু াউন্ড এর জন্য ড্রয়েবল রিসোর্স নির্দিষ্ট করতে background প্যারামিটারে উভাররাইড করতে পারবেন।

যদি আপনার অ্যাপস navigation tabs বা split action bar ব্যবহার করে থাকে, তাহলে আপনি এই বারের জন্য আলদা আলদাভাবে backgroundStacked এবং backgroundSplit প্যারামিটার ব্যবহার করে বয়স্ক গরু াউন্ড নির্দিষ্ট করতে পারেন।

সতর্কতা: এটা গুরুত্বপূর্ণ যে আপনি একটা যথাযথ প্যারেন্টিস ডিফল্ট করার করতে পারেন যাতে আপনি আপনার কাস্টম থিম এবং স্টাইল তৈরি করে স্টাইল পেয়ে থাকে। একটা প্যারেন্টিস স্টাইল ছাড়া, আপনার একশনবার অনেক স্টাইল প্যারামিটার ছাড়াই একশনবার হবে যদি না আপনি সর্বোচ্চ নিজেই তৈরি করে স্টাইল ডিফল্ট করার না করেন।

শুধুমাত্র অ্যান্ড্রয়েড ৩.০ এবং এর চেয়ে উন্নত সংস্করণের জন্য

যখন অ্যান্ড্রয়েড এবং এর চেয়ে উন্নত সংস্করণকে সাপোর্ট করবে,
আপনি একশনবার এর ব্যাকগ্রাউন্ডকে এভাবে বিনিময় করতে পারেন:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<!-- the theme applied to the application or activity -->
<style name="CustomActionBarTheme"
    parent="@style/Theme.Holo.Light.DarkActionBar">
    <item name="android:actionBarStyle">@style/MyActionBar</item>
</style>

<!-- ActionBar styles -->
<style name="MyActionBar"
    parent="@style/Widget.Holo.Light.ActionBar.Solid.Inverse">
    <item name="android:background">@drawable/actionbar_background</item>
</style>
</resources>
```

এরপর আপনার থিম আপনার পূর্ণ রাউন্ড অ্যাপে বাসবতন্ত্র
একটি টিভি টিতে পরিয়ে যা করতে পারেন:

```
<application android:theme="@style/CustomActionBarTheme" ... />
```

অ্যান্ড্রয়েড ২.১ এবং এর চেয়ে উন্নত সংস্করণের জন্য

যখন আপনি সাপোর্ট লাইব্রেরী ব্যবহার করেন, উপরের একই থিম পরিবর্তন হয়ে অবশ্যই এরকম দেখাবে:

res/values/themes.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<!-- the theme applied to the application or activity -->
<style name="CustomActionBarTheme"
    parent="@style/Theme.AppCompat.Light.DarkActionBar">
    <item name="android:actionBarStyle">@style/MyActionBar</item>

    <!-- Support library compatibility -->
    <item name="actionBarStyle">@style/MyActionBar</item>
</style>

<!-- ActionBar styles -->
<style name="MyActionBar"
    parent="@style/Widget.AppCompat.Light.ActionBar.Solid.Inverse">
    <item name="android:background">@drawable/actionbar_background</item>

    <!-- Support library compatibility -->
    <item name="background">@drawable/actionbar_background</item>
</style>
</resources>
```

এরপর আপনার থিম আপনার পূর্ণ রাউন্ড অ্যাপে বাসবতন্ত্র
একটি টিভি টেপ রেয়াগ করতে পারেন:

```
<application android:theme="@style/CustomActionBarTheme" ... />
```

টেক্সট কালার কাস্টমাইজ

আপনার একশন বার পরিবর্তন বা পরিমার্জন করতে, পরিত্যাগ টেক্সট এলিমেন্ট এর জন্য বিভিন্ন প্যারামিটার ওভাররাইড করা দরকার:

- একশন বার টাইটেল: একটি কাস্টম স্টাইল তৈরী করুন যা `textColor` প্যারামিটার থেকে সুনির্দিষ্ট করে এবং আপনার কাস্টম `actionBarStyle` এর মধ্যে `titleTextStyle` এর জন্য ওই স্টাইলকে সুনির্দিষ্ট করে।

নোট: `TitleTextStyle` তে কাস্টম স্টাইল পরে যোগ করতে

`TextAppearance.Holo.Widget.ActionBar.Title` কে প্যারামিটার স্টাইল হিসাবে ব্যবহার করা উচিত।

- একশন বার ট্যাবস: আপনার একটি টিভি টিম এর মধ্যে `actionBarTabTextStyle` কে ওভাররাইড করুন
- একশন বাটন: একটি টিভি টিম এর মধ্যে `actionMenuTextColor` কে ওভাররাইড করুন

শুধুমাত্র অ্যান্ড্রয়েড ৩.০ এবং এর চেয়ে উন্নত সংস্করণের জন্য

যখন অ্যান্ড্রয়েড ৩.০ এবং এর চেয়ে উন্নত সংস্করণকে সাপোর্ট করা হয়, তখন আপনার স্টাইল এক্সএমএল ফাইল এই রকম দেখাবে:

res/values/themes.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<!-- the theme applied to the application or activity -->
<style name="CustomActionBarTheme"
    parent="@style/Theme.Holo">
    <item name="android:actionBarStyle">@style/MyActionBar</item>
    <item name="android:actionBarTabTextStyle">@style/MyActionBarTabText</item>
    <item name="android:actionMenuTextColor">@color/actionbar_text</item>
</style>

<!-- ActionBar styles -->
<style name="MyActionBar"
    parent="@style/Widget.Holo.ActionBar">
    <item name="android:titleTextStyle">@style/MyActionBarTitleText</item>
</style>

<!-- ActionBar title text -->
<style name="MyActionBarTitleText"
    parent="@style/TextAppearance.Holo.Widget.ActionBar.Title">
    <item name="android:textColor">@color/actionbar_text</item>
</style>

<!-- ActionBar tabs text styles -->
<style name="MyActionBarTabText"
    parent="@style/Widget.Holo.ActionBar.TabText">
    <item name="android:textColor">@color/actionbar_text</item>
</style>
</resources>
```

অ্যান্ড্রয়েড ২.১ এবং এর চেয়ে উন্নত সংস্করণের জন্য

যখন সাপোর্ট লাইব্রেরী ব্যবহার করা হবে, তখন আপনার স্টাইল এক্সএমএল ফাইল এই রকম দেখাবে:

res/values/themes.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<!-- the theme applied to the application or activity -->
<style name="CustomActionBarTheme"
    parent="@style/Theme.AppCompat">
    <item name="android:actionBarStyle">@style/MyActionBar</item>
    <item name="android:actionBarTabTextStyle">@style/MyActionBarTabText</item>
    <item name="android:actionMenuTextColor">@color/actionbar_text</item>

    <!-- Support library compatibility -->
    <item name="actionBarStyle">@style/MyActionBar</item>
    <item name="actionBarTabTextStyle">@style/MyActionBarTabText</item>
    <item name="actionMenuTextColor">@color/actionbar_text</item>
</style>

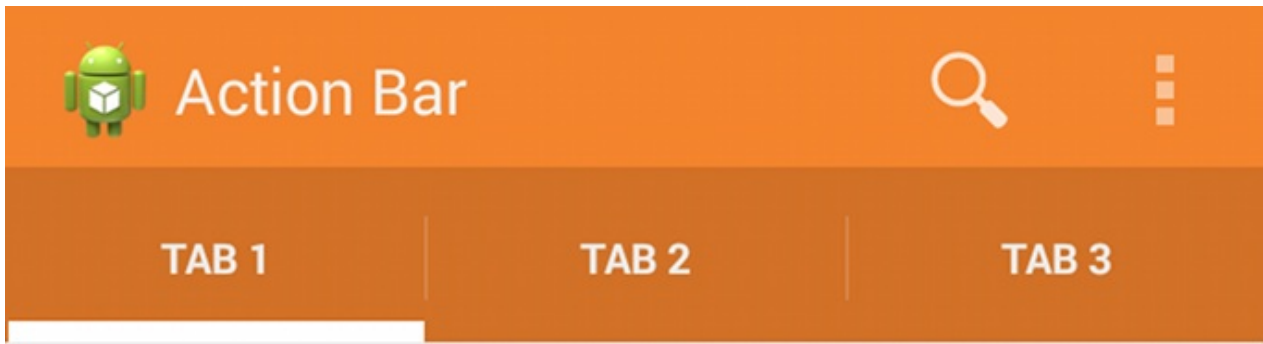
<!-- ActionBar styles -->
<style name="MyActionBar"
    parent="@style/Widget.AppCompat.ActionBar">
    <item name="android:titleTextStyle">@style/MyActionBarTitleText</item>

    <!-- Support library compatibility -->
    <item name="titleTextStyle">@style/MyActionBarTitleText</item>
</style>

<!-- ActionBar title text -->
<style name="MyActionBarTitleText"
    parent="@style/TextAppearance.AppCompat.Widget.ActionBar.Title">
    <item name="android:textColor">@color/actionbar_text</item>
    <!-- The textColor property is backward compatible with the Support Library -->
</style>

<!-- ActionBar tabs text -->
<style name="MyActionBarTabText"
    parent="@style/Widget.AppCompat.ActionBar.TabText">
    <item name="android:textColor">@color/actionbar_text</item>
    <!-- The textColor property is backward compatible with the Support Library -->
</style>
</resources>
```


ট্যাব ইন্ডিকেটর কাস্টমাইজ করা



Navigation tabs এর জন্য ব্যবহার করা ইন্ডিকেটর পরিবর্তন করতে, একটা একটি টিভি টিথিম তৈরী করুন যা actionBarTabStyle পর্পার্টিটরকে ওভাররাইড করে। এই পর্পার্টিট অনয স্টাইল রিসোর্সকে নির্দেশ করে যার মধ্যে আপনি background পর্পার্টিটকে ওভাররাইড করতে পারবেন যাতে একটি স্টেট-লিস্ট ড্রয়েবল নির্দিষ্ট করা উচিত।

নোট: একটি স্টেট-লিস্ট ড্রয়েবল গুরুত্বপূর্ণ যেতে বর্তমানে ট্যাব ব্যবহার করা উদ্দিষ্টে এর স্টেট সিলেক্ট করে যা অন্যান্য ট্যাবে বর্তমানে চলেই থাকে। একটি ড্রয়েবল রিসোর্স যা মাল্টিপল বাটন স্টেট ধারণ করে তাকীভাবে তৈরী করা যায় সেসম্পর্কে জানতে পড়ুন State List (<http://developer.android.com/guide/topics/resources/drawable-resource.html#StateList>) ডকুমেন্টেশন।

উদাহরণ এর জন্য, এখানে একটি স্টেট-লিস্ট ড্রয়েবল দেয়া হচ্ছে, যা একটি একশনবার ট্যাবে বর্তমান অবস্থি নির্দেশ করে। স্টেটস এর জন্য সুনির্দিষ্ট বয়স্ক গার্ডাউন্ড ইমেজকে ডেকে ফলাফল করে:

res/drawable/actionbar_tab_indicator.xml

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
```

```

<!-- Non focused states -->
<item android:state_focused="false" android:state_selected="false"
    android:state_pressed="false"
    android:drawable="@drawable/tab_unselected" />
<item android:state_focused="false" android:state_selected="true"
    android:state_pressed="false"
    android:drawable="@drawable/tab_selected" />

<!-- Focused states (such as when focused with a d-pad or mouse hover) -->
<item android:state_focused="true" android:state_selected="false"
    android:state_pressed="false"
    android:drawable="@drawable/tab_unselected_focused" />
<item android:state_focused="true" android:state_selected="true"
    android:state_pressed="false"
    android:drawable="@drawable/tab_selected_focused" />

<!-- STATES WHEN BUTTON IS PRESSED -->

<!-- Non focused states -->
<item android:state_focused="false" android:state_selected="false"
    android:state_pressed="true"
    android:drawable="@drawable/tab_unselected_pressed" />
<item android:state_focused="false" android:state_selected="true"
    android:state_pressed="true"
    android:drawable="@drawable/tab_selected_pressed" />

<!-- Focused states (such as when focused with a d-pad or mouse hover) -->
<item android:state_focused="true" android:state_selected="false"
    android:state_pressed="true"
    android:drawable="@drawable/tab_unselected_pressed" />
<item android:state_focused="true" android:state_selected="true"
    android:state_pressed="true"
    android:drawable="@drawable/tab_selected_pressed" />
</selector>

```

শুধুমাত্র অ্যান্ড্রয়েড ৩.০ এবং এর চেয়ে উন্নত সংস্করণের জন্য

যখন অ্যান্ড্রয়েড ৩.০ এবং এর চেয়ে উন্নত সংস্করণকে সাপোর্ট করে, তখন আপনার স্টাইল এক্সএমএল ফাইল এই রকম দেখাবে:

res/values/themes.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<!-- the theme applied to the application or activity -->
<style name="CustomActionBarTheme"
    parent="@style/Theme.Holo">
    <item name="android:actionBarTabStyle">@style/MyActionBarTabs</item>
</style>

<!-- ActionBar tabs styles -->
<style name="MyActionBarTabs"
    parent="@style/Widget.Holo.ActionBar.TabView">
<!-- tab indicator -->
    <item name="android:background">@drawable/actionbar_tab_indicator</item>
</style>
</resources>
```

অ্যান্ড্রয়েড ২.১ এবং এর চেয়ে উন্নত সংস্করণের জন্য

যখন সাপোর্ট লাইব্রেরী ব্যবহার করা হবে, তখন আপনার স্টাইল এক্সএমএল ফাইল এই রকম দেখাবে:

res/values/themes.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<!-- the theme applied to the application or activity -->
<style name="CustomActionBarTheme"
    parent="@style/Theme.AppCompat">
    <item name="android:actionBarTabStyle">@style/MyActionBarTabs</item>

    <!-- Support library compatibility -->
    <item name="actionBarTabStyle">@style/MyActionBarTabs</item>
</style>

<!-- ActionBar tabs styles -->
<style name="MyActionBarTabs"
    parent="@style/Widget.AppCompat.ActionBar.TabView">
    <!-- tab indicator -->
    <item name="android:background">@drawable/actionbar_tab_indicator</item>

    <!-- Support library compatibility -->
    <item name="background">@drawable/actionbar_tab_indicator</item>
</style>
</resources>
```

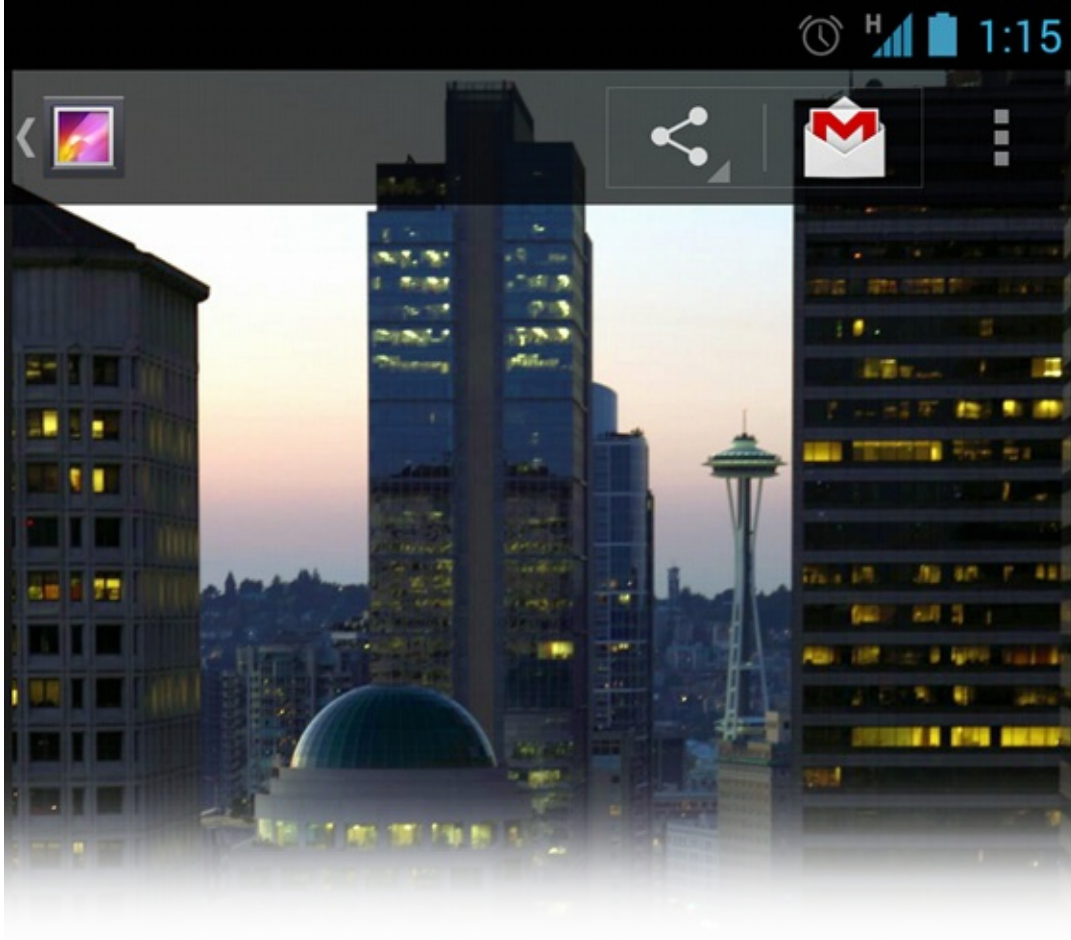
আরও কিছু তথ্য

- ActionBar গাইডেড তালিকা থেকে একশন বার এর জন্য আরও স্টাইল পর্যালোচনা দেখুন
- Style and Themes গাইড থেকে আরও শিখুন থিম কীভাবে কাজ করে
- এমনকি একশন বারের জন্য পূর্ণরঙ স্টাইল সম্পর্কে জানতে Android Action Bar Style Generator এ চেষ্টা করুন

একশন বার ওভারলে করা

(<http://developer.android.com/training/basics/actionbar/overlying.html>)

বাইন্ডিং ফল্ট, একশন বার আপনার একটি টিভি উইন্ডোর উপরের দিকে দৃশ্যমান হয়, আপনার একটি টিভি টির অবশিষ্ট লেআউট এর জন্য নির্ধারিত সেপারেশন পরিমাণ সামান্য কমিয়ে দেয়। ইউজারের যোগ্যে গের সময় যদি আপনি একশন বার হাইড বা শো করতে চান, আপনি এটা ActionBar এ `hide()` এবং `show()` দ্বারা করতে পারেন। যা হোক, এটা আপনার একটি টিভি টি কে লেআউট এর নতুন সাইজের উপর ভিত্তি করে রিকম্পাইট করা হবে।



ফিগার ১. ওভারলে মেড গ্যালারির একশন বার।

যখন একশন বার হাইড করে এবং শো করে, আপনার লেআউট রিসাইজ করা পরিহার করতে, আপনি একশন বারের জন্য **overlay mode** সিকর্য করতে পারেন। যখন ওভারলে মেড এ থাকবে, আপনার একটি টিভি টি লেআউট বিদ্যমান সেপারেশন সবটুকু ব্যবহার করেন, যেমন একশন বার সেখানে নেই এবং সিস্টেম আপনার লেআউটের সামনে একশন বার ড্র করে। এটা উপরের লেআউটের কিছু অংশ অস্পষ্ট করে দেয়, কিন্ত এখন একশন বার যখন হাইড বা শো করে, সিস্টেমটি লেআউট রিসাইজ করার কোন পর্যায়ে জড়িত নেই এবং এর পরিবর্তন নিখুঁত।

পরামর্শ: আপনি যদি চান আপনার লেআউট একশন বারের পেছনে আংশিকভাবে দৃশ্যমান হবে, একশন বারের জন্য আংশিক সবচ্ছন্ন বয়াকগরাউন্ড সহ একটি কাস্টম স্টাইল তৈরি করুন, যেমন ফিগার ১ দেখানো হয়েছে। একশন বার বয়াকগরাউন্ড কীভাবে নির্ধারণ করা হয় তা জানতে [Styling the Action Bar](#) পড়ুন।

ওভারলে মোড সিকর্য করুন

একশন বারের জন্য ওভারলে মোড সিকর্য করতে, আপনার একটি কাস্টম থিম তৈরী করে পরে যাজন যা বিদ্যমান একশন বার থিমকে বিসৃত করে এবং true তে android:windowActionBarOverlay পরপারিটে সট করুন।

তর্ অয ান ডর েড ৩.০ এব ং এর েচেয়ে উন নত স ংস্করনের জনয

যদি আপন ার minSdkVersion ১১ বা এর উপেরর লেভেলে সেট হয়ে থাকে, আপন ার
কাস্টম থিম এর উচিত আপন ার পয ারে রন্ট থিম হিসাবে Theme.Holo থিম (অথবা
এই গেতে রর্ অনয ি থিম) ব্যবহার করা। উদাহরনের জনয :

```
<resources>
<!-- the theme applied to the application or activity -->
<style name="CustomActionBarTheme"
    parent="@android:style/Theme.Holo">
    <item name="android:windowActionBarOverlay">true</item>
</style>
</resources>
```

অ্যান্ড্রয়েড ২.১ এবং এর চেয়ে উন্নত সংস্করণের জন্য

অ্যান্ড্রয়েড ৩.০ এর চেয়ে নীচের সংস্করণে সামঞ্জস্যপূর্ণ করার
জন্য যদি আপনার অ্যাপ সাপোর্ট লাইব্রেরী ব্যবহার করে থাকে,
আপনার কাস্টম থিম এর উচিত আপনার প্যারেন্ট থিম হিসাবে Theme.AppCompat
থিম (অথবা একই গেটের কোন একটি) ব্যবহার করা। উদাহরণের জন্য:

```
<resources>
<!-- the theme applied to the application or activity -->
<style name="CustomActionBarTheme"
    parent="@android:style/Theme.AppCompat">
    <item name="android:windowActionBarOverlay">true</item>

    <!-- Support library compatibility -->
    <item name="android:windowActionBarOverlay">true</item>
</style>
</resources>
```

এছাড়াও এটা মেনাটিশ করে যে, এই থিম windowActionBarOverlay স্টাইলের জন্য
দুইটা অর্থ বহন করে, একটি android:পিপারফিক্স সহ এবং অন্যটি android:
পিপারফিক্স ছাড়া। android:পিপারফিক্স সহিটি অ্যান্ড্রয়েড এর
সংস্করণগুলোর জন্য যা প্ল্যাটফর্মের স্টাইল অন্তর্ভুক্ত করে এবং
android:পিপারফিক্স ছাড়া অন্যটি পুরাতন সংস্করণের জন্য
সাপোর্ট লাইব্রেরী থেকে স্টাইল রিড করে।

লেআউট টপ মার্জিন নিদিষ্ট করুন

যখন একশন বার ওভারলে মোড থাকে, এটা আপনার লেআউটে টরিকিছু অংশকে অস্পষ্ট করে দিতে পারে যা দৃশ্যমান থাকার পরে যোজন। এটা নিশ্চিত করতে, এই আইটেমগুলোকে সব সময় একশন বারের নীচে রাখতে, actionBarSize দবারা নিম্নরূপে উচ্চতাব্যবহার করে ভিউ এর উপরে হয় মার্জিন নাহয় প্যাডিং এড করুন। উদাহরনের জন্য:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingTop="?android:attr/actionBarSize">
    ...
</RelativeLayout>
```

আপনি যদি একশন বার এর জন্য সাপোর্ট লাইব্রেরী ব্যবহার করে থাকেন, আপনার android:পিপারফিক্সটি সরিয়ে ফেলা দরকার। উদাহরনের জন্য:

```
<!-- Support library compatibility -->
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingTop="?attr/actionBarSize">
    ...
</RelativeLayout>
```

এই ক্ষেত্রে, পিপারফিক্স ছাড়া ?attr/actionBarSize ভ্যালুটি সকল স্যস্করেনে কাজ করে, এমনকি অ্যান্ডরয়েড ৩.০ এবং এর চেয়ে উন্নত স্যস্করেনেও।

বিভিন্ন ডিভাইসকে সাপোর্ট করানো

(<http://developer.android.com/training/basics/supporting-devices/index.html>)

সারা বিশ্বে অ্যান্ড্রয়েড ডিভাইস বিভিন্ন আকার আকৃতি নিয়ে আসে। এই যে বহুবিধ ধরনের ডিভাইসের বিস্তৃত এলাকা, এখানে আপনি পাবেন আপনার অ্যাপকে একটা বিশাল সংখ্যক গর্হিত করে পেছানোর সুযোগ। অ্যান্ড্রয়েডে যতটুকু সফল হওয়া সম্ভব তা হওয়ার জন্য আপনাকে আপনার অ্যাপকে এমনভাবে তৈরী করতে হবে যা ডিভাইসের বিভিন্ন কনফিগারেশনের মানানসই হয়। কিন্তু বৈচিত্র্যের বিষয়টা আপনাকে বিবেচনায় রাখতে হবে যেমন, বিভিন্ন ল্যাংগুয়েজ, স্ক্রিন এর সাইজ, এবং অ্যান্ড্রয়েড প্ল্যাটফর্মের বিভিন্ন সংস্করণ ইত্যাদি।

এই ক্লাস আপনাকে শেখাবে, কীভাবে মৌলিক প্ল্যাটফর্ম বৈশিষ্ট্যগুলো ব্যবহার করা যায় যা বিকল্প পরিবেশে এবং অন্যান্য বৈশিষ্ট্যগুলোকে নিয়ন্ত্রণ করে যাতে একটা সিমুলেটেড অ্যাপলিকে শন প্যাকেজ (APK) ব্যবহার করে, বিভিন্ন অ্যান্ড্রয়েড-উপযুক্ত ডিভাইসে আপনার অ্যাপ অপটিমাইজড ইউজার এক্সপেরিয়েন্স সরবরাহ করতে পারবে।

এই অধ্যায়ের অনুশীলন সমূহ

ভিন্ন ভিন্ন ভাষাকে সাপেপাটর্ করা

শিখন কীভাবে বিবকলপিস্টর ঞিরেসাসর িদেয় অনেকগুলো ভাষাকে সাপেপাটর্ করা যায়।

ভিন্ন ভিন্ন সিস্করনকে সাপেপাটর্ করা

শিখন বিভিন্ন সিস্করন সাইজ এবং যণতব্ এর জন্য কীভাবে ইউজার এক্সপেরিয়েন্সকে অপটিমাইজ করা যায়।

ভিন্ন ভিন্ন প্ল্যাটফর্ম সংস্করণকে সাপেপাটর্ করা

শিখন অ্যান্ডরয়েড এর নতুন সংস্করণে যে এপিআই গুলো (অচও ঞ) পাওয়া যায় তা কীভাবে অ্যান্ডরয়েডের পুরাতন সংস্করণে বয়্ বহার করা যায়।

ভিন্ন ভিন্ন ভাষাকে সাপোর্ট করা

(<http://developer.android.com/training/basics/supporting-devices/languages.html>)

আপনার অ্যাপকে ডেথেকে ইউজার ইন্টারফেস সিস্টেমকে বের করে ফেলা এবং একটা বাইরের ফাইলে রাখা, সবসময়ই একটা ভালো চর্চা। পরিত্যাগ অ্যান্ড্রয়েড পরজেক্টরিরে সাধারণ ডিরেক্টরী দিয়ে অ্যান্ড্রয়েড এটাকে সহজ করে দেয়।

আপনি যদি আপনার পরজেক্ট অ্যান্ড্রয়েড এসডিকে টুলস ব্যবহার করে তৈরী করে থাকেন, টুলসটি পরজেক্টর একদম উপরের স্তরে একটা res/ ডিরেক্টরী তৈরী করে। এই res/ ডিরেক্টরীর মধ্যে বিভিন্ন রিসোর্স টাইপের জন্য সাবডিরেক্টরী আছে। এখানে আরও কিছু ডিরেক্টরী আছে যেমন res/values/strings.xml যা আপনার সিস্টেম ভাষা লুকিয়ে ধারণ করে।

লোকালিডেরক্টরী এবং স্ট্রিং ফাইল তৈরী করা

আরও ভাষার জন্য সাপোর্ট এবং এড করে ত, res/ এর ভিতরে অতিরিক্ত values ডিরেক্টরী তৈরী করুন যা ডিরেক্টরীর একদম শেষে একটি হাইফেন এবং আইএসও কান্ট্রি ট্রেকোড অন্তর্ভুক্ত করে।

উদাহরণস্বরূপ, values-es/ হচ্ছে ল্যাটিন গুয়েজ কোড "es" দিয়ে লোকালের জন্য সাধারণ রিসোর্স গুলো লোক ধরের রাখার ডিরেক্টরী। অ্যান্ডর্য়েড ডিভাইস চলাকালীন সময়ে এর লোকাল সেটিং অনুসারে যথাযথ রিসোর্স লোড করে।

আপনি যদি ঠিক করে থাকেন কোন ভাষাকে সাপোর্ট করবেন, রিসোর্স সাব ডিরেক্টরী এবং স্ট্রিং রিসোর্স ফাইল তৈরী করুন।

উদাহরণস্বরূপ:

```
MyProject/  
res/  
  values/  
    strings.xml  
  values-es/  
    strings.xml  
  values-fr/  
    strings.xml
```

পরতিটা লোকালের জন্য স্ট্রিং ভ্যালু যথাযথ ফাইলে এবং এড করুন।

কাজ চলাকালীন সময়ে, অ্যান্ডর্য়েড সিস্টেম চলিত সময়ে ইউজারের ডিভাইসে যে লোকাল সেট করা হয়েছে তার উপর ভিত্তি করে সামগ্রিক সম্পূর্ণ স্ট্রিং রিসোর্স এর সেট ব্যবহার করে।

উদাহরণস্বরূপ, বিভিন্ন ভাষার জন্য কিছু ভিন্ন স্ট্রিং রিসোর্স নীচে দেওয়া আছে।

ইংরেজী (ডিফল্ট লোকাল), /values/strings.xml:

```
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
  <string name="title">My Application</string>  
  <string name="hello_world">Hello World!</string>  
</resources>
```

স্প্যানিশ, /values-es/strings.xml:

```
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
  <string name="title">Mi Aplicación</string>  
  <string name="hello_world">Hola Mundo!</string>  
</resources>
```

ফরাসী, /values-fr/strings.xml:

```
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
  <string name="title">Mon Application</string>  
  <string name="hello_world">Bonjour le monde !</string>  
</resources>
```

নোট: আপনি যে কোন রিসোর্সে টাইপের উপরে লোকাল ক্যাশালিফায়ার (অথবা যে কোন কনফিগারেশন ক্যাশালিফায়ার) ব্যবহার করতে পারেন, যেমন আপনি যদি আপনার বিটম্যাপ ড্রয়েবল এর লোকালাইজড সংস্করণ সরবরাহ করতে চান। আরও তথ্যের জন্য Localization (<http://developer.android.com/guide/topics/resources/localization.html>) দেখুন।

সিস্টার্ম রিসোর্স বয়বহার করুন

আপনি আপনার রিসোর্সকে কে ডে এবং অন্য কোন এক্সএমএল ফাইলে এ `<string>` এ লিখে নেন টার `name` এটি টার বিউট দবারা নিখরারিত রিসোর্স নাম বয়বহার করে আপনার সিস্টার্ম রিসোর্সকে উল্লেখ করতে পারেন।

আপনার রিসোর্সকে কে ডে আপনি সিনট্যাক্স `R.string.<string_name>` এর সাথে একটি সিস্টার্ম রিসোর্সকে উল্লেখ করতে পারেন। এখানে নাম ধরনের পদ্ধতি আছে যা সিস্টার্ম রিসোর্সকে এই উপায়ে গ্রহণ করে।

উদাহরণস্বরূপ:

```
// Get a string resource from your app's Resources
String hello = getResources().getString(R.string.hello_world);

// Or supply a string resource to a method that requires a string
TextView textView = new TextView(this);
textView.setText(R.string.hello_world);
```

অন্য এক্সএমএল ফাইলে আপনি সিনট্যাক্স `@string/<string_name>` এর সাথে একটি সিস্টার্ম রিসোর্সকে উল্লেখ করতে পারেন যখনই এক্সএমএল এটি টার বিউট একটি টিস্টার্ম ভ্যালু গ্রহণ করে।

উদাহরণস্বরূপ:

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/hello_world" />
```

ভিন্ন ভিন্ন সিস্কর্নকে সাপোর্ট করা

(<http://developer.android.com/training/basics/supporting-devices/screens.html>)

অ্যান্ড্রয়েড দুইটা সাধারণ পর্পাটিটর্ ব্যবহার করে ডিভাইস সিস্কর্নকে শরণীবিভাজন করে: তাহেচ্ছ এর সাইজ এবং ঘণতব্। আপনার আশা করা উচিত যে আপনার অ্যাপ সাইজ এবং ঘণতব্ উভয়ের রেঞ্জের মধ্যে থেকে সিস্কর্ন সহ ডিভাইসে ইনস্টল হবে। এর জন্য আপনার উচিত কিছু বিকল্প পিরেসাসর্ অন্তর্ভুক্ত করা যা বিভিন্ন মাপের সিস্কর্ন সাইজ এবং ঘণতেব্র জন্য আপনার অ্যাপের দৃশ্যমানতাকে অপটিমাইজ করেছে।

- এখনে চার ধরনের সাবর্জনীন সাইজ রয়েছে: স্মল, নরমাল, লার্জ এবং এক্সটর্ লার্জ
- এবং চার ধরনের সাবর্জনীন ঘণতব্ রয়েছে: লো (ldpi), মিডিয়াম (mdpi), হাই (hdpi) এবং এক্সটর্ হাই (xhdpi)

বিভিন্ন লোআউট এবং বিটম্যাপ ডিকে লয়ার করে বিভিন্ন সিস্কর্নের জন্য ব্যবহার করতে চাইলে, আপনাকে অবশ্যই এই বিকল্প পিরেসাসর্গুলোকে আলাদা ডিরেক্টরীতে রাখতে হবে, বিভিন্ন ভাষার স্টর্ এবং এর কেষ্টে যেনো ভাবে করেছে ন ঠিক সেভাবে বই করতে হবে।

আরও সতর্ক থাকতে হবে যে সিস্কর্ন ওরিয়েন্টেশন (ল্যান্ডস্কেপ অথবা পোর্ট্রেইট) এর বিষয়টাও সিস্কর্ন সাইজে রৈচিত্রের মধ্যে বিবেচনা করতে হবে, পর্তিটা ওরিয়েন্টেশনে ইউজার এক্সপেরিয়েন্স অপটিমাইজ করতে অনেক অ্যাপের লোআউট পূর্ণি বর্বেচনা করা উচিত।

ভিন্ন ভিন্ন লেআউট তৈরী করুন

ভিন্ন ভিন্ন সিস্করন সাইজে আপনার ইউজার এক্সপেরিয়েন্স সিকর্য করতে, আপনার উচিত পরিতটাসিস্করন সাইজের জনেয় একটি সবতন্তর লেআউট এক্সএমএল ফাইল তৈরী করা যা আপনিসাপোর্ট করতে চান। পরিতটালেআউট যথায়থিরিসাসসর্ ডিরেক্টরীতে সেভ করা উচিত, একটি <screen_size> সাফিক্স সহ দেয়ানাম। উদাহরণস্বরূপ, একটি লাজর সিস্করনের জনয় সবতন্তর লেআউট res/layout-large/ এর অধীনে সেভ করা উচিত।

নোট: সিস্করনকে সম্পূর্ণরূপে বমানানসই করার জনয় অ্যান্ডরয়েড সর্য়কিয়ন্তাবে আপনার লেআউটের সমতারক্ষাকরে। তাই ভিন্ন ভিন্ন সিস্করন সাইজের জনয় আপনার লেআউটের ইউজার ইন্টারফেস এলিমেন্টের পরিপূর্ণরূপে সাইজ নিয়ৈচিন্ততহওয়ার কোন কারণ নেই কিন্তু পরিবর্তের লেআউট কাঠামোর উপরে ফেকাস করা ইউজার এক্সপেরিয়েন্স পরভাব ফেলে (যেমন গুরুতব্পূর্ণরূপে ভিউ এর সাইজ অথবা অবস্থান সিবলিঙ ভিউ এর সাথে সম্পিকর্ত)।

উদাহরণস্বরূপ, এই পর্জেক্ট লাজর সিস্করনের জনয় একটা ডিফল্ট লেআউট এবং একটি বিকল্প লেআউট অন্তর্ভুক্ত করে:

```
MyProject/  
res/  
  layout/  
    main.xml  
  layout-large/  
    main.xml
```

ফাইলটির নাম অবশ্যই একই হতে হবে, কিন্ত অনুরূপ সিস্করন সাইজের জনয় অপটিমাইজড ইউজার ইন্টারফেস সরবরাহ করার জনয় তাদের বিষয়বস্তু ভিন্ন থাকে।

আপনার অ্যাপে সাধারণভাবে শুধুমাত্র লেআউট ফাইলটি উল্লেখ করুন:

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
}
```

সিস্টেমটি ডিভাইসে সিস্করনের সাইজ যেখানে আপনার অ্যাপরান করেছে তার উপর ভিত্ত করে যথায়থ লেআউট ডিরেক্টরী থেকে লেআউট ফাইলটি লোড করে। কীভাবে অ্যান্ডরয়েড যথায়থিরিসাসসর্ বাছাই করে এসম্পর্কে আরও তথ্য Providing Resources গাইডে আছে।

ভিন্ন একটা উদাহরণ হিসাবে, এখানে দেয়া হলে ল্যান্ডস্কেপ ওরিয়েন্টেশনের জনয় বিকল্প লেআউট সহ একটি পর্জেক্ট:

```
MyProject/  
res/  
  layout/  
    main.xml  
  layout-land/  
    main.xml
```

বাই ডিফল্ট, layout/main.xml ফাইলটি পেোর্ট ইট ওরিয়েন্টেশন এর জনয় ব্যবহার করা হয়।

আপনি যদি ল্যান্ডস্কেপের জন্য একটি সেপশাল লেআউট দিতে চান, ক্যান
লাজর হিস্করনের ক্ষেত্রেও, তখন আপনার large এবং land উভয়
কেয়ালিফায়ার ব্যবহার করা উচিত:

```
MyProject/  
  res/  
    layout/          # default (portrait)  
      main.xml  
    layout-land/     # landscape  
      main.xml  
    layout-large/    # large (portrait)  
      main.xml  
    layout-large-land/ # large landscape  
      main.xml
```

নোট: অ্যান্ড্রয়েড ৩.২ এবং এর চেয়ে উন্নত সংস্করণ হিস্করন সাইজ
নিধরারণে উন্নত পদ্ধতিতে সাপোর্ট করে, যা আপনাকে ডিজিটাল
সব্বাধীন পক্ষে সল এর শেত্রে সবরনিম্নপেরস্থ এবং উচ্চতার ভিত্তিতে
হিস্করন সাইজের জন্য রিসেসাসরকে সুনির্দিষ্ট করতে অনুমোদন
দেয়। এই অনুশীলনী এই নতুন কৌশলকে ধারণ করেন না। আরও তথ্যের জন্য,
বিভিন্ন ধরনের হিস্করনের জন্য ডিজাইন করা অধ্যয়টি পড়ুন।

বিভিন্ন নথরনের বিটময্ াপ তৈরী করুন

আপন াকে অবশয্ ই সবসময্ বিটময্ াপ রিসেসাসর্ দেয া উচিত য া সাধ ারন যণতব্ ম াপক াঠি অনুস ারে যথ াযথ ভ াবে প রিম াপ হয়: যেমন লে া, মি ডিয় াম, হ াই এব ং এক্ সটর্ া হ াই। এট া আপন াকে ভ ালে া গর্ াফিক য্ াল কে ায় ালি টি অজর্নে এব ং সকল সিস্কর্নে কাজ করে ত সহ ায়ত া করে ব।

এই ইমেজ তৈরী করতে, আপন ার উচিত ভেক্টর ফর্মেট এ 'ব' রিসেসাসর্ নিয়ে শুরুর করা এব ং নিম্নে াকৃত সাইজে সকল বয্ বহার করে প রি ত ট া যণতে ব্ র জনয্ ইমেজ তৈরী করা:

- xhdpi: 2.0
- hdpi: 1.5
- mdpi: 1.0 (baseline)
- ldpi: 0.75

এর ম ানে আপনি যদি xhdpi ডিভ াইসে সর জনয্ ২০০x২০০ ইমেজ তৈরী করেন, তাহলে আপন ার একই রিসেসাসর্ hdpi ডিভ াইসে সর জনয্ ১৫০x১৫০, mdpi ডিভ াইসে সর জনয্ ১০০x১০০ এব ং ldpi ডিভ াইসে সর জনয্ ৭৫x৭৫ এ তৈরী করা উচিত।

এরপর যথ াযথ ডর্য়েবল রিসেসাসর্ ডিরেক্টরীতে ফ াইলি টি রেখে দিন:

```
MyProject/  
res/  
  drawable-xhdpi/  
    awesomeimage.png  
  drawable-hdpi/  
    awesomeimage.png  
  drawable-mdpi/  
    awesomeimage.png  
  drawable-ldpi/  
    awesomeimage.png
```

যে কে ান সময্ আপনি @drawable/awesomeimage উলে লখ করতে প ারেন, সিসেস্ টমি টি সিস্কর্নের যণতে ব্ র উপর ভিত্তি করে যথ াযথ বিটময্ াপ বেছে নেন।

নে াট: লে াডেনজি টি/যণতব্ (ldpi) রিসেসাসর্ সব সময্ পর্য়ে াজনীয় নয়। যখন আপনি hdpi এসেট পর্দ ান করেন, সিসেস্ টমি টি যথ াযথ ভ াবে ldpi সিস্কর্নের সাথে মান ানসই করতে একে প রিম াপ করে অধর্ক ন ামিয়ে নিয়ে আসে।

আপন ার অয্ াপে সর জনয্ আইকন এসেট তৈরী করার বিষয়ে আরও জ ানতে দেখুন [Iconography design guide \(http://developer.android.com/design/style/iconography.html\)](http://developer.android.com/design/style/iconography.html)

ভিন্ন প্ল্যাটফর্ম সংস্করণকে সাপোর্ট করা

(<http://developer.android.com/training/basics/supporting-devices/platforms.html>)

যখন অ্যান্ড্রয়েডের সর্বশেষ সংস্করণে আপনার অ্যাপের জন্য
এপিআই (APIs) প্রদান করে, আপনার উচিত অ্যান্ড্রয়েডের পুরাতন
সংস্করণকে সাপোর্ট করতে পারেন সেভাবে কাজ করা যতক্ষণ না আরও
ডিভাইস আপডেট না হচ্ছে। এই অনুশীলনী আপনাকে দেখাবে কীভাবে
সর্বশেষ এপিআই (APIs) থেকে সুবিধা নিয়ে পুরাতন সংস্করণকে সাপোর্ট
করতে পারেন সেভাবে কাজ করা।

অ্যান্ড্রয়েডের প্রতিটা সংস্করণে রান করা সিকর্য ডিভাইস এর
বন্টন দেখাতে Platform Versions এর জন্য ড্যাশবোর্ড টিনিয়মিতভাবে
আপডেট হচ্ছে, এটা ডিভাইসের সংখ্যার উপর ভিত্তি করে করা, যা গুগল
সেটআপের পরিদর্শন করে। সাধারণভাবে, ৯০% সিকর্য ডিভাইসকে সাপোর্ট
করাটা একটা ভালো চর্যা, যখন আপনার অ্যাপ সর্বশেষ সংস্করণের
প্রতি লক্ষ্য নির্ধারণ করে থাকে।

পরামর্শ: কিছু অ্যান্ড্রয়েড সংস্করণকে সবচেয়ে ভালো বৈশিষ্ট্য
এবং কায়রক্সমতাপ্রদান করার জন্য আপনার অ্যাপ Android Support Library
ব্যবহার করা উচিত, যা আপনাকে পুরাতন সংস্করণের উপর কিছু নতুন
প্ল্যাটফর্ম এপিআই (APIs) ব্যবহার অনুমোদন করে।

মিনিমাম (ন্যূনতম) এবং টার্গেট এপিআই লেভেল সুনির্দিষ্ট করা

AndroidManifest.xml ফাইলটি আপনার অ্যাপ সম্পর্কে বিস্তারিত আলাচনা করে থাকে এবং কান সংস্করণের অ্যান্ডরয়েড এটাকে সাপোর্ট করবে তা চিহ্নিত করে। বিশেষ করে, <uses-sdk এলিমেন্ট এর জন্য minSdkVersion এবং targetSdkVersion এটি র‍্যাবিউটিটি সবার নিন্মে এপিআই লেভেল চিহ্নিত করে যেটা সহ আপনার অ্যাপ সব কিছুর সাথে এবং সর্বোচ্চ এপিআই লেভেল খাপ খাওয়াবেনার মতো যার বিপরীতে আপনি আপনার অ্যাপ ডিজাইন করেছেন এবং টেস্ট করেছেন।

উদাহরণস্বরূপ:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android" ... >
  <uses-sdk android:minSdkVersion="4" android:targetSdkVersion="15" />
  ...
</manifest>
```

যেহেতু অ্যান্ডরয়েডে ডরনতুন সংস্করণ এসেছে, কিছু স্টাইল এবং আচরণ পরিবর্তন হতে পারে। আপনার অ্যাপকে এই পরিবর্তনগুলো সুরক্ষিত করার বিষয়টুকরতে দিতে এবং নিশ্চিত করতে যে আপনার অ্যাপ সকল ডিভাইসে সমানানসই হবে, সর্বশেষ সংস্করণের সাথে খাপ খাওয়াতে targetSdkVersion ভ্যালু সেট করা উচিত।

রানটাইমে সিস্টেম সংস্করণকে চেক করুন

অ্যান্ড্রয়েড Build কনস্ট্রাক্টর ক্লাসের মধ্যে, প্রতিটা প্ল্যাটফর্ম সংস্করণের জন্যে চিহ্নিত একটি ইউনিক (সবতন্ত্র) কোড। আপনার অ্যাপের মধ্যে থেকে এই কোডগুলো ব্যবহার করে কনডিশনাল তৈরি করা যাবে কোডটিকে নিশ্চিত করে এবং যা উচ্চতর এপিআই লেভেল সংযুক্ত হওয়ার উপর নির্ভর করে, শুধুমাত্র তখন যখন ওই এপিআই সিস্টেমে থাকে।

```
private void setUpActionBar() {  
    // Make sure we're running on Honeycomb or higher to use ActionBar APIs  
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {  
        ActionBar actionBar = getActionBar();  
        actionBar.setDisplayHomeAsUpEnabled(true);  
    }  
}
```

নোট: যখন এক্সএমএল রিসোর্সের এপারিসরং করা হয়, অ্যান্ড্রয়েড এক্সএমএল এটি রিবিউটকে পরিহার করে যেটাকে বর্তমান ডিভাইস সাপোর্ট করে না। সুতরাং আপনি নিরাপদভাবে এক্সএমএল এটি রিবিউট ব্যবহার করতে পারেন যেটাকে শুধুমাত্র নতুন সংস্করণ সাপোর্ট করে থাকে পুরাতন সংস্করণের বৈকিক সম্পর্কে কোন ধরনের দুশ্চিন্তা ছাড়াই, যখন এটা ওই কোডকে একাউন্ট করার করে।
উদাহরণস্বরূপ, আপনি যদি `targetSdkVersion="11"` টি করে থাকেন, আপনার অ্যাপ অ্যান্ড্রয়েড ৩.০ বা এর চেয়ে উন্নত সংস্করণের ক্ষেত্রে বাইন্ডিং ফল্ট ActionBar অন্তর্ভুক্ত করে। তখন একশন বারের মেনু আইটেম সেট করতে, আপনার মেনু রিসোর্সের এক্সএমএল এ `android:showAsAction="ifRoom"` সেট করা দরকার। এটা কর্তৃক-ভাস্কর্য এক্সএমএল ফাইলে করাটো নিরাপদ, কারণ পুরাতন সংস্করণের অ্যান্ড্রয়েড সব্ভাব্যতাই `showAsAction` এটি রিবিউট পরিহার করে (তাই, `res/menu-v11/` এ আপনার কোন পৃথক সংস্করণের দরকার নেই)।

প্ল্যাটফর্ম স্টাইল এবং থিম ব্যবহার

অ্যান্ড্রয়েড ইউজার এক্সপেরিয়েন্স থিম প্রদান করে যা অ্যাপকে মৌলিক/আসল অপারেটিং সিস্টেম এর লুক এবং ফিল দেয়। এই থিমগুলো মিনিফেস্ট ফাইলের মধ্যে থেকে আপনার অ্যাপ পরিয়ে যাগ হতে পারে। এই বিবল্ট ইন থিম এবং স্টাইল ব্যবহার করে আপনার অ্যাপ সবার্ভািবকভাবে নতুন রিলিজ হওয়া পরিত্রা অ্যান্ড্রয়েডের সবার্ভেষলুক এবং ফিলকে অনুসরণ করবে।

আপনার একটি টিভি টেকে ডায়ালগ বক্স এর মতো করে তৈরী করতে:

```
<activity android:theme="@android:style/Theme.Dialog">
```

আপনার একটি টিভি টেকে সবচ্ছব অ্যাকগর্ডাউন্ড সহ তৈরী করতে:

```
<activity android:theme="@android:style/Theme.Translucent">
```

/res/values/styles.xml তে নিধর্ারিত আপনার নিজস্ব কাস্টম থিম পরিয়ে যাগ করতে:

```
<activity android:theme="@style/CustomTheme">
```

আপনার সম্পূর্ণ অ্যাপ (সকল একটি টিভি ট) একটি থিম পরিয়ে যাগ, <application> এ লিমেন্ট android:theme এ ট্রিবিউট এ্যাড করুন:

```
<application android:theme="@style/CustomTheme">
```

থিম তৈরী এবং ব্যবহার সম্পর্কিত আরও তথ্যের জন্য [Styles and Themes](#) গাইডিটি পড়ুন।

একটি টিভি টলাইফসাইকেল বয়বস্থাপনা

(<http://developer.android.com/training/basics/activity-lifecycle/index.html>)

প্লেটফর্মের স্টাইল এবং থিম বয়বহারে যেহেতু একজন ইউজার আপনার অ্যাপের মধ্যে দিয়ে, বাইরে বা অ্যাপে পরিণত আসার জন্য নিয়ন্ত্রণ করে, আপনার অ্যাপের Activity ইনস্ট্যান্স তার লাইফসাইকেলে বিভিন্ন পরিবর্তিত অবস্থানের মধ্যে থাকে। উদাহরণস্বরূপ যখন আপনার একটি টিভি টপ থমবালের মতো শুরু হয়, এটি সিস্টেমের সম্মুখভাগে চলে আসে এবং ইউজার ফোকাস গ্রহণ করে। এই প্রক্রিয়ার সময়, অ্যাপের ডায়ালগ সিস্টেম একটি টিভি টপ উপরে একটি টথারাবিহক লাইফসাইকেল পদ্ধতিতে কল করে যেখানে আপনি ইউজার ইন্টারফেস এবং অন্যান্য উপাদান সেট করেন। যদি ইউজার একটি একশন সম্পাদন করে থাকে যা অন্য একটি টিভি টপ শুরু করে বা অন্য অ্যাপে পরিবর্তন করে দেয়, সিস্টেম আরেকটি লাইফসাইকেল পদ্ধতির সেটকে কল করে যেহেতু এটি বয়াকগ্রাউন্ডে ডায়ালগে ধাবিত হয় (যেখানে একটি টিভি টপ আর দৃশ্যমান নয়, কিন্ত ইনস্ট্যান্স এবং এর ধরণ অবিকৃত থাকে)।

লাইফসাইকেল কলবয়াক পদ্ধতির মধ্যে থেকে আপনি ডিক্লয়ার করতে পারেন যে আপনার একটি টিভি টপে কমন আচরণ হবে যখন ইউজার একটি টিভি টপ থেকে বের হবে এবং পূর্ণ পরবেশ করবে। উদাহরণস্বরূপ, আপনি যদি একটা সিস্টেমের মিনিভিউতে প্লয়ার বানাতে থাকেন, যখন ইউজার আরেকটি অ্যাপে চলে যায় তখন আপনার ভিডিওতে একটা পজ দেয়া লাগতে পারে এবং নেটওয়ার্ক কানেকশন কে বন্ধ করে দেয়া লাগতে পারে। যখন ইউজার ফিরে আসে, আপনি নেটওয়ার্ককে পূর্ণ রেস্যোগ করতে পারেন এবং ইউজারকে ভিডিওটিতে যে অবস্থায় রেখে যাওয়া হয়েছিল সে অবস্থাতে থেকে শুরু করতে অনুরোধাদন করতে পারেন।

এই ক্লাস গুরুত্বপূর্ণ লাইফসাইকেল কলবয়াক মেথডকে বিবেচনা করে যা প্রতিটা Activity ইনস্ট্যান্স গ্রহণ করে এবং কীভাবে এদের বয়বহার করতে পারেন যাতে ইউজার যেভাবে চায় সেভাবে কাজ করতে পারে এবং সিস্টেমের রিসোর্সকে বয়বহার করেন না যখন আপনার একটি টিভি টপের জন্য এর পরে যোজন হয় না।

এই অধ্যায়ের অনুশীলন সমূহ

একটি ভিডিও শুরু করা

একটি ভিডিও লাইফসাইকেল এর মেমোরিয়াল বিষয় সম্পর্কে শিখুন, কীভাবে ইউজার আপনার অ্যাপ শুরু করবে, এবং কীভাবে মেমোরিয়াল একটি ভিডিও তৈরী করার বিষয়টি সম্পাদন করা যায়।

একটি ভিডিও পিজ্ঞ এবং রিজিউম

যখন আপনার একটি ভিডিওতে পজ দেয়া হবে (আংশিকভাবে বিহীন থাকে) এবং রিজিউম করা হবে তখন কী হবে এবং যখন এই অবস্থানের পরিবর্তন হবে তখন আপনার কী করা উচিত সে বিষয়ে শিখুন।

একটি ভিডিও স্টোপ এবং রিস্টার্ট

যখন ইউজার সম্পূর্ণভাবে আপনার একটি ভিডিও ত্যাগ করবে এবং এটাতে আবার ফিরে আসবে তখন কী হবে সে বিষয়ে শিখুন।

একটি ভিডিওর পুননির্মাণ

যখন আপনার একটি ভিডিও সম্পূর্ণভাবে ধবংস হবে তখন কী হবে এবং কীভাবে আপনি একটি ভিডিও স্টোপ পুননির্মাণ করতে পারবেন যখন এর পরে যাজন হবে সে বিষয়ে শিখুন।

একটি টিভি টিশুরুর করা

(<http://developer.android.com/training/basics/activity-lifecycle/starting.html>)

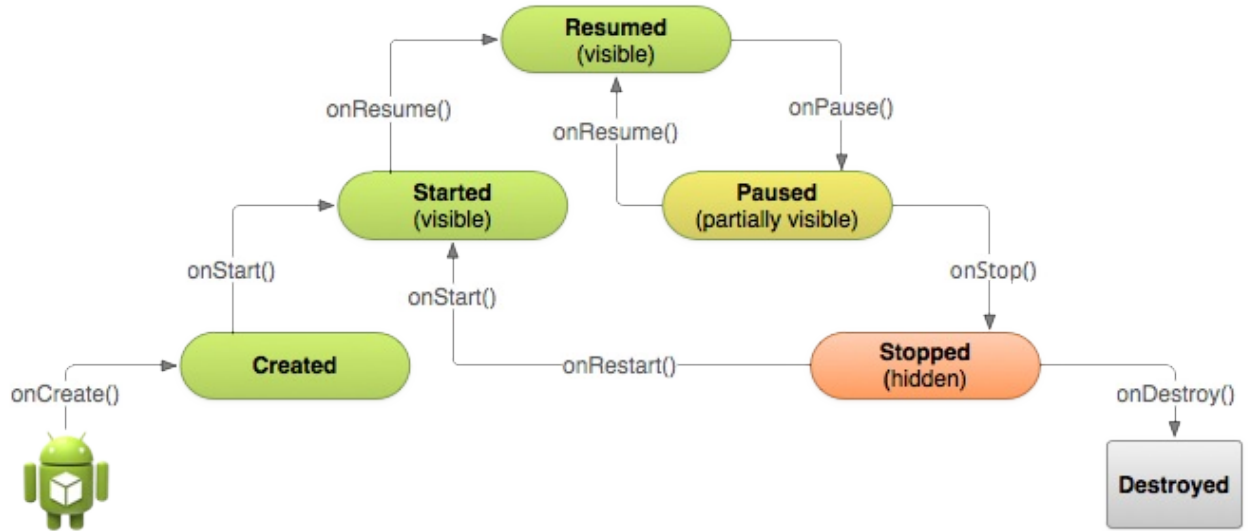
অন্য পেরাগরামি পয়ারাডাইম এর মতানয় যেখানে অ্যাপস একটি main() পদ্ধতি কতৃক তার যাত্রা শুরু করে, অ্যান্ডরয়েড সিস্টেম সুনির্দিষ্ট কলব্যাক পদ্ধতি আবাহনের মাধ্যমে একটি Activity ইনস্ট্যান্স এর মধ্যে কোড তৈরী করে যা এর লাইফসাইকেলের সুনির্দিষ্ট ধাপে যোগাযোগ করে। এখানে কলব্যাক পদ্ধতির একটা অনুকরম (সিকোয়েন্স) আছে যা একটি টিভি টি তৈরী করে এবং কলব্যাক পদ্ধতির একটি অনুকরম (সিকোয়েন্স) আছে যা একটি টিভি টির বিমোচন ঘটায়।

এই অনুশীলনী সবচেয়ে গুরুত্বপূর্ণর লাইফসাইকেল পদ্ধতির একটা সাধারণ ধারণা পর্দান করে এবং দেখায় যে কীভাবে প্রথম লাইফসাইকেল কলব্যাক পরিচালনা করতে হয় যা আপনার একটি টিভি টির জন্য একটি নতুন ইনস্ট্যান্স তৈরী করে

লাইফ সাইকেল কল বয়াক সম্বন্ধে ধারণা

একটি একটি ভিউটির জীবদ্দশাকালীন সময়ে, সিস্টেম লাইফ সাইকেল পদ্ধতির একটি মূল সটকে কর্মানুসারে কল করে যা একটি পিরামিডের ধাপের মতো। তাই একটি ভিউটি লাইফ সাইকেল এর প্রতিটা ধাপ পিরামিডের পৃথক ধাপ। যখন সিস্টেম একটি নতুন একটি ভিউটি ইনস্ট্যান্স তৈরী করে, প্রতিটা কল বয়াক পদ্ধতি একটি ভিউটি স্টটকে উপরের দিকে এক ধাপ এগিয়ে নেয়। পিরামিডের সবচেয়ে উপরের দিক হচ্ছে সেই পয়েন্ট যাতে একটি ভিউটি সম্মুখভাগে গরান করে এবং ইউজার এর সাথে যোগাযোগ করতে পারে।

যখন ইউজার একটি ভিউটি ত্যাগ করা শুরু করে, সিস্টেম আরেকটি পদ্ধতি কল করে যা একটি ভিউটিকে বিচ্ছিন্ন করার জন্য একটি ভিউটি স্টটকে পিরামিডের নীচের দিকে নামিয়ে আনে। কিছু কিছু ক্ষেত্রে, একটি ভিউটি শুধুমাত্র একটি অংশ নামিয়ে নিয়ে আসে এবং অপেক্ষা করে (যেমন, যখন ইউজার অন্য অ্যাপে চলে যায়), যে পয়েন্ট থেকে একটি ভিউটি আবার উপরে উঠে যেতে পারে (যদি ইউজার একটি ভিউটিতে ফিরে আসে) এবং যেখানে ইউজার এটারে থেঁতলায় সেখানে থেকেন শুরু করে।



ফিগার ১. একটি ভিউটি লাইফ সাইকেলের একটি সরলিকৃত ডায়াগ্রাম, একটি পিরামিড স্ট্রাকচারের মতো। এটা দেখায় কীভাবে একটি ভিউটি রিজিউম স্টট এর উপরের দিকে নিয়ে যেতে প্রতিটা কল বয়াক বহুত হয়, এখানে আরও একটি কল বয়াক পদ্ধতি আছে যা একটি ভিউটিকে নীচের ধাপে নামিয়ে যায়। একটি ভিউটি পজড স্টট (অবস্থান) এবং স্টপড স্টট থেকে রিজিউম স্টটে ফিরে আসতে পারে।

আপনার একটি ভিউটির জটিলতার উপর নির্ভর করে, সম্ভবত আপনার সকল লাইফ সাইকেল মেথড বাস্তবায়ন করার দরকার নেই। যাইহোক এটা গুরুত্বপূর্ণ যে আপনি প্রতিটা বিষয় বুঝতে পেরেছেন এবং বাস্তবায়ন করেছেন যা ইউজার যা আশা করছে আপনার অ্যাপে সম্ভাব্য কাজ করবে। আপনার একটি ভিউটি লাইফ সাইকেল পদ্ধতির যথাযথ বাস্তবায়ন আপনার অ্যাপকে বিভিন্ন উপায়ে সুন্দরভাবে কাজ করাটাকে নিশ্চিত করবে, যার মধ্যে রয়েছে:

- আপনার অ্যাপ বহুত্ব করার সময় যদি ইউজারের ক্যানফোন কল আসে বা অন্য কোন অ্যাপে চলে যায় তখন এটা একবার বন্ধ হয়ে যায় না।
- ইউজার এটা কেসিকর যভাবে বহুত্ব না করলে মূল যবান সিস্টেম

রিমেসাসর্ বয্য কের না

- যখন ইউজার অর্থাৎ পেইডে চলে যায় এবং পেইডে আবার ফিরে আসে সেসময় যে তর্ এটা ইউজারের অর্গগিতিকে হারিয়ে যেতে দেয় না
- সিস্করন যখন লয়ান্ডে স্কপ এবং পেটের ইট এর মধ্যে রেটেট করে তখন এটা ইউজারের অর্গগিতিকে হারিয়ে যেতে অথবা বন্ধ হতে দেয় না

যখন আপনি নিম্নোক্ত অনুশীলনী শিখবেন, সেখানে বেশ কিছু অবস্থা আছে যেখানে একটি টিভি ভিডিওটি বিভিন্ন স্টেটের (অবস্থানের) মধ্যে পরিবর্তন হয় যা ফিগার ১ এ দেখানো হয়েছে। কিন্ত এই স্টেটগুলোর মধ্যে শুধুমাত্র তিনটা অনড় থাকে। তাই, একটি টিভি একটি বিধর্ত সময়ের জন্য তিনটার মধ্যে একটিতে অবস্থান করতে পারে:

Resumed/রিজিউমড

এই স্টেট (অবস্থান), একটি টিভি সামনে থাকে এবং ইউজার এটার সাথে যোগাযোগ করতে পারে। (মাঝে এটাকে “রাইন” স্টেট হিসাবে উল্লেখ করা হয়)

Paused/পজড

এই স্টেট, একটি টিভি অন্য একটি দ্বারা আংশিকভাবে ঢাকা থাকে-অন্য একটি টিভি যেটা সামনে থাকে সেটা অর্ধস্বচ্ছ বা সম্পূর্ণরূপে স্করনে ঢেকে দেয় না। পজড একটি টিভি টি ইউজার ইনপুট গ্রহণ করেনা এবং কোন ক্র্যড সম্পাদন করেনা।

Stopped/স্টপড

এই স্টেট, একটি টিভি সম্পূর্ণরূপে ঢাকা থাকে এবং ইউজার এটা দেখতে পায় না; এটা পেছনভাগে থাকার বিষয়টা বিবেচনা করে। যখন একটি টিভি বন্ধ হয় তখন একটি টিভি ইনস্টেন্স এবং এর সকল স্টেট তথ্য যেমন মেমবারিভিরিয়েবলসকে ধরে রাখা হয় কিন্ত এটাকে কোন ক্র্যড সম্পাদন করতে পারেনা।

অন্য স্টেট (কিরিয়েট এবং স্টাটেরড) হচ্ছে অস্থায়ী এবং সিসেমটি অতিদ্রুত তাৎদর থেকে পরবর্তী স্টেটে চলে যায় পরবর্তী লাইফসাইকেল কলব্যাক মেথডকে কল করার মাধ্যমে। তাই সিসেমটি onCreate()কে কল করার পর, এটা দ্রুত onStart()কে কল করে, যা দ্রুত তার সাথে onResume()কে অনুসরণ করে।

এটা মৌলিক একটি টিভি লাইফসাইকেল এর জন্য. এখন আপনি কিছু সুনির্দিষ্ট লাইফসাইকেল আচরন সম্পর্কে শিখবেন।

আপনার অ্যাপের লঞ্চার একটি টিভি নিদির্ষ্ট করুন

যখন হোম স্ক্রিন থেকে ইউজার আপনার অ্যাপ আইকন বাছাই করে, তখন সিস্টেম আপনার অ্যাপের Activity এর জন্য onCreate() পদ্ধতিতে কল করে, যাকে আপনি “লঞ্চার” (অথবা প্রধান/“main”) একটি টিভি হিসাবে ঘোষণা করে। এটা সেই একটি টিভি যা আপনার অ্যাপের ইউজার ইন্টারফেসের প্রধান পর্বে পথ হিসাবে কাজ করে।

আপনি নিধরারন করতে পারেন যে অ্যান্ড্রয়েড মেনিফেস্ট ফাইলে কোন একটি টিভি টিট প্রধান একটি টিভি হিসাবে বর্নিত হবে, AndroidManifest.xml, যা আপনার পর্বেজক্ট ডিরেক্টরীর উৎসে আছে।

আপনার অ্যাপের জন্য প্রধান একটি টিভি অবশ্যই <intent-filter> সহ মেনিফেস্টে ডিক্লেয়ার করতে হবে যা MAIN একশন এবং LAUNCHER ক্যাটাগরি কে অন্তর্ভুক্ত করে। উদাহরণস্বরূপ:

```
<activity android:name=".MainActivity" android:label="@string/app_name">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
```

নোট: যখন আপনি অ্যান্ড্রয়েড এসডি কে টুলস দিয়ে একটি নতুন অ্যান্ড্রয়েড পর্বেজক্ট তৈরী করেন, Activity ক্লাস সহ ডিফল্ট পর্বেজক্ট ফাইলটি এই ফিল্টার দিয়ে মেনিফেস্টে ডিক্লেয়ার হয়।

যদি আপনার কোন একটা একটি টিভির জন্য হয় MAIN একশন বা LAUNCHER ক্যাটাগরি ডিক্লেয়ার করা না হয়ে থাকে, তখন হোম স্ক্রিনের অ্যাপের তালিকায় আপনার অ্যাপ আইকন দেখানো হবে না।

17.3 একটি নতুন ইনস্টেন্স তৈরী করুন

অধিকাংশ অ্যাপ কিছুর ভিত্তিতে একটি ভিউ অন্তর্ভুক্ত করে যা ইউজার কতটুকু বিভিন্ন একশন সম্পাদন করে। অনেক অ্যাপই হতে পারে একটি একটি ভিউ পর্ধান একটি ভিউ, যেটা তৈরী হয় যখন ইউজার আপনার অ্যাপ আইকনে ক্লিক করে অথবা একটি সর্বোত্তম একটি ভিউ যা আপনার অ্যাপ একটা ইউজার একশনের সপক্ষে এগিয়ে আসে, সিস্টেমটি Activity এর পরে টিউন ইনস্ট্যান্স onCreate() পদ্ধতিতে কল করার মাধ্যমে তৈরী করে থাকে।

আপনাকে অবশ্যই বেসিক অ্যাপ লিকেশন স্টাটর আপ লজিক সম্পাদন করতে onCreate() পদ্ধতি বাস্তুবায়ন করতে হবে। উদাহরণস্বরূপ, আপনার onCreate() এর বাস্তুবায়নে ইউজার ইন্টারফেসকে নিখরার করবে দেওয়া উচিত এবং সম্ভাব্য ক্লাস-স্কেপ ভেরিয়েবলসকে ইনস্ট্যান্সিয়েট করা।

উদাহরণস্বরূপ, নিম্নোক্ত onCreate() পদ্ধতির উদাহরণ দেখায় কিছুর কোড যা একটি ভিউ টির জন্ম দিচ্ছে মেইলিক সেটআপ সম্পাদন করে থাকে, যেমন, ইউজার ইন্টারফেস ডিকে ক্লয়ার করা (একটি এক্সএমএল লেআউট ফাইলে নিখরার), মেমবার ভেরিয়েবল নিখরার, এবং কিছুর ইউজার ইন্টারফেস কনফিগার করা।

```
TextView mTextView; // Member variable for text view in the layout

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    // Set the user interface layout for this Activity
    // The layout file is defined in the project res/layout/main_activity.xml file
    setContentView(R.layout.main_activity);

    // Initialize member TextView so we can manipulate it later
    mTextView = (TextView) findViewById(R.id.text_message);

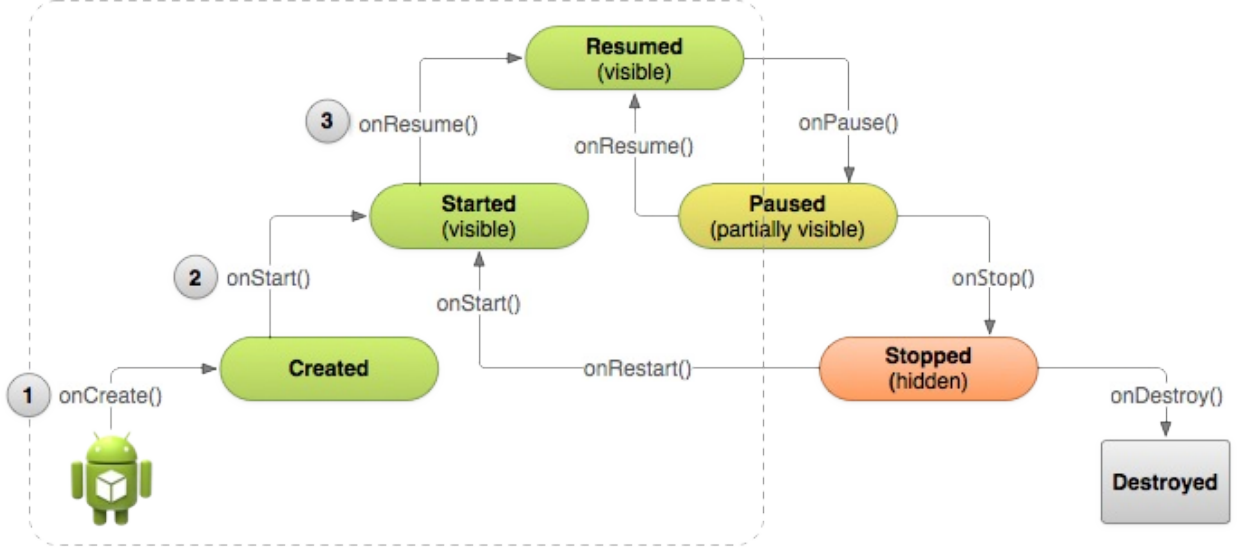
    // Make sure we're running on Honeycomb or higher to use ActionBar APIs
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
        // For the main activity, make sure the app icon in the action bar
        // does not behave as a button
        ActionBar actionBar = getActionBar();
        actionBar.setHomeButtonEnabled(false);
    }
}
```

সতর্কতা: নতুন এপিআই (APIs) থেকে পুরাতন সিস্টেমকে পরিহার করতে SDK INT ব্যবহার করে অ্যান্ড্রয়েড ২.০ বা এর চেয়ে উন্নত সংস্করণে এই উপায়ে কাজ করে। পুরাতন সংস্করণ রানটাইম এক্সপেশনকে এনকাউন্টার করে।

যখন onCreate() বাস্তুবায়ন শেষ করে, সিস্টেম onStarted() এবং onResume() পদ্ধতিতে কল দরুত কর্তমান সারেকল করে। আপনার একটি ভিউ কখনই ক্রিয়েটেড এবং স্টাটর ডেস্টেট থাকে না। কৌশলগতভাবে, একটি ভিউ ইউজারের কাছে দৃশ্যমান হয় যখন onStarted() কল করা হয়, কিন্ত onResume() দরুত অনুসরণ করে এবং একটি ভিউ টিরিজিউমে স্টেট থাকে যতক্ষণ পর্যন্ত না এটাকে পরিবর্তন করার জন্য কৈনিকিছুর না ঘটতে থাকে, যেমন যখন একটা ফোন কল রিসিভ করা হয়, ইউজার আরেকটি একটি ভিউ টিতে নৈভিগেট করে অথবা ডিভাইসিস্করন বন্ধ করা হয়।

অন্য অনুশীলনে যা অনুসরণ করে, আপনি দেখতে পাবেন যে স্টাটর আপ পদ্ধতি onStarted() এবং onResume() কীভাবে আপনার একটি ভিউ টির লাইফসাইকেলের জন্ম পরে জন্ম যখন পজড এবং স্টপডেস্টেট থেকে একটি ভিউ টি পুনরায় শুরু করতে ব্যবহৃত হয়।

নেট: onCreate() পদ্ধতিটি savedInstanceState নামে একটি প্যারামিটার
অনুভব করছে যা পরবর্তী অনুশীলনে আলোচনা করা হবে।



ফিগার ২. তিনটি প্রধান কলব্যাক এর উপরে দেয়া একটি ভিডিও
লাইফসাইকেল কাঠামোর আরেকটি চিত্র যাতে সিস্টেম কর্তৃক মানুসের
কলকরে যখন একটি ভিডিওর একটানতুন ইনসট্যান্স তৈরী করা হয়: onCreate(),
onStart(), এবং onResume()। যখন এই কলব্যাকের কর্ম শেষ হয়, একটি ভিডিওটির জিউম
সিস্টেম এ পেঁছায় যেখানে ইউজার একটি ভিডিওর সাথে যোগাযোগ করতে
পারে যতক্ষণ না তার অন্যান্য একটি ভিডিওতে চলে যায়।

একটি টিভিটির বিবেচনা

onCreate() যখন একটি টিভিটির প্রথম লাইফসাইকেল কলকলব্যাক হয়, onDestroy() টা তখন শেষ কলকলব্যাক হয়। সিস্টেমটি ফাইনালিসগনয়ালিহসাবে আপনার একটি টিভিটির উপর এই পদ্ধতি কল করে যা আপনার একটি টিভিট ইনসট্যান্সকে সমাপ্ত করার হুপে সিস্টেম মেমরিতে থেকে সরিয়ে ফেলে।

অধিকাংশ অ্যাপস এর এই পদ্ধতি বাস্তবায়ন করার দরকার নেই কারণ লোকাল ক্লাসের ফোরেনস একটি টিভিটির সাথে থিবেলোপ হয়ে যায় এবং onPause() এবং onStop() এর সময় আপনার একটি টিভিটির সবচেয়ে পরিষ্কার করা উচিত। যাহোক যদি আপনার একটি টিভিটি বয়াকগর্ডাউন ডেথের ডান্ডন্তভূর্ত করে থাকে যা আপনি onCreate() অথবা অন্য ল্যারানিরিসাসর এর সময় তৈরী করেছেন যা মেমরিতে কবের করে দিতে পারে (leak) যদি নাতাযথায় থাবে বন্ধ করা হয়, onDestroy() এর সময় আপনার এগুলো থেকে ধবংস করা উচিত।

```
@Override
public void onDestroy() {
    super.onDestroy(); // Always call the superclass

    // Stop method tracing that the activity started during onCreate()
    android.os.Debug.stopMethodTracing();
}
```

নোট: সিস্টেমটি ইতিমধ্যে onPause() এবং onStop() কে কল করার পর onDestroy() কে সব সময় কল করে একটি অব্যতীকর্ম ছাড়া: যখন আপনি onCreate() পদ্ধতির মধ্যে থেকে finish() কে কল করবেন। কিছু কিছু ক্ষেত্রে যখন আপনার একটি টিভিটি আরেকটি একটি টিভিটি লঞ্চ করতে অস্থায়ী সিস্টেমের গর্হণকারী হিসাবে কাজ পরিচালনা করে, আপনি একটি টিভিটি বিবেচনা (destroy) করতে onCreate() এর মধ্যে থেকে finish() কে কল করতে পারেন। এই ক্ষেত্রে সিস্টেমটি অন্য কোন লাইফসাইকেল পদ্ধতি গর্হণ না করে তা কখনো কখনো onDestroy() কে কল করে।

একটি একটি ভিউ টির পজ এবং রিজিউম করা

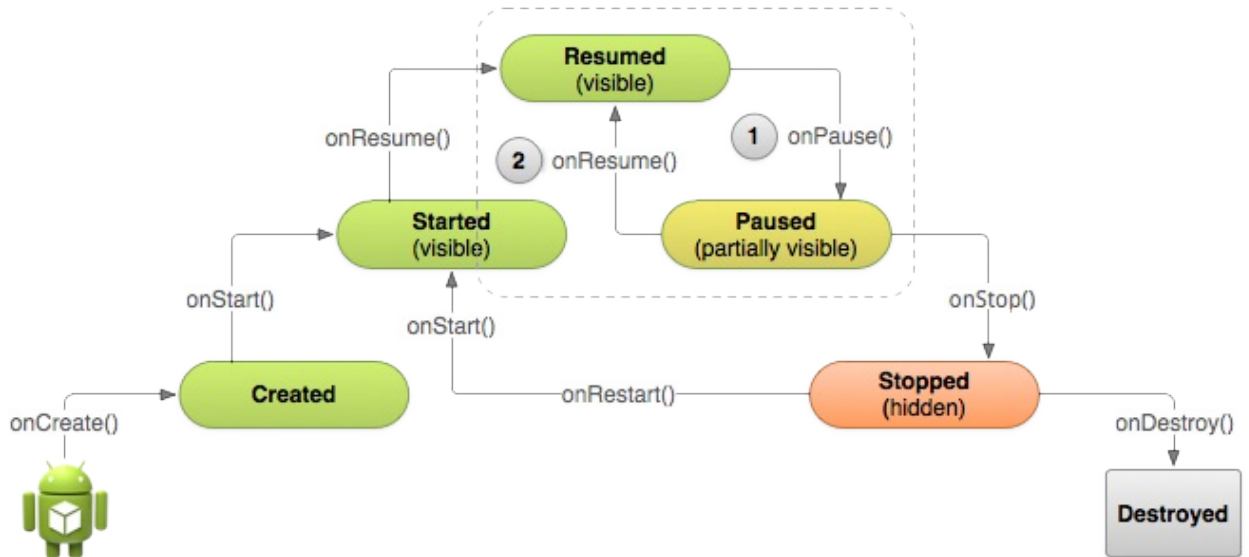
(<http://developer.android.com/training/basics/activity-lifecycle/pausing.html>)

অর্থাৎ পরস্পর ঋণাত্মক ব্যবহারকালীন সময়ে সাময়িকভাবে একটি ভিউ টি অনেক সময় অনবস্থিত থাকে যা উপদান করতে বাধ্য করে। প্ত হয় যা একটি ভিউ টির পজ এর কারণ। উদাহরণস্বরূপ, যখন একটি অধর্ষ সর্চ্চ একটি ভিউ টি ওপেন করা হয় (যেমন একটি ডায়ালগ বক্সে একটি স্টাইল) তখন পূর্ববর্তী একটি ভিউ টি পজ হয়ে যায়। যতক্ষণ এই একটি ভিউ টি আংশিকভাবে দৃশ্যমান থাকবে কিন্ত ফোকাসে থাকবে না, এটা পজ হিসাবে থাকবে।

যােহা একটি ভিউ টি যখন সম্পূর্ণভাবে বাতাপরাপ্ত হবে এবং দৃশ্যমান হবে না তখন এটা স্টপ (বন্ধ) হয়ে যাবে (এ বিষয়ে পরবর্তী অনুশীলনীতে আলোচনা করা হবে)।

যেহেতু আপনার একটি ভিউ টি পজ অবস্থায় পরবেশ করেছে, সিস্টেম আপনার Activity র উপর onPause() কল করে, যেটা আপনাকে একটি চলমান একশন/কমান্ড কে থামিয়ে দিতে দেয়, যা পজ অবস্থায় চালিয়ে যাওয়া উচিত নয় (যেমন ভিডিও) অথবা কোন তথ্য ধরে রাখা যা একজন ইউজারের আপনার অর্থাৎ ত্যাগ করার সময় স্থায়ীভাবে সেভ করে রাখা উচিত। যদি ইউজার পজ অবস্থান থেকে আপনার একটি ভিউ টিতে ফিরে আসে, সিস্টেমটি এটাকে পূরণায় শুরু করে এবং onResume() মেথডে কল করে।

নোট: যখন আপনার একটি ভিউ টি onPause() কল গ্রহণ করবে, এটা একটানি দের্শনা হবে যে একটি ভিউ টি কিছুর সময়ের জন্য পজ হবে এবং ইউজার আপনার একটি ভিউ টিতে আবার ফিরে আসতে পারে। যােহা, এটা সাধারণত প্রথম দিকনির্দেশনা যে ইউজার আপনার একটি ভিউ টি ত্যাগ করেছে।



ফিগার ১. যখন একটি অধর্ষ সর্চ্চ একটি ভিউ টি আপনার একটি ভিউ টিকে ডেকে দিবে, সিস্টেম onPause() কল করবে এবং একটি ভিউ টি পজ অবস্থায় অপেক্ষা করে (১)। যদি পজ থাকা অবস্থায় ইউজার একটি ভিউ টিতে ফিরে আসে, সিস্টেমটি onResume() কল করে (২)।

একটি টিভি টিতে পজ দেওয়া

যখন সিসে স্টমিটি আপনার একটি টিভি টির জন্য `onPause()` কল করে, এটা কৌশলগতভাবে বেবায় যে আপনার একটি টিভি টি এখনও আংশিকভাবে দৃশ্যমান, কিন্তু পরে যাই এরকম নিদর্শন না থাকে যে ইউজার একটি টিভি টি ত্যাগ করেছে এবং এটা খুব শিঘ্রই স্টপ অবস্থায় পরবে। নিম্নলিখিত কয়েকটি তথ্য আপনার `onPause()` কলবয়স্কাকবয়স্ক ব্যবহার করা উচিত:

- এনিমেশন বা অন্যান্য চলমান একশন/কমরন্ড বন্ধ করা যা সিসিপিইউ কেবয়স্ক ব্যবহার করেছে
- সেভ করা হয় নাই এমন পরিবর্তনগুলো সেভ করা, কিন্তু শুধুমাত্র তখনই যখন ইউজার একটি টিভি টি ত্যাগ করে তখন স্থায়ীভাবে এই চেঞ্জগুলো সেভ হতে আশা করে। (যেমন, একটি ড্রাফট ইমেইল)
- যখন ইউজারের জন্য এটার পরে যোজন নাই কিন্তু এটা পজ অবস্থায় থাকে বয়স্কটারীর আয়ুস্কালের উপর পরভাব ফেলে তখন বর্ডকাস্ট রিসভার, সেন্সর করতে জিপিএস (GPS) বা অন্যান্য কোন সিসে স্টম রিসেসার্স গুলোকে রিলিজ করে দিন।

উদাহরণস্বরূপ, যদি আপনার অ্যাপলিকেশন `Camera` ব্যবহার করে, এটাকে রিলিজ করবে `onPause()` পদ্ধতি একটি ভালে জায়গা

```
@Override
public void onPause() {
    super.onPause(); // Always call the superclass method first

    // Release the Camera because we don't need it when paused
    // and other activities might need to use it.
    if (mCamera != null) {
        mCamera.release();
        mCamera = null;
    }
}
```

সাধারণভাবে, ইউজার চেঞ্জগুলোকে স্থায়ী স্টারে স্টার করতে `onPause()` বয়স্ক ব্যবহার করা উচিত না (যেমন, একটি ফর্মের বয়স্ক তপতথ্য পরবেশ করানো)। আপনি শুধুমাত্র তখনই ইউজার চেঞ্জগুলোকে স্থায়ী স্টারে `onPause()` এর মধ্যে থেকে ধরে রাখতে পারেন যখন আপনি নিশ্চিত হবেন যে ইউজার আশা করবে যে এই চেঞ্জগুলো অটোমেটিকভাবে সেভ হবে (যেমন, যখন একটি ইমেইল ড্রাফট করা হয়ে থাকে)। যাহোক, `onPause()` এর সময় আপনার সিসিপিইউ-ইনেন্টিসভ কাজ করা পরিহার করার উচিত, যেমন একটি ডাটাবেজে লেখার কাজ, কারণ এটা পরবর্তী একটি টিভি টিতে দৃশ্যমান পরিবর্তনকে ধীরগতির করে দেয় (পরিবর্তে, `onStop()` এর সময় আপনার হেভিভ লোড শাটডাউন অপারেশন করা উচিত)।

যদি আপনার একটি টিভি টি আসলেই স্টপ/বন্ধ হয় তখন ইউজারের পরবর্তী লেক্ষয় দ্রুত চলে যাওয়ার বিষয়টা (a speedy transition) সমর্থন করতে `onPause()` পদ্ধতির মধ্যে অপেক্ষাকৃত সহজভাবে যে পরিমাণ অপারেশন গুলো সম্পাদিত হয়েছে তা আপনার ধরে রাখা উচিত।

নোট: যখন আপনার একটি টিভি টি পজ করা হয়, তখন `Activity` ইনস্টেন্সকে মেমরিতে রাখা হয় এবং এটাকে রিকল করা হয় যখন এটা আবার পুনরায় শুরু করা হয়। আপনার উপাদানগুলোকে পুনরায় শুরুর করার দরকার নেই যারিজিউম অবস্থায় নিয়ে যেতে যে কোন কলবয়স্কাকমেথড এর সময় তৈরী করা হয়।

আপনার একটি টিভি রিজিউম (পুনরায় শুরুর) করা

যখন ইউজার পজ অবস্থাতে থাকে আপনার একটি টিভি পুনরায় শুরুর করে (রিজিউম), তখন সিস্টেম `onResume()` পদ্ধতিতে কল করে।

এ বিষয়ে সতর্ক থাকার উচিত যে, আপনার একটি টিভি সম্মুখভাগে (ফোরগ্রাউন্ড) আসার ক্ষেত্রে সব সময়ই সময়ই সিস্টেমটি এই পদ্ধতিতে কল করে। যথা, কমেপোনেন্ট শুরুর করতে আপনার `onResume()` বাস্তবায়ন করার উচিত যা আপনি `onResume()` এর সময় মুক্ত করে দিয়েছিছেন এবং অন্য যে কোন কিছু শুরুর করার সময় যা একটি টিভি রিজিউম অবস্থায় পরবে শের পরিত্যাগেই যে টথাকে (যেমন, যখন শুধুমাত্র একটি টিভি টর ইউজার ফোকাস থাকে তখন এনিমেশন শুরুর করা এবং কমেপোনেন্ট শুরুর করা)।

নিম্নে নাক্ত `onResume()` এর উদাহরণটি উপরের `onResume()` এর উদাহরণের পরিতরূপ, এটাকয়ামেরাক শুরুর করায় যা একটি টিভি পজের সময় মুক্ত করা হয়েছিল।

```
@Override
public void onResume() {
    super.onResume(); // Always call the superclass method first

    // Get the Camera instance as the activity achieves full user focus
    if (mCamera == null) {
        initializeCamera(); // Local method to handle camera init
    }
}
```

একটি টিভি টস্‌টপ এবং রিস্‌ট্যাটার্‌ করার

(<http://developer.android.com/training/basics/activity-lifecycle/stopping.html>)

একটি টিভি ট লাইফসাইকেলের মধ্যে য় সাবিসর্কভাবে আপনার একটি টিভি টস্‌টপ এবং রিস্‌ট্যাটার্‌ হওয়া একটা গুরুত্বপূর্ণ পর্কর যা যা ইউজারের এ বিষয়ে উপলবির্ধ কিনির্শ্চত করবে যাতে আপনার অ্যাপ সবসময় জিবন্ত থাকবে এবং তােদর যে অর্গর্গিত তাধের রাখবে। নিচে কিছু চিত্র তুলে ধরা হলে যাতে আপনার একটি টিভি টস্‌টপ এবং রিস্‌ট্যাটার্‌ করবে:

- যখন ইউজার আপনার সাম্পর্কিতক অ্যাপ ওপেন করবে এবং আপনার অ্যাপ থেকে অনয় অ্যাপে চলে যাবে। আপনার অ্যাপের একটি টিভি ট যেটা বতর্মানেন সম্মুখভাগে (ফোরগর্রাউন্ড) আছে তা বন্ধ হলে। যদি হোম সিস্কর্ন লঞ্্চার আইকন অথবা সাম্পর্কিতক অ্যাপ উইন্ডো সথেকে ইউজার আপনার অ্যাপে ফিরে আসে, একটি টিভি ট পুনরায় শুরুর করবে।
- ইউজার আপনার অ্যাপে এমন একটা কাজ করে, যা আরেকটি নতুন একটি টিভি ট শুরুর করিয়ে দেয়। যখন দিবর্তীয় একটি টিভি ট তেরী হয় তখন চলিত একটি টিভি টটি বন্ধ (স্‌টপ) হয়ে যায়। এই সময় ইউজার যদি ***Back button** চাপে তাহলে পর্থম একটি টিভি টটি আবাব শুরুর হবে (রিস্‌ট্যাটের্ড)।
- যখন ইউজার আপনার অ্যাপ ব্যবহার করার সময় একটি টিফোন কলিরিস্ত করে।

Activity ক্লাস `onStop()` এবং `onRestart()` এই দুই ধরনের লাইফসাইকেল পদ্ধতি পর্দান করে, যা আপনার একটি টিভি ট কীভাবে স্‌টপ এবং রিস্‌ট্যাটার্‌কে ধারণ করবে তা আপনাকে সুনির্দির্ষ্‌তভাবে করতে দেয়। এটা পজ অবস্‌থানের মতো নয় যা একটি আর্শিক ইউজার ইন্টারফেস বাধাকে চিহ্নিত করে বর্স্‌টপ অবস্‌থা এর নিশ্চয়তাে দেয় যে ইউজার ইন্টারফেস আর দৃশয়মান হবে না এবং ইউজারের ফোকাস একটি টিভি ন একটি টিভি টতে থাকবে (অথবা একটা সম্পূর্ণ পর্গর্গিত ভিন্ন অ্যাপে)।

েনোট: কারণ সিস্‌টেমটি আপনার Activity ইনস্টেন্সকে সিস্‌টেম মেমিরিতে ধরে রাখে যখন এটা বন্ধ (স্‌টপড) হয়, এটা সম্ভবে আপনার `onStop()` এবং `onRestart()` অথবা এমনিক `onStart()` পদ্ধতিতে ও বাস্‌তবায়নের দরকার নেই। অধিকাংশ একটি টিভি ট যা অপেক্ষাকৃ্তভাবে সহজ, একটি টিভি টসুন দর্ভাবে স্‌টপ এবং রিস্‌ট্যাটার্‌ করতে পারবে এবং চলিত কাজকে পজিদিতে এবং সিস্‌টেম রিসোসর্থে থেকে স্যেগািবির্চছন্ন করতে আপনার শুধুমাত্র `onPause()` ব্যবহার করতে হতে পারে।

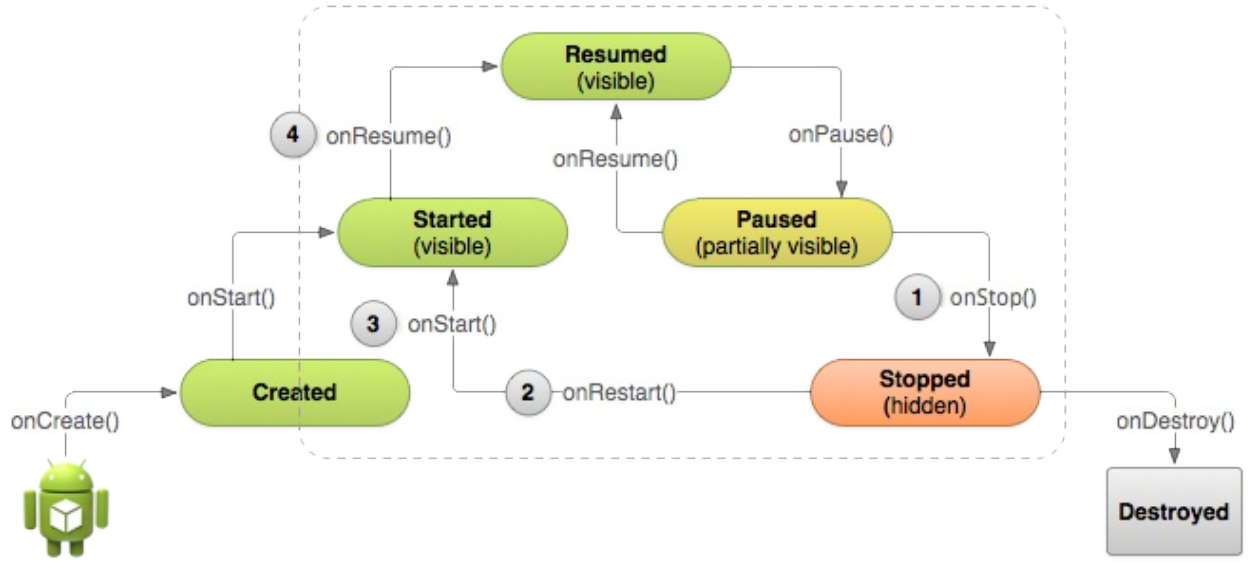


Figure 1. যখন ইউজার আপনার একটি টিভি টি ত্যাগ করে, সিস্টেমটি একটি টিভি টি বন্ধ (স্টপ) করতে `onStop()` কে কল করে (১)। যদি ইউজার একটি টিভি টি বন্ধ (স্টপড) থাকে অবস্থায় যখন আবার ফিরে আসে, সিস্টেমটি দরুত `onStart()` (৩) এবং `onResume()` (৪) কে অনুসরণ করে `onRestart()` কে কল করে (২)। উল্লেখ্য যে কোন বিষয়টা একটি টিভি টি কে স্টপ করার কারণ এটা কোন বিষয় নয়, সিস্টেম সবসময়ই `onStop()` কে কল করার পূর্বে `onPause()` কে কল করে।

আপনার একটি টিভি টবন্ধ (স্টপ) করুন

যখন আপনার একটি টিভি ট onStop() পদ্ধতি এ একটি কল রিসিভ করে, এটা আর দৃশ্যমান থাকে না এবং অধিকাংশ রিসোর্সসর্ ইয়ার পরে যোজন নাই যখন ইউজার এটা বন্ধ করার করে ছাড়া তামুকত (রিলিজ) করে দেয়া উচিত। যখন আপনার একটি টিভি টবন্ধ (স্টপ) হবে, সিসে স্টম ইনস্টেন্সকে ধবংস করে দিবে যদি এর সিসে স্টম মেমরির উন্নতি করার পরে যোজন হয়। চরম পর্যায়ে সিসে স্টমি ট একটি টিভি টর চুড়ানত onDestroy() কলব্যাংককে কলনা করেই আপনার অ্যাপ পরে সিসে সরাসির মেমের ফেলতে পারে, সুবরাং এটা গুরুতবং পূর্ণরূপে মেমরিরে কলিক করতে পারে এমন রিসোর্সসর্ গুলে লোকে রিলিজ করতে onStop() ব্যব্যহার করতে পারেন।

যদিও onPause() কে পদ্ধতি onStop() এর পূর্বে বন্ধ কল করা হয়, আরও ভালোভাবে কায়র সম্ভাদন করতে, আরও সিসি পইউ-ইনস্টেন্সিভ শাট-ডাউন অপারেশন করতে আপনার onStop() ব্যব্যহার করা উচিত, যেমন, একটি ডাটাবেজে তথ্য লেখার কাজ করতে।

উদাহরণস্বরূপ, এখনে onStop() এর একটি বাস্তবায়ন দেয়া আছে যা একটি ড্রাফট নোটের কনটেন্ট অনট্রেস্টারে জেসভ করে:

```
@Override
protected void onStop() {
    super.onStop(); // Always call the superclass method first

    // Save the note's current draft, because the activity is stopping
    // and we want to be sure the current note progress isn't lost.
    ContentValues values = new ContentValues();
    values.put(NotePad.Notes.COLUMN_NAME_NOTE, getCurrentNoteText());
    values.put(NotePad.Notes.COLUMN_NAME_TITLE, getCurrentNoteTitle());

    getContentResolver().update(
        mUri, // The URI for the note to update.
        values, // The map of column names and new values to apply to them.
        null, // No SELECT criteria are used.
        null // No WHERE columns are used.
    );
}
```

যখন আপনার একটি টিভি টবন্ধ (স্টপ) হবে, Activity অবজেক্টটি মেমরিতে রাখা হয় এবং যখন একটি টিভি টপূর্ণরায় শুরুর (রিজিউমড) হয় তখন এটাকে আবার রিকল করা হয়। কর্মে কর্মে মিরিজিউম অবস্থায় নিয়ে যেতে যেকোন কলব্যংক মেথড এর সময় তৈরী করা কমে পানেন্ট এর পূর্ণপর্বতরনের পরে যোজন নেই। সিসে স্টমি টেলেআউটের মধ্যে পর্যন্ত টা View এর জন্য চলতি অবস্থার টর্যংক গুলে লোকে ধরেও রাখে, সুতরাং ইউজার যদি EditText উইজিটের (widget) মধ্যে টেক্সট পর্বশ করায়, ওই কনটেন্ট যথাস্থানে রাখা থাকে তাই আপনার এটাকে সভকার এবং রিসে স্টার করার পরে যোজন নেই।

নোট: এমনকি যদি সিসে স্টমি ট আপনার একটি টিভি টকে ধবংস করে দেয় যখন এটা বন্ধ থাকে, এটা তখনও একটি Bundle এ (িক্তয্যালু পেয়ার এর একটি টিবন্ধদু) View অবজেক্টের (যেমন, একটি EditText এর মধ্যে টেক্সট) সেস্টটকে ধরে রাখে এবং তােদরেক রিসে স্টার করে যদি ইউজার আপনার একটি টিভি টর একই ইনস্টেন্স এ নেনিভিগেট করে ফিরে আসে (পরবর্তী অনুষাীলনীতে (next lesson) আপনার একটি টিভি ট ধবংস হওয়া বা পূর্ণনিমণের রেক্ষেত্রে অন্যান্য সিসে স্টট ডাটাবেসভ করতে Bundle ব্যব্যহার করার বিষয় সম্ভেকর্ বিসত্ারিত আলাচনা করা হবে)।

আপনার একটি টিভি টি স্টাটর্/রিস্টাটর্ করুন

যখন আপনার একটি টিভি টি স্টপ অবস্থাতে থাকে সম্মুখভাগে (ফোরগরাউন্ড) ফিরে আসে, এটা `onRestart()` এ একটাকলিরিসভকরে। সিস্টেমটি

`onStart()` পদ্ধতিতে কল করে যা আপনার একটি টিভি টির পর্তিবারদৃশ্যমান হওয়া কেসে তর্ই কল করে (যখন পর্তথমবারের মতে ারিস্টাটর্ করা বা তৈরী করা হয়)। যােহাক `onRestart()` পদ্ধতিতে কল শুরুমাতর্ তখনই কল করা হয় যখন একটি টিভি টি স্টপ অবস্থাতে থাকে আব্বার পূর্ণরায় কাজ শুরুর করে, সুতরাং এটা আপনিনিবশেষিরিস্টাটারেশন কাজে বয়বহার করেত পারবেন যা শুরুমাতর্ তখনই পর্তয়োজনীয় হতে পারে একটি টিভি টি টি যদি পূর্বের ইবনধ (স্টপড) হয়ে থাকে, কিন্ত খুব সনা হয়।

এটা কমই দেখা যায় যে একটি অয়্পের একটি টিভি টির অবস্থারিস্টার করেত `onRestart()` বয়বহার করার দরকার হয়, সুতরাং এখানে এই পদ্ধতির জনয় কান গাইডলাইন নেই যা অয়্পের সাধারণ পপুলেশনে পর্তযোগ করা হয়। যােহাক যেহেতু আপনার `onStop()` পদ্ধতির উচিত আপনার একটি টিভি টির সকল রিসেসার্সেরিস্কার করে ফেলা, যখন একটি টিভি টি টি রিস্টাটর্ করেত তখন আপনার এগুলােক রিস-ইনসেটনিসেয়ট করা দরকার। যখন আপনার একটি টিভি টি পর্তথমবারের মতে াৈতরী করেবন (যখন একটি টিভি টির কান অিস্ততব্ ছিল না) আপনার তখনও এটাকে ইনসেটনিসেয়ট করেত হবে। এই কারনে, আপনার `onStop()` পদ্ধতি এর পর্তিরূপ হিসাবে `onStart()` কলবয়াক পদ্ধতি সব্বাভাবিকভাবে বয়বহার করা উচিত, কারন সিস্টেম `onStart()` উভয়ই কল করে যখন এটা আপনার একটি টিভি টি তৈরী করে এবং যখন এটা বনধ (স্টপড) অবস্থাতে থাকে একটি টিভি টি কেরিস্টাটর্ করে।

উদাহরণসব্বরূপ, যেহেতু ইউজার এখানে ফিরে আসার আগে দীঘর্ একটা সময়ের জনয় আপনার অয়্প থেকে চলে যােছ, `onStart()` পদ্ধতি এটা পর্তমান করার একটা ভালো জায়গা যে পর্তয়োজনীয় সিস্টেম ফিচার সিকর্য আছে:

```
@Override
protected void onStart() {
    super.onStart(); // Always call the superclass method first

    // The activity is either being restarted or started for the first time
    // so this is where we should make sure that GPS is enabled
    LocationManager locationManager =
        (LocationManager) getSystemService(Context.LOCATION_SERVICE);
    boolean gpsEnabled = locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);

    if (!gpsEnabled) {
        // Create a dialog here that requests the user to enable GPS, and use an intent
        // with the android.provider.Settings.ACTION_LOCATION_SOURCE_SETTINGS action
        // to take the user to the Settings screen to enable GPS when they click "OK"
    }
}

@Override
protected void onRestart() {
    super.onRestart(); // Always call the superclass method first

    // Activity being restarted from stopped state
}
```

যখন সিস্টেমটি আপনার একটি টিভি টি/খব্বাস করে দেয়, এটা আপনার `Activity` জনয় `onDestroy()` পদ্ধতি কল করে। কারন সব্বাভাবিকভাবে আপনিনিবশেষিরিস্টাটারেশনকরে দি়েছেন, িঠক এই সময়ে আপনিনিবশেষিরিসভকরেছেন, অধিকাংশ অয়্পের এখানে খুব বেশী কিস্করার দরকার নেই। এই পদ্ধতিরিসেসার্সেরিস্কার করে ফেলার একটা শেষ সুযোগ যাে মরিরিরেকর দিকেনিয়ে যেতে পাওে, সুতরাং আপনার নিশ্চত হওয়া উচিত যে অতিরিক্ত থের্ড খব্বাস হয়েছ এবং মেথড

টের্‌সিএর মতো দীঘল মেয়াদী কর্মকাণ্ডও বন্ধ (স্টপড) হয়েছে।

একটি টিভি টপুনরায় তৈরী করা

(<http://developer.android.com/training/basics/activity-lifecycle/recreating.html>)

এমন অনেক অবস্থা আছে যেখানে আপনার একটি টিভি টপ সবার অভাবিক অস্থাপন কমরক্ষা নেওড় কারণে ধবংস হয়, যেমন ইউজার যখন **Back** button চাপে অথবা আপনার একটি টিভি টপ `finish()` কল করে এটা নিয়ে জে কেই ধবংস করে। সিস্টেমটি আপনার একটি টিভি টপকে ধবংস করতে পারে যদি এটা বর্তমানে বন্ধ (স্টপড) থাকে এবং দীর্ঘ সময় এর কোন ব্যবহার করা না হয় বা ফোরগর্ডাউনড একটি টিভি টপ আরও রিসেসাসর্দাবী করে যার ফলে সিস্টেম অবশ্যই মেমরি রিকভার করতে ব্যবহাকর্গর্ডাউনড পরেসসেক শাটডাউন করে দেয়।

ইউজারের **Back** চাপার কারণে অথবা একটি টিভি টপ নিয়ে জেই এটাকে শেষ করে দেওয়ার কারণে যখন আপনার একটি টিভি টপ ধবংস পরাপ্ত হয়, ওই Activity ইনস্টেন্স এর সিস্টেমের ধারণা চিরতরে চলে যায় কারণ এই কমরক্ষা নেওড় নিদের্শ করে যে একটি টিভি টপ আর কোন পরয়োজন নাই। যাহোক, সিস্টেমের সীমাবদ্ধতার কারণে (সবার অভাবিক কমরক্ষা নেওড় না হলে) যদি সিস্টেম একটি টিভি টপ ধবংস করে দেয়, তখনও আসল Activity ইনস্টেন্স চলে যাবে, সিস্টেম এটাকে স্মরণ করবে যে এটার অস্তিত্ব আছে যদি ইউজার নেভিগেট করে ফিরে আসে, সিস্টেমটি এক সেটেসভ করা ডাটাব্যবহার করে একটি টিভি টপের একটানতুন ইনস্টেন্স তৈরী করে যা একটি টিভি টপের অবস্থার বনর্নাকের যখন এটা ধবংস হয়েছিল। সেভ করা ডাটাকে সিস্টেম পূর্ববর্তের অবস্থা রিস্টার করতে ব্যবহার করে তাকে "instance state" বলে এবং এটা হচ্ছে Bundle অবজেক্ট সেটের হওয়া কিত্যলুপেয়ারস এর কালেকশন।

সতর্কতা: ইউজার স্ক্রিনেরেটেট করলে পর্তিবারই আপনার একটি টিভি টপ ধবংস এবং নতুন করে তৈরী হবে। যখন স্ক্রিন এর ওরিয়েন্টেশন বদলাবে তখন সিস্টেম ফোরগর্ডাউনড একটি টিভি টপ ধবংস করবে এবং নতুন করে তৈরী করবে কারণ স্ক্রিন কনফিগারেশন পরিবর্তন হয়েছে এবং আপনার একটি টিভি টপ বিকল্প রিসেসাসর্লেড করার দরকার হতে পারে (যেমন, লেআউটটি)।

বাইডিফল্ট, আপনার একটি টিভি টপ লেআউটের পর্তিতটা View অবজেক্ট সম্প্রকিত তথ্য সেভ করতে সিস্টেমটি Bundle ইনস্টেন্স অবস্থাব্যবহার করে (যেমন, টেক্সট ভ্যালু EditText অবজেক্ট পর্তেবশ করে)। সুতরাং, যদি আপনার একটি টিভি টপ ইনস্টেন্স ধবংস হয় বা নতুন করে তৈরী হয়, আপনার চাওয়া কানেকাড ছাড়াই লেআউটের অবস্থা এর পূর্ববর্তাস্থায় রিস্টার হয়। যাহোক আপনার একটি টিভি টপ সম্ভবত আরও অবস্থাগত তথ্য আছে যা আপনি রিস্টার করতে চান, যেমন মেমব্রারভেরিয়েবল যা একটি টিভি টপেই ইউজারের অর্গতিতে কনসরন করে।

নোট: অস্থানওড়য়েড সিস্টেমের জন্ম আপনার একটি টিভি টপ ভিউয়ের অবস্থার রিস্টার করতে, `android.id` এটির বিউট কতর্ক পর্তদত্পর্তিতটা ভিউয়ের অবশ্যই সবার্তনতর্ আইডি থাকাউচিত।

একটি টিভি টপ অবস্থাসম্প্রকের অতিরিক্ত ডাটাসেভ করতে, আপনাকে অবশ্যই `onSaveInstanceState()` কলব্যাক মেথডকে অর্গাহয় করতে হবে। যখন ইউজার আপনার একটি টিভি টপ ত্যাগ করে তখন সিস্টেম এটাকে কল করে এবং এটাকে Bundle অবজেক্ট পাস করে দেয় যা ইভেন্ট সেভ হয়ে যাবে যাতে আপনার একটি টিভি টপ অনাকিওখতভাবে ধবংস হয়ে যায়। যদি সিস্টেম অবশ্যই পরবর্তর্ীতে একটি টিভি টপ ইনস্টেন্স নতুন করে তৈরী করে, এটা একই Bundle অবজেক্ট `onRestoreInstanceState()` এবং `onCreate()` উভয় পদ্ধতির দিকে পাস করে দেয়।



ফিগার ২. যেহেতু সিসেম্‌স্টেম আপনার একটি ভিউটি স্টপ করতে শুরু করেছে, এটা onSaveInstanceState() কে কল করে (১) যাতে ইনস্ট্যান্স অবশ্যই নতুন করে তৈরী করতে হবে এমন ক্ষেত্রে আপন অতিরিক্ত স্টেট ডাটাকে সুনির্দিষ্ট করে প্যারেন্থাইসিসে সেভ করতে চান। যদি একটি ভিউটি ধ্বংস হয়ে থাকে এবং একই ইনস্ট্যান্স অবশ্যই নতুন করে তৈরী করতে হবে, সিসেম্‌স্টেম স্টেট ডাটাকে (১) এ নিধারন করা হয়েছে তাকে onCreate() পদ্ধতি (২) এবং onRestoreInstanceState() উভয় (৩) পদ্ধতির দিকে পাস করে।

আপনার একটি টিভি টেস্ট (অবস্থান) সেভ করা

যেহেতু সিস্টেম আপনার একটি টিভি টেস্ট উপকরণে শুরু হয়েছে, এটা `onSaveInstanceState()` কল করে (১) যাতে আপনার একটি টিভি টেস্ট একটি কী ভ্যালু কালেকশনের সাথে সিস্টেম তথ্যকে সেভ করতে পারে। এই পদ্ধতি ডিফল্ট বাস্‌তবায়ন একটি টিভি টেস্টের উইজট হায়ারারিকি সিস্টেম সম্পর্কিত তথ্য সেভ করে, যেমন `EditText` উইজট (widget) এর মধ্যে টেক্সট অথবা `ListView` এর স্ক্রলের অবস্থান।

আপনার একটি টিভি টেস্টের জন্য অতিরিক্ত সিস্টেম তথ্য সেভ করতে, আপনাকে অবশ্যই ডিফল্ট `onSaveInstanceState()` এবং `onRestoreInstanceState()` মেথড কী ভ্যালু পেয়ারস বাস্‌তবায়ন করতে হবে। উদাহরণস্বরূপ:

```
static final String STATE_SCORE = "playerScore"; static final String STATE_LEVEL = "playerLevel"; ...
```

```
@Override
public void onSaveInstanceState(Bundle savedInstanceState) {
    // Save the user's current game state
    savedInstanceState.putInt(STATE_SCORE, mCurrentScore);
    savedInstanceState.putInt(STATE_LEVEL, mCurrentLevel);

    // Always call the superclass so it can save the view hierarchy state
    super.onSaveInstanceState(savedInstanceState);
}
```

সতর্কতা: সবসময় `onSaveInstanceState()` এর সুপারক্লাস ইমপিপ্লমেন্টেশন কল করুন যাতে ডিফল্ট ইমপিপ্লমেন্টেশন উইজট হায়ারারিকি সিস্টেম সেভ করতে পারে।

আপনার একটি টিভি স্টেটকে (অবস্থান) রিস্টোর করুন

আপনার একটি টিভি যখন পূর্বে বর্ধিত হয়ে আবার নতুন করে তৈরী হয়, আপনি Bundle থেকে আপনার স্টেটকে উদ্ধার করতে পারেন যা সিস্টেম আপনার একটি টিভি তে পাস করে। onCreate() এবং onRestoreInstanceState() উভয় কলব্যাক মেথডই একই Bundle গর্হণ করে যা ইনস্ট্যান্স স্টেট তথ্য ধারণ করে।

কারন onCreate() পদ্ধতিতে কল করা হয় যখন সিস্টেম আপনার একটি টিভির একটা নতুন ইনস্ট্যান্স তৈরী করেছে অথবা পূর্বে বর্ধিত একটাকে নতুন করে তৈরী করেছে, এটা পড়তে শুরুর করার পূর্বেই আপনাকে অবশ্যই চেক করতে হবে যে স্টেট Bundle অর্হণ হয়েছে কিনা। যদি এটা অর্হণ হয়, তখন সিস্টেম একটি টিভির নতুন ইনস্ট্যান্স তৈরী করতে থাকে, পূর্বে বর্ধিত একটা যা ধর্মসহে ছাড়াই রিস্টোর না করে।

উদাহরণস্বরূপ, এখানে দেখানো হচ্ছে কীভাবে আপনি onCreate() এর মধ্যে স্টেট ডাটার রিস্টোর করতে পারেন:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState); // Always call the superclass first

    // Check whether we're recreating a previously destroyed instance
    if (savedInstanceState != null) {
        // Restore value of members from saved state
        mCurrentScore = savedInstanceState.getInt(STATE_SCORE);
        mCurrentLevel = savedInstanceState.getInt(STATE_LEVEL);
    } else {
        // Probably initialize members with default values for a new instance
    }
    ...
}
```

onCreate() এর সময়ে স্টেট রিস্টোর না করে আপনি onRestoreInstanceState() কে বাস্তবায়ন করতে হবে নিতে পারেন, যা সিস্টেম onStart() পদ্ধতির পড়ে কল করে। সিস্টেম শুধুমাত্র onRestoreInstanceState() কল করে যদি সেখানে রিস্টোর করতে একটি সেভ করা স্টেট থাকে, সূতরাং আপনাকে আর পরীক্ষা করতে হবে না যে Bundle গর্হণ হয়েছে:

```
public void onRestoreInstanceState(Bundle savedInstanceState) {
    // Always call the superclass so it can restore the view hierarchy
    super.onRestoreInstanceState(savedInstanceState);

    // Restore state members from saved instance
    mCurrentScore = savedInstanceState.getInt(STATE_SCORE);
    mCurrentLevel = savedInstanceState.getInt(STATE_LEVEL);
}
```

সতর্কতা: সবসময়ই onRestoreInstanceState() এর সুপারক্লাস ইমপিপ্লমেন্টেশন কল করুন যাতে ডিফল্ট ইমপিপ্লমেন্টেশন ভিত্তি হারানোর ঝুঁকি রিস্টোর করতে পারে।

রান টাইমের সময় ইভেন্ট রিস্টোর্ করার জন্য আপনার একটি টিভি নতুন করে শুরুর করা (যেমন, যখন স্ক্রিন রোট করে) সম্ভবিকৃত আরও তথ্য জানতে Handling Runtime Changes (<http://developer.android.com/guide/topics/resources/runtime-changes.html>) পড়ুন।

ফর্গেমেন্ট সহ ডাইনামিক ইউজার ইন্টারফেস তৈরী করা

(<http://developer.android.com/training/basics/fragments/index.html>)

অ্যান্ড্রয়েডে একটি ডাইনামিক এবং মাল্টি-পেন ইউজার ইন্টারফেস তৈরী করতে, আপনার ইউজার ইন্টারফেসকে ম্পানেন্ট এবং একটি ভিউ আচরণকে মডিউল এসয় ক্ত করা দরকার যাতে আপনি আপনার একটি ভিউর ভিতরে বা বাইরে অদলবদল করতে পারেন। আপনি এই মডিউল Fragment ক্লাস দিয়ে তৈরী করতে পারবেন, যা যেকোনভাবেই হোক একটি নেস্টেড একটি ভিউর মতো আচরণ করে যাতে আপনি এর নিজস্ব লেআউট নিধারন করতে পারেন এবং এর নিজস্ব লাইফসাইকেল ব্যবস্থাপনা করতে পারেন।

যখন একটি ফর্গেমেন্ট এর নিজস্ব লেআউট সুনির্দিষ্ট করে, বিভিন্ন নিস্কর্ন সাইজের জন্য আপনার লেআউট কনফিগারেশন পরিবর্তন করতে একটি একটি ভিউর মধ্যে অ্যান্ড্রয়েড ফর্গেমেন্টের সাথে বিভিন্ন কমিউনিকেশনের মধ্যে কনফিগার হতে পারে (একটি ছোট নিস্কর্ন এক সাথে একটি ফর্গেমেন্ট দেখাতে পারে, কিন্ত একটি বড় নিস্কর্ন দুই বা ততোধিক ফর্গেমেন্ট দেখাতে পারে)।

এই ক্লাস আপনাকে দেখাবে কীভাবে ফর্গেমেন্ট সহ একটি ডাইনামিক ইউজার এক্সপেরিয়েন্স তৈরী করতে পারে এবং বিভিন্ন নিস্কর্ন সাইজ সহ ডিভাইসের জন্য আপনার অ্যাপের ইউজার এক্সপেরিয়েন্সকে অপটিমাইজ করে, যা সারাক্ষণ অ্যান্ড্রয়েড ১.৬ এর মতো পুরাতন ভাসরুনের রান করা ডিভাইসকে সমর্থন করে।

এই অধ্যায়ের অনুশীলন সমূহ

একটি ফরমাগে মেন্টে তৈরী করা

শিক্ষুন কীভাবে একটি ফরমাগে মেন্টে তৈরী করতে হয় এবং এটার কলব্যাক মেথডের মধ্যে থেকে কেমৌলিক আচরণ বাস্তবায়ন করতে হয়।

একটি নমনীয় (ফ্লিক্সিবল) ইউজার ইন্টারফেস তৈরী করা

শিক্ষুন কীভাবে লআউটসহ আপনার অ্যাপস তৈরী করবেন যা বিভিন্ন সিস্করনের জন্য বিভিন্ন ফরমাগে মেন্ট কনফিগারেশন প্রদান করে।

অন্যায় ফরমাগে মেন্টের সাথে যোগাযোগ স্থাপন

শিক্ষুন কীভাবে একটি ফরমাগে মেন্ট থেকে একটি ভিউতে বা অন্য ফরমাগে মেন্ট কমিউনিকেশন/যোগাযোগের উপায় সেট করতে হয়।

একটি ফর্গে মেন্ট তৈরী করা

(<http://developer.android.com/training/basics/fragments/creating.html>)

আপনি একটি একটি ভিউ ট্রান্সফর্মার সেকশন হিসাবে একটি ফর্গে মেন্ট চিন্তা করতে পারেন, এর নিজস্ব লাইফসাইকেল আছে, এটাই নিজস্ব ইনপুট ইভেন্ট গ্রহণ করে এবং একটি ভিউ যখন রান করে তখন আপনি এটাকে যাজন বারিয়ে যাজন করতে পারেন (কতকংশে “সব একটি ভিউ ট্রান্সফর্মার” মতো যা আপনি বিভিন্ন একটি ভিউ ট্রান্সফর্মার বহুরকর করতে পারেন)। এই অনুশীলনী আপনাকে শেখাবে কীভাবে Support Library ব্যবহার করে Fragment টিকে বিন্যস্ত করা যায় যাতে আপনার অ্যাপ অ্যান্ড্রয়েড ১.৬ এর মতো পুরাতন ভার্শন ডিভাইসে ব্যবহার করার মতো উপযুক্ত রাখা যায়।

নোট: আপনি যদি চান যে আপনার অ্যাপের জন্য এপিআই লেভেল কমপক্ষে ১১ বা এর চেয়ে বেশী হবে, আপনার সাপোর্ট লাইব্রেরী ব্যবহার করার দরকার নেই এবং পরিবর্তে আপনি ফর্গে মেন্টের বিল্ট ইন Fragment ক্লাস এবং এসম্পটিক রিউ এপিআই ব্যবহার করতে পারেন। শুধু সতর্ক থাকবেন যে এই অনুশীলনী সাপোর্ট লাইব্রেরী থেকে এপিআই ব্যবহার করাটিকে ফ্যাকাস করে, যা প্লাটফর্মের অন্তর্ভুক্ত ভার্শনের চেয়ে একটি স্পেসিফিক প্যাকেজিং সগেনচার এবং মাঝে মাঝে সামান্য ভিন্ন এপিআই নাম ব্যবহার করে।

এই অনুশীলনী শুরুর করার পূর্বে, সাপোর্ট লাইব্রেরী ব্যবহার করার জন্য অবশ্যই আপনার অ্যান্ড্রয়েড পর্জেক্টে সেটআপ করতে হবে। আপনি যদি ইতিপূর্বে সাপোর্ট লাইব্রেরী ব্যবহার করেন না থাকেন, Support Library Setup (<http://developer.android.com/tools/support-library/setup.html>) ডকুমেন্ট অনুসরণ করে v4 লাইব্রেরী করতে আপনার পর্জেক্টে সেটআপ করুন। আবার v7 appcompat লাইব্রেরী ব্যবহার করার পরিবর্তে আপনি আপনার একটি ভিউ টিতে action bar (<http://developer.android.com/guide/topics/ui/actionbar.html>) অন্তর্ভুক্ত করতে পারেন, যা অ্যান্ড্রয়েড ২.১ (এপিআই লেভেল ৭) এর উপযুক্ত এবং Fragment এপিআইও অন্তর্ভুক্ত করে।

একটি ফর্াগমেন্ট ক্লাস তৈরী করুন

ফর্াগমেন্ট তৈরী করতে Fragment ক্লাসটি বর্ধিত করুন, এরপর আপনার অর্াপ লজিকে কিলাইফসাইকেল মেথড ওভাররাইড করুন, িঠক একই রকম ভাবে যেমন একটি Activity ক্লাসের সাথে থাকে বন।

যখন Fragment তৈরী করা হয় তখন একটা িভিন্তা তাকে তাহলে লেআউট িনধর্ারন করতে অবশ্যই আপনাকে onCreateView() কলবর্াক বর্ভহার করতে হবে। মূলত, এটা একমাত্র কলবর্াক একটা ফর্াগমেন্ট রানি অবস্থায় পাওয়ার জন্য যেটা আপনার পর্য়েয়াজন। উদাহরনসবর্রূপ, এখানে একটি সহজ ফর্াগমেন্ট আছে যা এর িনজসবর্লেআউটকে িনধর্ারন করে:

```
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.ViewGroup;

public class ArticleFragment extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.article_view, container, false);
    }
}
```

িঠক একটি একটি িভিটির মতো, একটি ফর্াগমেন্টর অনযর্ানযর্ লাইফসাইকেল কলবর্াক বাস্তবায়ন করা উচিত যা আপনাকে এর স্টট এর বর্বস্খাপনা করতে িদেবে যেহতু এটা একটি িভিটি থেকে সর্য়েয়াজিত বা িবর্য়োজিত হবে এবং যেহতু একটি িভিটি এর লাইফসাইকেল স্টটের মধ্যে পর্িবতর্ন হবে। উদাহরনসবর্রূপ, যখন একটি িভিটির onPause() মেথডকে কল করা হবে, একটি িভিটির যেকোন ফর্াগমেন্টও onPause() এ একটা কল িরিসভ করে।

ফর্াগমেন্ট লাইফসাইকেল এবং কলবর্াক মেথডের জন্য আরও তথ্য Fragments ডেভেলপ গাইডে আছে।

XML ব্যবহার করে একটি টিভিটেত ফর্য়গমেন্ট স্যেজন করুন

যখন ফর্য়গমেন্ট পূর্ণবর্য়ব্যহারে যোগ্য, মডিউলার ইউজার ইন্টারফেস
কম্পোনেন্ট, একটি `Fragment` ক্লাসের পরিতটাইনসেটনস অবশ্যই একটি
পর্্যারেন্ট `FragmentActivity` এর সাথে সম্পৃক্ত থাকতে হবে। আপনার একটি টিভিটে
লেআউট এক্সএমএল ফাইলের মধ্যে পরিতটা ফর্য়গমেন্ট নিখরারন করে
আপনি এই সম্পৃক্ততা অজরন করতে পারেন।

েনাট: এপিআই লেভেল ১১ এর চেয়ে পুরাতন সিসে স্টিমভাসরেন ফর্য়গমেন্ট
কে ধারন করতে সাপোর্টের রাইবেররীতে পরদেয় `FragmentActivity` একটা বিশেষ
একটি টিভিটে। যদি নুনযতম সিসে স্টিমভাসরেন এপিআই লেভেল ১১ বা এর চেয়ে
বেশী হয় যাক আপনি সাপোর্টের করছেন, তখন আপনি নিয়মিত `Activity` ব্যবহার
করতে পারেন।

এখানে লেআউট ফাইলের একটা উদাহরন দেয়া হেলা যা একটি একটি টিভিটে
দুটি ফর্য়গমেন্ট যুক্ত করে যখন মডিউলাইসিস্করন “লাজর”(বড়)
হিসাবে বিবেচয় হয় (ডিউরেক্টিরেনম এ large ক্যালিফায়ার দবারা
নিখরারিত)

res/layout-large/news_articles.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <fragment android:name="com.example.android.fragments.HeadlinesFragment"
        android:id="@+id/headlines_fragment"
        android:layout_weight="1"
        android:layout_width="0dp"
        android:layout_height="match_parent" />

    <fragment android:name="com.example.android.fragments.ArticleFragment"
        android:id="@+id/article_fragment"
        android:layout_weight="2"
        android:layout_width="0dp"
        android:layout_height="match_parent" />

</LinearLayout>
```

টিপস: ভিন্ভিন্ভন সিস্করনের জনয় লেআউট তৈরী করা সম্প্রিকর্ত আরও
তথ্য জানতে ভিভিন্ভন খরেনর সিস্করন সাইজ সাপোর্টের করা অথয়্যটি
পড়ুন।

এরপর আপনার একটি টিভিটেতে লেআউটটি পর্যোগ করুন:

```
import android.os.Bundle;
import android.support.v4.app.FragmentActivity;

public class MainActivity extends FragmentActivity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.news_articles);
    }
}
```

যদি আপনি v7 appcompat library ব্যবহার করে থাকেন, আপনার একটি টিভিটর
পরিবর্তে `ActionBarActivity` ব্যবহার করা উচিত, যা `FragmentActivity` এর সাবক্লাস
(আরও জানতে অফরহম ংযব অপারডহইখ পড়ুন)।

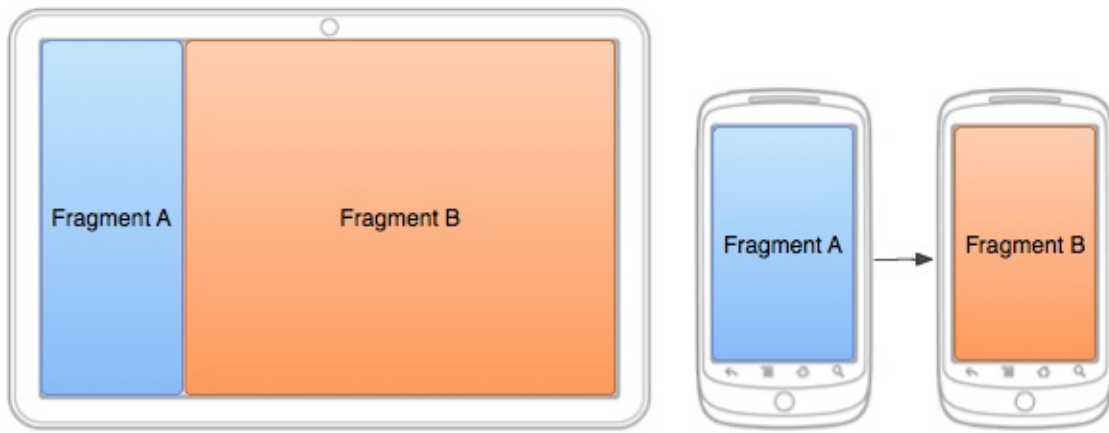
নেট: লেআউট এক্সএমএল ফাইলে ফর্গে মেন্টিনি ধরার নকরার মাধ্যমে
যখন আপনি একটি টিভি টেলেআউটে ফর্গে মেন্টিনি সাজান করবেন, রানটাইমে
এই ফর্গে মেন্টিনি কে বিযুক্ত করতে পারবেন না। আপনি যদি ইউজার
ইন্টারেকশনের সময় আপনার ভিত্তিতে এবং বাইরের অদলবদল করার
পরিবর্তন পনাকের থাকেন, আপনাকে অবশ্যই একটি টিভি টেলেআউটে ফর্গে মেন্টিনি
যুক্ত করতে হবে যখন একটি টিভি টেলেআউট প্রথম শুরু করে, পরবর্তী অনুশীলনীতে
যেভাবে দেখানো হয়েছে সেভাবে।

একটি নমনীয় (ফ্লিক্সিবল) ইউজার ইন্টারফেস তৈরী করুন

(<http://developer.android.com/training/basics/fragments/fragment-ui.html>)

যখন আপনার অ্যাপলিকেশনকে সিস্করন সাইজে বয়্যাপকিবস্ তৃতিতে সোপাটর্ করতে তৈরী করা হবে, বিদয়মান সিস্করন সাইজের উপর ভিত্তি করে ইউজার এক্সপেরিয়েন্স অপটিমাইজ করতে আপনিনি বিভিন্ন লেআউট কনফিগারেশনে আপনার ফর্াগমেন্টকে পূর্ণবর্য়ব্যহার করতে পারেন।

উদাহরণস্বরূপ, একটি হ্যান্ডসেটি ডিভাইসে একটি সিঙ্গেল-পেন ইউজার ইন্টারফেসে সরজনয় একবারে শুধু একটা ফর্াগমেন্ট পর্দশর্ন করাটা উপযুক্ত হবে। একইভাবে আপনি একটি ট্যাবলেটে যেতে আছে ইউজারকে আরও তথ্য পর্দশর্ন করার একটি পর্শস্ সিস্করন তাতে ফর্াগমেন্ট গুলেকে পাশাপাশি রাখতে চাইতে পারেন



িফগার ১. দুইটা ফর্াগমেন্ট ভিন্ন ভিন্ন সিস্করন সাইজে একই একটি ভিত্তির জনয় ভিন্ন ভিন্ন কনফিগারেশন পর্দশর্ন করেছে। বড় সিস্করনে দুইটা ফর্াগমেন্টই পাশাপাশি ফিট করেছে কিন্ত একটি হ্যান্ডসেটি ডিভাইসে একবারে শুধু একটা পর্াগমেন্ট ফিট করে সুতরাং ইউজার যখন নেভিগেট করে তখন একটা ফর্াগমেন্টের বদলে আরেকটি ফর্াগমেন্ট আসে।

FragmentManager ক্লাসটি মেথড সরবরাহ করে যা একটি ডাইনামিক এক্সপেরিয়েন্স সৈতরী করার জনয় একটি একটি ভিত্তিতে ফর্াগমেন্ট সঞ্যাজন, বিযেযাজন এবং পর্তিস্ থাপন করতে অনুমোদন করে।

রানটাইমে একটি টিভিভিটেত ফর্গামেন্ট স্যেজেন (এস্‌ড) করুন

যেভাবে <Fragment> এলিমেন্ট সহ পূর্ববর্তরী অনুশীলনীতে (previous lesson) দেখানো হয়েছে, লেআউটের মধ্যে একটি টিভিভিটের জন্য ফর্গামেন্ট মিনধরার মন করার বদলে আপনি একটি টিভিভিটের রানটাইমে একটি টিভিভিটেত একটি ফর্গামেন্ট স্যেজেন করে দিতে পারেন। এটা পরে যোজনীয় যদি আপনি একটি টিভিভিট অিস্তব্ শীল থাকাকালীন সময়ে ফর্গামেন্ট পরিবর্তন করার পরিকল্পনা করে থাকেন।

একটি ফর্গামেন্ট স্যেজেন বা বিয়েজন করার মতো বিভিন্ন ময় কয়র সম্পাদন করতে, একটি FragmentTransaction তৈরী করতে আপনাকে অবশ্যই FragmentManager ব্যবহার করতে হবে, যা স্যেজেন, বিয়েজন, পরিতস্থাপন এবং অন্যরানয় ফর্গামেন্ট বিভিন্ন ময় করতে এপিআই সরবরাহ করে থাকে।

এদি আপনার একটি টিভিভিট ফর্গামেন্ট কে বিয়েজিত বা পরিতস্থাপিত হতে অনুমোদন করে, আপনার উচিত একটি টিভিভিটের onCreate() মেথডের সময় একটি টিভিভিটেত পরাথমিক ফর্গামেন্ট স্যেজেন করা।

ফর্গামেন্ট মিনয়ে কাজ করার সময় একটি গুরুতব্ পূর্ণ রনিনয়ম-বিশেষত এ সকল সকল যার রানটাইমে স্যেজেন করা হয়েছে-এ মিনয়মিট হেচুছে য ফর্গামেন্টের অবশ্যই লেআউটের মধ্যে একটা কনটেইনার View থাকতে হবে যার মধ্যে ফর্গামেন্টের লেআউট অবস্থান করে।

মিনে মনাক্ত লেআউট পূর্ববর্তরী অনুশীলনীতে দেখানো লেআউটের বিকল্প যা একবারের শুধু একটা ফর্গামেন্ট পরদশরন করে। একটা ফর্গামেন্ট কে আর একটা ফর্গামেন্ট পরিতস্থাপন করতে একটি টিভিভিটের লেআউট একটা থালি FrameLayout অন্তর্ভুক্ত করে যা ফর্গামেন্ট কনটেইনার হিসাবে কাজ করে।

দেখা যায় যে, পূর্ববর্তরী অনুশীলনীর লেআউট ফাইলের মধ্যে ফাইলেনমিট একই রকম, কিন্তু লেআউট ডিরেক্টরীতে যধ মবকেয়ালি ফয়ারেনেই, তাই ডিভাইস সিস্করন বড় হওয়ার চেয়ে যখন ছোট হয় তখন এই লেআউটটি বব্যবহৃত হয় কারন সিস্করন একই সাথে উভয় ফর্গামেন্টের সাথে ফিট করে না।

res/layout/news_articles.xml:

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/fragment_container"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

আপনার একটি টিভিভিটের মধ্যে সাপোর্টার লাইব্ররী এপিআই ব্যবহার করে একটি FragmentManager পেতে getSupportFragmentManager() কল করুন। তারপর FragmentTransaction তৈরী করতে beginTransaction() কল করুন এবং একটি ফর্গামেন্ট স্যেজেন করতে add() কল করুন।

আপনি একই FragmentTransaction ব্যবহার করে একটি টিভিভিটের জন্য মাল্টিপল ফর্গামেন্টের রানজেকশন সম্পাদন করতে পারেন। যখন আপনি এই পরিবর্তন করতে পরস্তুত হবেন, আপনাকে অবশ্যই commit() কল করতে হবে।

উদাহরনসব্ রূপ, পূর্ববর্তরী লেআউট কীভাবে একটি ফর্গামেন্ট স্যেজেন করা যায় তা দেখা হলে:

```

import android.os.Bundle;
import android.support.v4.app.FragmentActivity;

public class MainActivity extends FragmentActivity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.news_articles);

        // Check that the activity is using the layout version with
        // the fragment_container FrameLayout
        if (findViewById(R.id.fragment_container) != null) {

            // However, if we're being restored from a previous state,
            // then we don't need to do anything and should return or else
            // we could end up with overlapping fragments.
            if (savedInstanceState != null) {
                return;
            }

            // Create a new Fragment to be placed in the activity layout
            HeadlinesFragment firstFragment = new HeadlinesFragment();

            // In case this activity was started with special instructions from an
            // Intent, pass the Intent's extras to the fragment as arguments
            firstFragment.setArguments(getIntent().getExtras());

            // Add the fragment to the 'fragment_container' FrameLayout
            getSupportFragmentManager().beginTransaction()
                .add(R.id.fragment_container, firstFragment).commit();
        }
    }
}

```

কারণ ফর্গে মেন্টরান টাইমে FrameLayout কনটেইনারের সংযুক্ত হয়ে ছ-
একটি <fragment> এলিমেন্ট দিয়ে একটি ভিউ টির লেআউটে এটানিখর্ধারণ করার
পরিবর্তে - একটি ভিউ টি ফর্গে মেন্ট টিকে বিবেচ্যিজিত করতে পারে এবং
ভিন্ন একটার সাথে এটাকে পরর্তিস্থাপন করতে পারে।

একটি ফ্রাগমেন্টের সাথে আরেকটি ফ্রাগমেন্টের প্রতিস্থাপন (রিপ্লেস)

একটি ফ্রাগমেন্ট প্রতিস্থাপনের পরিকর্যাসংযোজন পরিকর্যার মতই, কিন্ত `add()` এর পরিবর্তে `replace()` মেথড পর্যোজন হয়।

মেন রাখবেন যে যখন আপনি ফ্রাগমেন্ট ট্রানজেকশন সম্পাদন করবেন, যেমন ক্যান্টারি বিয়োজন বা প্রতিস্থাপন, ইউজারকে পরিবর্তনটিকে "undo" করতে নেভিগেট করে পেছনে নিয়ে আসাটা অনুমোদন করতে এটা বরাবরই যথায়থ। ফ্রাগমেন্ট ট্রানজেকশনের মাধ্যমে ইউজারকে পেছন দিকে নেভিগেট করা অনুমোদন করতে, `FragmentManager` করার পূর্বে বরং আপনাকে অবশ্যই `addToBackStack()` কল করতে হবে।

নেটি: যখন আপনি একটি ফ্রাগমেন্ট প্রতিস্থাপন বা বিয়োজন করবেন এবং বাকস্ট্যাক ট্যাবে ট্রানজেকশনটি সংযুক্ত করবেন, ফ্রাগমেন্টটি যাবি বিয়োজিত হয়েছে তা থেকে (স্ট্যাক) যায় (ধবংস হয় না)। যদি ইউজার ফ্রাগমেন্ট ট্রানজেকশন করতে নেভিগেট করে পেছন দিকে আসে, এটা রিস্টার্ট করে। যদি আপনি বাকস্ট্যাক এ ট্রানজেকশন সংযোজন না করে থাকেন, যখন ফ্রাগমেন্ট প্রতিস্থাপন বা বিয়োজন করা হয় তখন এটা ধবংস হয়ে যায়।

একটি ফ্রাগমেন্টের সাথে আরেকটির প্রতিস্থাপনের উদাহরণ:

```
// Create fragment and give it an argument specifying the article it should show
ArticleFragment newFragment = new ArticleFragment();
Bundle args = new Bundle();
args.putInt(ArticleFragment.ARG_POSITION, position);
newFragment.setArguments(args);

FragmentManager transaction = getFragmentManager().beginTransaction();

// Replace whatever is in the fragment_container view with this fragment,
// and add the transaction to the back stack so the user can navigate back
transaction.replace(R.id.fragment_container, newFragment);
transaction.addToBackStack(null);

// Commit the transaction
transaction.commit();
```

`addToBackStack()` মেথডটি একটি এন্ট্রি স্ট্যাকের অপারামিটার নয় যা ট্রানজেকশনের জন্য একটি ইউনিক নাম সুনির্দিষ্ট করে। নামটির কোন পর্যোজন নেই যদি না আপনি `FragmentManager.BackStackEntry` এপিআই ব্যবহার করে উন্নত ফ্রাগমেন্ট অপারেশন সম্পাদন করার পরিকল্পনা করে থাকেন।

অন্য ফ্রাগমেন্টের সাথে যোগাযোগ

(<http://developer.android.com/training/basics/fragments/communicating.html>)

ফ্রাগমেন্ট ইউজার ইন্টারফেস কম্পোনেন্ট ব্যবহারের জন্য, আপনাকে সম্ভবত সবসময় পুনরুৎপাদিত টায়াঁ ডিউলার কমেপোনেন্ট তৈরী করা উচিত যা এর নিজস্ব লেআউট এবং আচরণ নিধারণ করে। পূর্বে আপনি যদি এই পূর্ণ ব্যবহারযোগ্য ফ্রাগমেন্টগুলোকে নিরূপন করে থাকেন, আপনি তাদেরকে একটি টিভিভিটির সাথে একত্ব করে দিতে পারেন এবং সাবস্ক্রিপশন ইউজার ইন্টারফেসকে অনুধাবন করতে অ্যাপলিকেশন লজিকে সাথে এদের সংযুক্ত করে দিন।

মাঝে মাঝে আপনি চাবেন একটি ফ্রাগমেন্ট আরেকটি ফ্রাগমেন্টের সাথে যোগাযোগ করুক, উদাহরণস্বরূপ একটা ইউজার ইন্টারফেস উপর ভিত্তি করে কন্টেন্ট পরিবর্তন করা। সকল ফ্রাগমেন্ট থেকে ফ্রাগমেন্টের যোগাযোগটা সংযুক্ত করে একটি টিভিভিটির মাধ্যমে ঘটে থাকে। দুইটা ফ্রাগমেন্টের সরাসরি যোগাযোগ করা উচিত নয়।

একটি ইন্টারফেস নিধন করা

একটি ফর্গেমন্টকে এর একটি ভিউটি পযর্ন্ত তৈরিযোগে যোগ করা
অনুমেয়াদন করতে আপনি ফর্গেমন্ট ক্লাসে একটাই ইন্টারফেস
নিধন করার পোন্ন এবং একটি ভিউটির মধ্যে এর বাস্তবায়ন করুন।
ফর্গেমন্টটি এর `onAttach()` লাইফসাইকেলে মেথড সময়ে ইন্টারফেস
বাস্তবায়নকে কয়্যাপচার করে এবং তারপর একটি ভিউটির সাথে
যোগাযোগ স্থাপনের জন্য ইন্টারফেস মেথডকে কল করে।

এখন একটি ভিউটি কমিউনিকেশনে ফর্গেমনেন্টের একটি উদাহরণ দেয়া
হবে।

```
public class HeadlinesFragment extends ListFragment {
    OnHeadlineSelectedListener mCallback;

    // Container Activity must implement this interface
    public interface OnHeadlineSelectedListener {
        public void onArticleSelected(int position);
    }

    @Override
    public void onAttach(Activity activity) {
        super.onAttach(activity);

        // This makes sure that the container activity has implemented
        // the callback interface. If not, it throws an exception
        try {
            mCallback = (OnHeadlineSelectedListener) activity;
        } catch (ClassCastException e) {
            throw new ClassCastException(activity.toString()
                + " must implement OnHeadlineSelectedListener");
        }
    }

    ...
}
```

এখন ফর্গেমন্ট `OnHeadlineSelectedListener` ইন্টারফেসের `mCallback` ইনসটেন্স
ব্যবহার করে `onArticleSelected()` কে কল করার মাধ্যমে একটি ভিউটিতে
ময়াজ/বাতর্যাপাঠাতে পারে।

উদাহরণস্বরূপ, নিম্নোক্ত ফর্গেমনেন্টের মধ্যে ক্যার মেথডটিকে
করা হয় যখন ইউজার লিস্ট আইটেম ক্লিক করে। ফর্গেমন্টটি
পয়্যারেন্ট একটি ভিউটিতে ইভেন্টটি পেয়ে ছিঁদতে কলব্যয়্যাক ইন্টারফেস
ব্যবহার করে।

```
@Override
public void onListItemClick(ListView l, View v, int position, long id) {
    // Send the event to the host activity
    mCallback.onArticleSelected(position);
}
```


ইন্টারফেস বাস্তবায়ন করা

ফর্গামেন্ট থেকে ইন্টকলব্যাক গ্রহণ করতে, একটি টিভিটি যা একে আশ্রয় দেয় তার উচিত ফর্গামেন্ট ক্লাসে নিম্নরূপে ইন্টারফেস অবশ্যই বাস্তবায়ন করা।

উদাহরণস্বরূপ, নিম্নোক্ত একটি টিভিটি উপরে অঙ্কিত উদাহরণ থেকে ইন্টারফেস বাস্তবায়ন করেছে।

```
public static class MainActivity extends Activity
    implements HeadlinesFragment.OnHeadlineSelectedListener{
    ...

    public void onArticleSelected(int position) {
        // The user selected the headline of an article from the HeadlinesFragment
        // Do something here to display that article
    }
}
```

ফর্াগমে ন্ট বাতর্ া (মেসজ) পে াঁ ছি দিন

হে াস্ ট একটি টি ভি টি findFragmentById() দিয়ে Fragment কে কয় াপ চ ার কর ার মা ধ য় মে ফর্ াগমে ন্ট মেসজ পে াঁ ছি দি তে প াঁ রে, ত ার পর সর াসরি ফর্ াগমে ন্ট র প াবিলি ক মে থ ড কে কল করে।

উদ াহরণ সব রূ প, চিন্ত া করুন যে একটি টি ভি টি যে ট া উপ রে দে খ া নে া হ য়ে ছি আরে কটি ট ফর্ াগমে ন্ট কে ধ ারন করে তে প াঁ রে য া উপ রে া ক্ত কলব য় া ক মে থ ডে ডর মে ধ য় ি ফ রে আস া ড া ট া দ ব় ার া নি ধ র় া রিত আইটে ম কে প র্ দ শ র্ ন করে ত া। একে ষ্ টে ত র় একটি টি ভি টি কলব য় া ক মে থ ডে রি স ভ কর া ত থ য় অন য় ফর্ াগমে ন্ট প াস করে দি তে প াঁ রে য া আইটে ম কে প র্ দ শ র্ত কর বে।

```
public static class MainActivity extends Activity
    implements HeadlinesFragment.OnHeadlineSelectedListener{
    ...

    public void onArticleSelected(int position) {
        // The user selected the headline of an article from the HeadlinesFragment
        // Do something here to display that article

        ArticleFragment articleFrag = (ArticleFragment)
            getSupportFragmentManager().findFragmentById(R.id.article_fragment);

        if (articleFrag != null) {
            // If article frag is available, we're in two-pane layout...

            // Call a method in the ArticleFragment to update its content
            articleFrag.updateArticleView(position);
        } else {
            // Otherwise, we're in the one-pane layout and must swap frags...

            // Create fragment and give it an argument for the selected article
            ArticleFragment newFragment = new ArticleFragment();
            Bundle args = new Bundle();
            args.putInt(ArticleFragment.ARG_POSITION, position);
            newFragment.setArguments(args);

            FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();

            // Replace whatever is in the fragment_container view with this fragment,
            // and add the transaction to the back stack so the user can navigate back
            transaction.replace(R.id.fragment_container, newFragment);
            transaction.addToBackStack(null);

            // Commit the transaction
            transaction.commit();
        }
    }
}
```

ডাটাসেভ করা

(<http://developer.android.com/training/basics/data-storage/index.html>)

অধিকাংশ অ্যান্ড্রয়েড অ্যাপে ডাটাসেভ করার দরকার হয়, এমনকি শুধুমাত্র onPause() এর সময় অ্যাপ স্টেট সম্পর্কিত তথ্য সেভ করা যাতে ইউজারের অগ্রগতি হারিয়ে না যায়। অধিকাংশ নতুন আইডিয়াল অ্যাপে ইউজার সেটিংস সেভ করতে হয়, এবং কিছু অ্যাপে অবশ্যই ফাইলে এবং ডাটাবেজে অনেক তথ্য বসে রাখতে হয়। এই ক্লাস আপনাকে অ্যান্ড্রয়েডের পিরমিন্সপাল ডাটাসেটের জাপশন এর সাথে পরিচয় করিয়ে দেবে, যার মধ্যে রয়েছে:

- একটি টেশয়ারডিপারফরেন্স ফাইলে সিম্পল ডাটাইপ এর কি-ভ্যালু পেয়ার সেভ করা
- অ্যান্ড্রয়েডের ফাইল সিস্টেম আরবিটরি ফাইল সেভ
- SQLite কতৃক পরিচালিত ডাটাবেজ ব্যবহার

এই অধ্যায়ের অনুশীলনীসমূহ

কিক-ভয্যালুটেসটেসভ

কিক-ভয্যালুটেপয়ারেছেটআকারেবতথয্টেস্টারকরারজনয্শেযারড
িপর্ফারেবন্সফাইলবয্বহারিশিখুন।

ফাইলটেসভকরা

বেসিকফাইলটেসভকরাশিখুন,যেমনলমব্াকর্মেবডাটাস্টারকরাযা
সাধারনতকর্মানুসারেপড়তেহয়।

SQL ডাটাবেজেডাটাসভকরা

কাঠামোবদ্ধডাটাপড়তেএবংলিখতেSQLiteডাটাবেজবয্বহারকরতে
শিখুন।

কিক-ভয়ালু স্টেসভ করা

(<http://developer.android.com/training/basics/data-storage/shared-preferences.html>)

যদি আপনার কিক-ভয়ালুর অপেক্ষাকৃত ছোট সংগ্রহ থাকে যা আপনি সেভ করতে চান, আপনার SharedPreferences এপিআই ব্যবহার করা উচিত। একটি SharedPreferences অবজেক্ট ফাইল কন্টেইনিং কিক-ভয়ালু পেমেন্ট সেকেন্ডারি করে এবং তাদের পড়তে এবং লিখতে সরল মেথড সরবরাহ করে। প্রতিটা SharedPreferences ফাইল ফর্মওয়ার্ড কর্তৃক পরিচালিত হয় এবং এটা পরাইভেট বা শেয়ারড হতে পারে।

এই ক্লাস আপনাকে দেখাবে সহজ সরল ভয়ালু স্টোর এবং তা উদ্ধার করতে কীভাবে SharedPreferences এপিআই ব্যবহার করতে হয়।

নোট: SharedPreferences এপিআই শুধুমাত্র কিক-ভয়ালু পেমেন্ট পড়া এবং লেখার জন্য এবং আপনি Preferences এর সাথে এটাকে মিলিয়ে ফেলি দ্ব্যর্থতা নিবৃত্তি হবে না, কারণ Preferences আপনার অ্যাপ সিস্টেম এর জন্য ইউজার ইন্টারফেস তৈরীতে সহায়তা করে (যদিও তারা অ্যাপ সিস্টেম সেভ করতে তাদের বাস্তবায়ন হিসাবে SharedPreferences ব্যবহার করে)। Preferences এপিআই ব্যবহার করা সম্পর্কিত তথ্যের জন্য Settings (<http://developer.android.com/guide/topics/ui/settings.html>) গাইড দেখুন।

একটি Shared Preferences এ হয্‌ ানে ন্‌ ডল ল াভ করা

দুইটার মধ্যে একটা মেথড কল করার মাধ্যমে আপনি একটি টন শেয়ারড
পারফারেন্স তৈরী করতে পারেন বা বিদ্যমান একটি টে
প্রেবশে যাগ য় তা তৈরী করতে পারেন:

- `getSharedPreferences()` বয্‌ বহার করুন, আপনার যদি নাম দব্‌ ারা চিহ্ন নত
মািল্‌ টপল শেয়ারড পারফারেন্স দরকার হয়, যা আপনি পৰ্থম
পয্‌ ারা মিটার দি়ে সুনিনিদর্ষ্‌ ট করেন। আপনি এটা আপনার অয্‌ াপে
যেকোন Context থেকে কল করতে পারেন।
- `getPreferences()`-একটা একটি টিভি থেকে এটা বয্‌ বহার করুন যদি আপনার
একটি টিভি টির জন্য শূধু একটি শেয়ারড পারফারেন্স ফাইল
বয্‌ বহারের পরে য়াজন হয়। কারণ এটা একটি ডিফল্‌ ট শেয়ারড
পারফারেন্স উদধার করে যা একটি টিভি টির সাথে থাকে, আপনার একটা
নাম পরদান করার পরে য়াজন নেই।

উদাহরণসব্‌ রূপ, নিম্নে নাক্ত কোড একটি Fragment এর মধ্যে স্‌ষ্টিত
হয়েছে। এটা শেয়ারড পারফারেন্স ফাইল অনুপ্রেবশ করায় যা ারিসে সাসর্
িস্‌ টর্ ঁ `R.string.preference_file_key` কত্‌ র্‌ক চিহ্ন নত এব্‌ পর াইভে টে মাড
বয্‌ বহার করে এটা ওপেন করে যাতে ফাইলটি শূধু মাতর্‌ আপনার অয্‌ াপ
কত্‌ র্‌ক পরে বশে যাগ য় হয়।

```
Context context = getActivity();  
SharedPreferences sharedPref = context.getSharedPreferences(  
    getString(R.string.preference_file_key), Context.MODE_PRIVATE);
```

যখন আপনার শেয়ারড পারফারেন্স ফাইলের চিহ্ন নতকরন করা হয়, একটা
সব্‌ তন্‌ তর্‌ এব্‌ আপনার অয্‌ াপে সহ্‌ জি চিহ্ন নত করা যায় আপনার উচি
তেমন একটা নাম বয্‌ বহার করা, যেমন `"com.example.myapp.PREFERENCE_FILE_KEY"`

অপরিদকে যদি আপনার একটি টিভি টির জন্য শূধু একটি শেয়ারড
পারফারেন্স ফাইল দরকার হয়, আপনি `getPreferences()` মেথড বয্‌ বহার করতে
পারেন:

```
SharedPreferences sharedPref = getActivity().getPreferences(Context.MODE_PRIVATE);
```

সতর্‌ তা: আপনি যদি `MODE_WORLD_READABLE` বা `MODE_WORLD_WRITEABLE` দি়ে একটি
শেয়ারড পারফারেন্স ফাইল তৈরী করেন, তাহলে অন্য যেকোন অয্‌ াপ যা
জানে যে ফাইল আইডে নটি ফায়ার আপনার ডাটাতে অনুপ্রেবশ করতে পারবে।

শেয়ারড প্রিফারেন্স লেখা (রাইট)

একটি শেয়ারড প্রিফারেন্স ফাইল লিখতে, আপনার `SharedPreferences` উপরে `edit()` কল করার মাধ্যমে একটি `SharedPreferences.Editor` তৈরী করতে পারেন।

কি এবং ভ্যালু টা পাস করুন যা আপনি `putInt()` এবং `putString()` এর মাধ্যমে থ ডি দিয়ে লিখতে চান। তারপর পরিবর্তনটাকে সেভ করতে `commit()` কল করুন।

উদাহরণস্বরূপ:

```
SharedPreferences sharedPref = getActivity().getPreferences(Context.MODE_PRIVATE);
SharedPreferences.Editor editor = sharedPref.edit();
editor.putInt(getString(R.string.saved_high_score), newHighScore);
editor.commit();
```

শেয়ারড পিয়ারফরেন্স থেকে পাঠ করা

একটি শেয়ারড পিয়ারফরেন্স ফাইল থেকে ভ্যালু উদ্ধার করতে, `getInt()` এবং `getString()` এর মতো মেথড কল করুন, যে ভ্যালুটি আপনি চান তার জন্য যেকোনো সর্বসহায়ক করুন এবং যদি কীবোর্ডে যখন না থাকে ফিরে আসে তৈরি হচ্ছে তাহলে একটি ডিফল্ট ভ্যালু। উদাহরণস্বরূপ:

```
SharedPreferences sharedPref = getActivity().getPreferences(Context.MODE_PRIVATE);
int defaultValue = getResources().getInteger(R.string.saved_high_score_default);
long highScore = sharedPref.getInt(getString(R.string.saved_high_score), defaultValue);
```


ফাইল সেভ করা

(<http://developer.android.com/training/basics/data-storage/files.html>)

অ্যান্ড্রয়েড একটা ফাইল সিস্টেম ব্যবহার করে যা অ্যান্ড্রয়েড প্ল্যাটফর্মের রিডস্ক্রিপ্টেড ফাইল সিস্টেমের মতো। এই অনুশীলনী আলোচনা করে File এপিআই দিয়ে লিখতে বা পড়তে কীভাবে অ্যান্ড্রয়েড ফাইল সিস্টেম দিয়ে কাজ করতে হয়।

একটি File অবজেক্ট ক্যানরকমিস্কপ করা ছাড়াই শুরু থেকে শেষ পর্যন্ত কর্মানুসারে ব্যাপক সংখক ডাটালিখতে বা পড়তে পারার জন্য যথোপায়কৃত। উদাহরণস্বরূপ, এটাইমেজ ফাইল বা একটি নোটওয়ার্ডের ওপরে পরিবর্তিত ক্যানকিছুর জন্য ভালো।

এই অনুশীলনী দেখাবে কীভাবে আপনার অ্যান্ড্রয়েডে বেসিক ফাইল সম্পর্কিত কাজ সম্পাদন করা হয়। এই অনুশীলনী অনুমান করে যে আপনি বেসিক লিনাক্স ফাইল সিস্টেম এবং java.io এর মধ্যে স্ট্যান্ডার্ড ফাইল ইনপুট/আউটপুট এপিআই এর সাথে পরিচিত।

ইন্টারনাল বা এক্সট্যান্সাল স্টোরেজ পছন্দ করা

সকল অ্যান্ড্রয়েড ডিভাইসের দুইটা ফাইল সিস্টেমের জেলা আছে: "internal" এবং "external" সিস্টেমের জেলা। এই নামগুলো অ্যান্ড্রয়েডের প্রথমাদিককার দিনগুলো থেকেই চলে আসছে, যখন অধিকাংশ ডিভাইস বিল্ট-ইন নন-ভল্যাটাইল মেমরী (ইন্টারনাল সিস্টেমের জেলা) দিত, এর সাথে রিমুভযোগ্য সিস্টেমের জেলা মিমিডিয়াম যেমন একটি টমাইকের এএসডি কার্ড (এক্সট্যান্সাল সিস্টেমের জেলা) দিত। কিছু ডিভাইস "internal" এবং "external" পাঠিত্রশাল এই স্থায়ী সিস্টেমের জেলা সেপেটিভ ভক্ট ছিল, সুতরাং একটি রিমুভযোগ্য সিস্টেমের জেলা মিমিডিয়াম থাকার সাথে তবুও এখানে সবসময় দুইটা সিস্টেমের জেলা সেপস থাকে এবং এপিআই আচরণ একই থাকে, এক্সট্যান্সাল সিস্টেমের জেলা রিমুভযোগ্য বল হোক বা না হোক। নিম্নে এক তালিকাগুলো পরিতটাস্টোরেজ সেপস সম্বন্ধিকরত ফরাক্টস গুলোর সঙ্কিষপ্ত বর্ণনা।

ইন্টারনাল সিস্টেমের জেলা:

- এটা সবসময় পাওয়া যায়
- ফাইলটা এখানে সেভ হয় যা শুধুমাত্র বাইডিফল্ট আপনার অ্যাপ কন্ট্রল করবে বশেষাগয়
- যখন ইউজার আপনার অ্যাপ আনইনস্টল করে, সিস্টেমটি ইন্টারনাল সিস্টেমের জেলা থেকে আপনার অ্যাপের সমস্ত ফাইল অপসারণ করে

যখন আপনি নিশ্চিত হতে চাইবেন যে ইউজার বা অন্য অ্যাপ আপনার ফাইলে প্রবেশ করতে পারবেন না সেক্ষেত্রে ইন্টারনাল সিস্টেমের জেলাই উত্তম।

এক্সট্যান্সাল সিস্টেমের জেলা:

- এটা সবসময় পাওয়া যায় না, কারণ ইউজার ইউএসবি সিস্টেমের জেলা হিসাবে এক্সট্যান্সাল সিস্টেমের জেলাকে পরসারিত করতে পারে এবং কিছু ক্ষেত্রে এটাকে ডিভাইস থেকে রিমুভ করে ফেলতে পারে।
- এটা ওয়াল্ডের-রিভেবল, তাই ফাইলগুলো এখানে সেভ হয় আপনার নিয়ন্ত্রণের বাইরে থেকে পঠিত হতে পারে
- যখন ইউজার আপনার অ্যাপ আনইনস্টল করে, সিস্টেমটি আপনার অ্যাপের সমস্ত ফাইল অপসারণ করে শুধুমাত্র তখনই আপনি যখন `getExternalFilesDir()` থেকে ডিরেক্টরিতে তাদের সেভ করে থাকেন।

এক্সট্যান্সাল সিস্টেমের জেলা হতেছে সেই সকল ফাইলের জন্য যে সবচেয়ে ভালো জায়গা যা প্রবেশের ক্ষেত্রে বিধিনিষেধ দাবী করেনা এবং সেই সকল ফাইলের জন্য যা আপনি অন্য অ্যাপের সাথে শেয়ার করতে চান অথবা ইউজারকে একটি কম্পউটার দিয়ে এখানে প্রবেশ করাটা অনুমোদন করেন।

টিপ: যদিও অ্যাপ ইন্টারনাল সিস্টেমের জেলা বাইডিফল্ট ইনস্টলড হয়, আপনি আপনার মেনিফেস্টে `android:installLocation` এটি রিবিউটিটি সুনির্দিষ্ট করতে পারেন যাতে আপনার অ্যাপ এক্সট্যান্সাল সিস্টেমের জেলা ইনস্টলড হতে পারে। ইউজার এই অপশনকে সাদরে গ্রহণ করে যখন এপিকে সাইজ অনেক বড় হয় এবং যাদের একটা এক্সট্যান্সাল সিস্টেমের জেলা থাকে যা ইন্টারনাল সিস্টেমের জেলা থেকে বড় হয়। আরও তথ্যের জন্য App Install Location (link: <http://developer.android.com/guide/topics/data/install-location.html>) দেখুন।

এক্সটানর্নাল স্টোরেজের জন্য অনুমতি গ্রহণ

এক্সটানর্নাল স্টোরেজে লিখতে, আপনার মেনিফেস্টের মধ্যে `WRITE EXTERNAL STORAGE` পারমিশনকে অবশ্যই আবেদন করতে হয়:

```
<manifest ...>
  <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
  ...
</manifest>
```

সতর্কতা: বর্তমানে সকল অ্যাপের কোন বিশেষ অনুমোদন ছাড়াই এক্সটানর্নাল স্টোরেজ রিড করার সামর্থ্য আছে। যাহোক, এটা ভবিষ্যতে পরিবর্তন হবে। যদি আপনার অ্যাপের এক্সটানর্নাল স্টোরেজ রিড করার প্রয়োজন হয় (কিন্তু এটাতে লেখা নয়), তখন আপনার `READ EXTERNAL STORAGE` অনুমোদনকে/পারমিশন ডিকে লয় করার প্রয়োজন। আপনার অ্যাপ পর্তুগীশা অনুযায়ী কাজ করেছে এটা নিশ্চিত করতে আপনার উচিত এই পারমিশন ডিকে লয় করার, পরিবর্তনটা ফলাফল গ্রহণ করার পূর্বেই।

```
<manifest ...>
  <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
  ...
</manifest>
```

যাহোক, যদি আপনার অ্যাপ `WRITE EXTERNAL STORAGE` পারমিশন ব্যবহার করে, তখন এটাতে সম্ভব হতে পারে এক্সটানর্নাল স্টোরেজ রিড করারও পারমিশন থাকে।

ইন্টারনাল স্টোরেজে ফাইল সেভ করতে আপনার কোন পারমিশন প্রয়োজন নেই। আপনার অ্যাপের সবসময়ই এর ইন্টারনাল ডিরেক্টরির মধ্যে কার ফাইল লেখা বা পড়ার পারমিশন থাকে।

ইন্টারনাল স্টোরেজ ফাইল সেভ করা

যখন ইন্টারনাল স্টোরেজে একটি ফাইল সেভ করবেন, দুইটা মেথডের একটি কল করে আপনি একটি File হিসাবে একটি যথাযথ ডিরেক্টরীতে প করেন:

getFilesDir()

আপনার অ্যাপের জন্য একটা ইন্টারনাল ডিরেক্টরীর পর্তিনিধিত্বকারী একটা File ফিরিয়ে আন

getCacheDir()

আপনার অ্যাপের অস্থায়ী পথপযব ফাইলের জন্য একটা ইন্টারনাল ডিরেক্টরীর পর্তিনিধিত্বকারী একটা ফাইল ফিরিয়ে আন। যখন আর পর্য়েয়োজন হবে না তখন পর্তিতটা ফাইল ডিলিট করাটানিশ্চিত করুন এবং যেকোন নিদির্ষ্ট সময়ে যে পরিমান মেমরী আপনি বয্ বহার করেন তার সাথে সামঞ্জস্যপূর্ণ সাইজ লিমিট বাস্তবায়ন করুন, যেমন ১ মেগাবাইট (MB)। সিসে্টমটি যদি খুব কম স্টোরেজ এরান করা শুরূ করে, এটাকোন পূবর্ সতর্কতা ছাড়াই আপনার পথপযব ফাইল ডিলিট করে দিতে পারে।

এই ডিরেক্টরীর যে কোন একটাতে একটানতুন ফাইল তৈরী করতে, উপরোক্ত মেথড যা আপনার ইন্টারনাল স্টোরেজ ডিরেক্টরীকে সুনিদির্ষ্ট করে সেটাকতূর্ক পর্ দত্ File পাস করতে আপনি File() কনস্ট্রাকটর বয্ বহার করতে পারেন। উদাহরণসব্রূপ:

```
File file = new File(context.getFilesDir()  
filename);
```

অপরপক্ষে, আপনি একটি FileOutputStream পেতে openFileOutput() কল করতে পারেন যা আপনার ইন্টারনাল ডিরেক্টরীর একটি ফাইল লিখে। উদাহরণসব্রূপ, এখানে দেয়া হলোকীভাবে কিকছু টেক্সট ফাইল লিখতে হয়:

```
String filename = "myfile";  
String string = "Hello world!";  
FileOutputStream outputStream;  
  
try {  
    outputStream = openFileOutput(filename, Context.MODE_PRIVATE);  
    outputStream.write(string.getBytes());  
    outputStream.close();  
} catch (Exception e) {  
    e.printStackTrace();  
}
```

অথবা যদি আপনার কিকছু ফাইল গুপ্ত অবস্থায় সঞ্চয় করার পর্য়োজন হয়, পরিবর্তে আপনার createTempFile() বয্ বহার করা উচিত। উদাহরণসব্রূপ, নিম্নোক্ত মেথড URL থেকে ফাইল নামকে পৃথক করে এবং আপনার অ্যাপের ইন্টারনাল কয্াশ ডিরেক্টরীতে ওই নামে একটি ফাইল তৈরী করে:

```

public File getTempFile(Context context, String url) {
    File file;
    try {
        String fileName = Uri.parse(url).getLastPathSegment();
        file = File.createTempFile(fileName, null, context.getCacheDir());
    } catch (IOException e) {
        // Error while creating file
    }
    return file;
}

```

নোট: আপন‌ার অ‌্য‌াপ‌র ইন্‌টারন‌াল‌ স‌ে‌স‌্‌ট‌া‌র‌ে‌জ‌ ড‌ির‌েক‌্‌ট‌র‌ী অ‌্য‌ান‌্‌ড‌‌র‌্‌য়ে‌ড‌ ফ‌াই‌ল‌ স‌িস‌ে‌স‌্‌ট‌েম‌র‌ এক‌টি‌ ট‌ি‌ব‌ে‌শ‌ ল‌ে‌া‌ক‌শ‌েন‌ আপন‌ার অ‌্য‌াপ‌‌র‌ প‌য‌্‌া‌ক‌জ‌ন‌া‌ম‌ নি‌দি‌র‌্‌ষ‌্‌ট‌ক‌ের‌। ক‌ৌ‌শ‌ল‌গ‌ত‌ভ‌া‌ে‌ব‌, অন‌য‌্‌ অ‌্য‌াপ‌ আপন‌ার ইন্‌টারন‌াল‌ ফ‌াই‌ল‌ প‌ড়‌ে‌ত‌ প‌া‌র‌ে‌ যদি‌ আপ‌নি‌ ফ‌াই‌ল‌ মে‌া‌ড‌ে‌ক‌ি‌র‌ে‌ ড‌ব‌ল‌ হ‌ে‌ত‌ সে‌ট‌ক‌ের‌ন‌। ক‌িন‌্‌ত‌ু‌, অন‌য‌্‌ অ‌্য‌াপ‌‌ক‌ে‌ও‌ আপন‌ার অ‌্য‌াপ‌ প‌য‌্‌া‌ক‌জ‌ন‌েম‌ এব‌্‌ ফ‌াই‌ল‌ নে‌ম‌জ‌ান‌ে‌ত‌ হ‌ে‌ব‌। অন‌য‌্‌ অ‌্য‌াপ‌ আপন‌ার ইন্‌টারন‌াল‌ ড‌ির‌েক‌্‌ট‌র‌ী ব‌র‌্‌া‌উ‌স‌ ক‌র‌ে‌ত‌ প‌া‌র‌ে‌ব‌ না‌ এব‌্‌ ল‌ে‌খ‌া‌র‌ বা‌ প‌ড়‌া‌র‌ প‌র‌্‌ে‌ব‌ শ‌া‌ি‌ধ‌ক‌া‌র‌ থ‌া‌ক‌ে‌ব‌ না‌ যদি‌ না‌ আপ‌নি‌ প‌ি‌র‌ষ‌্‌ক‌া‌র‌ভ‌া‌ে‌ব‌ ফ‌াই‌ল‌টি‌ ট‌ি‌র‌ে‌ ড‌ব‌ল‌ বা‌ রা‌ই‌ে‌ট‌ব‌ল‌ হ‌ে‌ত‌ সে‌ট‌না‌ ক‌ের‌ন‌। সু‌ত‌র‌া‌্‌ য‌ত‌ক‌্‌ষ‌ণ‌ আপ‌নি‌ ইন্‌টারন‌াল‌ স‌ে‌স‌্‌ট‌া‌র‌ে‌জ‌র‌ উপ‌র‌ আপন‌ার ফ‌াই‌ল‌র‌ জন‌য‌্‌ MODE PRIVATE ব‌য‌্‌ ব‌হ‌া‌র‌ ক‌র‌ে‌ব‌ন‌, ত‌া‌র‌া‌ অন‌য‌্‌ অ‌্য‌াপ‌‌র‌ জন‌য‌্‌ প‌র‌্‌ে‌ব‌ শ‌ে‌য‌া‌গ‌য‌্‌ হ‌ে‌ব‌ না‌।

এক্সটানর্নাল স্টোরেজ ফাইল সেভ করা

কারণ এক্সটানর্নাল স্টোরেজ সহজপার্শ্বিক নাও হতে পারে-যেমন যখন ইউজার একটা পিসিতে স্টোরেজ পরিসীমিত করে অথবা এসডি কার্ডের যা এক্সটানর্নাল স্টোরেজ সরবরাহ করে তারিমুভ/অপসারণ করে-আপনার সবসময় যাচাই করা উচিত যে এটাতে পর্বেশ করার পূর্বে বর্তমান অবস্থায় থাকা। আপনি `getExternalStorageState()` কল করে এক্সটানর্নাল স্টোরেজ অবস্থা সম্পর্কে অনুসন্ধান করতে পারেন। যদি ফিরতি স্ট্রিং `MEDIA_MOUNTED` এর মতো হয় তাহলে আপনি আপনার ফাইল লিখতে বা পড়তে পারেন। উদাহরণস্বরূপ, নিম্নে নাক্ষত্রিক স্টোরেজ এর বিবরণ তুলে ধরা হল:

```
/* Checks if external storage is available for read and write */
public boolean isExternalStorageWritable() {
    String state = Environment.getExternalStorageState();
    if (Environment.MEDIA_MOUNTED.equals(state)) {
        return true;
    }
    return false;
}

/* Checks if external storage is available to at least read */
public boolean isExternalStorageReadable() {
    String state = Environment.getExternalStorageState();
    if (Environment.MEDIA_MOUNTED.equals(state) ||
        Environment.MEDIA_MOUNTED_READ_ONLY.equals(state)) {
        return true;
    }
    return false;
}
```

যদিও এক্সটানর্নাল স্টোরেজ ইউজার এবং অন্য অ্যাপ কন্ট্রোল
পরিবর্তনযোগ্য, সেখানে দুই ধরনের ফাইল আছে যা আপনি এখানে সেভ
করতে পারেন:

পাবলিক ফাইল

ফাইল যা অন্য অ্যাপ এবং ইউজারের জন্য মুক্তভাবে সহজলভ্য হওয়া
উচিত। যখন ইউজার আপনার অ্যাপ আনইনস্টল করবে, এই সকল ফাইল ইউজারের
কাছে থাকে উচিত।

উদাহরণ হিসাবে, আপনার অ্যাপ কন্ট্রোল কয়টি অপারেশন করা ফাইল
অন্য অ্যাপ ডাউনলোড করা ফাইল।

প্라이ভেট ফাইল

ফাইল যা আইনগতভাবে আপনার অ্যাপের সাথে থাকা উচিত এবং যখন ইউজার
আপনার অ্যাপ আনইনস্টল করে তখন এটা ডিলিট হয়ে যাওয়া উচিত। যদিও এই
সকল ফাইল কৌশলগতভাবে ইউজার বা অন্য অ্যাপ কন্ট্রোল পর্বেশগম্য
কারণ তারা এক্সটানর্নাল স্টোরেজ এ আছে, এই ফাইল যৌক্তিকভাবে
আপনার অ্যাপের প্রাইভেট কান্ট্রোল ইউজারকে ভ্যালু পর্দান করে না। যখন
ইউজার আপনার অ্যাপ আনইনস্টল করে, সিস্টেমটি আপনার অ্যাপের
এক্সটানর্নাল প্라이ভেট ডিরেক্টরীর সকল ফাইল ডিলিট করে দেয়।

উদাহরণস্বরূপ, আপনার অ্যাপ কন্ট্রোল ডাউনলোড করা অতিরিক্ত
রিসোর্স অথবা অস্থায়ী মিডিয়া ফাইল।

যদি আপনি পাবলিক ফাইল এক্সটানর্নাল স্টোরেজে সেভ করতে চান,
এক্সটানর্নাল স্টোরেজ যথাযথ কন্ট্রোল

পর্যন্ত তিনটি থিম ব্রকারী একটি File পেতে `getExternalStoragePublicDirectory()` মেথড ব্যবহার করুন। এই মেথড যে ধরনের ফাইল আপনাকে সেভ করতে চান তা সুনির্দিষ্ট করে একটি আলাদা চ্যানেলে নিয়ে আসে যাতে তার অনন্য পাবলিক ফাইলের সাথে যৌক্তিকভাবে সংগঠিত হতে পারে, যেমন, DIRECTORY MUSIC বা DIRECTORY PICTURES। উদাহরণস্বরূপ:

```
public File getAlbumStorageDir(String albumName) {  
    // Get the directory for the user's public pictures directory.  
    File file = new File(Environment.getExternalStoragePublicDirectory(  
        Environment.DIRECTORY_PICTURES), albumName);  
    if (!file.mkdirs()) {  
        Log.e(LOG_TAG, "Directory not created");  
    }  
    return file;  
}
```

আপনি যদি ফাইল সেভ করতে চান যা আপনার অ্যাপের কাছে পরবর্তীতে, আপনি `getExternalFilesDir()` কল করে যথার্থ ডিরেক্টরী পেতে পারেন এবং ডিরেক্টরীর ধরন যা আপনি পছন্দ করেন তা নির্দেশ করে এটা একটানে মেমোরি পাস করে দিন। এই এই উপায়ে তৈরী করে পরিত্যাগ ডিরেক্টরী পর্যন্ত ডিরেক্টরীতে যুক্ত হয় যা আপনার অ্যাপের এক্সটানসিভ স্টোরেজের সকল ফাইলকে একটি রত্ন করে রাখা যাঁসে স্টেম ডিলিট করে দেয় যখন ইউজার আপনার অ্যাপ আনইনস্টল করে দেয়।

উদাহরণস্বরূপ, এখানে একটা মেথড দেয়া আছে যা আপনি একটি একক ফ্রেম এলবামের জন্য একটি ডিরেক্টরী তৈরীতে ব্যবহার করতে পারেন:

```
public File getAlbumStorageDir(Context context, String albumName) {  
    // Get the directory for the app's private pictures directory.  
    File file = new File(context.getExternalFilesDir(  
        Environment.DIRECTORY_PICTURES), albumName);  
    if (!file.mkdirs()) {  
        Log.e(LOG_TAG, "Directory not created");  
    }  
    return file;  
}
```

যদি পূর্বনির্ধারিত সাব ডিরেক্টরীর কোন নামই আপনার ফাইলের সাথে না যায়, পরীক্ষণ করে আপনি `getExternalFilesDir()` কল করতে পারেন এবং হঠাৎ পাস করতে পারেন। এটা এক্সটানসিভ স্টোরেজে আপনার অ্যাপের পরবর্তীতে ডিরেক্টরীর জন্য রুট ডিরেক্টরী ফেরত নিয়ে আসে।

যদি মনে রাখতে যে `getExternalFilesDir()` একটি ডিরেক্টরীর মধ্যে একটি ডিরেক্টরী তৈরী করে, যা ডিলিট হয় যখন ইউজার আপনার অ্যাপ আনইনস্টল করে। যদি ফাইলটি যা আপনি সেভ করতে চান তা ইউজার কতটুকু আপনার অ্যাপ আনইনস্টল করার পরও বিদ্যমান থাকতে হয়- যেমন আপনার অ্যাপ হচ্ছে একটি ক্যামেরা এবং ইউজার ফ্রেমটোরা রাখতে চায়- পরিবর্তন আপনার `getExternalStoragePublicDirectory()` ব্যবহার করা উচিত।

যদি কোন কারণে আপনি ফাইলের জন্য `getExternalStoragePublicDirectory()` ব্যবহার করেন যা শয়ার্ড বা ফাইলের জন্য `getExternalFilesDir()` যা আপনার অ্যাপের কাছে পরবর্তীতে, এটা গুরুত্বপূর্ণ যে আপনি এপিআই কনস্টেন্ট কর্তৃক প্রদত্ত DIRECTORY_PICTURE এর মতো ডিরেক্টরী নেম ব্যবহার করেছেন। এই ডিরেক্টরী নেম নিশ্চিত করে যে ফাইলটি সিস্টেম কর্তৃক যথার্থভাবে যত্ন পাবে, উদাহরণস্বরূপ, DIRECTORY_RINGTONES সেভ হওয়া ফাইল সিস্টেম মিডিয়া সংরক্ষণ করার কতটুকু মিমিউজিকের পরিবর্তন রিএন্টান্সিভেবল শরণীবদ্ধ হয়।

ফির্ স্পেস অনুষন্ধান

যদি আপনি আগে থেকে জানেন নিক পরিমাণ ডাটা আপনি সেভ করছেন, `getFreeSpace()` বা `getTotalSpace()` কল করে একটি `IOException` না যাটি টেই আপন খুঁজে বের করতে পারেন যে কৈখায় পযর্াপ্ত স্পেস আছে। এই মেথডগুলো বিদ্যমান পরাপ্তসাধয় স্পেস এবং স্টোরেজ ভলিউমের সবর্মেটি স্পেস আলাদাভাবে সরবরাহ করে। এই তথ্য নিদির্ষ্ট সীমার বাইরে স্টোরেজ ভলিউম পরিপূর্ণ করাপরিহার করতে উপকারী।

যাহাকিসেস্টমিনিশ্চয়তাদেয়নাযে আপন `getFreeSpace()` কতর্ক নিদির্ষিত যতখুশিততবাইটসিলখতে পারেন। আপন যে সাইজের ডাটা সেভ করতে চান তার চাইতে কৈয়কমেগাবাইটে বেশী যদি নামবর্ারিটি ফেরত দেয়, বা ফাইলিসেস্টম ৯০% এরচেয়ে কম পূর্ণ হয়, তাহলে এটা এগিয়ে নেয়ার জনয় নিরাপদ। অন্যথায় আপনার সম্ভবত স্টোরেজে লেখা উচিত নয়।

নেটি: আপনার ফাইলে সেভ করার পূর্বে বর্বিদ্যমান স্পেসের পরিমাণ চেক করা আপনার জনয় পর্য়েয়োজনীয় নয়। আপন পরিবর্তে স্গ স্গ তা লিখতে চেষ্টা করতে পারেন, তারপর একটি `IOException` কয়্াচ করুন, যদি একটি য়েটি থাকে। আপনার হয়তে এটা করার দরকার হতে পারে যদি আপনি সঠিকভাবে না জানেন আপনার নিক পরিমাণ স্পেসের পর্য়েয়োজন। উদাহরণস্বরূপ, PNG ইমেজ JPEG তে কনভার্ট করার মাধ্যমে এটা সেভ করার পূর্বে বর্ আপন যদি ফাইলের এনেকোডিং পরিবর্তন করেন, আপন পূর্বে বর্ ইফাইলের সাইজ জানতে পারবেন না।

ফাইল ডিলিট করা

যে সকল ফাইলের আর দরকার নেই তা আপনার সব সময় ডিলিট করে ফেলা উচিত। একটা ফাইল ডিলিট করার সবচেয়ে সোজা সাপোর্টারাস্তাহচ্ছে ওপেনড ফাইলের ফােরেন্স যা নিয়ে জেত delete()কল করে।

```
myFile.delete();
```

যদি ফাইল ইন্টারনাল ফাইলে সেভ হয়, আপনি deleteFile()কল করে একটি ফাইলের স্থান চিহ্নিত করা এবং ডিলিট করতে Contextকে জিজ্ঞাসাও করতে পারেন:

```
myContext.deleteFile(fileName);
```

নোট: যখন ইউজার আপনার অ্যাপ আনইনস্টল করে, অ্যান্ডরয়েড সিস্টেম নিজে থেকে কত বিষয়গুলো ডিলিট করে:

- যে সকল ফাইল আপনি ইন্টারনাল স্টোরে জে সেভ করেছেন
- getExternalFilesDir() ব্যবহার করে যে সকল ফাইল আপনি এক্সটারনাল স্টোরে জে সেভ করেছেন

যাহোক, আপনার উচিত ম্যানুয়ালি getCacheDir()দ্বারা তৈরী সকল ক্যাশ ফাইল নিয়মিতভাবে ডিলিট করা এবং অন্যান্য ফাইল যা আপনার আর দরকার নেই তাও নিয়মিতভাবে ডিলিট করুন।

SQL ডাটাবেজে ডাটাসেভ করা

(<http://developer.android.com/training/basics/data-storage/databases.html>)

পুনরাবৃত্তি তহওয়া ডাটাব্যবহারে মাবদ্ধ ডাটার জন্য ডাটাবেজে
ডাটাসেভ করাটা ভালো, যেমন কনটাক্ট ইনফর্মেশন। এই ক্লাস অনুমান
করে যে আপনি যেটা মডিউলটি ডাটাবেজে SQL ডাটাবেজে এর সাথে পরিচিত এবং
আপনাকে অ্যান্ডরয়েড SQLite ডাটাবেজে সহকাজ শুরু করতে সাহায্য
করে। এপিআই যা অ্যান্ডরয়েড ডাটাবেজে ব্যবহার করতে আপনার
পরে যাজন হবে তা `android.database.sqlite` প্যাকেজে আছে।

স্কিমা (Schema) এবং কন্ট্রাক্ট নির্ধারণ

SQL ডাটাবেজে অন্য়তম পদ্ধতি নীতি হচ্ছিস্কিমা: কীভাবে ডাটাবেজ
বিনয়্যাস্ত হবে তার একটা আনুষ্ঠানিক ঘোষণা। স্কিমা SQL
সেট্টেমেন্ট পৰ্তিফলন হয় যা আপনি আপনার ডাটাবেজ তৈরীতে ব্যবহার
করেন। আপনি দেখবেন যে একটা সিঙ্গ ক্লাস তৈরীতে এটা সাহাজ্য করে, যা
কন্ট্রাক্ট ক্লাস হিসাবে পরিচিত, যা সম্পষ্টভাবে সিসেটেমেটিক এবং
সেল্ফ-ডকুমেমিন্টেড উপায়ে আপনার স্কিমার লেআউট সুনির্দিষ্ট করে।

একটি কন্ট্রাক্ট ক্লাস কনস্টেন্ট এর জন্য কন্টেইনার যা URLs, টেবল এবং
কলাম এর জন্য নাম নির্ধারণ করে। কন্ট্রাক্ট ক্লাস আপনার একই
পয়াকে জরম্বে অন্য় সকল ক্লাসে একই কনস্টেন্ট ব্যবহার করাটাকে
অনুমেদন করে। এটা আপনাকে একটা স্থানে কলাম নাম পরিবর্তন করতে
দেয় এবং আপনাকে ডেডের সর্বতর সংচ্চারিত হতে দেয়।

একটি কন্ট্রাক্ট ক্লাস বিনয়্যাস করতে ভালো উপায় হচ্ছসংজ্ঞা
আপেরা করা যা ক্লাসের রুটলেভেল সম্পূর্ণরূপে ডাটাবেজে বৈশিষ্ট্য হয়।
তারপর পর্তিটাবেটবলের জন্য একটা ইনার ক্লাস তৈরী করুন যা এর
কলামকে গণনা করে।

নোট: BaseColumns ইন্টারফেসটি বাস্তবায়ন করার মাধ্যমে, আপনার ইনার
ক্লাস _ID যাকিছু অন্য়ান্ডরয়েড ক্লাস নামে একটা পৰ্াথমিক কিক
ফিল্ড পেতে পারে যেমন কারসর অন্য়াদাপটর এটার থাকাকে পর্তয্যাশা
করে। এটার পর্যেয়াজন নেই তবে এটা অন্য়ান্ডরয়েড ফর্মওয়াকর্দি দিয়ে
আপনার ডাটাবেজ চম কারভাবে কাজ করতে পারবে।

উদাহরণসব্বরূপ, এই স্নেপেট একটি সিঙ্গলেটিবলের জন্য টেবিল নাম
এবং কলাম নাম নির্ধারণ করে দেয়:

```
public final class FeedReaderContract {  
    // To prevent someone from accidentally instantiating the contract class,  
    // give it an empty constructor.  
    public FeedReaderContract() {}  
  
    /* Inner class that defines the table contents */  
    public static abstract class FeedEntry implements BaseColumns {  
        public static final String TABLE_NAME = "entry";  
        public static final String COLUMN_NAME_ENTRY_ID = "entryid";  
        public static final String COLUMN_NAME_TITLE = "title";  
        public static final String COLUMN_NAME_SUBTITLE = "subtitle";  
        ...  
    }  
}
```

SQL হেল্পার ব্যবহার করে ডাটাবেজ তৈরী

পূর্বে বর্ত আপনি যদি ঠিক করে থাকেন আপনার ডাটাবেজ দেখতে কেমন হবে, আপনাকে মেথড বাস্‌তবায়ন করতে হবে যা ডাটাবেজ এবং টেবিল তৈরী এবং দেখানো করে। এখানে কিছু সাধারণ বিবৃতি আছে যা একটি টেবিল তৈরী এবং ডিলিট করে:

```
private static final String TEXT_TYPE = " TEXT";
private static final String COMMA_SEP = ",";
private static final String SQL_CREATE_ENTRIES =
    "CREATE TABLE " + FeedEntry.TABLE_NAME + " (" +
    FeedEntry._ID + " INTEGER PRIMARY KEY," +
    FeedEntry.COLUMN_NAME_ENTRY_ID + TEXT_TYPE + COMMA_SEP +
    FeedEntry.COLUMN_NAME_TITLE + TEXT_TYPE + COMMA_SEP +
    ... // Any other options for the CREATE command
    ")";

private static final String SQL_DELETE_ENTRIES =
    "DROP TABLE IF EXISTS " + FeedEntry.TABLE_NAME;
```

ঠিক ডিভাইসে সরি ইন্টারনাল স্টোরেজ (internal storage) সেভ করা ফাইলের মতো, অ্যান্ডরয়েড আপনার ডাটাবেজ পরাইভেটিভিস্‌ক স্‌পেস জমা করে যা স্‌কুইলিট অ্যাপলিকেশন। আপনার ডাটাবিনরাপদ, কারন বাইডিফল্ট এই এলাকা অনয্‌ অ্যাপলিকেশনের জন্য উন্মুক্ত নয়।

SQLiteOpenHelper ক্লাসে উপকারী এক সেট এপিআই পাওয়া যায়। যখন আপনি আপনার ডাটাবেজে রেফারেন্স পেতে এই ক্লাস ব্যবহার করেন, সিস্টেমটি ডাটাবেজ তৈরী এবং আপডেট এর সম্ভাব্য ল্য-রাইনিং অপারেশন সম্পাদন করে শুধুমাত্র যখন পর্যেয়াজন হয় তখন, এবং অ্যাপ শুরুর সময়ে নয়। আপনার এই সব করতে `getWritableDatabase()` এবং `getReadableDatabase()` কল করা।

নোট: কারন তারাল্য-রাইনিং হতে পারে, নিশ্চিত হোন যে আপনি একটি ব্যব্যাকগর্‌াউন্ড থের্ড এ `getWritableDatabase()` বা `getReadableDatabase()` কল করেছেন যেমন `AsyncTask` বা `IntentService` দিয়ে।

SQLiteOpenHelper ব্যবহার করতে একটি সাবক্লাস তৈরী করুন যা `onCreate()`, `onUpgrade()` এবং `onOpen()` কল ব্যব্যাক মেথডকে ওভাররাইড করে। আপনি হয়তো `onDowngrade()` বাস্‌তবায়ন করতে চান, কিন্তু এটার পর্যেয়াজন নেই।

উদাহরণস্বরূপ, এখানে SQLiteOpenHelper এর একটি বাস্‌তবায়ন আছে যা উপরে দেখানো কমান্ডেরিকছু ব্যবহার করে:

```

public class FeedReaderDbHelper extends SQLiteOpenHelper {
    // If you change the database schema, you must increment the database version.
    public static final int DATABASE_VERSION = 1;
    public static final String DATABASE_NAME = "FeedReader.db";

    public FeedReaderDbHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(SQL_CREATE_ENTRIES);
    }
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // This database is only a cache for online data, so its upgrade policy is
        // to simply to discard the data and start over
        db.execSQL(SQL_DELETE_ENTRIES);
        onCreate(db);
    }
    public void onDowngrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        onUpgrade(db, oldVersion, newVersion);
    }
}

```

আপনার ডাটাবেজে পর্বেশ করতে, আপনার SQLiteOpenHelper এর সাবক্লাস শুরু করুন:

```

FeedReaderDbHelper mDbHelper = new FeedReaderDbHelper(getContext());

```

তথ্য একটা ডাটাবেজে রাখুন

insert() মেথডে ContentValues অবজেক্ট পাস করে ডাটাবেজে ডাটাপ্রবেশ করান:

```
// Gets the data repository in write mode
SQLiteDatabase db = mDbHelper.getWritableDatabase();

// Create a new map of values, where column names are the keys
ContentValues values = new ContentValues();
values.put(FeedEntry.COLUMN_NAME_ENTRY_ID, id);
values.put(FeedEntry.COLUMN_NAME_TITLE, title);
values.put(FeedEntry.COLUMN_NAME_CONTENT, content);

// Insert the new row, returning the primary key value of the new row
long newRowId;
newRowId = db.insert(
    FeedEntry.TABLE_NAME,
    FeedEntry.COLUMN_NAME_NULLABLE,
    values);
```

insert() এর পরে থম আলাচনা হচ্ছে শুধুমাত্র টেবিল নাম। দ্বিতীয় আলাচনা (আগুরে মেন্ট) একটা কলামের নাম প্রদান করে যেখানে ফর্মওয়ার্ক ইভেন্ট NULL প্রবেশ করতে পারবে যেতে ContentValues খালি হয় (আপনি যদি পরিবর্তন "null" এ এটাসেট করেন, যখন এখানে কোন ভ্যালু থাকবে না তখন ফর্মওয়ার্ক একটি রো প্রবেশ করাবে না)

ডাটাবেজ থেকে তথ্য পঠ করা

ডাটাবেজ থেকে পড়তে, `query()` মেথড ব্যবহার করুন, আপনার বাছাই পরীক্ষার জন্য এবং কান্ডিত কলামে এটি পাস করে। মেথডটি `insert()` এবং `update()` এর এলিমেন্টগুলি একটি রত করে, ব্যতিক্রম শূন্য কলাম লিস্ট, আপনি যে ডাটাবেজে চান তা নির্ধারণ করে, ডাটাবেজ ক্যারেনার পূর্বের। অনুসন্ধানে ফলাফল আপনার কাছে একটি `Cursor` অবজেক্টের মধ্যে ফিরে আসবে।

```
SQLiteDatabase db = dbHelper.getReadableDatabase();

// Define a projection that specifies which columns from the database
// you will actually use after this query.
String[] projection = {
    FeedEntry._ID,
    FeedEntry.COLUMN_NAME_TITLE,
    FeedEntry.COLUMN_NAME_UPDATED,
    ...
};

// How you want the results sorted in the resulting Cursor
String sortOrder =
    FeedEntry.COLUMN_NAME_UPDATED + " DESC";

Cursor c = db.query(
    FeedEntry.TABLE_NAME, // The table to query
    projection,            // The columns to return
    selection,             // The columns for the WHERE clause
    selectionArgs,         // The values for the WHERE clause
    null,                  // don't group the rows
    null,                  // don't filter by row groups
    sortOrder              // The sort order
);
```

কাসরের মধ্যে একটি রেখে দেওয়া `Cursor` মুভ মেথড ব্যবহার করুন, রিডিং ভ্যালু শুরুর করার পূর্বের আপনার অবশ্যই সবসময় যাক কল করতে হবে। সাধারণত, `moveToFirst()` কল করার মাধ্যমে আপনার শুরুর করা উচিত, যা "read position" কে রেজাল্টের প্রথম পর্বের উপরে প্লস করে। পরিত্যাগের এজন্য, আপনি একটি কলামের ভ্যালু পড়তে পারেন যে কোন একটি `Cursor` গেটে মেথড কল করার মাধ্যমে, যেমন, `getString()` বা `getLong()`। পরিত্যাগে গেটে মেথডের জন্য আপনার অবশ্যই আপনার কান্ডিত কলামের ইন্ডেক্স পি জিশন পাস করতে হবে, যা আপনি `getColumnIndex()` বা `getColumnIndexOrThrow()` কল করে পেতে পারেন। উদাহরণস্বরূপ:

```
cursor.moveToFirst();
long itemId = cursor.getLong(
    cursor.getColumnIndexOrThrow(FeedEntry._ID)
);
```

ডাটাবেজে থেকে তথ্য ডিলিট করা

একটি টেবিল থেকে রেডিও ডিলিট করতে, আপনার সিলেকশন করা ইন্ট্রির যা পর্দান করা উচিত যার থেকে চিহ্নিত করে। ডাটাবেজ এপিআই সিলেকশন করা ইন্ট্রির জন্য একটি মেসেজ বসে পর্দান করে যা SQL ইন্জেকশন এর কাছে থেকে প্রতিরোধ করে। মেসেজ বসে টি সিলেকশন সেস্ট্রি সফিকেশনকে সিলেকশন কলজ এবং সিলেকশন আগুরমেন্ট এভাগ করে। কলজ কলামকে নির্ধারণ করে দেখতে এবং কলাম টেস্ট একটি রত করতে আপনাকে অনুমোদনও করতে দেয়। আরগুমেন্ট হচ্ছে ছত্য়ালু বিপরীত পরিক্ষা যা কলজের মধ্যে আবদ্ধ হয়। কারণ নিয়মিত SQL সেস্ট্রি টেমেন্টের মতো একটি ডাটাবেজ ল টপরিচালিত হয় না, এটা SQL ইন্জেকশন থেকে নিরাপদ।

```
// Define 'where' part of query.  
String selection = FeedEntry.COLUMN_NAME_ENTRY_ID + " LIKE ?";  
// Specify arguments in placeholder order.  
String[] selectionArgs = { String.valueOf(rowId) };  
// Issue SQL statement.  
db.delete(table_name, selection, selectionArgs);
```


ডাটাবেজ আপডেট করা

যখন আপনার ডাটাবেজ ভ্যালুর একটি সাবেকটি পরিবর্তন করতে হবে, তখন `update()` মেথড ব্যবহার করুন।

টেবিল আপডেট করে `update()` এর বিষয়বসিনটয়াক্স দিয়ে `insert()` এর কন্টেন্ট ভ্যালু সিনটয়াক্স একটিরত করে।

```
SQLiteDatabase db = mDbHelper.getReadableDatabase();

// New value for one column
ContentValues values = new ContentValues();
values.put(FeedEntry.COLUMN_NAME_TITLE, title);

// Which row to update, based on the ID
String selection = FeedEntry.COLUMN_NAME_ENTRY_ID + " LIKE ?";
String[] selectionArgs = { String.valueOf(rowId) };

int count = db.update(
    FeedReaderDbHelper.FeedEntry.TABLE_NAME,
    values,
    selection,
    selectionArgs);
```

অন্য অ্যাপের সাথে সম্পর্ক স্থাপন

(<http://developer.android.com/training/basics/intents/index.html>)

একটি অ্যান্ড্রয়েডের সব্‌ভাষিকভাবে বিকল্প একটি ভিত্তি থাকা (activities)। প্রতিটি একটি ভিত্তি একটি ইউজার ইন্টারফেস প্রদর্শন করে যা ইউজারকে একটি সুনির্দিষ্ট কাজ করতে দেয় (যেমন একটি ম্যাপ দেখা বা একটি ফটো নেওয়া)। ইউজারকে একটি ভিত্তি থেকে আরেকটি একটি ভিত্তিতে তিনিয়ে যেতে আপনার অ্যাপকে অবশ্যই ইচ্ছা করতে, অ্যাপের "intent" কনিধরারণ করতে একটি Intent ব্যবহার করা উচিত। যখন আপনি startActivity() এর মতো একটি মেথড দিয়ে সিস্টেমে একটি Intent পাঠান, সিস্টেম যথাযথ অ্যাপ উপাদান খুঁজে বের করতে এবং শুরু করতে Intent ব্যবহার করে। ইন্টেন্ট ব্যবহার আপনার অ্যাপকে অন্য অ্যাপের সাথে একটি একটি ভিত্তি শুরু করতে অনুমোদন করে।

একটি নির্দিষ্ট কাজ সম্পাদনা (একটি নির্দিষ্ট Activity ইনস্ট্যান্স) শুরু করার জন্য একটি Intent *explicit* হতে পারে অথবা কোন সম্পাদনা পরিচালিত কাজ (যেমন একটি ফটো ধারণ করা) করে তা শুরু করতে ইন্টেন্ট *implicit* হতে পারে।

এই ক্লাস আপনাকে দেখাবে কীভাবে অন্য অ্যাপের সাথে ইচ্ছা মেসেজিং মিথস্ক্রিয়ার কাজ সম্পাদন করতে একটি Intent ব্যবহার করা হয়, যেমন অন্য অ্যাপ শুরু করতে, ওই অ্যাপ থেকে রজাল্ট গ্রহণ করতে এবং আপনার অ্যাপকে অন্য অ্যাপের ইন্টেন্টের প্রতিক্রিয়া জানানোর মতো সক্ষম করা।

এই অধ্যায়ের অনুশীলন সমূহ

ইউজারকে অনয অয াপ পাঠানো

দেখানো হবে অনয অয াপ যা একটি কাজ করতে পারবে তা শুরুর করতে আপন কীভাবে ইমপ্লিসিট ইনটেন্ টৈতরী করতে পারবেন

একটি একটি ভিডিও থেকে রেজাল্ ট পাওয়া

দেখানো হবে কীভাবে অনয একটি ভিডিও শুরুর করতে হয় এবং একটি ভিডিও থেকে কীভাবে একটি রেজাল্ ট গরহণ করতে হয়

অনয অয াপকে আপনার অয াপ শুরুর করতে দেয়া

দেখানো হবে ইনটেন্ টি ফল্ টার যা আপনার অয াপ কতরক গরহণ কৃত ইমপ্লিসিট ইনটেন্ টি কে ডিউক্লয়ার করে তানিধরারণ করার মাধ্যমে অনয অয াপ কতরক বযবহার করার জন্য কীভাবে আপনার অয াপের একটি ভিডিও তৈরী করতে হয়।

ইউজারকে অনয্য অয্যাপ পাঠানো

(<http://developer.android.com/training/basics/intents/sending.html>)

একটি অয্যান্ডরয়েডের সবচেয়ে গুরুত্বপূর্ণ বৈশিষ্ট্য হচ্ছে একটি ক্যাজের (যা করতে চাওয়া হয়) উপরিত্তিত করে ইউজারকে অনয্য অয্যাপ পাঠানোর ক্ষেত্রে একটি অয্যাপের সামথর। উদাহরণস্বরূপ, যদি আপনার অয্যাপের একটি ব্যবসার ঠিকানা থাকে যা আপনি পরদর্শন করতে চান, ময্যাপ পরদর্শনের জন্য আপনার অয্যাপ একটি টিভিটি তৈরী করার দরকার নেই। পরিবর্তে আপন একটি Intent ব্যবহার করে ঠিকানা দেওয়ার জন্য একটি রিকোয়েস্ট তৈরী করতে পারেন। তখন অয্যান্ডরয়েড সিসেইটম একটি অয্যাপ শুরুর করে বয্যাময্যাপের উপর ঠিকানা শো করতে সামথর্য্য হবে।

পরথমক্লাস (Building Your First App) যেভাবে ব্যাখ্যা করা হয়েছে, আপনার অয্যাপের একটি টিভিটির মধ্যমেনিভিগেট করতে আপনাকে অবশ্যই ইনটেন্ট ব্যবহার করতে হবে। আপনি সাধারণভাবে তা এক্সপ্লিসিট ইনটেন্ট দিয়ে করতে পারবেন, যাকমেপানেন্টের সঠিক ক্লাস নাম নিখরারন করে যা আপনি শুরুর করতে চান। কিন্ত যখন আপনি চান একটি কাজ করতে একটি আলাদা অয্যাপ থাকুক, যেমন “একটি ময্যাপ দেখা” আপনাকে তখন অবশ্যই একটি ইমপ্লিসিট ইনটেন্ট ব্যবহার করতে হবে।

এই অনুশীলনী আপনাকে শোখাবে একটি সুনির্দিষ্ট ক্যাজের জন্য একটি ইমপ্লিসিট ইনটেন্ট তৈরী করা যায় এবং কীভাবে এটাকে ব্যবহার করে একটি টিভিটি শুরুর করা যায় যা অনয্য অয্যাপ কাজটি সম্পাদন করে।

একটি ইমপিপ্লিসিটি ইন্টেন্ট তৈরী করা

ইমপিপ্লিসিটি ইন্টেন্ট শুরুর করতে কেম্পোনেন্ট এর ক্লাসে নিম্ন ডিকে লয়ার করে নীচের তলু পরিবর্তিত সম্পাদন করতে একটি একশন ডিকে লয়ার করে। একশনটি যে বিষয়গুলো আপন করতে চান তা নিদিষ্ট করে, যেমন, কোন কিছুরি ভউ, এডিট, সেন্ড, গেট ইত্যাদী। ইন্টেন্ট মাঝে মাঝে একশনের সাথে সম্পৃক্ত ডাটা অন্তর্ভুক্ত করে, যেমন ঠিকানাটি যা আপন দেখতে চান অথবা ইমেইল মেসেজ যা আপন সেন্ড করতে চান। আপন কী ধরনের ইন্টেন্ট তৈরী করতে চান তার উপর নির্ভর করে, ডাটা একটি Uri হতে পারে, অন্য আরও ডাটার একটি অথবা ইন্টেন্টের আদৌ ডাটার দরকার নেই।

আপনার ডাটা যদি Uri হয়, একটা সরল Intent() আছে যা আপন একশন এবং ডাটা নির্ধারণ করতে ব্যবহার করতে পারেন।

উদাহরণস্বরূপ, এখানে দেখানো হয়েছে টেলিফোন নামবার সুনির্দিষ্ট করতে Uri ডাটা ব্যবহার করে একটা ফোন কল আরম্ভ করতে কীভাবে একটি ইন্টেন্ট তৈরী করতে হয়:

```
Uri number = Uri.parse("tel:5551234");
Intent callIntent = new Intent(Intent.ACTION_DIAL, number);
```

যখন আপনার অ্যাপ startActivity() কল করে এই ইন্টেন্টকে আবাহন করে, ফোন অ্যাপটি পরদত্ত ফোন নামবার একটি কল করে।

এখানে আছে একেজাড়া অন্য ইন্টেন্ট এবং তাদের একশন এবং Uri ডাটার জোড়া:

- View a map:/ ম্যাপ ভউ একটি:

```
// Map point based on address Uri location = Uri.parse("geo:0,0?q=1600+Amphitheatre+Parkway,+Mountain+View,+California"); // Or map point based on latitude/longitude // Uri
location = Uri.parse("geo:37.422219,-122.08364?z=14"); // z param is zoom level Intent mapIntent = new
Intent(Intent.ACTION_VIEW, location);
```

- View a web page:/ ওয়েব পেজ ভউ:

```
Uri webpage = Uri.parse("http://www.android.com");
Intent webIntent = new Intent(Intent.ACTION_VIEW, webpage);
```

ইমপিপ্লিসিটি ইন্টেন্টের অন্য ধরনে “এক্সট্রা” ডাটার পরয়োজন যা নির্ধারন ডাটা টাউপস পরদান করে, যেমন একটি স্ট্রিং। আপন নামা ধরনের putExtra() মেথড ব্যবহার করে এক্সট্রা ডাটার এক বা একাধিক অংশ সংযুক্ত করতে পারেন।

বাইন্ডিং ফল্ট, সিসেস্টমটি Uri ডাটা যা অন্তর্ভুক্ত আছে তার উপর নির্ভিত করে ইন্টেন্ট কতরকচাওয়া যথায়থ MIME টাইপটি নির্ধারণ করে। যদি আপন একটি Uri ইন্টেন্ট অন্তর্ভুক্ত না করেন, সবার্ভাবিকভাবে ইন্টেন্টের সাথে সম্পৃক্ত ডাটার টাইপ সুনির্দিষ্ট করতে আপনার উচিত setType() ব্যবহার করা। MIME টাইপটি সেটিং করে আরও সুনির্দিষ্ট করার এই ধরনের একটি ভিভিটিগুেলার ইন্টেন্ট গ্রহণ করা উচিত।

এখানে আরও কিছু ইন্টেন্ট আছে যা কাঙ্ক্ষিত একশন সুনির্দিষ্ট করতে এক্সট্রা ডাটা এড করে:

- এটা চেষ্টা মেন্টসহ একটা ইমেইল পাঠান:

```
Intent emailIntent = new Intent(Intent.ACTION_SEND);
// The intent does not have a URI, so declare the "text/plain" MIME type
emailIntent.setType(HTTP.PLAIN_TEXT_TYPE);
emailIntent.putExtra(Intent.EXTRA_EMAIL, new String[] {"jon@example.com"}); // recipients
emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Email subject");
emailIntent.putExtra(Intent.EXTRA_TEXT, "Email message text");
emailIntent.putExtra(Intent.EXTRA_STREAM, Uri.parse("content://path/to/email/attachment"));
// You can also attach multiple items by passing an ArrayList of Uris
```

- একটি ক্যালেন্ডার ইভেন্ট তৈরী করুন:

```
Intent calendarIntent = new Intent(Intent.ACTION_INSERT, Events.CONTENT_URI);
Calendar beginTime = Calendar.getInstance().set(2012, 0, 19, 7, 30);
Calendar endTime = Calendar.getInstance().set(2012, 0, 19, 10, 30);
calendarIntent.putExtra(CalendarContract.EXTRA_EVENT_BEGIN_TIME, beginTime.getTimeInMillis());
calendarIntent.putExtra(CalendarContract.EXTRA_EVENT_END_TIME, endTime.getTimeInMillis());
calendarIntent.putExtra(Events.TITLE, "Ninja class");
calendarIntent.putExtra(Events.EVENT_LOCATION, "Secret dojo");
```

নোট: ক্যালেন্ডার ইভেন্টের জন্য এই ইন্টেন্ট শুধুমাত্র এপিআই লেভেল ১৪ এবং এর চেয়ে উপরের লেভেল সাপোর্ট করে।

নোট: এটা গুরুত্বপূর্ণ যে আপনি যতটুকু সুনির্দিষ্ট করা সম্ভব ঠিক সেভাবে আপনার Intent নিধরারন করেছেন। উদাহরণস্বরূপ, যদি আপনি ACTION_VIEW ইন্টেন্ট ব্যবহার করে একটি ইমেজ পর্দা দেখান, আপনার image/* এর একটি MIME টাইপ সুনির্দিষ্ট করা উচিত। এটা অ্যাপস পরিহার করে যা ইন্টেন্ট কতৃক শুরু হওয়া থেকে অন্য ধরনের ডাটা (যেমন একটি ম্যাপ) কে "ভিউ" করতে পারে।

ইন্টেন্ট টিগিং করলে একটি অ্যাপ আফ্রিকান যাত্রাই করুন

যদিও অ্যান্ড্রয়েড প্ল্যাটফর্মের নিশ্চিততায় দেখে যেকোন ইন্টেন্ট যেকোন একটি বিল্ট-ইন অ্যাপস এসুইচিং করে থাকবে (যেমন, ফোন, ইমেইল বা ক্যালেন্ডার অ্যাপ), একটি ইন্টেন্টকে আবাহন করার পূর্বের আপনার সব সময় একটি যাত্রাইকরণ পদক্ষেপ থাকে।

সতর্কতা: আপনি যদি একটা ইন্টেন্ট আবাহন করেন এবং ডিভাইসে ইন্টেন্ট টিকে কয়েক কাজ করার মতো কোন অ্যাপ না থাকে, আপনার অ্যাপ ক্র্যাশ করবে।

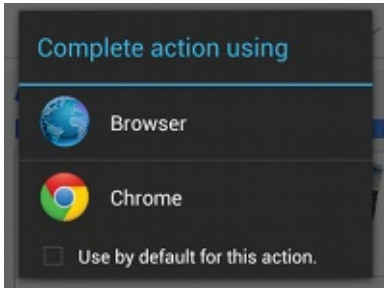
যাত্রাই করার জন্য একটা একটি ভিডিও আফ্রিকান ইন্টেন্টের সপন করে, আপনার Intent কে কয়েক কাজ করতে পারে এমন একটি ভিডিওর তালিকা পেতে QueryIntentActivities() কে কল করুন। যদি ফিরতি List থাকে না থাকে, আপনি নিরাপত্তা কাজ শুরু করতে পারেন। উদাহরণস্বরূপ:

```
PackageManager packageManager = getPackageManager();
List<ResolveInfo> activities = packageManager.queryIntentActivities(intent, 0);
boolean isIntentSafe = activities.size() > 0;
```

যদি isIntentSafe, true হয়, তখন কমপক্ষে একটি অ্যাপ ইন্টেন্টের প্রতি সপন করে। যদি এটা false হয়, তখন ইন্টেন্টকে কয়েক কাজ করার মতো কোন অ্যাপ থাকে না।

নোট: আপনার উচিত এই যাত্রাই কাজ সম্পাদন করা যখন আপনার একটি ভিডিও পরামর্শ শুরু করে, যদি আপনার চিহ্নটি নিশ্চিত করার দরকার হয় যে চিহ্ন ইন্টেন্টটি ইউজার কতটুকু ব্যবহারের পর্যায়ে পৌঁছেছে বা ব্যবহার করে থাকে। আপনি যদি কোন নিশ্চিত অ্যাপ সম্পর্কে জানেন যেটা ইন্টেন্টকে কয়েক কাজ করতে পারে, আপনি অ্যাপটি ইউজার কতটুকু ডাউনলোড করার জন্য একটি লিংক সরবরাহ করতে পারেন। (দেখুন গুগল পেলে কীভাবে আপনার পেগে লিংক যুক্ত করবেন) (<http://developer.android.com/distribute/googleplay/promote/linking.html>)

ইন্টেন্ট দিয়ে একটি এক্টিভিটি শুরুর ক্রম



ফিগার ১. সিস্টেম ডায়ালগের উদাহরণ যা দৃশ্যমান হয় যখন একাধিক অ্যাপ একটি এক্টিভিটি নিয়ন্ত্রণ করতে পারে।

পূর্বে আপনি যদি আপনার Intent তৈরী করে থাকেন এবং এক্সট্রা ইনফোর্মেশন ক্রয়ের থাকেন, এটি সিস্টেম স্টেজে সনাক্ত করে `startActivity()` কল করেন। যদি সিস্টেমটি একের অধিক এক্টিভিটি চিহ্নিত করে থাকে যা ইন্টেন্টটিকে নিয়ন্ত্রণ করতে পারে, এটি ইউজারের জন্য কোন অ্যাপটি ব্যবহার করতে বাতাবেছে নিতে একটি ডায়ালগ প্রদর্শন করে, ফিগার ১ এ যেভাবে দেখানো হয়েছে। যদি শুধু একটি এক্টিভিটি থাকে যা ইন্টেন্ট নিয়ন্ত্রণ করে, সিস্টেমটি তা স্বয়ংক্রিয়ভাবে এটা শুরু করে।

`startActivity(intent);`

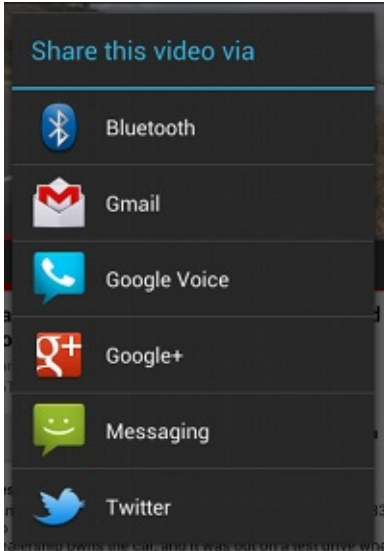
এখানে একটা সম্পূর্ণ উদাহরণ যা দেখাবে কীভাবে একটা অ্যাপ দেখার জন্য একটি ইন্টেন্ট তৈরী করতে হয়, ইন্টেন্ট নিয়ন্ত্রণ করতে একটা অ্যাপ বিদ্যমান আছে তা যাচাই করা, তারপর এটা শুরু করা:

```
// Build the intent
Uri location = Uri.parse("geo:0,0?q=1600+Amphitheatre+Parkway,+Mountain+View,+California");
Intent mapIntent = new Intent(Intent.ACTION_VIEW, location);

// Verify it resolves
PackageManager packageManager = getPackageManager();
List<ResolveInfo> activities = packageManager.queryIntentActivities(mapIntent, 0);
boolean isIntentSafe = activities.size() > 0;

// Start an activity if it's safe
if (isIntentSafe) {
    startActivity(mapIntent);
}
```


একটি অ্যাপ চুজার পরদাশরন করা



ফিগার ২. একটি চুজার ডায়ালগ

লক্ষ্য করুন যে যখন `startActivity()` তে `Intent` পাস করার মাধ্যমে আপনি একটি একটি ভিউটি শুরু করেন এবং সেখানে একের অধিক অ্যাপ আছে যা ইন্টেন্টের পরিত্রাসপনস করে তখন ইউজার বাছাই করতে পারবে যে কোন অ্যাপ বাই ডিফল্ট বয়বহার করতে হবে (ডায়ালগের একদমনীচে একটি চেকবক্স সিলেক্ট করে, ফিগার ১. দেখুন)। এটা ভালো, যখন একটা একশন সম্পাদন করা হয় যার জন্য ইউজার সাধারণত পরিতবার একই অ্যাপ বয়বহার করতে চায়, যেমন যখন একটি ওয়েব পেজ ওপেন করা হয় (ইউজার সম্ভবত মাত্র একটা ওয়েব বরাউজার বয়বহার করে) বা একটা ফটো নেওয়া হয় (ইউজার সম্ভবত একটা ক্যামেরা পছন্দ করে)।

কিন্তু, যদি সম্পাদিত একশন বহুবিধ অ্যাপস কর্তৃক নিয়ন্ত্রিত হয় এবং ইউজার পরিতবার ভিন্ন ভিন্ন অ্যাপ পছন্দ করতে পারে-যেমন একটি “শেয়ার” একশন, যার জন্য ইউজারের অনেকগুলো অ্যাপস থাকতে পারে যার মাধ্যমে তারা একটি আইটেম শেয়ার করতে পারে-আপনার উচিত সম্প্রতিভাবে একটা চুজার ডায়ালগ পরদাশরন করা যেভাবে ফিগার ২ তে দেখানো হয়েছে। চুজার ডায়ালগ পরিতবার একশনের জন্য কোন অ্যাপ বয়বহার করতে হবে তা বেছে নিতে ইউজারকে বাধ্য করে (ইউজার একশনের জন্য ডিফল্ট অ্যাপ বেছে নিতে পারেনা)।

একটি চুজার পরদাশরন করতে `createChooser()` বয়বহার করে একটি `Intent` তরী করুন এবং `startActivity()` পরিত এটা পাস করে দিন। উদাহরণস্বরূপ:

```
Intent intent = new Intent(Intent.ACTION_SEND);
...

// Always use string resources for UI text.
// This says something like "Share this photo with"
String title = getResources().getString(R.string.chooser_title);
// Create and start the chooser
Intent chooser = Intent.createChooser(intent, title);
startActivity(chooser);
```

এটা একটি অ্যাপের তালিকা সহ একটি ডায়ালগ পরদাশরন করে যা `createChooser()` মেথডকে পাস করে ইন্টেন্টের পরিত্রাসপনস করে এবং ডায়ালগ টাইটেল হিসাবে সরবরাহকৃত টেক্সটটি বয়বহার করে।

একটি একটি ভিডিও থেকে রেজাল্ট পাওয়া

(<http://developer.android.com/training/basics/intents/result.html>)

অন্য একটি ভিডিও শুরু করাটা এখনুই খহেত হবেনা। আপনি একটা একটি ভিডিও শুরুও করতে পারেন এবং একটা ফিরতি রেজাল্ট গ্রহণ করতে পারেন। একটা রেজাল্ট পেতে `startActivityForResult()` কল করুন (`startActivity()` এর পরিবর্তে)।

উদাহরণস্বরূপ, আপনার অ্যাপ একটা ক্যাশেরা শুরু করতে পারে এবং ফলাফল হিসাবে ক্যাশেরা করা ফটো পাাবে। অথবা ইউজার কতটুকু একটি কনটেন্ট ব্রাউজিং করেছেন ওয়ার জন্য আপনি পিপিএল অ্যাপ শুরু করতে পারেন এবং ফলাফল হিসাবে আপনি কনটেন্ট ব্রাউজিং টাইমসেপেতে পারেন।

অবশ্যই, একটি ভিডিও টি টায়ের সপনস করে তাহে এমনভাবে ডিজাইন করতে হবে যাতে একটা ফিরতি রেজাল্ট আসে। যখন এটা হবে, এটা অন্য `Intent` অবজেক্ট হিসাবে রেজাল্ট টিকে সনড করবে। আপনার একটি ভিডিও এটা `onActivityResult()` কলব্যাংক রিসিভ করবে।

নোট: যখন আপনি `startActivityForResult()` কল করবেন আপনি এক সিম্পলিস্ট বা ইমপ্লিসিট ইন্টেন্ট ব্যবহার করতে পারেন। একটা রেজাল্ট রিসিভ করতে আপনার নিজস্ব একটি ভিডিওর মধ্যে যখন একটি শুরু করবেন, আপনার উচিত একটা এক সিম্পলিস্ট ইন্টেন্ট ব্যবহার করা এটা নিশ্চিত করতে যে আপনি একটা ক্যাশেরা রেজাল্ট গ্রহণ করবেন।

একটি টিভি টিশুরুর করা

আপনার Intent অবজেক্ট ব্যবহার করাটা তেমন বিশেষ কিছু নয় যখন একটা রেজাল্টের জন্য একটা একটি টিভি টিশুরুর করা হয়, কিন্ত আপনি `startActivityForResult()` মেথডে একটি অতিরিক্ত ইনটিজার আরগুমেণ্ট পাস করা দরকার।

ইনটিজার আরগুমেণ্ট হচ্ছে একটি “রিকোয়েস্ট কোড” যা আপনার রিকোয়েস্ট চিহ্নিত করে। যখন আপনি রেজাল্ট Intent গ্রহণ করেন, কলব্যাকটি একই রিকোয়েস্ট কোড সরবরাহ করে যাতে আপনার অ্যাপ যথাযথভাবে রেজাল্টটি চিহ্নিত করতে পারে এবং নিয়ন্ত্রণ করতে পারে কীভাবে এটা নিয়ন্ত্রণ করতে হয়।

উদাহরণস্বরূপ, এখানে দেয়া আছে কীভাবে একটি একটি টিভি টিশুরুর করতে হয় যা ইউজারকে একটি কন্টাক্ট বাছাই করতে অনুমোদন করে:

```
static final int PICK_CONTACT_REQUEST = 1; // The request code
...
private void pickContact() {
    Intent pickContactIntent = new Intent(Intent.ACTION_PICK, Uri.parse("content://contacts"));
    pickContactIntent.setType(Phone.CONTENT_TYPE); // Show user only contacts w/ phone numbers
    startActivityForResult(pickContactIntent, PICK_CONTACT_REQUEST);
}
```

রেজাল্ টি ট গ র্ হ ণ (ি রি স ভ) ক র া

যখন ইউজার পরবর্তী একটি টিভি টি দিয়ে শেষ করবেন এবং ফিরবেন, সিস্টেমটি আপনার একটি টিভি টির onActivityResult() মেথডকে কল করে। এই মেথডে তিনটা আরগুমেন্ট অন্তর্ভুক্ত করে:

- রিকোয়েস্ট কে কোডটি আপনি startActivityForResult() এ পাঠিয়েছেন
- দিব্যতীয় একটি টিভি টি কত রকম সুনির্দিষ্ট একটি রেজাল্ টি কে কোড। এটা RESULT_OK হয় যদি অপারেশন কৃতকায় হইয়ে থাকে অথবা RESULT_CANCELED হয় যদি ইউজার ফিরে আসে বা কান কারনে অপারেশন ব্যর্থ হইয়ে থাকে।
- একটি Intent যা রেজাল্ টি ডাটা বহন করে।

উদাহরণস্বরূপ, এখানে দেয়া আছে আপনি কীভাবে “পিক কন্টেন্ট” ইন্টেন্ট এর জন্য রেজাল্ টি কে নিয়ন্ত্রণ করতে পারেন:

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    // Check which request we're responding to
    if (requestCode == PICK_CONTACT_REQUEST) {
        // Make sure the request was successful
        if (resultCode == RESULT_OK) {
            // The user picked a contact.
            // The Intent's data Uri identifies which contact was selected.

            // Do something with the contact here (bigger example below)
        }
    }
}
```

এই উদাহরণে, রেজাল্ টি Intent অর্ডার দিয়ে ডের কন্টেন্ট ক্রক্ ট কত রকম ফিরে আসে অথবা পিক পল অর্প একটি কন্টেন্ট Uri সরবরাহ করে যা ইউজার যে কন্টেন্ট বাছাই করে তা চিহ্নিত করে।

রেজাল্ টি কে সফলভাবে হ্যান্ডল করতে, আপনাকে অবশ্যই বুঝতে হবে রেজাল্ টি Intent এর ফরমেটিক হতে পারে। এমনটি করা সহজ যখন একটি টিভি টি একটি রেজাল্ টি ফরত নিয়ে এসে আপনার নিজস্ব একটি টিভি টির একটি হয়। আন্ডরয়েড প্ল্যাটফর্মের সাথে অন্তর্ভুক্ত অর্প সতাদের নিজস্ব এপিআইগুলাপের স্তাব করে যাতে আপনি নির্দিষ্ট রেজাল্ টি ডাটার উপর নির্ভর করতে পারেন। উদাহরণস্বরূপ পিক পল অর্প (কিছু পুরাতন সংস্করণের উপর কন্টেন্ট অর্প) সবসময় কন্টেন্ট ইউআরআই দিয়ে একটা রেজাল্ টি ফরত আনেন যা বাছাইকৃত কন্টেন্ট চিহ্নিত করে, এবং ক্যামেরা অর্প “ডাটা” এক্সট্রাতে একটি Bitmap ফরত নিয়ে আসে (Capturing Photos সম্পর্কিত ক্লাসটি দেখুন)।

বোনাস: কনট্যাক্ট ডাটাপড়া

উপরে কন্টেকাডটি দেখায় যে পিপল অয়্যাপ থেকে কীভাবে রেজাল্ট পেতে হয় যা আসলে কীভাবে রেজাল্ট থেকে ডাটাপড়তে হয় সে সম্পর্কে এবং বিস্তারিত আলাচনায় যায় না, কারণ এটা content providers সম্পর্কে আরও উন্নত আলাচনার দাবী করে। কিন্তু আপনি যদি উ সাহীহে হান, এখানে আরও কিছু কোড আছে যা আপনাকে দেখাবে বাছাই করা কনট্যাক্ট থেকে ফোন নামব্বার পেতে কীভাবে রেজাল্ট ডাটাকে অনুসন্ধান করতে হয়:

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    // Check which request it is that we're responding to
    if (requestCode == PICK_CONTACT_REQUEST) {
        // Make sure the request was successful
        if (resultCode == RESULT_OK) {
            // Get the URI that points to the selected contact
            Uri contactUri = data.getData();
            // We only need the NUMBER column, because there will be only one row in the result
            String[] projection = {Phone.NUMBER};

            // Perform the query on the contact to get the NUMBER column
            // We don't need a selection or sort order (there's only one result for the given URI)
            // CAUTION: The query() method should be called from a separate thread to avoid blocking
            // your app's UI thread. (For simplicity of the sample, this code doesn't do that.)
            // Consider using CursorLoader to perform the query.
            Cursor cursor = getContentResolver()
                .query(contactUri, projection, null, null, null);
            cursor.moveToFirst();

            // Retrieve the phone number from the NUMBER column
            int column = cursor.getColumnIndex(Phone.NUMBER);
            String number = cursor.getString(column);

            // Do something with the phone number...
        }
    }
}
```

নোট: অয়্যান্ডরয়েড ২.৩ (এপিআই লেভেল ৯) এর পূর্বে, Contacts Provider এর উপর অনুসন্ধান করতে (উপরে যেমনটি দেখানো হয়েছে) পরয়োজন্যে আপনার অয়্যাপ READ CONTACTS পারমিশনকে ডিকে ক্লয়ার করে (Security and Permissions দেখুন)। কিন্তু অয়্যান্ডরয়েড ২.৩ দিয়ে শুরু করলে, কনট্যাক্ট/পিপল অয়্যাপ কনট্যাক্ট পর্ভাইডার থেকে পড়তে আপনার অয়্যাপকে একটি অস্থায়ী অনুমতি পর্দান করে যখন এটা আপনাকে একটা রেজাল্ট ফেরত দেবে। অস্থায়ী অনুমতি শুধুমাত্র নির্দিষ্ট কনট্যাক্টের ক্ষেত্রেই পর্যোগ হয়, যার ফলে যেটা ইন্টেন্টির Uri কতর্ক সূচিন্দির্ষ্ টেস্টা ছাড়া আপনি একটি কনট্যাক্টকে অনুসন্ধান করতে পারবেন না, যদি না আপনি READ CONTACTS পারমিশন ডিকে ক্লয়ার করেন

অনয় অয় াপকে আপন ার অনয় াপ শুরুর করতে দেয়া

(<http://developer.android.com/training/basics/intents/filters.html>)

পূর্ববর্তরী দুইটা অনুশীলনী গেল্পর একটা দিক ফোকাস করে: আপন ার অনয় াপ থেকে অনয় অয় াপের একটি ভিডিও শুরুর করা। কিন্তু যদি আপন ার অনয় াপ একটা কাজ সম্পাদন করতে পারে যা অনয় অয় াপের জন্য উপকারী হতে পারে, আপন ার অনয় াপকে অনয় অয় াপ থেকে একশন রিকোয়েস্ট এর সম্পন্ন করার মতো করে পরিস্তুত করা উচিত। উদাহরণস্বরূপ, আপনি যদি একটা সামাজিক অনয় াপ তৈরী করেন যা ইউজারের বন্ধুদের সাথে মেসেজ বা ফোটা শেয়ার করতে পারে, এটা আপন ার আগর্হের কেন্দ্র থাকা ব ACTION SEND ইন্টেন্টকে সাপোর্ট কর়াটা, ফলে ইউজার অনয় অয় াপ থেকে একটি “শেয়ার” একশন চালু করতে পারবে এবং একশনটি সম্পাদন করতে আপন ার অনয় াপ শুরুর করবে।

আপন ার একটি ভিডিও শুরুর করতে অনয় অয় াপকে অনুমোদন দিতে, আপন ার পরে যা জন অনুরূপ <activity> এ লিমেন্ট এর জন্য আপন ার মিনিফেস্ট ফাইলে < intent-filter> এ লিমেন্ট সংযুক্ত করা।

যখন আপন ার অনয় াপ একটি ডিভাইসে ইনস্টল করবেন, সিস্টেম আপন ার ইন্টেন্ট ফিল্টারস চিহ্নিত করে এবং ইনস্টল করা সকল অনয় াপ কতরক সাপোর্ট কর়া ইন্টেন্টের ইন্টানরাল কয় াটালে গে ইনফরমেশনগুলো সংযুক্ত করে। যখন একটি অনয় াপ একটা ইমপিপ্লিসিট ইন্টেন্ট দিয়ে startActivity() বা startActivityForResult() কলে, সিস্টেমটি কোন একটি ভিডিও (বা একটি ভিডিও গুলো) ইন্টেন্টের সম্পন্ন করতে পারে।

ইনটেন্ট ফিল্টার সংযোজন (এন্ড) করা

কোন ইনটেন্ট আপনার একটি ভিত্তি ধারণ করতে পারবে তা যথাযথভাবে বিন্দবর্ণারণ করার জন্য, আপনি সংযুক্ত করেছেন এমন পরীতিটাই ইনটেন্ট ফিল্টারের একটি ভিত্তি গ্রহণ করা একশন এবং ডাটার টাইপের মতো করে যতদূর সম্ভব নির্দিষ্ট হওয়া উচিত।

সিস্টেমটি একটি ভিত্তি থেকে একটি নির্দিষ্ট Intent সন্ধান করতে পারে যদিও এই একটি ভিত্তির একটি ইনটেন্ট ফিল্টার থাকে যা Intent অবজেক্টের নির্দিষ্ট বৈশিষ্ট্যগুলি পূরণ করে:

একশন

একটি সিস্টার্ম পরিচালনা করতে একশনের নামকরণ করে। সাধারণত প্লাটফর্ম-নির্ভর ভাষাগুলি যেমন ACTION_SEND বা ACTION_VIEW।

<action> এলিমেন্টটি দিয়ে আপনার ইনটেন্ট ফিল্টারের এটা সূচিন্দর্শক করুন। ভাষাগুলি যেটা আপনি এই এলিমেন্ট সূচিন্দর্শক করেছেন তা অবশ্যই একশনের জন্য পূরণ সিস্টার্মের মতো হবে, এপিআই কমসেটেন্ট এর পরিবর্তে (নীচের উদাহরণ দেখুন)

ডাটা

ইনটেন্টের সাথে সংযুক্ত ডাটার একটি বর্ণনা।

এলিমেন্টটি দিয়ে আপনার ইনটেন্ট ফিল্টারের এটা সূচিন্দর্শক করুন। এই এলিমেন্ট এক বা একাধিক এটির বিউট বয়বহার করতে, আপনি শুধু একটি MIME টাইপ, শুধু একটি ইউআরআই পেরিফক্স, শুধু একটি ইউআরআই স্কিম সূচিন্দর্শক করতে পারেন বা এগুলার সমন্বয় এবং অন্য কোন যোগ্রহন করা ডাটা টাইপ নির্দেশ করে।

নোট: যদি ডাটা Uri সম্বন্ধে আপনার সূচিন্দর্শকটি ডেকে লয়ালের দরকার না হয় (যেমন যখন আপনার একটি ভিত্তি অন্যান্য ধরনের “এক সটার্” ডাটা হয়ান্ডল করে, একটি ইউআরআই এর পরিবর্তে), আপনার একটি ভিত্তির হয়ানেন্ডল করা ডাটার ধরনের ডেকে লয়াল করতে আপনার শুধু android:mimeType এটির বিউটিটি সূচিন্দর্শক করা উচিত, যেমন *text/plain* or *image/jpeg*

ক্যাটাগরি

ইনটেন্ট পরিচালনা করা একটি ভিত্তির বর্ণনাকরতে একটা অতিরিক্ত উপায় সরবরাহ করুন, সাধারণত ইউজার চিহ্ন বা লোকেশন যেখানে থেকে এটা শুরু করেছিল। সঠিক সিস্টেম দ্বারা সাপোর্ট পাওয়া বিভিন্ন ক্যাটাগরি আছে, কিন্তু অধিকাংশ খুব কমই ব্যবহার হয়। যাহোক সকল ইমপিপ্লিসিট ইনটেন্ট বাইন্ডিং ফল্ট CATEGORY_DEFAULT দিয়ে সংজ্ঞায়িত।

এটা <category> এলিমেন্ট দিয়ে আপনার ইনটেন্ট ফিল্টারের চিহ্নিত করুন।

আপনার ইনটেন্ট ফিল্টারের, <intent-filter> এলিমেন্ট থাকার সমাগতির্য এক সমএল এলিমেন্ট দিয়ে তাদের পরীতিটাই ডেকে লয়াল করার মাধ্যমে আপনার একটি ভিত্তি কোন মানদণ্ড গ্রহণ করে তা আপনি ডেকে লয়াল করতে পারেন।

উদাহরণস্বরূপ, এখন একটি ইনটেন্ট ফিল্টার সহ একটি ভিত্তি যা ACTION_SEND ইনটেন্ট পরিচালিত করে যখন ডাটা টাইপ হয়, টেক্সট বা ইমেজ হয়:

```
<activity android:name="ShareActivity">
  <intent-filter>
    <action android:name="android.intent.action.SEND"/>
    <category android:name="android.intent.category.DEFAULT"/>
    <data android:mimeType="text/plain"/>
    <data android:mimeType="image/*"/>
  </intent-filter>
</activity>
```

পৰ্ৱৰ্তীত টা পৰ্বে বৈশাদযত ইনটেন্‌ট শূধু মাতৰ্ একি ট একশন এব ঞ একি ট ডাটা টাইপ সূনি নিদৰ্ষ্ টকৰে, কিকন তু পৰ্ৱৰ্তীত টা <intent-filter> এর মধ্যে য় <action>, <category> এব ঞ <data> এলি লে মেন্‌টৰ মালি টপল ইনস্‌টেন্‌স কৰতে এটা িঠক আছে।

যদি একশন এব ঞ ডাটার কোনে জোড়া তা দেৱ আচৰনে পাস্‌পৰিকভাবে খাপ না খায়, কোন একশন গৰ্হণে যাগ য় যখন কোন ডাটা টাইপে পৰসাথে জোড়া হব সেটা কোনটা হব তা নিনিদৰ্ষ্ টকৰতে আপনাৰ আলাদা ইনটেন্‌ট তৈৰী কৰা উচিত।

উদাহরণ সব্ৰূপ, মেনে কৰুন আপনাৰ একি টি ভি ট ACTION SEND এব ঞ ACTION SENDTO ইনটে ন্‌টৰ উভয়েৰ জনয্ টেক্‌সট এব ঞ ইমেজ উভয়ই পৰিচালনা কৰে। এই কেষ্টে তৰ্ আপনা কৈ অবশ্য ই দুই টা একশনেৰ জেনে য় দুই টা আলাদা ইনটে ন্‌ট নিধৰ্ াৰন কৰতে হব ক াৰন send বা sendto ইউআৰআই সি স্কম বয্ বহাৰ কৰে গৰ্হা হকৈ কৰি িঠকানা সূনি নিদৰ্ষ্ টকৰতে একি ট ACTION SENDTO ইনটে ন্‌টৰ অবশ্য ই ডাটা Uri বয্ বহাৰ কৰা। উদাহরণ সব্ৰূপ:

```
<activity android:name="ShareActivity">
  <!-- filter for sending text; accepts SENDTO action with sms URI schemes -->
  <intent-filter>
    <action android:name="android.intent.action.SENDTO"/>
    <category android:name="android.intent.category.DEFAULT"/>
    <data android:scheme="sms" />
    <data android:scheme="smsto" />
  </intent-filter>
  <!-- filter for sending text or images; accepts SEND action and text or image data -->
  <intent-filter>
    <action android:name="android.intent.action.SEND"/>
    <category android:name="android.intent.category.DEFAULT"/>
    <data android:mimeType="image/*"/>
    <data android:mimeType="text/plain"/>
  </intent-filter>
</activity>
```

নোট: ইমপি প্‌লিসিট ইনটে ন্‌ট গৰ্হণ কৰাৰ জনয্, আপনা কৈ অবশ্য ই ইনটে ন্‌ট ফিল্টাৰে CATEGORY DEFAULT কয্ াটা গৰি অন্তভূৰ্ ক্ত কৰতে হব। মেখড startActivity() এব ঞ startActivityForResult() সকল ইনটে ন্‌ট এর বয্ বহাৰ কৰে যেন তাৰা CATEGORY DEFAULT কয্ াটা গৰি ধাৰণ কৰে। আপনি যদি এটা ডিক্‌লয় াৰনা কৰেন, কোন ইমপি প্‌লিসিট ইনটে ন্‌ট আপনাৰ একি টি ভি ট নিধৰ্ াৰন কৰে বনা।

ACTION SEND ইনটে ন্‌ট যা সামজিক শেয়াৰ আচরণে সন্ড এব ঞ ৰিসিভ কৰে সে সম্পৰ্ কৰ্ আৰু তথ্যৰ জনয্, Receiving Simple Data from Other Apps সম্পৰ্ কৰ্ অনুশীলন াি টি দেখুন।

আপনার একটি ভিডিওতে ইন্টেন্ট বয় বস্‌থাপনা করা

আপনার একটি ভিডিওতে কিক একশন নেয়া যায় ঠিক করার জন্য, আপনি Intent পড়ে ত পারেন যা এটা শুরু করত তে বয় বহুত হয়।

যখন আপনার একটি ভিডিও শুরু করে, Intent কে উদব্‌ার করতে getIntent() কল করুন যা একটি ভিডিও শুরু করে। একটি ভিডিওর লাইফসাইকেল কালের যে কোন সময়ে আপনি এটা করতে পারেন, কিন্তু সাধাারনত আপনার এটা করা উচিত কল বয় াকে পর থমি দিকটায় যেমন onCreate() বা onStart()।

উদাহরণস্বরূপ:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    setContentView(R.layout.main);

    // Get the intent that started this activity
    Intent intent = getIntent();
    Uri data = intent.getData();

    // Figure out what to do based on the intent type
    if (intent.getType().indexOf("image/") != -1) {
        // Handle intents with image data ...
    } else if (intent.getType().equals("text/plain")) {
        // Handle intents with text ...
    }
}
```

একটি রেজাল্ট ফেরত নিয়ে আসা

যদি আপনি একটি টিভিটিতে একটি রেজাল্ট ফেরত চান যা আপনার টাআবাহন করে, রেজাল্টকে ক্যাড এবং রেজাল্ট Intent সূচনিতদর্শক করতে শূন্যমাত্রা setResult() কল করুন। যখন আপনার অপারেশন শেষ হবে এবং ইউজারকে মূল একটি টিভিটিতে ফিরে যেতে হবে, আপনার একটি টিভিটি বন্ধ করতে (এবং ধবংস করতে) finish() কল করুন। উদাহরণস্বরূপ: // Create intent to deliver some kind of result data Intent result = new Intent("com.example.RESULT_ACTION", Uri.parse("content://result_uri")); setResult(Activity.RESULT_OK, result); finish();

আপনাকে অবশ্যই সবসময় রেজাল্ট দিয়ে রেজাল্টকে ক্যাড সূচনিতদর্শক করতে হবে। সাধারণত, হয় এটা RESULT OK বা RESULT CANCELED। এরপর আপনি পরে যাঁজন অনুসারে একটা Intent দিয়ে অতিরিক্ত ডাটা সরবরাহ করতে পারেন।

নোট: রেজাল্টটি RESULT CANCELED এ বাইন্ডিফল্ট সেট হয়েছে, তাই, একশন শেষ করার পূর্ববর্তী আপনি রেজাল্ট সেট করার পূর্ববর্তী যদি ইউজার বন্ধক বাটন চাপে, মূল একটি টিভিটি "ক্যান্সেল" রেজাল্টটি রিসিভ করে।

আপনার যদি কেবল একটি ইনটিজার ফেরত আনার দরকার হয় যাকিহু রেজাল্ট অপশনের একটি টিনেদর্শক করে, আপনি ০ এর চেয়ে বেশী যে কোন ভ্যালুতে রেজাল্টকে ক্যাড সেট করতে পারেন। যদি আপনি একটি ইনটিজার ডেলিভারি দিতে রেজাল্টকে ক্যাড ব্যবহার করেন এবং আপনার Intent অন্তর্ভুক্ত করার কোন দরকার না থাকে, আপনি setResult() কল করতে পারেন এবং শুধু একটি রেজাল্টকে ক্যাড পাস করতে পারেন। উদাহরণস্বরূপ:

```
setResult(RESULT_COLOR_RED);  
finish();
```

এক্ষেত্রে, এখানে শূন্য অল্পিকিহু সম্ভাব্য রেজাল্ট হতে পারে, ফলে রেজাল্টকে ক্যাডটি হেচু একটি ক্যাছাকাছি সূচনিতদর্শক ইনটিজার (০ এর চেয়ে বেশী)। এটা ভালোভাবে কাজ করে যখন আপনার নিজস্ব অ্যাপে একটি টিভিটিতে রেজাল্ট ফেরত আনছেন, কারণ একটি টিভিটি টায়েট রেজাল্ট রিসিভ করেছে রেজাল্টকে ডের ভ্যালু নিখরারন করতে পাবলিক কনস্ট্যান্টকে রেফারেন্স করতে পারে।

নোট: আপনার একটি টিভিটিক startActivity() নাকি startActivityForResult() দিয়ে চলছে তা চেক পরে যাঁজন নেই। শূন্য setResult() কল করুন যদি ইনটেন্ট যা আপনার একটি টিভিটি শুরুর করেছে একটা রেজাল্ট আশা করতে পারে। যদি পরামিতিকভাবে শুরু করা একটি টিভিটি startActivityForResult() কল করে থাকে, তখন সিস্টেমটি এটা ডেলিভারি দেয়, রেজাল্ট যেটা আপনি setResult() এ সাপ্লাই করেছে; অন্যথায় রেজাল্ট উৎপিক্ষত হয়।

কন্টেইনার টেমপ্লেট শেয়ারিং সহ অ্যাপস তৈরী করুন

(<http://developer.android.com/training/building-content-sharing.html>)

এই অনুশীলনটি আপনাকে শেখাবে কীভাবে একটি অ্যাপস তৈরী করা যায় যা অ্যাপস এবং ডিভাইসের মধ্যে ডাটাবে শেয়ার করবে।

১. সাধারণ ডাটাবে শেয়ার করা

অন্য অ্যাপের সাথে তথ্য আদান প্রদান করার মাধ্যমে আপনার অ্যাপকে পরবর্তীতে ভেঙে কীভাবে যোগাযোগ করতে হয়, কীভাবে ফিরতি তথ্য গ্রহণ করতে হয়, এবং কীভাবে ইউজার কন্টেইনার টেমপ্লেট দিয়ে শেয়ার একশন সম্পাদন করতে একটি সহজ এবং মাপযোগ্য উপায় সরবরাহ করা যায়।

১. অন্য অ্যাপ সাধারণ ডাটাবে সন্ড করা
২. অন্য অ্যাপ থেকে সাধারণ ডাটারিভিসিভ করা
৩. সহজ শেয়ার একশন এয়ড করুন

২. ফাইল শেয়ার করা

একটি কন্টেইনার ইউআরআই এবং অস্থায়ী পর্বেশ অনুমতি বয়বহার করে আপনার অ্যাপের সাথে যুক্ত ফাইলে কীভাবে একটি নিরাপদ পর্বেশ সরবরাহ করা যায়।

১. ফাইল শেয়ার করা সেটআপ করা
২. একটি ফাইল শেয়ার
৩. শেয়ার করা ফাইল রিকোয়েস্ট করা
৪. ফাইল তথ্য পুনরুদ্ধার

৩. NFC দিয়ে ফাইল শেয়ার

NFC অ্যান্ডরয়েড বিম ফিচার বয়বহার করে কীভাবে ডিভাইসের মধ্যে ফাইল স্থানান্তর করা যায়।

১. অন্য ডিভাইসে ফাইল সন্ড করা
২. অন্য ডিভাইস থেকে ফাইল গ্রহণ

সাধারণ ডাটাবেশ করার

(<http://developer.android.com/training/sharing/index.html>)

অ্যান্ড্রয়েড অ্যাপ্লিকেশনের একটি বড় বৈশিষ্ট্য হচ্ছে তাদের এক অপরের সাথে যোগাযোগ করতে পারা এবং একীভূত হওয়ার ক্ষমতা। কেন একটা কার্যকরতার পুনরুদ্ভাবন করবেন যা আপনার অ্যাপ্লিকেশনের মধ্যে নেই যখন এটা অন্য একটা অ্যাপ্লিকেশনের মধ্যে ইতিমধ্যে বিদ্যমান রয়েছে?

এই ক্লাস Intent এপিআই এবং ActionProvider অবজেক্ট ব্যবহার করে অ্যাপ্লিকেশনের মধ্যে সাধারণ ডাটাবেসে সন্ড এবং রিসিভ করার কিছু পদ্ধতি উপায় অন্তর্ভুক্ত করে।

অনুশীলনীসমূহ

অনয় অয় াপে সাধারণ ডাটা সেন্ড করা

শিখুন ইন্টেন্ট দিয়ে অনয় অয় াপিলেকশনে টেক্সট এবং বাইনারি সেন্ড করতে সমর্থ করতে কীভাবে আপনার অয় াপিলেকশন সেট করতে হয়।

অনয় অয় াপ থেকে সাধারণ ডাটারি সিস্ত করা

শিখুন ইন্টেন্ট থেকে টেক্সট এবং বাইনারী ডাটারি সিস্ত করতে কীভাবে আপনার অয় াপিলেকশন সেট করতে হয়।

সহজে শয়ার একশন এয় াড করুন

শিখুন আপনার একশন বারের কীভাবে একটি “শয়ার” একশন আইটেম এয় াড করতে হয়।

অনয় অয় াপে সাধারণ ডাটাসেন্ড করা

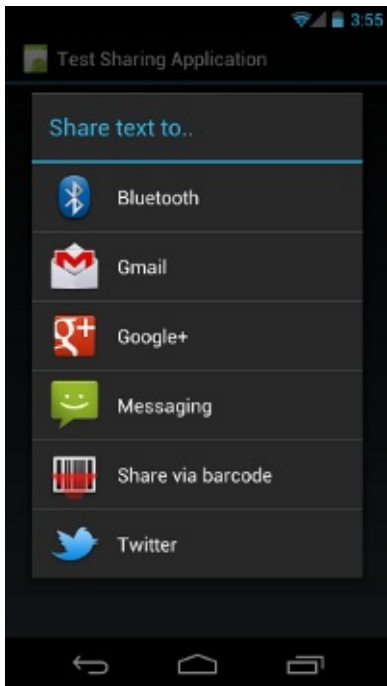
(<http://developer.android.com/training/sharing/send.html>)

যখন আপনি একটি ইন্টেন্টিগঠন করেন, আপনাকে অবশ্যই একশনকে সূচনাদর্শক করতে হবে আপনি যদি ইন্টেন্টিটি সিকর্য করতে চান ("trigger")। অয়ান্ডরয়েডকে একটি একশনকে নিধরারণ করে, যার মধ্যে রয়েছে ACTION_SEND যা নিদর্শক করে যে ইন্টেন্টি একটি টিভি থেকে আরেকটিতে ডাটাসেন্ড করেছে। অনয় একটি টিভি থেকে ডাটাসেন্ড করতে, আপনাদের সকলকে ডাটাসেন্ড এবং এর টাইপকে সূচনাদর্শক করার দরকার, সিস্টেমসবর্গর্হী রিসিসিভ একটি টিভি নিধরারণ করে এবং ইউজারের কাছে তা পর্দাশরন করে (যদি সেখানে বহুমুখি অপশন থাকে) অথবা তা ক্রমিকভাবে শুরুর করে (যদি শুধু একটি অপশন থাকে)। একইভাবে, আপনি ডাটাসেন্ড করার পরে পেরন যে আপনার একটি টিভি আপনার মেসেজিস্টেট তাদের সূচনাদর্শক করার মাধ্যমে অনয় অয় আপ থেকে রিসিসিভ করাকে সাপোর্ট করে।

ইন্টেন্টি দিয়ে অয় আপিলেকশনের মধ্যে ডাটাসেন্ড এবং রিসিসিভ করাটা হচ্ছে কন্টেন্টি এর সামাজিক শয়ারিং এর জন্য সবচেয়ে পর্চলিত বয়বহার। ইন্টেন্টি ইউজারকে দরুত এবং সহজে তথ্য শয়ার করতে দেয়, তাদের পছন্দের অয় আপিলেকশন বয়বহার করে।

নোট: একটি ActionBar এ একটি শয়ার একশন আইটেম যুক্ত করার সবচেয়ে ভালো উপায় হচ্ছে ShareActionProvider বয়বহার করা যা এপিআইলেভল ১৪তে পাওয়া যাবে। Adding an Easy Share Action সম্পর্কিত অনুলীলনীতে ShareActionProvider আলোচিত হয়েছে।

টেক্সট কন্টেন্ট সেন্ড করা



ফিগার ১. একটি হ্যান্ডডেস্ক ACTION SEND ইন্টেন্ট চুজারের একটি স্ক্রিনশট।

ACTION SEND একশনের সবচেয়ে সাজাসাপ্টা এবং সবচেয়ে বেশী ব্যবহার হচ্ছে একটি টিভি থেক আরেকটি টেটেক্সট কন্টেন্ট সেন্ড করা। উদাহরণস্বরূপ, বিল্ট-ইন বর্ডার অ্যাপ থেকে কোন অ্যাপলিকেশন দিয়ে টেক্সটটি হিসাবে চলিত-পর্দা শর্ত পেজের ইউআরএল শেয়ার করতে পারে। এটা ইমইল বা সামাজিক যোগাযোগের মাধ্যমে বন্ধুদের সাথে একটি আর্টিকেল বা ওয়েবসাইট শেয়ার করার জন্য উপকারী। এই ধরনের শেয়ার বাস্তুায়ন করতে এখন কেঁকাডটি আছে:

```
Intent sendIntent = new Intent();
sendIntent.setAction(Intent.ACTION_SEND);
sendIntent.putExtra(Intent.EXTRA_TEXT, "This is my text to send.");
sendIntent.setType("text/plain");
startActivity(sendIntent);
```

যদি একটি ফিল্টার সহ একটি ইনস্টল করা অ্যাপলিকেশন থাকে যা ACTION SEND এবং MIME টাইপ টেক্সট/প্লেন টেক্সট ম্যাসেজের, অ্যান্ড্রয়েড সিস্টেম এটারান করে; যদি একে অধিক অ্যাপলিকেশন ম্যাসেজের, সিস্টেম একাধারে বেঁধে ডায়ালগ (একটি চুজার) পর্দা শর্ত করে যা ইউজারকে একটি অ্যাপ পছন্দ করতে দেয়। যদি আপনি ইন্টেন্টের জন্য Intent.createChooser() কল করেন, অ্যান্ড্রয়েড সবসময় চুজারটি পর্দা শর্ত করে। এটার কিছু সুবিধা আছে:

- এমনকি যদি ইউজার পূর্বেই ইন্টেন্টের জন্য একটি ডিফল্ট একশন নির্ধারণ করে, চুজার তখনও পর্দা শর্ত করে।
- যদি কোন অ্যাপলিকেশন ম্যাসেজ না করে, অ্যান্ড্রয়েড একটি সিস্টেম ইমেজ পর্দা শর্ত করে।
- আপনি চুজার ডায়ালগের জন্য একটি টাইটেল নির্ধারণ করতে পারেন।

এটা একটি আপডেট কেঁকাড:

```
Intent sendIntent = new Intent();
sendIntent.setAction(Intent.ACTION_SEND);
sendIntent.putExtra(Intent.EXTRA_TEXT, "This is my text to send.");
sendIntent.setType("text/plain");
startActivity(Intent.createChooser(sendIntent, getResources().getText(R.string.send_to)));
```

ফলাফল ডায়ালগ ফিগার ১ এ দেখানো হয়েছে।

এটি চুঁইকভাবে, আপনি ইন্টেন্টির জন্য কীকছু স্ট্যান্ডার্ড এক্সটেন্সিভ স্ট্রিংস সেট করতে পারেন: EXTRA_EMAIL, EXTRA_CC, EXTRA_BCC, EXTRA_SUBJECT। কিন্ত যদি রিসিভ করা অ্যাপলিকেশন তাদের ব্যবহার করার মতো করে ডিজাইন করা না থাকে, তাহলে কীকছু ইন্টেন্সিভ না। আপনি কাস্টম এক্সটেন্সিভ ব্যবহার করতে পারেন, কিন্ত এখানে কোন পরেভার্স না যদি না রিসিভ করা অ্যাপলিকেশন এটা বুঝতে না পারে। সব্ভাব্য কভাবে আপনি রিসিভ করা অ্যাপলিকেশনের নিজেসব ধারানিধরারিত কাস্টম এক্সটেন্সিভ ব্যবহার করতে পারেন।

নোট: কীকছু ই-মেইল অ্যাপলিকেশন, যেমন জিমেইল, এক্সটেন্সিভ এর জন্য EXTRA_EMAIL এবং EXTRA_CC এর মতো একটি String[] এর পরে তথ্য শা করে, আপনার ইন্টেন্টি এটা যুক্ত করতে putExtra (String, String[]) ব্যবহার করে।

বাইনারি কন্টেন্ট সেন্ড করুন

বাইনারি ডাটাবেশ করার হয় যথাযথ MIME টাইপ সেটিং দ্বারা একটি রত ACTION SEND একশন ব্যবহার করে এবং EXTRA_STREAM নামে একটি এক্সট্রার মধ্যে ডাটাবেশ ইউআরআই সিন্বেশ করে। এটা সাধারণত একটি ইমেজ শেয়ার করার কাজে ব্যবহার করা হয় কিন্ত যেকোন ধরনের বাইনারি কন্টেন্ট শেয়ার করতে ব্যবহার করা যেতে পারে:

```
Intent shareIntent = new Intent();
shareIntent.setAction(Intent.ACTION_SEND);
shareIntent.putExtra(Intent.EXTRA_STREAM, uriToImage);
shareIntent.setType("image/jpeg");
startActivity(Intent.createChooser(shareIntent, getResources().getText(R.string.send_to)));
```

নিম্নে নাক্ত বিষয়গুলো লেনাট করুন:

- আপনি "/" এর MIME টাইপ ব্যবহার করতে পারেন, কিন্ত এটা শুধু একটি টিভি টি ম্যাচ করে যা জেনেরিক ডাটাসিস্টার্ম পরিচালনা করতে সক্ষম।
- রিসিভ করা অ্যাপের ডাটাবেশ পরবেশ করতে অনুমতি দরকার যা Uri নিদের্শ করে। এটা করার পরস্ তাবিত উপায় হচ্ছে:
 - আপনার নিজস্ব ContentProvider এ ডাটাবেস্টার করুন, এটা নিশ্চিত করতে যে আপনার পর্ভাইডার অন্য অ্যাপের পরবেশের সঠিক অনুমতি আছে। পরবেশ গম্যতা পর্দানের পছন্দসই মেকানিজম হচ্ছে per-URI permissions ব্যবহার করা যা অস্থায়ী এবং শুধুমাত্র রিসিভ করা অ্যাপিলেকশনে পরবেশ গম্যতা কে অনুমোদন করে। এটার মতো করে একটি ContentProvider তৈরী করার একটা সহজ উপায় হচ্ছে FileProvider হেলপার ক্লাস ব্যবহার করা।
 - সিস্টেম MediaStore ব্যবহার করুন। MediaStore পরাথমিকভাবে বিডিও, অডিও এবং ইমেজ MIME টাইপ লক্ষ্য করে করা, কিন্ত অ্যান্ডরয়েড ৩.০ (এপিআই লেভেল ১১) দিয়ে শুরু করলে এটা নন-মিডিয়া টাইপ (আরও জানতে দেখুন MediaStore.Files) সেষ্টার করতে পারে। ফাইল scanFile() ব্যবহার করে MediaStore এ পরবেশ করতে পারে, যেটার পরে শেয়ারের জন্য উপযুক্ত একটি content:// স্টাইল Uri কে পর্ভাইডকৃত onScanCompleted() কলব্যাকে পাস করে দেয়। উল্লেখ্য যে একবার সিস্টেম MediaStore যুক্ত হলে কন্টেন্ট টিভিডিওই সরে যে কোন অ্যাপে পরবেশ গম্য হবে।

কন্টেইনার এমালি টপল অংশ সেন্ড করুন

কন্টেইনার এমালি টপল অংশ শেয়ার করতে, কন্টেইনার টিনিদের শেয়ার করা ইউআরআই (URIs) এর একটি তালিকা সহ ACTION_SEND_MULTIPLE একশন ব্যবহার করুন। কন্টেইনার রিমশরণ যা আপনি ব্যবহার করেছেন সেই অনুসারে MIME টাইপের তারতম্য হয়। উদাহরণস্বরূপ, যদি আপনি JPEG ইমেজ শেয়ার করতে চান, টাইপ তখনও "image/jpeg"। একটি ইমেজ টাইপের মিস্ট্রের জন্য, একটি একটি টিভিটি যাতে কোন ধরনের ইমেজ পরিচালিত করে এই একটি টিভিটি ম্যাচ করতে এটা [image/*] হওয়া উচিত।

আপনার শুধুমাত্র [/*] ব্যবহার করা উচিত যদি আপনি অনেকগুলো বিভিন্ন টাইপ শেয়ার করে থাকেন। যেমনভাবে প্লেবের বর্ণনা করা হয়েছে, আপনার ডাটাবেকিবেশ লষণ এবং পরিকর্যা করাটা, রিসিভ করা অ্যাপিলেকশনের উপর বর্তায়। উদাহরণস্বরূপ:

```
ArrayList<Uri> imageUris = new ArrayList<Uri>();
imageUris.add(imageUri1); // Add your image URIs here
imageUris.add(imageUri2);

Intent shareIntent = new Intent();
shareIntent.setAction(Intent.ACTION_SEND_MULTIPLE);
shareIntent.putParcelableArrayListExtra(Intent.EXTRA_STREAM, imageUris);
shareIntent.setType("image/*");
startActivity(Intent.createChooser(shareIntent, "Share images to.."));
```

প্লেবের মতো, নিশ্চিত করুন যে পরদত্ত URIs ডাটার পরিতিনিদের শেয়ার করতে একটি রিসিভ করা অ্যাপিলেকশন পরবেশগম্য হতে পারে।

অনয় অয় াপ থে কে সাধ ারন ড াট া রিসিভ করা

(<http://developer.android.com/training/sharing/receive.html>)

ঠিক যেমন আপনার অয় াপ লি কশন অনয় অয় াপ লি কশনে ড াট া সেন্ড করতে পারে, তেমনিভাবে এটা সহজেই অনয় অয় াপ লি কশন থেকে পাঠানো ড াট া রিসিভও করতে পারে। কীভাবে ইউজার আপনার অয় াপ লি কশনের সাথে পারস্পরিক ক্রিয়া করবে এবং অনয় অয় াপ লি কশন থেকে ক্রিয়াকর্মের ড াট া আপন রিসিভ করতে পারেন সে সম্পর্কে রচনা করুন। উদাহরণস্বরূপ, একটি সামাজিক যোগাযোগের অয় াপ লি কশন অনয় অয় াপ লি কশন থেকে টেক্সট কন্টেন্ট, একটাকে কীভাবে লোড করবেন তা দেখাব ইউআরএল রিসিভ করতে আগ্রহী হতে পারে। গুগল প্লাস অয় ান ডায়েরিতে অয় াপ লি কশন (Google+ Android application: <https://play.google.com/store/apps/details?id=com.google.android.apps.plus>) টেক্সট এবং একক বা বহুমুখি ইমেজ উভয়ই গ্রহণ করে।

আপনার মেসেজস্ট আপডেট করা

ইন্টেন্ট ফিল্টার সিস্টমকে অবহিত করে যে একটি ইন্টেন্ট অ্যাপ্লিকেশনকে ম্যাপায়েনন্ট গর্হণ করতে ইচ্ছুক। একইভাবে অন্য অ্যাপ সাধারণ ডাটাসেন্ড করুন (Sending Simple Data to Other Apps) অনুশীলনের মতো ACTION SEND দিয়ে কীভাবে আপনি একটি ইন্টেন্ট তৈরী করেন, এই একশন দিয়ে ইন্টেন্ট গর্হণ করতে পারার জন্য আপনি ইন্টেন্ট ফিল্টার তৈরী করেন। <intent-filter> এ লিমেন্ট বয়বহার করে আপনি আপনার মেসেজস্ট একটি ইন্টেন্ট ফিল্টার নিধরারণ করেন। উদাহরণস্বরূপ, যদি আপনার অ্যাপ্লিকেশন টেক্সট কন্টেন্ট রিসিভ করা চালিত করে, যে কোন ধরনের একটি একক ইমেজ বা যে কোন ধরনের বহুমুখি ইমেজ, আপনার মেসেজস্ট এরকম দেখাবে:

```
<activity android:name=".ui.MyActivity" >
  <intent-filter>
    <action android:name="android.intent.action.SEND" />
    <category android:name="android.intent.category.DEFAULT" />
    <data android:mimeType="image/*" />
  </intent-filter>
  <intent-filter>
    <action android:name="android.intent.action.SEND" />
    <category android:name="android.intent.category.DEFAULT" />
    <data android:mimeType="text/plain" />
  </intent-filter>
  <intent-filter>
    <action android:name="android.intent.action.SEND_MULTIPLE" />
    <category android:name="android.intent.category.DEFAULT" />
    <data android:mimeType="image/*" />
  </intent-filter>
</activity>
```

নোট: ইন্টেন্ট ফিল্টার এবং ইন্টেন্ট রেজুলেশনের সম্পর্ক আরও জানতে Intents and Intent Filters (<http://developer.android.com/guide/components/intents-filters.html#ifs>) দেখুন।

যখন অন্য অ্যাপ্লিকেশন একটি ইন্টেন্ট গঠন করে এবং startActivity() র পর্তি এটা পাস করার মাধ্যমে এই বিষয়গুলো লার কৈকানিক ছু শেয়ার করার চেষ্টা করে, আপনার অ্যাপ্লিকেশন ইন্টেন্ট চুজারের মধ্যে একটি অপশন হিসাবে তালিকাভুক্ত হবে। যদি ইউজার আপনার অ্যাপ্লিকেশন বেছে নেয়, সন্শিল্ষ্ট একটি টিভিটি (উপরে রক্ত উদাহরনের মধ্যে .ui.MyActivity) শুরু হবে। তারপর আপনার কেড এবং ইউজার ইন্টারফেসের মধ্যে থেক কন্টেন্ট যথায় থাবে চালিত করা আপনার উপর নিভর করে।

ইনকামিং কন্টেন্ট বস্‌থাপনা করা

একটি Intent দ্বারা ডেলিভারি দেওয়া কন্টেন্ট চালনা করতে, Intent অবজেক্ট পেতে getIntent() কল করে শুরু করুন। আপনার যদি পূর্বে বর্হি অবজেক্টটি থেকে থাকে, পরবর্তীতে তিক করতে হবে তা নিবর্তারগ করতে এটার কন্টেন্ট পরীক্ষা করতে পারেন। মনে রাখবেন যে যদি এই একটি ভিটিটি সিস্টেমটির অনয অংশ থেকে শুরু করতে পারে, যেমন লঞ্চার, তখন এটাকে আপনার বিবেচনায় আনার দরকার হতে পারে যখন ইনটেন্টটি পরীক্ষা করা হয়।

```
void onCreate (Bundle savedInstanceState) {  
    ...  
    // Get intent, action and MIME type  
    Intent intent = getIntent();  
    String action = intent.getAction();  
    String type = intent.getType();  
  
    if (Intent.ACTION_SEND.equals(action) && type != null) {  
        if ("text/plain".equals(type)) {  
            handleSendText(intent); // Handle text being sent  
        } else if (type.startsWith("image/")) {  
            handleSendImage(intent); // Handle single image being sent  
        }  
    } else if (Intent.ACTION_SEND_MULTIPLE.equals(action) && type != null) {  
        if (type.startsWith("image/")) {  
            handleSendMultipleImages(intent); // Handle multiple images being sent  
        }  
    } else {  
        // Handle other intents, such as being started from the home screen  
    }  
    ...  
}  
  
void handleSendText(Intent intent) {  
    String sharedText = intent.getStringExtra(Intent.EXTRA_TEXT);  
    if (sharedText != null) {  
        // Update UI to reflect text being shared  
    }  
}  
  
void handleSendImage(Intent intent) {  
    Uri imageUri = (Uri) intent.getParcelableExtra(Intent.EXTRA_STREAM);  
    if (imageUri != null) {  
        // Update UI to reflect image being shared  
    }  
}  
  
void handleSendMultipleImages(Intent intent) {  
    ArrayList<Uri> imageUris = intent.getParcelableArrayListExtra(Intent.EXTRA_STREAM);  
    if (imageUris != null) {  
        // Update UI to reflect multiple images being shared  
    }  
}
```

সতর্কতা: ইনকামিং ডাটাকে চক করতে অতিরিক্ত যত্নেবান হোন, আপনি কখনই জানতে পারবেন না অনয অ্যাপলিকেশন আপনাকে কিসে সন্ড করতে পারে। উদাহরণসব্বরূপ, ভুল MIME টাইপ সেট হতে পারে, অথবা অতিরিক্ত বড় আকারে ইমেজ সেট হতে পারে। পর্ধান ("UI") থের্ডের মধ্যে করার চেয়ে একটি আলাদা থের্ডের মধ্যে বাইনারি ডাটাপর্সেসস করার বিষয়টাও মনে রাখবেন।

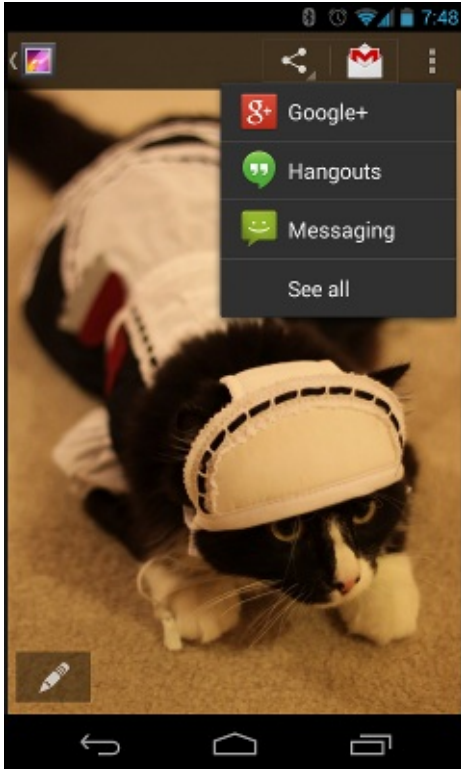
ইউজার ইন্টারফেস আপডেট করাটা একটি EditText পূরণ করার মতোই সহজ হতে পারে, অথবা এটা একটি ইমেজ একটা আকর্ষনীয় ফিটাইফল্টার পর্য়োগ করার মতো আরও জটিলও হতে পারে। এটা আসলেই আপনার অ্যাপলিকেশনে সুনিদিষ্ট টেপ পরবর্তীতে তিক হবে।

একটি সহজ শেয়ার একশন যুক্ত করা

(<http://developer.android.com/training/sharing/shareaction.html>)

অ্যান্ড্রয়েড ৪.০ (এপিআই লেভেল ১৪) এর মধ্যে `ActionProvider` এর পর্বতরন দ্বারা আপনার `ActionBar` এর মধ্যে কার্যকরী এবং ইউজার বান্ধব শেয়ার একশন বাস্তবায়ন করাটা এমনকি আরও সহজ করে তৈরী। একটি `ActionProvider`, একবার একশনবারের মধ্যে মেনু আইটেম স্যুজ্ঞিত হলে, ওই আইটেমের উপস্থিতি এবং আচরণ উভয়েই চালনা করে। `ShareActionProvider` এর ক্ষেত্রে, আপনি একটা শেয়ার ইন্টেন্ট সরবরাহ করেন এবং এটা বাকীটাকরে।

নোট: এপিআই লেভেল ১৪ এবং এর চেয়ে উন্নত সংস্করণ দিয়ে শুরু করা কাজে `ShareActionProvider` বিদ্যমান আছে।



ফিগার ১. গ্যালারি অ্যাপে `ShareActionProvider`।

মেনু ডিকে ক্লয়ারে শন আপডেট করা

ShareActionProviders দিয়ে শুরু করতে, আপনার menu resource ফাইলের মধ্যে
সংশ্লিষ্ট <item> এর জন্য android:actionProviderClass এটি রিবিউটিনিং ধারণ করুন:

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item
    android:id="@+id/menu_item_share"
    android:showAsAction="ifRoom"
    android:title="Share"
    android:actionProviderClass=
      "android.widget.ShareActionProvider" />
  ...
</menu>
```

এটা ShareActionProvider এর প্রতি আইটেমের উপস্থিতি এবং ক্যাবলীর
দায়িত্ব হস্তান্তর করে। কিন্তু আপনার সরবরাহকারীকে বলা দরকার যে
আপনি কী শয়্যার করতে চান।

শেয়ার ইন্টেন্ট সেট করা

ShareActionProvider কে কাজ করতে, আপনাকে অবশ্যই এটাকে একটি শেয়ার ইন্টেন্ট সরবরাহ করতে হবে। EXTRA_TEXT এবং EXTRA_STREAM এর মতো এক্সট্রার সাহায্যে একশন ACTION_SEND এবং অতিরিক্ত ডাটাসেটি দিয়ে (Sending Simple Data to Other Apps) অনুশীলনীতে যেভাবে আলোচনা করা হয়েছে, এই শেয়ার ইন্টেন্টের সরকম হওয়া উচিত। একটি শেয়ার ইন্টেন্টকে নিয়ুক্ত করতে, পর্ত্বেম সংশ্লষ্ট MenuItem খুঁজে বের করুন যখন আপনার Activity বা Fragment এর মধ্যে আপনার মেনুটির সাসর্ পর্সারিত করে। পরবর্তীতে ShareActionProvider এর একটি ইনস্ট্যান্স উদ্ধার করতে MenuItem.getActionProvider() কল করুন। ওই একশন আইটেমের সাথে যুক্ত শেয়ার আইটেম আপডেট করতে setShareIntent() ব্যবহার করুন। এখানে একটি উদাহরণ আছে:

```
private ShareActionProvider mShareActionProvider;
...

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate menu resource file.
    getLayoutInflater().inflate(R.menu.share_menu, menu);

    // Locate MenuItem with ShareActionProvider
    MenuItem item = menu.findItem(R.id.menu_item_share);

    // Fetch and store ShareActionProvider
    mShareActionProvider = (ShareActionProvider) item.getActionProvider();

    // Return true to display menu
    return true;
}

// Call to update the share intent
private void setShareIntent(Intent shareIntent) {
    if (mShareActionProvider != null) {
        mShareActionProvider.setShareIntent(shareIntent);
    }
}
```

আপনার শুধু আপনার মেনু তৈরীর সময় একবার শেয়ার ইন্টেন্ট সেট করার পরে যাজন হতে পারে, অথবা আপনি এটাসেট করতে চাইতে পারেন এবং ইউজার ইন্টারফেস পরিবর্তন হিসাবে এটাকে আপডেট করুন। উদাহরণস্বরূপ, যখন আপনি গ্যালারী অ্যাপে ফটো পূর্ণরিস্ক করেনিভুক্ত করেন, শেয়ারিং ইন্টেন্ট পরিবর্তন করে যাতে আপনি ফটোর মধ্যে যুক্ত হতে পারেন।

ShareActionProvider অবজেক্ট সম্পর্কিত আরও আলোচনার জন্য ActionBar গাইড দেখুন।

ফাইল শেয়ার করা

(<http://developer.android.com/training/secure-file-sharing/index.html>)

অ্যাপের পরর্তি তিনয়ত অনয্য অ্যাপে তােদর এক বা একাধিক ফাইল দেয়ার পর স্তাব করার দরকার হয়। উদাহরণস্বরূপ, একটি ইমেজ গ্যালারি ইমেজ এডিটরে ফাইল পাঠানোর পর স্তাব করতে পারে, অথবা একটি ফাইল ব্যবস্থাপনা অ্যাপ এক স্ট্যান্ডার্ড লেস্টারে জের এলাকার মধ্যে ইউজারকে ফাইল কপি এবং সেস্ট করতে দিতে চাইতে পারে। একমুখি রাস্তা একটি সেমিন্ড অ্যাপ একটি ফাইল শেয়ার করতে পারে সেটা হচ্ছে পরদত্তিরিকে যাে স্টর পর্তি রিসিভ অ্যাপ থেকে সাড়া পরদান।

সকলক্ষেত্রে আপনার অ্যাপ থেকে অনয্য অ্যাপে ফাইল পাঠানোর পর স্তাবে একমাত্র নিরাপদ উপায় হচ্ছে রিসিভ অ্যাপে ফাইলের কন্টেন্ট ইউআরআই পাঠানো এবং ইউআরআই অস্থায়ী পরবেশ অনুমতি দেওয়া। ইউআরআই অস্থায়ী পরবেশ অনুমতি সহ কন্টেন্ট ইউআরআই নিরাপদ কারন তার শ্রুধু অ্যাপে পরে যাগ করে যা ইউআরআই রিসিভ করে, এবং তার ব্যবহার যত্নে বশেষ হয়। একটি ফাইলের কন্টেন্ট ইউআরআই উপাদান করার জন্য অ্যান্ডরয়েড FileProvider কমেপানেন্টে মেথড `getUriForFile()` সরবরাহ করে।

আপনি যদি অ্যাপে সরমধ্যে অল্প পরিমানে টেক্সট বা সংখ্যাবাচক (নিউমেরিক) ডাটাকে শেয়ার করতে চান, আপনার উচিত একটা `Intent` সেন্ড করা যা ডাটাকে ধারণ করে। `Intent` দিয়ে কীভাবে সাধারণ ডাটাকে সেন্ড করা যায় তা শিখতে (Sharing Simple Data) ক্লাসটি দেখুন।

এই অনুশীলনী ব্যবখ্যাকরেব অ্যান্ডরয়েড FileProvider কমেপানেন্ট কতরকউ পািদত কন্টেন্ট ইউআরআই ব্যবহার করে আপনার অ্যাপ থেকে অনয্য অ্যাপে কীভাবে নিরাপদে ফাইল শেয়ার করা যায় এবং অস্থায়ী অনুমতি যা কন্টেন্ট ইউআরআই এর জন্য রিসিভ অ্যাপকে দেয়া হয়।

অনুশীলনীসমূহ

ফাইল শেয়ার করতে অ্যাপ সেটআপ করা

শিখুন ফাইল শেয়ার করতে কীভাবে আপনার অ্যাপ সেটআপ দিতে হয়

একটি ফাইল শেয়ার করা

শিখুন ফাইলের জন্য একটি কন্টেন্ট ইউআরআই উপাদান করার মাধ্যমে কীভাবে একটি ফাইল অন্য অ্যাপে পাঠানোর পর্স্বতাব করা হয়।

শেয়ার করা ফাইল রিকোয়েস্ট করা

শিখুন কীভাবে অন্য অ্যাপ কতৃক শেয়ার করা একটি ফাইলকে রিকোয়েস্ট করতে হয়, ফাইলের জন্য কন্টেন্ট ইউআরআই রিসিভ করতে হয় এবং ফাইলটি ওপেন করতে কন্টেন্ট ইউআরআই ব্যবহার করতে হয়।

ফাইল তথ্য পুনরুদ্ধার করা

শিখুন কীভাবে একটি অ্যাপ একটি FileProvider দ্বারা উপাদিত কন্টেন্ট ইউআরআই ব্যবহার করে ফাইল তথ্য পুনরুদ্ধার করে যার মধ্যে MIME টাইপ এবং ফাইল সাইজ অন্তর্ভুক্ত।

ফাইল শেয়ার করতে অ্যাপ সেট আপ করা

(<http://developer.android.com/training/secure-file-sharing/setup-sharing.html>)

আপনার অ্যাপ থেকে অন্য অ্যাপ একটি ফাইল নিরাপদে পাঠানোর পরিস্থিতিতে, কন্টেন্টের ইউআরআই এর ফর্মের ফাইলের পরিত একটি নিরাপদ বস্তু স্থাপনা পরিস্থিতিতে আপনাকে আপনার অ্যাপকে কনফিগারেশন করতে হবে। অ্যান্ড্রয়েড FileProvider কন্ট্রোল ফাইলের জন্য কন্টেন্ট ইউআরআই উপাদান করে, আপনার দ্বারা এক্সএমএল এর পরিস্থিতিতে সিসিফিকেশন উপরিত্ত করে। এই অনুশীলনী আপনাকে দেখাবে কীভাবে আপনার অ্যাপ FileProvider এর ডিফল্ট বাসস্থান এড করতে হয় এবং আপনার অ্যাপ থেকে অন্য অ্যাপ যে ফাইলটি পরিস্থিতিতে চান তা কীভাবে সন্নিবিষ্ট করতে হয়।

নোট: FileProvider ক্লাসটি v4 Support Library এর অংশ। আপনার অ্যাপের মধ্যে এই লাইব্রেরী অন্তর্ভুক্ত করা সম্পর্কিত তথ্যের জন্য (Support Library Setup) দেখুন।

ফাইল পর্ভাইডার (পর্দানকারী) নির্দিষ্ট করুন

আপনার অ্যাপের জন্য একটি FileProvider নির্ধারণ করা আপনার মেনিফেস্টের মধ্যে একটি এন্ট্রি চায়। এই এন্ট্রি উপস্থাপন করতে ইউআরআই ব্যবহার করতে অথরিটি নির্দিষ্ট করে, একইভাবে একটি এক্সএমএল ফাইলের নাম যা আপনার অ্যাপ শেয়ার করতে পারা ডিরেক্টরী সুনির্দিষ্ট করে।

নিম্নোক্ত চিত্রটি আপনাকে দেখাবে কীভাবে আপনার মেনিফেস্ট <provider> এন্ট্রি এ যুক্ত করতে হয় যা FileProvider ক্লাস, অথরিটি, এবং এক্সএমএল ফাইল নাম সুনির্দিষ্ট করে:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.myapplication">
    <application
        ...>
        <provider
            android:name="android.support.v4.content.FileProvider"
            android:authorities="com.example.myapplication.fileprovider"
            android:grantUriPermissions="true"
            android:exported="false">
            <meta-data
                android:name="android.support.FILE_PROVIDER_PATHS"
                android:resource="@xml/filepaths" />
        </provider>
        ...
    </application>
</manifest>
```

এই উদাহরণে, android:authorities এন্ট্রি বিউট ইউআরআই অথরিটি সুনির্দিষ্ট করে যা আপনি FileProvider কন্ট্রোল উপস্থাপন করতে ইউআরআই এর জন্য ব্যবহার করতে চাইতে পারেন। এই উদাহরণে, অথরিটি হচ্ছে com.example.myapplication.fileprovider। আপনার নিজস্ব অ্যাপের জন্য, অ্যাপের android:package ভ্যালুর অংশীভূত অথরিটি সুনির্দিষ্ট করা সাথে এন্ট্রিতে স্ট্রিং "fileprovider" যুক্ত করে। অথরিটি ভ্যালু সম্বন্ধে আরও জানতে টিপক (Content URIs) এবং android:authorities এন্ট্রি বিউটের জন্য ডকুমেন্টেশন দেখুন।

<provider> এর <meta-data> চাইল্ড এন্ট্রি এক্সএমএল ফাইলের পর্ভিত নির্দেশ করে যা আপনার শেয়ার করতে চাওয়া ডিরেক্টরী নির্দিষ্ট করে। android:resource এন্ট্রি বিউট .osis এক্সটেনশন ব্যতীত ফাইলের নাম এবং পথ। এই ফাইলের কন্টেন্ট পরবর্তী শেখেনে আলাচনা করা হবে।

শেয়ারে যোগ য় ডিরেক্টরী নিদির্ষ্টকরণ

পূর্বে বর্ আপনি যদি আপনার মেইনফেস্ট FileProvider যুক্ত করে থাকেন, আপনাকে ডিরেক্টরী নির্দির্ষ্ট করতে হবে যা সেই ফাইলকে ধরে রাখবে যা আপনি শেয়ার করতে চান। ডিরেক্টরী নির্দির্ষ্ট করতে, আপনার পরে জেক্টর res/xml/ সাবডির্ক্টরীর মধ্যে য় ফাইল filepaths.xml তৈরী করার মাধ্যমে শুরু করুন। এই ফাইলের মধ্যে, পরিতর্টি ডিরেক্টরীর জন্য একটি এক্সএমএল এলিমেন্ট যুক্ত করার মাধ্যমে ডিরেক্টরী নির্দির্ষ্ট করুন। নিম্নোক্ত অংশটি আপনাকে res/xml/filepaths.xml এর কন্টেইনার উদাহরণ দেখাবে। এই অংশটি আরও পরামান করে আপনার ইন্টারনাল স্টোরে জের এলাকার মধ্যে য় files/ ডিরেক্টরীর একটি সাবডির্ক্টরী কীভাবে শেয়ার করতে হয়:

```
<paths>
  <files-path path="images/" name="myimages" />
</paths>
```

এই উদাহরণের মধ্যে, <files-path> ট্যাগ আপনার অ্যাপের ইন্টারনাল স্টোরে জের files/ ডিরেক্টরীর মধ্যে য় থেক ডিরেক্টরী শেয়ার করতে পারে। path এটি ট্রিবিউট files/ এর images/ সাবডির্ক্টরী শেয়ার করে। name এটি ট্রিবিউট FileProvider কে files/images/ সাবডির্ক্টরীতে ফাইলের জন্য কন্টেইন্ট ইউআরআই এ পাঠ সেগমেন্ট myimages যুক্ত করতে বলে।

<paths> এলিমেন্টের মািল্টিপল চিলড্রন থাকতে পারে, শেয়ার করতে এর পরিতর্টি ভিন্ন ডিরেক্টরী নির্দির্ষ্ট করে। উপরন্তু এলিমেন্ট আপনি এক্সটানরাল সেটোরে জের ডিরেক্টরী শেয়ার করতে এলিমেন্ট ব্যবহার করতে পারেন। চাইল্ড এলিমেন্ট যা শেয়ার করা ডিরেক্টরী নির্দির্ষ্ট করে সে সম্পর্কে আরও জানতে, FileProvider রেফারেন্স ডকুমেন্টেশন দেখুন।

নোট: এক্সএমএল ফাইল হেচ্ছ আপনি যে ডিরেক্টরী শেয়ার করতে চান তা নির্দির্ষ্ট করার একমাত্র উপায়; আপনি পেরাগর্মেটিকভাবে একটি ডিরেক্টরী যুক্ত করতে পারেন না।

এখন আপনার একটি FileProvider এর একটি পূর্ণ রাঙা গেস্পিসিফিকেশন আছে যা আপনার অ্যাপের ইন্টানরাল সেটোরে জের files/ ডিরেক্টরীর ফাইলের জন্য অথবা files/ এর সাবডির্ক্টরীর ফাইলের জন্য কন্টেইন্ট ইউআরআই উপস্থাপন করে। যখন আপনার অ্যাপ একটি ফাইলের জন্য কন্টেইন্ট ইউআরআই উপস্থাপন করে এটা ফাইলের <provider> এলিমেন্ট (com.example.myapplication.fileprovider), পাথ myimages/, এবং নেম এর মধ্যে নির্দির্ষ্ট অথরিটিটিকে ধারণ করে।

উদাহরণস্বরূপ, এই অনুশীলনের এই চিত্রটি অনুসারে আপনি যদি একটি FileProvider নিধরারন করেন এবং আপনি যদি ফাইল ফবভধঃষঃথরসধমব. লটম এর জন্য একটি কন্টেইন্ট ইউআরআই রিকোয়েস্ট করেন, FileProvider নিম্নোক্ত ইউআরআই ফেরত দেয়:

```
content://com.example.myapplication.fileprovider/myimages/default_image.jpg
```

একটি ফাইল শেয়ার করা

(<http://developer.android.com/training/secure-file-sharing/share-file.html>)

পূর্বে বর্নিত কখনও কন্টেন্ট ইউআরআই ব্যবহার করে ফাইল শেয়ার করতে যদি আপনার অ্যাপ সেটআপ করে থাকেন, ওই ফাইলের জন্য অন্য অ্যাপের পাঠানো রিকোয়েস্টের সপন্থ করতে পারেন। এই রিকোয়েস্টের সপন্থ করার একটি উপায়, সান্তর্কার অ্যাপ থেকে একটি ফাইল সিলেকশন ইন্টারফেস সরবরাহ করা যাতে অন্য অ্যাপিলেকশন আবাহন করতে পারে। এই অ্যাপেরাচ একটি কলোয়ন্ট (ভেক্তা) অ্যাপিলেকশনকে সান্তর্কার অ্যাপ থেকে একটি ফাইল ইউজারকে বাছাই করতে দেওয়ার বিষয়টা অনুমোদন করে এবং তারপর নিবর্তাচিত ফাইলের কন্টেন্ট ইউআরআই রিসভ করে।

এই অনুশীলনী দেখাবে আপনার অ্যাপ কীভাবে একটি ফাইল নিবর্তাচনী Activity তৈরী করতে হয় যা ফাইলের জন্য রিকোয়েস্টের সপন্থ করে।

ফাইল িরেকায়েস্ ট িরিসভ করা

ক্লায়েন্ট অ্যাপ থেকে ফাইলের জন্য িরেকায়েস্ ট গ্রহণ করতে এবং কন্টেন্ট ইউআরআই দিয়ে রেসপন্স করতে, আপনার অ্যাপে উচ্চত একটি ফাইল িনবর্চনী Activity সরবরাহ করা। ক্লায়েন্ট অ্যাপ একশন ACTION PICK ধারণ করা একটি Intent দিয়ে startActivityForResult() কল করার মাধ্যমে এই Activity শুরু করে। যখন ক্লায়েন্ট অ্যাপ startActivityForResult() কল করে, আপনার অ্যাপ ক্লায়েন্ট অ্যাপে একটি রিজাল্ট ফেরত দিতে পারে, ইউজার যে ফাইল সিলেক্ট করেছে তার জন্য একটি কন্টেন্ট ইউআরআই আকরে।

একটি ক্লায়েন্ট অ্যাপে একটি ফাইলের জন্য িরেকায়েস্ ট বাস্ তাবায়ন করতে হয় কীভাবে তা শিখতে অনুশীলনী Requesting a Shared File দেখুন।

ফাইল সিলেকশন একটি টিভি টৈতরী করা

ফাইল সিলেকশন একটি টিভি টৈ সটআপ করতে, আপনার মেনিফেস্ট Activity নিদির্ষ্ট করার মাধ্যমে শুরু করুন, একটি ইন্টেন্ট ফিল্টার সাথে নিয়ে যা একশন ACTION_PICK, ক্যাটাগরি CATEGORY_DEFAULT, এবং CATEGORY_OPENABLE ম্যাচ করে। আপনার অ্যাপ অন্য অ্যাপে ফাইল পরিবেশন করে তার জন্য MIME টাইপ ফিল্টার যুক্তও করে। নীচের অংশটি আপনাকে দেখাবে নতুন একটি টিভি টৈ এবং ইন্টেন্ট ফিল্টার কীভাবে সুনির্দির্ষ্ট করতে হয়:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android">
...
    <application>
    ...
        <activity
            android:name=".FileSelectActivity"
            android:label="@string/File_Selector" >
            <intent-filter>
                <action
                    android:name="android.intent.action.PICK"/>
                <category
                    android:name="android.intent.category.DEFAULT"/>
                <category
                    android:name="android.intent.category.OPENABLE"/>
                <data android:mimeType="text/plain"/>
                <data android:mimeType="image/*"/>
            </intent-filter>
        </activity>
```


ফাইল সিস্টেম লকশন একটি টিভি টেকে ডি নির্ধারণ করুন

পরবর্তীতে, একটি Activity সাবক্লাস নির্ধারণ করুন যা ইন্টারনাল স্টোরেজ আপনার অ্যাপের files/images/ ডিরেক্টরী থেকে সহজপরিচয় ফাইল নির্দেশ করে এবং ইউজারকে কাঙ্ক্ষিত ফাইল তুলে নিতে অনুমোদন করে। নিম্নোক্ত অংশটি আপনাকে দেখায় কীভাবে এই Activity নির্ধারণ করতে হয় এবং ইউজারের সিস্টেম লকশনকে রিসপন্স করতে হয়:

```
public class MainActivity extends Activity {
    // The path to the root of this app's internal storage
    private File mPrivateRootDir;
    // The path to the "images" subdirectory
    private File mImagesDir;
    // Array of files in the images subdirectory
    File[] mImageFiles;
    // Array of filenames corresponding to mImageFiles
    String[] mImageFilenames;
    // Initialize the Activity
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        ...
        // Set up an Intent to send back to apps that request a file
        mResultIntent =
            new Intent("com.example.myapplication.ACTION_RETURN_FILE");
        // Get the files/ subdirectory of internal storage
        mPrivateRootDir = getFilesDir();
        // Get the files/images subdirectory;
        mImagesDir = new File(mPrivateRootDir, "images");
        // Get the files in the images subdirectory
        mImageFiles = mImagesDir.listFiles();
        // Set the Activity's result to null to begin with
        setResult(Activity.RESULT_CANCELED, null);
        /*
         * Display the file names in the ListView mFileListView.
         * Back the ListView with the array mImageFilenames, which
         * you can create by iterating through mImageFiles and
         * calling File.getAbsolutePath() for each File
         */
        ...
    }
    ...
}
```

ফাইল সিলেকশনকে রেসপন্স করা

কখনও একজন ইউজার একটি শেয়ার করা ফাইল নিবর্চন (সিলেক্ট) করে, আপনার অ্যাপ্লিকেশনকে অবশ্যই কৈমান ফাইল সিলেক্ট করা হয়েছে তা নিবর্চন করা এবং তারপর ফাইলের জন্য একটি কন্টেন্ট ইউআরআই তৈরী করতে হবে। যেহেতু Activity একটি ListView সহজপরাপয় ফাইলের তালিকা পরদর্শন করে, যখন ইউজার একটি ফাইল নামিক্লিক করে সিস্টেম মেথড onItemClick() কল করে, যার মধ্যে আপনি একটি নিবর্চন চিত ফাইল পেয়ে থাকে।

onItemClick() এর মধ্যে, নিবর্চন চিত ফাইলের ফাইল নামের জন্য একটি File অবজেক্ট পেতে এবং getUriForFile() একটি আরগুমেন্ট হিসাবে পাস করতে, অথরিটি সহ যা আপনি FileProvider এর জন্য <provider> এলিমেন্ট সূচিন্দির্ষ ট করেছেন। উদ্ভূত কন্টেন্ট ইউআরআই অথরিটি, ফাইলের ডিরেক্টরীর (এক্সএমএল মেটা-ডাটাতে নিদির্ষ টকরনের মতো) সাথে সংশ্লিষ্ট একটি প্যাথ সেগমেন্ট, এবং এক্সটেনশন সহ ফাইলের নাম ধারণ করে। কীভাবে FileProvider এক্সএমএল মেটা-ডাটা এর উপরিভিত্ত করে প্যাথ সেগমেন্ট ডিরেক্টরীগুণে লাময় প করে তা (Specify Sharable Directories) অধ্যয়ে আলোচনা করা হয়েছে।

নিম্নোক্ত অংশটি আপনাকে দেখায় কীভাবে নিবর্চন চিত ফাইল সনাক্ত করতে হয় এবং এটার জন্য একটি কন্টেন্ট ইউআরআই পেতে হয়:

```
protected void onCreate(Bundle savedInstanceState) {  
    ...  
    // Define a listener that responds to clicks on a file in the ListView  
    mFileListView.setOnItemClickListener(  
        new AdapterView.OnItemClickListener() {  
            @Override  
            /*  
             * When a filename in the ListView is clicked, get its  
             * content URI and send it to the requesting app  
             */  
            public void onItemClick(AdapterView<?> adapterView,  
                View view,  
                int position,  
                long rowId) {  
                /*  
                 * Get a File for the selected file name.  
                 * Assume that the file names are in the  
                 * mImageFilename array.  
                 */  
                File requestFile = new File(mImageFilename[position]);  
                /*  
                 * Most file-related method calls need to be in  
                 * try-catch blocks.  
                 */  
                // Use the FileProvider to get a content URI  
                try {  
                    fileUri = FileProvider.getUriForFile(  
                        MainActivity.this,  
                        "com.example.myapp.fileprovider",  
                        requestFile);  
                } catch (IllegalArgumentException e) {  
                    Log.e("File Selector",  
                        "The selected file can't be shared: " +  
                        clickedFilename);  
                }  
                ...  
            }  
        });  
    ...  
}
```

মেনরাখেবন যে আপনি শুধুমাত্র ফাইলের জন্য কন্টেন্ট ইউআরআই তৈরী করতে পারবেন, যে ফাইল একটি ডিরেক্টরীর মধ্যে থাকে যা আপনি মেটা-ডাটা ফাইল নিদির্ষ ট করেছেন, যা <paths> এলিমেন্ট ধারণ করে, যেভাবে পূর্ববর্তরী অধ্যয়ে (Specify Sharable Directories) আলোচিত হয়েছে। একটি প্যাথ এ

একটি File এর জন্য যদি আপনি `getUriForFile()` কল করেন যা আপনি নির্দিষ্ট করেন
নি, আপনি একটি `IllegalArgumentException` রিসভ করবেন।

ফাইলের জনয পারিমিশন (অনুমতি) পরদান

এখন অন্য অর্থাৎ পরসাথে শেয়ার করতে চাওয়া ফাইলের জনয আপনার একটি কন্টেন্ট ইউআরআই আছে, আপনার দরকার কলয়েন্ট অর্থাৎ পেকে ফাইলে পর্বেশ করতে দেয়া। পর্বেশ গময তা অনুমোদন করতে, একটি Intent এ কন্টেন্ট ইউআরআই যুক্ত করার মাধ্যমে কলয়েন্ট অর্থাৎ প পারিমিশন দিতে পারেন এবং Intent এর উপর পারিমিশন ফ্লাগ সেট করতে পারেন। যে পারিমিশন আপনি দিতেছেন তা অস্থায়ী এবং স্বয়ংক্রিয়ভাবে সমাপ্ত হবে যখন রিসিভ করা অর্থাৎ পরকাযর কর্ম শেষ হবে।

নিম্নে নাক্ত কড অংশটি দেখায় আপনি কীভাবে ফাইলের জনয রিড পারিমিশন সেট করে বন:

```
protected void onCreate(Bundle savedInstanceState) {  
    ...  
    // Define a listener that responds to clicks in the ListView  
    mFileListView.setOnItemClickListener(  
        new AdapterView.OnItemClickListener() {  
            @Override  
            public void onItemClick(AdapterView<?> adapterView,  
                View view,  
                int position,  
                long rowId) {  
                ...  
                if (fileUri != null) {  
                    // Grant temporary read permission to the content URI  
                    mResultIntent.addFlags(  
                        Intent.FLAG_GRANT_READ_URI_PERMISSION);  
                }  
                ...  
            }  
        }  
    );  
    ...  
}
```

সতর্কতা: অস্থায়ী পর্বেশ পারিমিশন ব্যবহার করে আপনার ফাইল
নিরপদভাবে পর্বেশ গময তা অনুমোদন করতে setFlags() কল করা হচ্ছে
একমাত্র উপায়। একটি ফাইলের কন্টেন্ট ইউআরআই এর জনয
Context.grantUriPermission() মেথড কল করা পিহার করুন, কারণ এই মেথড
পর্বেশ গময তা পরদান করে যাতে আপনি শুধুমাত্র
Context.revokeUriPermission() কল করার মাধ্যমে আবাহন করতে পারেন।

রিকোয়েস্ট করা অ্যাপের সাথে ফাইল শেয়ার

অ্যাপ যা ফাইল রিকোয়েস্ট করে এটাকে দিয়ে ফাইল শেয়ার করতে,
কন্টেন্ট ধারণ করা Intent পাস করুন এবং setResult() এ পরিমিশন দিন। যখন Activity যা
আপনি মাত্র ইনিশিয়াল করা হয়েছে তখন শেষ হলে, সিস্টেম কলোয়েন্ট
অ্যাপ কন্টেন্ট ইউআই ধারণ করা Intent সেন্ড করে। নিম্নোক্ত
কোড অংশটি দেখায় কীভাবে এটা করতে হয়:

```
protected void onCreate(Bundle savedInstanceState) {  
    ...  
    // Define a listener that responds to clicks on a file in the ListView  
    mFileListView.setOnItemClickListener(  
        new AdapterView.OnItemClickListener() {  
            @Override  
            public void onItemClick(AdapterView<?> adapterView,  
                View view,  
                int position,  
                long rowId) {  
                ...  
                if (fileUri != null) {  
                    ...  
                    // Put the Uri and MIME type in the result Intent  
                    mResultIntent.setDataAndType(  
                        fileUri,  
                        getContentResolver().getType(fileUri));  
                    // Set the result  
                    MainActivity.this.setResult(Activity.RESULT_OK,  
                        mResultIntent);  
                } else {  
                    mResultIntent.setDataAndType(null, "");  
                    MainActivity.this.setResult(RESULT_CANCELED,  
                        mResultIntent);  
                }  
            }  
        });  
};
```

ইউজারকে কলোয়েন্ট অ্যাপ তা ক্লিক করতে বিধির আসার জন্য একটা
উপায় পর্দান করুন যদি তারাইতিমধ্যে একটা ফাইল বেছে নেয়। এটা করার
একটা উপায় একটা চেকমাকর বা **Done** বাটন সরবরাহ করা। বাটনের android:onClick
এটি রিবিউট বয় বহার করে বাটনের সাথে একটা মেথড যুক্ত করুন।
উদাহরণস্বরূপ:

```
public void onDoneClick(View v) {  
    // Associate a method with the Done button  
    finish();  
}
```

একটি শেয়ার করা ফাইল রিকোয়েস্ট করা

(<http://developer.android.com/training/secure-file-sharing/request-file.html>)

যখন একটি অ্যাপ অন্য অ্যাপ দ্বারা শেয়ার করা একটি ফাইল পরবেশ করতে চায়, রিকোয়েস্ট করা (ক্লায়েন্ট) অ্যাপ সাধারণভাবে ফাইল শেয়ার করা অ্যাপকে একটি রিকোয়েস্ট সেন্ড করে (সারভার)। অধিকাংশ ক্ষেত্রে, সারভার অ্যাপ রিকোয়েস্টটি একটি একটি ভিডিও শুরু করে যা ফাইল পরদর্শন করে এটি শেয়ার করতে পারে। ইউজার একটি ফাইল তুলে নেয়, যার পরে সারভার অ্যাপ ক্লায়েন্ট অ্যাপ ফাইলের কন্টেন্ট ইউআরআই ফরম্যাট পাঠায়।

এই অনুশীলনীতে দেখায় কীভাবে একটি ক্লায়েন্ট অ্যাপ একটি সারভার অ্যাপ থেকে একটি ফাইল রিকোয়েস্ট করে, সারভার অ্যাপ থেকে ফাইলের কন্টেন্ট ইউআরআই গ্রহণ করে, এবং কন্টেন্ট ইউআরআই ব্যবহার করে ফাইলটি ওপেন করে।

ফাইলের জন্য একটি রিকোয়েস্ট সেন্ড (পাঠানো) করা

সারভার অ্যাপ থেকে একটি ফাইল রিকোয়েস্ট করতে, ক্লায়েন্ট অ্যাপ ACTION_PICK এবং একটি MIME টাইপের মতো একশন ধারণ করা Intent দিয়ে startActivityForResult কল করে যা ক্লায়েন্ট অ্যাপ চালিত করতে পারে।

উদাহরণস্বরূপ, নিম্নোক্ত কোড অংশটি দেখায় Activity शुरू করার জন্য একটি সারভার অ্যাপে কীভাবে একটি Intent সেন্ড করতে হয় যা Sharing a File এ আলাচনা করা হয়েছে:

```
public class MainActivity extends Activity {
    private Intent mRequestFileIntent;
    private ParcelFileDescriptor mInputPFD;
    ...
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mRequestFileIntent = new Intent(Intent.ACTION_PICK);
        mRequestFileIntent.setType("image/jpg");
        ...
    }
    ...
    protected void requestFile() {
        /**
         * When the user requests a file, send an Intent to the
         * server app.
         * files.
         */
        startActivityForResult(mRequestFileIntent, 0);
        ...
    }
    ...
}
```

রিকোয়েস্ট করা ফাইলে পরবেশ

সারভার অ্যাপ ফাইলের কন্টেন্ট ইউআরআই একটি Intent এর মধ্যে কলোয়েন্ট অ্যাপে ফেরত পাঠায়। এই Intent কলোয়েন্ট অ্যাপে এটার onActivityResult() এর ওভাররাইডে পাস হয়। কখনও কলোয়েন্ট অ্যাপে ফাইলের কন্টেন্ট ইউআরআই থাকে, এটা ফাইলে পরবেশ করতে পারে এটার FileDescriptor পাওয়ার মাধ্যমে।

ফাইল নিরাপত্তা এই পিকারয়ার মধ্যে সুরক্ষিত আছে কারণ কন্টেন্ট ইউআরআই হচ্ছে ডাটার একমাত্র অংশ যা কলোয়েন্ট অ্যাপ গ্রহণ করে। যেহেতু এই ইউআরআই একটি ডিরেক্টরী পথ ধারণ না করে, কলোয়েন্ট অ্যাপ সারভার অ্যাপের অন্য ফাইল আবিষ্কার এবং ওপেন করতে পারেনা। শুধুমাত্র কলোয়েন্ট অ্যাপ ফাইলে পরবেশ গম্যতা লাভ করে, এবং শুধুমাত্র সারভার অ্যাপ কতদূর প্যারামিশন প্রদান করা হয়। প্যারামিশনটা অস্থায়ী, তাই একসময় কলোয়েন্ট অ্যাপের কায়রকর্ম শেষ হলে, সারভার অ্যাপের বাইরে থেকে ফাইলের পরবেশ গম্যতা বিবদ্যমান থাকেনা। নিম্নোক্ত অংশটি দেখায় সারভার অ্যাপ থেকে পাঠানো Intent কে কলোয়েন্ট অ্যাপ কীভাবে চািলত করে, এবং ইউআরআই ব্যবহার করে কীভাবে কলোয়েন্ট অ্যাপ FileDescriptor পেয়ে থাকে:

```
/*
 * When the Activity of the app that hosts files sets a result and calls
 * finish(), this method is invoked. The returned Intent contains the
 * content URI of a selected file. The result code indicates if the
 * selection worked or not.
 */
@Override
public void onActivityResult(int requestCode, int resultCode,
    Intent returnIntent) {
    // If the selection didn't work
    if (resultCode != RESULT_OK) {
        // Exit without doing anything else
        return;
    } else {
        // Get the file's content URI from the incoming Intent
        Uri returnUrl = returnIntent.getData();
        /*
         * Try to open the file for "read" access using the
         * returned URI. If the file isn't found, write to the
         * error log and return.
         */
        try {
            /*
             * Get the content resolver instance for this context, and use it
             * to get a ParcelFileDescriptor for the file.
             */
            mInputPFD = getContentResolver().openFileDescriptor(returnUrl, "r");
        } catch (FileNotFoundException e) {
            e.printStackTrace();
            Log.e("MainActivity", "File not found.");
            return;
        }
        // Get a regular file descriptor for the file
        FileDescriptor fd = mInputPFD.getFileDescriptor();
        ...
    }
}
```

মেথড openFileDescriptor() ফাইলের জন্য একটি ParcelFileDescriptor ফেরত পাঠায়। এই অবজেক্ট থেকে, কলোয়েন্ট অ্যাপ একটি FileDescriptor অবজেক্ট লাভ করে, এটা পরবর্তীতে ফাইল রিড করতে ব্যবহৃত হয়।

ফাইল তথ্য উদ্ধার

(<http://developer.android.com/training/secure-file-sharing/retrieve-info.html>)

একটি ফাইল যার জন্য এর একটি কন্টেন্ট ইউআরআই আছে তার সাথে একটি ক্লায়েন্ট অ্যাপের কাজ করার চেষ্টা করার পূর্বে, অ্যাপটি সারভার অ্যাপ থেকে তথ্য সম্পর্কিত কয়েকটি করতে পারে, যার মধ্যে রয়েছে ডাটাইপ এবং ফাইল সাইজ। ডাটাইপ ক্লায়েন্ট অ্যাপে নিধরারন করতে সাহায্য করে যেতে এটা ফাইলকে চািলত করতে পারে এবং ফাইল সাইজ ক্লায়েন্ট অ্যাপে ফাইলের জন্য বাফারিং এবং ক্যাশিং সেটআপ করতে সহায়তা করে।

এই অনুশীলনী দেখায় একটি ফাইলের MIME টাইপ এবং সাইজ উদ্ধার করতে কীভাবে সারভার অ্যাপের FileProvider অনুসন্ধান করতে হয়।

একটি ফাইলের MIME টাইপ উদ্ধারণ

একটি ফাইলের ডাটাতাইপ ক্লায়েন্ট অ্যাপ নিবেদর্শকের কীভাবে এটার ফাইলের কন্টেন্ট চািলত করা উচিত। একটা শেয়ার করা ফাইলের ডাটাতাইপ পেতে এটার কন্টেন্ট ইউআরআই পর্দান করে, ক্লায়েন্ট অ্যাপ `ContentResolver.getType()` কল করে। এই মেথড ফাইলের MIME টাইপ ফেরত দেয়। বাইডিফল্ট, একটি `FileProvider` ফাইলের MIME টাইপ নিরূপন করে এর ফাইল নেম এক্সটেনশন থেকে।

নিম্নোক্ত কোড চিত্রটি দেখায় ক্লায়েন্ট অ্যাপ একটি ফাইলের MIME টাইপ উদ্ধারণ করে, কোন সময় যদি সারভার অ্যাপ ক্লায়েন্ট কন্টেন্ট ইউআরআই ফেরত পাঠায়:

```
...
/*
 * Get the file's content URI from the incoming Intent, then
 * get the file's MIME type
 */
Uri returnUrl = returnIntent.getData();
String mimeType = getContentResolver().getType(returnUrl);
...
```

একটি ফাইলের নাম ও সাইজ উদ্ধারণ

FileProvider ক্লাসের query() মেথডের একটি ডিফল্ট বাস্তুবায়ন আছে যা একটি Cursor এ একটি কন্টেন্ট ইউআরআই এর সাথে যুক্ত ফাইলের নাম ও সাইজকে ফেরত পাঠায়। ডিফল্ট বাস্তুবায়ন দুইটুকলাম ফেরত পাঠায়:

ডিসপ্লেইনাম (DISPLAY NAME)

ফাইলের নাম, একটি String হিসাবে। এই ভ্যালুটা File.getName() দ্বারা ফেরত আসে ভ্যালুর মতো একইভাবে।

সাইজ (SIZE)

ফাইলের সাইজ বাইটে হয়, একটি ষড়হম হিসাবে। এই ভ্যালুটা File.length() দ্বারা ফেরত আসে ভ্যালুর মতো একইভাবে।

কন্টেন্ট ইউআরআই এর জন্য ছাড়া null এ query() এর আরগুমেন্টের সব কিছু সেট করার মাধ্যমে ক্লায়েন্ট অ্যাপ একটি ফাইলের জন্য DISPLAY NAME এবং SIZE উভয়ই পেতে পারে। উদাহরণস্বরূপ, এই কোড চিত্রটি একটি ফাইলের DISPLAY NAME এবং SIZE উদ্ধারণ করে এবং প্রতিটুকু TextView এ প্রদর্শন করে:

```
...
/*
 * Get the file's content URI from the incoming Intent,
 * then query the server app to get the file's display name
 * and size.
 */
Uri returnUrl = returnIntent.getData();
Cursor returnCursor =
    getContentResolver().query(returnUrl, null, null, null, null);
/*
 * Get the column indexes of the data in the Cursor,
 * move to the first row in the Cursor, get the data,
 * and display it.
 */
int nameIndex = returnCursor.getColumnIndex(OpenableColumns.DISPLAY_NAME);
int sizeIndex = returnCursor.getColumnIndex(OpenableColumns.SIZE);
returnCursor.moveToFirst();
TextView nameView = (TextView) findViewById(R.id.filename_text);
TextView sizeView = (TextView) findViewById(R.id.filesize_text);
nameView.setText(returnCursor.getString(nameIndex));
sizeView.setText(Long.toString(returnCursor.getLong(sizeIndex)));
...
```

এনএফসি দিয়ে ফাইল শেয়ার করা

(<http://developer.android.com/training/beam-files/index.html>)

অ্যান্ড্রয়েড বিম ফাইল টরান্সফার বৈশিষ্ট্য ব্যবহার করে ডিভাইসে সরমধ্যে বড় ফাইল বদলি করতে অ্যান্ড্রয়েড আপনাকে অনুমোদন দেয়। এই বৈশিষ্ট্যের একটি সরল এপিআই আছে এবং শুধুমাত্র ডিভাইস ট্যাচ করার মাধ্যমে ইউজারকে বদলি পরীক্ষার টি শুরুর করতে দেয়। জবাবে, অ্যান্ড্রয়েড বিম ফাইল টরান্সফার একটি ডিভাইস থেকে আরেকটিতে সর্বোৎকর্ষভাবে ফাইল কপি করে এবং ইউজারকে এটা অবগত করে যখন এটা শেষ হয়।

যখন অ্যান্ড্রয়েড বিম ফাইল টরান্সফার এপিআই ব্যবহারকারী পরিমাণ ডাটামিনিয়াকাজ করে, অ্যান্ড্রয়েড ৪.০ (এপিআই লেভেল ১৪) এ চালিত অ্যান্ড্রয়েড বিম NDEF টরান্সফার এপিআই অল্প পরিমাণ ডাটামিনিয়াকাজ করে যেমন ইউআরআই বা অন্য ছোট বার্তা। উপরন্তু, অ্যান্ড্রয়েড বিম একমাত্র বৈশিষ্ট্য যা অ্যান্ড্রয়েড NFC ফর্মওয়ার্কের পাওয়া যায়, যেটা আপনাকে ট্যাগ থেকে NDEF মেসেজ পড়তে (রিড) অনুমোদন করে। অ্যান্ড্রয়েড বিম সম্পর্কে আরও জানতে Beaming NDEF Messages to Other Devices দেখুন এবং NFC ফর্মওয়ার্কের আরও জানতে Near Field Communication এপিআই গাইড দেখুন

অনুশীলনীসমূহ

অনয় ডিভাইসে ফাইল সন্ড করা

অনয় ডিভাইসে ফাইল সন্ড করতে শিখুন কীভাবে আপনার অয় প সেটআপ করতে হয়।

অনয় ডিভাইস থেকে ফাইল রিসভ করা

অনয় ডিভাইস থেকে পাঠানো ফাইল রিসভ করতে কীভাবে আপনার অয় প সেটআপ করতে হয়

মেমিনেফেস্টিফচারগুেলা (বৈশিষ্টগুেলা) ডেকে ক্লয়ার করা

পর্যন্ত, আপনার অর্থে পরে পরিমিশন এবং ফিচার পরে যাজন তা
ডেকে ক্লয়ার করতে আপনার অর্থে মেমিনেফেস্টি এডিট করুন।

পারমিশন (অনুমতি) রিকোয়েস্ট (অনুরোধ) করা

NFC ব্যবহার করে এক্সটানর্নাল স্টোরেজ থেকে ফাইল সেন্ড করতে আপনার অ্যাপকে অ্যান্ড্রয়েড বিম ফাইল ট্রান্সফার ব্যবহার করতে দিতে, আপনাকে অবশ্যই আপনার অ্যাপ মেনিফেস্টে স্টিনিমেন্টে পারমিশনগুলো রিকোয়েস্ট করতে হবে: NFC আপনার অ্যাপকে NFC এর উপরে ডাটাসেন্ড করতে দিন। এই পারমিশন নিম্নদর্শক করতে <manifest>এলিমেন্টের একটি চাইল্ড ইমপ্লিমেন্টে একটি এলিমেন্ট যুক্ত করুন:

```
<uses-permission  
    android:name="android.permission.NFC" />
```

READ_EXTERNAL_STORAGE

আপনার অ্যাপকে এক্সটানর্নাল স্টোরেজ থেকে রিড (পড়তে) করতে দিন। এই পারমিশন নিম্নদর্শক করতে <manifest>এলিমেন্টের একটি চাইল্ড ইমপ্লিমেন্টে একটি এলিমেন্ট যুক্ত করুন:

```
<uses-permission  
    android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

নোট: অ্যান্ড্রয়েড ৪.২.২ (এপিআই লেভেল ১৭) শুরুর ক্ষেত্রে এই পারমিশন বাধ্যতামূলক নয়। প্ল্যাটফর্মের ভবিষ্যত সংস্করণে অ্যাপের জন্য এটা চাইতে পাবেন যা এক্সটানর্নাল স্টোরেজ থেকে রিড করতে (পড়তে) চাইতে পারে। অগ্রেসর সবার্গ হনোযোগ্যতা নিশ্চিত করতে, এখন পারমিশন রিকোয়েস্ট করুন, এটা পরে যাজনীয় হওয়ার পূর্বেই

NFC বৈশিষ্ট্য গুলো (ফিচার গুলো) নিদির্ষ্ট করুন

নিদির্ষ্ট করুন যে আপনার অ্যাপ NFC ব্যবহার করছে, <manifest> এ লিমেন্টর একটি চাইল্ড হিসাবে একটি <uses-feature> element যুক্ত করার মধ্যে দিয়ে। আপনার অ্যাপ NFC উপস্থিতি ছাড়া কাজ করতে পারেনা এটা নিদির্ষ্ট করতে true এ android:required এটি ট্রিবিউটে সেট করে।

নিমেম্নাক্ত খনিডত অংশটি দেখায় কীভাবে <uses-feature> এ লিমেন্ট নিদির্ষ্ট করতে হয়:

```
<uses-feature  
  android:name="android.hardware.nfc"  
  android:required="true" />
```

উল্লেখ্য শুধুমাত্র NFC ব্যবহার করা যদি আপনার অ্যাপের একমাত্র অপশন হয়, যদি NFC উপস্থিতি না থাকে কিন্তু এখনও কাজ করছে, আপনার উচিত false এ android:required সেট করা এবং কেডে NFC এর জন্য পরীক্ষা করা।

অ্যান্ড্রয়েড বিম ফাইল ট্রান্সফার নিদির্ষ্ট করুন

যেহেতু অ্যান্ড্রয়েড বিম ফাইল ট্রান্সফার শুধুমাত্র
অ্যান্ড্রয়েড ৪.১ (এপিআই লেভেল ১৬) এবং পরবর্তী সংস্করণে পাওয়া
যায়, যদি আপনার অ্যাপ এর কার্যকর্যের একটি মূল অংশের জন্য
অ্যান্ড্রয়েড বিম ফাইল ট্রান্সফার এ উপরে নিভর্শীল হয় আপনাকে
অবশ্যই `android:minSdkVersion="16"` এটি ট্রিবিউট সহকারে `<uses-sdk>` এ লিখে মনট
নিদির্ষ্ট করতে হবে। অন্য পর্কারে আপনি অন্য অ্যাপ পর্যেয়াজন
অনুসারে `android:minSdkVersion` সেট করতে পারেন এবং কেড়ে প্লাটফর্ম
সংস্করণ পরীক্ষা করুন, যেভাবে বিনম্ে াক্ ত সেকশনে আলোচনা করা
হয়েছে।

অ্যান্ড্রয়েড বিম ফাইল ট্রান্সফার সাপোর্টের জনস্ব পরীক্ষা

আপনার অ্যাপ মেনিফেস্টে স্টিনিদির্ষ্ট করতে NFC হচ্ছে অপশনাল, আপনি নিম্নোক্ত এলিমেন্ট বয়বহার করতে পারেন:

```
<uses feature android:name="android.hardware.nfc" android:required="false" />
```

আপনি যদি `android:required="false"` সেট করেন, আপনাকে অবশ্যই কয়েড NFC সাপোর্ট এবং অ্যান্ড্রয়েড বিম ফাইল ট্রান্সফার সাপোর্টের জনস্ব পরীক্ষা করতে হবে।

কয়েড অ্যান্ড্রয়েড বিম ফাইল ট্রান্সফার সাপোর্টের জনস্ব পরীক্ষা করতে, পরীক্ষা দবারা শুরুর করুন যাতে আরগুমেণ্ট `FEATURE NFC` সহকারে `PackageManager.hasSystemFeature()` কল করার মাধ্যমে ডিভাইস NFC কে সাপোর্ট করে। পরে, চেক করুন যে অ্যান্ড্রয়েড ভাসরন SDK INT এর ভ্যালু পরীক্ষা করার মাধ্যমে অ্যান্ড্রয়েড বিম ফাইল ট্রান্সফারকে সাপোর্ট করে। যদি অ্যান্ড্রয়েড বিম ফাইল ট্রান্সফার সাপোর্ট পায়, NFC কন্ট্রোলারের একটি ইনস্টেন্স লাভ করে যা আপনাকে NFC হাডর্ ওয়ারের সাথে থেগাগােগ করতে দিবে।
উদাহরণস্বরূপ:

```
public class MainActivity extends Activity {
    ...
    NfcAdapter mNfcAdapter;
    // Flag to indicate that Android Beam is available
    boolean mAndroidBeamAvailable = false;
    ...
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        ...
        // NFC isn't available on the device
        if (!PackageManager.hasSystemFeature(PackageManager.FEATURE_NFC)) {
            /*
             * Disable NFC features here.
             * For example, disable menu items or buttons that activate
             * NFC-related features
             */
            ...
            // Android Beam file transfer isn't supported
        } else if (Build.VERSION.SDK_INT <
            Build.VERSION_CODES.JELLY_BEAN_MR1) {
            // If Android Beam isn't available, don't continue.
            mAndroidBeamAvailable = false;
            /*
             * Disable Android Beam file transfer features here.
             */
            ...
            // Android Beam file transfer is available, continue
        } else {
            mNfcAdapter = NfcAdapter.getDefaultAdapter(this);
            ...
        }
    }
    ...
}
```

একটি কলব্যাক মেথড তৈরী করুন যা ফাইল সরবরাহ করে

ইতিপূর্বে বর্ণনা আপনি যদি নিশ্চিত করেন যে ডিভাইসটি অ্যান্ড্রয়েড বিম ফাইল ট্রান্সফার সাপোর্ট করে, একটি কলব্যাক মেথড যুক্ত করুন যা সিস্টেমটি আবাহন করে যখন অ্যান্ড্রয়েড বিম ফাইল ট্রান্সফার চিহ্নিত করে যে ইউজার অন্য একটি NFC-এনাবলড ডিভাইসে একটি ফাইল পাঠাতে চায়। এই কলব্যাক মেথড, Uri অবজেক্টের একটি বিনয়ী সফরত দিন। অ্যান্ড্রয়েড বিম ফাইল ট্রান্সফার রিসিভিং ডিভাইসে এই সব ইউআরআই দ্বারা পরিবেশিত ফাইলটি টেককপি করে।

কলব্যাক মেথড যুক্ত করতে, `NfcAdapter.CreateBeamUriCallback` ইন্টারফেস এবং এর মেথড `createBeamUri()` বাস্তবায়ন করুন। নিম্নোক্ত খন্ডাংশটি দেখায় এটা কীভাবে করতে হয়:

```
public class MainActivity extends Activity {
    ...
    // List of URIs to provide to Android Beam
    private Uri[] mFileUri = new Uri[10];
    ...
    /**
     * Callback that Android Beam file transfer calls to get
     * files to share
     */
    private class FileUriCallback implements
        NfcAdapter.CreateBeamUriCallback {
        public FileUriCallback() {
        }
        /**
         * Create content URIs as needed to share with another device
         */
        @Override
        public Uri[] createBeamUri(NfcEvent event) {
            return mFileUri;
        }
    }
    ...
}
```

আপনি কখনও ইন্টারফেসটি বাস্তবায়ন করলে, `setBeamPushUriCallback()` কল করার মাধ্যমে অ্যান্ড্রয়েড বিম ফাইল ট্রান্সফারের কলব্যাক সরবরাহ করুন। নিম্নোক্ত খন্ডাংশটি আপনাকে দেখাবে কীভাবে এটা করতে হয়:

```
public class MainActivity extends Activity {
    ...
    // Instance that returns available files from this app
    private FileUriCallback mFileUriCallback;
    ...
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        ...
        // Android Beam file transfer is available, continue
        ...
        mNfcAdapter = NfcAdapter.getDefaultAdapter(this);
        /**
         * Instantiate a new FileUriCallback to handle requests for
         * URIs
         */
        mFileUriCallback = new FileUriCallback();
        // Set the dynamic callback for URI requests.
        mNfcAdapter.setBeamPushUriCallback(mFileUriCallback, this);
        ...
    }
    ...
}
```

নোট: আপনার অ্যাপের NfcAdapter ইনস্টেন্স এর মাধ্যমে সরাসরি NFC ফর্মওয়ার্ডকে করে আপনি Uri অবজেক্টের বিনয়্যাস পরদানও করতে পারেন। এই অ্যাপের রাচ বেছে নিলে, আপনি যদি NFC টাচ ইভেন্ট ঘটানোর পূর্বেই স্থানান্তর করতে ইউআরআই নির্ধারণ করতে পারেন। এই অ্যাপের রাচ সম্পর্কে আরও জানতে, NfcAdapter.setBeamPushUri() দেখুন।

সেন্ড করতে ফাইল নিদির্ষ্ট করুন

অনয় NFC- এনাবলড ডিভাইসে এক বা একাধিক ফাইল স্থানান্তর করতে, প্রতিটা ফাইলের জন্য একটি ফাইল ইউআরআই (একটি file সহকারে একটি URI) সংগ্রহ করুন এবং তারপর Uri অবজেক্টের একটি বিবনয়সে এটা যুক্ত করুন। একটি ফাইল স্থানান্তর করতে, আপনার অবশয়ই ফাইলের জন্য রিড করার স্থায়ী পর্বে বশগময়তা থাকতে হবে। উদাহরণস্বরূপ, নিম্নোক্ত খন্ডাংশটি দেখাবে একটি ফাইল নেম থেকে কীভাবে আপনি একটি ফাইল ইউআরআই পাবেন এবং তারপর ইউআরআইটি বিবনয়সে (অয়সে) যুক্ত করবেন:

```
/*
 * Create a list of URIs, get a File,
 * and set its permissions
 */
private Uri[] mFileUris = new Uri[10];
String transferFile = "transferimage.jpg";
File extDir = getExternalFilesDir(null);
File requestFile = new File(extDir, transferFile);
requestFile.setReadable(true, false);
// Get a URI for the File and add it to the list of URIs
fileUri = Uri.fromFile(requestFile);
if (fileUri != null) {
    mFileUris[0] = fileUri;
} else {
    Log.e("My Activity", "No File URI available for file.");
}
```

অন্য ডিভাইস থেকে ফাইল রিসভ করা

(<http://developer.android.com/training/beam-files/receive-files.html>)

অ্যান্ড্রয়েড বিম ফাইল ট্রান্সফার রিসিভিং ডিভাইসের একটি বিশেষ ডিরেক্টরিতে ফাইল কপি করে। এটা অ্যান্ড্রয়েড মিডিয়া স্কয়ার্নার দিয়ে কপি করা ফাইলের স্কয়ার্ন করে এবং MediaStore পর্তাইডারে মিডিয়া ফাইলের জন্য এন্ট্রি যুক্ত করে। এই অনুশীলনী দেখায় কীভাবে আপনি রেসপন্স করে বন যখন ফাইল কপি সম্পূর্ণ হয় এবং রিসিভিং ডিভাইস কপি করা ফাইলকে স্থাপন করেন।

ডাটাপর্দশর্ন করতে একটি রিকোয়েস্টের পর্তিরেসপন্স করা

অ্যান্ডরয়েড বিম ফাইল টর্ান্সফার যখন রিসিভিং ডিভাইসে ফাইল
কপি করার কাজ শেষ করে, এটা একশন ACTION VIEW সহকারে একটি Intent ধারণ করা
নেটিটিফিকেশন, পর্থম ফাইলের MIME টাইপ যা স্থানান্তরিত হয়েছে, এবং
একটি ইউআরআই যা পর্থম ফাইলের পর্তিতিনেদর্শ করে তা পেস্ট করে।
যখন ইউজার নেটিটিফিকেশন ক্লিক করে, এই ইন্টেন্টটি সিস্টেমের বাইরে
পঠানো হয়। আপনার অ্যাপের এই ইন্টেন্টের সপন্স করতে, Activity এর <
activity> এ লিখে মেন্টর জনয একটি <intent-filter> যুক্ত করুন যার রেসপন্স করা
উচিত। <intent-filter> এ লিখে মেন্টর মধ্যে নিম্নোক্ত চাইল্ড এলিমেন্ট
যুক্ত করুন:

```
<action android:name="android.intent.action.VIEW" />
```

নেটিটিফিকেশন থেকে পাঠানো ACTION VIEW ইন্টেন্ট ম্যাচ করুন।

```
<category android:name="android.intent.category.CATEGORY_DEFAULT" />
```

একটি Intent ম্যাচ করুন যার সুস্পষ্ট ক্যান ক্যাটাগরি থাকে না।

```
<data android:mimeType="mime-type" />
```

একটি MIME টাইপ ম্যাচ করুন। শুধুমাত্র ঐ সকল MIME টাইপ সুনির্দিষ্ট
করুন যা আনার অ্যাপ সামলাতে পারবে

উদাহরণস্বরূপ, নিম্নোক্ত খন্ডাংশটি আপনাকে দেখাবে কীভাবে
একটি ইন্টেন্ট ফিল্টার যুক্ত করতে হয় যা একটি ভিডিও
com.example.android.nfctransfer.ViewActivity চালাবে।

```
<activity  
    android:name="com.example.android.nfctransfer.ViewActivity"  
    android:label="Android Beam Viewer" >  
    ...  
    <intent-filter>  
        <action android:name="android.intent.action.VIEW"/>  
        <category android:name="android.intent.category.DEFAULT"/>  
        ...  
    </intent-filter>  
</activity>
```

নোট: অ্যান্ডরয়েড বিম ফাইল টর্ান্সফার একটি ACTION VIEW ইন্টেন্টের
একমাত্র উদাহরণ। রিসিভিং ডিভাইসের অন্য অ্যাপও এই একশন সহ একটি
Intent পাঠাতে পারে। এই অবস্থা সামলানোর বিষয়টি Get the directory from a content URI
সেকশনে আলোচনা করা হয়েছে।

ফাইল পারমিশন রিকোয়েস্ট

ফাইল পড়তে যা অ্যান্ড্রয়েড বিম ফাইল টরান্সফার ডিভাইসে কপি করেছেন, পারমিশন `READ_EXTERNAL_STORAGE` রিকোয়েস্ট করুন। উদাহরণস্বরূপ:

```
<uses permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

আপনি যদি আপনার অ্যাপের নিজস্ব সেস্টারেজ এলাকাতে স্থানান্তরিত ফাইল কপি করতে চান, পরিবর্তে `WRITE_EXTERNAL_STORAGE` রিকোয়েস্ট করুন। `WRITE_EXTERNAL_STORAGE` `READ_EXTERNAL_STORAGE` কে অন্তর্ভুক্ত করে।

নোট: অ্যান্ড্রয়েড ৪.২.২ (এপিআই লেভেল ১৭) এর মতো, একমাত্র পারমিশন `READ_EXTERNAL_STORAGE` জারি রাখা হয় যদি ইউজার এটা করতে পছন্দ করে। প্ল্যাটফর্মের ভবিষ্যৎ সংস্করণ সকল ক্ষেত্রে এই পারমিশন চাইতে পারে। অগরু সব রুগরু হনো যাগ য়তা নিশ্চিত করতে, এখন পারমিশন রিকোয়েস্ট করুন, এটা পরোজনাীয় হওয়ার পূবেই।

যেহতু আপনার অ্যাপের এর ইন্টারনাল সেস্টারেজ এলাকার উপর নিয়ন্ত্রন আছে, আপনার ইন্টারনাল সেস্টারেজ এলাকায় একটি স্থানান্তরিত ফাইল কপি করতে আপনাকে রাইট পারমিশন রিকোয়েস্ট করার দরকার হবে না।

কপি কৃত ফাইলের জনয্ ডিরেক্টরী পাওয়া

অ্যান্ডরয়েড বিম ফাইল টরান্সফার রিসিভিং ডিভাইসে সর একটি
ডিরেক্টরীতে একটি একক টরান্সফারে সকল ফাইল কপি করে।

অ্যান্ডরয়েড বিম ফাইল টরান্সফার ন্যাটিভ ফ্রেশন কতর্ক পাঠানো
কন্টেন্ট Intent এর মধ্যে য্কার ইউআরআই পর্থম স্খানান্ তিরত ফাইলকে
নির্দেশ করে। কিন্তু আপনার অ্যাপ অ্যান্ডরয়েড বিম ফাইল
টরান্সফার ছাড়া অনয্ কান উ স্খেকেও একটি ACTION VIEW ইন্টেন্ট
রিসিভ করতে পারে। ইনকামিং Intent কীভাবে চালিত করতে হয় তা নিধরারন
করতে, আপনার দরকার এর স্কিম এবং অথরিটি পরীক্ষা করা।

ইউআরআই এর জনয্ স্কিম পেতে, Uri.getScheme() কল করুন। নিম্নোক্ত কোড
খন্ডাংশটা আপনাকে দেখায় কীভাবে স্কিম নিধরারণ করতে হয় এবং
সেইমতে ইউআরআই চালিত করে:

```
public class MainActivity extends Activity {  
    ...  
    // A File object containing the path to the transferred files  
    private File mParentPath;  
    // Incoming Intent  
    private Intent mIntent;  
    ...  
    /*  
    * Called from onNewIntent() for a SINGLE_TOP Activity  
    * or onCreate() for a new Activity. For onNewIntent(),  
    * remember to call setIntent() to store the most  
    * current Intent  
    */  
    private void handleViewIntent() {  
        ...  
        // Get the Intent action  
        mIntent = getIntent();  
        String action = mIntent.getAction();  
        /*  
        * For ACTION_VIEW, the Activity is being asked to display data.  
        * Get the URI.  
        */  
        if (TextUtils.equals(action, Intent.ACTION_VIEW)) {  
            // Get the URI from the Intent  
            Uri beamUri = mIntent.getData();  
            /*  
            * Test for the type of URI, by getting its scheme value  
            */  
            if (TextUtils.equals(beamUri.getScheme(), "file")) {  
                mParentPath = handleFileUri(beamUri);  
            } else if (TextUtils.equals(  
                beamUri.getScheme(), "content")) {  
                mParentPath = handleContentUri(beamUri);  
            }  
        }  
        ...  
    }  
    ...  
}
```

একটি ফাইল ইউআরআই থেকে ডিৱেক্টরী লাভ

যদি ইনকামিং Intent একটি ফাইল ইউআরআই ধারণ করে, ইউআরআই ফাইলের একটি সম্পূর্ণ ফাইল নেম ধারণ করে, যার মধ্যে পূর্ণ ডিরেক্টরী প্যাথ এবং ফাইল নেম। অ্যান্ডরয়েড বিম ফাইল ট্রান্সফার এর জন্য, ডিরেক্টরী প্যাথ অন্য স্থানান্তরিত ফাইলের অবস্থান নির্দেশ করে, যদি কোন একটি থেকে থাকে। ডিরেক্টরী প্যাথ পেতে, ইউআরআই এর প্যাথ অংশ নিন যা এই file: প্রিফিক্স ছাড়া ইউআরআই এর সকল কিছু ধারণ করে। প্যাথ অংশ থেকে একটি File তৈরী করুন, তারপর File এর প্যাথের প্যাথ অর্জন করুন:

```
...
public String handleFileUri(Uri beamUri) {
    // Get the path part of the URI
    String fileName = beamUri.getPath();
    // Create a File object for this filename
    File copiedFile = new File(fileName);
    // Get a string containing the file's parent directory
    return copiedFile.getParent();
}
...
```

একটি কন্টেইনার ইউআরআই থেকে ডিডিরেক্টরী লভ করা

যদি ইনকামিং Intent একটি কন্টেইনার ইউআরআই ধারণ করে, ইউআরআই একটি ডিডিরেক্টরীর এবং ফাইল নেম যা MediaStore কন্টেইনার পর্ভাইডারে স্টোর করা আছে তার প্রতিভিন্দেদর্শ করে। ইউআরআই এর অথরিটি ভ্যালু পরীক্ষা করার মাধ্যমে এর MediaStore জনম্ব আপনি একটি কন্টেইনার সনাক্ত করতে পারেন। MediaStore এর জনম্ব একটি কন্টেইনার ইউআরআই অ্যান্ডরয়েড বিম ফাইল ট্রান্সফার বা অন্য অ্যাপ থেকে আসতে পারে, কিন্তু উভয় ক্ষেত্রেই কন্টেইনার ইউআরআই এর জনম্ব আপনি একটি ডিডিরেক্টরী এবং ফাইল নেম উদ্ধারণ করতে পারেন।

MediaStore এর চেয়ে একটি কন্টেইনার পর্ভাইডারের জনম্ব আপনি একটি কন্টেইনার ইউআরআই ধারণ করা একটি ইনকামিং ACTION VIEW ইন্টেন্সিটি গৃহণ করতে পারেন। এক্ষেত্রে, কন্টেইনার ইউআরআই MediaStore অথরিটি ভ্যালু ধারণ করেনা এবং কন্টেইনার ইউআরআই সাধারণত একটি ডিডিরেক্টরীর প্রতিভিন্দেদর্শ করেনা।

নোট: অ্যান্ডরয়েড বিম ফাইল ট্রান্সফার এর জনম্ব, ACTION VIEW ইন্টেন্সিটি এর মধ্যে আপনি একটি কন্টেইনার ইউআরআই গৃহণ করেন যদি প্রথম ইনকামিং ফাইলের "audio/*", "image/*", বা "video/*" এর একটি MIME টাইপ থাকে, এটা ভিন্দেদর্শ করে যে এটা মিডিয়া-সম্পৃক্ত। অ্যান্ডরয়েড বিম ফাইল ট্রান্সফার মিডিয়া ফাইলের ইন্ডেক্স করে এটা ডিডিরেক্টরীর রানিং মিডিয়া স্ক্যানার দ্বারা স্থানান্তর করে যেখানে এটা স্থানান্তরিত ফাইল সেট করা করে। মিডিয়া স্ক্যানার এর রেজাল্ট (ফলাফল) MediaStore কন্টেইনার পর্ভাইডারের রাইট করে এটা তারপর প্রথম ফাইলের জনম্ব একটি কন্টেইনার ইউআরআই অ্যান্ডরয়েড বিম ফাইল ট্রান্সফারে ফেরত পাঠায়। এই কন্টেইনার ইউআরআই হচ্ছে সেটাবেটা আপনি নোটিফিকেশন Intent এ গৃহণ করেছেন। প্রথম ফাইলের ডিডিরেক্টরী পেতে, কন্টেইনার ইউআরআই ব্যবহার করে MediaStore থেকে এটা আপনি উদ্ধারণ করতে পারেন।

কন্টেইনার পর্ভাইডার নিবনল্লয় করা

নিবনল্লয় করা করতে, আপনিল যদি কন্টেইনার ইউআরআই থেকে একটি ফাইল ডিরেক্টরী উদ্ধার করতে চান, ইউআরআই এর অথরিটি পেতে `Uri.getAuthority()` কল করে ইউআরআই এর সাথে সংশ্লষ্ট কন্টেইনার পর্ভাইডার নিবনল্লয় করা। ফলাফলের দুইটা সম্ভাব্য ভয়ালু আছে:

`MediaStore.AUTHORITY`

ইউআরআই একটি ফাইলের জন্য বা ফাইল `MediaStore` দ্বারা অনুল্লিখিত। `MediaStore` থেকে পূর্ণ ফাইল নেম উদ্ধার করুন এবং ফাইল নেম থেকে ডিরেক্টরী অর্জন করুন।

অন্য যে কোন অথরিটি ভয়ালু

অন্য কন্টেইনার পর্ভাইডার থেকে একটি কন্টেইনার ইউআরআই। কন্টেইনার ইউআরআই এর সাথে সংশ্লষ্ট ডাটাপর্দশর্ম করে কন্টেইনার ফাইল ডিরেক্টরী পায়না।

একটি `MediaStore` কন্টেইনার ইউআরআই এর জন্য ডিরেক্টরী পেতে, একটি অনুল্লিখিত রান করুন যা `Uri` আগুরমেন্ট এর জন্য ইনকামি কন্টেইনার ইউআরআই এবং পর্ভেজকশন এর জন্য কলাম `MediaColumns.DATA` নির্দর্শক করে। ইউআরআই ইকতরক পিরেবিশত ফাইলের জন্য ফেরত আসা `Cursor` পূর্ণ পথ এবং নাম ধারণ করে। এই পথ সকল অনুল্লিখিত ফাইল ধারণ করে যা অ্যান্ডরয়েড বিম ফাইল টরান্সফার মাত্রই ডিভাইসে কপি করেছে।

নিম্নোক্ত খন্ডাংশটি আপনাকে দেখায় কীভাবে কন্টেইনার ইউআরআই এর অথরিটি পরীক্ষা করতে হয় এবং স্থানান্তরিত ফাইলের জন্য পথ এবং নেম উদ্ধার করে:

```

...
public String handleContentUri(Uri beamUri) {
    // Position of the filename in the query Cursor
    int filenameIndex;
    // File object for the filename
    File copiedFile;
    // The filename stored in MediaStore
    String fileName;
    // Test the authority of the URI
    if (!TextUtils.equals(beamUri.getAuthority(), MediaStore.AUTHORITY)) {
        /*
         * Handle content URIs for other content providers
         */
        // For a MediaStore content URI
    } else {
        // Get the column that contains the file name
        String[] projection = { MediaStore.MediaColumns.DATA };
        Cursor pathCursor =
            getContentResolver().query(beamUri, projection,
                null, null, null);
        // Check for a valid cursor
        if (pathCursor != null &&
            pathCursor.moveToFirst()) {
            // Get the column index in the Cursor
            filenameIndex = pathCursor.getColumnIndex(
                MediaStore.MediaColumns.DATA);
            // Get the full file name including path
            fileName = pathCursor.getString(filenameIndex);
            // Create a File object for the filename
            copiedFile = new File(fileName);
            // Return the parent directory of the file
            return new File(copiedFile.getParent());
        } else {
            // The query didn't work; return null
            return null;
        }
    }
}
}
...

```

একটি কন্টেইনারে টপ লেভেল আইডার থেকে ডাটা উদ্ধার করা সম্পর্কে আরও জানতে, Retrieving Data from the Provider সেকশনটি দেখুন।

মালি টিমিডিয়া সহ অ্যাপ তৈরী

(<http://developer.android.com/training/building-multimedia.html>)

এই ক্লাস আপনাকে আপনাকে শেখাবে কীভাবে সমৃদ্ধ মালি টিমিডিয়া অ্যাপ তৈরী করতে হয় এবং ইউজার যা চায় তার মতো করে আচরণ করে।

১. অডিও প্লেবয়্যাক বস্‌স্থাপনা

কীভাবে হাউন্ডরুওয়ার অডিও ক্রিপের্সের সপনস করতে হয়, যখন অডিও প্লে করা হয় তখন অডিও ফোকাস রিকোয়েস্ট করা এবং অডিও ফোকাসের পরিবর্তনে যথাযথ রেসপন্স করা।

1. আপনার অ্যাপের ভিলিউম এবং প্লেবয়্যাক নিয়ন্ত্রণ
2. অডিও ফোকাস বস্‌স্থাপনা করা
3. অডিও আউটপুট হাউন্ডরুওয়ারের সাথে আদান পরদান করা

২. ফ্রেটাকন্ট্রোলার করা

ইউজারের ডিভাইসে ফ্রেটাতুলতে বা সরাসরি কন্ট্রোলার হাউন্ডরুওয়ার নিয়ন্ত্রণ এবং আপনার নিজস্ব কন্ট্রোলার অ্যাপ তৈরী করার ক্ষেত্রে কীভাবে বিদ্যমান কন্ট্রোলার অ্যাপকে কাজে লাগানো যায়।

1. ফ্রেটানৈয়া
2. ভিডিও রেকর্ড করা
3. কন্ট্রোলার নিয়ন্ত্রণ করা

৩. কন্টেন্টি পিরন্ট করা

কীভাবে আপনার অ্যাপ থেকে ফ্রেটাতা, HTML ডকুমেন্ট, এবং কাস্টম ডকুমেন্ট পিরন্ট করতে হয়।

1. ফ্রেটাতা
2. HTML ডকুমেন্ট
3. কাস্টম ডকুমেন্ট

অডিও প্লেব্যা্ক ব্যবস্থাপনা

(<http://developer.android.com/training/managing-audio/index.html>)

যদি আপনার অ্যাপ অডিও প্লে করে, এটা গুরুত্বপূর্ণ যে আপনার ইউজার অডিওকে কীভাবে তরীতে তৈরী করে তৈরী করে। একটি অসাধারণ ইউজার এক্সপেরিয়েন্স নিশ্চিত করতে, এটা গুরুত্বপূর্ণ যে একটি অ্যাপ একই সময়ে অডিও প্লে না করাটা নিশ্চিত করতে আপনার অ্যাপ অডিও ফ্যাকাস পরিচালনা করে।

এই ক্লাসে শেষে আপনি অ্যাপস তৈরীতে সমর্থ হবেন যা হার্ডওয়্যার অডিও কিপেরেসেস এর সপনস করে, যা অডিও প্লে সময় অডিও ফ্যাকাস রিকোয়েস্ট করে এবং সিস্টেম বা অন্য অ্যাপ লিকেশন কতটুকু অডিওর কোন পরিবর্তনকে যথাযথভাবে সপনস করে।

অনুশীলনীসমূহ

আপনার অর্থে পর ভিলিউম এবং পেম্বলবয়্যাক নিয়ন্ত্রণ

শিক্ষণ হাড়র ওয়ার বা সফটওয়ার ভিলিউম কন্ট্রোল বয়্যবহার করে আপনি কীভাবে নিশ্চিত করতে বন আপনার ইউজার অর্থে পর ভিলিউম নিয়ন্ত্রণ করতে পারবে এবং পেম্বল, স্টপ, পজ, স্কপ এবং পেরিভিয়াস পেম্বলবয়্যাক কীভাবে আচ্ছ।

অডিও ফোকাস বয়্যবস্থাপনা করা

একটি অর্থে পর সম্ভাব্য অডিও পেম্বলর ক্ষেত্রে এটি চিন্তা করাটা গুরুত্বপূর্ণর তাদেব কীভাবে পারস্পরিকভাবে কাজ করা উচিত। একই সময়ে পরে তৎকটামিউজিক অর্থে পর একসাথে পেম্বল করা পরিহার করতে, অর্থাৎ ডেড অডিও পেম্বলবয়্যাকে পরিমিত করতে অডিও ফোকাস বয়্যবহার করে। শিক্ষণ কীভাবে অডিও ফোকাস রিকোয়েস্ট করতে হয়, অডিও ফোকাসের একটি কক্ষিত কেনজর দিতে হয়, এবং কীভাবে রেসপন্স করতে হয় যখন এটি ঘটে।

অডিও আউটপুট হাড়র ওয়ারের সাথে কাজ করা অডিও বিভিন্ন উৎস থেকে পেম্বল হতে পারে। শিক্ষণ কীভাবে খুঁজে বের করতে হয় কীভাবে অডিও পেম্বল হচ্ছে এবং পেম্বলবয়্যাকে সময় বিচিহ্ন হওয়া হ্যান্ডেস্ট কীভাবে সামলাতে হয়।

আপনার অ্যাপের ভলিউম এবং প্লেবয়্যাক নিয়ন্ত্রণ

(<http://developer.android.com/training/managing-audio/volume-playback.html>)

একটি ভোলাইউজার এক্সপেরিয়েন্স একটি অনুমেয় বিষয়। আপনার অ্যাপ যদি মিডিয়া প্লে করে এটা গুরুত্বপূর্ণ যে আপনার ইউজার তাদের ডিভাইস, ব্লুটুথ, বা হেডফোনের হার্ডওয়্যার বা সফটওয়্যার ভলিউম কন্ট্রোল ব্যবহার করে আপনার অ্যাপের ভলিউম নিয়ন্ত্রণ করতে পারে।

একইভাবে, আপনার অ্যাপ কতটুকু ব্যবহৃত অডিও সিস্টেমের কাছাকাছি প্লে, স্টপ, পজ, স্কিপ এবং পেরিভিয়াস মিডিয়া প্লেবয়্যাক কিগুলা তাদের নিজ নিজ কার্য সম্পাদন করতে পারে এবং এগুলো কাছাকাছি থাকে।

কোন অডিও সিস্টার্ম বয্‌বহার করে ত হবে তা চিহ্নিত করা

একটি অনুষ্টেয় অডিও এক্সপেরিয়েন্স তৈরীর পর্থম ধাপ হ্‌ছে আপনার
অয্‌প কোন অডিও সিস্টার্ম বয্‌বহার করে তা বেঝা।

অয্‌নডর্‌য়েড মিউজিক, এলামর্‌, নেটিটিফিকেশন, ইনকামিঙ্‌কলিঙ্‌গার,
সিসেস্টম সাউন্‌ড, ইন-কল ভলিউম এব্‌ DTMF টোন প্‌ল করার জন্য্‌ একটি
আলাদা অডিও সিস্টার্ম বজায় রাখে। এটা করা হয় পর্থমিকভাবে পর্‌তিটা
সিস্টার্ম এর ভলিউম সবাধীনভাবে ইউজারকে নিয়ন্‌তরণ করে তি দেত।

এই অধিকাংশ সিস্টার্ম সিসেস্টম ইন্‌ভেন্টসীমাবদধ, সুতরাং যদি না
আপনার অয্‌প একটি বদলি এলামর্‌ কলকনা হয়, আপনি পর্থায়
নিশ্চিতভাবে STREAM MUSIC সিস্টার্ম বয্‌বহার করে আপনার অডিও প্‌ল
করবেন

আপনার অ্যাপে পর অডিও ভলিউম নিয়ন্ত্রণ করতে হাডর্ ওয়ার ভলিউমিক ব্যবহার করুন

বাই ডিফল্ট, ভলিউম কন্ট্রোল পের্স করা সিকর্য অডিও স্ট্রমকে পরিমিত করে। যদি আপনার অ্যাপ এখন কৈনিক ছু পে ল করে না থাকে, ভলিউমিক তে চাপ দিয়ে রিং এর ভলিউম সমনব্য করুন।

আপনি যদি একটি টেগম বা মিমিউজিক অ্যাপ পেয়ে থাকেন, তখন ভালো সম্ভাবনা থাকে যে যখন ইউজার ভলিউমিক হিট করে তার টেগম বা মিমিউজিকের ভলিউম নিয়ন্ত্রণ করতে চায়, এমনকি যদি তারাবতর্মানে গান বা চলিত টেগম লোকশনে মিমিউজিক নেই এমন হয়।

আপনি ভলিউমিক পের্স এর জন্য চেষ্টা করতে এবং শুনতে পর্লুব্ধ হতে পারেন এবং আপনার অডিও স্ট্রমের ভলিউম এই উপায়ে পরিবর্তন করতে পারেন। এটা করে বন না। অ্যান্ডরয়েড অডিও স্ট্রম যেটা আপনি নিদির্ষ্ট করেছেন তাতে ভলিউমিক পের্সে সস পরিচালনা করতে উপকারী `setVolumeControlStream()` মেথড পর্দান করে।

অডিও স্ট্রমি চিহ্ন ত থাকতে যা আপনার অ্যাপ লিকেশন ব্যবহার করতে থাকবে, আপনার এটাকে ভলিউম স্ট্রম টারগেট হিসাবে সেট করা উচিত। আপনার উচিত আপনার অ্যাপের লাইফসাইকেলের পর্থমিদিকে এই কল করে ফেলা- কারণ আপনার শুধু এটাকে একবার একটি টিভিটি লাইফসাইকেল সময়কালে কল করা দরকার, আপনার উচিত `onCreate()` মেথডের (Activity বা Fragment এর যা আপনার মিমিডয়ান নিয়ন্ত্রন করে) মধ্যে সাধারণভাবে এটাকে কল করা। এটা নিশ্চিত করে যে যখনই আপনার অ্যাপ দৃশ্যমান হবে, ভলিউম কাযর্কর্মে কৈনয়ন্ত্রন করে ইউজারের আকাঙ্খা অনুসারে।

```
setVolumeControlStream(AudioManager.STREAM_MUSIC);
```

এই বিন্দু থেকে, ডিভাইসের ভলিউমিক পের্স করা অডিও স্ট্রমকে পর্তাবিত করে যা আপনি নিদির্ষ্ট করেছেন (এক্ষেতর্ “মিমিউজিক”) যখনই টার্গেট একটি টিভি বা ফর্াগমেন্ট দৃশ্যমান হয়।

আপনার অ্যাপের অডিও প্লেবয়্যাক নিয়ন্ত্রণ করতে হাডর্ ওয়ার প্লেবয়্যাক নিয়ন্ত্রণিক বয়্যবহার করুন

মিডিয়া প্লেবয়্যাক বাটন, যেমন প্লে, পজ, স্টপ, স্কিপ এবং রিপিট অ্যাকশন
কিছু হ্যান্ডলেসেট আছে এবং অনেক স্যুইচ বাটন বর্তমানে হ্যান্ডলেসেটও আছে।
যখন একজন ইউজার এই হাডর্ ওয়ারিক এর একটি ট্যাপের স্ক্রিন, সিস্টেমটি
ACTION MEDIA_BUTTON একশন সহকারে একটি ইন্টেন্ট সম্ভার করে।

মিডিয়া বাটন ক্লিকের সপক্ষে করতে, আপনার প্রয়োজন আপনার
মেনিফেস্টে একটি BroadcastReceiver রেজিস্টার করা যা একশন বর্ডকাস্ট
শুনে থাকে যা নিম্নে দেয়া আছে।

```
<receiver android:name=".RemoteControlReceiver">
  <intent-filter>
    <action android:name="android.intent.action.MEDIA_BUTTON" />
  </intent-filter>
</receiver>
```

রিসিভার বাস্তুবায়ন তার নীচে জরুরি হওয়া দরকার যে কীবোর্ড কাস্ট
ঘটতে পেরে স্ক্রিন হয়। Intent এটা EXTRA_KEY_EVENT কি'র অধীনে অন্তর্ভুক্ত
করে, যখন KeyEvent ক্লাস একটি তালিকা KEYCODE MEDIA স্ট্যাটিক কনস্টেন্ট
অন্তর্ভুক্ত করে যা পরিত্রাস্তাবয়্য মিডিয়া বাটন পরিবেশন করে,
যেমন KEYCODE MEDIA_PLAY_PAUSE এবং KEYCODE MEDIA_NEXT।

নিম্নের খন্ডাংশটি দেখায় যে কীভাবে মিডিয়া বাটন পেরে সড স্ক্রিনে
ফেলতে হয় এবং সে অনুসারে মিডিয়া প্লেবয়্যাক কে পরে ভাবিত করতে হয়।

```
public class RemoteControlReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        if (Intent.ACTION_MEDIA_BUTTON.equals(intent.getAction())) {
            KeyEvent event = (KeyEvent) intent.getParcelableExtra(Intent.EXTRA_KEY_EVENT);
            if (KeyEvent.KEYCODE_MEDIA_PLAY == event.getKeyCode()) {
                // Handle key press.
            }
        }
    }
}
```

কারণ বহুমুখি (মাল্টিপল) অ্যাপলিকেশন মিডিয়া বাটন পেরে সেস
এরজন্য শুনতে চাইতে পারে, আপনাকে অবশ্যই পেরে আগের মেট্রিক্যাল
নিয়ন্ত্রণ করতে হবে যখন আপনার অ্যাপের মিডিয়া বাটন পেরে স্ক্রিন
গরহন করা উচিত।

AudioManager বয়্যবহার করে আপনার মিডিয়া বাটন ইন্ডেন্ট রিসিভার
রেজিস্টার এবং অডিও রেজিস্টার করতে নীচের কেডটি আপনার অ্যাপের
মধ্যে বয়্যবহার হতে পারে। যখন রেজিস্টার করা হয়, আপনার বর্ডকাস্ট
রিসিভার হেচু সকল মিডিয়া বাটন বর্ডকাস্টের এক্সক্লুসিভ
রিসিভার।

```
AudioManager am = mContext.getSystemService(Context.AUDIO_SERVICE);  
...  
  
// Start listening for button presses  
am.registerMediaButtonEventReceiver(RemoteControlReceiver);  
...  
  
// Stop listening for button presses  
am.unregisterMediaButtonEventReceiver(RemoteControlReceiver);
```

সাধারণভাবে, অ্যাপের উচিত তাদের অধিকাংশ রিসভার আনডেজিস্টার করা যখনই তারানিস্কর্য বা অদৃশ্য হয় (যেমন, onStop() কলব্যাংক কর সময়কালে)। কিন্ত এটা মিডিয়াপ্লব্যাংক অ্যাপের জন্য সহজনয়-কাযরত, মিডিয়াপ্লব্যাংক বাটনে রেসপন্স করা হেচ্ছ সবচেয়ে গুরুত্বপূর্ণর যখন আপনার অ্যাপিলেকশন দৃশ্যমান নয় এবং সেইজন্য অন-িস্করন ইউজার ইন্টারফেস কতৃক নিয়ন্তরত হেতপারেনা।

একটি ভালে অ্যাপেরাচ হেচ্ছ মিডিয়া বাটন ইভেন্ট রিসভার রেজিস্টার এবং আনডেজিস্টার করা যখন আপনার অ্যাপিলেকশন অডিও ফোকাসটি অজরন করে বা হারিয়ে ফেলে। এটা পরবর্তী অনুশীলনীতে বিস্তারিতভাবে আলাচনা করা হয়েছে।

অডিও ফোকাস ব্যবস্থাপনা

(<http://developer.android.com/training/managing-audio/audio-focus.html>)

বহুমুখী (মাল্টিপল) অ্যাপস দিয়ে সম্ভাব্য অডিও প্লে এর কেসে তরতারাকীভাবে পারস্পরিকভাবে কাজ করে সেই বিষয়ে চিন্তা করা গুরুত্বপূর্ণ। পরিতীমিউজিক অ্যাপ একই সময়ে প্লে করা পরিহার করতে, অ্যান্ডরয়েড অডিও প্লে ব্যব্যাকে পরিমিত করতে অডিও ফোকাস ব্যবহার করে- শুমাত্র সেই অ্যাপ যা অডিও ফোকাস ধারণ করে তার অডিও প্লে করা উচিত।

আপনার অ্যাপের অডিও প্লে শুরুর করার পূর্বেই এর অডিও ফোকাস রিকোয়েস্ট এবং রিসভ করা উচিত। একইভাবে এটার জানা উচিত অডিও ফোকাস লস হলে কীভাবে তা শোনা উচিত এবং এটা ঘটলে কীভাবে যথাযথভাবে রেসপন্স করতে হয়।

অডিও ফোকাস রিকোয়েস্ট

আপনার অ্যাপের কোন অডিও প্লে শুরুর করার পূর্বেই এটার উচিত সিস্টারমে যেটা বয় ব্যবহার হবে তার জন্যে জনস্বার্থে অডিও ফোকাস ধারণ করা উচিত। `requestAudioFocus()` একলি দিয়ে এটা সম্পূর্ণ হবেন যে `AUDIOFOCUS_REQUEST_GRANTED` ফেরত দেয় যদি আপনার রিকোয়েস্ট সফল হয়।

আপনাকে অবশ্যই ইনিদিরষ্টি করতে হবে কোন সিস্টারম আপন বয় ব্যবহার করতে যাচ্ছেন এবং আপনি কোনটা চান, অস্থায়ী বা স্থায়ী অডিও ফোকাস। অস্থায়ী ফোকাস রিকোয়েস্ট করা হয় যখন আপনি অল্প সময়ের জন্যে অডিও প্লে করতে চান (উদাহরণস্বরূপ যখন নোটিফিকেশন ইনস্টারাকশন প্লে করা হয়)। স্থায়ী অডিও ফোকাস রিকোয়েস্ট করা হয় যখন আপনি অনুরোধের বিষয়ের জন্যে অডিও প্লে করার পরিকল্পনা করবেন (উদাহরণস্বরূপ, যখন মিউজিক প্লে করা হয়)।

নিম্নোক্ত খন্ড চিত্রটি মিউজিক অডিও সিস্টারমে স্থায়ী অডিও ফোকাস রিকোয়েস্ট করে। আপনার প্লে বয়াক শুরুর করার পূর্বেই আপনাকে অবশ্যই তা ক্রমিকভাবে অডিও ফোকাস রিকোয়েস্ট করতে হবে, যেমন যখন ইউজার পরবর্তী গেম লেভেল শুরুর করার জন্যে প্লে বা বয়াক গরুন্ড মিউজিক পের্স করে।

```
AudioManager am = mContext.getSystemService(Context.AUDIO_SERVICE);
...

// Request audio focus for playback
int result = am.requestAudioFocus(afChangeListener,
    // Use the music stream.
    AudioManager.STREAM_MUSIC,
    // Request permanent focus.
    AudioManager.AUDIOFOCUS_GAIN);

if (result == AudioManager.AUDIOFOCUS_REQUEST_GRANTED) {
    am.unregisterMediaButtonEventReceiver(RemoteControlReceiver);
    // Start playback.
}
```

ইতিমধ্যেই আপনি যদি প্লে বয়াক সম্পূর্ণ করেন নিশ্চিত হতে `abandonAudioFocus()` কল করুন। এটা সিস্টেমটিকে জ্ঞাপন করে যে আপনি আর ফোকাস চান না এবং `AudioManager.OnAudioFocusChangeListener` কে আনরেজিস্টার করে। অস্থায়ী ফোকাস পরিত্যক্তের ক্ষেত্রে, এটা যে কোন বাধাগরুস্থ অ্যাপকে প্লে বয়াক জারির রাখতে অনুমোদন দেয়।

```
// Abandon audio focus when playback complete
am.abandonAudioFocus(afChangeListener);
```

যখন রিকোয়েস্ট করা অস্থায়ী অডিও ফোকাস আপনার কাছে অতিরিক্ত অপশন হিসাবে থাকে: যেটাই হোক আপনি "ducking" সিকর্য করতে চাইবেন। সবার্ভাবিকভাবে, যখন সদাচরন করা অডিও অ্যাপ অডিও ফোকাস লস করে এটা তা ক্রমিকভাবে এটার প্লে বয়াক নিরব করে দেয়। একটি অস্থায়ী অডিও ফোকাস যা ডালিক করতে দেয় সেটার রিকোয়েস্ট করার মাধ্যমে আপনি অন্য অডিও অ্যাপকে বলেন যে প্লে বজায় রাখতে এটা তাদের জন্যে গরুহনযোগ্য, যতক্ষণ না ফোকাসটি তাদের কাছে ফেরত আসে তারাতাদের ভলিউম নীচুরাখে।

```
// Request audio focus for playback
int result = am.requestAudioFocus(afChangeListener,
    // Use the music stream.
    AudioManager.STREAM_MUSIC,
    // Request permanent focus.
    AudioManager.AUDIOFOCUS_GAIN_TRANSIENT_MAY_DUCK);

if (result == AudioManager.AUDIOFOCUS_REQUEST_GRANTED) {
    // Start playback.
}
```

ডায়ালগ বক্সে শেষভাবে অর্থাৎ সর্বশেষ জনস্বার্থে উপযোগী যাতে কেতে কেতে অডিও
সিস্টেম বন্ধ করে, যেমন শব্দ বর্ণনা গাঢ় ডার্ক আইডিওর কন্ট্রোল।

যখন অন্য অর্থাৎ অডিও ফোকাসের কয়েকটি করে যেভাবে উপরে আলোচনা
করা হয়েছে, এটা যখন ফোকাসের কয়েকটি করা হয় তখন আপনি যে শব্দ তাতা
বেরি জিস্টার করেছেন তাই দরদর দরদর গরহণ করা স্থায়ী এবং অস্থায়ী
(অডিও ফোকাসের মধ্যে একটা পছন্দ করে।

অডিও ফোকাসের ক্ষতি/হারিয়ে ফেলাকে বস্‌বস্‌ত্ব পনাকরা

যদি আপনার অ্যাপ অডিও ফোকাসের কয়েস্ট করতে পারে, এটা অনুসরণ করে যে এটা পালকর্মে লস করে যখন অন্য অ্যাপ এটাকে বিরেকোয়েস্ট করে। আপনার অ্যাপ অডিও ফোকাসের লস কীভাবে বেরসপন্স করে তা নিভর করে ঐ লসের আচরনের উপর।

অডিও ফোকাসের `onAudioFocusChangeListener()` কলবস্‌বাক্‌মে খডেশর্‌াতা যা আপনি রেজিস্টার করেছেন তাকে পরিবর্তন করে যখন বিরেকোয়েস্ট অডিও ফোকাস একটি টমানদন্ড বস্‌বহার করে যা ফোকাস চেঞ্জ ইভেন্ট আলাচনাকরে। নিদির্ষ্‌টভাবে, সম্ভাবস্‌ ফোকাস লস ইভেন্টে ফোকাস বিরেকোয়েস্ট টাইপকে পর্‌তিফলিত করে-পূর্ববর্তর্‌ী সেকশন থেকে-স্থায়ী লস, অস্থায়ী লস এবং অস্থায়ী সাঁথে ডাক্‌ পারমিটেড।

সাধারণভাবে বলতে, অডিও ফোকাসের একটি অস্থায়ী লসের উচিত আপনার অ্যাপের অডিও স্টর্‌মিনিশ্‌চুপ করিয়ে দি়ে এর উপর পর্‌ভাব ফেলা, কিন্তু অন্যথায় একই অবস্থাবহার রেখে। আপনার উচিত অডিও ফোকাসের পরিবর্তন মনিটর করা অবস্‌হত রাখা এবং পেপ্লবস্‌াক পূরণায় শুরুর করতে পর্‌স্‌তুত হন যেখানে এটা পজ করা হয়েছিল আপনি পূর্‌নায় ফোকাস অজর্‌ন করবেন।

যদি অডিও ফোকাস লস স্থায়ী হয়, এটা গ্‌হিত হয়ে অন্য অ্যাপিলেকশন এখন অডিও শুনতে শুরুর করে এবং আপনার অ্যাপের কাযর্‌করীভাবে নিজেই এটার শেষ করা উচিত। বাস্‌তিবকপক্ষে, ওটার মানে পেপ্লবস্‌াক স্টপ করা, মিডিয়া বাটন লিসেনারের রিমুয়ন্ড করা-নতুন অডিও পেপ্লয়ারকে এক্সক্লুসিভভাবে ওই ইভেন্ট চালিত করতে অনুমোদন করে-এবং আপনার অডিও ফোকাস ত্যাগ করে। যে সময়ে, অডিও পেপ্ল পূরণায় শুরুর করার পূর্ববর্‌ই পর্‌য়োজন হবে এমন একটি ইউজার একশন আপনি আশা করতে পারেন(আপনার অ্যাপে পেপ্লপের্‌স করা)।

নিচের কেডখন্‌ডিতর্‌টিতে, আমরা পেপ্লবস্‌াক বাআমাদে রিমিডিয়া পেপ্লয়ার অব্‌জেক্ট পজ দেয় যদি অডিও লস অস্থায়ী হয় এবং আমরা এটা পূরণায় শুরুর করি যখন আমরা ফোকাস পূরণায় অজর্‌ন করি। যদি লস স্থায়ী হয়, এটা আমাদে রিমিডিয়া বাটন ইভেন্টের রিসিভার আনরেজিস্টার করে এবং অডিও ফোকাস এর পরিবর্তন মনিটরি করা বন্ধ করে।

```
OnAudioFocusChangeListener afChangeListener = new OnAudioFocusChangeListener() {
    public void onAudioFocusChange(int focusChange) {
        if (focusChange == AudioManager.AUDIOFOCUS_LOSS_TRANSIENT
            // Pause playback
        } else if (focusChange == AudioManager.AUDIOFOCUS_GAIN) {
            // Resume playback
        } else if (focusChange == AudioManager.AUDIOFOCUS_LOSS) {
            am.unregisterMediaButtonEventReceiver(RemoteControlReceiver);
            am.abandonAudioFocus(afChangeListener);
            // Stop playback
        }
    }
};
```

অডিও ফোকাসের একটি অস্থায়ী লসের ক্ষেতর্‌ ডাক্‌ অনুমিত পর্‌াপ্‌ত, পেপ্লবস্‌াক পজ নাকরে আপনি পরিবর্তর্‌ “ডাক” করতে ডারেন।

Duck! / ডাক!

ডাকিং হচ্ছে অন্য অ্যাপ থেকে অস্থায়ী অডিও বান্ধতে আপনার অডিও সিস্টারম্‌ আউটপুট ভলিউম নীচু করার একটি পদ্ধতি। যা, আপনার নিজস্ব অ্যাপ লিকেশন থেকে কৈকান রকম ব্যতীত ছাড়াই অডিও সহজভাবে শুনতে।

নীচের কোডের খিন্ডিত চিত্রটি আমাদের ডিফল্ট প্লেয়ার অবজেক্ট ভলিউম কমিয়ে দেয় যখন আমরা অস্থায়ীভাবে ফোকাস লস করি, তারপর এটাকে এর পূর্ববর্তরী লেভেলে ফেরত আনি যখন আমরা ফোকাস পুনরায় অর্জন করি।

```
OnAudioFocusChangeListener afChangeListener = new OnAudioFocusChangeListener() {  
    public void onAudioFocusChange(int focusChange) {  
        if (focusChange == AUDIOFOCUS_LOSS_TRANSIENT_CAN_DUCK) {  
            // Lower the volume  
        } else if (focusChange == AudioManager.AUDIOFOCUS_GAIN) {  
            // Raise it back to normal  
        }  
    }  
};
```

অডিও ফোকাসের লস হচ্ছে পদ্ধতিটির যা করতে সবচেয়ে পুরনু এবং পূর্ণ বর্ডকাস্ট, কিন্ত একমাত্র নয়। সিস্টেম ইউজারের অডিও এক্সপেরিয়েন্সের পরিবর্তনের পদ্ধতি সতর্ক করে তাকে একটি ইন্টেন্ট বর্ডকাস্ট করে। পরবর্তরী অনুশীলনী দেখাবে ইউজারের সাবস্ক্রিপ্ট এক্সপেরিয়েন্স উন্নত করতে কীভাবে তাদের মনিটর করা হয়।

অডিও আউটপুট হাউন্ডলিং ওয়ার ব্যবস্থাপনা

(<http://developer.android.com/training/managing-audio/audio-output.html>)

ইউজারের কিছু বিকল্প আছে যখন এটা তাদের অ্যান্ড্রয়েড ডিভাইস থেকে অডিও উপভোগ করতে আসে। অধিকাংশ ডিভাইসের বিল্ট-ইন স্পকার, তারসহেড সেটের জন্য হেডফোন জ্যাক এবং আরও অনেক কিছু আছে, এছাড়াও ব্লুটুথ কানেনকিটিভিটি এবং A2DP অডিও সমর্থন বৈশিষ্ট্য দেয়।

চেক করুন কোন হ্যাডসেট ওয়ার ব্যবহৃত হচ্ছে

কীভাবে আপনার অ্যাপ আচরণ করবে তা নির্ভর করে যে কোন হ্যাডসেট ওয়ার এর আউটপুট রুটিন ঠিক করা হয়েছে।

নির্ধারণ করা করতে পারেন যে যদি বর্তমানে অডিও টি ডিভাইস সিঙ্গলার, তার সহ হেডসেট, বা সংযুক্ত ব্লুটুথ ডিভাইস এর দিকে রুট করা হয়েছে আপনি এক্ষেত্রে **AudioManager** টি অনুলসন্ধান করতে পারেন যেভাবে নীচের খন্ড চিত্রটিতে দেখানো হয়েছে:

```
if (isBluetoothA2dpOn()) {  
    // Adjust output for Bluetooth.  
} else if (isSpeakerphoneOn()) {  
    // Adjust output for Speakerphone.  
} else if (isWiredHeadsetOn()) {  
    // Adjust output for headsets  
} else {  
    // If audio plays and noone can hear it, is it still playing?  
}
```

অডিও আউটপুট হাডল্ড ওয়ার এর পরিবর্তন সামলানো

যখন একটি টেহডেসটেক আনপ্লগড করা হয়, অথবা একটি ব্লুটুথ ডিভাইস
বিচিচ্ছন্ন করা হয় অডিও সিস্টেম সব্ব্যংকিয়ভাবে বিবল্ট-ইন
স্পকারের দিকে রুটিঠক করে। আপনি যদি আপনার মিউজিক শুনেন থাকেন
আমি যত জোর শুনছি ঠিক তেতাজোর, এটা একটি শব্দময়িবস্ময়হতে
পারে।

সেীবাগযবশত সিস্টেমটি একটি ACTION_AUDIO_BECOMING_NOISY ইনটেন্ট
বর্ডকাস্ট করে যখন এটা য়েট। একটি BroadcastReceiver রেজিস্টার করা একটা
ভােলাচচর্াযা এই ইনটেন্টের জনয শুনেন থাকে যখন আপনি এটা পেল
করেন। মিউজিক পেলয়ারের কেষ্টের, ইউজার চিরাচরতভাবে আশাকরে
পেলবযাক স্টপ হবে-যখন গেমের জনয আপনি উল্লখে যাগযভাবে
ভলিউম নীচু করতে পছন্দ করতে পারেন

```
private class NoisyAudioStreamReceiver extends BroadcastReceiver {  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        if (AudioManager.ACTION_AUDIO_BECOMING_NOISY.equals(intent.getAction())) {  
            // Pause the playback  
        }  
    }  
}  
  
private IntentFilter intentFilter = new IntentFilter(AudioManager.ACTION_AUDIO_BECOMING_NOISY);  
  
private void startPlayback() {  
    registerReceiver(myNoisyAudioStreamReceiver(), intentFilter);  
}  
  
private void stopPlayback() {  
    unregisterReceiver(myNoisyAudioStreamReceiver);  
}
```

ফটো ক্যাপচার করা

(<http://developer.android.com/training/camera/index.html>)

সমৃদ্ধ মিডিয়া সহজলভ্য হওয়ার আগে পৃথিবী নিরস এবং বৈশিষ্টহীন ছিল। Gopher কে স্মরণ করতে পারেন? আমরা এখন আর পারি না। আপনার অ্যাপকে আপনার ইউজারের জীবনের একটি অংশ করতে, তাদেরকে একটি সুযোগ দিন যাতে তারা তাদের জীবন কিছ' অংশ এটার মধ্যে দিয়েতে পারে। অন-বোডর্ ক্যামেরা ব্যবহার করে, আপনার অ্যাপলিকেশন ইউজার তার চারপাশ যা দেখে তা ছিঁড়ে দিয়ে দিতে সামর্থ্য তৈরী করে, একটি সর্বতনতর অ্যাপাউটার তৈরী, চারপাশ জিম্ব (ভূত-পের্ত) খুঁজে বেরােনা, অথবা শুধুমাত্র অভিজ্ঞতা বিবিনময়।

এই ক্লাস আপনাকে বিদ্যমান ক্যামেরা অ্যাপলিকেশনের সর্বোচ্চ সুবিধােনয়ার অতি-সহজ উপায় সহদরুত ক্লক করা অবস্থায় পাবে। পরবর্তী অনুশীলনীতে, আপনি আরও গভীরে যাবেন এবং ক্যামেরা হাডর্ ওয়ার সরাসরি কীভাবে বিনয়নতরংগ করা তা শিখতে পাবেন।

অনুশীলনীসমূহ

ফটো নেয়া

অন্য অ্যাপলিকেশনকে পর্ভাষিত করা এবং শুধুমাত্র কিকছু ক্যাডেডর
লাইন সহ ফটো ক্যাপচার করা

ভিডিও রেকর্ড করা

অন্য অ্যাপলিকেশনকে পর্ভাষিত করা এবং শুধুমাত্র কিকছু ক্যাডেডর
লাইন সহ ভিডিও রেকর্ড করা

ক্যামেরা নিয়ন্ত্রণ

ক্যামেরা হার্ডওয়্যার সরাসরি নিয়ন্ত্রণ করা এবং আপনার নিজস্ব
ক্যামেরা বাস্‌তায়ন করা ক্যাপচার করা

ছবি নেয়া

(<http://developer.android.com/training/camera/photobasics.html>)

এই অনুশীলনটি ব্যাখ্যা করে বকীভাবে একটি বিদ্যমান ক্যামেরা ব্যবহার করে একটি ফটো ধারণ করা যায়।

ধরুন আপনি একটি কর্‌আউড-সোসর্ড (জন-সম্পৃক্ত) আবহাওয়া সেবা দিয়ে থাকেন যা আপনার কলোয়েন্ট অ্যাপ রান করা ডিভাইস কত রক্তে তাল আকাশের ছবি একটি তরত করার মাধ্যমে একটি টেবিশব্ক আবহাওয়া ম্যাপ তৈরী করে। ফটো একটি তরত করা আপনার অ্যাপ লিকেশনের খুব কৃষুদর একটি অংশ। আপনি অতি ব্যস্ততানাদে দিখিয়ে ফটো তুলতে চান, ক্যামেরা অতি পরিবর্তন না করে। খুশির খবর হচ্ছে অধিকাংশ অ্যান্ড্রয়েড ডিভাইস কমপক্ষে একটি করে ক্যামেরা অ্যাপ লিকেশন ইনস্টল করেছে। এই অনুশীলনটিতে, আপনি শিখবেন কীভাবে এটাকে আপনার ছবি তেলার জন্য ব্যবানবেন।

কয্‌া়ে মর া প ার ি ম শ ন ি র ়ে ক া়ে য স্ ট ক র া

যদি আপন ার অয্‌ াপে পর একটি ট গু রু ত ব্ প ন র্ কাজ হয় ছবি তে ালা, তাহলে
গু গ ল পে প্ ল তে তি ডি ভ াই সে স য ার কয্‌ া়ে মর া আে ছ ত াতে এর দৃ শ য্ মান ত া
স ি ম া ব দ্ ধ ক রে দি ন। পর্ চ া রিত ক র া়ে আপন ার অয্‌ াপ লি ক শ ন কয্‌ া়ে মর া
থ া ক ার উপ র নি ভ র্ র শ ি ল, আপন ার তে ম নি ন ফে স্ ট ফ াই লে একটি <uses-feature> ট য্ া প
র া থু ন:

```
<manifest ... >
  <uses-feature android:name="android.hardware.camera" />
  ...
</manifest ... >
```

যদি আপন ার অয্‌ াপ লি ক শ ন ব য্ ব হ ার ক রে, কি ন তু কাজ ক র ার জন য্ কে ান
কয্‌ া়ে মর া ন া চ া য়, ট য্ া়ে গ android:required="false" যু ক্ ত ক রু ন। এ ট া ক র লে, গু গ ল
পে প্ ল কয্‌ া়ে মর া ছ া ড া ডি ভ াই সে স আপন ার অয্‌ াপ লি ক শ ন ড াউ ন লে া ড ক রে ত
দে য়। তা র পর hasSystemFeature(PackageManager.FEATURE_CAMERA) ক ল ক রে র া ন ট াই মে
কয্‌ া়ে মর া আে ছ কি ন া তা চে চ ক ক র া আপন ার দ া য় ত ব্। যদি কয্‌ া়ে মর া ন া
থ া কে, আপন ার উ চিত ত খ ন কয্‌ া়ে মর া ি ব ি শ ষ্ ট স কি র্ য ক র া।

কয াে মর া অয াপ ি দে য ফে ট া তু লু ন

অয ান ডর ে ডে ডিলে গি টি া উপ ায় অনয া অয াপ ি লে কশনে কাজ করা হে চ্ চ একটি Intent কে আহবান করা য়ে য়ে ট া আপ িন য া করতে চ ান তা বনর ্ না করে। এই পর ি কর ্ য া তিন ট া অ াশ কে য়ু ক ্ ত করে: Intent িন জে, এক ্ স ট ান র ্ াল Activity শুরুর ে করতে একটি কল এব া যখন ফে াক াস আপ ার একটি টি ভি টি তে তি ফের আেস তখন ই়ে মজ ড া ট া চ ালিত করার জনয া ক়ি ছু কে াড।

এখ ানে একটি ফ া াশন আে ছ য়ে য়ে ট া একটি ছবি তুলে ত একটি ইন া টে ন ্ ট কে আহবান করে।

```
private void dispatchTakePictureIntent(int actionCode) {  
    Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);  
    startActivityForResult(takePictureIntent, actionCode);  
}
```

অ িভনন ্ দন: এই কে াড ি দে য় আপ ার অয াপ ি লে কশন অনয া কয াে মর া অয াপ ি লে কশন কে এ টি ব ড করাে ন ার মে ত া সামথর ্ অজর ্ ন করে ছে! অবশয া ই, যদি কে ান সবর ্ উে প াে য া গ াী অয াপ ি লে কশন ইন া টে ন ্ ট ধর ার জনয া পর ্ স ্ তু ত না থ াে ক, তখন আপ ার অয াপ বয ্ থর ্ ত ায় পযর ্ বসিত হে ব। এখ ানে একটি ফ া াশন দে য় া হে ল া য ার মাধয াে ম আপ িন চে ক করতে পারে বন আপ ার অয াপ ট া আপ ার ইন া টে ন ্ ট ট া সামল াে ত পারে ব ক়িন া:

```
public static boolean isIntentAvailable(Context context, String action) {  
    final PackageManager packageManager = context.getPackageManager();  
    final Intent intent = new Intent(action);  
    List<ResolveInfo> list =  
        packageManager.queryIntentActivities(intent, PackageManager.MATCH_DEFAULT_ONLY);  
    return list.size() > 0;  
}
```

ফেটো ভিউ করা

যদি দৃষ্টিতে তত্ত্বের সাথে ছবি তৈরী করাটাই আপনার অর্থে পর চুড়ান্ত আকাঙ্ক্ষা না হয়, তাহলে আপনি তখন সম্ভবত ক্যামেরা অর্থাৎ থেক্সট ফিউচার চাইবেন এবং এটা দিয়ে অন্য কিছু করবেন।

অর্থাৎ ডেড ক্যামেরা অর্থাৎ পলিকেশন ফিউচার Intent এ ফেটোটি এনকোড করে, কি "data"র অধীনে এক স্ট্রিং এর মধ্যে একটি Bitmap বিটম্যাপ হিসাবে onActivityResult() এ ডিলিভারি দেয়। নিম্নোক্ত কোড এই ইমেজ উদ্ধার করে এবং ImageView এ প্রদর্শন করে।

```
private void handleSmallCameraPhoto(Intent intent) {  
    Bundle extras = intent.getExtras();  
    mImageBitmap = (Bitmap) extras.get("data");  
    mImageView.setImageBitmap(mImageBitmap);  
}
```

নোট: "data" থেকে এই থামবোন ইমেজ একটি আইকনের জন্য ভাঙে লাগতে পারে, কিন্ত এতে চেয়ে বেশী কিছু না। একটি পূর্ণ সাইজের ইমেজ নিয়ে কাজ করতে আরেকটু বেশী কাজ করতে হয়।

ফটো সেভ করা

অ্যান্ডরয়েড ক্যামেরা অ্যাপলিকেশন একটি পূর্ণরসাইজের ফটো সেভ করে যদি আপনি এটা সেভ করতে দেন। আপনাকে অবশ্যই একটি পথ সরবরাহ করতে হবে যা সেটোর জভিলিউম, ফোল্ডার এবং ফাইল নেম অন্তর্ভুক্ত করে।

এখানে ফটোর জন্য পথ পাওয়ার একটি সহজ উপায় আছে, কিন্ত এটা শুধু অ্যান্ডরয়েড ২.২ (এপিআই লেভেল ৮) এবং এর পরবর্তী সংস্করণে কাজ করে:

```
storageDir = new File(
    Environment.getExternalStoragePublicDirectory(
        Environment.DIRECTORY_PICTURES
    ),
    getAlbumName()
);
```

পর্যায়ক্রমিকভাবে এপিআই লেভেলের জন্য, আপনার নিজেকে ফটো ডিরেক্টরীর নাম সরবরাহ করতে হবে।

```
storageDir = new File (
    Environment.getExternalStorageDirectory()
    + PICTURES_DIR
    + getAlbumName()
);
```

নোট: পথ কেম্পানেন্ট PICTURE_DIR হচ্ছে শুধু Pictures/, শয়ার ফটোর জন্য যথায় থেলোকেশন (স্থান) হচ্ছে এক্সটানরাল/শয়ারড সেটোর জ।

ফাইল নেম সেট করুন

পূর্ববর্তরী অধ্যায়ে যেভাবে দেখানো হয়েছে, একটি ইমেজের জন্য ফাইল লোকেশন অবশ্যই ডিভাইস এনভায়রনমেন্ট দ্বারা চা্লিত হওয়া উচিত। যার জন্য আপনার নিজে থেকে একটি সংঘর্ষ-রোধক ফাইল-নামকরণ স্কিম (a collision-resistant file-naming scheme) বেছে নিতে হবে। পরবর্তরী ব্যবহারের জন্য আপনি একটি মেমব্রার ভিত্তিতে বেলের মধ্যে পথ সেট করতে চাইতেও পারেন। এখানে একটি উদারণ সমাধান দেয়া হলো:

```
private File createImageFile() throws IOException {
    // Create an image file name
    String timeStamp =
        new SimpleDateFormat("yyyyMMdd_HHmmss").format(new Date());
    String imageFileName = JPEG_FILE_PREFIX + timeStamp + "_";
    File image = File.createTempFile(
        imageFileName,
        JPEG_FILE_SUFFIX,
        getAlbumDir()
    );
    mCurrentPhotoPath = image.getAbsolutePath();
    return image;
}
```

ইন্টেন্‌টর উপর ফাইল নেম যুক্ত করুন

যদি কখনও আপনার ইমেজ সেভ করার একটি জায়গা থাকে, Intent এর মাধ্যমে এ
লেবেল কখনও কখনও মেরা অর্থাৎ পালকে শেনে পাস করে দিন।

```
File f = createImageFile();  
takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT, Uri.fromFile(f));
```

একটি গ্যালারি রিটে ফটো সংযোজন করুন

যখন ইন্টেন্টিভ মাধ্যমে আপনি একটি ফটো তৈরি করে বন, আপনার জানা উচিত ইমেজ কীভাবে রাখা হয়েছে, কারণ আপনি বলেছেন পর্বে থাকা স্থানের মধ্যে এটা কে রাখা যেতে হবে। বাকীদেব জনস্ব, সম্ভবত সহজ উপায়ে আপনার ফটোকে পর্বে বসে যাগয করাহে চ্ছিসে স্টেমের মিডিয়া পর্ভাইডার থেকে এটাকে পর্বে বসে যাগয করা।

নিম্নে নাক্ত উদাহরণ মেথড দেখায় মিডিয়া পর্ভাইডার ডাটাবেজ আপনার ফটো যুক্ত করে কীভাবে সিসে স্টেমের মিডিয়া স্কয়ারানারকে আহবান করে, অর্ন্তদরয়েড গ্যালারী অর্াপিলি কশন এবং অনর্ অর্াপে এটাকে সহজ পর্পয করা যায়।

```
private void galleryAddPic() {  
    Intent mediaScanIntent = new Intent(Intent.ACTION_MEDIA_SCANNER_SCAN_FILE);  
    File f = new File(mCurrentPhotoPath);  
    Uri contentUri = Uri.fromFile(f);  
    mediaScanIntent.setData(contentUri);  
    this.sendBroadcast(mediaScanIntent);  
}
```

একটি স্কেলড ইমেজ ডেকেড করা

সীমিত আকারের মেমরীতে বহুমুখি ইমেজ ব্যবস্থাপনা জটিল হতে পারে। আপনি যদি দেখেন যে আপনার অ্যাপ্লিকেশন অল্প কিছু ছবি দেখিয়েই মেমরী শেষ করে ফেলে, আপনি নটকীয়ভাবে একটি ইমেমরী বিনয়্যাসের মধ্যে বর্ধিত করার দ্বারা ব্যবহৃত ডায়নামিক হিপ গুলে লাকমিয়ে ফেলতে পারেন যা ইতিমধ্যে গন্তব্য ভিত্তি এর সাইজের সাথে ম্যাচ করানোর জন্য মাপা হয়েছে। নিম্নোক্ত উদাহরণে দেখানো হয়েছে এই কৌশল দেখায়।

```
private void setPic() {  
    // Get the dimensions of the View  
    int targetW = mImageView.getWidth();  
    int targetH = mImageView.getHeight();  
  
    // Get the dimensions of the bitmap  
    BitmapFactory.Options bmOptions = new BitmapFactory.Options();  
    bmOptions.inJustDecodeBounds = true;  
    BitmapFactory.decodeFile(mCurrentPhotoPath, bmOptions);  
    int photoW = bmOptions.outWidth;  
    int photoH = bmOptions.outHeight;  
  
    // Determine how much to scale down the image  
    int scaleFactor = Math.min(photoW/targetW, photoH/targetH);  
  
    // Decode the image file into a Bitmap sized to fill the View  
    bmOptions.inJustDecodeBounds = false;  
    bmOptions.inSampleSize = scaleFactor;  
    bmOptions.inPurgeable = true;  
  
    Bitmap bitmap = BitmapFactory.decodeFile(mCurrentPhotoPath, bmOptions);  
    mImageView.setImageBitmap(bitmap);  
}
```


ভিডিও রেকর্ড করা

(<http://developer.android.com/training/camera/ videobasics.html>)

এই অনুশীলনীতে দেখাবে কীভাবে বিবদ্যমান কয়ামেরা অয়্যাপিলিকেশন দিয়ে ভিডিও কয়্যাপচার করতে হয়।

আপনার অয়্যাপিলিকেশনের করার জন্য কাজ আছে, এবং ভিডিও সমন্বয় করা তার একটি কক্ষমদর অংশ। আপনি অতিবয়স্ ততানাদেখিয়ে ভিডিও করতে চান, কয়্যামকডরার অতিপরিবর্তন নাকরে। খুশির খবর হচ্ছে অধিকাংশ অয়্যান্ডরওয়েড ডিভাইসে একটি কয়্যামেরা অয়্যাপিলিকেশন আছে যা ভিডিও রেকর্ড করে। এই অনুশীলনীতে, আপনি এটা আপনার জন্য তাকরার জন্য বানাবেন।

কয্‌া়ে মর া প ার ি ম শ ন ি র়ে ক া়ে য স্ ট

পর্চ ার ক়ে ত়ে য়, াপ ন ার অয্ াপ ি ল়ে ক শ ন াক ি ট কয্‌া়ে মর া থ াক ার ঁপ র ি ন ঙ র়় ক়ে, ি ম ি ন ি ফ়ে স্ ট ফ াই়ে ল াক ি ট <uses-feature> টয্ াগ র া থ় ন :

```
<manifest ... >
  <uses-feature android:name="android.hardware.camera" />
  ...
</manifest ... >
```

যদি াপ ন ার অয্ াপ ি ল়ে ক শ ন বয্‌ বহ ার ক়ে, ি ক ন় ত়ু ক াজ ক়ার জনয্ ক়ে ান কয্‌া়ে মর া ন া চ ায়, টয্ াগ android:required="false" য় ক় ত ক়় ন। ঁ ট া ক়ে ল, গ় গ ল় প় ল কয্‌া়ে মর া ছ াড় া ি ড ঙ াই়ে স াপ ন ার অয্ াপ ি ল়ে ক শ ন ড াউ ন়ে ল াড ক়ে ত়ে দ়। ত ার প় hasSystemFeature(PackageManager.FEATURE_CAMERA) ক় ক়ে র া ন ট াই়ে ম কয্‌া়ে মর া ঁ ছ ি ক ন া ত া চ় ক় ক় া াপ ন ার দ া ি য় ত ব়। যদি কয্‌া়ে মর া ন া থ াক়ে, াপ ন ার ঁ চ় ত ত খ ন কয্‌া়ে মর া ব় ি শ য়্ ট স ি ক় য় ক় া।

কয ামে র া অয াপ ি দে য ি ভি ড ও রে ক ড র া কর া

অয ান ড র া য়ে ড ডে ডি লগে গি ট া উপ ায় অনয া অয াপ ি লকে শনে ক াজ কর া হচে চ া
একি ট Intent কে আহব ান কর া য়ে ট া আপ ি ন য া করে ত চ ান ত া বন র া ন া করে। এই
পর ি কর া য় া তি ন ট া অ াশ কে য় ক ত করে: Intent ি ন জে, এক স ট া ন র া ল Activity
শুরুর করে ত একি ট কল এব া যখন ফে ক াস আপ ন ার একি টি ভি ডি টে ত ফিরে আসে তখন
ভি ডি ও চ ালি ত কর ার জনয া ি ক হু কে ক াড।

এখ ানে একি ট ফ া াশ ন আে ছে য়ে ট া একি টি ভি ডি ও করে ত একি ট ইনটে ট ন্ ট কে
আহব ান করে।

```
private void dispatchTakeVideoIntent() {  
    Intent takeVideoIntent = new Intent(MediaStore.ACTION_VIDEO_CAPTURE);  
    startActivityForResult(takeVideoIntent, ACTION_TAKE_VIDEO);  
}
```

এ ট া এক ট া ভ ালে া চি ন ত া, একি ট অয াপ একি ট ইনটে ট ন্ ট কে আহব ান কর ার
পূে ব র া স ামল ানে ার জনয া অি স ত ত ব া শীল থ াকে। এখ ানে একি ট ফ া াশ ন দে য় া
আে ছে য া অয াপে র জনয া চ ক করে য া আপ ন ার ইনটে ট ন্ ট স ামল াতে প ারবে:

```
public static boolean isIntentAvailable(Context context, String action) {  
    final PackageManager packageManager = context.getPackageManager();  
    final Intent intent = new Intent(action);  
    List<ResolveInfo> list =  
        packageManager.queryIntentActivities(intent,  
            PackageManager.MATCH_DEFAULT_ONLY);  
    return list.size() > 0;  
}
```

ভিডিও ভিউ

অ্যান্ড্রয়েড ক্যামেরা অ্যাপ্লিকেশন Intent এর মাধ্যমে ভিডিও ফরম্যাট দিয়ে Uri হিসাবে onActivityResult() এর পরে ডিলিভারি করে সেট করে ভিডিও লোকেশন নির্দেশ করে। নিম্নে কোড এই ভিডিও উদাহরণের এবং VideoView এ এটা প্রদর্শন করে।

```
private void handleCameraVideo(Intent intent) {  
    mVideoUri = intent.getData();  
    mVideoView.setVideoURI(mVideoUri);  
}
```

কয ামে মর া নিয়ন তর্ ণ কর া

(<http://developer.android.com/training/camera/cameradirect.html>)

এই অনুষ্টীলনীতে আমরা আলোচনা করব ফর্ মওয়াকর্ এপিআই বয বহার করে সরাসরি কীভাবে কয ামে মর া হাডর্ ওয়ার নিয়ন তর্ ণ করা যায়।

সরাসরি একটি ডিভাইস কয ামে মর া নিয়ন তর্ ণ বিদয মান কয ামে মর া অয াপিলেকশন থেকে ছবি বা ভিডিও রেকর্ডে স্ট করার চেয়ে বেশী ক্যাড চায়। কিন্ত যদি আপনি বিশেষ কয ামে মর া অয াপিলেকশন বা আপনার অয াপে সমিনব ত অবস্থায় থাকান য় িকছু তৈরী করতে চান, এই অনুষ্টীলনী আপনাকে এটা করতে শেখাবে।

কয্‌া়ে মর া অব়ে জক্‌ ট ও়ে পন কর া

Camera অব়ে জক্‌ ট এর একটি ইনস্টেন্স পাওয়া হ়ে চ়্‌ছ সরাসরি কয্‌া়ে মর া নিয়ন্‌তরন্‌ন পর়্‌কির য়়ার পর়্‌থম ধাপ। য়ে হ়ে তু অয্‌ান্‌ডর়্‌য়ে ড়ে র নিজসব্‌ কয্‌া়ে মর া অয্‌াপিল়ে কশন করে থ়়ে ক, কয্‌া়ে মর া তে পর়্‌বে শ করার পর়্‌স্‌ত া বিত উপায় হ়ে চ়্‌ছ একটি প্‌থক থ়ে র্‌ড য়়া onCreate() থ়ে কেশুরু হ়ে ত ার উপর Camera ও়ে পন কর া। এই অয্‌াপে র়্‌াচ একটি ভ়়ে ল া ধ ারনা য়ে হ়ে তু এট া িক হ়ু সময় বয্‌য় করতে প়়ে এর ং ইউজ়ার ইন্‌টারফেস থ়ে র্‌ড়ে ক অগর়্‌সর হ়ে ত ব াধ া দ়ে ত প়়ে র। একটি আরও ম়ে ািলক ব াস্‌ত া য়ে নের ম়ে ধয্‌, ক়ে াড প্‌হনর়্‌বয্‌় বহার করতে সঞ্‌চ ালন করতে ং ই নিয়ন্‌তরন্‌গ পর়্‌ব াহ ধ়ে র রাথ়ে ত কয্‌া়ে মর া ও়ে পন কর া onResume() মথ়ে ড়ি বলমব্‌ করতে প়়ে র।

Camera.open() কল কর া একটি বয্‌়িতকর়্‌ম করতে প়়ে র যদি কয্‌া়ে মর া ইতিম়ে ধয্‌ অনয্‌ অয্‌াপ কত্‌র়্‌ক একটি বয্‌়বহার়ে র ম়ে ধয্‌ থ়়ে ক, তাই আম ার া এট া ক ং হ়ু ব়্‌ল়ে ক য়ু কত করে রে থ়ে দ়ে ত প়়ে র।

```
private boolean safeCameraOpen(int id) {
    boolean qOpened = false;

    try {
        releaseCameraAndPreview();
        mCamera = Camera.open(id);
        qOpened = (mCamera != null);
    } catch (Exception e) {
        Log.e(getString(R.string.app_name), "failed to open Camera");
        e.printStackTrace();
    }

    return qOpened;
}

private void releaseCameraAndPreview() {
    mPreview.setCamera(null);
    if (mCamera != null) {
        mCamera.release();
        mCamera = null;
    }
}
```

যদি এপিআই লেভল ৯ হয়, কয্‌া়ে মর া ফ়ে র়্‌মওয়াকর়্‌ বহু মুখি কয্‌া়ে মর া স়ে প াটর়্‌ করে। যদি আপনি িলগয্‌ ািস এপিআই বয্‌়বহার করে ন ং ই একটি আরগু়ে মন্‌ট হ়া ড় াই open() কল করে ন, আপনি পর়্‌থম িরয় ার ফ়ে িস ং কয্‌া়ে মর া প়়ে বন।

কয ় ামে মর া ি প র ় ি ভ উ তৈ র ি কর া

একটি ট ছবি বেনেয়ার জনয ় সাধ ারনত প র ় য়ে াজন যে আপন ার ইউজ ার শ াট ার
টেপ ার আে গ ত ার সাব জেক্ট র পি প র ় ি ভ উ তৈ র ি কর া। এট া কর তে, আপনি য া
কয ় ামে মর া সেন্সর তুলে বন ত ার পি প র ় ি ভ উ তৈ র ি কর তে SurfaceView বয ় বহ ার
কর তে প ারেন।

পির্ভিউ ক্লাস

পির্ভিউ পর্দাশর্ন করে (ডিসপ্লে) শুরু করতে, আপনার পির্ভিউ ক্লাস দরকার। পির্ভিউ `android.view.SurfaceHolder.Callback` ইন্টারফেসের বাস্তবায়ন চায়, যা ক্যামেরা হাউন্ডার থেকে অ্যাপিলিকেশনে ইমেজ ডাটাপাস করার কাজে ব্যবহার করা হয়।

```
class Preview extends ViewGroup implements SurfaceHolder.Callback {  
  
    SurfaceView mSurfaceView;  
    SurfaceHolder mHolder;  
  
    Preview(Context context) {  
        super(context);  
  
        mSurfaceView = new SurfaceView(context);  
        addView(mSurfaceView);  
  
        // Install a SurfaceHolder.Callback so we get notified when the  
        // underlying surface is created and destroyed.  
        mHolder = mSurfaceView.getHolder();  
        mHolder.addCallback(this);  
        mHolder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);  
    }  
    ...  
}
```

লাইভ ইমেজ পির্ভিউ শুরু হওয়ার পূর্বে বর্ধিত পির্ভিউ ক্লাস অবশ্যই `Camera` তপাস করতে হবে, যেভাবে পরবর্তী অধ্যায়ে দেখানো হয়েছে।

পরিমিত উইন্ডো সেট করা এবং শুরু করা

একটি ক্যামেরা ইনস্ট্যান্স এবং এর সাথে সম্পর্কিত পরিমিত উইন্ডো অবশ্যই একটি নির্দিষ্ট ক্রম অনুসারে তৈরি হতে হবে, ক্যামেরা অবজেক্ট দিয়ে পরে থাকা হবে। নীচে চরিত্র উল্লিখিত ক্রমে ক্যামেরা আরম্ভের পরিকল্পনা সংযুক্ত করা হয়েছে যাতে `setCamera()` মেথড দ্বারা `Camera.startPreview()` কল করা হয়, যখনই ইউজার ক্যামেরার পরিবর্তনের জন্য ক্লিক করে। পরিমিত উইন্ডো অবশ্যই পরিমিত উইন্ডো ক্লাস `surfaceChanged()` কলব্যাক মেথডে ডিরেক্টরি পরিচালনা করতে হবে।

```
public void setCamera(Camera camera) {
    if (mCamera == camera) { return; }

    stopPreviewAndFreeCamera();

    mCamera = camera;

    if (mCamera != null) {
        List<Size> localSizes = mCamera.getParameters().getSupportedPreviewSizes();
        mSupportedPreviewSizes = localSizes;
        requestLayout();

        try {
            mCamera.setPreviewDisplay(mHolder);
        } catch (IOException e) {
            e.printStackTrace();
        }

        /*
        Important: Call startPreview() to start updating the preview surface. Preview must
        be started before you can take a picture.
        */
        mCamera.startPreview();
    }
}
```

কয্‌া়ে মর া়ে সি ট ং প ি র ব ত র ং ন ক র া

কয্‌া়ে মর া়ে সি ট ং য়ে উপ া়ে কয্‌া়ে মর া ছি ব়ে ন়ে ত া প ি র ব ত র ং ন ক র়ে, জু ম
লে ভ ল থে ক়ে শুরুর ক়ে এক স়ে প া জ া র ক়ে ম়ে প ন়ে স শ ন প য় র ং ন়ে ত। এই উ দ া হ র ণ
শু ধ়ে ম া ত র ং ি প র ং ি ভ উ স া ই জ প ি র ব ত র ং ন ক়ে র; আর ক়ে ক া ন ি ক ছ়ে র জ ন য়
কয্‌া়ে মর া অ য় া প ি ল়ে ক শ়ে ন়ে র়ে স া স র ং ক়ে ক া ড়ে দ খ়ে ন।

```
public void surfaceChanged(SurfaceHolder holder, int format, int w, int h) {  
    // Now that the size is known, set up the camera parameters and begin  
    // the preview.  
    Camera.Parameters parameters = mCamera.getParameters();  
    parameters.setPreviewSize(mPreviewSize.width, mPreviewSize.height);  
    requestLayout();  
    mCamera.setParameters(parameters);  
  
    /*  
     Important: Call startPreview() to start updating the preview surface. Preview must be  
     started before you can take a picture.  
    */  
    mCamera.startPreview();  
}
```

পিপার্ডিভিউওরিয়েন্টেশনসেটকরুন

অধিকাংশ ক্যামেরা অ্যাপলিকেশন ডিউসেপ্লকেলয়ান্ডেস্কপ
পদ্ধতিতে আটকে ফেলার কারণে এটুক্যামেরাসেসনসেরসব্ভাবিক
ওরিয়েন্টেশন। এইসিটিংআপনাকেপোর্টেটর্ইট-মোডফেটানিতে
বাধাদিবেনাকারনডিভাইসেরওরিয়েন্টেশন EXIF হেডারেরকডর্করা
আছে। কীভাবেইমেজেরকডর্করাহেচ্ছতাকেপর্ভাবানিবর্তনাকরেই
কীভাবেপিপার্ডিভিউপর্দাশর্ত(ডিউসেপ্ল)হবেতা setCameraDisplayOrientation() মেথড
আপনাকেপরিবর্তনকরতেদিবে। এপিআইলেভল ১৪ এরআগর
অ্যান্ডর্য়েডআপনকেঅবশ্যইওরিয়েন্টেশনপরিবর্তনকরার
পূর্বেরআপনারপিপার্ডিভিউবন্ধকরতেহবেএবংতারপরএটারিস্টাটর্
করতেহবে।

একটি ছবি নেয়া

যখনই পির্ভিউ শুরু হবে একটি ছবি তুলতে `Camera.takePicture()` ব্যবহার করুন। আপনি `Camera.PictureCallback` এবং `Camera.ShutterCallback` অবজেক্ট তৈরি করতে পারেন এবং তাদের `Camera.takePicture()` এ পাস করে দিতে পারেন।

যদি আপনি কর্মগতভাবে ইমেজ নিতে চান, আপনি একটি `Camera.PreviewCallback` তৈরি করতে পারেন যা `onPreviewFrame()` বাস্তুবায়ন করে। এর মধ্যে যেকিছু করতে, আপনি শুধু বাছাই করা পির্ভিউ ফ্রেমস ক্যাপচার করতে পারেন, অথবা `takePicture()` কল করে একটি ডেরি করা একশন সেটআপ করতে পারেন।

রিভিউ পূরণায় শুরুর করিন (রিস্টার্ট)

একটি ছবি নেয়া শেষ হলে, ইউজার আরেকটি ছবি নেয়ার পূর্বেই আপনাকে অবশ্যই রিভিউ রিস্টার্ট করতে হবে। এই উদাহরণে, শাটার বাটন ওভারলোড করার মাধ্যমে রিস্টার্ট করা হয়েছে।

```
@Override
public void onClick(View v) {
    switch(mPreviewState) {
        case K_STATE_FROZEN:
            mCamera.startPreview();
            mPreviewState = K_STATE_PREVIEW;
            break;

        default:
            mCamera.takePicture( null, rawCallback, null);
            mPreviewState = K_STATE_BUSY;
    } // switch
    shutterBtnConfig();
}
```

পির্ভিভিউ বন্ধ করুন এবং ক্যামেরা রিলিজ (হেঁড়ে দেয়া) করুন

একসময় আপনার অ্যাপ্লিকেশন ক্যামেরা ব্যবহার করতে শেষ করবে, এটা এখন পির্ভিউ করার করার সময়। বিবেচনা করে, আপনাকে অবশ্যই Camera অবজেক্ট রিলিজ করতে হবে, অথবা আপনি অন্য অ্যাপ্লিকেশন কর্তৃক শাসন করার ঝুঁকি নিবেন, যার মধ্যে আপনার নিজস্ব অ্যাপ্লিকেশনের নতুন ইনস্ট্যান্স রয়েছে।

আপনাকে কখন পির্ভিভিউ বন্ধ এবং ক্যামেরা রিলিজ করতে হবে? আপনার পির্ভিভিউ সারফেস ধ্বংস হওয়া একটি ভালো নির্দেশন যে পির্ভিভিউ বন্ধ এবং ক্যামেরা রিলিজ করতে এটাই উপযুক্ত সময়, Preview ক্লাস থেকে এই মেথডের মধ্যে যেভাবে দেখানো হয়েছে।

```
public void surfaceDestroyed(SurfaceHolder holder) {
    // Surface will be destroyed when we return, so stop the preview.
    if (mCamera != null) {
        /*
         * Call stopPreview() to stop updating the preview surface.
         */
        mCamera.stopPreview();
    }

    /**
     * When this function returns, mCamera will be null.
     */
    private void stopPreviewAndFreeCamera() {

        if (mCamera != null) {
            /*
             * Call stopPreview() to stop updating the preview surface.
             */
            mCamera.stopPreview();

            /*
             * Important: Call release() to release the camera for use by other applications.
             * Applications should release the camera immediately in onPause() (and re-open() it in
             * onResume()).
             */
            mCamera.release();

            mCamera = null;
        }
    }
}
```

এই অনুশীলনের পরে থমকিয়ে, এই পর্কিত যাও setCamera() এর একটি অংশ ছিল, তাই একটি ক্যামেরা শুরু করাটা সবসময় পির্ভিভিউ বন্ধ করার মাধ্যমে শুরু হয়।

কনটেটন টি পি র ন ট ক র া

(<http://developer.android.com/training/printing/index.html>)

অয ান ডর েড ইউজ ার কর ্মাগতভ াবে ত াে দর ডিভ াইসে শু ধ ুম াতর ্ কনটেটন টি ভিউ করে (েদেখ)। িকন ্তু একট সময় আসে ত প াের কে ান তথয ্ শেয ার কর ার জনয ্ কাউকে শু ধ ুম াতর ্ িস ্কর ্ নে দেখ ানে াট া পয াপ ্ত উপ ায় ন াও হে ত প াের। আপন ার অয ্ াপ িলেকশন থে কে তথয ্ িপ র ্ ন টি দে ত প ার াট া একজন ইউজ ারকে আপন ার অয ্ াপ থে কে একট বড় আক ারে র কনটেটন টি দে খ ার বা অনয ্ বয ্ াক িক ্ত িয িন আপন ার অয ্ াপ বয ্ বহ ার করে ন ন া ত ার স াথে এই তথয ্ শেয ার কর ার একট উপ ায় প র ্ দ ান করে। িপ র ্ িন ট ্ ত াে দর তথয ্ র (ইনফ ে মর ্ শন) একট ি-œ ্ য াপ শ টি তৈরী করে ত দে য য া ডিভ াইস থ াকা বা ন া থ াকা, পয র ্ াপ ্ত বয ্ াট ারী প াওয়ার, বা ওয়ারলেস নেটওয়ার্ক ক ানে কশন এর উপ র িনভর ্ র করে ন া।

অয ান ডর েড ৪.৪ (এপিআই লেভেল ১৯) এ ব ্ এর চেয়ে উপের স ্ স ্কর ্ নে র মধে য ্, ফে র ্ মওয়ার্ক স র াস ির অয ্ ান ডর েড অয ্ াপ িলেকশন থে কে িপ র ্ ন টি ক র া ইমেজে এ ব ্ ডকু মেন টস এর জনয ্ স াভি র ্ স প র ্ দ ান করে। এই প র ্ িশ ক ্ষন আলে াচনা করবে আপন ার অয ্ াপ িলেকশনে ক াভ াবে িপ র ্ িন ট ্ কে স িক র ্ য ক র া য ায য ার মধে য ্ রে য়ে ছ িপ র ্ িন ট ্ ইমেজ, HTML পেজে স এ ব ্ িপ র ্ ন টি ক র ার জনয ্ ক াস ্ টম ডকু মেন টস তৈরী ক র া।

অনুশীলনীসমূহ

ফটোটিপর্নট

এই অনুশীলনীতে দেখাবে কীভাবে ইমেজটিপর্নট করতে হয়

HTML ডকুমেন্টটিপর্নট

এই অনুশীলনীতে দেখাবে কীভাবে HTML ডকুমেন্টটিপর্নট করতে হয়

কাস্টম ডকুমেন্টসটিপর্নট

এই অনুশীলনীতে দেখাবে কীভাবে আপনি অ্যান্ডারুয়েডটিপর্নট
ম্যানুয়ালভাবে রসথে সংযোগ করে বন, একটিটিপর্নট এন্ডাপটর তৈরী
করবেন এবং পর্নটের জন্য কন্টেন্ট তৈরী করবেন

ফটো প্রিন্ট

(<http://developer.android.com/training/printing/photos.html>)

মোবাইল ডিভাইসের জন্য একটি ফটো তৈরি করা এবং শেয়ার করা সবচেয়ে জনপ্রিয় ব্যবহার। আপনার অ্যাপ্লিকেশন যদি ফটো নেয়, তাহলে এর প্রদর্শন করে এবং ইউজারকে তৈরি করার অনুরোধ করে, সে ক্ষেত্রে আপনার অ্যাপ্লিকেশন থেকে এই ইমেজ প্রিন্টকে সিকর্য করার বিষয়টি কয়েকটি বিবেচনা করা উচিত। Android Support Library (<http://developer.android.com/tools/support-library/index.html>) একটি ন্যূনতম পরিমাণে কোড এবং প্রিন্ট লেআউট অপশনের একটি সহজ সেট ব্যবহার করে ইমেজ প্রিন্টিংকে সিকর্য করার জন্য একটি যথাযথ ফাংশন সরবরাহ করে।

এই অনুশীলনীতে দেখাবে v4 সাপোর্টার লাইব্রেরী PrintHelper ক্লাস ব্যবহার করে কীভাবে ইমেজ প্রিন্ট করতে হয়।

ইমেজ পি প র ্ ন ্ ট

অয ্ ান ্ ডর ্ য়েড সাপেপাটর ্ লাইবের ্ রী PrintHelper ক ্ লাস ইমেজের পি প র ্ ন ্ ট করতে একটি সহজ উপায় প র ্ দান করে। ক ্ লাসিটির একটি একক লেআউট অপশন আছে, setScaleMode(), যা দুইয়ের মধ্যে একটি অপশন দিয়ে আপনাকে পি প র ্ ন ্ ট করতে দেয়:

- SCALE_MODE_FIT – এই অপশন ইমেজকে আকার প র ্ দান করে যাতে সম্পূর্ণ ইমেজ পেজটির পি প র ্ ন ্ টে যোগ্য এলাকার মধ্যে দেখা যায়।
- SCALE_MODE_FIT - এই অপশন ইমেজকে পরিমাপ করে যাতে এটা পেজটির সম্পূর্ণ পি প র ্ ন ্ টে যোগ্য এলাকা পূর্ণ করে। এই সেটিং পছন্দ করলে বে া ঝ া য়ে উপরের বানী চরিত্র অংশ বা ডান অথবা বাম পাশের চিত্রের অংশ পি প র ্ ন ্ ট হবে না। এই অপশন হচ্ছে ছি ডি ফল ্ ট ভয় ্ াল ্ য় যদি না আপনি সে সকল মোড পরিবর্তন করে দেন।

setScaleMode() এর জন্য উভয় স্কেলিং অপশনই ইমেজ ইনটয় ্ াক ্ ট এর বিদ্যমান অনুপাত ধরে রাখে। নীচের কোড উদাহরণে দেখায় কীভাবে PrintHelper ক ্ লাসের একটি ইনস্ট্যান্স তৈরি করতে হয়, কীভাবে সে স্কেলিং অপশন সেট করতে হয় এবং পি প র ্ িন ্ ট ্ প র ্ িক র ্ য া শ ্ র ্ করতে হয়:

```
private void doPhotoPrint() {
    PrintHelper photoPrinter = new PrintHelper(getActivity());
    photoPrinter.setScaleMode(PrintHelper.SCALE_MODE_FIT);
    Bitmap bitmap = BitmapFactory.decodeResource(getResources(),
        R.drawable.droids);
    photoPrinter.printBitmap("droids.jpg - test print", bitmap);
}
```

এই মেথডটি একটি মেনু আইটেমের জন্য একশন হিসাবে বলা যেতে পারে। উল্লেখ্য, একশনের জন্য মেনু আইটেম যা সবসময় সাপেপাটর ্ লাত্ত করে না (যেমন পি প র ্ িন ্ ট ্) ওভারফ্লো মেনুতে পেলস করা উচিত। আরও জানতে Action Bar (<http://developer.android.com/design/patterns/actionbar.html>) ভিজাইন গাইডে দেখুন।

printBitmap() কল করার পর, আপনার অ্যাপ থেকে আরেকান একশনের পরে যোজন হয় না। অয ্ ান ্ ডর ্ য়েড পি প র ্ ন ্ ট ইউজার ইন্টারফেস আবিভূর্ত হয়, ইউজারকে একটি পি প র ্ ন ্ টার এবং পি প র ্ িন ্ ট ্ অপশন বেছে নিতে দেয়। ইউজার এরপর ইমেজটি পি প র ্ ন ্ ট করতে পারে বা একশনটি ক্যান্সেল করতে পারে। যদি ইউজার ইমেজ পি প র ্ ন ্ ট করাট বেছে নেয়, একটি পি প র ্ ন ্ ট জব তৈরি হবে এবং সিসে ্ টম বারে একটি পি প র ্ িন ্ ট ্ নোটিফিকেশন দৃশ্যমান হবে।

আপনি যদি শুধু ইমেজ ছাড়াও অতিরিক্ত কোন কন্টেন্ট আপনার পি প র ্ ন ্ ট আউটে অন্তর্ভুক্ত করতে চান, আপনাকে অবশ্যই একটি পি প র ্ ন ্ ট ডকুমেন্ট তৈরি করতে হবে। পি প র ্ িন ্ ট ্ এর জন্য ডকুমেন্ট তৈরীর বিষয়ে তথ্য পেতে Printing an HTML Document বা Printing a Custom Document অনুশীলনী দেখুন

HTML ডকুমেন্ট প্রিন্ট

(<http://developer.android.com/training/printing/html-docs.html>)

অ্যান্ড্রয়েড শুধু একটি ফ্রেমওয়ার্ক কন্টেন্ট প্রিন্ট আউটের জন্য প্রয়োজন একটি প্রিন্ট ডকুমেন্টের ভিতরে টেক্সট এবং গ্রেফিক্সকে প্যাক করা। একটি ডকুমেন্টকে প্যাক করতে এবং একটি নুয়নতম ক্যাডি দিয়ে এটি প্রিন্ট করতে অ্যান্ড্রয়েড ফর্মওয়ার্ক HTML ব্যবহার করার একটি উপায় সরবরাহ করে।

অ্যান্ড্রয়েড ৪.৪ (এপিআই লেভেল ১৯) এর মধ্যে, `WebView` ক্লাস HTML কন্টেন্ট প্রিন্টিং সিকর্য করতে আপডেট হচ্ছে। ক্লাস একটি স্থানীয় HTML রিসোর্স লোড করতে বা ওয়েব থেকে একটি পেজ ডাউনলোড করতে, একটি প্রিন্ট জব তৈরি করতে এবং অ্যান্ড্রয়েড প্রিন্ট সার্ভিসের পরে এটি হস্তান্তর করতে আপনাকে অনুমোদন করে।

এই অনুশীলনী আপনাকে দেখাবে কীভাবে টেক্সট এবং গ্রেফিক্স ধারণ করা HTML ডকুমেন্ট অতি দ্রুত তৈরি করা যায় এবং এটি প্রিন্ট করতে `WebView` ব্যবহার করা যায়।

HTML ডকুমেন্ট লোড করুন

WebView দিয়ে একটি HTML ডকুমেন্ট প্রদর্শন করে HTML রিসোর্স লোড করা বা একটি সিস্টেমের ইন্টারফেসে একটি HTML ডকুমেন্ট তৈরী করার বিষয়গুলোকে অন্তর্ভুক্ত করে। এই অধ্যায়ে আলোচনা করে কীভাবে একটি HTML সিস্টেম তৈরী করতে হয় এবং প্রদর্শন করে এর জন্য WebView এর মধ্যে এটাকে লোড করে।

এই ভিউ অবজেক্ট একটি টিভি টেলিউইটের অংশ হিসাবে সাধারণভাবে ব্যবহার হয়ে থাকে। যদি আপনার অ্যাপলিকেশন WebView ব্যবহার করেনা থাকে, আপনি ক্লাসের একটি ইনস্ট্যান্স তৈরী করতে পারেন স্ট্রিনিং রিস্ট্রাক্টভাবে প্রদর্শন করে এর উদ্দেশ্য। এই কাস্টম প্রদর্শন টিভি তৈরী করার পদ্ধতি ধাপগুলো হচ্ছে:

1. একটি WebViewClient তৈরী করুন যা HTML রিসোর্স লোড হওয়ার পর একটি প্রদর্শন জব শুরু করে।
2. WebView অবজেক্টের মধ্যে HTML রিসোর্স লোড করা

নিম্নোক্ত কোড নমুনা দেখায় কীভাবে একটি সহজ WebViewClient তৈরী করা যায় এবং অতিদ্রুত চলমান অবস্থায় তৈরী HTML ডকুমেন্ট লোড করা হয়:

```
private WebView mWebView;

private void doWebViewPrint() {
    // Create a WebView object specifically for printing
    WebView webView = new WebView(getActivity());
    webView.setWebViewClient(new WebViewClient() {

        public boolean shouldOverrideUrlLoading(WebView view, String url) {
            return false;
        }

        @Override
        public void onPageFinished(WebView view, String url) {
            Log.i(TAG, "page finished loading " + url);
            createWebPrintJob(view);
            mWebView = null;
        }
    });

    // Generate an HTML document on the fly:
    String htmlDocument = "<html><body><h1>Test Content</h1><p>Testing, " +
        "testing, testing...</p></body></html>";
    webView.loadDataWithBaseURL(null, htmlDocument, "text/HTML", "UTF-8", null);

    // Keep a reference to WebView object until you pass the PrintDocumentAdapter
    // to the PrintManager
    mWebView = webView;
}
```

নোট: পূর্ববর্তী অধ্যায়ে আপনি তৈরী করেছিলেন, সেই WebViewClient এর onPageFinished() মেথডের মধ্যে হওয়া একটি প্রদর্শন জব তৈরী করার জন্য আপনার কল মিনিশ্চত করুন। যতদূর সম্ভব পজেলোডিং শেষ না হতে ততদূর যদি আপনি অপেক্ষা করতে না পারেন, প্রদর্শন এর আউটপুট অসম্পূর্ণ রাখা হতে বা সম্ভব হতে পারে।

নোট: উপরের উদাহরণ কোড WebView একটি ইনস্ট্যান্স ধরে রাখা যেতে প্রদর্শন জব তৈরী হওয়ার পূর্বে এটা যাতে গারবেজ কালেকশন না হয়। এটা মিনিশ্চত করুন যে আপনি আপনার নিজস্ব বাস্তুচালনা একইভাবে করেছেন অন্যথায় প্রদর্শন টি প্রক্রিয়াকরণে ব্যর্থ হতে পারে।

আপনি যদি পেজ রিফ্রেশ অন্তর্ভুক্ত করতে চান, আপনার প্রজেক্টের assets/ ডিরেক্টরীর মধ্যে গرافিক ফাইল প্লস করুন এবং loadDataWithBaseURL() মেথডের প্রথম প্যারামিটারের মধ্যে একটি বেজ URL

সুনির্দিষ্ট করতে, যেভাবে নীচের কোড উদাহরণে দেখানো হয়েছে:

```
webView.loadDataWithBaseURL("file:///android_asset/images/"  
htmlBody  
"text/HTML"  
"UTF-8"  
null);
```

আপনি `loadDataWithBaseURL()` এর যেখানে ডর সাথে `loadUrl()` বদল করার মাধ্যমে
পিপার্নিট এর জন্য একটি ওয়েব পেজ লোড করতে পারেন, যেভাবে নীচে
দেখানো হলো।

```
// Print an existing web page (remember to request INTERNET permission!)  
webView.loadUrl("http://developer.android.com/about/index.html");
```

যখন আপনি পিপার্নিট ডকুমেন্ট তৈরী করার জন্য `WebView` ব্যবহার করে
থাকেন, আপনাকে নীচের সীমাবদ্ধতা বিষয়ে সতর্ক থাকতে হবে:

1. আপনি পেজে হেডার, ফুটার যুক্ত করা বা পেজ নামব্রার যুক্ত করতে
পারবেন না
2. HTML ডকুমেন্টের জন্য পিপার্নিট অপশনগুলো পিপার্নিটে পজ
েরঞ্জকে অন্তর্ভুক্ত করেনা, যেমন: ১০ পেজের HTML ডকুমেন্টের ২
থেকে ৪ পর্যন্ত পিপার্নিট করা হবে, এই বিষয়গুলো সাপোর্ট করা হয়
না।
3. `WebView` একটি ইনস্ট্যান্স একসাথে শুধুমাত্র একটি পিপার্নিট জব
পর্য্যাক্ষর করতে পারে
4. HTML ডকুমেন্ট CSS পিপার্নিট এটির বিউটি ধারণ করে, যেমন
ল্যান্ডস্কেপ পারিটি সাপোর্ট করা হয় না
5. পিপার্নিট শুরুর করতে একটি HTML ডকুমেন্ট এর মধ্যে `JavaScript`
ব্যবহার করতে পারবেন না

নোট: একটি ডবনঠরিব এর কন্টেন্ট যা একটি লাইউটে অন্তর্ভুক্ত, এটা
পিপার্নিট ডও হতে পারে যখনই এটা একটি ডকুমেন্ট লোড করে।

আপনি যদি আরও পরিমিত পিপার্নিট আউটপুট তৈরী করতে চান এবং পিপার্নিট
হওয়া পেজের উপরে ডর করা কন্টেন্টের উপর সম্পূর্ণ নিয়ন্ত্রণ
রাখতে চান: পরবর্তী অনুশীলনে (Printing a Custom Document) চলে যান।

পিরন্ট জব তৈরী করুন

একটি WebView তৈরী এবং আপনার HTML কন্টেন্ট লোড করার পর, আপনার অ্যাপ্লিকেশন এর পিরন্টিং পর্কিত করার অংশটুকু পরায় সমাপ্ত করে ফেলেছেন. পরবর্তী ধাপ হচ্ছে PrintManager পর্বেশ, একটি পিরন্ট অ্যাডাপটার তৈরী, এবং সবরশেষ একটি পিরন্ট জব তৈরী করা। নীচের উদাহরণ ব্যাখ্যাকরে কীভাবে এই ধাপগুলো সম্পাদন করতে হয়:

```
private void createWebPrintJob(WebView webView) {  
  
    // Get a PrintManager instance  
    PrintManager printManager = (PrintManager) getActivity()  
        .getSystemService(Context.PRINT_SERVICE);  
  
    // Get a print adapter instance  
    PrintDocumentAdapter printAdapter = webView.createPrintDocumentAdapter();  
  
    // Create a print job with name and adapter instance  
    String jobName = getString(R.string.app_name) + " Document";  
    PrintJob printJob = printManager.print(jobName, printAdapter,  
        new PrintAttributes.Builder().build());  
  
    // Save the job object for later status checking  
    mPrintJobs.add(printJob);  
}
```

এই উদাহরণ অ্যাপ্লিকেশন দ্বারা ব্যবহার করার জন্য PrintJob অবজেক্টের একটি ইনস্ট্যান্সকে সত্বে, যার কোন পর্যেয়াজন নেই। আপনার অ্যাপ্লিকেশন পিরন্ট জবের অগ্রগতি অনুসরণ করতে এই অবজেক্ট ব্যবহার করতে পারেন যে এটা পর্কিত করা জাত হচ্ছে। এই অ্যাডাপ্টার উপকারী তখনই যখন আপনি আপনার অ্যাপ্লিকেশনের পিরন্ট জবের অবস্থা মনিটরিং করতে চান যেমন, কাজটা শেষ হয়েছে কিনা, অকার্যকর হয়েছে কিনা বা ইউজার ক্যানসেল করেছে কিনা। ইন-অ্যাপ্লিকেশনটিফিকেশন তৈরী করার কোন পর্যেয়াজন নেই, কারন পিরন্ট ফর্মওয়ার্ক সবর্যাক্তি করে যভাবে পিরন্ট জবের জন্য একটি সিস্টেম নোটিফিকেশন তৈরী করে দেয়।

কাস্ টম ডকুমেণ্টস প্ৰিন্ট

(<http://developer.android.com/training/printing/custom-docs.html>)

কিছু অ্যাপ্লিকেশনের জন্য, যেমন ডকুমেন্ট অ্যাপস, পেজ লেআউট অ্যাপস এবং অন্যান্য অ্যাপস যা গ্ৰাফিক আউটপুটকে ফালাস করে, সুন্দর প্ৰিন্ট করা পেজ তৈরী করা হচ্ছে পৰ্ধান বৈশিষ্ট। এই কেসে তর একটি ইমেজ বা একটি HTML ডকুমেণ্ট প্ৰিন্ট করাটাই যথেষ্ট নয়। এই ধরনের অ্যাপ্লিকেশনের প্ৰিন্ট আউটপুটে তর জন্য একটি পেজের মধ্যে যাকিছু যায় তার সব কিছুর উপর নিদিষ্ট স্টাইল নিয়ন্ত্রণ পৰ্যেয়োজন, যার মধ্যে রয়েছে ফন্ট, টেক্সট ফ্লো, পেজ বর্ক, হেডার, ফুটার এবং গ্ৰাফিক এলিমেন্ট।

একটি প্ৰিন্ট আউটপুট তৈরী করা যা সম্ভব পৰ্ভাবে আপনার অ্যাপ্লিকেশনের জন্য পৰ্যেয়োজ্য করার জন্য পৰ্যেয়োজন পূর্ববর্তী আলোচনার অ্যাপে পৰ্চে চলে যে অনেক বৈশিষ্ট পৰ্গামি বিবিন্যোগ করে। আপনাকে অবশ্যই কমে পোনে টৈ তরী করতে হবে যা প্ৰিন্ট ফর্ম ওয়াকে র সাথে যোগাযোগ করতে পারে, প্ৰিন্টার সেটিং এর সাথে থাপ থায়, পেজ এলিমেন্ট ডর্করে, বহুমুখি পেজ প্ৰিন্ট পিচালনা করা।

এই অনুশীলনী আপনাকে দেখায় কীভাবে আপনি প্ৰিন্ট ম্যােনেজারের সাথে যুক্ত হতে পারবেন, একটি প্ৰিন্ট এডাপটর তৈরী করতে পারবেন এবং প্ৰিন্টারের জন্য কন্টেন্ট বানতে পারবেন।

পরিবর্তন টিম যোগাযোগের সাথে যুক্ত করুন

যখন আপনার অ্যাপলিকেশন সরাসরি পরিবর্তন টিম পরিচালনা করা চালাতে চলে, আপনার ইউজারের কাছ থেকে একটি পরিবর্তন টিমের কয়েকটি রিসিস্ত করার পর প্রথম ধাপ হচ্ছে অ্যান্ডারয়েড পরিবর্তন টিমের ফর্মওয়ার্ড করা এবং PrintManager ক্লাসের একটি ইনস্ট্যান্স লাভ করা। এই ক্লাস আপনাকে একটি পরিবর্তন টিম জব আরম্ভ করতে এবং পরিবর্তন টিম লাইফসাইকেল শুরু করতে অনুমোদন দেয়। নিম্নলিখিত কোড উদাহরণ আপনাকে দেখায় কীভাবে পরিবর্তন টিম যোগাযোগের পেতে হয় এবং পরিবর্তন টিম পরিচালনা শুরু করতে হয়।

```
private void doPrint() {  
    // Get a PrintManager instance  
    PrintManager printManager = (PrintManager) getActivity()  
        .getSystemService(Context.PRINT_SERVICE);  
  
    // Set job name, which will be displayed in the print queue  
    String jobName = getActivity().getString(R.string.app_name) + " Document";  
  
    // Start a print job, passing in a PrintDocumentAdapter implementation  
    // to handle the generation of a print document  
    printManager.print(jobName, new MyPrintDocumentAdapter(getActivity()),  
        null);  
}
```

উপরের কোড উদাহরণ দেখায় কীভাবে একটি পরিবর্তন টিম জবের নাম নির্দিষ্ট হয় এবং PrintDocumentAdapter ক্লাসের একটি ইনস্ট্যান্স সেট করতে হয় যা পরিবর্তন টিম লাইফসাইকেলের ধাপগুলো পরিচালনা করে। পরিবর্তন টিম এডাপ্টার ক্লাসের বাস্তবায়ন পরবর্তী সেকশনে আলোচনা করা হয়েছে।

নোট: print() মেথডের শেষ প্যারামিটার একটি PrintAttributes অবজেক্ট নয়। পূর্ববর্তী পরিবর্তন টিম লাইফসাইকেল এর ভিত্তিতে পরিবর্তন টিম ফর্মওয়ার্ড করা এবং পরিবর্তন টিম সেট অপশনে হিন্টস প্রদান করতে আপনি এই প্যারামিটার ব্যবহার করতে পারেন, এই উপায়ে ইউজার এক্সপেরিয়েন্স উন্নত করে। আপনি অপশন সেট করতেও প্যারামিটার ব্যবহার করতে পারেন যেটা যে কন্টেন্ট পরিবর্তন টিম হবে তার জন্য বৈশিষ্ট্য উপযুক্ত, যেমন ল্যান্ডস্কেপ ওরিয়েন্টেড শন সেট করুন যখন একটি ফটো পরিবর্তন টিম করবেন যা ওরিয়েন্টেড শনে আছে।

একটি টি প র ্ ন ্ ট অ য ্ া ড া প ট া র তৈ ত র ী ক র ু ন

একটি টি প র ্ ন ্ ট অ য ্ া ড া প ট া র অ য ্ া ন ্ ড র ্ য়ে ড টি প র ্ ন ্ ট ফে র ্ ম ও য় া কের ্ র সা থে য়ে া গ া য়ে া গ কেরে এ ব ্ া টি প র ্ ি ন ্ ট ্ প র ্ ি ক র ্ য় া র ধ া প গ ু লে া প রি চ া ল ন া ক রে। এই প র ্ ি ক র ্ য় া র জন য় প র ্ য়ে া জ ন টি প র ্ ন ্ ট ক র া র জন য় একটি ড ক ু মে ন ্ ট তৈ ত র ী ক র া র প ্ হে ব র ্ ই উ জ া র ক ত ্ র ্ ক টি প র ্ ন ্ ট া র এ ব ্ া টি প র ্ ন ্ ট অ প শ ন নি ব র ্ া চ ন ক র া। এই নি ব র ্ া চ ন চ ু ড া ন ্ ত আ উ ট প ু ট কে প র ্ ভ া বি ত ক র তে প া রে য়ে হ ত ু ই উ জ া র ভি ন ্ ন আ উ ট প ু ট সা ম থ র ্ , ভি ন ্ ন পে জ সা ই জ, ব া ভি ন ্ ন পে জ ও রি য়ে নে ন ্ ট শ ন দি য়ে টি প র ্ ন ্ ট া র প হ ন ্ দ ক রে। য়ে হ ত ু এই নি ব র ্ া চ ন তৈ ত র ী ক র া হ য়ে ছে, টি প র ্ ন ্ ট ফে র ্ ম ও য় া কের ্ আপ ন া র টি প র ্ ন ্ ট অ য ্ া ড া প ট া র কে একটি টি প র ্ ন ্ ট ড ক ু মে ন ্ ট প রি ক ল ্ প ন া ক র তে ব লে এ ব ্ া এ ট া তৈ ত র ী ক র তে ব লে। য খ ন ই এক জন ই উ জ া র টি প র ্ ন ্ ট ব া ট ন চ া পে, ফে র ্ ম ও য় া কের ্ চ ু ড া ন ্ ত টি প র ্ ন ্ ট ড ক ু মে ন ্ ট নে য় এ ব ্ া এ ট া কে টি প র ্ ন ্ ট প র ্ ভ া ই ড া রে আ উ ট প ু টে র জন য় প া ঠ া য়। টি প র ্ ি ন ্ ট ্ প র ্ ি ক র ্ য় া সা ম য় ক া লে ই উ জ া র টি প র ্ ন ্ ট এক শ ন ক য় া ন সে ল ক র তে প া রে, সু ত রা ্ আপ ন া র টি প র ্ ন ্ ট অ য ্ া ড া প ট া র কে অব শ য় ই একটি ক য় ন সে ল ক র া রি কে া য়ে স ্ ট শ ু ন তে হ বে এ ব ্ া এর প র ্ ি ত প র ্ ি তি ক র ্ য় া ক র তে হ বে।

PrintDocumentAdapter অ য ্ া ব স ্ ট র ্ া ক ্ ট ক ্ ল া স টি প র ্ ন ্ ট া র ল া ই ফ সা ই কে ল প রি চ া ল ন া ক র া র জন য় ডি জ া ই ন ক র া হ য়ে ছে, য া র চ া রি টি প র ্ ধ া ন ক ল ব য় া ক মে থ ড আ ছে। আপ ন া কে অব শ য় ই টি প র ্ ন ্ ট ফে র ্ ম ও য় া কের ্ র সা থে যথ া য় থ ভ া বে য়ে া গ া য়ে া গ ক র া র জন য় আপ ন া র টি প র ্ ন ্ ট অ য ্ া ড া প ট া রে র সা থে এই মে থ ড গ ু লে া ব া স ্ ত ব া য় ন ক র তে হ বে:

- **onStart()** – টি প র ্ ন ্ ট প র ্ ি ক র ্ য় া র শ ুর ু তে শ ু ধ ু এক ব া র ক ল ক র া হ য়। য দি আপ ন া র অ য ্ া প লি ক শ নে র সম ্ প া দ ন ক র া র জন য় য়ে কে া ন ট া ই ম-টি প র ্ প া রে শ ন ক া জ থ া কে, য়ে ম ন য়ে ড া ট া টি প র ্ ন ্ ট ক র তে হ বে ত া র একটি “œ” য া প শ টি প া ও য় া, সে গ ু লে া কে এ থ া নে সম ্ প া দ ন ক র ু ন। আপ ন া র অ য ্ া ড া প ট া রে এই মে থ ড ব া স ্ ত ব া য় ন ক র া র কে া ন প র ্ য়ে া জ ন নে ই।
- **onLayout()** – এক জন ই উ জ া র য ত ব া র একটি টি প র ্ ন ্ ট সে টি ্ প রি ব ত র ্ ন ক রে, প র ্ ি ত ব া র ই এ ট া ক ল ক র া হ য়, য়ে ম ন একটি টি ভি ন ্ ন পে জ সা ই জ, ব া পে জ ও রি য়ে নে ন ্ ট শ ন, আপ ন া র অ য ্ া প লি ক শ ন কে য়ে পে জ টি প র ্ ন ্ ট হ বে ত া র একটি টে ল আ উ ট গ ণ ন া ক র া র একটি স ু য়ে া গ দে য়। ন ু য় ন ত ম ভ া বে, এই মে থ ড অব শ য় ই টি প র ্ ন ্ ট ক র া ড ক ু মে ন ্ ট এর ম ধ য় ক ত গ ু লে া পে জ আ শ া ক রে ত া রি ট া ন র ্ ক রে।
- **onWrite()** – টি প র ্ ন ্ ট হ বে এ ম ন ফ া ই লের ম ধ য় রে ন ্ ড া র টি প র ্ ন ্ ট পে জে ক ল ক র া হ য়। এই মে থ ড কে প র ্ ি ত ট া onLayout() ক লের প র এক ব া এক া ধি ক ব া র ক ল হ তে প া রে।
- **onFinish()** – টি প র ্ ন ্ ট প র ্ ি ক র ্ য় া র শে ষে এক ব া র ক র া হ য়। য দি আপ ন া র অ য ্ া প লি ক শ নে র কে া ন ও য় া ন ট া ই মি টি য় া র-ড া উ ন ক া জ থ া কে, এ থ া নে সম ্ প া দ ন ক র ু ন। আপ ন া র অ য ্ া ড া প ট া রে এই মে থ ড ব া স ্ ত ব া য় ন ক র া র কে া ন প র ্ য়ে া জ ন নে ই।

নিচে র অ ধ য় া য় আ লে া চ ন া ক রে ক া ভ া বে লে আ উ ট এ ব ্ া রা ই ট মে থ ড ব া স ্ ত ব া য় ন ক র তে হ য়, য া একটি টি প র ্ ন ্ ট অ য ্ া ড া প ট া রে ক া জ ক র া থ ু ব জি ট ল।

নে া ট: এই অ য ্ া ড া প ট া রে মে থ ড কে আপ ন া র অ য ্ া প লি ক শ নে র মে ই ন থে র ্ ডে ক ল ক র া হ য়। য দি আপ নি আপ ন া র অ য ্ া প লি ক শ নে এই মে থ ড গ ু লে া ক া জ ক র া তে উ লে ল থ য়ে া গ য় প রি ম া ন সা ম য় নি তে আ শ া ক রে ন, ত া দে র ক া জ ক র া তে একটি প ্ থ ক থে র ্ ডে ব া স ্ ত ব া য় ন ক র ু ন। উ দ া হ র ন স ব ্ র ু প, আপ নি প ্ থ ক AsyncTask শ অব জে ক ্ টে ল আ উ ট ব া টি প র ্ ন ্ ট ড ক ু মে ন ্ টে লে খ া র ক া জ স ্ য ু ক ত ক র তে প া রে ন।

প্ৰিন্ট ডকুমেণ্ট ইন্‌ফো কম্পাউণ্ডিং

`PrintDocumentAdapter` ক্লাসে সৰু একাধিক বাস্তুৰ বাবে মধ্যস্থতা, আপোনাৰ অ্যাপলিকেশ্যনৰ বাবে ই ডকুমেণ্টৰ ধৰণ নিৰ্দিষ্ট কৰিবলৈ সমৰ্থ হ'ব যাঁহে এটা তৈৰী কৰিছে। এই প্ৰিন্ট জেবৰ জনসংখ্যা সৰ্বমুঠ পৃষ্ঠাৰ সংখ্যা গণনা কৰে, প্ৰিন্ট কৰাৰ পূৰ্বে পৃষ্ঠাৰ সংখ্যা পৰিৱৰ্তন কৰিব পাৰিব। অ্যাপটৰ `onLayout()` মেথডে ডকুমেণ্টৰ বাস্তুৰ বাবে এক `PrintDocumentInfo` ক্লাসৰ মধ্যস্থতা প্ৰিন্ট জেবৰ কাৰ্য্যকৰী আউটপুট সম্বন্ধিত এই গণনাৰ প্ৰক্ৰিয়া তৈৰী কৰিবলৈ পৰিৱৰ্তন কৰে, পৃষ্ঠাৰ সংখ্যাৰ এক কন্টেন্ট টাইপ সহ। নিম্নলিখিত কোড উদাহৰণ এক `PrintDocumentAdapter`ৰ জনসংখ্যা `onLayout()` মেথডৰ মেথ'ডিক বাস্তুৰ বাবে দেখায়:

```
@Override
public void onLayout(PrintAttributes oldAttributes,
    PrintAttributes newAttributes,
    CancellationSignal cancellationSignal,
    LayoutResultCallback callback,
    Bundle metadata) {
    // Create a new PdfDocument with the requested page attributes
    mPdfDocument = new PrintedPdfDocument(getActivity(), newAttributes);

    // Respond to cancellation request
    if (cancellationSignal.isCancelled()) {
        callback.onLayoutCancelled();
        return;
    }

    // Compute the expected number of printed pages
    int pages = computePageCount(newAttributes);

    if (pages > 0) {
        // Return print information to print framework
        PrintDocumentInfo info = new PrintDocumentInfo
            .Builder("print_output.pdf")
            .setContentType(PrintDocumentInfo.CONTENT_TYPE_DOCUMENT)
            .setPageCount(pages);
        info.build();
        // Content layout reflow is complete
        callback.onLayoutFinished(info, true);
    } else {
        // Otherwise report an error to the print framework
        callback.onLayoutFailed("Page count calculation failed.");
    }
}
```

`onLayout()` মেথড সম্বন্ধিত প্ৰিন্টৰ তথ্য ফলাফল থাকিব: কম্পিলেশ্যন (শেষ কৰা), কম্পাইলেশ্যন (বাতিল কৰা) বা ফাইলিং (অসফল)। এয়া ঘটনাৰ যথেষ্ট লেআউটৰ কম্পাইলেশ্যন (গণনা) সমাপ্ত হয় না।

`PrintDocumentAdapter.LayoutResultCallback` অবজেক্টৰ যথেষ্ট মেথড টকল কৰাৰ মাধ্যমে আপোনাক অবশ্যই এই ফলাফল প্ৰক্ৰিয়াৰ কোন একাধিক নিৰ্দেশ কৰিবলৈ হ'ব।

নোট: `onLayoutFinished()` মেথডে ডকুমেণ্টৰ বুলিয়ান পৰামিতিৰ লেআউট কন্টেন্ট আসলেই পৰিৱৰ্তন হ'ব নাই। ইয়াত নিৰ্দেশ কৰিবলৈ শেষৰিকোয়েষ্ট থকাৰ বাবে এই পৰামিতিৰ যথেষ্ট ভাবে সেৱা কৰিবলৈ প্ৰিন্ট ফৰ্মাটত `onWrite()` মেথডে অপৰ্য্যেয় জনসংখ্যাৰ কল কৰা, পৃষ্ঠাৰ লেআউট প্ৰিন্ট ডকুমেণ্ট অপৰিহাৰ্য্য ভাবে জৰুৰীয়াৰ বাবে পৰিৱৰ্তন কৰিবলৈ উন্নয়ন কৰাৰ বাবে কৰিবলৈ তথ্য দিব।

`onLayout()`ৰ পৰিৱৰ্তন কৰা হ'ব পৃষ্ঠাৰ সংখ্যাৰ কম্পাইলেশ্যন কৰাৰ বাবে আউটপুট হিচাপে আকাঙ্ক্ষা কৰা হয়। প্ৰিন্টৰ এটাৰ বিৱৰ্তন দিয়া কৰিবলৈ আপোনাৰ এই নামৰ কম্পাইলেশ্যন কৰিবলৈ আপোনাৰ অ্যাপলিকেশ্যন প্ৰিন্টিংৰ এক জনসংখ্যাৰ জৰুৰীয়া কৰাৰ বাবে উপৰ সম্বন্ধিত কৰিব।

নিম্নের শীল। নীচে চরকে ডাউনহরন একটি টবাস্ তবায়ন দেখায় যেখানে
পেজের সংখ্যা পির্ন ট ওরিফেনে টশন কতরু কনিধর ারিত হয়ে থাকে:

```
private int computePageCount(PrintAttributes printAttributes) {  
    int itemsPerPage = 4; // default item count for portrait mode  
  
    MediaSize pageSize = printAttributes.getMediaSize();  
    if (!pageSize.isPortrait()) {  
        // Six items per page in landscape orientation  
        itemsPerPage = 6;  
    }  
  
    // Determine number of print items  
    int printItemCount = getPrintItemCount();  
  
    return (int) Math.ceil(printItemCount / itemsPerPage);  
}
```

একটি টিপি র্ণ ন্ট ডকুমে ন্ট ফাইল লেখা (রাইট)

একটি টিফাইলে টিপি র্ণ ন্ট আউটপুট লেখার সময়, অ্যান্ডারৱেয়েড টিপি র্ণ ন্ট ফর্ম ওয়াকর আপনার অ্যান্ডাপিলিকেশনের `PrintDocumentAdapter` ক্লাসের `onWrite()` মেথড কল করে। মেথডের প্যারামিটার ক্যান পেজ লেখা উচিত এবং আউটপুট ফাইল ব্যবহার হবে তা নির্ধারণ করে। আপনার এই মেথডের বাস্তবায়ন অবশ্যই তারপর কন্টেন্টের রিকোয়েস্ট করা পরিত্যাগ পেজ মালি টি-পেজ **PDF** ডকুমে ন্ট ফাইলে পেশ করতে হবে। যখন এই পরিকর্য সম্পন্ন হবে আপনি কলব্যাক অবজেক্টের `onWriteFinished()` মেথড কল করেন।

\$েনাট: অ্যান্ডারৱেয়েড টিপি র্ণ ন্ট ফর্ম ওয়াকর `onLayout()` এ পরিত্যাগ কলের জন্য এক বা একাধিকবার `onWrite()` মেথড কল করতে পারে। এই কারণে, `false` এ `onLayoutFinished()` মেথডের বুলিয়ান (boolean) প্যারামিটার সেট করাটা গুরুত্বপূর্ণ যখন টিপি র্ণ ন্ট কন্টেন্ট লেআউট পরিবর্তন হয় না, টিপি র্ণ ন্ট কন্টেন্টের অপরেয়োজনীয় রি-রাইট পরিহার করার জন্য।

েনাট: `onLayoutFinished()` মেথডের বুলিয়ান প্যারামিটার লেআউট কন্টেন্ট আসলেই পরিবর্তিত হয়েছে কিনা তা নির্ধারণ করে শেষ রিকোয়েস্ট থেকে অদয় বোধ। এই প্যারামিটার যথাযথভাবে সেট করতে টিপি র্ণ ন্ট ফর্ম ওয়াকরকে `onLayout()` মেথড অপরেয়োজনীয়ভাবে কল করা, পূর্বের লেখা টিপি র্ণ ন্ট ডকুমে ন্ট অপরিহার্যভাবে জিমিয়ে রাখা এবং পারফরমেন্স উন্নয়ন করা থেকে বিরত রাখতে দিন।

নিম্নোক্ত নমুনা একটি **PDF** ফাইল তৈরী করতে `PrintedPdfDocument` ক্লাস ব্যবহার করে এই পরিকর্যার মৌলিক মেকানিকস দেখায়:

```

@Override
public void onWrite(final PageRange[] pageRanges,
    final ParcelFileDescriptor destination,
    final CancellationSignal cancellationSignal,
    final WriteResultCallback callback) {
    // Iterate over each page of the document,
    // check if it's in the output range.
    for (int i = 0; i < totalPages; i++) {
        // Check to see if this page is in the output range.
        if (containsPage(pageRanges, i)) {
            // If so, add it to writtenPagesArray. writtenPagesArray.size()
            // is used to compute the next output page index.
            writtenPagesArray.append(writtenPagesArray.size(), i);
            PdfDocument.Page page = mPdfDocument.startPage(i);

            // check for cancellation
            if (cancellationSignal.isCancelled()) {
                callback.onWriteCancelled();
                mPdfDocument.close();
                mPdfDocument = null;
                return;
            }

            // Draw page content for printing
            drawPage(page);

            // Rendering is complete, so page can be finalized.
            mPdfDocument.finishPage(page);
        }
    }

    // Write PDF document to file
    try {
        mPdfDocument.writeTo(new FileOutputStream(
            destination.getFileDescriptor()));
    } catch (IOException e) {
        callback.onWriteFailed(e.toString());
        return;
    } finally {
        mPdfDocument.close();
        mPdfDocument = null;
    }
    PageRange[] writtenPages = computeWrittenPages();
    // Signal the print framework the document is complete
    callback.onWriteFinished(writtenPages);

    ...
}

```

এই নমুনা `drawPage()` মেথডে PDF পেজ কন্টেইনারের রেন্ডারিং হস্তান্তর করে, যা পরবর্তী অধ্যায়ে আলোচনা করা হবে।

যেহেতু লেআউটের সাথে, `onWrite()` মেথড কায়রু সম্পাদনের তিনটা ফলাফল থাকতে পারে: কমপ্লিশন (শেষ করা), ক্যান্সেলেশন (বাতিতল করা) বা ফেইলিওর (অসফল) এর ঘটনা যেখানে কন্টেইনার টেলে থাওয়া হয় না।

`PrintDocumentAdapter.WriteResultCallback` অবজেক্টের যথাযথ মেথড টিকল করার মাধ্যমে আপনাকে অবশ্যই এই ফলাফলগুলো লারেকান একটি টিনেদর্শ করতে হবে।

নোট: পিরিন্টিং এর জন্য একটি ডকুমেন্টের রেন্ডারিং একটি রিসোর্স-ইনটেইনিসভ অপারেশন হতে পারে। আপনার অ্যাপ্লিকেশনের পর্দা ইউজার ইন্টারফেসে থের্ড ব্লক করাকে পরিহার করার জন্য, আপনার উচিত একটি পৃথক থের্ডেপেজের রেন্ডারিং বারাইটিটি অপারেশন সম্পাদন করার বিষয়টাকে বিবেচনা করা উচিত, উদাহরণস্বরূপ, একটি `AsyncTask` এর মধ্যে। অসিঙ্ক্রোনাস (asynchronous) কাজের মতো এটিকে সিকিউশন থের্ডের সাথে কাজ করা সম্পর্কিত আরও তথ্য জানতে, `Processes and Threads` (<http://developer.android.com/guide/components/processes-and-threads.html>) দেখুন।

PDF পেজ কন্টেন্ট ড্র করা

যখন আপনার অ্যাপ্লিকেশনটি পরিদর্শন করে, আপনার অ্যাপ্লিকেশন অবশ্যই একটি PDF ডকুমেন্ট তৈরী করে এবং পরিদর্শন করার জন্য এটা অ্যান্ডারুয়েডে পরিদর্শন ফর্মওয়াটেকর্পাঠিয়ে দেয়। এই উদ্দেশ্যে আপনি যে কোন PDF ডকুমেন্টের রাইবেরী ব্যবহার করতে পারেন। এই অনুশীলনীতে দেখায় আপনার কন্টেন্ট থেকে PDF পেজ তৈরী করতে কীভাবে PrintedPdfDocument ক্লাস ব্যবহার করতে হয়।

PrintedPdfDocument ক্লাস একটি PDF পেজ এলিমেন্ট ড্র করতে একটি Canvas অবজেক্ট ব্যবহার করে, একটা একটি টিভি টেলিআউটে ড্র করার মতো করে। ড্র Canvas মেথড ব্যবহার করে পরিদর্শন করার পেজ আপনি এলিমেন্ট ড্র করতে পারেন। নিম্নোক্ত উদাহরণে কোডে দেখায় এই মেথডগুলো ব্যবহার করে একটি PDF ডকুমেন্ট কীভাবে ব্রিকহু সহজ এলিমেন্ট ড্র করতে হয়:

```
private void drawPage(PdfDocument.Page page) {
    Canvas canvas = page.getCanvas();

    // units are in points (1/72 of an inch)
    int titleBaseLine = 72;
    int leftMargin = 54;

    Paint paint = new Paint();
    paint.setColor(Color.BLACK);
    paint.setTextSize(36);
    canvas.drawText("Test Title", leftMargin, titleBaseLine, paint);

    paint.setTextSize(11);
    canvas.drawText("Test paragraph", leftMargin, titleBaseLine + 25, paint);

    paint.setColor(Color.BLUE);
    canvas.drawRect(100, 100, 172, 172, paint);
}
```

যখন PDF পেজ ড্র করতে Canvas ব্যবহার করা হয়, এলিমেন্ট পেয়েন্ট নির্দিষ্ট হয়, যা এক ইঞ্চি ১/৭২। নির্দিষ্ট করুন পেজ এলিমেন্টের সাইজ নির্দিষ্ট করার জন্য আপনি এই পরিমাপের একক ব্যবহার করেছেন। ড্র করা এলিমেন্টের অবস্থানের ঠিক করার জন্য, সমন্বয়কারী সিস্টেম থেকে শুরু করে, পেজের উপরের বাম কোণের জন্য ০।

টিপ: যখন ঈধহাধ অবজেক্ট একটি PDF ডকুমেন্টের সীমানায় আপনাকে পরিদর্শন এলিমেন্ট রাখতে দেয়, অনেক পরিদর্শনার পেপারের কোণের অংশটি পরিদর্শন দিতে সমর্থ হয় না। নির্দিষ্ট করুন যে আপনি পেজের আনুপাতিক টবলকোনা (এজ) বন্ধ করেছেন, যখন আপনি এই ক্লাস দিয়ে একটি পরিদর্শন ডকুমেন্ট তৈরী করেন।

গরুাফিক্‌স এবং অযুাণিসেশন িদেয় অযুাপস তৈরী

(<http://developer.android.com/training/building-graphics.html>)

এই ক্লাস আপনাকে শেখাবে কীভাবে গরুাফিক্‌স িদেয় কাযরু সম্পাদন করতে হয় যা আপনার অযুাপকে পরুতিযোগীতায় িকছু সুবিধা পরুদান করতে পারে। আপনি যদি মেমোরীলক ইউজার ইন্টারফেসের সীমা ছাড়িয়ে একটি সুন্দর িভজুয়্যাল এক্সপেইয়ন্স তৈরী করতে চান, এই ক্লাসগুলো আপনাকে সেখানে পেঁাছে িদেতে সহায়তা করবে।

১. দক্‌ষতার সাথে িবটমযুাপ পরুদশরু

কীভাবে িবটমযুাপ লোড এবং পরুিকরুয়াজাত করতে হয় যখন আপনার ইন্টারফেসকে সবেদনশীল (রেসপন্সিভ) রাখা হয় এবং মেমরীর মাতরুাকে ছাড়িয়ে যাওয়াকে পরিহার করে

১. বড় আকারের িবটমযুাপ লোড করা
২. ইউআই থেরুড অফ করতে িবটমযুাপ পরুেসস করা
৩. িবটমযুাপ জমিয়ে (Caching) রাখা
৪. িবটমযুাপ মেমরী পরিচালনা করা
৫. আপনার ইউজার ইন্টারফেসে িবটমযুাপ পরুদশরু

২. OpenGL ES সহকারে গরুাফিক্‌স পরুদশরু

অযুানুডরুয়েড অযুাপ ফেরুয়ওয়াকের মধ্যে কীভাবে উচুহাএখ গরুাফিক্‌স তৈরী করতে হয় এবং টাচ ইনপুটে রেসপন্স করতে হয়।

১. OpenGL ES পরিবেশ তৈরী করা
২. আকার (েশপ) িননরুয়
৩. আকার (েশপ) ডরু করা
৪. পরুজেক্ট এবং কযুােমরা িভুউ আবেদন
৫. মেশন যুুক্ত করা
৬. টাচ ইভেন্টে রেসপন্স করা

৩. অযুাণিসেশন যুুক্ত করা

কীভাবে আপনার ইউজার ইন্টারফেসে পরিবর্তন সূচক অযুাণিসেশন যুুক্ত করতে হয়।

১. দুইটা িভুউ করু সফড করুন (একটার পর আরেকটি আসা)
২. িস্করুন সল্লাইডের জন্য িভুউপেজার ব্যুহহার করা
৩. কাডরু িফ্লপ অযুাণিসেশন পরুদশরু
৪. িভুউজুয়ম করা
৫. লআউট পরিবর্তন অযুাণিসেশন করা

দক্ষতার সাথে বিটম্যাপ পরিদর্শন

(<http://developer.android.com/training/displaying-bitmaps/index.html>)

শিখুন কীভাবে সাধারণত কীভাবে বিটম্যাপ ব্যবহার করে Bitmap অবজেক্ট লোড এবং
পরিবর্তিত করা যায় এবং আপনি আপনার ইন্টারফেসকে স্কেলেবল (স্কেলেবল
(রেসপনসিভ)) রাখা হয় এবং মেমরীর মাত্রার ক্ষেত্রে যাওয়া থেকে পরিহার
করে। আপনি যদি সতর্ক না হোন, বিটম্যাপ আপনার বিটম্যাপ মেমরী
ব্যয় থেকে দ্রুত খরচ করে ফলে আপনার ডেভাইস এক্সপেন্সিভ কারণে
একটি অ্যাপলিকেশনকে ধবংস হওয়ার পথে নিয়ে যায়:

java.lang.OutOfMemoryError: bitmap size exceeds VM budget.

এখানে কিছু কারণ আছে যেখানে দেখায় আপনার অ্যান্ড্রয়েড
অ্যাপলিকেশন বিটম্যাপ লোড করছে কতগুলি:

- মোবাইল ডিভাইসের সাধারণত ব্যবহৃত সিস্টেম রিসোর্স আছে।
অ্যান্ড্রয়েড ডিভাইস একটি একক অ্যাপলিকেশন ১৬ গিগাবাইটের
মোট মেমরী থাকতে পারে। Android Compatibility Definition Document (CDD),
সেকশন ৩.৭. Virtual Machine Compatibility বিবর্তন সিস্টেম সাইজ এবং যন্ত্রের
পরিমাপনীয় ন্যূনতম অ্যাপলিকেশন মেমরী দেয়। অ্যাপলিকেশন এই
ন্যূনতম মেমরী সীমার মধ্যে থেকে সম্পাদন করার মতো করে
অপটিমাইজ করা উচিত। কিন্ত মেনে রাখতে বাকি ডিভাইস উচ্চ
মাত্রার কনফিগার করা হয়েছে।
- বিটম্যাপ অনেক মেমরী নিয়ে নেয়, বিশেষ করে ফটোগ্রাফের মতো উচ্চ
মাত্রার ইমেজের জন্য। উদাহরণস্বরূপ, Galaxy Nexus একবারে 2592x1936
pixels (5 megapixels) পর্যন্ত ফটো নিয়ে থাকে। যদি বিটম্যাপ কনফিগারেশন
ARGB 8888 ব্যবহার করে থাকে (অ্যান্ড্রয়েড ২.৩.৩ আরও অগ্রেসিভ
ডিফল্ট) তখন মেমরীতে এই ইমেজ লোড করতে মেমরীর প্রায় ১৯ MB নেয়
(25921936 bytes), কিছু ডিভাইসে পিআর-অ্যাপলিকেশন কন্ট্রোল
করে দেয়।
- অ্যান্ড্রয়েড অ্যাপ ইউজার ইন্টারফেস নিয়ন্ত্রিতভাবে একই সাথে
কতগুলি বিটম্যাপ লোড হওয়ার আকাঙ্ক্ষা করে। কন্ট্রোল
ListView, GridView এবং ViewPager সাধারণত একই সাথে বহুবিধ বিটম্যাপ
অন-সিস্টেম অন্তর্ভুক্ত করে, শুধু একটি আর্গুমেন্টের ট্যাকায়
সম্ভাবনাময় অনেক অফ-সিস্টেম পরিদর্শন হওয়ার জন্য প্রস্তুত
আছে।

অনুশীলনীসমূহ

বড়আকারেরবিটম্যাপলেডকরা

এইঅনুশীলনীপর্ষিতঅ্যাপিলেকশনেরমেমরীলিমিটঅতিক্রমনাকরেই
বড়আকারেরবিটম্যাপিডেকাডকরতেহয়কীভাবেতাপনাকেদেখাবে

বিটম্যাপপর্ষিকর্যাইউআইথের্ডবন্ধকরে

বিটম্যাপপর্ষিকর্যা(রিসাইজকরা,একটিদূরবতর্ীসেসার্থেকে
ডাউনলেডকরা,ইত্যাদী)করাকখনইইউজারইন্টারফেসথের্ডেরজায়গা
নেওয়াউচিতনা।এইঅনুশীলনীআপনাকেদেখাবেAsyncTaskব্যবহারকরে
একটিব্যাকগ্রাউন্ডথের্ডেরমধ্যেবিটম্যাপপর্ষেসসকরতেহয়এবং
ব্যবহারকেরকীভাবেকনকারেনিংসইস্যুয়পরিচালনাকরতেহয়।

বিটম্যাপজিমেরাখা

এইঅনুশীলনীআপনাকেদেখাবেযখনমাল্টিপলবিটম্যাপলেডকরাহয়
তখনআপনারইউজারইন্টারফেসেরসপিন্সভেনেসএবংফ্লুয়িডিটি
উন্নতকরতেকীভাবেমেমরীএবংিডস্কবিটম্যাপক্যাশব্যবহার
করতেহয়

বিটম্যাপমেমরীপরিচালনাকরা

এইঅনুশীলনীব্যবহারকেরআপনারঅ্যাপপারফরেন্সবাড়াতে
কীভাবেবিটম্যাপমেমরীপরিচালনাকরতেহয়।

আপনারইউজারইন্টারফেসেবিটম্যাপপর্ষদর্শন

এইঅনুশীলনীসবিকছুএকসাথেখিনিয়েআসে,আপনাকেদেখাবেএটি
ব্যাকগ্রাউন্ডথের্ডএবংবিটম্যাপক্যাশব্যবহারকরে
ViewPagerএবংGridViewমতাকমেপানেন্টরমধ্যেকীভাবেমাল্টিপল
বিটম্যাপলেডকরতেহয়

বড় আকারের বিটম্যাপ দক্ষতার সাথে লোড করা

(<http://developer.android.com/training/displaying-bitmaps/load-bitmap.html>)

ইমেজ সকল আকার ও গঠন নিয়ে আসতে পারে। অনেকক্ষেত্রে একটি সাধারণ অ্যাপলিকেশন ইউজার ইন্টারফেসের জন্য প্রয়োজনের চেয়ে বড় হতে পারে। উদাহরণস্বরূপ, সিস্টেম গ্যালারী অ্যাপলিকেশন আপনার অ্যান্ড্রয়েড ডিভাইসের কয় মরাবয় বহার করে নেয়া ফটো পর্দাশর্ন করে যা সাধারণত আপনার স্ক্রিনের যতটুকু চেষ্টা করে উচ্চ রেজুলেশন যুক্ত।

দেয়া আছে যে আপনি সীমিত মেমরী নিয়ে কাজ করেছেন, এটাই আদর্শ যে আপনি শুধু মেমরীতে কম রেজুলেশনের সংস্করণ লোড করবেন। কম রেজুলেশনের সংস্করণ আপনার ইউজার ইন্টারফেসের কম্পোনেন্ট যা এটা পর্দাশর্ন করে তার সাথে সাথে ম্যাচ করা উচিত। একটি উচ্চ রেজুলেশন যুক্ত ইমেজকে কান দৃশ্যমান সুবিধা পর্দাশর্ন করে না, কিন্ত এখনও মূল্যবান মেমরী নিয়ে নেয়া এবং ফ্লাই সেকিল এ যুক্ত তার কারণে অতিরিক্ত পারফরমেন্স ওভারহেড ডেকে আনে।

এই অনুশীলনীতে একটি ছোট সাবসাম্পলড সংস্করণ লোড করার মাধ্যমে পর্দাশর্ন অ্যাপলিকেশনের মেমরীর সীমা অতিক্রম না করে বড় আকারের বিটম্যাপ ডিকেডি করার কাজটি দেখাবে।

বিটম্যাপিং পেরমাতর্ (ডাইমেনশন) এবং টাইপ পড়ুন

বিভিন্ন উৎস থেকে একটি Bitmap তৈরী করার জন্য BitmapFactory ক্লাস কিছু ডিকোডিং মেথড (decodeByteArray(), decodeFile(), decodeResource() ইত্যাদী) সরবরাহ করে। আপনার ইমেজ ডাটাসেটের উপর ভিত্তি করে সবচেয়ে উপযুক্ত মেথড পছন্দ করুন। এই মেথডগুলো গঠনহওয়া বিটম্যাপিং এর জন্য মেমরী বরাদ্দ দরপদক্ষেপ নেয় এবং সহজেই একটি OutOfMemory এক্সপেশনের মধ্যে ফলাফল নিয়ে আসে। ডিকোডের পরে টাধরনের অতিরিক্ত সিগনেচার থাকে যা আপনাকে BitmapFactory.Options ক্লাসের মধ্যে ডিকোডিং অপশন নির্দিষ্ট করতে দেয়। true এ inJustDecodeBounds পরপাট্র সেট করা যখন ডিকোডিং মেমরী বরাদ্দ (এলোকেশন) পরিহার করে, বিটম্যাপিং অবজেক্টের জন্য হাষফেরত আনেকিন্ত outWidth, outHeight এবং outMimeType সেট করে। এই কৌশল বিটম্যাপিং এর নিম্নরানের (এবং মেমরী এলোকেশন) পূর্বে আপনার ইমেজ ডাটার ডাইমেনশন এবং টাইপ পড়তে দেয়

```
BitmapFactory.Options options = new BitmapFactory.Options();
options.inJustDecodeBounds = true;
BitmapFactory.decodeResource(getResources(), R.id.myimage, options);
int imageHeight = options.outHeight;
int imageWidth = options.outWidth;
String imageType = options.outMimeType;
```

java.lang.OutOfMemory এক্সপেশন পরিহার করতে, ডিকোড করার পূর্বে একটি বিটম্যাপিং পের ডাইমেনশন চেক করুন, যদি না আপনি সম্পূর্ণরূপে সেটসেরকে বিশ্বাস করেন যা আপনাকে অনুমিত সাইজের ইমেজ ডাটাসেট সহজেই বিদ্যমান মেমরীতে ফিট করে বতাপর্দান করতে।

মেমরীতে একটি সেস্কলডাউন- সংস্করণ লোড করুন

এখন ইমেজ ডাইমেনশন জানা হয়েছে, মেমরীতে পূর্ণ ইমেজ লোড করা উচিত না কি পরিবেশের একটি সাবসেট অ্যাম্পলড সংস্করণ লোড করা উচিত তা ঠিক করতে তাদের এখন ব্যবহার করা যেতে পারে। এখানে কিছু বিষয় বিবেচনার জন্য তুলে ধরা হলো:

- মেমরীতে পূর্ণ ইমেজ লোড করার পূর্বর্তী ভাষা করা মেমরী ব্যবহার
- মেমরীর যে পরিমাণ এই ইমেজ লোড করার জন্য আপনি ঠিক করে রেখেছেন তা যদি আপনার অ্যাপ্লিকেশনের অন্য কোন মেমরী দাবী করে।
- ট্যাগের ImageView এর ডাইমেনশন বা ইউজার ইন্টারফেস কম্পোনেন্ট যার মধ্যে ইমেজ লোড হবে
- চলতি ডিভাইসের সংস্করণ সাইজ এবং যন্ত্র

উদাহরণস্বরূপ, এটা মেমরীতে একটি 1024x768 pixel ইমেজ লোড করার যে অর্থাৎ নয় যদি এটা একটি ImageView এর মধ্যে আগে বা পরে একটি 128x96 pixel থাম্বনেইল পরদা দর্শিত হয়।

ইমেজকে সাবসেট অ্যাম্পল করতে ডিকোডারকে বলতে, মেমরীতে একটি ছোট সংস্করণ লোড করুন, আপনার BitmapFactory.Options অবজেক্টের true এ inSampleSize সেট করুন। উদাহরণস্বরূপ, 2048x1536 রেজলুশন সহকারে একটি ইমেজ যা ৪ এর inSampleSize দিয়ে ডিকোড করা একটি আনুমানিক 512x384 এর একটি বিটম্যাপ উপস্থাপন করে। এটা মেমরীতে লোড করতে পূর্ণ ইমেজের (ARGB_8888 এর একটি বিটম্যাপ কনফিগারেশন ধারণ করে) জন্য ১২ MB না ব্যবহার করে ০.৭৫ MB ব্যবহার করে। সংস্করণ সাইজ ভয়াল ক্যালকুলেট করতে এখানে একটি মেথড দেয়া আছে যা একটি ট্যাগের টাইমস এবং উচ্চতার উপর ভিত্তি করে দুইয়ের ক্রমত।

```
public static int calculateInSampleSize(
    BitmapFactory.Options options, int reqWidth, int reqHeight) {
    // Raw height and width of image
    final int height = options.outHeight;
    final int width = options.outWidth;
    int inSampleSize = 1;

    if (height > reqHeight || width > reqWidth) {

        final int halfHeight = height / 2;
        final int halfWidth = width / 2;

        // Calculate the largest inSampleSize value that is a power of 2 and keeps both
        // height and width larger than the requested height and width.
        while ((halfHeight / inSampleSize) > reqHeight
            && (halfWidth / inSampleSize) > reqWidth) {
            inSampleSize *= 2;
        }
    }

    return inSampleSize;
}
```

নোট: দুইটা ভয়াল প্যাওয়ার ক্যালকুলেট করা হয় কারণ দুইয়ের নিকটস্থ প্যাওয়ারের রাউন্ড ডাউন করার মাধ্যমে ডিকোডার একটি ফাইনাল ভয়াল ব্যবহার করে, inSampleSize ডকুমেন্টেশন অনুসারে।

এই মেথড ব্যবহার করতে, inJustDecodeBounds সহকারে পৃথক ডিকোড true এ সেট করা, এর মাধ্যমে অংশন পাস করা, তারপর ডিকোড করুন আবার নতুন inSampleSize ভয়াল এবং inJustDecodeBounds ব্যবহার করে false এ সেট করুন:

```

public static Bitmap decodeSampledBitmapFromResource(Resources res, int resId,
    int reqWidth, int reqHeight) {

    // First decode with inJustDecodeBounds=true to check dimensions
    final BitmapFactory.Options options = new BitmapFactory.Options();
    options.inJustDecodeBounds = true;
    BitmapFactory.decodeResource(res, resId, options);

    // Calculate inSampleSize
    options.inSampleSize = calculateInSampleSize(options, reqWidth, reqHeight);

    // Decode bitmap with inSampleSize set
    options.inJustDecodeBounds = false;
    return BitmapFactory.decodeResource(res, resId, options);
}

```

এই মেথড একটা `ImageView` এর মধ্যে ইচ্ছামত বড় সাইজের একটি বিটম্যাপ লোড করাকে এটা সহজ করে দেয় যা একটি `100x100 pixel` থামব্‌নেইল পর্দাশর্ন করে, যেভাবে নীচের উদাহরণকে ডেডওয়াইয়েছে:

```

mImageView.setImageBitmap(
    decodeSampledBitmapFromResource(getResources(), R.id.myimage, 100, 100));

```

আপনি অন্য রিসোর্স থেকে বিটম্যাপ ডিকোড করতে একই মেথড অনুসরণ করতে পারেন, যথায় `BitmapFactory.decode*` মেথড পরয়োজন অনুসারে পরিস্‌থাপন করার মাধ্যমে।

বিটম্যাপ পরিকর্যা ইউআই থের্ড বন্ধ করে

(<http://developer.android.com/training/displaying-bitmaps/process-bitmap.html>)

BitmapFactory.decode* মেথড, Load Large Bitmaps Efficiently (বড় আকারের বিটম্যাপ দক্ষতার সাথে লোড করা) অনুশীলনীতে আলোচনা করা হয়েছে, পর্ধান ইউজার ইন্টারফেস থের্ডে কায়র কর করা উচিত নয় যদি সে সাসর্ ডাটা ডিস্ক বা একটি নেটওয়ার্ক লোকেশন থেকে (বাসতিয়কার অথের অনয সাসর্ মেমরী নয়) পড়া হয়েছে। লোড করতে ডাটা যে সময়ে নেয় তা অনুমান করা যায় না এবং বিভিন্ন ঘটনার উপর নির্ভর করে (ডিস্ক বা নেটওয়ার্ক থেকে পড়ার স্পড, ইমেজ সাইজ, সিসিপিইউ এর পাওয়ার ইত্যাদি)। যদি এই কাজগুলোর একটি ইউআই থের্ড ব্লক করে দেয়, সিস্টেমটি আপনার অ্যাপলিকেশন নন-রেসপনসিভ হিসাবে থামিয়ে দেয় এবং ইউজারের এটা বন্ধ করাটাকে বোঝাতে হয় (আরও তথ্যের জন্য Designing for Responsiveness: <http://developer.android.com/training/articles/perf-anr.html> দেখুন)।

এই অনুশীলনী আপনাকে দেখায় কীভাবে AsyncTask ব্যবহার করে একটি ব্যাকগ্রাউন্ডে থের্ডে বিটম্যাপ পরেসিস করতে হয় এবং আরও দেখায় কীভাবে কনকারেন্স ইস্যুয় চালানা করতে হয়।

Async Task ব্যবহার করুন

AsyncTask ক্লাস একটি ট্যাকগরান্ড থ্রেডে ডিক্রু কাজ সম্পাদনের জন্য সহজ রাস্তা পর্দান করে এবং ইউআই থ্রেডে ফলাফল পর্কাশ করে। এটা ব্যবহার করতে, একটি সাবক্লাস তৈরী করুন এবং পর্দত তে মথড ওভাররাইড করুন। এখানে AsyncTask এবং decodeSampledBitmapFromResource() ব্যবহার করে একটি ImageView এর মধ্যে একটি বড় আকারের ইমেজ লোড করার উদাহরণ দেয়া হলে:

```
class BitmapWorkerTask extends AsyncTask<Integer, Void, Bitmap> {
    private final WeakReference<ImageView> imageViewReference;
    private int data = 0;

    public BitmapWorkerTask(ImageView imageView) {
        // Use a WeakReference to ensure the ImageView can be garbage collected
        imageViewReference = new WeakReference<ImageView>(imageView);
    }

    // Decode image in background.
    @Override
    protected Bitmap doInBackground(Integer... params) {
        data = params[0];
        return decodeSampledBitmapFromResource(getResources(), data, 100, 100);
    }

    // Once complete, see if ImageView is still around and set bitmap.
    @Override
    protected void onPostExecute(Bitmap bitmap) {
        if (imageViewReference != null && bitmap != null) {
            final ImageView imageView = imageViewReference.get();
            if (imageView != null) {
                imageView.setImageBitmap(bitmap);
            }
        }
    }
}
```

ImageView এর পর্তি WeakReference নিশ্চিত করে যে AsyncTask ImageView কে এবং গারে বজ হওয়া কে কান ক্রু কে রফারেন্স করে তা বন্ধ করে না। এটার কান নিশ্চয়তানেই যখন কাজ শেষ হবে তখনও আশপাশে ImageView থাকবে, তাই আপনাকে অবশ্যই onPostExecute() এর মধ্যে রফারেন্সে চেক করতে হবে। ImageView এর অস্ ততব্ আর নাও থাকতে পারে, যদি উদাহরণ সর্বরূপ, একটি টিভি টেথেকে নেভিগেট করে চলে যায় বা যদি কাজ শেষ হওয়ার পূর্বেই কনফিগারেশনের পর্বতর্ন ঘটে থাকে

বিটম্যাপ আপ আিসংকে র্ানাসিল (asynchronously) লোড শুরুর করতে চাইলে, শুধু নতুন কাজ তৈরী করুন এবং সম্পাদন করুন:

```
public void loadBitmap(int resId, ImageView imageView) {
    BitmapWorkerTask task = new BitmapWorkerTask(imageView);
    task.execute(resId);
}
```

কনকারেবিন্স চালনা করা

কমন ভিউ কমেপোনেন্ট যেমন ListView এবং GridView অন্য হিসুয়ে শুরুর করে যখন AsyncTask দিয়ে কনজাঙ্কশনে ব্যবহার করা হয় যেভাবে পূর্ববর্তের সেকশনে দেখানো হয়েছে। মেমরীর সাথে সুসংগঠিত হওয়ার জন্য, এই কমেপোনেন্ট গুলো চাইল্ড ভিউকে ইউজার স্ক্রলের মতো রিসাইকেল করে। যদি পর্তিটা চাইল্ড ভিউ একটি AsyncTask সিকর্য করে, এখানে কোন নিশ্চয়তাই নেই যে যখন এটা শেষ হবে, সহায়তাকারী ভিউ অন্য চাইল্ড ভিউ এবং ব্যবহারের জন্য ইতিমধ্যে রিসাইকেল হয় নাই। এছাড়াও, এখানে কোন নিশ্চয়তাই নেই যে বিনয়াস যার মধ্যে সিঙকের নাস এর কাজ শুরু হয় সেটা সেই বিনয়াস যা তার সম্পূর্ণ করে।

ব্লগ স্পট Multithreading for Performance (<http://android-developers.blogspot.com/2010/07/multithreading-for-performance.html>) কনকারেবিন্স নিয়ে কাজ করার বিষয়ে বিস্তারিত আলোচনা করেছে, এবং একটি সমাধান পরস্ তাব করেছে যেখানে ImageView একটি রেফারেন্স সদয় AsyncTask এ স্টার করে যা কে পরবর্তীতে চেক করা হয় যখন কাজ শেষ হয়। একটি সমপয়র্ ায়ের মেথড ব্যবহার করা, AsyncTask পূর্ববর্তী সেকশনে থেকে একটি সমরূপ বৈশিষ্ট্য অনুসরণ করে AsyncTask সম্পর্সারিত হতে পারে।

ওয়াকরার টাসে ক্রিফের একটি রেফারেন্স সে স্টার করতে একটি নিবেদিত Drawable সাবক্লাস তৈরী করুন। এক্ষেত্রে একটি BitmapDrawable ব্যবহার করা হয় যাতে একটি প্লেসহোল্ডার ইমেজ ImageView এর মধ্যে পর্দিশর্ত হতে পারে যখন কাজটি সম্পূর্ণ হবে:

```
static class AsyncDrawable extends BitmapDrawable {
    private final WeakReference<BitmapWorkerTask> bitmapWorkerTaskReference;

    public AsyncDrawable(Resources res, Bitmap bitmap,
        BitmapWorkerTask bitmapWorkerTask) {
        super(res, bitmap);
        bitmapWorkerTaskReference =
            new WeakReference<BitmapWorkerTask>(bitmapWorkerTask);
    }

    public BitmapWorkerTask getBitmapWorkerTask() {
        return bitmapWorkerTaskReference.get();
    }
}
```

BitmapWorkerTask সম্পাদন করার পূর্বে, আপনি একটি AsyncDrawable তৈরী করেন এবং এটা কে টাগের্ট ImageView এ স্যু ক্ত করেন:

```
public void loadBitmap(int resId, ImageView imageView) {
    if (cancelPotentialWork(resId, imageView)) {
        final BitmapWorkerTask task = new BitmapWorkerTask(imageView);
        final AsyncDrawable asyncDrawable =
            new AsyncDrawable(getResources(), mPlaceholderBitmap, task);
        imageView.setImageDrawable(asyncDrawable);
        task.execute(resId);
    }
}
```

কেড স্যাম্পল এর মধ্যে রেফারেন্স করা cancelPotentialWork মেথড উপরে চেক করে যদি অন্য রািনিং কাজ ImageView এর সাথে ইতিমধ্যে যুক্ত হয়। যদি তাই হয়, cancel() কল করার মাধ্যমে পূর্ববর্তের কাজ ক্যান্সেল করার পর্চেষ্টা চালানো হয়। খুব কম কেষ্টে নতুন টাস্ক ডাটা বিদয়মান টাস্ক চেক করে এবং আর কিছু ইটার পর্য়োজন হয় না। এখানে cancelPotentialWork: এর বাস্ তাবায়ন আছে:


```

public static boolean cancelPotentialWork(int data, ImageView imageView) {
    final BitmapWorkerTask bitmapWorkerTask = getBitmapWorkerTask(imageView);

    if (bitmapWorkerTask != null) {
        final int bitmapData = bitmapWorkerTask.data;
        if (bitmapData != data) {
            // Cancel previous task
            bitmapWorkerTask.cancel(true);
        } else {
            // The same work is already in progress
            return false;
        }
    }
    // No task associated with the ImageView, or an existing task was cancelled
    return true;
}

```

একটি সাহাজ্যকারী মেথড, `getBitmapWorkerTask()`, উপরে বয়বহার করা হয়েছে একটি বিশেষ ও সমন্বিত রবটিকার সাথে যুক্ত কাজকে পূরণরুদ্ধ করার করতে:

```

private static BitmapWorkerTask getBitmapWorkerTask(ImageView imageView) {
    if (imageView != null) {
        final Drawable drawable = imageView.getDrawable();
        if (drawable instanceof AsyncDrawable) {
            final AsyncDrawable asyncDrawable = (AsyncDrawable) drawable;
            return asyncDrawable.getBitmapWorkerTask();
        }
    }
    return null;
}

```

শেষ ধাপ হচ্ছে `BitmapWorkerTask` এ `onPostExecute()` অপারেটর করা যাতে এটা চেক করতে পারেন কাজটি কয়ান সেল হয়েছে কিনা এবং চলিত কাজটি `ImageView` এর সাথে সমাপ্তিকরত এককটি ময়চ করে ছে কিনা:

```

class BitmapWorkerTask extends AsyncTask<Integer, Void, Bitmap> {
    ...

    @Override
    protected void onPostExecute(Bitmap bitmap) {
        if (isCancelled()) {
            bitmap = null;
        }

        if (imageViewReference != null && bitmap != null) {
            final ImageView imageView = imageViewReference.get();
            final BitmapWorkerTask bitmapWorkerTask =
                getBitmapWorkerTask(imageView);
            if (this == bitmapWorkerTask && imageView != null) {
                imageView.setImageBitmap(bitmap);
            }
        }
    }
}

```

এই বাস্তুবায়ন এখন `ListView` এবং `GridView` কেম্পোনেন্ট একই সাথে অনয় কেম্পোনেন্ট বয়বহার করার জন্য উপযুক্ত যাতায়ে দরচাইল্ডিভিউ রিসাইকেল করে। সাধারণভাবে `loadBitmap` কল করুন যেখানে আপনি সব্রাভািবকভাবে আপনার `ImageView` এ একটি ইমেজে সেট করে ছন। উদাহরণসব্রূপ, একটি `getView` বাস্তুবায়নের মধ্যে য় এটা বয় াকি অয়্রাডাপ্রটের `getView()` মেথডের মধ্যে য় হতে পারে।

বিটম্যাপ জিমের রাখা (ক্যাশিং)

(<http://developer.android.com/training/displaying-bitmaps/cache-bitmap.html>)

একটি সিস্টেমে গুলি বিটম্যাপ আপনার ইউজার ইন্টারফেসে লোড করা খুব সহজ, কিন্তু তুলে বিষয়টিকে জটিল করে দেয় যদি আপনাকে একটি বড় আকারের ইমেজ সেট একই সাথে লোড করতে হয়। অনেক ক্ষেত্রে (যেমন, ListView, GridView বা ViewPager এর মতো কন্ট্রোল প্যানেল টি দিয়ে), সিস্টেমের মেমোরি ইমেজের পরিমাণ কমানোর ইমেজ দ্বারা একটি রত যাকি শর ই সিস্টেমের উপরে স্ক্রল হবে তা মূলত সীমাহীন।

মেমরী ব্যবহার এটার মতো কন্ট্রোল প্যানেল টির সাথে নীচে রাখা হয় চাইল্ড ডিভিউ রিসাইকেল করার মাধ্যমে যেভাবে তার অফ-সিস্টেম মুভ করে। গারবেজ কালেক্টর (অপর্যে জনীয় তথ্য) আপনাকে লোড করা বিটম্যাপ ছেড়ে দেয়, আপনাকে জানিয়ে দেয় যাকি কোন দীর্ঘায়ুর রেফারেন্স রাখা বেন না। এগুলো লাভালা, কিন্তু একটি ফ্লুইড এবং ফাস্ট-লোডিং ইউজার ইন্টারফেস রাখার জন্য আপনাকে এই ইমেজ গুলোর পরিত্যক্ত পরিকল্পনা করা পরিহার করতে চাইতে পারেন পরিত্যক্ত তার যাখন অন-সিস্টেম ফিরে আসে। একটি মেমরী এবং ডিস্ক ক্যাশিং এখানে নিয়মিত সাহায্য করতে পারে, পরেস করা ইমেজ রিলোড করতে কন্ট্রোল প্যানেল টিকে অনুরোধ দিতে।

এই অনুশীলনী আপনাকে দেখাবে যখন মাল্টিপল বিটম্যাপ লোড করা হয় তখন কীভাবে আপনাকে ইউজার ইন্টারফেসে ফ্লুইডিটি এবং রেসপনসিভনেস উন্নত করতে একটি মেমরী এবং ডিস্ক বিটম্যাপ ব্যবহার করতে হয়।

একটি মেমরী কয়্‌শে বয়্‌বহার

একটি মেমরী কয়্‌শে মূল্যবান অয়্‌পলিকেশন মেমরীর খরচে চরিত্বিনময়ে বিটময়্‌পে দরুত পরবেশগময়্‌তাপরস্‌তাবকরে। LruCache কল্লাস (এটা Support Library তে ও পাওয়া যায়, এপিআই লেভেল ৪ এ ফিরে বয়্‌বহারের জন্যে) কয়্‌শি শিবিটময়্‌পে পরটােস্‌কর জনয়্‌বিশেষ করে উপযোগী, একটি শিক্‌তশালী রেফারেন্স করা LinkedHashMap এর মধ্যে সম্‌পর্‌তি রেফারেন্স করা বস্‌তু রাখে এবং কয়্‌শের নিধর্‌রিত সাইজ ছাড়িয়ে যাওয়ার পূর্‌বের্‌ কম সাম্‌পর্‌তিক বয়্‌বহৃত মেমব্‌র উচ্‌ছদ করে।

নেট: অতীতে, একটি জনপিয় মেমরী কয়্‌শ বাস্‌তবায়ন ছিল SoftReference বা WeakReference বিটময়্‌প কয়্‌শে, কিন্তু এটা সুপারিশ করা হয় না।

অয়্‌ান্ডরুয়েড ২.৩ (এপিআই লেভেল ৯) থেকে শুরু করে সফ্‌ট/উইক রেফারেন্স কালেকশন করা সহগারে বজ্‌কালে ক্‌টর আরও আকর্‌মনাতব্‌ক যা তাে দরেক কিকছু টা অকায়র্‌কর করে। উপরন্‌তু, অয়্‌ান্ডরুয়েড ৩.০ (এপিআই লেভেল ১১) এর পূর্‌বের্‌, একটি বিটময়্‌পে বয়্‌কি ডাটা একটি স্‌থানীয় (নেটিভ) মেমরীতে সেস্‌টার হতে যা অনুমেয় উপায়ে রিলিজ হয় না, সম্‌ভাবয়্‌ একটি অয়্‌পলিকেশনকে একটি স্‌কিষপ্‌ত সময়ের জন্যে এর মেমরী সীমাকে অতিকর্‌ম করার এবং ধব্‌স করার কারণ হয়।

একটি LruCache এর জন্যে উপযুক্ত সাইজ পছন্‌দ করার জন্যে, কয়েকটি ফয়্‌ক্‌টর বিবেচনাযরাখতে হবে:

- কতটুকু মেমরী ইন্‌টেনসিভ হচ্‌ছ আপনার একটি ভিডিও এবং/বা অয়্‌পলিকেশনের বাকী অংশ?
- একসাথে কতগুলো ইমেজ অন-িস্কর্‌ন হতে পারে? কতগুলো অন-িস্কর্‌নে পরস্‌তুত হয়ে আসতে সহজপর্‌াপয়্‌ হওয়ার পর্‌য়োজন?
- ডিভাইসের সিস্কর্‌নের সাইজ এবং ঘনতব্‌কি? Galaxy Nexus এর মতে একটি এক্‌সটর্‌াহাই ডেনজিটি সিস্কর্‌নের (xhdpi) একটি মেমরী Nexus S (hdpi) এর মতে ডিভাইসে সাথে তুলনীয় তার মধ্যে একই স্‌খয়্‌ক ইমেজ ধারন করতে একটি বড় আকর্‌রের কয়্‌শে দরকার হবে।
- বিটময়্‌পে ডাইমেনশন এবং কনফিগারেশনিক এবং তারপর পর্‌তিটাকি পরিমান মেমরী নিবে?
- কত যগ ঘন এই ইমেজে পর্‌বেশ করা হতে পারে? কোনটাতে কি অনয়্‌টার চেয়ে বেশী বার পর্‌বেশ করা হতে পারে? যদি হয়, সম্‌ভবত আপনি সবসময় কিকছু আইটেম মেমরীতে রাখতে চাইতে পারেন বা বিটময়্‌পে পরিভিন্‌ন গর্‌পে জনয়্‌ এমনিক মাল্‌টপল LruCache অবজেক্ট আছে।
- আপনি কি পরিমাণের বিপরীতে গুণ এর ভারসাময়্‌রক্‌ষাকরতে পারবেন? কিকছু ক্‌ষেতর্‌ এটাকম গুণসম্পন্‌ন বিটময়্‌পে একটি বড় স্‌খয়্‌কে সেস্‌টার করার জন্যে আরও উপকারী হতে পারে, অনয়্‌ বয়্‌কগর্‌াউন্‌ড টােস্‌কর মধ্যে একটি উচ্‌চ গুণসম্পন্‌ন স্‌স্করণ সম্‌ভাবয়্‌লে ডকরার জন্যে।

এখানে কোন নির্‌দস্‌ট সাইজ এবং ফমুর্‌লানেই যা সকল অয়্‌পলিকেশনের জন্যে উপযোগী, এটা নিভর্‌র করে আপনার বয়্‌বহারের বিশ্‌লষণ এবং একটি সুন্‌দর সমাধানে আসার উপর। একটি কয়্‌শে যা খুবই ছোট তাে কোন সুবিধা ছাড়াই অতিরিক্‌ত ও ভারে হড যটায়, একটি কয়্‌শে যা বেশী বড় java.lang.OutOfMemory এক্‌সেসপশন ঘটায় এবং আপনার অয়্‌পে বাকী ক্‌ষুদর মেমরী এর সাথে কাজ করার জন্যে ছেড়ে দেয়।

এখানে বিটময়্‌পে জনয়্‌ একটি LruCache সেটআপে একটি উদাহরণ আছে:

```

private LruCache<String, Bitmap> mMemoryCache;

@Override
protected void onCreate(Bundle savedInstanceState) {
    ...
    // Get max available VM memory, exceeding this amount will throw an
    // OutOfMemory exception. Stored in kilobytes as LruCache takes an
    // int in its constructor.
    final int maxMemory = (int) (Runtime.getRuntime().maxMemory() / 1024);

    // Use 1/8th of the available memory for this memory cache.
    final int cacheSize = maxMemory / 8;

    mMemoryCache = new LruCache<String, Bitmap>(cacheSize) {
        @Override
        protected int sizeOf(String key, Bitmap bitmap) {
            // The cache size will be measured in kilobytes rather than
            // number of items.
            return bitmap.getByteCount() / 1024;
        }
    };
    ...
}

public void addBitmapToMemoryCache(String key, Bitmap bitmap) {
    if (getBitmapFromMemCache(key) == null) {
        mMemoryCache.put(key, bitmap);
    }
}

public Bitmap getBitmapFromMemCache(String key) {
    return mMemoryCache.get(key);
}

```

নোট: এই উদাহরনে, আউটভাগের একভাগ অ্যাপলিকেশন মেমরী আপনার কয়ালে শের জন্য বরাদ্দ। একটি নরমাল/hdpi ডিভাইসে এটানুয্যনতম ৪ MB এর কাছাকাছি (৩২/৮)। একটি ডিভাইসে ৮০০x৪৮০ রেজুয়লেশন সহকারে একটি ফুল স্ক্রিন GridView এ পূরণ করে থাকাইমেজ পরায় ১.৫ MB (৮০০x৪৮০x৪ bytes) ব্যবহার করতে পারে, সুতরাং এটানিদে নপক্ষে ইমেজে র পরায় ২.৫ পজ মেমরীতে জিমিয়ে রাখা।

যখন একটি ImageView এর মধ্যে একটি বিটম্যাপ লোড করা হয়, LruCache পর্থমে চেক করা হয়। যদি একটি এনিম্টর্পাওয়া যায়, এটাতা ক্রনিকভাবে ImageView আপডেট করতে ব্যবহৃত হয়, অন্যথায় ইমেজ পরেসস করতে একটি ব্যব্রাকগর্রাউন্ড থের্ডউপন্ন হয়:

```

public void loadBitmap(int resId, ImageView imageView) {
    final String imageKey = String.valueOf(resId);

    final Bitmap bitmap = getBitmapFromMemCache(imageKey);
    if (bitmap != null) {
        mImageView.setImageBitmap(bitmap);
    } else {
        mImageView.setImageResource(R.drawable.image_placeholder);
        BitmapWorkerTask task = new BitmapWorkerTask(mImageView);
        task.execute(resId);
    }
}

```

BitmapWorkerTask এর মেমরী কয়ালে শেতে যুক্ত হতে আপডেট হওয়া পর্যেজন:

```
class BitmapWorkerTask extends AsyncTask<Integer, Void, Bitmap> {
    ...
    // Decode image in background.
    @Override
    protected Bitmap doInBackground(Integer... params) {
        final Bitmap bitmap = decodeSampledBitmapFromResource(
            getResources(), params[0], 100, 100);
        addBitmapToMemoryCache(String.valueOf(params[0]), bitmap);
        return bitmap;
    }
    ...
}
```

ডিস্ক ক্যাশে ব্যবহার

একটি মেমরী ক্যাশে সাম্প্রতিক ভিউ হওয়া বিটম্যাপ একটি গতিশীল পরবেশের জন্য উপকারী, কিন্তু আপনি এই ক্যাশেতে ইমেজের সহজপাঠ্য হওয়ার উপর নির্ভর করতে পারেন না। বড় আকারের ডাটাসেট সহকারে GridView এর মতো কেম্পোনেন্ট সহজেই একটি মেমরী ক্যাশে পূরণ করতে পারে। আপনার অ্যাপলিকেশন অন্য কোন টাসকে যেমন একটি ফোন কল দ্বারা বাধাগ্রস্ত হতে পারে, এবং যখন ব্যাকগ্রাউন্ডের মধ্যে এটা হতে পারে এবং মেমরী ক্যাশে ধ্বংস হবে। যখনই ইউজার আবার শুরুর করে, আপনার অ্যাপলিকেশনকে পরিত্যাগ ইমেজ আবার শুরুর করতে হয়।

একটি ডিস্ক ক্যাশে এই ক্ষেত্রে পরে সসকরা বিটম্যাপ অটলভাবে চািলিয়ে যাওয়ার কাজে ব্যবহার করা যেতে পারে এবং লোড করার সময় কমতে পারে যেখানে ইমেজ মেমরী ক্যাশের মধ্যে আর পাওয়া যাবে না। অবশ্যই, ডিস্ক ক্যাশে ইমেজ আনয়ন করা মেমরী থেকে লোড করার চেয়ে ধীরগতির এবং অবশ্যই ব্যাকগ্রাউন্ডে থের্ডেড সম্পন্ন করতে হবে, যেহেতু ডিস্ক ক্যাশে সব সময় অননুমোদিত।

নোট: ক্যাশে ইমেজ সেট করার করতে ContentProvider একটা ভালো জায়গা হতে পারে, যদি তাতে বারবার পরবেশ করা হয়, উদাহরণস্বরূপ, একটি ইমেজ গ্যালারী অ্যাপলিকেশনের মধ্যে।

এই ক্লাসের নমুনা কোড একটি DiskLruCache বাস্তবায়ন ব্যবহার করে যাকে Android source (<https://android.googlesource.com/platform/libcore/+jb-mr2-release/luni/src/main/java/libcore/io/DiskLruCache.java>) থেকে টানা হয়। এখানে একটি আপডেট উদাহরণে কোড আছে যা বিদ্যমান মেমরী ক্যাশে অতিরিক্ত হিসাবে একটি ডিস্ক ক্যাশে যুক্ত করে:

```
private DiskLruCache mDiskLruCache;
private final Object mDiskCacheLock = new Object();
private boolean mDiskCacheStarting = true;
private static final int DISK_CACHE_SIZE = 1024 * 1024 * 10; // 10MB
private static final String DISK_CACHE_SUBDIR = "thumbnails";

@Override
protected void onCreate(Bundle savedInstanceState) {
    ...
    // Initialize memory cache
    ...
    // Initialize disk cache on background thread
    File cacheDir = getDiskCacheDir(this, DISK_CACHE_SUBDIR);
    new InitDiskCacheTask().execute(cacheDir);
    ...
}

class InitDiskCacheTask extends AsyncTask<File, Void, Void> {
    @Override
    protected Void doInBackground(File... params) {
        synchronized (mDiskCacheLock) {
            File cacheDir = params[0];
            mDiskLruCache = DiskLruCache.open(cacheDir, DISK_CACHE_SIZE);
            mDiskCacheStarting = false; // Finished initialization
            mDiskCacheLock.notifyAll(); // Wake any waiting threads
        }
        return null;
    }
}

class BitmapWorkerTask extends AsyncTask<Integer, Void, Bitmap> {
    ...
    // Decode image in background.
    @Override
    protected Bitmap doInBackground(Integer... params) {
        final String imageKey = String.valueOf(params[0]);

        // Check disk cache in background thread
        Bitmap bitmap = getBitmapFromDiskCache(imageKey);
    }
}
```

```

        if (bitmap == null) { // Not found in disk cache
            // Process as normal
            final Bitmap bitmap = decodeSampledBitmapFromResource(
                getResources(), params[0], 100, 100));
        }

        // Add final bitmap to caches
        addBitmapToCache(imageKey, bitmap);

        return bitmap;
    }
    ...
}

public void addBitmapToCache(String key, Bitmap bitmap) {
    // Add to memory cache as before
    if (getBitmapFromMemCache(key) == null) {
        mMemoryCache.put(key, bitmap);
    }

    // Also add to disk cache
    synchronized (mDiskCacheLock) {
        if (mDiskLruCache != null && mDiskLruCache.get(key) == null) {
            mDiskLruCache.put(key, bitmap);
        }
    }
}

public Bitmap getBitmapFromDiskCache(String key) {
    synchronized (mDiskCacheLock) {
        // Wait while disk cache is started from background thread
        while (mDiskCacheStarting) {
            try {
                mDiskCacheLock.wait();
            } catch (InterruptedException e) {}
        }
        if (mDiskLruCache != null) {
            return mDiskLruCache.get(key);
        }
    }
    return null;
}

// Creates a unique subdirectory of the designated app cache directory. Tries to use external
// but if not mounted, falls back on internal storage.
public static File getDiskCacheDir(Context context, String uniqueName) {
    // Check if media is mounted or storage is built-in, if so, try and use external cache dir
    // otherwise use internal cache dir
    final String cachePath =
        Environment.MEDIA_MOUNTED.equals(Environment.getExternalStorageState()) ||
        !isExternalStorageRemovable() ? getExternalCacheDir(context).getPath() :
        context.getCacheDir().getPath();

    return new File(cachePath + File.separator + uniqueName);
}

```

নেটাট: ডিস্ক ক্যাশে আরম্ভ করা ডিস্ক অপারেশন আশা করতে পারে এবং সেজন্য মেইন থ্রেডে ঘটা উচিত না। কিন্ত তু, এটা বেবায় সেখানে একটি সুযোগ আছে ক্যাশে আরম্ভ হওয়ার পূর্বেই সেখানে পরবেশ করা হয়। এটা সনাক্ত করতে, উপেরাক্ত বাস্তুবায়নে, একটি লক করা অবজেক্ট নিশিচত করে যে অ্যাপ ডিস্ক ক্যাশে শেখেকে পড়না যতক্ষন না ক্যাশে আরম্ভ করা হয়।

যখন মেমরী ক্যাশে ইউআই থ্রেডের মধ্যে চেক করা হয়, ডিস্ক ক্যাশে ব্যাকগ্ৰাউন্ডে থ্রেডে চেক করা হয়। ডিস্ক অপারেশন ইউআই থ্রেডে স্যফিটিত হওয়া উচিন নয়। যখন ইমেজ পরেসস করা সম্পূর্ণ হয়, তখন চুড়ান্ত বিটম্যাপ মেমরী এবং ডিস্ক ক্যাশে দুইটাতেই ভবিষ্যত ব্যবহারের জন্য যুক্ত হয়।

কনফিগারেশন পরিবর্তন ব্যবস্থাপনা

রানটাইম কনফিগারেশন পরিবর্তন, যেমন একটি টিস্করন ওরিয়েন্টেশন পরিবর্তন, অ্যান্ডরয়েডে কনতুন কনফিগারেশন দিয়ে একটি চলমান একটি টিভিটিকে ধব্বাসবারিস্ টাটর্ করায় (এই আচরন সম্পর্কে আরও জানতে Handling Runtime Changes: <http://developer.android.com/guide/topics/resources/runtime-changes.html> দেখুন)।

আপনি পুনরায় আপনার সকল ইমেজ পর্সেস হওয়া ক পরিহার করতে চাইতে পারেন যাতে ইউজারের একটি সবচ্ছন্দ এবং গতিময় এক্সপেরিয়েন্স থাকে যখন একটি কনফিগারেশন ঘটে থাকে।

সীমিত বসত, আপনার বিটময়্যাপের একটি সুনন্দর মেমরী কয়্যেশ আছে যা আপনি Use a Memory Cache সেকশনে তৈরী করেছেন। এই কয়্যেশ একটি Fragment যা setRetainInstance(true) কল করার মাধ্যমে স্মরক্কষত তা ব্যবহার করে একটি নতুন একটি টিভিটিকে পাস হতে পারে। একটি টিভিটি পুননুমর্ত হওয়ার পরে, এই রক্কষত Fragment পুনরায় অয়্যাটচ করা হয় এবং আপনি বিদয়মান কয়্যেশ অবজেক্ট পর্তে বশগময় তা পাবেন, ইমেজকে ImageView অবজেক্টের মধ্যে দরুত স্মর্গর্হিত এবং রি-পপুলেট হতে অনুমোদন করে।

এখানে একটি Fragment ব্যবহার করে কনফিগারেশন পরিবর্তন জুরে একটি LruCache অবজেক্ট ধরে রাখার উদাহরণ আছে:

```
private LruCache<String, Bitmap> mMemoryCache;

@Override
protected void onCreate(Bundle savedInstanceState) {
    ...
    RetainFragment retainFragment =
        RetainFragment.findOrCreateRetainFragment(getFragmentManager());
    mMemoryCache = retainFragment.mRetainedCache;
    if (mMemoryCache == null) {
        mMemoryCache = new LruCache<String, Bitmap>(cacheSize) {
            ... // Initialize cache here as usual
        };
        retainFragment.mRetainedCache = mMemoryCache;
    }
    ...
}

class RetainFragment extends Fragment {
    private static final String TAG = "RetainFragment";
    public LruCache<String, Bitmap> mRetainedCache;

    public RetainFragment() {}

    public static RetainFragment findOrCreateRetainFragment(FragmentManager fm) {
        RetainFragment fragment = (RetainFragment) fm.findFragmentByTag(TAG);
        if (fragment == null) {
            fragment = new RetainFragment();
            fm.beginTransaction().add(fragment, TAG).commit();
        }
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setRetainInstance(true);
    }
}
```

এটা পরীক্কষা করতে, Fragment ধরে রেখে এবং ধরনা রেখে একটি ডিভাইন রেটে টট করার চেষ্টা করতে পারেন। আপনার একটি লক্কষয় করা উচিত, দেরী না করা যেহেতু ইমেজ মেমরী থেকে পর্তায় তা ক্কষনিকভাবে একটি টিভিটিকে অধুয়র্ষিত করে যখন আপনি কয়্যেশ ধারন করবেন। কোন ইমেজ মেমরীতে থুজে পাওয়া না গেলে আশাকরা যাবে যে এটা ডিস্ক কয়্যেশে আছে, যদি না

থাকে তার াযথ ার াঁতি পর্েসে সহয়েছে।

বিটম্যাপ মেমরী বয়বস্থাপনা

(<http://developer.android.com/training/displaying-bitmaps/manage-memory.html>)

Caching Bitmaps এ যে আলাচনা করা হয়েছে তা ছাড়াও, গারবেজ কালেকশন এবং বিটম্যাপ রিইউজ সঞ্চারন করতে আপনিনি নিদির্ষ্ টিভােবিকিছু করতে পারবেন। সুপারিশকৃত কৌশল নিভর্র করে আপনিনি অ্যান্ডরয়েডেরিক ধরনের সঙ্করনকে লক্ষ্য হিসাবে রেখেছেন। ইরঃসধৃঋঃহস্যাম্পল অ্যাপ এই কল্লােসর অন্ততভূরংকৃত, আপনােকে দেখায় অ্যান্ডরয়েডেরি বিভিন্ন সঙ্করন জুরে দক্ষতার সাথে কাজ করতে কীভাবে আপনার অ্যাপ কে ডিজাইন করতে হয়।

এই অনুশীলনের জন্য ধাপগুলো সেট করতে, এখানে আছে কীভাবে অ্যান্ডরয়েডেরি বিটম্যাপ মেমরীর বয়বস্থাপনা গড়ে ওঠে:

- অ্যান্ডরয়েড ২.২ (এপিআই লেভেল ৮) এবং এর চেয়ে নীচের সংস্করণে, যখন গারবেজ কালেকশন ঘটে, আপনার অ্যাপের থ্রেড থেমে যায়। এটা একটি টিবিলামব্ব্যটায় যা পারফরমেন্স এর অবনতি ঘটায়। অ্যান্ডরয়েড ২.৩ সমবতরী গারবেজ কালেকশন যুক্ত করে, যেটাে বাঝায় যে, একটি বিটম্যাপ আর রেফারেন্স না হওয়ার পরে মেমরী খুব শিঘ্রই পুনরুদ্ধার হবে।
- অ্যান্ডরয়েড ২.২.৩ (এপিআই লেভেল ১০) এবং এর চেয়ে নীচের সংস্করণে, একটি বিটম্যাপের জন্য বয়বিক্িপিকে সল ডাটা স্থানীয় মেমরীতে সেটার হয়। এটা নিজেই বিটম্যাপ থেকে পৃথক, যেটা ডালিভিকি হেপে সেটার হয়। নিটিভ মেমরীর পিকে সল ডাটা একটি অনুষ্টেয় উপায়ে রিলিজ হয় না, একটি অ্যপিরিকশনকে ক্ষুদ্রভাবে এর মেমরী সীমা অতিকর্মে এবং ধবংস হওয়ার সম্ভাব্য কারণ ঘটাবে। অ্যান্ডরয়েড ৩.০ (এপিআই লেভেল ১১) এর মতো, পিকে সল ডাটা স্যুদ্র বিটম্যাপের সাথে ডালিভিকি হেপে সেটার হয়।

নিচের সেকশন আলাচনা করে বিভিন্ন অ্যান্ডরয়েড সংস্করণে জন্য কীভাবে বিটম্যাপ মেমরী বয়বস্থাপনা অপটিমাইজ করতে হয়।

অ্যান্ডরয়েড ২.২.৩ এবং এরচেয়ে নীচে চরসংস্করণে মেমরী ব্যবস্থাপনা

অ্যান্ডরয়েড ২.২.৩ (এপিআই লেভেল ১০) এবং এরচেয়ে নীচে চরসংস্করণে, `recycle()` ব্যবহার সুপারিশ করা হয়। আপনি যদি আপনার অ্যাপে বড় আকারের বিটম্যাপ ডাটাপর্দাশর্ন করতে চান, আপনার `OutOfMemoryError` এর সম্ভাব্য রান করার সম্ভাবনা। `recycle()` মথড একটি অ্যাপকে যত দরুত সম্ভব মেমরী পুনরুদ্ধার করতে দেয়।

সতর্কতা: আপনার উচিত `recycle()` ব্যবহার করা শুধুমাত্র তখন যখন আপনি নিশ্চিত হবেন যে বিটম্যাপ আর ব্যবহার হচ্ছে না। আপনি যদি `recycle()` কল করেন এবং পরবর্তীতে বিটম্যাপ ডর্ করতে চেষ্টা করেন, আপনি এর পাবেন:

"Canvas: trying to use a recycled bitmap".

নিম্নোক্ত কোডের খন্ডচিত্রটি `recycle()` কলের একটি উদাহরণ দেয়। এটা একটি বিটম্যাপ বর্তমান পরদাশর্ত হচ্ছে না কিস্যে তে আছে তা খুঁজে বের করতে রেফারেন্স কাউন্ট (ভেরিয়েবল `mDisplayRefCount` এবং `mCacheRefCount` এর সম্ভাব্য) ব্যবহার করে। কোড বিটম্যাপকে রিসাইকেল করে যখন এই অবস্থার সম্মুখীন হয়:

- রেফারেন্স কাউন্ট `mDisplayRefCount` এবং `mCacheRefCount` উভয়ের জন্য ০ হয়
- বিটম্যাপ `null` নয় এবং এটা এখনও রিসাইকেল হয় নাই

```
private int mCacheRefCount = 0; private int mDisplayRefCount = 0; ... // Notify the drawable that the displayed state
has changed. // Keep a count to determine when the drawable is no longer displayed. public void
setIsDisplayed(boolean isDisplayed) {
```

```
    synchronized (this) {
        if (isDisplayed) {
            mDisplayRefCount++;
            mHasBeenDisplayed = true;
        } else {
            mDisplayRefCount--;
        }
    }
    // Check to see if recycle() can be called.
    checkState();
```

```
}
```

```
// Notify the drawable that the cache state has changed. // Keep a count to determine when the drawable is no longer
being cached. public void setIsCached(boolean isCached) {
```

```
    synchronized (this) {
        if (isCached) {
            mCacheRefCount++;
        } else {
            mCacheRefCount--;
        }
    }
    // Check to see if recycle() can be called.
    checkState();
```

```
}
```

```
private synchronized void checkState() {
```

```
    // If the drawable cache and display ref counts = 0, and this drawable  
    // has been displayed, then recycle.  
    if (mCacheRefCount <= 0 && mDisplayRefCount <= 0 && mHasBeenDisplayed  
        && isValidBitmap()) {  
        getBitmap().recycle();  
    }
```

```
}
```

```
private synchronized boolean isValidBitmap() {
```

```
    Bitmap bitmap = getBitmap();  
    return bitmap != null && !bitmap.isRecycled();
```

```
}
```

অ্যান্ডরয়েড ৩.০ এবং এরচেয়ে উপরের সংস্করণে মেমরী ব্যবস্থাপনা

অ্যান্ডরয়েড ৩.০ (এপিআই লেভেল ১১) `BitmapFactory.Options.inBitmap` ফিল্ড চালু করে। যদি এই অপশন সেট করা হয়, ডিকোড মেথড যা `Options` অবজেক্ট নেয় তা একটি টিবিটম্যান বিটম্যাট্রির ইউজ করার উদ্দেশ্যে নয় যখন কন্টেন্ট লোড করা হয়। এটার মানে হচ্ছে টিবিটম্যাটে পরমেমরী পুনর্ব্যবহার হচ্ছে, উন্নত পারফরমেন্স পরিনত হতে এবং মেমরী এলোকেশন এবং ডি-এলোকেশন উভয় বাদ দিতে। কিন্ত সেখানে কিছু বাধ্যবাধকতা আছে কীভাবে `inBitmap` ব্যবহার হতে পারে। বিবেশত, অ্যান্ডরয়েড ৪.৪ (এপিআই লেভেল ১৯) এর পূর্ববর্তী শৃঙ্খমাতর্ক সমান সাইজে রিবিটম্যাট্রিপকে সাপোর্ট করা হতে। বিসত্ত্ব বিবরনের জন্য `inBitmap` ডকুমেন্টেশন দেখুন।

পরবর্তী ব্যবহারের জন্য একটি বিটম্যাপ সেভ করা

নিম্নে একটি কৃত্রিম উদাহরণ দেওয়া হয়েছে যা সম্ভাব্য পল অর্থে পরবর্তীতে সম্ভাব্য ব্যবহারের জন্য কীভাবে একটি বিটম্যাপ সেভ করা হয়। যখন একটি বিটম্যাপ অ্যান্ডারৱেড ৩.০ বা এর চেয়ে উপরের লেভেলের রান করে এবং একটি বিটম্যাপ LruCache থেকে উদ্ধৃত হয়, বিটম্যাপের পরপরিত একটি সফটরেফারেন্স একটি HashSet এর মধ্যে স্থাপিত হয়, inBitmap সহকারে পরবর্তীতে সম্ভাব্য ব্যবহারের জন্য:

```
Set<SoftReference<Bitmap>> mReusableBitmaps;
private LruCache<String, BitmapDrawable> mMemoryCache;

// If you're running on Honeycomb or newer, create a
// synchronized HashSet of references to reusable bitmaps.
if (Utils.hasHoneycomb()) {
    mReusableBitmaps =
        Collections.synchronizedSet(new HashSet<SoftReference<Bitmap>>());
}

mMemoryCache = new LruCache<String, BitmapDrawable>(mCacheParams.memCacheSize) {

    // Notify the removed entry that is no longer being cached.
    @Override
    protected void entryRemoved(boolean evicted, String key,
        BitmapDrawable oldValue, BitmapDrawable newValue) {
        if (RecyclingBitmapDrawable.class.isInstance(oldValue)) {
            // The removed entry is a recycling drawable, so notify it
            // that it has been removed from the memory cache.
            ((RecyclingBitmapDrawable) oldValue).setIsCached(false);
        } else {
            // The removed entry is a standard BitmapDrawable.
            if (Utils.hasHoneycomb()) {
                // We're running on Honeycomb or later, so add the bitmap
                // to a SoftReference set for possible use with inBitmap later.
                mReusableBitmaps.add
                    (new SoftReference<Bitmap>(oldValue.getBitmap()));
            }
        }
    }
};
....
}
```

একটি বিবদয্মান বিটময্াপ বয্বহার

রান করা অয্াপের মধ্যে, ডিকোডার মেথডে দেখার জন্য চেক করে যে যদি সেখানে কোন বিবদয্মান বিটময্াপ থাকে যা তার ব্যবহার করতে পারে।
উদাহরণস্বরূপ:

```
public static Bitmap decodeSampledBitmapFromFile(String filename,
    int reqWidth, int reqHeight, ImageCache cache) {

    final BitmapFactory.Options options = new BitmapFactory.Options();
    ...
    BitmapFactory.decodeFile(filename, options);
    ...

    // If we're running on Honeycomb or newer, try to use inBitmap.
    if (Utils.hasHoneycomb()) {
        addInBitmapOptions(options, cache);
    }
    ...
    return BitmapFactory.decodeFile(filename, options);
}
```

পরবর্তী খন্ডাংশটি দেখায় addInBitmapOptions() মেথড যাকে উপরের খন্ডাংশে ত
ব্যবহার করা হয়েছে। এটা inBitmap এর ভ্যালু হিসাবে সেট করতে একটি
বিবদয্মান বিটময্াপ অনুসন্ধান করে। উল্লেখ্য এই মেথড শুধুমাত্র
inBitmap এর জন্য ভ্যালু সেট করে যদি এটা একটি উপযুক্ত ময্াচতুর্জ
পায় (আপনার কেডের কখনই অনুমান করা উচিত নয় যে একটি ময্াচতুর্জ
পাওয়া যাবে): private static void addInBitmapOptions(BitmapFactory.Options options, ImageCache cache) { //
inBitmap only works with mutable bitmaps, so force the decoder to // return mutable bitmaps. options.inMutable = true;

```

if (cache != null) {
    // Try to find a bitmap to use for inBitmap.
    Bitmap inBitmap = cache.getBitmapFromReusableSet(options);

    if (inBitmap != null) {
        // If a suitable bitmap has been found, set it as the value of
        // inBitmap.
        options.inBitmap = inBitmap;
    }
}

// This method iterates through the reusable bitmaps, looking for one
// to use for inBitmap:
protected Bitmap getBitmapFromReusableSet(BitmapFactory.Options options) {
    Bitmap bitmap = null;

    if (mReusableBitmaps != null && !mReusableBitmaps.isEmpty()) {
        synchronized (mReusableBitmaps) {
            final Iterator<SoftReference<Bitmap>> iterator
                = mReusableBitmaps.iterator();
            Bitmap item;

            while (iterator.hasNext()) {
                item = iterator.next().get();

                if (null != item && item.isMutable()) {
                    // Check to see if the item can be used for inBitmap.
                    if (canUseForInBitmap(item, options)) {
                        bitmap = item;

                        // Remove from reusable set so it can't be used again.
                        iterator.remove();
                        break;
                    }
                } else {
                    // Remove from the set if the reference has been cleared.
                    iterator.remove();
                }
            }
        }
    }
    return bitmap;
}

```

চুড়ান্ তভাবে, এই মেথড নিখর্ ারণ করে একটি পর্ াথর্ াী বিটময্ াপ inBitmap এর জনয্ বয্ বহর্ ার কর্ াহেব এমন সাইজ কর্ াইতে টিরয় াকে সন্ তুষ্ ট করে:


```

static boolean canUseForInBitmap(
    Bitmap candidate, BitmapFactory.Options targetOptions) {

    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT) {
        // From Android 4.4 (KitKat) onward we can re-use if the byte size of
        // the new bitmap is smaller than the reusable bitmap candidate
        // allocation byte count.
        int width = targetOptions.outWidth / targetOptions.inSampleSize;
        int height = targetOptions.outHeight / targetOptions.inSampleSize;
        int byteCount = width * height * getBytesPerPixel(candidate.getConfig());
        return byteCount <= candidate.getAllocationByteCount();
    }

    // On earlier versions, the dimensions must match exactly and the inSampleSize must be 1
    return candidate.getWidth() == targetOptions.outWidth
        && candidate.getHeight() == targetOptions.outHeight
        && targetOptions.inSampleSize == 1;
}

/**
 * A helper function to return the byte usage per pixel of a bitmap based on its configuration.
 */
static int getBytesPerPixel(Config config) {
    if (config == Config.ARGB_8888) {
        return 4;
    } else if (config == Config.RGB_565) {
        return 2;
    } else if (config == Config.ARGB_4444) {
        return 2;
    } else if (config == Config.ALPHA_8) {
        return 1;
    }
    return 1;
}

```

আপনার ইউআই একটি টিবিটময্যাপ পর্দাশর্ন

(<http://developer.android.com/training/displaying-bitmaps/display-bitmap.html>)

এই অনুশীলনী পূর্ববর্তরী অনুশীলনীর সব কিছু এক সাথে নিয়ে আসে, আপনাকে দেখায় একটি ব্যব্যাকগর্াউন্ড থের্ড এব্যবিটময্যাপ কয্যশ ব্যবহার করে ViewPager এব্য GridView কয্যপােনেন্টের মধ্যে কীভাবে মাল্টিপল টিবিটময্যাপ লোড করতে হয়, যখন কনক্রেটিন্স এব্য কনফিগারেশন পরিবর্তন নিয়ে কাজ করতে হয়।

ভিউ পেজ বাস্তুত্বায়নে বিটম্যাপ লেআউড করা

একটি ইমেজ গ্যালারির বিস্তুতির ভিউতে গট করতে swipe view pattern একটি চমক করার উপায়। একটি PagerAdapter দ্বারা সমর্থিত একটি ViewPager ব্যবহার আপনি এই প্যাটার্ন বাস্তুত্বায়ন করতে পারেন। কিন্ত এর চেয়ে বেশী উপযুক্ত ব্যবহারিক ক্ষেত্রে অ্যাডাপ্টার হচ্ছে সাবক্লাস FragmentStatePagerAdapter যা ViewPager এর মধ্যে Fragments এর সেটটি সর্বোচ্চ কার্যক্ষমভাবে ধরবে এবং সেভ করে যেভাবে তার অফ-স্ক্রিন দৃশ্যমান করে, মেমরী ব্যবহার নীচে রাখবে।

নোট: আপনার যদি অল্প সংখ্যক ইমেজ থাকে এবং আপনি যদি আত্মবিশ্বাসি হোন যে সবগুলোই অ্যাপলিকেশনে মেমরী সীমার মধ্যে ফিট করবে, তাহলে একটি নিয়মিত PagerAdapter বা FragmentPagerAdapter ব্যবহার করাটা অধিক যুক্তিতত্ত্বকৃত হবে।

এখানে ImageView চিলেডের সহকারে একটি ViewPager এর বাস্তুত্বায়ন আছে।
মেইন একটি ভিউটি ViewPager এবং অ্যাডাপ্টার ধারণ করে:

```
public class ImageDetailActivity extends FragmentActivity {
    public static final String EXTRA_IMAGE = "extra_image";

    private ImagePagerAdapter mAdapter;
    private ViewPager mPager;

    // A static dataset to back the ViewPager adapter
    public final static Integer[] imageResIds = new Integer[] {
        R.drawable.sample_image_1, R.drawable.sample_image_2, R.drawable.sample_image_3,
        R.drawable.sample_image_4, R.drawable.sample_image_5, R.drawable.sample_image_6,
        R.drawable.sample_image_7, R.drawable.sample_image_8, R.drawable.sample_image_9};

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.image_detail_pager); // Contains just a ViewPager

        mAdapter = new ImagePagerAdapter(getSupportFragmentManager(), imageResIds.length);
        mPager = (ViewPager) findViewById(R.id.pager);
        mPager.setAdapter(mAdapter);
    }

    public static class ImagePagerAdapter extends FragmentStatePagerAdapter {
        private final int mSize;

        public ImagePagerAdapter(FragmentManager fm, int size) {
            super(fm);
            mSize = size;
        }

        @Override
        public int getCount() {
            return mSize;
        }

        @Override
        public Fragment getItem(int position) {
            return ImageDetailFragment.newInstance(position);
        }
    }
}
```

এখানে বিস্তুতির Fragment এর একটি বাস্তুত্বায়ন আছে যা imageView চিলেডের
ধারণ করে। এটামনে হতে পারে একটি পুরাপুরি যুক্তিসংগত
অ্যাডাপ্টার, কিন্ত আপনি কি এই বাস্তুত্বায়নের উদ্ভব কল্পনা
পাচ্ছেন? এটা কীভাবে উন্নত হবে?

```

public class ImageDetailFragment extends Fragment {
    private static final String IMAGE_DATA_EXTRA = "resId";
    private int mImageNum;
    private ImageView mImageView;

    static ImageDetailFragment newInstance(int imageNum) {
        final ImageDetailFragment f = new ImageDetailFragment();
        final Bundle args = new Bundle();
        args.putInt(IMAGE_DATA_EXTRA, imageNum);
        f.setArguments(args);
        return f;
    }

    // Empty constructor, required as per Fragment docs
    public ImageDetailFragment() {}

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        mImageNum = getArguments() != null ? getArguments().getInt(IMAGE_DATA_EXTRA) : -1;
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        // image_detail_fragment.xml contains just an ImageView
        final View v = inflater.inflate(R.layout.image_detail_fragment, container, false);
        mImageView = (ImageView) v.findViewById(R.id.imageView);
        return v;
    }

    @Override
    public void onActivityCreated(Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        final int resId = ImageDetailActivity.imageResIds[mImageNum];
        mImageView.setImageResource(resId); // Load image into ImageView
    }
}

```

আশা করা যায় যে আপনি এই ইস্যুয়টি লক্ষ্য করেছেন: ইমেজ ইউআই থের্‌ডে রিসোর্সিং থেকে পড়া হচ্ছে, যা একটি অ্যাপলিকেশনকে ঝুলে থাকা বা জোরপূর্বক বন্ধ হয়ে যাওয়ার দিকে নিয়ে যায়। Processing Bitmaps Off the UI Thread অনুলীলনীতে যেভাবে আলোচনা করা হয়েছে সেভাবে একটি AsyncTask ব্যবহার, একটি ব্যব্যাকগর রাউন্ড থের্‌ডে ইমেজ লোড এবং পরে সেস করতে করোনাটো সাজাসাপটো:

```

public class ImageDetailActivity extends FragmentActivity {
    ...

    public void loadBitmap(int resId, ImageView imageView) {
        mImageView.setImageResource(R.drawable.image_placeholder);
        BitmapWorkerTask task = new BitmapWorkerTask(mImageView);
        task.execute(resId);
    }

    ... // include BitmapWorkerTask class
}

public class ImageDetailFragment extends Fragment {
    ...

    @Override
    public void onActivityCreated(Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        if (ImageDetailActivity.class.isInstance(getActivity())) {
            final int resId = ImageDetailActivity.imageResIds[mImageNum];
            // Call out to ImageDetailActivity to load the bitmap in a background thread
            ((ImageDetailActivity) getActivity()).loadBitmap(resId, mImageView);
        }
    }
}

```

যে কোন অতিরিক্ত পরেসে (যেমন, নেটওয়ার্ক থেকে ইমেজ রিসাইজ করা

বাঁনিযে আসা) মেইন ইউআই এর রেসপনসিভনেস কে ক্যানভাবে পর্তা বিতনা করইে BitmapWorkerTask এর মধ্যে সঁযটি ত করেত পারে। যদি বয়াকগর্াউন্ড থের্ড সরাসরি ডিস্ক থেকে একটি ইমেজ লোড করার চাইতেও বেশী কঁছুরে, এটা একটি মেমরী এবঁ/বাঁ ডিস্ক কয়ঁাশে য়ুক্ত করে লাভবান হতে পারে, যেভাবে Caching Bitmaps অনুশীলনীতে আলোচনা করা হয়েছে। এখানে একটি মেমরী কয়ঁাশের অতিরিক্ত পরিবর্তন দেয়া আছে:

```
public class ImageDetailActivity extends FragmentActivity {
    ...
    private LruCache<String, Bitmap> mMemoryCache;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        ...
        // initialize LruCache as per Use a Memory Cache section
    }

    public void loadBitmap(int resId, ImageView imageView) {
        final String imageKey = String.valueOf(resId);

        final Bitmap bitmap = mMemoryCache.get(imageKey);
        if (bitmap != null) {
            mImageView.setImageBitmap(bitmap);
        } else {
            mImageView.setImageResource(R.drawable.image_placeholder);
            BitmapWorkerTask task = new BitmapWorkerTask(mImageView);
            task.execute(resId);
        }
    }

    ... // include updated BitmapWorkerTask from Use a Memory Cache section
}
```

সকল অঁশ একসাথে করান য়ুনতম লোডিঁল য়াে টিনিঁস (ক্যান কঁছুর পর্তেসস করার সময়কাল) এবঁ আপনার ইমেজে যতটুকু দরকার সেইমতে কম বাঁ বেশী বয়াকগর্াউন্ড পর্তেসসিঁ সঁ করার সামথর্ সহ আপনাে একটি রেসপনসিভ ViewPager বাঁ স্তবায়ন পর্তদান করবে।

একটি গিগার্ডিভিউ বাস্‌তবায়নে বিটম্যাপ লোড করা

grid list building block ইমেজ ডাটাসেটে দেখানোর জন্য উপকারী এবং একটি GridView কন্ট্রোল নতুন বয় ব্যবহার করে বাস্‌তবায়িত হতে পারে যার মধ্যে অনেক ইমেজ যেকোন সময় একসাথে অন-স্ক্রিন হতে পারে এবং আরও অনেকগুলো দৃশ্যমান হতে পরস্পর তুলে ধরা দরকার যদি ইউজার আপ বা ডাউন স্ক্রল করে। যখন এই ধরনের নিয়ন্ত্রণ বাস্‌তবায়ন করা হয়, আপনাকে নিশ্চিত করতে হবে যে ইউআই ফ্লুইড ধরে রাখছে, যেমন বয় ব্যবহার নিয়ন্ত্রণ রাখে এবং কনকোর্ডিন্স সঠিকভাবে বয় বাস্‌থপান করা হয় (GridView চিলড্রেন ভিউরিসাইকেল করার কারণে)।

শুরুতে, ImageView চিলড্রেন সহকারে একটি আদর্শ GridView বাস্‌তবায়ন একটি Fragment এর মধ্যে স্থাপন করে। পূরণায়, এটাকে মনে হতে পারে একটি সম্পূর্ণ রেঞ্জের যৌক্তিক অর্থে পেরোচ, কিন্ত এটাকে কিকভাবে লাগতে পারে?

```

public class ImageGridFragment extends Fragment implements AdapterView.OnItemClickListener {
    private ImageAdapter mAdapter;

    // A static dataset to back the GridView adapter
    public final static Integer[] imageResIds = new Integer[] {
        R.drawable.sample_image_1, R.drawable.sample_image_2, R.drawable.sample_image_3,
        R.drawable.sample_image_4, R.drawable.sample_image_5, R.drawable.sample_image_6,
        R.drawable.sample_image_7, R.drawable.sample_image_8, R.drawable.sample_image_9};

    // Empty constructor as per Fragment docs
    public ImageGridFragment() {}

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        mAdapter = new ImageAdapter(getActivity());
    }

    @Override
    public View onCreateView(
        LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        final View v = inflater.inflate(R.layout.image_grid_fragment, container, false);
        final GridView mGridView = (GridView) v.findViewById(R.id.gridView);
        mGridView.setAdapter(mAdapter);
        mGridView.setOnItemClickListener(this);
        return v;
    }

    @Override
    public void onItemClick(AdapterView<?> parent, View v, int position, long id) {
        final Intent i = new Intent(getActivity(), ImageDetailActivity.class);
        i.putExtra(ImageDetailActivity.EXTRA_IMAGE, position);
        startActivity(i);
    }

    private class ImageAdapter extends BaseAdapter {
        private final Context mContext;

        public ImageAdapter(Context context) {
            super();
            mContext = context;
        }

        @Override
        public int getCount() {
            return imageResIds.length;
        }

        @Override
        public Object getItem(int position) {
            return imageResIds[position];
        }

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

        @Override
        public View getView(int position, View convertView, ViewGroup container) {
            ImageView imageView;
            if (convertView == null) { // if it's not recycled, initialize some attributes
                imageView = new ImageView(mContext);
                imageView.setScaleType(ImageView.ScaleType.CENTER_CROP);
                imageView.setLayoutParams(new GridView.LayoutParams(
                    LayoutParams.MATCH_PARENT, LayoutParams.MATCH_PARENT));
            } else {
                imageView = (ImageView) convertView;
            }
            imageView.setImageResource(imageResIds[position]); // Load image into ImageView
            return imageView;
        }
    }
}

```

আর একবার, এই বাস্তবায়নের সমস্যা হচ্ছে যে ইমেজ ইউআই থের্ডে সেট হচ্ছে। যখন এটা ছোট, সহজ ইমেজে রজন্য কাজ করে (সিস্টেম রিসোর্স লোড এবং ক্যাশে শেয়ার করার কারণে), যদি দেকান অতিরিক্ত পরিসেস করার

দরকার হয়,আপনার ইউআই একটি হেলে টি ধবংস হয়।

পূর্ববর্তরী সেকশন থেকে একই আিসঙেকর্ ানাস পরেসস এবং কয াশে করার মেথড এখানে বস্ তবায়ন করা যেতে পারে। কিন্তু,আপনার কনকারেনিন্‌স ইস্যু য়ে ত সতকর্ হওয়া পর্‌য়োজন কারন GridView চাইল্ড ডিভিউ রিসাইকেল করে। এটা পরিচালনা করতে,Processing Bitmaps Off the UI Thread অনুশীলনীতে আলোচিত কৌশল ব্যবহার করুন। এখানে আপডেট সমাধান দেয়া হেলা:


```

public class ImageGridFragment extends Fragment implements AdapterView.OnItemClickListener {
    ...

    private class ImageAdapter extends BaseAdapter {
        ...

        @Override
        public View getView(int position, View convertView, ViewGroup container) {
            ...
            loadBitmap(imageResIds[position], imageView)
            return imageView;
        }
    }

    public void loadBitmap(int resId, ImageView imageView) {
        if (cancelPotentialWork(resId, imageView)) {
            final BitmapWorkerTask task = new BitmapWorkerTask(imageView);
            final AsyncDrawable asyncDrawable =
                new AsyncDrawable(getResources(), mPlaceholderBitmap, task);
            imageView.setImageDrawable(asyncDrawable);
            task.execute(resId);
        }
    }

    static class AsyncDrawable extends BitmapDrawable {
        private final WeakReference<BitmapWorkerTask> bitmapWorkerTaskReference;

        public AsyncDrawable(Resources res, Bitmap bitmap,
            BitmapWorkerTask bitmapWorkerTask) {
            super(res, bitmap);
            bitmapWorkerTaskReference =
                new WeakReference<BitmapWorkerTask>(bitmapWorkerTask);
        }

        public BitmapWorkerTask getBitmapWorkerTask() {
            return bitmapWorkerTaskReference.get();
        }
    }

    public static boolean cancelPotentialWork(int data, ImageView imageView) {
        final BitmapWorkerTask bitmapWorkerTask = getBitmapWorkerTask(imageView);

        if (bitmapWorkerTask != null) {
            final int bitmapData = bitmapWorkerTask.data;
            if (bitmapData != data) {
                // Cancel previous task
                bitmapWorkerTask.cancel(true);
            } else {
                // The same work is already in progress
                return false;
            }
        }
        // No task associated with the ImageView, or an existing task was cancelled
        return true;
    }

    private static BitmapWorkerTask getBitmapWorkerTask(ImageView imageView) {
        if (imageView != null) {
            final Drawable drawable = imageView.getDrawable();
            if (drawable instanceof AsyncDrawable) {
                final AsyncDrawable asyncDrawable = (AsyncDrawable) drawable;
                return asyncDrawable.getBitmapWorkerTask();
            }
        }
        return null;
    }

    ... // include updated BitmapWorkerTask class

```

নোট: একই কৈড `ListView` সাথে কাজ করতেও মাঝিনে য়িনতে পারে।

এই বাস্ তবায়ন ফেলিক্ সবিবিলিটির জনয্ অনুমোদন করে যেভাবে ইমেজ পর্েসস এবং লেড হয় ইউআই এর মস্ নতাকৈ বধাগর্স্ থনাকেরই।
 বয্াকগর্ াউন্ড টাস্ ক আপিন নৈটওয়াকর্ থেকৈ ইমেজ লেড করতে
 পারেন বা বড়ি ডিজিটাল কয্ ামরাফেটা এবং ইমেজ দৃশয্ মান তারিসাইজ
 করতে পারেন যেহেতু টাস্ ক পর্েসস করাে শষ করে।

এটার এবং অন্য কনসেপ্টের যা এই অনুশীলনটিতে আলোচনা করা হয়েছে তার
পূর্ণরূপে উদাহরণের জন্য, দয়াকরে অন্য তথ্যের কত সম্ভাব্য
অ্যাপলিকেশন দেখুন।

OpenGL ES দিয়ে গরুাফিক্স পর্দশর্ন

(<http://developer.android.com/training/graphics/opengl/index.html>)

অর্ান্ডর্য়েডফর্মওয়াকর্ আকর্ষর্নীয, ফাংশনাল গর্াফিকযাল
ইউজার ইন্টারফেস তৈরী করার জন্য বেষিকছু আদর্শ টুলস সরবরাহ করে
থাকে। কিন্তু স্কর্নে যা অঙ্কন করা হচ্ছ তার উপর আপনি যদি আরও
নিয়ন্ত্রণ চান, বা তিরম্িতর্ক গর্াফিক্স এর মধ্যে কাজ করতে
উদযাগী হান, আপনার বিবিভিন্ টুলস ব্যবহার করা পর্য়োজন।
অর্ান্ডর্য়েডফর্মওয়াকর্ দব্বারা পর্দত OpenGL ES APIs মূলযবান
অর্াণিন্মেটেড গর্াফিক্স যাশুধুমাত্র আপনার চিন্তাশীলতা দব্বারা
সীমাবদ্ধ তা পর্দশর্ন করার জন্য একসেট টুল পর্স্তাব করে এবং
এছাড়াও অনেক অর্ান্ডর্য়েড ডিভাইস পর্দত একসেলেরশন অব
গর্াফিক্স পর্সেসিং ইউনিট (GPUs) থেকে লাভবান হতে পারেন।

এই ক্লাস OpenGL ব্যবহার করা অর্াপিলিকেশন ডেভেলপ করার মেৌলিক
বিষয়গুলোে আপনাকে দেখাবে, যার মধ্যে রয়েছে সেটআপ, ডর্িয়
অবজেক্ট, ম্িতির্ অন এলিমেন্ট এবং টাচ ইনপুটে রেসপন্স।

এই ক্লাসে সরম্েয উদাহরণ কেড OpenGL ES 2.0 APIs ব্যবহার করে, যা চলতি
অর্ান্ডর্য়েড ডিভাইসে সাথে ব্যবহার করতে সুপারিশকৃত এপিআই
সংস্করণ। OpenGL ES এর সংস্করণ সম্পর্কে আরও তথয জানতে OpenGL
ডেভেলপার গাইড দেখুন।

নোট: সতর্ক থাকবেন, OpenGL ES 1.x API কে OpenGL ES 2.0 methods বলে ভুল করবেন না!
এই দুই টা অদল-বদল করার মেেতানয এবং দুই টাকে একসাথে ব্যবহার করার
ফলে আপনি শুধু হতাশই হবেন এবং কষ্টও বাড়বে।

অনুশীলনীসমূহ

একটি **OpenGL ES** পরিবেশ তৈরী করুন

শিখুন **OpenGL** গ্রাফিক্স অঙ্কন করতে সমর্থ হতে কীভাবে একটি অ্যান্ড্রয়েড অ্যাপলিকেশন সেটআপ করতে হয়

গঠননিধর্যারণ করুন

শিখুন কীভাবে গঠন(শেপ) নিধর্যারণ করতে হয় এবং কেন আপনাকে ফেস এবং উইন্ডিং সম্বন্ধে জানতে হবে

গঠন ডর/অঙ্কন করুন

শিখুন আপনার অ্যাপলিকেশনে কীভাবে গঠন(শেপ) অঙ্কন করতে হয়

পর্জেক্ট এবং কয়্যামেরাপর্যোগ করুন

শিখুন আপনার ডর অন অবজেক্ট একটি নতুন পের্ক্যাপটেপে কীভাবে পর্জেকশন এবং কয়্যামেরা ভিউ ব্যবহার করতে হয়

মেশন যুক্ত করুন

শিখুন **OpenGL** দিয়ে কীভাবে ডর অন অবজেক্ট এর মৌলিক গতিবিধি এবং অ্যানিমেশন করতে হয়

ট্যাচ ইভেন্টের সপনস করা

শিখুন **OpenGL** গ্রাফিক্স সিদেয় কীভাবে মৌলিক যোগাযোগ করতে হয়

একটি OpenGL ES পরিবেশ তৈরী করা

(<http://developer.android.com/training/graphics/opengl/environment.html>)

আপনার অ্যান্ড্রয়েড অ্যাপ্লিকেশনে OpenGL ES দিয়ে গ্রাফিক্স অঙ্কন করার জন্য, আপনাকে অবশ্যই তাঁদের জন্য একটি ভিউ কন্ট্রোলার তৈরী করতে হবে। এটা করার সাজা-সাপটা অনেকগুলো উপায়ে একটি হচ্ছে একটি GLSurfaceView এবং একটি GLSurfaceView.Renderer উভয়ই বাস্‌তবায়ন করা। একটি GLSurfaceView হচ্ছে OpenGL দিয়ে গ্রাফিক্স অঙ্কন করার জন্য একটি ভিউ কন্ট্রোলার, এবং GLSurfaceView.Renderer হচ্ছে এঁ ভিউয়ের মধ্যে যাঁ অঙ্কন করা হচ্ছে তা নিয়ন্ত্রণ করে। এই ক্লাস সম্পর্কে আরও তথ্যের জন্য OpenGL ES (<http://developer.android.com/guide/topics/graphics/opengl.html>) ডেভেলপার গাইড দেখুন।

GLSurfaceView হচ্ছে আপনার অ্যাপ্লিকেশনের মধ্যে OpenGL ES গ্রাফিক্স সংযুক্ত করার একটি উপায়। একটি ফুল স্ক্রিন বা ফুল স্ক্রিনের কাছাকাছি গ্রাফিক্স ভিউ এর জন্য এটা একটা যৌক্তিক পছন্দ। ডেভেলপার যারা তাঁদের লেআউটের একটি টেবু-টু-অ্যাক্সেস OpenGL ES গ্রাফিক্স সংযুক্ত করতে চায় তাঁদের TextureView টি এক নজরে দেখা উচিত। সত্যিকার অর্থে “ডু-ইট-ইয়েয়ারে সল্‌ফ” ডেভেলপার, এটাও সম্ভব যে SurfaceView ব্যবহার করে একটি OpenGL ES ভিউ তৈরী করা কিন্ত এটার একটু বেশী অতিরিক্ত কোড লেখার পরেই এজন্য হয়।

এই অনুশীলনী ব্যবখ্যাকের একটি সরল অ্যাপ্লিকেশন কীভাবে GLSurfaceView এবং GLSurfaceView.Renderer এর একটি নুয়নতম বাস্‌তবায়ন সম্পূর্ণ করতে হয়।

মেনিফেস্ট **OpenGL ES** ব্যবহার ডেকে ফলাফল করার ন

আপনার অ্যাপ্লিকেশনের জন্য OpenGL ES 2.0 API ব্যবহার করতে, আপনাকে অবশ্যই আপনার মেনিফেস্ট টিমে মনাক্ত ডেকে ফলাফল শনয়ু ক্ত করতে হবে:

```
<uses  
feature android:glEsVersion="0x00020000" android:required="true" />
```

যদি আপনার অ্যাপ্লিকেশন টেক্সচার কম্পার্সন ব্যবহার করে, আপনাকে অবশ্যই কৈন কম্পার্সন ফরময্ াট আপনি সাপোর্ট করে বন সেটাও ডেকে ফলাফল করতে হবে যাতে যে ডিভাইস এই সকল ফরময্ াট সাপোর্ট করে না তারা আপনার অ্যাপ্লিকেশন রান করার চেষ্টা করতে না পারে:

```
<supports gl texture android:name="GL_OES_compressed_ETC1_RGB8_texture" />  
<supports gl texture android:name="GL_OES_compressed_paletted_texture" />
```

টেক্সচার কম্পার্সন ফরময্ াট সম্পর্কে আরও তথ্য জানতে, OpenGL (<http://developer.android.com/guide/topics/graphics/opengl.html>) ডেভেলপার গাইডে দেখুন।

OpenGL ES গরুাফিক্স এর জন্য একটি একটিভিটি তৈরী করুন

অ্যান্ড্রয়েড অ্যাপ্লিকেশন যা OpenGL ES ব্যবহার করে তার একটিভিটি আছে, অন্য অ্যাপ্লিকেশনের মতোই যার একটি ইউজার একটিভিটি আছে। অন্য অ্যাপ্লিকেশন থেকে এর প্রধান পাঠ্যকর্মে হচ্ছে আপনার একটিভিটির জন্য লোআউটে আপনিক রেখেছেন। যেখানে অনেক অ্যাপ্লিকেশন আপন TextView, Button এবং ListView ব্যবহার করতে পারেন, OpenGL ES ব্যবহার করা অ্যাপে, আপনি আরও একটি GLSurfaceView যুক্ত করতে পারেন।

নিম্নোক্ত কোড উদাহরণ একটি একটিভিটির নুয়নতম বাস্তবায়ন যা এর প্রাথমিক ভিউ হিসাবে একটি GLSurfaceView ব্যবহার করে দেখায়:

```
public class OpenGLES20Activity extends Activity {  
  
    private GLSurfaceView mGLView;  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        // Create a GLSurfaceView instance and set it  
        // as the ContentView for this Activity.  
        mGLView = new MyGLSurfaceView(this);  
        setContentView(mGLView);  
    }  
}
```

নোট: OpenGL ES 2.0 চায় অ্যান্ড্রয়েড ২.২ (এপিআই লেভেল ৮) বা এর চেয়ে উপরের সংস্করণ, তাই নিশ্চিত করুন যে আপনার অ্যান্ড্রয়েড প্রজেক্ট API বা এর চেয়ে উপরের সংস্করণে ট্যাগেট করে।

একটি GLSurfaceView অবজেক্ট তৈরী করুন

একটি GLSurfaceView হচ্ছে ছবি বৈশিষ্ট্যযুক্ত ভিউ যেখানে আপনি OpenGL ES গ্রাফিক্স অঙ্কন করতে পারেন। এটি নিজেকে তৈরী করে না। অবজেক্টের যথাযথ অঙ্কন GLSurfaceView.Renderer এর মধ্যে নিয়ন্ত্রণ হয়ে থাকে যা আপনি এই ভিউ এস্টেট করেছেন। মূলত, এই অবজেক্টের জন্য কেউ টিউন আপ করা, আপনি একে পরিসরিত করা এডিট করে যেতে পারেন এবং শুধু একটি অপরিমিত GLSurfaceView ইনস্ট্যান্স তৈরী করতে পারেন কিন্ত এটা করবেন না। টাচ ইভেন্ট ক্যাপচার করার জন্য আপনাকে অবশ্যই এই ক্লাস পরিসরিত করার পরে যাজন, যা Responding to Touch Events ক্লাসে আলোচনায় এসেছে।

একটি GLSurfaceView এর জন্য পরে যাজনীয় কেউ হচ্ছে নতুন (মিনিমাল), সুতরাং একটি দ্রুত বাস্তবায়নের জন্য, একটি একটি ভিউ টাচ এটা ব্যবহার করে তার মধ্যে শুধু একটি ইনার ক্লাস তৈরী করা একটা সাধারণ বিষয়:

```
class MyGLSurfaceView extends GLSurfaceView {

    public MyGLSurfaceView(Context context){
        super(context);

        // Set the Renderer for drawing on the GLSurfaceView
        setRenderer(new MyRenderer());
    }
}
```

যখন OpenGL ES 2.0 ব্যবহার করা হয়, আপনাকে অবশ্যই আপনার GLSurfaceView কনস্ট্রাক্টরের আরেকটি কল যুক্ত করতে হবে, এটা উল্লেখ করে যে আপনি ২.০ এপিআই ব্যবহার করতে চান:

```
// Create an OpenGL ES 2.0 context
setEGLContextClientVersion(2);
```

নোট: যদি আপনি OpenGL ES 2.0 API ব্যবহার করতে থাকেন, নিশ্চিত করুন আপনি এটা আপনার অ্যাপলিকেশন মেনিফেস্টে স্ট্যাটাস ডিক্লেয়ার করেছেন। আরও তথ্যের জন্য Declare OpenGL ES Use in the Manifest দেখুন।

আপনার GLSurfaceView ইমপিপ্লেমেন্টেশনে অন্যান্য আরেকটি এঁচ্ছক সংযোজন হচ্ছে GLSurfaceView.RENDERMODE_WHEN_DIRTY সেটিং ব্যবহার করে শুধু ভিউ ড্র করার জন্য রেন্ডার মোড সেট করা যখন আপনার ড্রিং ডাটাবে পরিবর্তন থাকে:

```
// Render the view only when there is a change in the drawing data
setRenderMode(GLSurfaceView.RENDERMODE_WHEN_DIRTY);
```

আপনি যতক্ষণ না requestRender() কল করেন এই সেটিং GLSurfaceView ফোর্সে রিড্র হওয়ার থেকে বিরত রাখবে, যেটা এই সম্বন্ধে পল অ্যাপের জন্য অধিক কায়র করী।

রেন্ডার ক্লাস তৈরী করা

GLSurfaceView.Renderer ক্লাসে সরবাস্ত বায়ন, বা রেন্ডারার, একটি অয়্যাপলিকেশনের মধ্যে যা OpenGL ES ব্যবহার করে যেখানে বিষয়বস্তু চমককার হওয়া শুরু হয়। এই ক্লাস GLSurfaceView এ ডর করা হয় তা নিয়ন্ত্রণ করে যার সাথে এটা সংযুক্ত। একটি রেন্ডারের মধ্যে তিনটা পদ্ধতি আছে যাকে অয়্যান্ডরয়েড সিস্টেম কল করে, একটি GLSurfaceView একী এবং কীভাবে ডর করতে হয় তা খুঁজে বের করার জন্য:

- onSurfaceCreated() – ভিউয়ের ওভারহাউল বহার ডেসবহঃ সেটআপ করতে একবার কল করা হয়
- onDrawFrame() - ভিউয়ের প্রতিটা রিডর এর জন্য কল করা হয়
- onSurfaceChanged() – কল করা হয়, যদি ভিউয়ের জয়্যামিতি পরিবর্তন করে, উদাহরণস্বরূপ যখন ভিউ হাইসেরিস্কর্ন ওরিয়েন্টেশন পরিবর্তন করে

এখানে একটি OpenGL ES রেন্ডারের একটি মেমোরিলক বাস্তু বায়ন আছে, যা GLSurfaceView এর মধ্যে একটি গের্ ব্যবহার করে। উদাহরণস্বরূপ GLSurfaceView এর মধ্যে একটি গের্ ব্যবহার করে।

```
public class MyGLRenderer implements GLSurfaceView.Renderer {  
  
    public void onSurfaceCreated(GL10 unused, EGLConfig config) {  
        // Set the background frame color  
        GLES20.glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  
    }  
  
    public void onDrawFrame(GL10 unused) {  
        // Redraw background color  
        GLES20.glClear(GLES20.GL_COLOR_BUFFER_BIT);  
    }  
  
    public void onSurfaceChanged(GL10 unused, int width, int height) {  
        GLES20.glViewport(0, 0, width, height);  
    }  
}
```

এতে যা আছে তাই সব! উপরের কোড উদাহরণ একটি সহজ অয়্যান্ডরয়েড অয়্যাপলিকেশন তৈরীর, যা OpenGL ব্যবহার করে একটি গের্ স্কর্ন প্রদর্শন করে। যখন এই কোড খুব চমককার কৈনিকিছু না করে, এই ক্লাসগুলো তৈরী করে, OpenGL দিয়ে ডরিয়েস গের্ স্কর্ন এলিমেন্ট শুরু করার জন্য প্রয়োজনীয় ফাউন্ডেশন রাখে।

নোট: আপনি হয়তো অবাক হবেন, এই পদ্ধতিগুলো কেবল একটি GL10 পয়্যারামিটার থাকে, যখন আপনি OpenGL ES 2.0 APIs ব্যবহার করেন। এই পদ্ধতি সিস্টেম চার অয়্যান্ডরয়েড ফর্মওয়ার্কের কোড সিম্পলার ধরনের সাথে 2.0 APIs এর জন্য পুনর্ব্যবহার হয়।

আপনি যদি OpenGL ES APIs এর সাথে পরিচিত থাকেন, আপনার এখন আপনার অয়্যাপলিকেশন OpenGL ES পরিবেশে সেটআপ এবং গের্ স্কর্ন ডর করা শুরু করতে পারার সামর্থ্য থাকার উচিত। কিন্ত আপনি যদি OpenGL শুরুর করার জন্য আর একটু সাহায্য প্রয়োজন হয়, আরও কিছু ইন্টিগ্রেটেড পেরবর্তী অনুষ্টান তৈরী চলে যান।

গঠন/আকৃতি নিধর্যারণ করা

(<http://developer.android.com/training/graphics/opengl/shapes.html>)

একটি OpenGL ES ভিউ এর পেরে ক্রমশ পটের মধ্যে অঙ্কিত হওয়া শেপ/গঠন নিধর্যারণ করতে পারা হচ্ছে ছাড়া আপনার হাই-এন্ড গ্লারফিক্স মাস্টার্স পিস তৈরী করার পথ ধাপ। আপনার গ্লারফিক্স অবজেক্ট নিধর্যারণ করাটাকা কীভাবে OpenGL ES আশাকরে সে সম্পর্কে নাজেন OpenGL ES দিয়ে ডর করাটা একটু জটিল হতে পারে।

এই অনুশীলনী একটি অ্যান্ডরয়েড ডিভাইস সিস্করনের সাথে সম্পর্কিত OpenGL ES সমন্বয় সিস্টেম, একটি শেপ/গঠন, নিধর্যারণের মৌলিক সূত্র, শেপ ফেস, একই সাথে একটি তির্যুজ এবং একটি বর্গক্ষেত্র বিবয়ে বখাখয়াকরে।

একটি তির্ভুজ (ট্রায়াঙ্গেল) নির্ধারণ

OpenGL ES তির্ভুজ তির্ভুজ স্পেসকে অরিয়েন্টেশন বহু বহুরকের ড্রইয়া
অবজেক্ট নির্ধারণ করতে দেয়। সুতরাং একটি তির্ভুজ অঙ্কন করতে
পারার আগে, আপনাকে অবশ্যই এর কোঅর্ডিনেটস ঠিক করে নিতে হবে। OpenGL
ES এর মধ্যে সাধারণভাবে এটাকরা হচ্ছে কোঅর্ডিনেটস এর জন্য
ফ্ল্যাটিং পয়েন্ট নামবহার এর একটি ভার্ভেক্স অর্গানাইজেশন।
সবরোচ্চ কায়রকোঁরিতার জন্য, আপনি এই সকল কোঅরিয়েন্টেশন একটি
ByteBuffer এর মধ্যে লিখেন, যা পরে সিস্টেম হওয়ার জন্য OpenGL ES গ্লোফিক্স
পাইপলাইনে পাস হয়ে যায়।

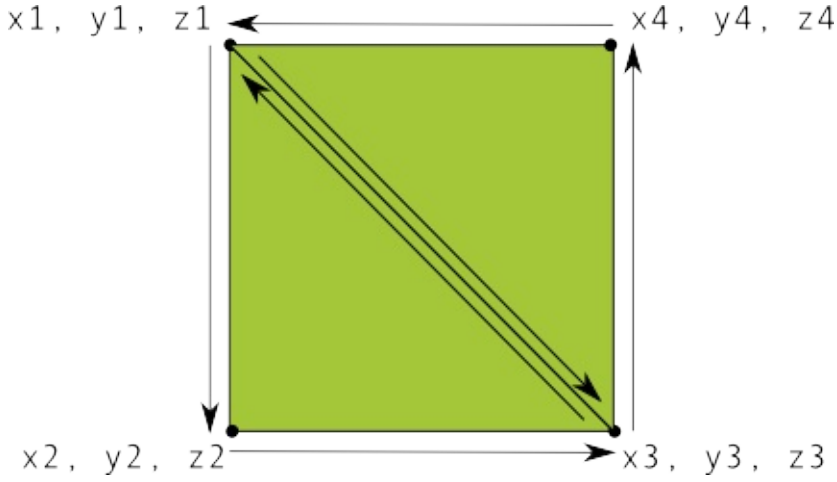
```
public class Triangle {  
  
    private FloatBuffer vertexBuffer;  
  
    // number of coordinates per vertex in this array  
    static final int COORDS_PER_VERTEX = 3;  
    static float triangleCoords[] = { // in counterclockwise order:  
        0.0f, 0.622008459f, 0.0f, // top  
        -0.5f, -0.311004243f, 0.0f, // bottom left  
        0.5f, -0.311004243f, 0.0f // bottom right  
    };  
  
    // Set color with red, green, blue and alpha (opacity) values  
    float color[] = { 0.63671875f, 0.76953125f, 0.22265625f, 1.0f };  
  
    public Triangle() {  
        // initialize vertex byte buffer for shape coordinates  
        ByteBuffer bb = ByteBuffer.allocateDirect(  
            // (number of coordinate values * 4 bytes per float)  
            triangleCoords.length * 4);  
        // use the device hardware's native byte order  
        bb.order(ByteOrder.nativeOrder());  
  
        // create a floating point buffer from the ByteBuffer  
        vertexBuffer = bb.asFloatBuffer();  
        // add the coordinates to the FloatBuffer  
        vertexBuffer.put(triangleCoords);  
        // set the buffer to read the first coordinate  
        vertexBuffer.position(0);  
    }  
}
```

বাইন্ডিং ফল্ট, OpenGL ES মেনের একটি কোঅরিয়েন্টেশন সিস্টেম যেখানে [0,0,0]
(X,Y,Z) GLSurfaceView ফর্মের কেন্দ্র নির্দিষ্ট করে দেয়, [1,1,0] ফর্মের
উপরের ডান কোণ, [-1,-1,0] ফর্মের নীচের বাম কোণ নির্দিষ্ট করে। এই
কোঅরিয়েন্টেশন সিস্টেম সম্পর্কে আরও তথ্য পেতে OpenGL ES ডেভেলপার
গাইড দেখুন।

উল্লেখ্য যে, এই গঠনের/শেপ কোঅরিয়েন্টেশন কাউন্টার-ক্লক-ওয়াইজ
কর্মে নির্ধারণিত হয়। ড্র করার কর্ম থাকা গুরুত্বপূর্ণ কারণ এটা
ঠিক করে শেপের কোন পাশটি ফ্রন্ট ফেস হবে, যেটা সাধারণভাবে আপনি
ড্রইয়াতে চান, এবং বয়স্ক ফেস, যেটা ড্র না করার জন্য বেছে নেন,
OpenGL ES কাল (cull) ফেস বৈশিষ্ট্য বহু বহুরকের। ফেস এবং কালিং সম্পর্কে
আরও তথ্যের জন্য OpenGL ES ডেভেলপার গাইড দেখুন।

একটি বগর্কেষতর্ (স্কয়ার) নিধর্ারণ

OpenGL এ একটি তিতর্ভুজ নিধর্ারণ করা থুবই সহজ, কিন্তু আপনি যদি একতু বেষীজটি লিকছু চান তখন কী হবে? ধরুন বগর্কেষতর্? এটাকরার অনেক উপায় আছে, কিন্তু OpenGL এর মধ্যে ঐশেপ/গঠন অঙ্কন করার একটি সাধর্ারণ পথ হচ্ছে দুইটি তিতর্ভুজ একসাথে অঙ্কন করা:



ফিগার ১. দুইটি তিতর্ভুজ ব্যবহার করে একটি বগর্কেষতর্ অঙ্কন করুন

পুনরায়, উভয় তিতর্ভুজের জন্য একটি কাউন্টার-ক্লক-ওয়াইজ কর্মে মধ্য আপনার শীষর্বিবন্দ্গুলা (vertices) নিধর্ারণ করা যা এইশপকে পর্তিনিধিতব্ করে, ভয়্ালু একটি ByteBuffer এর মধ্যে রাখে। পর্তিট তিতর্ভুজ কতর্ক দুইবার শেয়ার হওয়া দুইটাকে অরিডিনেটসকে নিধর্ারণ করা পরিহার করতে, OpenGL ES গর্াফিক্স পাইপলাইনকে বলার জন্য যে এই শীষর্বিবন্দ্গুলা (vertices) কীভাবে ডর্ করতে হয় তার জন্য ডর্িয়লিস্ট ব্যবহার করুন। এখন এইশেপের জন্য কেডেয়াহেলা:

```

public class Square {

    private FloatBuffer vertexBuffer;
    private ShortBuffer drawListBuffer;

    // number of coordinates per vertex in this array
    static final int COORDS_PER_VERTEX = 3;
    static float squareCoords[] = {
        -0.5f, 0.5f, 0.0f, // top left
        -0.5f, -0.5f, 0.0f, // bottom left
        0.5f, -0.5f, 0.0f, // bottom right
        0.5f, 0.5f, 0.0f }; // top right

    private short drawOrder[] = { 0, 1, 2, 0, 2, 3 }; // order to draw vertices

    public Square() {
        // initialize vertex byte buffer for shape coordinates
        ByteBuffer bb = ByteBuffer.allocateDirect(
            // (# of coordinate values * 4 bytes per float)
            squareCoords.length * 4);
        bb.order(ByteOrder.nativeOrder());
        vertexBuffer = bb.asFloatBuffer();
        vertexBuffer.put(squareCoords);
        vertexBuffer.position(0);

        // initialize byte buffer for the draw list
        ByteBuffer dlb = ByteBuffer.allocateDirect(
            // (# of coordinate values * 2 bytes per short)
            drawOrder.length * 2);
        dlb.order(ByteOrder.nativeOrder());
        drawListBuffer = dlb.asShortBuffer();
        drawListBuffer.put(drawOrder);
        drawListBuffer.position(0);
    }
}

```

এই উদাহরণ আপনাকে OpenGL দিয়ে আরও জটিল শেপ তৈরী করার বিষয়গুলো দেখাবে। সাধারণভাবে, আপনি অবজেক্ট ডরং করতে তির্ভুজের সংগ্রহ ব্যবহার করেন। পরবর্তী অনুশীলনীতে আপনি শিখবেন কিভাবে এই শেপগুলো অঙ্কন করা হবে।

শেপ/আকৃতি অঙ্কন করা

(<http://developer.android.com/training/graphics/opengl/draw.html>)

OpenGL দিয়ে অঙ্কন করার জন্য শেপ নিখরহে রাখার পর, আপনি সম্ভবত তাদের অঙ্কন করতে চাইবেন। OpenGL 2.0 দিয়ে অঙ্কন করাটা আপনি যা চিন্তা করেছেন তার চেয়েও বেশী ক্যাডেন্স, কারণ এপিআই গ্লোবাল ফক্সরেন্ডারিং পাইপলাইনের উপর আরও নিয়ন্ত্রণ প্রদান করে।

এই অনুশীলনী ব্যয়াক্ষর পূর্ববর্তী অনুশীলনীতে নিখরহে রাখার শেপ OpenGL ES 2.0 API ব্যবহার করে কীভাবে ড্র/অঙ্কন করা যায়।

শেপ/আকৃতি আরম্ভ করা

কোন ড্রয়িং/অঙ্কন করার পূর্বে, আপনি যে শেপ ড্র করার পরিকল্পনা করেছেন তা অবশ্যই আরম্ভ করা এবং লোড করতে হবে। যতক্ষণ না আপনার পের্গাগর্গামে বয়বহার করা শেপের কাঠামো (মূল কেরিডনেটস) কাযরুনিবর্গাহে রগতিপথ সময়কালীন পরিবর্তন না হয়, মেমরী এবং পরসেসিস কাযরুকািরতার জনয় আপনার উচিত তােদর আপনার েন্ডারগের onSurfaceCreated() পদ্ধতির মধ্যে আরম্ভ করা

```
public void onSurfaceCreated(GL10 unused, EGLConfig config) {  
    ...  
  
    // initialize a triangle  
    mTriangle = new Triangle();  
    // initialize a square  
    mSquare = new Square();  
}
```

শেপ অঙ্কন/ডর করা

OpenGL ES 2.0 ব্যবহার করে একটি টিনিথর রাঁচিত শেপ অঙ্কন করার জন্য একটি উল্লেখযোগ্য পরিমান কোড পর্যালোচনা হয়, কারণ আপনাকে অবশ্যই গ্লোবাল ফ্রেমওয়ার্কের ডারিং পাইপলাইনে পরচুর তথ্য-উপাত্ত দিতে হবে, বিশেষ করে নীচের বিষয়গুলো টিনিথর রাঁগ করে দিতে হবে:

- **Vertex Shader** – একটি শেপের শীষর বিন্দুগুলো (vertices) এর ডারিং করার জন্য OpenGL ES গ্লোবাল ফ্রেমওয়ার্ক কোড
- **Fragment Shader** – কালার এবং টেক্সচার দিয়ে একটি শেপের ফ্রেমওয়ার্কের ডারিং করার জন্য OpenGL ES গ্লোবাল ফ্রেমওয়ার্ক কোড
- **Program** – একটি OpenGL ES অবজেক্ট যা শেডার ধারণ করে যা আপনি এক বা একাধিক শেপ ডর করার জন্য ব্যবহার করতে চান

আপনার একটি শেপ ডর করার জন্য কমপক্ষে একটি ভার্টেক্স শেডার লাগবে এবং এটি শেপ রঙ করতে একটি ফ্র্যাগমেন্ট শেডার লাগবে। এই শেডগুলো অবশ্যই সংকলিত হতে হবে এবং তারপর একটি OpenGL ES program এ যুক্ত হতে হবে, তারপর এটাকে শেপ ডর করার কাজে ব্যবহার করা হবে। এখন একটি উদাহরণ আছে, কীভাবে একটি টেমপ্লেট শেডার টিনিথর রাঁগ করতে হয় যা আপনি একটি শেপ ডর করার কাজে ব্যবহার করতে পারেন:

```
private final String vertexShaderCode =
    "attribute vec4 vPosition;" +
    "void main() {" +
    "    gl_Position = vPosition;" +
    "}";

private final String fragmentShaderCode =
    "precision mediump float;" +
    "uniform vec4 vColor;" +
    "void main() {" +
    "    gl_FragColor = vColor;" +
    "}";
```

শেডার OpenGL Shading Language (GLSL) কোড ধারণ করে যা OpenGL ES পরিবেশে ব্যবহারের পূর্বের অবশ্যই সংকলিত হতে হবে। এই কোড সংকলন করতে আপনার ফ্রেমওয়ার্ক লাসে একটি ইউটিলিটি পদ্ধতি তৈরি করুন:

```
public static int loadShader(int type, String shaderCode){

    // create a vertex shader type (GL_VERTEX_SHADER)
    // or a fragment shader type (GL_FRAGMENT_SHADER)
    int shader = GLES20.glCreateShader(type);

    // add the source code to the shader and compile it
    GLES20.glShaderSource(shader, shaderCode);
    GLES20.glCompileShader(shader);

    return shader;
}
```

আপনার শেপ ডর করার জন্য, আপনাকে অবশ্যই শেডার কোড সংকলন করতে হবে, তাদের একটি OpenGL ES প্রোগ্রামে যুক্ত করুন এবং তারপর প্রোগ্রামটি লোড করুন। এটা আপনার ডর করা অবজেক্টের কনস্ট্রাক্টর করুন, ফলে এটা একবার ঘটবে।

নোট: সংকলিত OpenGL ES শেডার এবং লোড করার প্রোগ্রামিং সিপিইউ সাইকেল এবং প্রসেসর সময়ের শেডার ব্যবহুল, তাই আপনাকে এটা একবারের বেশী করাটা পরিহার করতে হবে। আপনি যদি রান টাইমে আপনার

শেডারের কন্টেইনারটি সম্বন্ধে না জানেন, আপনার উচিত আপনার ক্যাডট্রেরী
করা যেমনভাবে তারাশুধু ত্রেরী করা জিনিসটা পেয়ে থাকে এবং পরবর্তী
বয়বহারের জন্য জমিয়ে রাখা। public class Triangle() { ...

```
int vertexShader = loadShader(GL_ES20.GL_VERTEX_SHADER, vertexShaderCode);
int fragmentShader = loadShader(GL_ES20.GL_FRAGMENT_SHADER, fragmentShaderCode);

mProgram = GL_ES20.glCreateProgram(); // create empty OpenGL ES Program
GL_ES20.glAttachShader(mProgram, vertexShader); // add the vertex shader to program
GL_ES20.glAttachShader(mProgram, fragmentShader); // add the fragment shader to program
GL_ES20.glLinkProgram(mProgram); // creates OpenGL ES program executables
}
```

এই মুহূর্তে, আপনি যথাযথ কল যুক্ত করতে পারেন। তখনই আপনি
আপনার শেডার করে। OpenGL ES দিয়ে শেডার করা চায়েরেন্ডারিং
পাইপলাইনকে বলতে পান যে আপনি কল করতে চান এবং কীভাবে এটা ডর
করতে চান সেই জন্য আপনি কিছু প্যারামিটার নির্দিষ্ট করুন।
যেহেতু ডরিং অপশন শেডার ভিন্ন ভিন্ন হতে পারে, তাই এটা
একটা ভার্গিচন তাকে আপনার শেডার ক্লাস তে ডিফিনিজেশন ডরিং লজিক
(যুক্তি) ধারণ করে।

শেডার করার জন্য একটি draw() পদ্ধতি ত্রেরী করুন। এই ক্যাডে শেডার
ভার্টেক্স শেডার এবং ফরগেমনট শেডারের অবস্থান (পজিশন) এবং
কালার ভ্যালু সেট করে, এবং তারপর ডর করার কাজ সম্পাদন করে।

```
public void draw() {
    // Add program to OpenGL ES environment
    GL_ES20.glUseProgram(mProgram);

    // get handle to vertex shader's vPosition member
    mPositionHandle = GL_ES20.glGetAttribLocation(mProgram, "vPosition");

    // Enable a handle to the triangle vertices
    GL_ES20.glEnableVertexAttribArray(mPositionHandle);

    // Prepare the triangle coordinate data
    GL_ES20.glVertexAttribPointer(mPositionHandle, COORDS_PER_VERTEX,
                                   GL_ES20.GL_FLOAT, false,
                                   vertexStride, vertexBuffer);

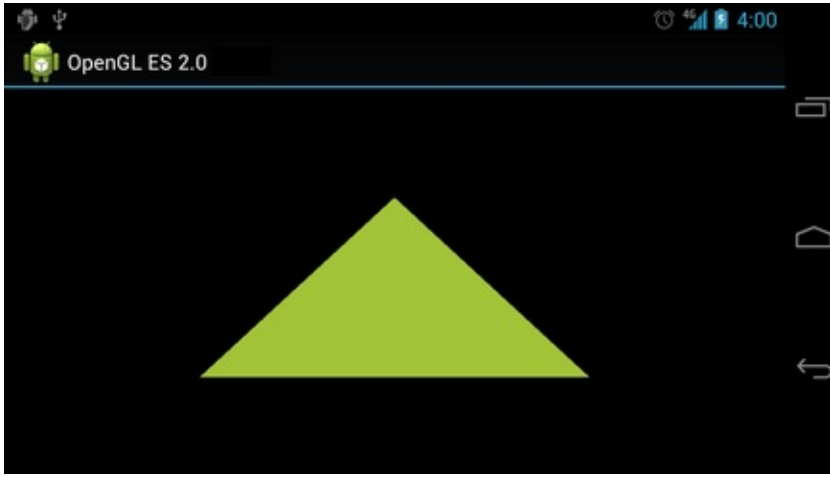
    // get handle to fragment shader's vColor member
    mColorHandle = GL_ES20.glGetUniformLocation(mProgram, "vColor");

    // Set color for drawing the triangle
    GL_ES20.glUniform4fv(mColorHandle, 1, color, 0);

    // Draw the triangle
    GL_ES20.glDrawArrays(GL_ES20.GL_TRIANGLES, 0, vertexCount);

    // Disable vertex array
    GL_ES20.glDisableVertexAttribArray(mPositionHandle);
}
```

যখনই আপনার এই সকল ক্যাড যথাযথ না থাকবে, এই অবজেক্ট অঙ্কন করতে
শুধু আপনার রেন্ডারিংয়ের onDrawFrame() পদ্ধতির মধ্যে থেক পদ্ধতিতে
একটি কল করা পড়ে যাজন। যখনই আপনি অ্যাপলিকে কল করা করবেন, এটা
দেখতে অনেকটা এরকম দেখাবে:



ফিগার ১. একটি পর্জেকশন বাকয্যে মেরা ভিউ ছাড়া অ্যাক্রিক তিতর ভুজ

এই কোড উদাহরনে কিছু সমসয়া আছে। পর্থমটি হচ্ছে, এটা আপনার বন্ধুদেরকে কাছে এটাকে কান পর্ভাব ফেলবেনা। দিব্যতীয়ত, তিতর ভুজটি একটু চাপাচাপি অবস্থায় থাকে এবং আপনি যখন ডিভাইসে রিস্কর্ন ওরিয়েন্টেশন পরিবর্তন করবেন তখন এটা শেপ পরিবর্তন করে। তিতর ভুজটি অসমান (স্কউড) হওয়ার কারণে কিছু ঘটনা যা হচ্ছে অবজেক্টের শীর্ষবিন্দু গুলোর (vertices) স্কর্ন এলাকা যেখানে GLSurfaceView পর্দা শরত হয়ে সেখানকার অনুপাত সঠিকভাবে হয় নাই। পর্বতরী অনুশীলনীতে একটি পর্জেকশন এবং কয্যে মেরা ভিউ ব্যবহার করে আপনি এই সমসয়া ঠিক করতে পারেন।

সবরশেষটা হচ্ছে তিতর ভুজটি, অনড়, ডেটা একটু বিবরিত কর। Adding Motion অনুশীলনীতে আপনি এই শপকে রেটেট করতে পারবেন এবং OpenGL ES গর্ফিক্স পাইপলাইন এর ব্যবহার আরও আকর্ষণীয় করবে।

পর্জেকশন এবং কয়ামেরাভিউ পর্য়োগ

(<http://developer.android.com/training/graphics/opengl/projection.html>)

OpenGL ES পরিবেশে, পর্জেকশন এবং কয়ামেরাভিউ অঙ্কনকৃত অবজেক্ট একটা উপায়ে পর্দাশরন করতে দেয় যা আপনি নিজে চাখে বাস্ তেব যা দেখেন তার খুব কাচাকাছি স্বব্দ শ্রুত পূনর। বাস্ তেবর মতে থাকে দেখার বিষয়টি যটেট অঙ্কিত অবজেক্ট কে কাঅরিডেনেটস এর গাণিতিক রূপান্তরের মাধ্যমে:

- পর্জেকশন- এই ট্রান্সফর্মেশন GLSurfaceView যেখানে তার পর্দাশরত হয় তার দৈর্ঘ্য এবং উচ্চতার উপর ভিত্তি করে অঙ্কিত অবজেক্টের কাঅরিডেনেটস সমন্বয় করে। এই কয়ালকুলেশন ব্যতীত, OpenGL ES দবারা অঙ্কিত অবজেক্ট ভিউ উইন্ডোর অসম অনুপাতের দবারা অসমান হয়। একটি পর্জেকশন ট্রান্সফর্মেশন সাধারণভাবে শুধুমাত্র তখন কয়ালকুলেট হওয়া উচিত যখন OpenGL ভিউ এর অনুপাত আপনার রেন্ডার এর onSurfaceChanged() পদ্ধতির মধ্যে প্রতিতিষ্ঠিত হয় বা পরিবর্তিত হয়। OpenGL ES পর্জেকশন এবং কাঅরিডেনেট ময়্যাপিং সম্বন্ধে আরও তথ্যের জন্য Mapping Coordinates for Drawn Objects দেখুন।
- কয়ামেরাভিউ- এই ট্রান্সফর্মেশন একটি ভাচুর য়াল কয়ামেরা অবস্থানের উপর ভিত্তি করে অঙ্কিত অবজেক্টের কাঅরিডেনেটস সমন্বয় করে। এটা খয়াল করা জরুরী, OpenGL ES একটি আসল কয়ামেরা অবজেক্ট নির্ধারণ করেনা, কিন্তু পরিবর্তিত ইউটিলিটি পদ্ধতি সরবরাহ করে যা অঙ্কিত অবজেক্টের পর্দাশরনের ট্রান্সফর্মেশন ঘটানোর দবারা একটি কয়ামেরা কে অনুকরন করে। একটি কয়ামেরা ভিউ ট্রান্সফর্মেশন শুধু একবার কয়ালকুলেট হয় যখন আপনি আপনার GLSurfaceView প্রতিতিষ্ঠিত করেন, বা ইউজারের একশন এবং আপনার অয়্যাপলিকেশনের ফাংশন এর উপর ভিত্তি করে সিকর্য পরিবর্তন করে।

এই অনুশীলনী আলাচনাকরে কীভাবে একটি পর্জেকশন এবং কয়ামেরাভিউ তৈরী করতে হয় এবং আপনার GLSurfaceView এর মধ্যে শেপ অঙ্কন করতে কীভাবে এটা পর্য়োগ করতে হয়।

একটি পর্জেকশন নিধরার

একটি পর্জেকশন রূপান্তরের জন্য ডাটা আপনার GLSurfaceView.Renderer ক্লাসের onSurfaceChanged() পদ্ধতির মধ্যে ক্যালকুলেট হয়। নিম্নোক্ত উদাহরণ কোড GLSurfaceView এর উচ্চতা এবং দৈর্ঘ্য নেয় এবং Matrix.frustumM() পদ্ধতি ব্যবহার করে একটি পর্জেকশন ট্রান্সফর্মেশন Matrix পুনরুৎপাদন করে এটা ব্যবহার করে:

```
@Override
public void onSurfaceChanged(GL10 unused, int width, int height) {
    GLES20.glViewport(0, 0, width, height);

    float ratio = (float) width / height;

    // this projection matrix is applied to object coordinates
    // in the onDrawFrame() method
    Matrix.frustumM(mProjectionMatrix, 0, -ratio, ratio, -1, 1, 3, 7);
}
```

এই কোড একটি পর্জেকশন মেট্রিক্স mProjectionMatrix পুনরুৎপাদন করে যা আপনি পরবর্তীতে onDrawFrame() পদ্ধতির মধ্যে একটি ক্যামেরা ভিউ রূপান্তরের সাথে কম্বিনেট করতে পারবেন, যা পরবর্তী অধ্যায়ে আলোচনা করা হবে।

নোট: আপনার অ্যাক্টিভ অবেজেক্ট শুধু একটি পর্জেকশন ট্রান্সফর্মেশন পর্য়েোগ করার সাধারণ ফলাফল হচ্ছে একটি থ্রিডি স্পেস। সাধারণভাবে, সিস্টরেনে কোন কিছু পর্দার জন্য আপনাকে অবশ্যই একটি ক্যামেরা ভিউ ট্রান্সফর্মেশন পর্য়েোগ করতে হবে।

একটি ক্যামেরা ভিউ ট্রান্সফর্মেশন

অঙ্কন করার পরিকল্পনার অংশ হিসাবে ক্যামেরা ভিউ ট্রান্সফর্মেশন যুক্ত করার মাধ্যমে আপনার অঙ্কিত অবজেক্টের রূপান্তরিত করার পরিকল্পনা সম্পূর্ণ করুন। নিম্নোক্ত উদাহরণ কোডে ডিফল্ট `Matrix.setLookAtM()` পদ্ধতি ব্যবহার করে ক্যামেরা ভিউ ট্রান্সফর্মেশন ক্যালকুলেট হয় এবং পূর্বের নথায় ক্যালকুলেট হওয়া পর্জেকশন মেট্রিক্স এর সাথে সমিমিলিত হয়। সমিমিলিত ট্রান্সফর্মেশন মেট্রিক্স তখন পের্সপেক্টিভ অঙ্কন করার জন্য।

```
@Override
public void onDrawFrame(GL10 unused) {
    ...
    // Set the camera position (View matrix)
    Matrix.setLookAtM(mViewMatrix, 0, 0, 0, -3, 0f, 0f, 0f, 1.0f, 0.0f);

    // Calculate the projection and view transformation
    Matrix.multiplyMM(mMVPMatrix, 0, mProjectionMatrix, 0, mViewMatrix, 0);

    // Draw shape
    mTriangle.draw(mMVPMatrix);
}
```

পর্জেকশন এবং ক্যাঁমেরা ট্রান্সফর্মেশন পর্য়োগ

পূবর্বতরী অধ্য্যে়ে দেখােনা সিম্মিলিত পর্জেকশন এবং ক্যাঁমেরা
িভিউ ট্রান্সফর্মেশন মেটিটর্ক্ সব্যবহার করার জন্য, সিম্মিলিত
ট্রান্সফর্মেশন মেটিটর্ক্ সগর্হন করতে এবং শেপ এটা পর্য়োগ
করতে আপনার গর্াফিক অব্জেক্টর draw() পদ্ধতি পরিমিত করুন:

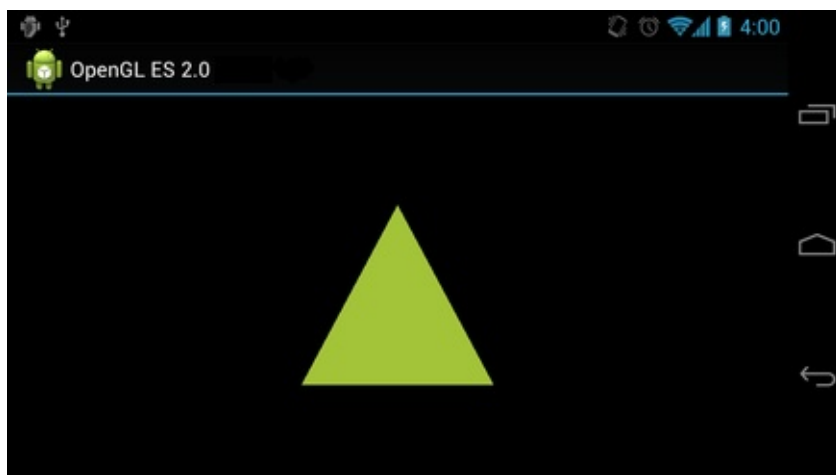
```
public void draw(float[] mvpMatrix) { // pass in the calculated transformation matrix
    ...

    // get handle to shape's transformation matrix
    mMVPMatrixHandle = GLES20.glGetUniformLocation(mProgram, "uMVPMatrix");

    // Pass the projection and view transformation to the shader
    GLES20.glUniformMatrix4fv(mMVPMatrixHandle, 1, false, mvpMatrix, 0);

    // Draw the triangle
    GLES20.glDrawArrays(GLES20.GL_TRIANGLES, 0, vertexCount);
    ...
}
```

যখনই আপনি সঠিকভাবে পর্জেকশন এবং ক্যাঁমেরািভিউ ট্রান্সফর্মেশন
ক্যাঁলকুলেট এবং পর্য়োগ করতে পারবেন, আপনার গর্াফিক অব্জেক্ট
সঠিক অনুপাতে অঙ্কিত হবে এবং দেখতে এরকম হবে:



িফগার ১. একটি পর্জেকশন এবং ক্যাঁমেরািভিউ পর্য়োগ করে তির্ভুজ
অঙ্কন।

এখন আপনার একটি অ্যাপলিকেশন আছে যা আপনার শেপ সঠিক অনুপাতে
পর্দশর্ন হবে, এখন আপনার শেপ গতিশীলতা (মোশন) যুক্ত করার সময়।

গতিশীলতা (মেশন) যুক্ত করা

(<http://developer.android.com/training/graphics/opengl/motion.html>)

স্বাক্ষরিত অবজেক্ট অঙ্কন করা OpenGL এর একটি মৌলিক কাজ, কিন্ত আপনি অন্যান্য অ্যান্ড্রয়েড গারফিক্স ফ্রেমওয়ার্ক ক্লাস দিয়েও এটা করতে পারবেন যার মধ্যে রয়েছে Canvas এবং Drawable অবজেক্ট।

তির্যম্যতির কবান্য সবতন্তর উপায়ে বাধ্যতাইউজার এক্সপেরিয়েন্স তৈরী করতে অঙ্কিত অবজেক্টকে চলন্ত এবং রূপান্তরিত করার জন্য OpenGL ES অতিরিক্ত সামর্থ্য প্রদান করে।

এই অনুশীলনীতে, রেগিষ্ট্রেশন দিয়ে কীভাবে একটি শেপ গতিশীলতা (মেশন) যুক্ত করা যায় এটা শেখার মাধ্যমে OpenGL ES ব্যবহারের কেসের আরেক পাসামনে আগাবেন।

শেপেরাে টট করা

OpenGL ES 2.0 দিয়ে একটি অঙ্কিত অবজেক্টেরাে টট করা তুলনামূলকভাবে সহজ। আপনি অন্যরকমি টটরান্ সফরেশন মেটরক্‌স (একি টেরাে টশন মেটরক্‌স) তৈরী করুন এবং তারপর আপনার পর্জেকশন এবং কয়্ােরাি ভিউ টরান্ সফেরশন মেটরক্‌স এর সাথে এটাে কসিম্ মিলিত করুন:

```
private float[] mRotationMatrix = new float[16];
public void onDrawFrame(GL10 gl) {
    ...
    float[] scratch = new float[16];

    // Create a rotation transformation for the triangle
    long time = SystemClock.uptimeMillis() % 4000L;
    float angle = 0.090f * ((int) time);
    Matrix.setRotateM(mRotationMatrix, 0, angle, 0, 0, -1.0f);

    // Combine the rotation matrix with the projection and camera view
    // Note that the mMVPMatrix factor *must be first* in order
    // for the matrix multiplication product to be correct.
    Matrix.multiplyMM(scratch, 0, mMVPMatrix, 0, mRotationMatrix, 0);

    // Draw triangle
    mTriangle.draw(scratch);
}
```

যদি আপনার তির্ভুজ এই পরিবর্তন করার পরেরাে টট না করে, নিশ্চিত করুন আপনি GLSurfaceView.RENDERMODE_WHEN_DIRTY সেটিং কে মন্ ট আউট করেছেন, যেভাবে পরবর্তী অধ্যয়ে আলোচনা করা হয়েছে।

অবিবর্তনীয় ডায়ালগ বক্স

যদি আপনি এই পয়েন্ট এই ক্লাসের মধ্যে উদাহরণ কোড সম্বন্ধে অনুসরণ করেন, নিশ্চিত করুন আপনি লাইন কন্ট্রোল আউট করেন যা রেন্ডার মেড সেট করে শুধুমাত্র যখন অঙ্কন অপরিষ্কার হয়, অন্যথায় শুধুমাত্র একবারে পেরাটেট বৃদ্ধি করা হয় এবং তারপর GLSurfaceView কন্ট্রোলার থেকে requestRender() পরে একটি কলের জন্য অপেক্ষা করা হয়:

```
public MyGLSurfaceView(Context context) {  
    ...  
    // Render the view only when there is a change in the drawing data.  
    // To allow the triangle to rotate automatically, this line is commented out:  
    //setRenderMode(GLSurfaceView.RENDERMODE_WHEN_DIRTY);  
}
```

যদি নাকোন ইউজারের সাথে যোগাযোগ ছাড়াই আপনার অবজেক্ট পরিবর্তন থাকে, এটি সাধারণত ভালো চিন্তা মানদণ্ড চালু রাখা। এই কোড আনকন্ট্রোল করার জন্য পরস্তুত হোন, কারণ পরবর্তী অনুশীলনী এই কলেকআর একবার পরে আসি উৎসাহিত করবে।

ট্যাচ ইভেন্টের সম্পর্ক

(<http://developer.android.com/training/graphics/opengl/touch.html>)

চিত্রভুক্ত করে দেওয়া একটি পূর্ববর্তী পত্রে পরাগর্শম অনুসারে অবজেক্ট ম্যুয়ভ করা, কিছু মেনে যাগ পাওয়ার জন্য উপকারী, কিন্ত আপন যদি চান আপনার OpenGL ES graphics সাথে ইউজারের পারস্পরিক যোগাযোগ থাকুক তখন কী হবে? আপনার OpenGL ES অ্যাপ্লিকেশন ট্যাচ যোগাযোগ তৈরী করার উপায়, ট্যাচ ইভেন্ট এর জন্য লিসেন করতে `onTouchEvent()` ওভাররাইড করতে `GLSurfaceView` বাস্তবায়ন পরিসারিত করেছে।

এই অনুশীলনী আপনাকে দেখায় ইউজারকে একটি OpenGL ES অবজেক্টকে রেখে দেওয়া করতে দিয়ে ট্যাচ ইভেন্টের জন্য লিসেন করতে হয়।

ট্যাচ লিসেনার সেটআপ

আপনার OpenGL ES অ্যাপ্লিকেশনকে ট্যাচ ইভেন্টের সপন্থে সরার জন্য তৈরী করতে, আপনাকে অবশ্যই আপনার GLSurfaceView ক্লাসে onTouchEvent() পদ্ধতি বাস্তবায়ন করতে হবে। নীচে চরবাস্তবায়ন উদাহরণে দেখায় কীভাবে MotionEvent.ACTION_MOVE ইভেন্ট লিসেন করতে হয় এবং একটি টেপের জন্য রেজেষ্ট্রেশনের একটি এয়াজেক্ট গেলতাদের অনুবাদ করতে হয়।

```
@Override
public boolean onTouchEvent(MotionEvent e) {
    // MotionEvent reports input details from the touch screen
    // and other input controls. In this case, you are only
    // interested in events where the touch position changed.

    float x = e.getX();
    float y = e.getY();

    switch (e.getAction()) {
        case MotionEvent.ACTION_MOVE:

            float dx = x - mPreviousX;
            float dy = y - mPreviousY;

            // reverse direction of rotation above the mid-line
            if (y > getHeight() / 2) {
                dx = dx * -1;
            }

            // reverse direction of rotation to left of the mid-line
            if (x < getWidth() / 2) {
                dy = dy * -1;
            }

            mRenderer.setAngle(
                mRenderer.getAngle() +
                ((dx + dy) * TOUCH_SCALE_FACTOR); // = 180.0f / 320
            requestRender();
        }

        mPreviousX = x;
        mPreviousY = y;
        return true;
    }
}
```

উল্লেখ্য যে রেজেষ্ট্রেশন অ্যাজেক্ট গেলকয়ালকুলেকশন করার পর, এই পদ্ধতি requestRender() কে কল করে রেন্ডারকে বলতে চায় যে এটা ফর্মের রেন্ডার করার সময়। এই উদাহরণে এই অ্যাপেরাচ সবচেয়ে কার্যকরী কারণ ফর্মটি কেমনতুন করে অঙ্কন করার কোন পরয়োজন নেই যদি না সেখানে রেজেষ্ট্রেশন একটি পরিবর্তন হয়। কিন্ত, এটা কার্যকরিতার উপর কোন পর্ভাব ফেল না যদি না আপনি আরও রিকোয়েস্ট করেন যে রেন্ডার তখনই রিডর্ (পুনরায় অঙ্কন) করতে হবে যখন setRenderMode() পদ্ধতি ব্যবহার করে পরিবর্তন করে, সুতরাং নিশ্চিত করুন যে এই লাইন রেন্ডারের মধ্যে য় আনকে মেনে টড হয়:

```
public MyGLSurfaceView(Context context) {
    ...
    // Render the view only when there is a change in the drawing data
    setRenderMode(GLSurfaceView.RENDERMODE_WHEN_DIRTY);
}
```

রেণ্ডারিং টাশন অফ এঙ্গেল উইন্স ম্যাচন করুন

উপরের উদাহরণ কোড চায়ে যে একটি পাবলিক মেমবের যুক্ত করার
মাধ্যমে আপনার রেন্ডারারের মধ্যে যুক্তিদিয়ে আপনি রেণ্ডারিং টাশন অফ এঙ্গেল
উইন্স ম্যাচন করুন। যেহেতু রেন্ডারার কোড আপনার অফপিলিকে শেনের
পর্ধান ইউজার ইন্টারফেসে থর্ড থেকে একটি পৃথক থর্ডে রান করে ছে,
আপনাকে অবশ্যই এই পাবলিক ভেরিয়েবলকে volatile হিসাবে ডিক্লয়ার করতে
হবে। এখানে এটা করার একটি কোড দেওয়া হলো:

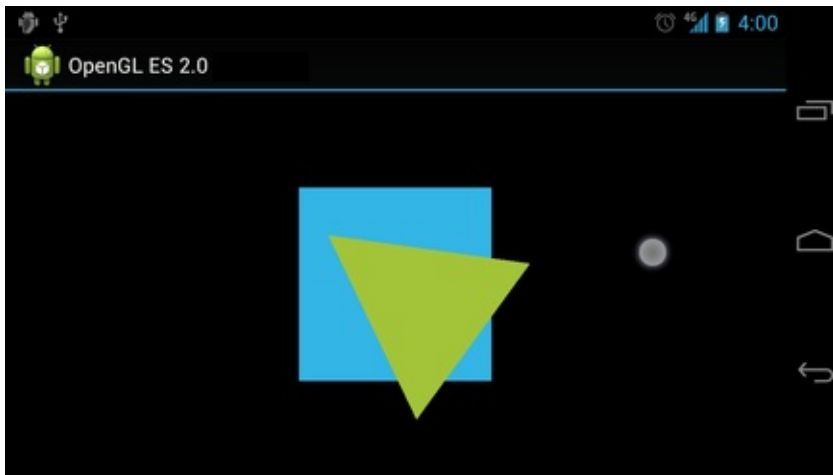
```
public class MyGLRenderer implements GLSurfaceView.Renderer {  
    ...  
    public volatile float mAngle;
```

রেটেটশন পর্য়েয়োগ

ট্যাচ পয়েন্ট দিয়ে তৈরী রেটেটশন পর্য়েয়োগ করতে, কেকাডিট কে মন্ট আউট করুন যা একটি অয় াংগে ল তৈরী করে এবং mAngle যুক্ত করে যা ট্যাচ পয়েন্ট দিয়ে তৈরী অয় াংগে ল ধারন করে:

```
public void onDrawFrame(GL10 gl) {  
    ...  
    float[] scratch = new float[16];  
  
    // Create a rotation for the triangle  
    // long time = SystemClock.uptimeMillis() % 4000L;  
    // float angle = 0.090f * ((int) time);  
    Matrix.setRotateM(mRotationMatrix, 0, mAngle, 0, 0, -1.0f);  
  
    // Combine the rotation matrix with the projection and camera view  
    // Note that the mVPMMatrix factor *must be first* in order  
    // for the matrix multiplication product to be correct.  
    Matrix.multiplyMM(scratch, 0, mVPMMatrix, 0, mRotationMatrix, 0);  
  
    // Draw triangle  
    mTriangle.draw(scratch);  
}
```

যখন আপনি উপরে বিনির্ভর ধাপগুলো সম্পূর্ণ করবেন, পের্াগর্ মরান করবেন এবং তির্ভুজ (ট্রায়াঙ্গল) রেটেট করতে সিঙ্কর্ন জুওঁে আপনার আঙ্গল টেনে আনুন:



ফিগার ১. ট্যাচ পয়েন্ট দিয়ে তির্ভুজ রেটেট হচ্ছে (বৃত্তটি ট্যাচ লোকেশন দেখাচ্ছে)

অ্যানিমেশন সঙ্ঘোজন করা

(<http://developer.android.com/training/animation/index.html>)

অ্যানিমেশন সূক্ষ্ম ষ্ঠিভিজুয়্যালিকউৎযোগ করতে পারেন যা ইউজারকে জানান যে আপনার অ্যাপের কি হচ্ছে এবং আপনার অ্যাপের ইন্টারফেসের মেন্টাল মডেলকে উন্নত করে। অ্যানিমেশন বিশেষভাবে উপকারী যখন স্ক্রিন অবস্থা (স্টেট) পরিবর্তন করে, যেমন যখন কন্টেন্ট লোড করে বা নতুন একশন সহজলভ্য হয়। অ্যানিমেশন আপনার অ্যাপের একটি সুশীলভিত দৃশ্যমানতা সংযোজন করতে পারে, যা আপনার অ্যাপকে একটি উচ্চ গুণ সম্পন্ন হওয়ার অনুভূতি দিবে।

যদিও মনে রাখবেন, যে অ্যানিমেশনের অতিব্যবহার বা তাদের অসময়ে ব্যবহার করা ট্যাক্সটিকারক হতে পারে, যেমন তারা যখন বিলম্ব করে। এই পরিশীলন ক্লাস আপনাকে দেখাবে কীভাবে কিছু সাধারণ ধরনের অ্যানিমেশন বাস্তবায়ন করা যায় যা ব্যবহারযোগ্যতা বাড়িদ্ধ করে এবং আপনার ইউজারকে বিরক্ত না করেই ফেলয়ার সংযোজন করে।

অনুশীলনীসমূহ

দুইটা ভিত্তিতে ককর্ সফড (একটার পরে আরেকটি আসা) করা

শিখন কীভাবে দুইটা ওভারল্যাপ করা ভিত্তিতে ককর্ সফড করা হয়। এই অনুশীলনীতে দেখায় একটি ভিত্তিতে একটি উন্নয়ন মানদন্ডকে ককর্ সফড করতে হয় যাতে ককর্ সট কনটেন্ট ধারণ করে।

সিকরন স্লাইডের জন্য ভিত্তিতে পজ বয় বহার করুন

শিখন কীভাবে স্লাইডিং ট্রানজিশন (স্লাইড পরিবর্তন) অনুভূমিকভাবে সিন্‌নিত সিকরনে গুলোর মধ্যে অ্যানিমেট করা যায়

কাদর ফিল্প অ্যানিমেশন পরদশরন

শিখন কীভাবে দুইটা ভিত্তিতে যর মধ্যে একটি টিফলিপ এমেশন সহকারে অ্যানিমেট করা যায়

ভিত্তিকে জুম করা

শিখন কীভাবে একটি ভিত্তিকে একটি টাচ-টু-জুম-অ্যানিমেশন দিয়ে বড় করা যায়।

লেআউট পরিবর্তন অ্যানিমেশন করা

শিখন কীভাবে বিল্ট-ইন-অ্যানিমেশন সিকর্য করতে হয় যখন লেআউটে চাইল্ড ভিত্তিকে স্যেজান, বিয়েজান বা আপডেট করা হয়।

দুইটা ভিউকে ক্রসফেড করা

(<http://developer.android.com/training/animation/crossfade.html>)

ক্রসফেড অ্যানিমেশন (ডিডলভিহিসাবেও পরিচিত) ধীরে ধীরে একটি ইউজার ইন্টারফেসকে ম্পানেন্টে অদৃশ্য করে তথাকথন একইসাথে অন্য আরেকটি টিকে দৃশ্যমান করা হয়, এই অ্যানিমেশন পরিস্থিতির জন্য উপকারী যেখানে আপনি কন্টেন্ট বা ভিউকে আপনার অ্যাপে বিনিময় করতে চান। ক্রসফেড খুবই সুকৃষ্ট এবং ছোট্ট কনসেপ্টে একটিটি সংক্রমণ থেকে পরবর্তীটিতে সহজভাবে পরিবর্তন হওয়ার সুযোগ প্রস্তুত করে। যখন আপনি এগুলো ব্যবহার করেছেন না, তথাপি পরিবর্তন মাঝে মাঝে হঠাৎ বা ব্যস্ততা অনুভব করে।

এখানে কিছু টেকস্ট কন্টেন্টে একটি উন্নয়ন মানদণ্ড থেকে একটি ক্রসফেডের উদাহরণ দেয়া আছে।

Crossfade animation

আপনি যদি সামনে এগিয়ে যেতে চান এবং একটি পুনরুৎপাদন উদাহরণ দেখতে চান, সম্ভ্রম পল/নমুনা অ্যাপটি ডাউনলোড (<http://developer.android.com/shareables/training/Animations.zip>) এবং রান করুন এবং ক্রসফেড উদাহরণ নিবন্ধ চাচন করুন, ক্যাডবাস্তবায়নের জন্য নিম্নলিখিত ফাইলগুলো দেখুন:

- src/CrossfadeActivity.java
- layout/activity_crossfade.xml
- menu/activity_crossfade.xml

ভিউ গুলো তৈরী করুন

ভিউ দুইটি তৈরী করুন যা কর্‌সেফ ডকুমেন্টে চায়। নিম্নে একটি উদাহরণ
একটি উন্নয়ন মানদণ্ড তৈরী করে এবং একটি স্ক্রল করা যায় এমন
টেক্সট ভিউ তৈরী করে:

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/content"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView style="?android:textAppearanceMedium"
            android:lineSpacingMultiplier="1.2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/lorem_ipsum"
            android:padding="16dp" />

    </ScrollView>

    <ProgressBar android:id="@+id/loading_spinner"
        style="?android:progressBarStyleLarge"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center" />

</FrameLayout>
```

অ্যানিমেশন সেটআপ করা

অ্যানিমেশন সেটআপ করতে:

1. ভিউয়ের জন্য মেমব্রার ভেরিয়েবল তৈরী করুন যা আপনি কর্তৃক সেফড করতে চান। আপনার পরবর্তীতে এই সকল রেফারেন্স পরিয়ে আন হবে যখন অ্যানিমেশনের সময়কালে ভিউ পরিমিত করা হবে।
2. ভিউয়ের জন্য যা দৃশ্যমান হচ্ছে, এর দৃশ্যমানতাকে GONE এ সেট করুন। এটা ভিউকে লোড করে সেট সেন্সে থাকা অবস্থায় রাখবে এবং এটা লোড হওয়ার পর লোড শেড করা হবে, পরে সেটকে আরও দ্রুত করতে হবে।
3. একটি মেমব্রার ভেরিয়েবলের মধ্যে config_shortAnimTime সিস্টেম পরপারটি জমিয়ে রাখুন (কয়েশ)। এই পরপারটি অ্যানিমেশনের জন্য একটি মানসম্পন্ন "ছোট" ("short") সময়কাল নির্ধারণ করে দেয়। এই সময়কাল সুক্লে অ্যানিমেশন বাঁধে অ্যানিমেশন বারবার ঘটে তার জন্য আদর্শ। config_longAnimTime এবং config_mediumAnimTime টাও সহজপাওয়া যায় যদি আপনি তাদের ব্যবহার করার ইচ্ছা পরকাশ করেন।

একটি ভিউ কন্ট্রোল ভিউ হিসাবে পূর্ববর্তীকে ডিচতর্ক থেকে লোড করা হবে বারবার একটি উদাহরণ এখানে দেয়া আছে:

```
public class CrossfadeActivity extends Activity {  
  
    private View mContentView;  
    private View mLoadingView;  
    private int mShortAnimationDuration;  
  
    ...  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_crossfade);  
  
        mContentView = findViewById(R.id.content);  
        mLoadingView = findViewById(R.id.loading_spinner);  
  
        // Initially hide the content view.  
        mContentView.setVisibility(View.GONE);  
  
        // Retrieve and cache the system's default "short" animation time.  
        mShortAnimationDuration = getResources().getInteger(  
            android.R.integer.config_shortAnimTime);  
    }  
}
```

ভিউ ক্রসফেড

এখন ভিউগুলো সম্ভ্রমণর ভাবে সেটআপ করা হয়েছে, নীচের কাজগুলো করে তাদের ক্রসফেড করুন:

- যে ভিউ দৃশ্যমান হতে যাচ্ছে (fading in) তার জন্য, ০ এ আলফা ভ্যালু এবং দৃশ্যমানতা **VISIBLE** এ সেট করুন। (যদি রাখেন তবে, এটা পরে ঠিকমতো **GONE** এ সেট করা হয়েছিল) এটা ভিউকে দৃশ্যমান করবে কিনহুত সম্ভ্রমণর ভাবে সবচ্ছথাকবে।
- যে ভিউ দৃশ্যমান হতে যাচ্ছে তার জন্য, ০ থেকে ১ এ এর আলফা ভ্যালু অয়ানিমেট করুন। একই সময়ে যে ভিউ বিবনর হয়ে গেছে (fading out) তার জন্য আলফা ভ্যালু ১ থেকে ০ তে সেট করুন।
- একটি `Animator.AnimatorListener` এর মধ্যে `onAnimationEnd()` ব্যবহার করে, ভিউয়ের দৃশ্যমানতাকে সেট করুন যা **GONE** এ অদৃশ্য হয়েছিল। এমনকি যদিও আলফা ভ্যালু হচ্ছ ০, **GONE** এ ভিউয়ের দৃশ্যমানতাকে সেট করাটা লোউটে স্পসনেয়াথেকে বিবর রাখবে এবং এটা লোউট কয়ালকুলেশন থেকে বাদ দিয়ে দিবে, পরে সসকে আরও দরুততর করবে।

নীচের পদ্ধতিটি এটাকীভাবে করতে হয় তার একটি উদাহরণ দেখায়: `private View mContentView; private View mLoadingView; private int mShortAnimationDuration;`

```
...

private void crossfade() {

    // Set the content view to 0% opacity but visible, so that it is visible
    // (but fully transparent) during the animation.
    mContentView.setAlpha(0f);
    mContentView.setVisibility(View.VISIBLE);

    // Animate the content view to 100% opacity, and clear any animation
    // listener set on the view.
    mContentView.animate()
        .alpha(1f)
        .setDuration(mShortAnimationDuration)
        .setListener(null);

    // Animate the loading view to 0% opacity. After the animation ends,
    // set its visibility to GONE as an optimization step (it won't
    // participate in layout passes, etc.)
    mLoadingView.animate()
        .alpha(0f)
        .setDuration(mShortAnimationDuration)
        .setListener(new AnimatorListenerAdapter() {
            @Override
            public void onAnimationEnd(Animator animation) {
                mLoadingView.setVisibility(View.GONE);
            }
        });
}
```

সিস্করন স্লাইড এর জনয্ ভিউ পেজ বয্ বহার

(<http://developer.android.com/training/animation/screen-slide.html>)

সিস্করন স্লাইড একটি পুনরু সিস্করন থেকে আরেকটি সিস্করনের মধ্যে পরিবর্তন হয় এবং ইউআই এ এটা খুব সাধারণ যেমন সেটআপ উইজার্ড বা স্লাইড শো। এই অনুশীলনী দ্বারা পরদত একটি ViewPager দিয়ে কীভাবে এটা করতে হয় তা আপনাকে দেখায়। ViewPagers সব যেকোনো সিস্করন স্লাইড অ্যানিমেশন করতে পারে। এখানে আছে একটি সিস্করন স্লাইড দেখতে কমন হয় যা একটি কন্টেন্টের সিস্করন থেকে পরিবর্তীটতে পরিবর্তন করে:

Screen slide animation / সিস্করন স্লাইড অ্যানিমেশন

আপনি যদি সামনে এগিয়ে যেতে চান এবং একটি সম্পূর্ণ কার্যরত উদাহরণ দেখেন, নমুনা অ্যাপটি <http://developer.android.com/shareables/training/Animations.zip> করুন এবং রান করুন এবং সিস্করন স্লাইড উদাহরণ নিবর্তাচন করুন। কেড বাস্তবায়নের জনয্ নীচে রফাইলগুলো দেখুন:

- src/ScreenSlidePageFragment.java
- src/ScreenSlideActivity.java
- layout/activity_screen_slide.xml
- layout/fragment_screen_slide_page.xml

ভিউ তৈরী করুন

একটি লেআউট ফাইল তৈরী করুন যা আপনি পরবর্তীতে একটি ফর্গায়েমেন্টের কন্টেন্টের জন্য ব্যবহার করবেন। নিম্নোক্ত উদাহরণিক ছোটেক্সট পর্দাশরন করতে একটি টেক্সট ভিউ ধারণ করে:

```
<com.example.android.animationsdemo.ScrollView
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:id="@+id/content"
  android:layout_width="match_parent"
  android:layout_height="match_parent">

  <TextView style="?android:textAppearanceMedium"
    android:padding="16dp"
    android:lineSpacingMultiplier="1.2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/lorem_ipsum" />

</com.example.android.animationsdemo.ScrollView>
```

ফর্াগে মেন্ট তৈরী করুন

একটি ফর্াগে মেন্ট ক্লাস তৈরী করুন যা onCreateView() পদ্ধতির মধ্যে মাত্র তৈরী করলে আউটটি ফেরত দেয়। তারপর আপনি পর্গের একটি ভিউটির মধ্যে এই ফর্াগে মেন্টের ইনসটেন্স তৈরী করতে পারেন যখনই ইউজারের কাছে পর্দা শর্ন করতে আপনার একটি নতুন পেজ দরকার হবে:

```
public class ScreenSlidePageFragment extends Fragment {  
  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container,  
        Bundle savedInstanceState) {  
        ViewGroup rootView = (ViewGroup) inflater.inflate(  
            R.layout.fragment_screen_slide_page, container, false);  
  
        return rootView;  
    }  
}
```

ভিউ পেজ যুক্ত করা

ViewPager এর পেজ জুড়ের পরিবর্তন করতে বিল্ট-ইন সুইপ জেশচার আছে, এবং তার বাইরে ডিফল্ট সিস্টেমের স্লাইড অ্যানিমেশন পর্দা পরিদর্শন করে, তাই আপনাকে কোন টাইটেল বাক্সে তৈরি করতে হবে না। ViewPager গুলো নতুন পেজ পর্দা পরিদর্শন করার জন্য একটি যোগান হিসাবে PagerAdapter ব্যবহার করে, সুতরাং PagerAdapter ফরগেটমেন্ট ক্লাস ব্যবহার করে যেখানে পূর্বে তৈরি করা হয়েছিল।

শুরু করার জন্য, একটি লেআউট তৈরি করুন একটি ViewPager ধারণ করে:

```
<android.support.v4.view.ViewPager
xmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/pager"
android:layout_width="match_parent"
android:layout_height="match_parent" />
```

একটি একটি ভিউ তৈরি করুন যা নিম্নোক্ত বিষয়গুলো করে থাকে:

- ViewPager সহকারে লেআউট হতে কন্টেন্ট ভিউ সেট করুন
- একটি ক্লাস তৈরি করুন যা FragmentStatePagerAdapter এবং স্ট্রাকচার ক্লাসকে পর্দা পরিদর্শন করে এবং নতুন পেজ হিসাবে ScreenSlidePageFragment এর ইনস্ট্যান্স যোগান দিতে getItem() পদ্ধতি বাস্তবায়ন করুন। পেজ অ্যাডাপ্টার আরও চায় যে আপনি getCount() পদ্ধতি বাস্তবায়ন করুন যা অ্যাডাপ্টার পেজ তৈরি করে তার সংখ্যা কয়েক ফেরত আনে (উদাহরণে ৫ টি)।
- ViewPager এ চধমব অফসেট যুক্ত করুন।
- ফরগেটমেন্টের ডাটাবেস স্ট্রাকচারের মধ্যে ডিভাইসের বাক্স বাটন পেছন দিয়ে দেয়ার মাধ্যমে চালাত করুন।

```
public class ScreenSlidePagerActivity extends FragmentActivity {
```

```

/**
 * The number of pages (wizard steps) to show in this demo.
 */
private static final int NUM_PAGES = 5;

/**
 * The pager widget, which handles animation and allows swiping horizontally to access previous
 * and next wizard steps.
 */
private ViewPager mPager;

/**
 * The pager adapter, which provides the pages to the view pager widget.
 */
private PagerAdapter mPagerAdapter;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_screen_slide_pager);

    // Instantiate a ViewPager and a PagerAdapter.
    mPager = (ViewPager) findViewById(R.id.pager);
    mPagerAdapter = new ScreenSlidePagerAdapter(getFragmentManager());
    mPager.setAdapter(mPagerAdapter);
}

@Override
public void onBackPressed() {
    if (mPager.getCurrentItem() == 0) {
        // If the user is currently looking at the first step, allow the system to handle the
        // Back button. This calls finish() on this activity and pops the back stack.
        super.onBackPressed();
    } else {
        // Otherwise, select the previous step.
        mPager.setCurrentItem(mPager.getCurrentItem() - 1);
    }
}

/**
 * A simple pager adapter that represents 5 ScreenSlidePageFragment objects, in
 * sequence.
 */
private class ScreenSlidePagerAdapter extends FragmentStatePagerAdapter {
    public ScreenSlidePagerAdapter(FragmentManager fm) {
        super(fm);
    }

    @Override
    public Fragment getItem(int position) {
        return new ScreenSlidePageFragment();
    }

    @Override
    public int getCount() {
        return NUM_PAGES;
    }
}

```

```

}

```


পেজ ট্রান্সফরমার দিয়ে অ্যানিমেশন কাস্টমাইজ করা

ডিফল্ট হিস্করেন স্লাইড অ্যানিমেশন থেকে একটি ভিন্ন অ্যানিমেশন প্রদর্শন করতে, `ViewPager.PageTransformer` ইন্টারফেস বাস্তবায়ন করুন এবং ভিউপেজারের সাপ্লাই করুন। ইন্টারফেস একটি একক পদ্ধতি `transformPage()` উন্মোচন করে। হিস্করেনের পরিবর্তনের প্রতিটা পয়েন্টে, এই পদ্ধতি প্রতিটা দৃশ্যমান পেজের জন্য একবার ইকল হয় (সাধারণভাবে এখন একটি ইদৃশ্যমান পেজ থাকে) পাশবর্তী পেজের জন্য শূন্য স্ক্রকরেনিটি বন্ধ করে দিন। উদাহরণস্বরূপ, যদি তিন নম্বর পেজ দৃশ্যমান হয় এবং ইউজার চার নম্বর পেজের যায়, ২, ৩ এবং চার নাম্বার পেজের জন্য অঙ্গতিগত/ইংগিত প্রতিটা ধাপে `transformPage()` কল করা হয়।

আপনার `transformPage()` বাস্তবায়নে, হিস্করেন পেজের পজিশনের উপর ভিত্তি করে কৈন পেজ পরিবর্তন হওয়া পর্যন্তে জটিলতা নির্ধারণ করার জন্য আপনি তারপরে কাস্টম স্লাইড এনিমেশন করতে পারেন, যা `transformPage()` পদ্ধতির `position` প্যারামিটার থেকে ধারণ করা হয়।

`Position` প্যারামিটার নির্দেশ করে কৈথায় একটি পর্দা পেজ আছে যা হিস্করেনের কেন্দ্রের সমপিকর্ত। এটা একটা ডাইনামিক পারিটি যা ইউজারের হিস্করেন জুরে স্কল করার মতো করে পরিবর্তন করে। যখন একটি পেজ হিস্করেন পূনর করে, এটার অবস্থান (পজিশন) ভ্যালু ০। যখন পেজ শূন্য হিস্করেনের ডান পাশ বন্ধ করে অকন করা হয়, এটার অবস্থান ভ্যালু ১। যদি ইউজার এবং ২ পেজের মধ্যে অধরকপথ স্কল করে, প্রথম পেজের অবস্থান -০.৫ এবং দুই নম্বর পেজের ০.৫ অবস্থান থাকে। হিস্করেন পেজের পজিশনের ভিত্তি করে, `setAlpha()`, `setTranslationX()`, বা `setScaleY()` এর মতো পদ্ধতি দিয়ে পেজ পারিটি সেট করার মাধ্যমে আপনি কাস্টম স্লাইড অ্যানিমেশন তৈরি করতে পারেন।

যখন আপনার `PageTransformer` এর একটি বাস্তবায়ন থাকে, আপনার কাস্টম অ্যানিমেশন পরে যোগ করতে আপনার বাস্তবায়ন দিয়ে `setPageTransformer()` কল করুন। উদাহরণস্বরূপ, আপনার যদি `ZoomOutPageTransformer` নামে একটি `PageTransformer` থাকে, আপনি এটার মতো করে আপনার কাস্টম অ্যানিমেশন করতে পারেন:

```
ViewPager pager = (ViewPager) findViewById(R.id.pager);
...
pager.setPageTransformer(true, new ZoomOutPageTransformer());
```

উদাহরণের জন্য `Zoom-out page transformer` এবং `Depth page transformer` সেকশন দেখুন এবং `PageTransformer` এর ভিডিও দেখুন।

পেজ ট্রান্সফর্মার জুম আউট করা

এই পেজ ট্রান্সফর্মার পেজকে কিঞ্চিৎ এবং ম্লান করে যখন পাশের বর্তনী পেজের মধ্যে সংকরল করে হয়। যেহেতু একটি পেজ কেন্দ্রের নিকটস্থ হয়, এটা এর সর্বাভাবিক সাইজে ফিরে আসে এবং দৃশ্যমান হয়।

ZoomOutPageTransformer উদাহরণ

```
public class ZoomOutPageTransformer implements ViewPager.PageTransformer {
    private static float MIN_SCALE = 0.85f;
    private static float MIN_ALPHA = 0.5f;

    public void transformPage(View view, float position) {
        int pageWidth = view.getWidth();
        int pageHeight = view.getHeight();

        if (position < -1) { // [-Infinity,-1]
            // This page is way off-screen to the left.
            view.setAlpha(0);

        } else if (position <= 1) { // [-1,1]
            // Modify the default slide transition to shrink the page as well
            float scaleFactor = Math.max(MIN_SCALE, 1 - Math.abs(position));
            float vertMargin = pageHeight * (1 - scaleFactor) / 2;
            float horzMargin = pageWidth * (1 - scaleFactor) / 2;
            if (position < 0) {
                view.setTranslationX(horzMargin - vertMargin / 2);
            } else {
                view.setTranslationX(-horzMargin + vertMargin / 2);
            }

            // Scale the page down (between MIN_SCALE and 1)
            view.setScaleX(scaleFactor);
            view.setScaleY(scaleFactor);

            // Fade the page relative to its size.
            view.setAlpha(MIN_ALPHA +
                (scaleFactor - MIN_SCALE) /
                (1 - MIN_SCALE) * (1 - MIN_ALPHA));

        } else { // (1,+Infinity]
            // This page is way off-screen to the right.
            view.setAlpha(0);
        }
    }
}
```

ডেপথ পেজ ট্রান্সফর্মার

এই পেজ ট্রান্সফর্মার বাম পাশের স্লাইডিং ডিপেজের জন্য ডিফল্ট স্লাইড অ্যানিমেশন বয়বহার করে, যখন ডান পাশের স্লাইডিং ডিপেজের জন্য একটি “ডেপথ” অ্যানিমেশন বয়বহার করা হয়। এই ডেপথ অ্যানিমেশন পেজ আউটকে ফেড করে, এবং সুসঙ্গতভাবে আনুপাতিক হারে কমায়।

DepthPageTransformer উদাহরণ

নোট: ডেপথ অ্যানিমেশনের সময়, ডিফল্ট অ্যানিমেশন (একটি স্ক্রিন স্লাইড) তখনও জায়গা নেয়, সুতরাং আপনাকে অবশ্যই একটি নৈগটিভ X অনুবাদ দিয়ে স্ক্রিন স্লাইড নিবারণ করতে হবে। উদাহরণ:

```
view.setTranslationX(
    1 * view.getWidth() * position);
```

নিম্নোক্ত উদাহরণ দেখায় একটি কায়ররত পেজ ট্রান্সফর্মার কীভাবে ডিফল্ট স্ক্রিন স্লাইড অ্যানিমেশন নিবারণ করা হয়:

```
public class DepthPageTransformer implements ViewPager.PageTransformer {
    private static float MIN_SCALE = 0.75f;

    public void transformPage(View view, float position) {
        int pageWidth = view.getWidth();

        if (position < -1) { // [-Infinity,-1]
            // This page is way off-screen to the left.
            view.setAlpha(0);

        } else if (position <= 0) { // [-1,0]
            // Use the default slide transition when moving to the left page
            view.setAlpha(1);
            view.setTranslationX(0);
            view.setScaleX(1);
            view.setScaleY(1);

        } else if (position <= 1) { // (0,1]
            // Fade the page out.
            view.setAlpha(1 - position);

            // Counteract the default slide transition
            view.setTranslationX(pageWidth * -position);

            // Scale the page down (between MIN_SCALE and 1)
            float scaleFactor = MIN_SCALE
                + (1 - MIN_SCALE) * (1 - Math.abs(position));
            view.setScaleX(scaleFactor);
            view.setScaleY(scaleFactor);

        } else { // (1,+Infinity]
            // This page is way off-screen to the right.
            view.setAlpha(0);
        }
    }
}
```

ক্যাডফ্লিপ অ্যানিমেশন পরিদর্শন করা

(<http://developer.android.com/training/animation/cardflip.html>)

এই অনুশীলনী আপনাকে শেখাবে কাস্টম ফর্গেমন্ট অ্যানিমেশনের সাথে একটি ক্যাডফ্লিপ অ্যানিমেশন কীভাবে করতে হয়। ক্যাডফ্লিপ একটি অ্যানিমেশন দেখানোর মাধ্যমে কন্টেন্টের ভিত্তির মধ্যে অ্যানিমেশন করে যা একটি ক্যাডফ্লিপ করার অনুকরণ করে।

এখানে দেয়া আছে একটি ফ্লিপ দেখতে কেমন হবে:

Card flip animation

আপনি যদি সামনে এগিয়ে যেতে চান এবং একটি পুনরুৎপাদন উদাহরণ দেখতে চান, নমুনা অ্যাপ [Download](#) এবং রান করুন এবং ক্যাডফ্লিপ উদাহরণ নিবর্তন করুন। ক্যাডফ্লিপ তবায়নের জন্য নিম্নলিখিত ফাইল দেখুন:

- src/CardFlipActivity.java
- animator/card_flip_right_in.xml
- animator/card_flip_right_out.xml
- animator/card_flip_left_in.xml
- animator/card_flip_left_out.xml
- layout/fragment_card_back.xml
- layout/fragment_card_front.xml

অয ় ানি নে ম ট র তৈ ত র ী ক র ু ন

ক াড র ্ ফ ় লে প র জন য ্ অ য ্ ানি নে ম শ ন তৈ ত র ী ক র ু ন। আপ ন ার জন য ্ দু ই ট া
অ য ্ ানি নে ম ট র ল াগে ব য খ ন ক াড র ্ র স ম ্ ম ্ খ ভ াগ অ য ্ ানি নে ম ট আ উ ট ক রে এ ব ্
ব াম প াশে র দি কে এ ব ্ অ য ্ ানি নে ম ট ই ন ক রে এ ব ্ ব াম প াশে থ কে। আপ ন ার জন য ্
আ র ও দু ই ট া অ য ্ ানি নে ম ট র ল াগে ব য খ ন ক াড র ্ র প শ ্ চ াদ ভ াগ অ য ্ ানি নে ম ট
ই ন ক রে এ ব ্ ড ান প াশে থ কে এ ব ্ অ য ্ ানি নে ম ট আ উ ট ক রে ড ান প াশে র দি কে।

card_flip_left_in.xml

```
<set xmlns:android="http://schemas.android.com/apk/res/android">
  <!-- Before rotating, immediately set the alpha to 0. -->
  <objectAnimator
    android:valueFrom="1.0"
    android:valueTo="0.0"
    android:propertyName="alpha"
    android:duration="0" />

  <!-- Rotate. -->
  <objectAnimator
    android:valueFrom="-180"
    android:valueTo="0"
    android:propertyName="rotationY"
    android:interpolator="@android:interpolator/accelerate_decelerate"
    android:duration="@integer/card_flip_time_full" />

  <!-- Half-way through the rotation (see startOffset), set the alpha to 1. -->
  <objectAnimator
    android:valueFrom="0.0"
    android:valueTo="1.0"
    android:propertyName="alpha"
    android:startOffset="@integer/card_flip_time_half"
    android:duration="1" />
</set>
```

card_flip_left_out.xml

```
<set xmlns:android="http://schemas.android.com/apk/res/android">
  <!-- Rotate. -->
  <objectAnimator
    android:valueFrom="0"
    android:valueTo="180"
    android:propertyName="rotationY"
    android:interpolator="@android:interpolator/accelerate_decelerate"
    android:duration="@integer/card_flip_time_full" />

  <!-- Half-way through the rotation (see startOffset), set the alpha to 0. -->
  <objectAnimator
    android:valueFrom="1.0"
    android:valueTo="0.0"
    android:propertyName="alpha"
    android:startOffset="@integer/card_flip_time_half"
    android:duration="1" />
</set>
```

card_flip_right_in.xml

```
<set xmlns:android="http://schemas.android.com/apk/res/android">
  <!-- Before rotating, immediately set the alpha to 0. -->
  <objectAnimator
    android:valueFrom="1.0"
    android:valueTo="0.0"
    android:propertyName="alpha"
    android:duration="0" />

  <!-- Rotate. -->
  <objectAnimator
    android:valueFrom="180"
    android:valueTo="0"
    android:propertyName="rotationY"
    android:interpolator="@android:interpolator/accelerate_decelerate"
    android:duration="@integer/card_flip_time_full" />

  <!-- Half-way through the rotation (see startOffset), set the alpha to 1. -->
  <objectAnimator
    android:valueFrom="0.0"
    android:valueTo="1.0"
    android:propertyName="alpha"
    android:startOffset="@integer/card_flip_time_half"
    android:duration="1" />
```


card_flip_right_out.xml

```
<set xmlns:android="http://schemas.android.com/apk/res/android">
  <!-- Rotate. -->
  <objectAnimator
    android:valueFrom="0"
    android:valueTo="-180"
    android:propertyName="rotationY"
    android:interpolator="@android:interpolator/accelerate_decelerate"
    android:duration="@integer/card_flip_time_full" />

  <!-- Half-way through the rotation (see startOffset), set the alpha to 0. -->
  <objectAnimator
    android:valueFrom="1.0"
    android:valueTo="0.0"
    android:propertyName="alpha"
    android:startOffset="@integer/card_flip_time_half"
    android:duration="1" />
</set>
```

ভিউ তৈরী করুন

ক্যাডব্রুও ট্রিটিউ প্যাসব্রু পৃথক লেআউট যা আপনি যা চান সেই কন্টেন্ট ধারণ করে, যেমন টেক্সটের দুইটা স্ক্রিন, দুইটা ইমেজ, বা ফিল্ডে পরম্পরায় ভিউয়ের যেকোন সমাহার। আপনি তখন ফরগেট দুইটা লেআউট ব্যবহার করতে পারেন যা আপনি পরে অ্যানিমেশন করতে পারবেন। নীচের লেআউট একটি ক্যাডব্রুও একটি প্যাসব্রু তৈরী করে যা টেক্সট দেখায়:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#a6c"
    android:padding="16dp"
    android:gravity="bottom">

    <TextView android:id="@android:id/text1"
        style="?android:textAppearanceLarge"
        android:textStyle="bold"
        android:textColor="#fff"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/card_back_title" />

    <TextView style="?android:textAppearanceSmall"
        android:textAllCaps="true"
        android:textColor="#80ffff"
        android:textStyle="bold"
        android:lineSpacingMultiplier="1.2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/card_back_description" />

</LinearLayout>
```

এক ক্যাডব্রুর অপর প্যাসব্রু যা একটি ImageView প্রদর্শন করে:

```
<ImageView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:src="@drawable/image1"
    android:scaleType="centerCrop"
    android:contentDescription="@string/description_image_1" />
```

ফর্াগমেন্ট তৈরী করা

কাডের সম্মুখ এবং পিছন ভাগের জন্য ফর্াগমেন্ট ক্লাস তৈরী করুন। এই ক্লাসগুলো লেআউটে ফরত দেয় যা আপনি পূর্বে বর্ণিত টা ফর্াগমেন্টের onCreateView() পদ্ধতিতে তৈরী করেছিলেন। এরপর আপনি পর্গেরেন্ট একটি ভিউতে এই ফর্াগমেন্টের ইনস্টেন্স তৈরী করে ত পর্গেরেন্টে যথেন আপনি কাডের পর্দাশর্ন করতে চান। নীচের উদাহরণ পর্গেরেন্ট একটি ভিউটির মধ্যে থাকা ফর্াগমেন্ট ক্লাস দেখায় যা তাদেব ব্যবহার করে:

```
public class CardFlipActivity extends Activity {
    ...
    /**
     * A fragment representing the front of the card.
     */
    public class CardFrontFragment extends Fragment {
        @Override
        public View onCreateView(LayoutInflater inflater, ViewGroup container,
            Bundle savedInstanceState) {
            return inflater.inflate(R.layout.fragment_card_front, container, false);
        }
    }

    /**
     * A fragment representing the back of the card.
     */
    public class CardBackFragment extends Fragment {
        @Override
        public View onCreateView(LayoutInflater inflater, ViewGroup container,
            Bundle savedInstanceState) {
            return inflater.inflate(R.layout.fragment_card_back, container, false);
        }
    }
}
```

কাডর্ ফ্লপ অর্ ানি মেট করা

এখন আপনার একটি পর্েরন্ট একটি টিভিটির মধ্যে ফর্াগেমন্টি পর্দশর্ন করার পর্েয়োজন্ এটা করতে, পর্থমে আপনার একটি টিভিটির জনয্ লেআউট তৈরী করুন। নীচে চর উদাহরন একটি FrameLayout করে ছে যােত আপনি ফর্াগেমন্টির রানটাইমের সময় এেত এটা সেট করতে পারবেন:

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

একটি টিভিটি কে ােড, লেআউট হেত কনটেইন্টিভিউ সেট করুন যা আপনি মাতর্ তৈরী করে ছেন। একটি ডিফল্টি ফর্াগেমন্টি দেখােনা একটি ডালে াচন্ তা যখন একটি টিভিটি তৈরী হয়, সুতরাং ানি মেন্ নাক্ ত উদাহরন একটি টিভিটি আপনার কে দেখায় বাই ডিফল্টি কীভাবে কাডর্র সময় ম্খভাগ পর্দশর্ন করতে হয়:

```
public class CardFlipActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_activity_card_flip);

        if (savedInstanceState == null) {
            getSupportFragmentManager()
                .beginTransaction()
                .add(R.id.container, new CardFrontFragment())
                .commit();
        }
        ...
    }
}
```

এখন আপনার সময় ম্খভাগ পর্দশর্ন করার বয্ বস্ থা আেছ, আপনি একটি যথায়থ সময়ে ফ্লপ অর্ ানি মেশন দিয়ে কাডর্ ও পেছনভাগ পর্দশর্ন করতে পারেন। কাডর্র অনয্ পাশবর্ পর্দশর্ন করতে একটি পদ্ধতি তৈরী করুন যা ানি মেন্ নাক্ ত বিষয়গুলো সম্ পাদন করে:

- কাস্ টম অর্ ানি মেশন সেট করুন যা ফর্াগেমন্টি পর্িবতর্ন করার জনয্ আপনি ইতিপূেবর্ তৈরী করে ছেন।
- একটি নতুন ফর্াগেমন্টির সােথ বতর্মােন পর্দিশর্ত ফর্াগেমেন্ট কে পর্তিস্ থাপন করুন এবং কাস্ টম অর্ ানি মেশন যা আপনি তৈরী করে ছেন তা দিয়ে এই ইেভন্ট অর্ ানি মেট করুন।
- ফর্াগেমন্টি বয্ াক সে্ টকে পূেবর্ পর্দিশর্ত ফর্াগেমন্টি যুক্ ত করুন তাই যখন ইউজার বাক বাটনে পর্স করবে, কাডর্ ফ্লপ করে টেছেন চেল যাবে।

```
private void flipCard() {
```

```

if (mShowingBack) {
    getFragmentManager().popBackStack();
    return;
}

// Flip to the back.

mShowingBack = true;

// Create and commit a new fragment transaction that adds the fragment for the back of
// the card, uses custom animations, and is part of the fragment manager's back stack.

getFragmentManager()
    .beginTransaction()

    // Replace the default fragment animations with animator resources representing
    // rotations when switching to the back of the card, as well as animator
    // resources representing rotations when flipping back to the front (e.g. when
    // the system Back button is pressed).
    .setCustomAnimations(
        R.animator.card_flip_right_in, R.animator.card_flip_right_out,
        R.animator.card_flip_left_in, R.animator.card_flip_left_out)

    // Replace any fragments currently in the container view with a fragment
    // representing the next page (indicated by the just-incremented currentPage
    // variable).
    .replace(R.id.container, new CardBackFragment())

    // Add this transaction to the back stack, allowing users to press Back
    // to get to the front of the card.
    .addToBackStack(null)

    // Commit the transaction.
    .commit();

}

```

ভিউ জুম করা

(<http://developer.android.com/training/animation/zoom.html>)

এই অনুশীলনটি আপনাকে কেরে দেখাবে কীভাবে একটি টাচ-টু-জুম অ্যানিমেশন মশন করা যায়, যা অ্যাপের জনম পরে যাজনীয় যেমন একটি পূর্ণরূপ সাইজের ইমেজে একটি থামবুনেইল থেকে একটি ভিউ অ্যানিমেশন ট করার জনম ফ্রেটা গয়ালারী যা সিস্করুন পূনর করে।

এখানে একটি টাচ-টু-জুম অ্যানিমেশন দেখতে কেমন হবে তা দেখা হয়েছে যা সিস্করুনকে পূর্ণরূপ করে একটি ইমেজ থামবুনেইল পরে সারিত করে:

Zoom animation/জুম অ্যানিমেশন

আপনি যদি সামনে এগিয়ে যেতে চান এবং একটি পূর্ণরূপ কায়রুরত উদাহরণ দেখতে চান, নমুনা অ্যাপ [Download](#) এবং রান করুন এবং জুম উদাহরণ নিবর্চন করুন। কোড বাস্তবায়নের জনম নিম্ন নাক্ত ফাইল দেখুন:

- src/TouchHighlightImageButton.java (a simple helper class that shows a blue touch highlight when the image button is pressed)
- src/ZoomActivity.java
- layout/activity_zoom.xml

ভিউ তৈরী করা

একটি টেলিআউট ফাইল তৈরী করুন যা কন্টেইনার বড় এবং ছোট স্ক্রিন ধারণ করে যা আপনি জুম করতে চান। নীচে উদাহরণটি ক্লিকে যাগ য় ইমেজ থামব্ নইল এর জন্য য় একটি ImageButton তৈরী করে এবং একটি ImageView যা ইমেজের বড় ভিউ পর্ দশর্ ন করে:

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="16dp">

        <ImageButton
            android:id="@+id/thumb_button_1"
            android:layout_width="100dp"
            android:layout_height="75dp"
            android:layout_marginRight="1dp"
            android:src="@drawable/thumb1"
            android:scaleType="centerCrop"
            android:contentDescription="@string/description_image_1" />

    </LinearLayout>

    <!-- This initially-hidden ImageView will hold the expanded/zoomed version of
    the images above. Without transformations applied, it takes up the entire
    screen. To achieve the "zoom" animation, this view's bounds are animated
    from the bounds of the thumbnail button above, to its final laid-out
    bounds.
    -->

    <ImageView
        android:id="@+id/expanded_image"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:visibility="invisible"
        android:contentDescription="@string/description_zoom_touch_close" />

</FrameLayout>
```

জুম অ্যানিমেশন সেটআপ করা

ইতিপূর্বে বর্ আপন যদি আপনার লেআউট পর্যালোচনা করে থাকেন, ইন্ডেন্ট হ্যান্ডেলারস সেটআপ করুন যা জুম অ্যানিমেশনকে সিকর্য করে। নীচের উদাহরণটি জুম অ্যানিমেশন সম্পাদন করতে ImageButton এ একটি View.OnClickListener যুক্ত করে যখন ইউজার ইমেজ বাটনে ক্লিক করে:

```
public class ZoomActivity extends FragmentActivity {
    // Hold a reference to the current animator,
    // so that it can be canceled mid-way.
    private Animator mCurrentAnimator;

    // The system "short" animation time duration, in milliseconds. This
    // duration is ideal for subtle animations or animations that occur
    // very frequently.
    private int mShortAnimationDuration;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_zoom);

        // Hook up clicks on the thumbnail views.

        final View thumb1View = findViewById(R.id.thumb_button_1);
        thumb1View.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                zoomImageFromThumb(thumb1View, R.drawable.image1);
            }
        });

        // Retrieve and cache the system's default "short" animation time.
        mShortAnimationDuration = getResources().getInteger(
            android.R.integer.config_shortAnimTime);
    }
    ...
}
```


ভিউ জুম করুন

যখন সূয়ে যাগ হবে আপনি তখন সাধারণ সাইজের ভিউ থেকে জুম ভিউয়ে অর্গানিমেট করতে পারেন। সাধারণভাবে আপনার এক সাধারণ সাইজের ভিউয়ের সীমা থেকে বড়-আকারের ভিউ এর দিকে অর্গানিমেট করতে হবে। নীচের পদ্ধতি দেখায় কীভাবে একটি জুম অর্গানিমেশন পর্য়ে যাগ করা যায় যা নীচ কাজগুলো করার মাধ্যমে একটি বড়করা ভিউয়ে একটি ইমেজ থামব্‌নেইল জুম করে:

1. হিডেন "জুম-ইন" (বড় করা) `ImageView` এ অধিক সংখ্যক ইমেজ বরাদ্দ করুন। নীচের উদাহরণ সরলতার জন্য ইউআই থের্ডে একটি বড় আকারের ইমেজ রিসোর্স লোড করে। ইউআই থের্ডে ব্লিক্‌বন্ড করতে একটি পৃথক থের্ডে লোড করার মাধ্যমে আপনি এটা করতে চাইতে পারেন এবং তারপর ইউআই থের্ডে বিটম্যাপ সেট করতে পারেন।
2. `ImageView` এর জন্য বার্ডাউন্ড শুরুর এবং শেষকে গণনা করুন।
3. `X`, `Y`, (`SCALE_X`, এবং `SCALE_Y`) এ ৪ টি অবস্থানের এবং সাইজের পন্টিট পর্পারটির অর্গানিমেশন করুন, একইভাবে শুরুর সীমা থেকে শেষের সীমার মধ্যে। এই চার ধরনের অর্গানিমেশন একটি `AnimatorSet` এ যুক্ত করা হয় যাতে তারা একইসাথে শুরুর করতে পারেন।
4. একই ধরনের অর্গানিমেশন রান করলে জুম পেছনে চলে যায় কিন্ত বিপরীতভাবে ইউজার স্কর্ন টাচ করে যখন ইমেজ জুম ইন করা হয়। `ImageView` এ একটি `View.OnClickListener` যুক্ত করে আপনি এটা করতে পারেন। যখন ক্লিক করা হয়, `ImageView` ইমেজ থামব্‌নেইল সাইজে পিছিয়ে আসা কে মিনিমাইজ কওে এবং এটা তার দৃশ্যমানতাকে `GONE` এ সেট করে এটাকে হাইড করে।

```
private void zoomImageFromThumb(final View thumbView, int imageResId) {
    // If there's an animation in progress, cancel it
    // immediately and proceed with this one.
    if (mCurrentAnimator != null) {
        mCurrentAnimator.cancel();
    }

    // Load the high-resolution "zoomed-in" image.
    final ImageView expandedImageView = (ImageView) findViewById(
        R.id.expanded_image);
    expandedImageView.setImageResource(imageResId);

    // Calculate the starting and ending bounds for the zoomed-in image.
    // This step involves lots of math. Yay, math.
    final Rect startBounds = new Rect();
    final Rect finalBounds = new Rect();
    final Point globalOffset = new Point();

    // The start bounds are the global visible rectangle of the thumbnail,
    // and the final bounds are the global visible rectangle of the container
    // view. Also set the container view's offset as the origin for the
    // bounds, since that's the origin for the positioning animation
    // properties (X, Y).
    thumbView.getGlobalVisibleRect(startBounds);
    findViewById(R.id.container)
        .getGlobalVisibleRect(finalBounds, globalOffset);
    startBounds.offset(-globalOffset.x, -globalOffset.y);
    finalBounds.offset(-globalOffset.x, -globalOffset.y);

    // Adjust the start bounds to be the same aspect ratio as the final
    // bounds using the "center crop" technique. This prevents undesirable
    // stretching during the animation. Also calculate the start scaling
    // factor (the end scaling factor is always 1.0).
    float startScale;
    if ((float) finalBounds.width() / finalBounds.height()
        > (float) startBounds.width() / startBounds.height()) {
        // Extend start bounds horizontally
        startScale = (float) startBounds.height() / finalBounds.height();
        float startWidth = startScale * finalBounds.width();
    }
```

```

        float deltaWidth = (startWidth - startBounds.width()) / 2;
        startBounds.left -= deltaWidth;
        startBounds.right += deltaWidth;
    } else {
        // Extend start bounds vertically
        startScale = (float) startBounds.width() / finalBounds.width();
        float startHeight = startScale * finalBounds.height();
        float deltaHeight = (startHeight - startBounds.height()) / 2;
        startBounds.top -= deltaHeight;
        startBounds.bottom += deltaHeight;
    }

    // Hide the thumbnail and show the zoomed-in view. When the animation
    // begins, it will position the zoomed-in view in the place of the
    // thumbnail.
    thumbView.setAlpha(0f);
    expandedImageView.setVisibility(View.VISIBLE);

    // Set the pivot point for SCALE_X and SCALE_Y transformations
    // to the top-left corner of the zoomed-in view (the default
    // is the center of the view).
    expandedImageView.setPivotX(0f);
    expandedImageView.setPivotY(0f);

    // Construct and run the parallel animation of the four translation and
    // scale properties (X, Y, SCALE_X, and SCALE_Y).
    AnimatorSet set = new AnimatorSet();
    set
        .play(ObjectAnimator.ofFloat(expandedImageView, View.X,
            startBounds.left, finalBounds.left))
        .with(ObjectAnimator.ofFloat(expandedImageView, View.Y,
            startBounds.top, finalBounds.top))
        .with(ObjectAnimator.ofFloat(expandedImageView, View.SCALE_X,
            startScale, 1f)).with(ObjectAnimator.ofFloat(expandedImageView,
            View.SCALE_Y, startScale, 1f));
    set.setDuration(mShortAnimationDuration);
    set.setInterpolator(new DecelerateInterpolator());
    set.addListener(new AnimatorListenerAdapter() {
        @Override
        public void onAnimationEnd(Animator animation) {
            mCurrentAnimator = null;
        }

        @Override
        public void onAnimationCancel(Animator animation) {
            mCurrentAnimator = null;
        }
    });
    set.start();
    mCurrentAnimator = set;

    // Upon clicking the zoomed-in image, it should zoom back down
    // to the original bounds and show the thumbnail instead of
    // the expanded image.
    final float startScaleFinal = startScale;
    expandedImageView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if (mCurrentAnimator != null) {
                mCurrentAnimator.cancel();
            }

            // Animate the four positioning/sizing properties in parallel,
            // back to their original values.
            AnimatorSet set = new AnimatorSet();
            set.play(ObjectAnimator
                .ofFloat(expandedImageView, View.X, startBounds.left))
                .with(ObjectAnimator
                    .ofFloat(expandedImageView,
                        View.Y, startBounds.top))
                .with(ObjectAnimator
                    .ofFloat(expandedImageView,
                        View.SCALE_X, startScaleFinal))
                .with(ObjectAnimator
                    .ofFloat(expandedImageView,
                        View.SCALE_Y, startScaleFinal));
            set.setDuration(mShortAnimationDuration);
            set.setInterpolator(new DecelerateInterpolator());
            set.addListener(new AnimatorListenerAdapter() {
                @Override
                public void onAnimationEnd(Animator animation) {
                    thumbView.setAlpha(1f);

```

```
        expandedImageView.setVisibility(View.GONE);
        mCurrentAnimator = null;
    }

    @Override
    public void onAnimationCancel(Animator animation) {
        thumbView.setAlpha(1f);
        expandedImageView.setVisibility(View.GONE);
        mCurrentAnimator = null;
    }
});
set.start();
mCurrentAnimator = set;
}
});
}
```

লেআউট পরিবর্তন অ্যানিমেশন করা

(<http://developer.android.com/training/animation/layout.html>)

একটি লেআউট অ্যানিমেশন হচ্ছে একটি পূর্বে লোড করা অ্যানিমেশন যা লেআউট কনফিগারেশন থেকে কান পরিবর্তন করার সময় পরিত্যক্ত। সিস্টেম রান করে থাকে। এই সব করতে আপনার যাদরকার হবে তা হচ্ছে লেআউটে একটি এটির বিউটে স্ট করুন অ্যান্ডারয়েড সিস্টেমে এই সকল লেআউট পরিবর্তন অ্যানিমেশন করা কথাবলতে এবং সিস্টেম ডিফল্ট অ্যানিমেশন আপনার জন্য সম্পন্ন হয়।

টিপ: আপনি যদি কাস্টম লেআউট অ্যানিমেশন সরবরাহ করতে চান, একটি `LayoutTransition` অবজেক্ট তৈরি করুন এবং `setLayoutTransition()` পদ্ধতি দিয়ে লেআউটে সরবরাহ করুন।

এখানে দেয়া আছে যখন একটি লিস্ট আইটেম যুক্ত হবে তখন ডিফল্ট লেআউটে দিতে কমন হবে:

Layout animation/ লেআউট অ্যানিমেশন

আপনি যদি সামনে এগিয়ে যেতে চান এবং একটি পূর্ন কাযের উদাহরণ দেখতে চান, নমুনা অ্যাপ [Download](#) এবং রান করুন এবং কর্তৃক সফেড উদাহরণ নিবরণ চান করুন। কেডবাস্তবায়নের জন্য নিম্নলিখিত ফাইল দেখুন:

1. `src/LayoutChangesActivity.java`
2. `layout/activity_layout_changes.xml`
3. `menu/activity_layout_changes.xml`

একটি লেআউট তৈরী করা

আপনার একটি ভিউটির লেআউট XML ফাইলে, লেআউটের জন্য `android:animateLayoutChanges` এটির বিউট `true` সেট করুন যা আপনি অ্যানিমেশন সিকর্য করার জন্য চান। উদাহরণস্বরূপ:

```
<LinearLayout android:id="@+id/container"
    android:animateLayoutChanges="true"
    ...
/>
```

লেআউটে থেক আইটেম এয়াড (যুক্ত), আপডেট বারিমুয্ত (বিয়োজন) করা

এখন আপনার লেআউটে আইটেম এয়াড (যুক্ত), আপডেট বারিমুয্ত (বিয়োজন) করা পরেয়োজন এবং আইটেমগুলো সব্বয় ঞিকর় য়ভা়ে অয়া়িনে়ে মট হ়েব:

```
private ViewGroup mContainerView;  
...  
private void addItem() {  
    View newView;  
    ...  
    mContainerView.addView(newView, 0);  
}
```

কানেক্টিভিটি এবং ক্লাউড দিয়ে অ্যাপস তৈরী

(<http://developer.android.com/training/building-connectivity.html>)

এই ক্লাস আপনাকে শেখাবে আপনার অ্যাপকে ইউজারের ডিভাইসে র‍‌বারাইরেও কীভাবে বিবেশব্র সাথে সংযুক্ত করা যাবে। আপনি জানবেন কীভাবে এলাকার মধ্যে অন্য ডিভাইসে সরাসরি যুক্ত করতে হয়, ইন্টারনেটের সাথে যুক্ত করতে হয় আপনার ডাটার ব্যাকআপ এবং সিঙ্ক (sync) এবং অন্য অন্য কাজগুলো করতে হয়।

১. ডিভাইসে তারিবিহীন সংযোগ

নেটওয়ার্কের সািভরসিডসেকাভারি ব্যবহার করে কীভাবে বেরাকাল ডিভাইসে সংযোগে বের করতে হয় এবং সংযোগ করতে হয় এবং ওয়াই-ফাই দিয়ে পিয়ার টু পিয়ার সংযোগ তৈরী করতে হয়।

- নেটওয়ার্কের সািভরসিডসেকাভারি ব্যবহার
- ওয়াই-ফাই দিয়ে পিয়ার টু পিয়ার (P2P) সংযোগ তৈরী
- সািভরসিডসেকাভারি জন্য ওয়াই-ফাই পিয়ার টু পিয়ার ব্যবহার

২. নেটওয়ার্ক অপারেশন সম্পাদন

কীভাবে একটি নেটওয়ার্ক সংযোগ তৈরী করতে হয়, কানেক্টিভিটিতে পরিবর্তনের জন্য সংযোগ মনিটর করতে হয় এবং XML ডাটার সাথে লেনদেন সম্পন্ন করতে হয়।

- নেটওয়ার্কের সংযুক্ত করা
- নেটওয়ার্ক ব্যবহার ব্যবস্থাপনা
- XML ডাটা পারিসিঙ্ক করা

৩. ব্যাটারী খরচ না করে ডাটা সংস্থানান্তর করা

ব্যাটারীর উপর আপনার অ্যাপের প্রভাব কীভাবে কমে যাবে আনবেন যখন ডাউনলোড বা অন্য অন্য নেটওয়ার্ক লেনদেন সংঘটিত হয়ে থাকে।

- কার্যকরী নেটওয়ার্ক পরবেশে যোগ্যতার জন্য ডাইনলোড অপটিমাইজ করা
- নিয়মিত আপডেটের প্রভাব কমে যাবে আনা
- অপ্রয়োজনীয় ডাউনলোড প্রয়োজন নেই
- কানেক্টিভিটির উপর ভিত্তি করে পর্যাপ্ত পরিবর্তন করা

৪. ক্লাউডে সিঙ্ক করা

কীভাবে অ্যাপ এবং ইউজারের ডাটাকে ক্লাউডের দুর্বতরী ওয়েব সািভরসিডসিঙ্ক এবং ব্যাকআপ করতে হয় এবং ডাটা কীভাবে বহুবিধ ডিভাইসে রিস্টার করতে হয়।

- ব্যাকআপ API ব্যবহার
- গুগল ক্লাউড মেসেজিং এর অধিকাংশ তৈরী করেন

৫. ক্লাউডে সভা সংঘাত সমাধান করা

অ্যাপের জন্য কীভাবে একটি শক্তিশালী সংঘাত সমাধান কৌশল ডিজাইন করা যায় যা ক্লাউড ডাটাকে সভা করে।

৬. সিঙ্ক অ্যাডাপ্টির ব্যবহার করে ডাটা সংস্থানান্তর

অয্যান্ ডরয়েড সিসঙ্ক অয্যাদাপ্ টর ফের্ম ওয়াকর বয্যবহার করে
কীভাবে বক্ লাউড এবং ডিভাইসে সরমধয্য ডাটা স্থানান্ তর করা যায়।

1. স্টাব অি থিনিং টেকটর তৈরী করা
2. স্টাব কন্টেটন্ টপর্দানকরী তৈরী করা
3. সিসঙ্ক অয্যাদাপ্ টর তৈরী করা
4. সিসঙ্ক অয্যাদাপ্ টর রান করা

ডিভাইসে ওয়ারলেস সংযোগ

(<http://developer.android.com/training/connect-devices-wirelessly/index.html>)

ক্লাউড এর সাথে কমিউনিকেশন সিকর্য করার ছাড়াও, অ্যান্ড্রয়েডের তারবিহীন APIs একই লোকাল নেটওয়ার্কে অর্থাৎ ডিভাইসের সাথে ও কমিউনিকেশন সিকর্য করে এবং এমনকি যেটাকে কান নেটওয়ার্কে নেই কিন্ত আংশপাংশ উপস্থিত আছে তার সাথেও। নেটওয়ার্ক সািভর্স ডিউস্কাভারি (এনএসডি) এর সংযোগজন একটি অ্যাপলিকেশনকে নিকটবর্তী চলমান ডিভাইস যার সাথে এটা সংযোগ স্থাপন করতে পারে তাকে খুঁজে বের করে তাকে দিয়ে এটাকে সামনে নিয়ে যায়। আপনার অ্যাপলিকেশনের মধ্যে এই কায়রুকারিতাকে একিভূত করা আপনাকে ব্যাপক আকারের বৈশিষ্ট্য প্রদান করা, যেমন একই কেক্ষ ইউজারের সাথে গেম খেলা, একটি নেটওয়ার্ক করা এনএসডি সিকর্য-ওয়েবকন্সাম থেকে ইমেজ আনা, বা একই নেটওয়ার্কে অর্থাৎ মেশিনের মধ্যে লগ ইন করাতে সহায়তা করে।

এই ক্লাস আপনার অ্যাপলিকেশন থেকে অর্থাৎ ডিভাইস খুঁজে বের করা বা এর সাথে যুক্ত হওয়ার জন্য যে কিক APIs সে বিষয়ে আরো চান। বিবেশ করে, এটা সহজপরাপয় সািভর্স খুঁজে ডাওয়ার জন্য এনএসডি APIs আলাচনা করে এবং পিয়ার টু পিয়ার (P2P) ওয়ারেরস সংযোগ করার জন্য ওয়াই-ফাই পিয়ার টু পিয়ার (P2P) APIs নিয়েও আলাচনা করে। এই ক্লাস আরও আলাচনা করে একটি ডিভাইস কতরুপ পরস্তুতিবত সািভর্স চিহ্নিত করতে কীভাবে এনএসডি এবং ওয়াই-ফাই (P2P) একসাথে সনিবৃত্তভাবে ব্যবহার করা হয় এবং ডিভাইসের সাথে সংযোগ করতে হয় যখন কোন ডিভাইস নেটওয়ার্কে সাথে যুক্ত থাকেনা।

অনুশীলনীসমূহ

নেটওয়ার্কিং সার্ভিসেস ডিস্ট্রিবিউটরিং বয়স্ক বহুর

শিক্ষণ কীভাবে আপনার নিজস্ব অর্থোপলেকশন দব্ধারাপরস্ তাব করা
সার্ভিসেস সম্ভার করে ত হয়, লোকাল নেটওয়ার্কিং পরস্ তাবিত
সার্ভিসেস খুজে বের করতে হয় এবং আপনিয়ে সার্ভিসেসের সাথে যুক্ত হতে
চান তার জন্য স্ফুটিক্তর বিস্ তাবিত বিবরণ নিধর্ ারন করতে এনএসডি
বয়স্ক বহুর করা হয়।

ওয়াই-ফাই দিয়ে পিয়ার টু পিয়ার (P2P) স্ফুটোগ তৈরী

শিক্ষণ কীভাবে একটি নিকটস্থ পিয়ার ডিভাইস থেকে একটি লিস্ টপাওয়া
যায়, লিগয়্ াসি ডিভাইসের জন্য একটি একেসসপয়েন্ট তৈরি করা হয় এবং
ওয়াই-ফাই (P2P) স্ফুটোগ সামথর্ অনয্ ডিভাইসে স্ফুটোগ স্ তাপন করা হয়।

সার্ভিসেস ডিস্ট্রিবিউটরিংর জন্য ওয়াই-ফাই পিয়ার টু পিয়ার বয়স্ক বহুর

শিক্ষণ কীভাবে ওয়াই-ফাই পিয়ার টু পিয়ার বয়স্ক বহুর করে একই নেটওয়ার্ক
না হয়েই নিকটস্থ ডিভাইস দব্ধারাপর্ কাশিত সার্ভিসেস খুজে বের করতে
হয়।

নেটওয়ার্কিং সার্ভিসে ডেলিভারি ব্যবহার

(<http://developer.android.com/training/connect-devices-wirelessly/nsd.html>)

নেটওয়ার্কিং সার্ভিসে ডেলিভারি (এনএসডি) আপনার অ্যাপে যুক্ত করা আপনার ইউজারকে লোকাল নেটওয়ার্কে অন্য ডিভাইস চিহ্নিত করতে দেয় যা আপনার অ্যাপের পরিবেশে কয়েক সেকেন্ডের মধ্যে পৌঁছানোর ক্ষমতা দেয়। এটি বিবিধ নীতিমূলক ট্রাফিকের অ্যাপলিকেশনের জন্য উপকারী যেমন, ফাইল শেয়ার করা বা মাল্টি-প্ল্যাটফর্মের গেম খেলা। অ্যান্ড্রয়েডের এনএসডি এপিআই আপনাকে এই বৈশিষ্ট্য বাস্তবায়ন করার জন্য প্রয়োজনীয় পর্বে চেষ্টা করে দেয়।

এই অনুশীলনী আপনাকে দেখায় কীভাবে একটি অ্যাপলিকেশন তৈরি করা হয় যা এর নাম এবং স্যেগমেন্টের তথ্য লোকাল নেটওয়ার্কে সম্পর্কিত করে এবং একইভাবে অন্য অ্যাপলিকেশন থেকে তথ্যের জন্য অনুসন্ধান করে থাকে। চূড়ান্তভাবে, এই অনুশীলনী দেখায় কীভাবে অন্য ডিভাইসের সাথে একই অ্যাপলিকেশনের সাথে যোগাযোগ করতে হয়।

নেটওয়ার্কে আপনার সার্ভিস রেজিস্ট্রেশন করা

নোট: এই ধাপ ঐচ্ছিক। লোকাল নেটওয়ার্ক জুড়ে আপনার অর্থাৎ পের সার্ভিস সম্পর্কে চারিত্র্যহওয়া বিষয়ে আপনি তখন গুরুত্ব না দেন, আপনি পরবর্তীতে সেক্ষেত্রে যেতে পারেন, Discover Services on the Network।

লোকাল নেটওয়ার্কে আপনার সার্ভিস রেজিস্ট্রার করতে, পরেই একটি NsdServiceInfo অবজেক্ট তৈরি করুন। এই অবজেক্ট তথ্য প্রদান করে যাতে নেটওয়ার্কের অন্য ডিভাইস ব্যবহার করে যখন তার ঐচ্ছিক করে কখন আপনার সার্ভিসের সাথে সংযোগ স্থাপন করতে পারে।

```
public void registerService(int port) {  
    // Create the NsdServiceInfo object, and populate it.  
    NsdServiceInfo serviceInfo = new NsdServiceInfo();  
  
    // The name is subject to change based on conflicts  
    // with other services advertised on the same network.  
    serviceInfo.setServiceName("NsdChat");  
    serviceInfo.setServiceType("_http._tcp.");  
    serviceInfo.setPort(port);  
    ....  
}
```

এই কোড খন্ডচিত্রটি "NsdChat"এ সার্ভিস নাম সেট করে। নামটি নেটওয়ার্কে যে কোন ডিভাইসে দৃশ্যমান হয় যাকাল সার্ভিস খুঁজে এনএসডি ব্যবহার করে। মনে রাখবেন যে, নেটওয়ার্কে যে কোন সার্ভিসের জন্য নামটি অবশ্যই সর্বত্র হতে হবে এবং অর্থাৎ ডায়েরি সব যেকির যতাবে সংঘাত সমাধান পরিচালনা করে। যদি নেটওয়ার্ক দুইটি ডিভাইসই "NsdChat" অর্থাৎ পলিকেশন ইনস্টল করে থাকে, এদোমধ্যে একটি সর্ব যেকির যতাবে তার সার্ভিস নাম পরিবর্তন করে, যেমন হতে পারে "NsdChat (1)"।

দিবর্তীয় প্যারামিটার সার্ভিস টাইপ সেট করে, কোন পরেটাকল এবং ট্রান্সপোর্ট প্রোট্রলেয়ার অর্থাৎ পলিকেশন ব্যবহার করে তানিধাররণ করে। সিনট্যাক্স হচ্ছে ":"। কোড খন্ডাংশটিতে, সার্ভিসটি TCP ধরে চলমান HTTP পরেটাকল ব্যবহার করে। একটি অর্থাৎ পলিকেশন একটি ট্রান্সপোর্ট সার্ভিস সম্পর্কে তাবে করে (উদাহরণস্বরূপ, একটি নেটওয়ার্ক প্রিং টার) "_ipp._tcp"এ সার্ভিস টাইপ সেট করতে পারে।

নোট: দয় ইন্টারন্যাশনাল অর্থাৎ আইএনএনএমএর অথরিটি (IANA) সার্ভিস সিডেস্কাভারি পরেটাকল দবারাব্যবহৃত সার্ভিস টাইপের একটি কেন্দ্রীয়, নিভর্যযোগ্য তালিকা ব্যবস্থাপনা করে, যেমন এনএসডি এবং ইউএলডি। আপনি the IANA list of service names and port numbers (<http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml>) থেকে তালিকাটি ডাউনলোড করে নিতে পারেন। আপনি যদি একটি নতুন সার্ভিস টাইপ ব্যবহার করার পরিকল্পনা করে থাকেন, IANA Ports and Service registration form পূরণ করার মাধ্যমে আপনার এর মজুদ রাখা উচিত।

যখন আপনার সার্ভিসের জন্য প্রোট্র সেটিং করা হয়, এটাকে হাডরকে ডিই করা পরিহার করুন, যেহেতু এটা অন্য অর্থাৎ পলিকেশনের সাথে সংঘাত করে। উদাহরণস্বরূপ, ধরুন যে আপনার অর্থাৎ পলিকেশন সবসময় প্রোট্র ১৩৩৭ ব্যবহার করে যা একই প্রোট্র ব্যবহার করা অন্য ইনস্টল করা অর্থাৎ পলিকেশনকে সম্ভাব্য সংঘাতের দিকে ঠেলে দেয়। পরিবর্তে ডিভাইসের অন্য ানয় বিদ্যমান প্রোট্র ব্যবহার করুন। কারণ এই ইনফর্মেশন একটি সার্ভিস সম্পর্কে চারিত্র্যের মাধ্যমে অন্য অর্থাৎ পের সরবরাহ করা হয়, এখানে কম্পাইল সময়ে অন্য অর্থাৎ পলিকেশন কতরুর পরিচিত হওয়ার জন্য আপনার অর্থাৎ পলিকেশনের কোন প্রোট্র ব্যবহার

করার দরকার নেই। পরিবর্তে অ্যাপ্লিকেশনটি আপনার সার্ভিস বর্ডকাস্ট থেকে এই তথ্য পেতে পারে, আপনার সার্ভিসেস স্যুইচ হওয়ার ঠিক পূর্বে।

আপনি যদি কোন সেক্টর সাথে কাজ করতে থাকেন, এখানে আছে কীভাবে আপনি একটি সেক্টরে কোন বিদ্যমান পোর্টের শুরুর করতে পারেন শুরুর মাত্র এটাকে ০ তে সেটিং করার মাধ্যমে।

```
public void initializeServerSocket() {  
    // Initialize a server socket on the next available port.  
    mServerSocket = new ServerSocket(0);  
  
    // Store the chosen port.  
    mLocalPort = mServerSocket.getLocalPort();  
    ...  
}
```

এখন আপনি NsdServiceInfo অবজেক্ট নিধারণ করেছেন, আপনার RegistrationListener ইন্টারফেস বাস্তবায়ন করা পরে যোজন। এই ইন্টারফেস অ্যান্ডারয়েড দ্বারা ব্যবহৃত কলব্যাক ধারণ করে, আপনার অ্যাপ্লিকেশনকে সার্ভিসের রিজিস্ট্রেশন এবং আনরেজিস্ট্রেশনের সাফল্য বা ব্যর্থতার বিষয়ে সতর্ক করতে। public void initializeRegistrationListener() { mRegistrationListener = new NsdManager.RegistrationListener() {

```
@Override  
public void onServiceRegistered(NsdServiceInfo NsdServiceInfo) {  
    // Save the service name. Android may have changed it in order to  
    // resolve a conflict, so update the name you initially requested  
    // with the name Android actually used.  
    mServiceName = NsdServiceInfo.getServiceName();  
}  
  
@Override  
public void onRegistrationFailed(NsdServiceInfo serviceInfo, int errorCode) {  
    // Registration failed! Put debugging code here to determine why.  
}  
  
@Override  
public void onServiceUnregistered(NsdServiceInfo arg0) {  
    // Service has been unregistered. This only happens when you call  
    // NsdManager.unregisterService() and pass in this listener.  
}  
  
@Override  
public void onUnregistrationFailed(NsdServiceInfo serviceInfo, int errorCode) {  
    // Unregistration failed. Put debugging code here to determine why.  
}  
};  
}
```

এখন আপনার সার্ভিসের রিজিস্ট্রেশন করতে সকল অংশ আপনার আছে। registerService() পদ্ধতি কল করুন। উল্লেখ্য যে এই পদ্ধতিটি আসিঙ্ক্রোনাস, সুতরাং যে কোন ক্যাড যা সার্ভিসের রিজিস্ট্রেশন হওয়ার পর রান করা পরে যোজন তা অবশ্যই onServiceRegistered() পদ্ধতির মধ্যে যাওয়া উচিত।

```
public void registerService(int port) {  
    NsdServiceInfo serviceInfo = new NsdServiceInfo();  
    serviceInfo.setServiceName("NsdChat");  
    serviceInfo.setServiceType("_http_tcp.");  
    serviceInfo.setPort(port);  
  
    mNsdManager = Context.getSystemService(Context.NSD_SERVICE);  
  
    mNsdManager.registerService(  
        serviceInfo, NsdManager.PROTOCOL_DNS_SD, mRegistrationListener);  
}
```

নেটওয়ার্কিং সার্ভিস উইথ ম্যাচিং করা

নেটওয়ার্কিং জীবনে পরিপূর্ণ, নতুন নেটওয়ার্কিং পরিবর্তন আনছে।
শুরু করে নিম্নলিখিত নেটওয়ার্কিং ওয়েবকন্সাম পয়সে নতুন, নিম্নলিখিত, নিম্নলিখিত
টিক-ট্যাক-টেমপ্লেয়ার এর ভয়ংকর যুদ্ধ পয়সে নতুন। আপনার
অ্যাপলিকেশনকে এই জীবনে তৈরি করে সিস্টেমটি দেখতে দেওয়ার উপায়
হচ্ছে। সার্ভিস সিস্টেমের কাঠামো (খুঁজো বের করা)। কোন সার্ভিস
সহজপরাপে আছে তা দেখতে আপনার অ্যাপ্লিকেশনের নেটওয়ার্কিং
সার্ভিস সম্ভার (বর্ডকাস্ট) শোনা পরে যোজন এবং অ্যাপ্লিকেশন
যেটার সাথে কাজ করতে পারবে সেটা বেছে নেয়া।

সার্ভিস সিস্টেমের কাঠামো (খুঁজো বের করা), সার্ভিস সেরিজিস্ট্রির
মতো কোন ধাপ নেই: পরাসিঙ্গ কলব্যাকের সাথে একটি সিস্টেমের
লিসেনার সেট করা, এবং `discoverServices()` এর করা একটি একক আনুগত্যের
এপিআই তৈরি করা।

পর্যন্ত, একটি নামহীন ক্লাস শুরু করুন যা `NsdManager.DiscoveryListener`
বাস্তবায়ন করে। নিম্নলিখিত খন্ডাংশটি একটি সহজ উদাহরণ দেখাচ্ছে:

```

public void initializeDiscoveryListener() {

    // Instantiate a new DiscoveryListener
    mDiscoveryListener = new NsdManager.DiscoveryListener() {

        // Called as soon as service discovery begins.
        @Override
        public void onDiscoveryStarted(String regType) {
            Log.d(TAG, "Service discovery started");
        }

        @Override
        public void onServiceFound(NsdServiceInfo service) {
            // A service was found! Do something with it.
            Log.d(TAG, "Service discovery success" + service);
            if (!service.getServiceType().equals(SERVICE_TYPE)) {
                // Service type is the string containing the protocol and
                // transport layer for this service.
                Log.d(TAG, "Unknown Service Type: " + service.getServiceType());
            } else if (service.getServiceName().equals(mServiceName)) {
                // The name of the service tells the user what they'd be
                // connecting to. It could be "Bob's Chat App".
                Log.d(TAG, "Same machine: " + mServiceName);
            } else if (service.getServiceName().contains("NsdChat")){
                mNsdManager.resolveService(service, mResolveListener);
            }
        }

        @Override
        public void onServiceLost(NsdServiceInfo service) {
            // When the network service is no longer available.
            // Internal bookkeeping code goes here.
            Log.e(TAG, "service lost" + service);
        }

        @Override
        public void onDiscoveryStopped(String serviceType) {
            Log.i(TAG, "Discovery stopped: " + serviceType);
        }

        @Override
        public void onStartDiscoveryFailed(String serviceType, int errorCode) {
            Log.e(TAG, "Discovery failed: Error code:" + errorCode);
            mNsdManager.stopServiceDiscovery(this);
        }

        @Override
        public void onStopDiscoveryFailed(String serviceType, int errorCode) {
            Log.e(TAG, "Discovery failed: Error code:" + errorCode);
            mNsdManager.stopServiceDiscovery(this);
        }
    };
}

```

NSD API আপনার অয্যাপিলিকেশনের জ্ঞাতাথে থর্ এই ইন্টারফেস পদ্ধতিটি টবযবহার করে যখন ডিসকোভারি শুরু হয়, যখন এটা বয থর্ হয় এবং সািভর্ সপাওয়া যায় এবং হারিয়ে যায় (হারিয়ে যাওয়া মানে "এখন আর সহজপর্পায নয")। মনেরাখিবন এই খন্ডাংশটি কিকচুেচক করেনা যখন একটি সািভর্ সপাওয়া যায়।

1. খুেজপাওয়া সািভর্ সের সািভর্ সনাম লেকাল সািভর্ সের সািভর্ সনামের সােথ তুলনা করা হয়, এটা নিধর্ারন করতে যে যদি ডিভাইস শুধু তার নিজসব সমর্পর্চার (বর্ডকাস্ট) তুলে নেয়।
2. এটা এমন ধরেন সািভর্ সযার সােথ আপনার অয্যাপিরিকেশন স্যেযাগ সর্থাপন করতে পারে, তাই সািভর্ সটাইপেচক করা হয়।
3. সঠিক অয্যাপিরিকেশন স্যেযাগ যাচাই করতে সািবর্ সনামেচক করা হয়

সািভর্ সনামেচক করা সবসময় জরুরী নয এবং আপনি যদি কেকান নিদির্ স্ট অয্যাপিলিকেশনের সােথ যুক্ ত হেত চান তখনই শুধু এটা পর্সিঙ্গক।
উদাহরনসব্ রূপ, অয্যাপিলিকেশন অন্য ডিভাইসেরান করার সময় এর নিজসব

ইনস্টেটনস এর সাথে যুক্ত হতে চাইবে। কিন্ত, যদি অ্যাপলিকেশন একটি নেটওয়ার্ক কানেকশনের সাথে যুক্ত হতে চায় এটা দেখাই যে যে সার্ভিস টাইপ হচ্ছে "_ipp._tcp"।

লিসেনার সেটআপ করার পর, discoverServices() কল করুন, আপনার অ্যাপলিকেশনের যাকে কাজে উচিত সেই সার্ভিস টাইপের মধ্যে পাস করতে, ডিসকভারি পোর্টাকল ব্যবহার করতে এবং লিসেনার আপনি যামাতর্ তৈরী করেছেন।

```
mNsdManager.discoverServices(  
    SERVICE_TYPE, NsdManager.PROTOCOL_DNS_SD, mDiscoveryListener);
```


নেটওয়ার্কিং সার্ভিসে যুক্ত করুন

যখন আপনার অ্যাপ্লিকেশন নেটওয়ার্কিং একটি সার্ভিস সংযুক্ত করার
জন্য খুঁজে পায়, `resolveService()` পদ্ধতি ব্যবহার করে ঐ সার্ভিসের
কোনকোন তথ্য অবশ্যই পরামর্শ গ্রহণ করা উচিত। এই পদ্ধতির
মধ্যে পাস করতে একটি `NsdManager.ResolveListener` বাস্তবায়ন করুন এবং কোনকোন
তথ্য ধারণ করা একটি `NsdServiceInfo` পাওয়ার জন্য এটা ব্যবহার করুন।

```
public void initializeResolveListener() {
    mResolveListener = new NsdManager.ResolveListener() {

        @Override
        public void onResolveFailed(NsdServiceInfo serviceInfo, int errorCode) {
            // Called when the resolve fails. Use the error code to debug.
            Log.e(TAG, "Resolve failed" + errorCode);
        }

        @Override
        public void onServiceResolved(NsdServiceInfo serviceInfo) {
            Log.e(TAG, "Resolve Succeeded. " + serviceInfo);

            if (serviceInfo.getServiceName().equals(mServiceName)) {
                Log.d(TAG, "Same IP.");
                return;
            }
            mService = serviceInfo;
            int port = mService.getPort();
            InetAddress host = mService.getHost();
        }
    };
}
```

একসময় সার্ভিস স্থির হবে, আপনার অ্যাপ্লিকেশন সার্ভিস তথ্য
সম্পর্কে বিস্তারিত গ্রহণ করে ব্যার মধ্যে রয়েছে একটি আইপি
এড্রেস এবং পোর্ট নামবো। সার্ভিসে আপনার নিজেই নেটওয়ার্কিং
কোনকোন তথ্য ধারণ করে আপনাকে এই সব করতে হবে।

অয্যাপিলেকশন বন্ধ করার সময় আপনার সার্ভিস আনরেজিস্টার করুন

NSD এর কার্যকারিতা সিকর্য এবং নিস্কর্য করা গুরুত্বপূর্ণ অয্যাপিলেকশনের লাইফসাইকেলের সময়কালে যেভাবে উপযুক্ত হয়। আপনার অয্যাপিলেকশন আনরেজিস্টার করুন যখন এটা বন্ধ হবে, এটা অন্য অয্যাপিলেকশনের চিন্তা করা থেকে বিবর্ত রাখে সাহায্য করে যে এখনও এটা সিকর্য আছে এবং এটার সাথে সংযোগ স্থাপনের চেষ্টা করবে। এছাড়াও সার্ভিস ডিসকভারি একটি বয়্যবহুল অপারেশন হতে পারে এবং এটা থামা না উচিত হবে যখন পর্য্যন্ত একটি টিভিটিতে পজ দেওয়া হবে এবং সিকর্য করা হবে যখন একটি টিভি পূর্ণরায় শুরু হবে। আপনার পর্য্যধান একটি টিভিটির লাইফসাইকেল পদ্যতি ওভাররাইড করুন এবং সার্ভিস বর্ডকাস্ট এবং ডিসকভারি যেভাবে উপযুক্ত হয় স্টপ এবং স্টাট করতে কেড পরবেশ করান।

//In your application's Activity

```
@Override
protected void onPause() {
    if (mNsdHelper != null) {
        mNsdHelper.tearDown();
    }
    super.onPause();
}

@Override
protected void onResume() {
    super.onResume();
    if (mNsdHelper != null) {
        mNsdHelper.registerService(mConnection.getLocalPort());
        mNsdHelper.discoverServices();
    }
}

@Override
protected void onDestroy() {
    mNsdHelper.tearDown();
    mConnection.tearDown();
    super.onDestroy();
}

// NsdHelper's tearDown method
public void tearDown() {
    mNsdManager.unregisterService(mRegistrationListener);
    mNsdManager.stopServiceDiscovery(mDiscoveryListener);
}
```

ওয়াই-ফাই সহকারে P2P কানেকশন তৈরী করা

(<http://developer.android.com/training/connect-devices-wirelessly/wifi-direct.html>)

ওয়াই-ফাই পিয়ার টু পিয়ার (P2P) এপিআই একটি নতুন ওয়াকার্বাহটস্‌পেট যুক্ত না হয়েই অ্যাপলিকেশনকে নিকটস্থ ডিভাইসে যুক্ত হতে দেয় (অ্যান্ডরয়েডের ওয়াই-ফাই পিয়ার টু পিয়ার (P2P) ফর্মওয়ার্ক ওয়াকার্ব Wi-Fi Direct™ সার্টিফিকেশনের অনুবর্তী হয়)। ওয়াই-ফাই পিয়ার টু পিয়ার (P2P) আপনার অ্যাপলিকেশনকে ব্লুটুথ এর সাথে থর্রসীমার বাইরের নিকটবর্তী ডিভাইসকে দ্রুত খুঁজে বের করা বা এর সাথে যোগাযোগ করতে অনুমোদন করে।

এই অনুশীলনী আপনাকে দেখায় ওয়াই-ফাই পিয়ার টু পিয়ার (P2P) ব্যবহার করে কীভাবে নিকটবর্তী ডিভাইসকে খুঁজে বের করা বা এর সাথে যোগাযোগ করতে হয়।

অফলাইন পিয়ারিং প্যারামিটার সেটআপ করা

ওয়াই-ফাই পিয়ারিং টু পিয়ারিং (P2P) ব্যবহার করার জন্য, আপনার মনিফেস্টে `CHANGE_WIFI_STATE`, `ACCESS_WIFI_STATE` এবং `INTERNET` প্যারামিটার যুক্ত করুন। ওয়াই-ফাই পিয়ারিং টু পিয়ারিং (P2P) এর ইন্টারনেট সংযোগের পরয়োজন নেই, কিন্তু এটি স্ট্যান্ডার্ড জাভা সেক্টর ব্যবহার করতে পারে, যা `INTERNET` প্যারামিটার চায়। সুতরাং ওয়াই-ফাই পিয়ারিং টু পিয়ারিং (P2P) ব্যবহার করতে আপনার মনিফেস্টে নাক্ত প্যারামিটার পরয়োজন হবে।

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.android.nsdchat"
    ...

    <uses-permission
        android:required="true"
        android:name="android.permission.ACCESS_WIFI_STATE"/>
    <uses-permission
        android:required="true"
        android:name="android.permission.CHANGE_WIFI_STATE"/>
    <uses-permission
        android:required="true"
        android:name="android.permission.INTERNET"/>
    ...
```

একটি বর্ডকাস্ট রিসভার এবং ওয়াই-ফাই পিয়ার টু পিয়ার (P2P) ম্যানাজার সেটআপ করুন

ওয়াই-ফাই পিয়ার টু পিয়ার (P2P) ব্যবহার করতে, আপনার বর্ডকাস্ট ইনটেন্সিটি কেশানার চেষ্টা করুন যা আপনার অ্যাপ্লিকেশনকে বলবে যখন নির্দিষ্ট টিকিছু ইভেন্ট ঘটবে। আপনার অ্যাপ্লিকেশনকে, একটি `IntentFilter` শুরু করুন এবং নিচের বিষয়গুলো শুনতে এটা সেট করুন:

WIFI_P2P_STATE_CHANGED_ACTION

ওয়াই-ফাই পিয়ার টু পিয়ার (P2P) কনাতা নির্দেশ করে

WIFI_P2P_PEERS_CHANGED_ACTION

নির্দেশ করে যে বিদ্যমান পিয়ার লিস্ট পরিবর্তন করেছে

WIFI_P2P_CONNECTION_CHANGED_ACTION

ওয়াই-ফাই পিয়ার টু পিয়ার (P2P) কানেকটিভিটি পরিবর্তন করেছে তার অবস্থান নির্দেশ করে

WIFI_P2P_THIS_DEVICE_CHANGED_ACTION

এই ডিভাইসের কনফিগারেশন ডিটেইলস পরিবর্তন করেছে নির্দেশ করে

```
private final IntentFilter intentFilter = new IntentFilter();
...
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    // Indicates a change in the Wi-Fi P2P status.
    intentFilter.addAction(WifiP2pManager.WIFI_P2P_STATE_CHANGED_ACTION);

    // Indicates a change in the list of available peers.
    intentFilter.addAction(WifiP2pManager.WIFI_P2P_PEERS_CHANGED_ACTION);

    // Indicates the state of Wi-Fi P2P connectivity has changed.
    intentFilter.addAction(WifiP2pManager.WIFI_P2P_CONNECTION_CHANGED_ACTION);

    // Indicates this device's details have changed.
    intentFilter.addAction(WifiP2pManager.WIFI_P2P_THIS_DEVICE_CHANGED_ACTION);

    ...
}
```

`onCreate()` পদ্ধতিতে শেষ, একটি `WifiP2pManager` এর ইনস্ট্যান্স পাবেন, এবং এর `initialize()` পদ্ধতি কল করুন। এই পদ্ধতি একটি `WifiP2pManager.Channel` অবজেক্ট ফেরত দেয়, যা আপনি পরবর্তীতে ওয়াই-ফাই পিয়ার টু পিয়ার (P2P) ফর্মওয়ার্কের আপনার অ্যাপ যুক্ত করার সময় ব্যবহার করতে পারেন।

```
@Override

Channel mChannel;

public void onCreate(Bundle savedInstanceState) {
    ....
    mManager = (WifiP2pManager) getSystemService(Context.WIFI_P2P_SERVICE);
    mChannel = mManager.initialize(this, getMainLooper(), null);
}
```

এখন একটি নতুন `BroadcastReceiver` ক্লাস তৈরী করুন যা সিস্টেমের ওয়াই-ফাই পরিবার টুপি পরিবার (P2P) অবস্থার পরিবর্তন শুনতে ব্যবহার করতে পারবেন। `onReceive()` পদ্ধতির মধ্যে, উপের লিস্ট করা পর্টিটো (P2P) অবস্থা পরিবর্তনকে চালনা করতে একটি শতর্য় যুক্ত করুন।

```
@Override
public void onReceive(Context context, Intent intent) {
    String action = intent.getAction();
    if (WifiP2pManager.WIFI_P2P_STATE_CHANGED_ACTION.equals(action)) {
        // Determine if Wifi P2P mode is enabled or not, alert
        // the Activity.
        int state = intent.getIntExtra(WifiP2pManager.EXTRA_WIFI_STATE, -1);
        if (state == WifiP2pManager.WIFI_P2P_STATE_ENABLED) {
            activity.setIsWifiP2pEnabled(true);
        } else {
            activity.setIsWifiP2pEnabled(false);
        }
    } else if (WifiP2pManager.WIFI_P2P_PEERS_CHANGED_ACTION.equals(action)) {
        // The peer list has changed! We should probably do something about
        // that.

    } else if (WifiP2pManager.WIFI_P2P_CONNECTION_CHANGED_ACTION.equals(action)) {
        // Connection state changed! We should probably do something about
        // that.

    } else if (WifiP2pManager.WIFI_P2P_THIS_DEVICE_CHANGED_ACTION.equals(action)) {
        DeviceListFragment fragment = (DeviceListFragment) activity.getFragmentManager()
            .findFragmentById(R.id.frag_list);
        fragment.updateThisDevice((WifiP2pDevice) intent.getParcelableExtra(
            WifiP2pManager.EXTRA_WIFI_P2P_DEVICE));
    }
}
```

চুড়ান্তভাবে, ইন্টেন্টিফল্টার এবং বর্ডকাস্ট রিসিভার রেজিস্টার করতে কেড যুক্ত করুন যখন আপনার পর্ধান একটি টিভিটি সিকর্য় থাকে, এবং তােদর আনরেজিস্টার করুন যখন একটি টিভিটি পজ অবস্থায় থাকে। এটা করার সবচেয়ে ভালো স্থান হেচ্ছ `onResume()` এবং `onPause()` পদ্ধতি।

```
/** register the BroadcastReceiver with the intent values to be matched */
@Override
public void onResume() {
    super.onResume();
    receiver = new WifiDirectBroadcastReceiver(mManager, mChannel, this);
    registerReceiver(receiver, intentFilter);
}

@Override
public void onPause() {
    super.onPause();
    unregisterReceiver(receiver);
}
```

পিয়ার ডিসকভারী শুরু করুন

ওয়াই-ফাই পিয়ার টু পিয়ার (P2P) সহকারে নিকটস্থ ডিভাইস অনুসন্ধান শুরু করতে, `discoverPeers()` কল করুন। এই পদ্ধতি নিম্নোক্ত আলোচনা গর্হণ করে:

- যখন আপনি পিয়ার টু পিয়ার ম্যানেজার শুরু করে বন আপনি `WifiP2pManager.Channel` ফিরে পাবেন
- পদ্ধতিগুলো সহকারে `WifiP2pManager.ActionListener` এর একটি বাস্তবায়ন, সিস্টেমটি সফল এবং ব্যর্থতার ডিসকভারী আহবান করে।

```
mManager.discoverPeers(mChannel, new WifiP2pManager.ActionListener() {
```

```
    @Override
    public void onSuccess() {
        // Code for when the discovery initiation is successful goes here.
        // No services have actually been discovered yet, so this method
        // can often be left blank. Code for peer discovery goes in the
        // onReceive method, detailed below.
    }

    @Override
    public void onFailure(int reasonCode) {
        // Code for when the discovery initiation fails goes here.
        // Alert the user that something went wrong.
    }
}
```

```
});
```

মেন রাখবেন যে এটা শুধু পিয়ার ডিসকভারী শুরু করে। `discoverPeers()` পদ্ধতি ডিসকভারী পরিকল্পনা শুরু করে এবং তারপর তা ক্রমিকভাবে ফেরত দেয়। সিস্টেমটি আপনাকে জানায় যদি পিয়ার ডিসকভারী পরিকল্পনা পরদত্ত একশন লিসনারের মধ্যে পদ্ধতি কল করার মাধ্যমে সফলভাবে শুরু হয়। এছাড়াও, ডিসকভারী সিকর্য থাকে যতক্ষণ না একটি কানেকশন শুরু হয় বা একটি পিয়ার টু পিয়ার গরুপ শুরু হয়।

পিয়ার লিস্ট নিয়ে আসা

এখন কেডলি লিখুন যা পিয়ারের লিস্ট নিয়ে আসে এবং পরীক্ষা করে।
পর্যবেক্ষণে `WifiP2pManager.PeerListListener` ইন্টারফেস বাস্তবায়ন করুন, যা পিয়ার সম্বন্ধে তথ্য প্রদান করে যেতে ওয়াই-ফাই পিয়ার টু পিয়ার চিহ্নিত হয়। নিম্নোক্ত খন্ডাংশটি এটা সম্বন্ধে বিস্তারিত আলাচনা করে:

```
private List peers = new ArrayList();
...

private PeerListListener peerListListener = new PeerListListener() {
    @Override
    public void onPeersAvailable(WifiP2pDeviceList peerList) {

        // Out with the old, in with the new.
        peers.clear();
        peers.addAll(peerList.getDeviceList());

        // If an AdapterView is backed by this data, notify it
        // of the change. For instance, if you have a ListView of available
        // peers, trigger an update.
        ((WifiPeerListAdapter) getListAdapter()).notifyDataSetChanged();
        if (peers.size() == 0) {
            Log.d(WiFiDirectActivity.TAG, "No devices found");
            return;
        }
    }
}
```

এখন `requestPeers()` কল করতে আপনার বর্ডকাস্ট রিসিভারের `onReceive()` পদ্ধতি পরিবর্তন করুন যখন একশন `WIFI_P2P_PEERS_CHANGED_ACTION` সহকারে একটি ইন্টেন্ট রিসিভ করা হয়। আপনাকে যেভাবে বই হোক এই লিসেনারকে রিসিভারের মধ্যে পাস করে দিন। বর্ডকাস্ট রিসিভারের কনস্ট্রাক্টরে একটি আরগুমেন্ট হিসাবে এটাসেন্ড করার একটি উপায়।

```
public void onReceive(Context context, Intent intent) {
    ...
    else if (WifiP2pManager.WIFI_P2P_PEERS_CHANGED_ACTION.equals(action)) {

        // Request available peers from the wifi p2p manager. This is an
        // asynchronous call and the calling activity is notified with a
        // callback on PeerListListener.onPeersAvailable()
        if (mManager != null) {
            mManager.requestPeers(mChannel, peerListListener);
        }
        Log.d(WiFiDirectActivity.TAG, "P2P peers changed");
    }...
}
```

এখন, একশন `WIFI_P2P_PEERS_CHANGED_ACTION` ইন্টেন্ট সহকারে একটি ইন্টেন্ট আপডেট পিয়ার লিস্টের জন্য রিকোয়েস্ট চেক চালু করবে।

একটি টি পয় াের র স াথে স ংয াগ

একটি টি পয় াের স ংয াগ স্ থাপনের জনয, একটি নতুন WifiP2pConfig অবজেক্ট তৈরী করুন, এবং যে ডিভাইসে আপনি যুক্ত হতে চান তার পরিচিতি নিখতব করা WifiP2pDevice এর থেকে এর মধ্যে ডাটাকপি করুন।

```
@Override
public void connect() {
    // Picking the first device found on the network.
    WifiP2pDevice device = peers.get(0);

    WifiP2pConfig config = new WifiP2pConfig();
    config.deviceAddress = device.deviceAddress;
    config.wps.setup = WpsInfo.PBC;

    mManager.connect(mChannel, config, new ActionListener() {

        @Override
        public void onSuccess() {
            // WiFiDirectBroadcastReceiver will notify us. Ignore for now.
        }

        @Override
        public void onFailure(int reason) {
            Toast.makeText(WiFiDirectActivity.this, "Connect failed. Retry.",
                Toast.LENGTH_SHORT).show();
        }
    });
}
```

WifiP2pManager.ActionListener এই খন্ডাংশে বাস্ তব ায়িত হয় শুধুমাত্র আপনাকে জন ায় কখন শুরটাসফল বাবয্ থর্। ক ানেকশন অবস্ থার মধ্যে পরিবর্তন শ ানার জনয, WifiP2pManager.ConnectionInfoListener ইন্টারফেস বাস্ তব ায়ন করুন। এর onConnectionInfoAvailable() কলবয্ াক আপনাকে জ ান াবে কখন ক ানেকশনের অবস্ থা পরিবর্তন করে। একটি একক ডিভাইসে স ংযুক্ত হতে যাওয়া ম াল্ টপল ডিভাইসে সর কে ষ্ তে তর্ যেমন, একটি গে মিকন্ তু তিনজন প্ লয়ার বা একটি চয্ াট অয্ াপ) একটি ডিভাইস "ম ড ঙ্ ট ডি হব "হিস াবে বিবেচিত হবে।

```
@Override
public void onConnectionInfoAvailable(final WifiP2pInfo info) {

    // InetAddress from WifiP2pInfo struct.
    InetAddress groupOwnerAddress = info.groupOwnerAddress.getHostAddress();

    // After the group negotiation, we can determine the group owner.
    if (info.groupFormed && info.isGroupOwner) {
        // Do whatever tasks are specific to the group owner.
        // One common case is creating a server thread and accepting
        // incoming connections.
    } else if (info.groupFormed) {
        // The other device acts as the client. In this case,
        // you'll want to create a client thread that connects to the group
        // owner.
    }
}
```

এখন বর্ডকাস্ টিরিসভ াের onReceive() পদ্ ধতি তে তিফের য ান এবং সেকশন পরিবর্তন করুন যা একটি WIFI_P2P_CONNECTION_CHANGED_ACTION ইন্টেন্ট শূনে থাকে। যখন এই ইন্টেন্টিরিসভ করা হয়, requestConnectionInfo() কল করুন। এটা একটি স ঙ্গে কর ান াস কল, সুতরাং আপনি য াকে একটি পয্ ার ামিটার হিস াবে পর্দ ান করে ছেন সেই ক ানেকশন ইন্ফ ালিসেন ার দব্ ার া ফল াফলিরিসভ করা হয়।

```
...
} else if (WifiP2pManager.WIFI_P2P_CONNECTION_CHANGED_ACTION.equals(action)) {

    if (mManager == null) {
        return;
    }

    NetworkInfo networkInfo = (NetworkInfo) intent
        .getParcelableExtra(WifiP2pManager.EXTRA_NETWORK_INFO);

    if (networkInfo.isConnected()) {

        // We are connected with the other device, request connection
        // info to find group owner IP

        mManager.requestConnectionInfo(mChannel, connectionListener);
    }
    ...
}
```

সািভর্স ডিসকভারির জন্য ওয়াই-ফাই P2P ব্যবহার করা

(<http://developer.android.com/training/connect-devices-wirelessly/nsd-wifi-direct.html>)

এই ক্লাসের পূর্ববর্তী অনুশীলনীতে, Using Network Service Discovery, আপনাকে দেখায় কীভাবে সািভর্স খুঁজে বের করা যায় যা একটি লোকাল নেটওয়ার্কের যুক্ত। যাহোক, ওয়াই-ফাই-ফাই পিয়ার টু পিয়ার সািভর্স ডিসকভারী ব্যবহার করা আপনাকে নিকটস্থ ডিসকভারীর সািভর্স সরাসরি খুঁজে বের করতে দেয়, কোন নেটওয়ার্কের যুক্ত না হয়েই। আপনার ডিভাইসে রান করা সািভর্সের পরিচারণাও আপনি করতে পারেন। এই সামগ্র্যগুলি আপনাকে অ্যাপের মধ্যে যোগাযোগ সহায়তাকরে, এমনকি যখন কোন লোকাল নেটওয়ার্ক বা হটস্পট না থাকে।

যখন APIs এর এই সেট নেটওয়ার্কের সািভর্স ডিসকভারী APIs যার রূপের খা পূর্ববর্তী অনুশীলনীতে দেখা হয়েছে তার যদি উদ্দেশ্য একই হয়, তাদেরকে ডেবাস্‌তবায়ন করা খুব ভিন্ন। এই অনুশীলনী আপনাকে দেখায় ওয়াই-ফাই-ফাই পিয়ার টু পিয়ার ব্যবহার করে অনয ডিভাইস থেকে বিদ্যমান সািভর্স কীভাবে খুঁজে ত হয়। এই অনুশীলনী ধারণাকরে আপনি পূর্ববর্ত থেকেই সম্ভব করে Wi-Fi-P2P API জানেন।

মেমিনেফেস্টে স্টআপ করুন

ওয়াই-ফাই পিয়ার টু পিয়ার (P2P) ব্যবহার করার জন্য আপনার মেমিনেফেস্টে CHANGE_WIFI_STATE, ACCESS_WIFI_STATE এবং INTERNET যুক্ত করুন। যদিও ওয়াই-ফাই পিয়ার টু পিয়ার (P2P) ইন্টারনেট কানেকশন চায় না, এটা স্ট্যান্ডার্ড জাভা সেক্টর ব্যবহার করে, এবং অ্যান্ডরয়েড এই সব ব্যবহার করা রিকোয়েস্ট করা পারমিশন চায়।

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.android.nsdchat"
    ...

    <uses-permission
        android:required="true"
        android:name="android.permission.ACCESS_WIFI_STATE"/>
    <uses-permission
        android:required="true"
        android:name="android.permission.CHANGE_WIFI_STATE"/>
    <uses-permission
        android:required="true"
        android:name="android.permission.INTERNET"/>
    ...
```

লোকাল সার্ভিস যুক্ত করুন

যদি আপনি একটি লোকাল সার্ভিস সরবরাহ করতে থাকেন, আপনার এটা সার্ভিস ডিসকভারির জন্য রেজিস্টার করতে হবে। একসময় আপনার লোকাল সার্ভিস রেজিস্টার হেয়ে থাকে, পিয়ার থেকে সার্ভিস ডিসকভারী রিকোয়েস্ট এ ফের্ম ওয়াকর্স সর্ব্বক্ষণিক র্যভাবে রেসপন্স করবে।

একটি লোকাল সার্ভিস তৈরী করতে:

1. একটি WifiP2pServiceInfo অবজেক্ট তৈরী করুন
2. আপনার সার্ভিস সম্পর্কিত তথ্য এটার সাথে পর্চািরিত করুন
3. সার্ভিস ডিসকভারির জন্য লোকাল সার্ভিস রেজিস্টার করতে addLocalService() কল করুন

```
private void startRegistration() {
    // Create a string map containing information about your service.
    Map record = new HashMap();
    record.put("listenport", String.valueOf(SERVER_PORT));
    record.put("buddyname", "John Doe" + (int) (Math.random() * 1000));
    record.put("available", "visible");

    // Service information. Pass it an instance name, service type
    // _protocol._transportlayer , and the map containing
    // information other devices will want once they connect to this one.
    WifiP2pDnsSdServiceInfo serviceInfo =
        WifiP2pDnsSdServiceInfo.newInstance("_test", "_presence._tcp", record);

    // Add the local service, sending the service info, network channel,
    // and listener that will be used to indicate success or failure of
    // the request.
    mManager.addLocalService(channel, serviceInfo, new ActionListener() {
        @Override
        public void onSuccess() {
            // Command successful! Code isn't necessarily needed here,
            // Unless you want to update the UI or add logging statements.
        }

        @Override
        public void onFailure(int arg0) {
            // Command failed. Check for P2P_UNSUPPORTED, ERROR, or BUSY
        }
    });
}
```

নিচক টস্ থ সাি ভর্ স উদয াটন (ি ড স ক ভ ার)

অয্ ান্ ডর্ য়েড সহজপর্ াপয্ সাি ভর্ স সম্ পর্ কর্ আপন ার
অয্ াপ লি ক শ ন কৈ অব ি হ ত ক র তৈ ক ল ব য্ াক প দ্ ধ তি ব য্ ব হ ার ক রৈ, সু ত র া ং
এ ট া ক র তৈ প র্ থ মে আপন া কৈ ঐ গ ল্ ল া সে স ট আপ ক র া। ইন ক া মি ং রেকডর্ শুন তৈ
এক টি WifiP2pManager.DnsSdTxtRecordListener তৈ ত রী ক রুন।

অয্ ান্ ডর্ য়েড আপন ার অয্ াপ লি ক শ ন কৈ বি দ য্ ম ান সাি ভর্ সের বি ষ য়ে
জ ান া তৈ ক ল ব য্ াক মে থ ড ব য্ ব হ ার ক রৈ, সু ত র া ং প র্ থ ম ক া জ হ চ্ ছ ঐ
গ ল্ ল া সে স ট আপ ক র া। ইন ক া মি ং রেকডর্ শুন তৈ এক টি
WifiP2pManager.DnsSdTxtRecordListener তৈ ত রী ক রুন। এই রেকডর্ ঐ চ্ ছ ক ভ া বে অ ন য্
ি ড ভ া ই স ক তি ত্ র ক ব র্ ড ক া স্ ট হ তৈ প া রৈ। যখন এক জন ি ভ তৈ রে আে স, ি ড ভ া ই স
এ ড র্ স এ ব ং চলি ত প দ্ ধ তি র ব া ই রের এক টি ড া ট া ক া ঠ া মে য া তৈ আপ নি য়ে
কৈ ান প র্ স ি ঙ্ গ ক ত থ য্ চ ান ত া ক পি ক রৈ, তাই আপ নি এ ট া তৈ প রৈ প র্ ব শ
ক র তৈ প া রৈ ব ন। নি মে ম্ ন া ক ত্ উ দ া হ রণ মৈ ন ক রৈ য়ে রেকডর্ এক টি "buddyname"
ি ফ ল্ ড ধ া রন ক রৈ, ইউ জ া রের আই ডে ন ি টি টি র সাে থ থ া কৈ।

```
final HashMap<String, String> buddies = new HashMap<String, String>();
...
private void discoverService() {
    DnsSdTxtRecordListener txtListener = new DnsSdTxtRecordListener() {
        @Override
        /* Callback includes:
         * fullDomain: full domain name: e.g "printer._ipp._tcp.local."
         * record: TXT record data as a map of key/value pairs.
         * device: The device running the advertised service.
         */

        public void onDnsSdTxtRecordAvailable(
            String fullDomain, Map record, WifiP2pDevice device) {
            Log.d(TAG, "DnsSdTxtRecord available -" + record.toString());
            buddies.put(device.deviceAddress, record.get("buddyname"));
        }
    };
    ...
}
```

সাি ভর্ স ইন ফৈ ম র্ শ ন পৈ ত, এক টি WifiP2pManager.DnsSdServiceResponseListener তৈ ত রী
ক রুন। এ ট া স ি ঠ ক ি ব ব রণ এ ব ং ক া নৈ ক শ ন ত থ য্ গ র্ হ ণ ক রৈ। পূ ব র্ ব ত র া
কৈ া ড অ ং শ ি টি এক টি ি ড ভ া ই স এ ড র্ স এ ব ং ব া ড ি নৈ ম এক ক র তৈ এক টি Map
অ বৈ জ ক্ ট ব া স্ ত ব া য় ন ক রৈ। সাি ভর্ সের স প ন্ স লি সৈ ন ার সমৈ গ া তি র্ য
সাি ভর্ স তৈ থ য্ র সাে থ DNS রেকডর্ লি ং ক ক র তৈ এ ট া ব য্ ব হ ার ক রৈ। যখন ই
উ ভ য় লি সৈ ন ার ব া স্ ত ব া য় ত হৈ ব, setDnsSdResponseListeners() প দ্ ধ তি ব য্ ব হ ার
ক রৈ তাে দর WifiP2pManager এ য় ক ত ক রুন।

```

private void discoverService() {
    ...

    DnsSdServiceResponseListener servListener = new DnsSdServiceResponseListener() {
        @Override
        public void onDnsSdServiceAvailable(String instanceName, String registrationType,
            WifiP2pDevice resourceType) {

            // Update the device name with the human-friendly version from
            // the DnsTxtRecord, assuming one arrived.
            resourceType.deviceName = buddies
                .containsKey(resourceType.deviceAddress) ? buddies
                .get(resourceType.deviceAddress) : resourceType.deviceName;

            // Add to the custom adapter defined specifically for showing
            // wifi devices.
            WifiDirectServicesList fragment = (WifiDirectServicesList) getFragmentManager()
                .findFragmentByld(R.id.frag_peerlist);
            WifiDevicesAdapter adapter = ((WifiDevicesAdapter) fragment
                .getListAdapter());

            adapter.add(resourceType);
            adapter.notifyDataSetChanged();
            Log.d(TAG, "onBonjourServiceAvailable " + instanceName);
        }
    };

    mManager.setDnsSdResponseListeners(channel, servListener, txtListener);
    ...
}

```

এখন একটি সাঁতিভর্স সিরে কাঁয়ে স্ টৈ তরী করুন এবং addServiceRequest() কল করুন। এই পদ্ধতিও সফলতা বা বাবর্ থর্ তাঁর পোঁটর্ করে ত একটি টিলে সনর্ নর্ থে থাঁক।

```

serviceRequest = WifiP2pDnsSdServiceRequest.newInstance();
mManager.addServiceRequest(channel,
    serviceRequest,
    new ActionListener() {
        @Override
        public void onSuccess() {
            // Success!
        }

        @Override
        public void onFailure(int code) {
            // Command failed. Check for P2P_UNSUPPORTED, ERROR, or BUSY
        }
    });

```

চুড়ান্ তভাঁব, discoverServices() এ কল করুন।

```

mManager.discoverServices(channel, new ActionListener() {

    @Override
    public void onSuccess() {
        // Success!
    }

    @Override
    public void onFailure(int code) {
        // Command failed. Check for P2P_UNSUPPORTED, ERROR, or BUSY
        if (code == WifiP2pManager.P2P_UNSUPPORTED) {
            Log.d(TAG, "P2P isn't supported on this device.");
        } else if (...)
            ...
    }
});

```

যদি সব িকছু ভাঁলে লাভাঁব হয়, আপনাঁক অভিনন্ দন আপনি এটাঁ করে তে পেরেঁ ছন। আর আপনি যদি কেঁান সমসয্ ার মুখে ামুখি হন, মনে রাঁখে বন যে

আপনি যে আসিঙে কর্ ান াস কল তৈতরী করেছেন তা একটি আরগুমেন্ট হিসাবে একটি WifiP2pManager.ActionListener নেয় এবং এটা আপনাকে কলব্যাক দিয়ে নিশ্চিত সফলতা বা ব্যর্থতা পরদান করবে। পদ্ধতি দ্বারা পরদত্ত এরকে ডপর্বে লমসম্পেকর্ অভাস দেয়। এখানে কিছু সম্ভাব্য এরর দেয়া হলে এবং তার কাবে বায় তা দেয়া হলে

P2P_UNSUPPORTED

ওয়াই-ফাই পিয়ার টু পিয়ার অর্থাপ রান করা ডিভাইস সাপোর্ট করেনা।
সিস্টেমটি রিকোয়েস্ট পরেস

BUSY

সিস্টেমটি ব্যাস্ত আছে রিকোয়েস্ট পরেস করার মতো যথেষ্ট সময় নেই

ERROR

একটি অ্যাক্শন এরর কারণে অপারেশন ব্যর্থ হয়েছে।

নেটওয়ার্ক অপারেশন সম্পাদন করা

(<http://developer.android.com/training/basics/network-ops/index.html>)

এই ক্লাস নেটওয়ার্কে সঙ্ঘুক্ত হওয়ার জন্য যে মৌলিক কাজগুলো রয়েছে সে সম্পর্কে বর্ণনা করা, নেটওয়ার্ক কানেকশন মনিটরিং করে (এর মধ্যে কানেকশন পরিবর্তন রয়েছে) এবং ইউজারকে একটি অ্যাপের নেটওয়ার্ক বয়বহারের উপর নিয়ন্ত্রণ প্রদান বিষয়ে আলোচনা করে। এটা এক্সএমএল ডাটাকীভাবে পাসর্ (parse) করা এবং বয়বহার করা হয়।

এই ক্লাস একটি নমুনা অ্যাপ্লিকেশন অন্তর্ভুক্ত করেছে যা সাধারণ নেটওয়ার্ক অপারেশন সম্পাদন করার বিষয়টা আলোচনা করে। আপনি নমুনাটি ডাউনলোড করতে পারেন (ডান পাশে দেয়া আছে) এবং এটা আপনার নিজের অ্যাপ্লিকেশনের জন্য একটি প্রদর্শন বয়বহারে যাগক্ষেড়ে সেসর্ হিসাবে বয়বহার করতে পারেন।

এই লেসনগুলোর মধ্যে গিয়ে আপনি অ্যাপ্লিকেশন তৈরী করার জন্য মৌলিক বিল্ডিং ব্লক পাবেন যা কায়র্ করীভাবে কন্টেন্ট এবং পাসর্ ডাটাদাউনলোড করে, যখন নেটওয়ার্ক টর্ফিফিকিমিনিমাইজ করা হয়।

অনুশীলনী সমূহ

নেটওয়ার্কিং যুক্ত হওয়া

শিখন কীভাবে নেটওয়ার্কিং যুক্ত হতে হয়, একটি HTTP ক্লায়েন্ট পছন্দ করতে হয়, এবং নেটওয়ার্কিং অপারেশন ইউজার ইন্টারফেসের কাছাকাছি সম্পাদন করতে হয়।

নেটওয়ার্কিং ব্যবহার ব্যবস্থাপনা

শিখন কীভাবে একটি ডিভাইসে নেটওয়ার্কিং সংযোগ চেক করতে হয়, নেটওয়ার্কিং ব্যবহার নিয়ন্ত্রণ করার জন্য একটি পিপিফরেন্স ইউজার ইন্টারফেস তৈরি করতে হয় এবং সংযোগ পরিবর্তন রেসপন্স করতে হয়।

এক্সএমএল ডাটাপার্স (parse) করা

শিখন কীভাবে এক্সএমএল ডাটাকীভাবে পার্স (parse) করা এবং ব্যবহার করা হয়

নেটওয়ার্কিং যুক্ত হওয়া

(<http://developer.android.com/training/basics/network-ops/connecting.html>)

এই অনুশীলনী আপনাকে দেখায় একটি সহজ অ্যাপ্লিকেশন কীভাবে বাস্তবায়ন করতে হয় যা নেটওয়ার্কিং যুক্ত করে। এটাকে ছোট কৃষ্ণ টি চর্চা হিসেবে শ্রেণিভুক্ত করে যা আপনার এমনকি একটি সহজ নেটওয়ার্কিং যুক্ত অ্যাপ টৈত্রী করার সময়ও অনুসরণ করা উচিত।

মনে রাখবেন যে এই অনুশীলনীতে আলোচনা করা নেটওয়ার্কিং অপারেশন সম্পাদন করতে, আপনার অ্যাপ্লিকেশন মিনিফেস্টের অবশ্যই নিম্নলিখিত পারমিশনগুলো অন্তর্ভুক্ত করতে হবে:

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

একটি HTTP ক্লায়েন্ট নিবর্াচন

অধিকাংশ নেটওয়ার্কিং-যুক্ত অ্যান্ড্রয়েড অ্যাপস ডাটাসেন্ড এবং
রিসিভ করতে HTTP ব্যবহার করে। অ্যান্ড্রয়েড দুইটা HTTP ক্লায়েন্ট
অন্তর্ভুক্ত করে: `URLConnection` এবং `HttpClient` অ্যাপাচে (Apache)। উভয়ই HTTPS,
স্ট্রিমিং আপলোড এবং ডাউনলোড, কনফিগারবল টাইমআউট, IPv6, এবং
কানেকশন পুলিং সাপোর্ট করে। আমরা জিঞ্জারবের্ড বা এর উপরের
সংস্করণের পরিতলক্ষ্য করে তৈরী করা অ্যাপলিকেশনের জন্য
`URLConnection` সুপারিশ করি। এই এরও আলোচনার জন্য, এই ব্লগে পাস্টি
Android's HTTP Clients দেখুন।

নেটওয়ার্ক কানেকশন চেক করুন

আপনার অ্যাপ নেটওয়ার্কের যুক্ত হওয়ার জন্য চেষ্টা করার পূর্বে, `getActiveNetworkInfo()` এবং `isConnected()` ব্যবহার করে এটার চেক করে দেখা উচিত যে কীভাবে একটি নেটওয়ার্ক কানেকশন সহজপরিাপ্য আছে। মনে রাখবেন ডিভাইসটি একটি নেটওয়ার্কের বাইরে থাকতে পারে, অথবা ইউজার ওয়াই-ফাই এবং মোবাইল ডাটাকে সসদুইটাই নিস্কর্য করে রাখতে পারে। এই বিষয়ে আরও আলোচনার জন্য Managing Network Usage অনুশীলনীটি দেখুন।

```
public void myClickHandler(View view) {  
    ...  
    ConnectivityManager connMgr = (ConnectivityManager)  
        getSystemService(Context.CONNECTIVITY_SERVICE);  
    NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();  
    if (networkInfo != null && networkInfo.isConnected()) {  
        // fetch data  
    } else {  
        // display error  
    }  
    ...  
}
```

একটি পৃথক থ্রেডে নেটওয়ার্ক অপারেশন সম্পাদন করুন

নেটওয়ার্ক অপারেশন অননুমেয় বিলম্ব সম্পৃক্ত করতে পারে। একটি দূর্বল ইউজার এক্সপেরিয়েন্স হওয়া থেকে বিরত রাখতে, ইউজার ইন্টারফেস থেকে একটি পৃথক থ্রেডে সব সময় নেটওয়ার্ক অপারেশন পারফর্ম করতে হয়। AsyncTask ক্লাস ইউজার ইন্টারফেস থেকে একটি নতুন কাজ বন্দ করার একটি অনন্তম সহজ উপায় প্রদান করে। এ বিষয়ে আরও আলোচনার জন্য এই Multithreading For Performance ব্লগ পোস্টটি দেখুন। (<http://android-developers.blogspot.com/2010/07/multithreading-for-performance.html>)

নিচের চিত্রটিতে, myClickHandler() পদ্ধতি new DownloadWebpageTask().execute(stringUrl) কে আহবান করে। DownloadWebpageTask ক্লাস AsyncTask এর একটি সাব-ক্লাস। DownloadWebpageTask নিচের AsyncTask পদ্ধতিগুলো বাস্তবায়ন করে:

- doInBackground() পদ্ধতি downloadUrl() টি সম্পাদন করে। এটা ওয়েব পেজ URL কে একটি প্যারামিটার হিসাবে পাস করে। পদ্ধতি downloadUrl() টি ওয়েব পেজ কন্টেন্ট ধারণ করে এবং প্রতিক্রিয়া জাত করে। যখন এটা শেষ করে, এটা একটি রেজাল্ট হিস্টরি ফেরত পাঠায়।
- onPostExecute() ফিরতি হিস্টরি টি গ্রহণ করে এবং ইউজার ইন্টারফেসে এটা প্রদর্শন করে।

```

public class HttpExampleActivity extends Activity {
    private static final String DEBUG_TAG = "HttpExample";
    private EditText urlText;
    private TextView textView;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        urlText = (EditText) findViewById(R.id.myUrl);
        textView = (TextView) findViewById(R.id.myText);
    }

    // When user clicks button, calls AsyncTask.
    // Before attempting to fetch the URL, makes sure that there is a network connection.
    public void myClickHandler(View view) {
        // Gets the URL from the UI's text field.
        String urlString = urlText.getText().toString();
        ConnectivityManager connMgr = (ConnectivityManager)
            getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();
        if (networkInfo != null && networkInfo.isConnected()) {
            new DownloadWebpageTask().execute(stringUrl);
        } else {
            textView.setText("No network connection available.");
        }
    }

    // Uses AsyncTask to create a task away from the main UI thread. This task takes a
    // URL string and uses it to create an HttpURLConnection. Once the connection
    // has been established, the AsyncTask downloads the contents of the webpage as
    // an InputStream. Finally, the InputStream is converted into a string, which is
    // displayed in the UI by the AsyncTask's onPostExecute method.
    private class DownloadWebpageTask extends AsyncTask<String, Void, String> {
        @Override
        protected String doInBackground(String... urls) {

            // params comes from the execute() call: params[0] is the url.
            try {
                return downloadUrl(urls[0]);
            } catch (IOException e) {
                return "Unable to retrieve web page. URL may be invalid.";
            }
        }
        // onPostExecute displays the results of the AsyncTask.
        @Override
        protected void onPostExecute(String result) {
            textView.setText(result);
        }
    }
    ...
}

```

এই চিত্রটির ইন্টেন্টিভ কর্তৃক মিনিচের বিষয়গুলোর মতো:

1. যখন ইউজার বাটনে ক্লিক করে যা `myClickHandler()` কে আহবান করে, অর্থাৎ এটি `AsyncTask` এর সাবক্লাস `DownloadWebpageTask` তৈরি করে এবং URL টি পাস করে।
2. `AsyncTask` পদ্ধতি `doInBackground()`, `downloadUrl()` পদ্ধতিতে কল করে।
3. The `downloadUrl()` পদ্ধতি একটি পয়ারামিটার হিসাবে একটি URL স্ট্রিং গ্রহণ করে এবং এটা একটি URL অবজেক্ট তৈরি করতে ব্যবহার করে।
4. URL অবজেক্ট একটি `HttpURLConnection` পরীক্ষা করতে ব্যবহৃত হয়।
5. যখন ইকোনেকশন পরীক্ষা হতে, `HttpURLConnection` অবজেক্ট একটি `InputStream` হিসাবে ওয়েব পেজ কন্টেন্ট ধারণ করে।
6. `InputStream readIt()` পদ্ধতিতে পাস হয়, যা স্ট্রিং মিকে একটি স্ট্রিং এ পরিবর্তন করে।
7. `AsyncTask` এর `onPostExecute()` পদ্ধতি প্রধান একটি ভিউটির ইউজার ইন্টারফেস

িস্ টর্ অ্ টি পর্ দর্শর্ ন ক়ে।

ডাটা সংযোগ এবং ডাউনলোড

আপনার থের্ডে আপনার নেটওয়ার্ক লেনদেন করতে, আপনার ডাটা ডাউনলোড বা একটি এউএসম্পাদন করতে আপনি `URLConnection` ব্যবহার করতে পারেন। `Connect()` কল করার পর, `getInputStream()` কল করার মাধ্যমে আপনি ডাটাতির একটি `InputStream` পেতে পারেন।

নিম্নোক্ত চিত্রটিতে, `doInBackground()` পদ্ধতিটি পদ্ধতি `downloadUrl()` কল করে। `downloadUrl()` পদ্ধতি পরদত্ত URL গ্রহণ করে এবং `URLConnection` হয়ে নেটওয়ার্কের সঙ্গে যুক্ত হতে এটাব্যবহার করে থাকে। যখনই একটি কানেকশন প্রতিষ্ঠিত হয় অর্থাৎ একটি `InputStream` হিসাবে ডাটা পূরণ হুদধার করতে পদ্ধতি `getInputStream()` ব্যবহার করে।

```
// Given a URL, establishes an HttpURLConnection and retrieves
// the web page content as a InputStream, which it returns as
// a string.
private String downloadUrl(String myurl) throws IOException {
    InputStream is = null;
    // Only display the first 500 characters of the retrieved
    // web page content.
    int len = 500;

    try {
        URL url = new URL(myurl);
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setReadTimeout(10000 /* milliseconds */);
        conn.setConnectTimeout(15000 /* milliseconds */);
        conn.setRequestMethod("GET");
        conn.setDoInput(true);
        // Starts the query
        conn.connect();
        int response = conn.getResponseCode();
        Log.d(DEBUG_TAG, "The response is: " + response);
        is = conn.getInputStream();

        // Convert the InputStream into a string
        String contentAsString = readIt(is, len);
        return contentAsString;

        // Makes sure that the InputStream is closed after the app is
        // finished using it.
    } finally {
        if (is != null) {
            is.close();
        }
    }
}
```

উল্লেখ্য যে, পদ্ধতি `getResponseCode()` কানেকশনের status code ফেরত দেয়। এটা কানেকশন সম্পর্কে অতিরিক্ত তথ্য পেতে এটা একটি কায়করী পথ। ২০০ এর স্ট্যাটাস কোড সফলতা নির্দেশ করে।

ইনপুট স্ট্রিম কে একটি স্ট্রিং এ পরিবর্তন (কনভার্ট)

একটি `InputStream` বাইটের (bytes) একটি পঠনযোগ্য স্রোত। যখনই আপনি একটি `InputStream` পান, এটা একটি সর্বাভাবিক ঘটনাকে এটাকে একটি বৃহৎ আকারের ডাটাস্ট্রিম হিসেবে ডিকোড বা পরিবর্তন (কনভার্ট) করা হয়। উদাহরণস্বরূপ, আপনি যদি ইমেজ ডাটাস্ট্রিম লোড করতে থাকেন, আপনি সম্ভবত এটার মতো করে ডিকোড বা পর্দাশরন করবেন:

```
InputStream is = null;
...
Bitmap bitmap = BitmapFactory.decodeStream(is);
ImageView imageView = (ImageView) findViewById(R.id.image_view);
imageView.setImageBitmap(bitmap);
```

উপরে আর্কট উদাহরণ দেখায়, `InputStream` একটি ওয়েব পেজের টেক্সটকে পর্দাশরন করে। এভাবেই উদাহরণটি `InputStream` কে একটি স্ট্রিং এ পরিবর্তন করে যাতে একটি ভিউটি ইউজার ইন্টারফেসের মধ্যে এটা পর্দাশরন করতে পারে।

```
// Reads an InputStream and converts it to a String.
public String readIt(InputStream stream, int len) throws IOException, UnsupportedEncodingException {
    Reader reader = null;
    reader = new InputStreamReader(stream, "UTF-8");
    char[] buffer = new char[len];
    reader.read(buffer);
    return new String(buffer);
}
```

নেটওয়ার্কিং বয় ব্যবহার বয় বস্ থাপনা

(<http://developer.android.com/training/basics/network-ops/managing.html>)

এই অনুশীলনটি আলোচনা করে কীভাবে অ্যাপ্লিকেশন লিখতে (রাইট করতে হয়) হয় যার নেটওয়ার্কিং রিসোর্স বয় ব্যবহারের একটি চমককার নিয়ন্ত্রণ আছে। যদি আপনার অ্যাপ্লিকেশন পর্চুর পরিমাণে নেটওয়ার্ক অপারেশন সম্পাদন করে থাকে, আপনার ইউজার সেটিংস পর্দান করা উচিত যা ইউজারকে আপনার অ্যাপের ডাটা অভ্যাস (হ্যাণ্ডেলিং) কে নিয়ন্ত্রণ করতে দেয়, যেমন, কীভাবে আপনার অ্যাপ ডাটা সঙ্কলিত, যখন ওয়াই-ফাই থাকা শুল্কমাত্র তখন আপলোড/ডাউনলোড করতে, রেয়াসিমি এর সময় ডাটা বয় ব্যবহার করতে এবং অন্যান্য কাজ করতে। তাদের কাছে এই নিয়ন্ত্রণগুলো সহজপাওয়া করার মাধ্যমে ইউজার বয় একগুণে উন্নত ডাটাতে আপনার অ্যাপের পর্বে বশগম্যতা অনেক কম নিশ্চয় করে, কারণ কতটুকু ডাটা আপনার অ্যাপ বয় ব্যবহার করে তার পরিবর্তে সঠিকভাবে তা নিয়ন্ত্রণ করে।

কীভাবে অ্যাপ রাইট করা যায় যা ডাউনলোড এবং নেটওয়ার্কিং কানেকশনের উপরে বয় টারিফের উপরে পর্ভাব কমিয়ে আনবে সে বিষয়ে সাধারণ গাইডলাইনের জন্য, [Optimizing Battery Life](#) এবং [Transferring Data Without Draining the Battery](#) দেখুন।

একটি টিডিভাইসে সরেনেটওয়ার্ক কানেকশন চেক করা

একটি টিডিভাইসে সরেনেটওয়ার্ক কানেকশন টাইপ থাকে। এই অনুষঙ্গীলন নীহয় ওয়াই-ফাই বা মোবাইল কানেকশনের উপর ফোকাস করে। সম্ভাব্য নেটওয়ার্কের ধরনের পূর্ণ তালিকার জন্য, `ConnectivityManager` দেখুন।

ওয়াই-ফাই সাধারণভাবে গতিশীল। মোবাইল ডাটাও মাঝে মাঝে পরিমাপ হয় যাব্য যবহুল হতে পারে। অতএব পরজন্য একটি সাধারণ কৌশল হচ্ছে শূন্য মাত্র বড় ডাটা ধারণ করা যদি একটি ওয়াই-ফাই থেকে থাকে।

নেটওয়ার্ক অপারেশন সম্পাদন করার পূর্বে, নেটওয়ার্ক কানেক্টিভিটির অবস্থান চেক করা একটি ভালো চর্চা। অন্যমনস্ক বিস্তারিত মধ্য এটা আপনার অ্যাপকে অসচেতনভাবে ভুল রেডিও ব্যবহার করতে থাকে। যদি একটি নেটওয়ার্ক কানেকশন না থাকে, আপনার অ্যাপ লিকশনের উচিত সাবলিলভাবে রেসপন্স করা। নেটওয়ার্ক কানেকশন চেক করতে, আপনি সাধারণভাবে নীচের ক্লাসগুলো ব্যবহার করতে পারেন।

- **ConnectivityManager:** নেটওয়ার্ক কানেক্টিভিটির অবস্থান সম্পর্কে অনুষঙ্গীলনের উত্তর দেয়। এটা অ্যাপ লিকশনকে জানানায়খন নেটওয়ার্ক কানেকশন পরিবর্তন করা হয়।
- **NetworkInfo:** একটি পর্দাত টাইপের (বর্তমানে হয় মোবাইল বা ওয়াই-ফাই) একটি নেটওয়ার্ক ইন্টারফেসের অবস্থান আলাচনা করে।

এই চিত্রটি ওয়াই-ফাই এবং মোবাইলের জন্য নেটওয়ার্ক কানেক্টিভিটি পরীক্ষা করে। এটা মনিস্ক্রিনের এই নেটওয়ার্ক ইন্টারফেস আছে কিনা (এটা হচ্ছে, নেটওয়ার্ক কানেক্টিভিটি সম্ভব কিনা) এবং/বা এটা সযুক্ত আছে কিনা (এটা হচ্ছে নেটওয়ার্ক কানেক্টিভিটি বিদ্যমান আছে কিনা এবং যদি সেক্ষেত্রে তথ্য প্রদান করা সম্ভব হয় এবং ডাটা পাস করা):

```
private static final String DEBUG_TAG = "NetworkStatusExample";
...
ConnectivityManager connMgr = (ConnectivityManager)
    getSystemService(Context.CONNECTIVITY_SERVICE);
NetworkInfo networkInfo = connMgr.getNetworkInfo(ConnectivityManager.TYPE_WIFI);
boolean isWifiConn = networkInfo.isConnected();
networkInfo = connMgr.getNetworkInfo(ConnectivityManager.TYPE_MOBILE);
boolean isMobileConn = networkInfo.isConnected();
Log.d(DEBUG_TAG, "Wifi connected: " + isWifiConn);
Log.d(DEBUG_TAG, "Mobile connected: " + isMobileConn);
```

উল্লেখ্য যে একটি নেটওয়ার্ক আছে কিনা তার উপর ভিত্তি করে আপনার সিদ্ধান্ত তৈরি করা উচিত নয়। নেটওয়ার্ক অপারেশন সম্পাদন করার পূর্বে আপনার উচিত সবসময় `isConnected()` চেক করা, যেহেতু `isConnected()` ফলস্বরূপ মোবাইল নেটওয়ার্ক, এয়ারপ্লেন মোড এবং সিমিত ব্যাকগ্রাউন্ড ডাটা চালিত করে।

একটি নেটওয়ার্ক ইন্টারফেস আছে কিনা তা চেক করতে নিম্নোক্ত চিত্রটির মতো আরও সঞ্চিপ্ত পথ। পদ্ধতি `getActiveNetworkInfo()` একটি `NetworkInfo` ইনস্ট্যান্স প্রথম খুঁজে পাওয়া যায় পর্দা পরিবর্তিত করণ নেটওয়ার্ক ইন্টারফেসের তথ্য, বা আর যদি কোন ইন্টারফেস খুঁজে পাওয়া না যায় তাহলে হঠাৎ ফেরত দেয় (বাঝায় যে ইন্টারনেট সংযোগ পাওয়া যাচ্ছে না):

```
public boolean isOnline() {  
    ConnectivityManager connMgr = (ConnectivityManager)  
        getSystemService(Context.CONNECTIVITY_SERVICE);  
    NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();  
    return (networkInfo != null && networkInfo.isConnected());  
}
```

আরও সুন্দর অবস্থা অনুসন্ধান করতে আপনি `NetworkInfo.DetailedState` ব্যবহার করতে পারেন, কিন্ত তার খুব কমই পর্যালোচনা ঘটে।

নেটওয়ার্ক বয় বহার বয় বস্ থাপনা

আপনি একটি পিওরফোন্স একটি টিভি ডিভাইস তবায়ন করতে পারেন যা আপনার অ্যাঞ্পেরনেটওয়ার্কেরিসেসোর্সের উপরে ইউজারকে পরিষ্কার নিয়ন্ত্রণ দিয়ে থাকে। উদাহরণস্বরূপ:

- আপনি ইউজারকে তখনই একটি টিভি ডিভাইস আপলোড করতে দিবেন যখন ডিভাইস একটি ওয়াই-ফাইয়ের সাথে যুক্ত থাকে।
- কিছু নির্দিষ্ট বৈশিষ্ট্য যেমন নেটওয়ার্কের উপস্থিতি/বিদ্যমানতা, সময়ের বিবরণী (ইন্টারভাল) এবং অন্য সব আপনি নিয়ন্ত্রণ করতে পারেন।

একটি অ্যাপ লিখতে যাঁনেটওয়ার্ক পরবেশগম্যতা এবং নেটওয়ার্ক বয় বহার বয় বস্ থাপনা সাপোর্ট করে আপনাকে মনিটরিং স্টাবশ্যই স্থিতি পরিমাপ এবং ইন্টেন্ট ফিল্টার থাকতে হবে।

- মনিটরিং স্ট্যান্ডেমনাক্ত পরিমাপনগুলো আর আকাঙ্ক্ষাকরে:
 - `android.permission.INTERNET`—অ্যাপ লিকে শনেকেনেটওয়ার্ক সেক্টপে নকরতে দেয়
 - `android.permission.ACCESS_NETWORK_STATE`— অ্যাপ লিকে শনেকেনেটওয়ার্ক সম্বন্ধে তথ্যের মধ্যে পরবেশ করতে দেয়।
- `ACTION_MANAGE_NETWORK_USAGE` একশনের জন্য আপনি ইন্টেন্ট ফিল্টার ডিফল্ট করার করতে পারেন নির্দেশ করতে যে আপনার অ্যাপ লিকে শন একটি একটি টিভি ডিভাইস নির্ধারণ করেছে যা ডাটাবে বয় বহার নিয়ন্ত্রণ করার পর স্তাবে দেয়। `ACTION_MANAGE_NETWORK_USAGE` একটি নির্দিষ্ট অ্যাপ লিকে শনের নেটওয়ার্ক ডাটাবে বয় বস্ থাপনাকরার জন্য সেটিং দেখায়। যখন আপনার অ্যাঞ্পের একটি সেটিং একটি টিভি ডিভাইস ইউজারকে নেটওয়ার্ক বয় বহার নিয়ন্ত্রণ করতে দেয়, আপনার উচিত এ একটি টিভি ডিভাইসের জন্য এই ইন্টেন্ট ফিল্টার ডিফল্ট করার। নমুনা অ্যাপ লিকে শনের মধ্যে এই একশন ক্লাস `SettingsActivity` কতর্কচালিত হয়ে থাকে, যা কখন একটি ফিডব্যাক লোড করতে হবে তা ইউজারকে দিয়ে ঠিক করতে একটি পিওরফোন্স ইউজার ইন্টারফেস পরদর্শন করে।

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.android.networkusage"
    ...>

    <uses-sdk android:minSdkVersion="4"
        android:targetSdkVersion="14" />

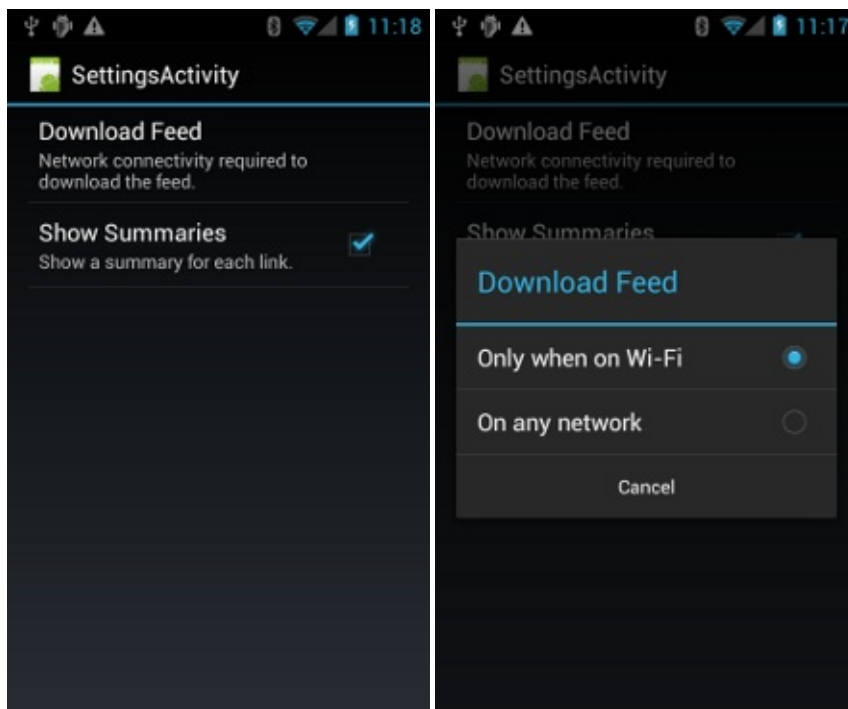
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

    <application
        ...>
        ...
        <activity android:label="SettingsActivity" android:name=".SettingsActivity">
            <intent-filter>
                <action android:name="android.intent.action.MANAGE_NETWORK_USAGE" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

একটি টিপি রফা রেন্স একটি টিভি টি বাস্তবায়ন

যেহেতু আপনি উপরে একটি টেমিনফেস্ট্র বনরনা দেতে পারেন, নমুনা অফ্রাে পর একটি টিভি টি SettingsActivity এর ACTION_MANAGE_NETWORK_USAGE একশনের জন্য একটি ইনটেন্ট আছে। SettingsActivity হচ্ছে PreferenceActivity এর একটি সাব-ক্লাস। এটা একটি টিপি রফা রেন্স সিস্করন পরদশরন করে (ফিগার ১ এ দেখানো হয়েছে) যা ইউজারকে নীচের বিষয়গুলো নিদিষ্ট করতে দেয়:

- প্রতিটা XML ফিড এন্ট্রির জন্য সারসংক্ষেপ পরদশরন করতে হবে কিনা অথবা প্রতিটা এন্ট্রির জন্য শুধু একটি লিংক।
- যদি কোন নেটওয়ার্ক কানেকশন থেকে থাকে XML ফিড ডাউনলোড করতে হবে কিনা অথবা যদি শুধুমাত্র ওয়াই-ফাই বিদ্যমান থেকে থাকে।



ফিগার ১. টিপি রফা রেন্স একটি টিভি টি এটা হচ্ছে SettingsActivity। উল্লেখ্য যে এটা OnSharedPreferenceChangeListener বাস্তবায়ন করে। যখন একজন ইউজার একটি টিপি রফা রেন্স পরিবর্তন করেন, এটা onSharedPreferenceChanged() বন্ধ করে দেয়, যেটা refreshDisplay সেট করে। এটা রিফ্রেশ করার জন্য পরদশরন করার কারণ ঘটায় যখন ইউজার পর্দান একটি টিভি টিতে ফিরে আসবেন: public class SettingsActivity extends PreferenceActivity implements OnSharedPreferenceChangeListener {

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    // Loads the XML preferences file
    addPreferencesFromResource(R.xml.preferences);
}

@Override
protected void onResume() {
    super.onResume();

    // Registers a listener whenever a key changes
    getPreferenceScreen().getSharedPreferences().registerOnSharedPreferenceChangeListener(this);
}

@Override
protected void onPause() {
    super.onPause();

    // Unregisters the listener set in onResume().
    // It's best practice to unregister listeners when your app isn't using them to cut down on
    // unnecessary system overhead. You do this in onPause().
    getPreferenceScreen().getSharedPreferences().unregisterOnSharedPreferenceChangeListener(this);
}

// When the user changes the preferences selection,
// onSharedPreferenceChanged() restarts the main activity as a new
// task. Sets the refreshDisplay flag to "true" to indicate that
// the main activity should update its display.
// The main activity queries the PreferenceManager to get the latest settings.

@Override
public void onSharedPreferenceChanged(SharedPreferences sharedPreferences, String key) {
    // Sets refreshDisplay to true so that when the user returns to the main
    // activity, the display refreshes to reflect the new settings.
    NetworkActivity.refreshDisplay = true;
}
}

```


পরিবর্তন সম্পর্কে কিভাবে জানতে পারবেন

যখন ইউজার সেটিংস পরিবর্তন করেন, এটাতে সাধারণভাবে অ্যাপের আচরণের জন্য ফলাফল থাকে। এই চিত্রটিতে অ্যাপ `onStart()` এর মধ্যে পরিবর্তন সেটিংস চেক করে থাকে। যদি সেখানে সেটিংস এবং ডিভাইসের স্টেটাসের কানেকশন এর মধ্যে মিল যায় (উদাহরণস্বরূপ, যদি সেটিংস "Wi-Fi" হয় এবং ডিভাইসের একটি Wi-Fi কানেকশন থাকে), অ্যাপটি ফিডব্যা্ক দেবে এবং ডিসপ্লে পরদা পরিবর্তন করে।

```
public class NetworkActivity extends Activity {
    public static final String WIFI = "Wi-Fi";
    public static final String ANY = "Any";
    private static final String URL = "http://stackoverflow.com/feeds/tag?tagNames=android&sort=newest";

    // Whether there is a Wi-Fi connection.
    private static boolean wifiConnected = false;
    // Whether there is a mobile connection.
    private static boolean mobileConnected = false;
    // Whether the display should be refreshed.
    public static boolean refreshDisplay = true;

    // The user's current network preference setting.
    public static String sPref = null;

    // The BroadcastReceiver that tracks network connectivity changes.
    private NetworkReceiver receiver = new NetworkReceiver();

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // Registers BroadcastReceiver to track network connection changes.
        IntentFilter filter = new IntentFilter(ConnectivityManager.CONNECTIVITY_ACTION);
        receiver = new NetworkReceiver();
        this.registerReceiver(receiver, filter);
    }

    @Override
    public void onDestroy() {
        super.onDestroy();
        // Unregisters BroadcastReceiver when app is destroyed.
        if (receiver != null) {
            this.unregisterReceiver(receiver);
        }
    }

    // Refreshes the display if the network connection and the
    // pref settings allow it.

    @Override
    public void onStart () {
        super.onStart();

        // Gets the user's network preference settings
        SharedPreferences sharedPrefs = PreferenceManager.getDefaultSharedPreferences(this);

        // Retrieves a string value for the preferences. The second parameter
        // is the default value to use if a preference value is not found.
        sPref = sharedPrefs.getString("listPref", "Wi-Fi");

        updateConnectedFlags();

        if(refreshDisplay){
            loadPage();
        }
    }

    // Checks the network connection and sets the wifiConnected and mobileConnected
    // variables accordingly.
    public void updateConnectedFlags() {
        ConnectivityManager connMgr = (ConnectivityManager)
            getSystemService(Context.CONNECTIVITY_SERVICE);
```

```
NetworkInfo activeInfo = connMgr.getActiveNetworkInfo();
if (activeInfo != null && activeInfo.isConnected()) {
    wifiConnected = activeInfo.getType() == ConnectivityManager.TYPE_WIFI;
    mobileConnected = activeInfo.getType() == ConnectivityManager.TYPE_MOBILE;
} else {
    wifiConnected = false;
    mobileConnected = false;
}
}

// Uses AsyncTask subclass to download the XML feed from stackoverflow.com.
public void loadPage() {
    if (((sPref.equals(ANY)) && (wifiConnected || mobileConnected))
        || ((sPref.equals(WIFI)) && (wifiConnected))) {
        // AsyncTask subclass
        new DownloadXmlTask().execute(URL);
    } else {
        showErrorPage();
    }
}
...
}
```

কানেকশন পরিবর্তন চিহ্নিত করা

পাঞ্জেলর চুড়ানত অংশ হচ্ছে BroadcastReceiver সাবক্লাস NetworkReceiver। যখন ডিভাইসে সর্নেটওয়াকর্ কানেকশন পরিবর্তন করে, NetworkReceiver একশন CONNECTIVITY_ACTION কে থামিয়ে দেয়, নিখরারণ করে দেয় যে সর্নেটওয়াকর্ কানেকশনের স্ট্যাটাসের অবস্থানিক, এবং ফ্ল্যাগস wifiConnected এবং mobileConnected তদানুসারে ট্রু/ফলস এসেট করে দেয়। পিরিনতি হচ্ছে যে পরবর্তী সময়ে ইউজার অ্যাপিফের আসবে, অ্যাপ শুধুমাত্র সর্বশেষ ফিডব্যাউনলেড করবে এবং ডিসপেপ্লআপে ডিট করে দেয় যদি NetworkActivity.refreshDisplay ট্রু তে সেট হয়।

একটি বর্ডকাস্টিরিসিভার (BroadcastReceiver) সেটআপ করা যা অপর্যেয়াজনীযভাবে সিসেস্টমিরেসোর্সের একটি ডেরইনকল করা হতে পারে। নমুনায় অ্যাপিলিকেশন onCreate() এর মধ্যে BroadcastReceiver NetworkReceiver রেজিস্টার করে, এবং onDestroy() এর মধ্যে এটা আনরেজিস্টার করতে পারেন। মেনিফেস্টের মধ্যে একটি <receiver> ডিকে ক্লয়ার করার চেয়েও এটা আরও অধিক লাইটওয়েট। যখন আপনি মেনিফেস্টে একটি <receiver> ডিকে ক্লয়ার করেন, এটা যে কোন সময় আপনার অ্যাপকে জাগিয়ে তুলতে পারে, এটাও হতে পারে আপনি এটা কে গত এক সপ্তাহ চালু করেননি। পরধান একটি ভিভিটর মধ্যে NetworkReceiver রেজিস্টার এবং আনরেজিস্টার করতে, আপনি নিশ্চিত করুন যে ইউজার অ্যাপ ত্যাগ করার পর অ্যাপটি জেগে উঠবে না। আপনি যদি মেনিফেস্টে একটি <receiver> ডিকে ক্লয়ার করেন এবং আপনি সঠিকভাবে জানেন যে এটা কে থায় পর্যেয়াজন, আপনি যথাযথ কাজ চিহ্নিত এবং এটা কে সিকর্য এবং নিস্কর্য করতে setComponentEnabledSetting() ব্যবহার করতে পারেন।

এটা হচ্ছে NetworkReceiver:

```
public class NetworkReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        ConnectivityManager conn = (ConnectivityManager)
            context.getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo networkInfo = conn.getActiveNetworkInfo();

        // Checks the user prefs and the network connection. Based on the result, decides whether
        // to refresh the display or keep the current display.
        // If the userpref is Wi-Fi only, checks to see if the device has a Wi-Fi connection.
        if (WIFI.equals(sPref) && networkInfo != null && networkInfo.getType() == ConnectivityManager.TYPE_WIFI) {
            // If device has its Wi-Fi connection, sets refreshDisplay
            // to true. This causes the display to be refreshed when the user
            // returns to the app.
            refreshDisplay = true;
            Toast.makeText(context, R.string.wifi_connected, Toast.LENGTH_SHORT).show();

            // If the setting is ANY network and there is a network connection
            // (which by process of elimination would be mobile), sets refreshDisplay to true.
        } else if (ANY.equals(sPref) && networkInfo != null) {
            refreshDisplay = true;

            // Otherwise, the app can't download content--either because there is no network
            // connection (mobile or Wi-Fi), or because the pref setting is WIFI, and there
            // is no Wi-Fi connection.
            // Sets refreshDisplay to false.
        } else {
            refreshDisplay = false;
            Toast.makeText(context, R.string.lost_connection, Toast.LENGTH_SHORT).show();
        }
    }
}
```

XML ডাটাপারিসংকরা

(<http://developer.android.com/training/basics/network-ops/xml.html>)

এক্সটেনসিবল মার্কআপ ল্যাঙ্গুয়েজ (XML) হচ্ছে মেশিন-রিডেবল ফর্মের ডকুমেন্ট এনেকোডিং করার জন্য এক্সটেনসিবল। XML হচ্ছে ইন্টারনেটে ডাটাবেশ করার একটি জনপ্রিয় ফরম্যাট। ওয়েবসাইট বা পরিত্যক্ত তথ্যের কন্টেন্ট আপডেট করে, যেমন কোন নিউজ সাইট বা ব্লগ, মাঝে মাঝে একটি XML ফিড সরবরাহ করে যাতে বাইরের পেরাগামগুলো পোশাপাশি কন্টেন্ট পরিবর্তন ধারণ করতে পারে। ডাটাপা আপলোড করা বা পাসর্করা নেটওয়ার্কে যুক্ত হওয়া অথবা পাসর্করা সাধারণ কাজ। এই অনুশীলনী ব্যাখ্যা করে কীভাবে XML ডকুমেন্ট পাসর্ক করতে হয় এবং তথ্য ডাটাবে ব্যবহার করতে হয়।

একটি পাসার্নার নিবর্নাচন করুন

আমরা XmlPullParser সুপারিশ করি, যা অ্যান্ডরয়েডে XML পাসার্ন করার একটি কাযর্নরী এবং তত্ববাবধানের উপযোগী পথ। ঐতিহাসিকভাবে অ্যান্ডরয়েডের এই ইন্টারফেসের দুইটা বাস্তুবায়ন আছে:

- KXmlParser via XmlPullParserFactory.newPullParser().
- ExpatPullParser, via Xml.newPullParser().

উভয় নিবর্নাচনই ঠিক আছে। এই অধ্যায়ে উদাহরণ Xml.newPullParser() হয়ে ExpatPullParser ব্যবহার করে।

ফিড বিবেশ লষণ

একটি ফিড পাসর্ করার পরে থম ধাপ হচ্ছে আপনিকে কান ফিল্ড কাজ করতে আগর্ হী তা ঠিক করা। পাসর্ ার ঐ ফিল্ড থেকে ডাটা নিয়ে থাকে এবং বাকীগুলোকে এড়িয়ে যায়।

এখানে ফিড থেকে একটি সংগ্রহ যা একটি নমুনায় অর্পণ থেকে পাসর্ করা হয়েছে। StackOverflow.com তে পরে তটাপে পাস্ট একটি entry হিসাবে ফিড দৃশ্যমান হয় যা বেশ কিছু বিবদ্যমান টয়োগসে ধারণ করে:

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom" xmlns:creativecommons="http://backend.userland.com/creativecommonsRssModule" ...
<title type="text">newest questions tagged android - Stack Overflow</title>
...
  <entry>
    ...
  </entry>
  <entry>
    <id>http://stackoverflow.com/q/9439999</id>
    <re:rank scheme="http://stackoverflow.com">0</re:rank>
    <title type="text">Where is my data file?</title>
    <category scheme="http://stackoverflow.com/feeds/tag?tagnames=android&sort=newest/tags" term="android"/>
    <category scheme="http://stackoverflow.com/feeds/tag?tagnames=android&sort=newest/tags" term="file"/>
    <author>
      <name>cliff2310</name>
      <uri>http://stackoverflow.com/users/1128925</uri>
    </author>
    <link rel="alternate" href="http://stackoverflow.com/questions/9439999/where-is-my-data-file" />
    <published>2012-02-25T00:30:54Z</published>
    <updated>2012-02-25T00:30:54Z</updated>
    <summary type="html">
      <p>I have an Application that requires a data file...</p>
    </summary>
  </entry>
  <entry>
    ...
  </entry>
  ...
</feed>
```

নমুনায় অর্পণ entry টয়োগ থেকে ডাটা সংগ্রহ করে এবং এটা title, link এবং summary টয়োগে রেখে দেয়

পাসওয়ার্ড ইনসেটনিসেট করা

পরবর্তী ধাপ হচ্ছে একটি পাসওয়ার্ড ইনসেটনিসেট করা এবং পারসিং পরিকল্পনা কে ছেড়ে দেয়া। এই চিত্রটিতে, নমুনা পাসওয়ার্ড ইনসেট করার জন্য এবং এর ইনপুট হিসাবে সরবরাহকৃত `InputStream` ব্যবহার করে একটি পাসওয়ার্ড চালু করা হয়েছে। এটা `nextTag()` একটি কল করার মধ্য দিয়ে পারসিং পরিকল্পনা শুরু করে, এবং `readFeed()` পদ্ধতিতে আহবান করে যা অ্যাপ যে ডাটার পরিত্যাগ গ্রহণে সেই ডাটানিয়ে আসে এবং পরিকল্পনা করে:

```
public class StackOverflowXmlParser {
    // We don't use namespaces
    private static final String ns = null;

    public List parse(InputStream in) throws XmlPullParserException, IOException {
        try {
            XmlPullParser parser = Xml.newPullParser();
            parser.setFeature(XmlPullParser.FEATURE_PROCESS_NAMESPACES, false);
            parser.setInput(in, null);
            parser.nextTag();
            return readFeed(parser);
        } finally {
            in.close();
        }
    }
    ...
}
```

ফিড পাঠ করা

`readFeed()` পদ্ধতি ফিড পরে সস করার সঠিক কাজটি করে। এটা রিকোর্ডের ভিত্তিতে ফিড পরীক্ষা করা করার পরে আরম্ভক পয়েন্ট হিসাবে উপাদান ট্যাগ করা "entry" এর অনুসন্ধান করে। একটি ট্যাগ যদি entry ট্যাগ না হয়, এটা সেটাকে ছেড়ে চলে যায়। যখনই সম্পূর্ণ ফিড রিকোর্ডের ভিত্তিতে পরীক্ষা করা হবে, `readFeed()` এন্ট্রি (নেস্টেড ডাটাবেস মব্বার সহ) ধারণ করা একটি List ফেরত দেয় যেগুলো সে ফিড থেকে গৃহণ করেছিল। এই তালিকা তারপর ডারসার দ্বারা ফেরত আসে।

```
private List readFeed(XmlPullParser parser) throws XmlPullParserException, IOException {
    List entries = new ArrayList();

    parser.require(XmlPullParser.START_TAG, ns, "feed");
    while (parser.next() != XmlPullParser.END_TAG) {
        if (parser.getEventType() != XmlPullParser.START_TAG) {
            continue;
        }
        String name = parser.getName();
        // Starts by looking for the entry tag
        if (name.equals("entry")) {
            entries.add(readEntry(parser));
        } else {
            skip(parser);
        }
    }
    return entries;
}
```


XML পাসর্ করা

XML পাসর্ করার ধাপগুলো নিম্নোক্ত:

১. যেভাবে **Analyze the Feed** এ আলোচনা করা হয়েছে, আপনি আপনার অ্যাপে যে ট্যাগ অন্তর্ভুক্ত করতে চান তা চিহ্নিত করুন। এই উদাহরণে **entry** ট্যাগের জন্য ডাটাতুলে আনতে এবং **title, link, এবং summary** ট্যাগে এগুলো রেখে দেয়।

২. নিম্নোক্ত পদ্ধতিগুলো তৈরি করুন:

- যে ট্যাগগুলোর প্রতি আপনি আগ্রহী তার প্রতিটির জন্য একটি "read" পদ্ধতি। উদাহরণস্বরূপ, `readEntry()`, `readTitle()` এবং অন্যান্য। পাসর্ করার ইনপুট সিস্টরমে থেকে ট্যাগ রিড করে। যখন এটা `entry`, `title`, `link` বা `summary` নামে একটি ট্যাগের মুঠো খামুঠি হয়, এটা এই ট্যাগের জন্য উপযোগী পদ্ধতি কল করে। অন্যথায় এটা ট্যাগটিকে ছেড়ে দিয়ে চলে যায়।
- প্রতিটি ভিন্ন ধরনের ট্যাগের জন্য ডাটাতুলে আনার জন্য এবং পাসর্ করার কৈরবতরী ট্যাগে এগিয়ে নিয়ে যাওয়ার জন্য পদ্ধতি তৈরি করা:
 - `Title` এবং `Summary` ট্যাগগুলোর জন্য পাসর্ করার `readText()` কল করে। এই পদ্ধতি `parser.getText()` কল করার মাধ্যমে এই ট্যাগগুলোর জন্য ডাটাতুলে আনে।
 - `link` ট্যাগের জন্য, পাসর্ করার লিংকের জন্য ডাটাতুলে আনে, পরে যেহি এটা নির্ধারণ করে নেয় যে লিংকটা সেই ধরনের যার প্রতি এটার আগ্রহ আছে। তারপর এটা লিংকের ভ্যালু নিয়ে আসতে `parser.getAttributeValue()` ব্যবহার করে।
 - `entry tag` ট্যাগের জন্য, পাসর্ করার `readEntry()` কল করে। এই পদ্ধতি এন্ট্রির নেস্টেড ট্যাগ পাসর্ করে এবং ডাটাতোমব্বার `title`, `link`, এবং `summary` সহ একটি `Entry` অবজেক্ট ফেরত পাঠায়।
- একটি হেল্পার (সহায়তাকারী) `skip()` পদ্ধতি যারিকারিসর্ভ। এই বিষয়ে আলোচনার জন্য `Skip Tags You Don't Care About` দেখুন।

এই চিত্রটিতে দেখানো হচ্ছে কীভাবে পাসর্ করার `entries`, `titles`, `links`, এবং `summaries` কৈ পাসর্ করে।

```
public static class Entry {
    public final String title;
    public final String link;
    public final String summary;

    private Entry(String title, String summary, String link) {
        this.title = title;
        this.summary = summary;
        this.link = link;
    }
}

// Parses the contents of an entry. If it encounters a title, summary, or link tag, hands them off
// to their respective "read" methods for processing. Otherwise, skips the tag.
private Entry readEntry(XmlPullParser parser) throws XmlPullParserException, IOException {
    parser.require(XmlPullParser.START_TAG, ns, "entry");
    String title = null;
    String summary = null;
    String link = null;
    while (parser.next() != XmlPullParser.END_TAG) {
        if (parser.getEventType() != XmlPullParser.START_TAG) {
            continue;
        }
        String name = parser.getName();
```

```

        if (name.equals("title")) {
            title = readTitle(parser);
        } else if (name.equals("summary")) {
            summary = readSummary(parser);
        } else if (name.equals("link")) {
            link = readLink(parser);
        } else {
            skip(parser);
        }
    }
}
return new Entry(title, summary, link);
}

// Processes title tags in the feed.
private String readTitle(XmlPullParser parser) throws IOException, XmlPullParserException {
    parser.require(XmlPullParser.START_TAG, ns, "title");
    String title = readText(parser);
    parser.require(XmlPullParser.END_TAG, ns, "title");
    return title;
}

// Processes link tags in the feed.
private String readLink(XmlPullParser parser) throws IOException, XmlPullParserException {
    String link = "";
    parser.require(XmlPullParser.START_TAG, ns, "link");
    String tag = parser.getName();
    String relType = parser.getAttributeValue(null, "rel");
    if (tag.equals("link")) {
        if (relType.equals("alternate")) {
            link = parser.getAttributeValue(null, "href");
            parser.nextTag();
        }
    }
    parser.require(XmlPullParser.END_TAG, ns, "link");
    return link;
}

// Processes summary tags in the feed.
private String readSummary(XmlPullParser parser) throws IOException, XmlPullParserException {
    parser.require(XmlPullParser.START_TAG, ns, "summary");
    String summary = readText(parser);
    parser.require(XmlPullParser.END_TAG, ns, "summary");
    return summary;
}

// For the tags title and summary, extracts their text values.
private String readText(XmlPullParser parser) throws IOException, XmlPullParserException {
    String result = "";
    if (parser.next() == XmlPullParser.TEXT) {
        result = parser.getText();
        parser.nextTag();
    }
    return result;
}
...
}

```

আপনার কাছে গুরুত্বপূর্ণ নতুন বিষয় টিয়গগুলা এডিয়ে (স্কিপ) যান

XML পাসার করার ধাপ যা উপরে আলোচনা করা হয়েছে তার মধ্যে একটি হচ্ছে পাসারের যে টিয়গাগে আগরহেনই সেগুলা এডিয়ে যাওয়া।
এখানে পাসারের skip() পদ্ধতি দেয়া আছে:

```
private void skip(XmlPullParser parser) throws XmlPullParserException, IOException {  
    if (parser.getEventType() != XmlPullParser.START_TAG) {  
        throw new IllegalStateException();  
    }  
    int depth = 1;  
    while (depth != 0) {  
        switch (parser.next()) {  
            case XmlPullParser.END_TAG:  
                depth--;  
                break;  
            case XmlPullParser.START_TAG:  
                depth++;  
                break;  
        }  
    }  
}
```

এভাবে এটা কাজ করে:

- এটা একটি বয়তিকরকে ছেড়ে দেয় যদি চলতি ইভেন্ট একটি START_TAG না হয়।
- এটা START_TAG এবং END_TAG এর সাথে ম্যাচ করা সকল ইভেন্ট বয়বহার করে।
- এটা নিশ্চিত করতে যে এটা সঠিক END_TAG এবং আসল START_TAG এর পরের পরে থাকা টিয়গাগের মধ্যে মুঠোখামুঠি হয় সেটাতে নয়, এটানি স্ট্রিং ডেপথের ট্র্যাক ধরে রাখা।

তাই যদি চলতি এলিমেন্টের নেস্টেড এলিমেন্ট থাকে, depth এর ভ্যালু ০ হবে না যতক্ষণ না পাসারের আসল START_TAG এবং এর ম্যাচ করা END_TAG এর মধ্যে সকল ইভেন্ট বয়বহার করে। উদাহরণস্বরূপ, কীভাবে পাসারের <author> এলিমেন্ট ছেড়ে দেয় তা বিবেচনা করা যার নেস্টেড এলিমেন্ট আছে, <name> এবং <uri>:

- while লুপের মধ্যে দিয়ে পরে থাকা <author> এর পর পাসারের যে টিয়গাগের মধ্যে মুঠোখামুঠি হয় তা হচ্ছে <name> এর জন্য START_TAG। depth এর জন্য ভ্যালু বেড়ে ২ হবে।
- while লুপের মধ্যে দিয়ে দ্বিতীয়বার, পরবর্তী টিয়গাগ পাসারের যার মধ্যে মুঠোখামুঠি হয় তা হচ্ছে END_TAG </name>। depth এর জন্য ভ্যালু কমে ১ হবে।
- while লুপের মধ্যে দিয়ে তৃতীয়বার, পরবর্তী টিয়গাগ পাসারের যার মধ্যে মুঠোখামুঠি হয় তা হচ্ছে START_TAG </uri>। depth এর জন্য ভ্যালু বেড়ে ২ হবে।
- while লুপের মধ্যে দিয়ে চতুর্থবার, পরবর্তী টিয়গাগ পাসারের যার মধ্যে মুঠোখামুঠি হয় তা হচ্ছে END_TAG </uri>। depth এর জন্য ভ্যালু কমে ১ হবে।
- while লুপের মধ্যে দিয়ে পঞ্চমবার এবং শেষবার, পরবর্তী টিয়গাগ পাসারের যার মধ্যে মুঠোখামুঠি হয় তা হচ্ছে END_TAG </author>। depth এর জন্য ভ্যালু কমে ০ হবে, এটানি দর্শন করতে যে <author> এলিমেন্টকে সফলার

সাথে সিঁকপ করা হয়েছে।

XML ডাটাবেস ব্যবহার

নমুনী উদাহরণটি অ্যাপলিকেশন একটি AsyncTask এর মধ্যে যথেষ্ট XML ফিড আনতে যায় এবং পাস করে। এটা পর্দা ইন্টারফেস থেকে বন্ধ করার পরিকল্পনা করে। যখন পরিকল্পনা শেষ হয় অ্যাপ পর্দা একটি টিভিটিভি (NetworkActivity) ইউআইকে আপডেট করে।

নীচের অংশটি দেখায়, loadPage() পদ্ধতি নীচের কাজগুলো করে:

- XML ফিডের জন্য URL সহকারে একটি সিস্টারিং ভেরিয়েবল আরম্ভ করা।
- যদি ইউজারের সেটিং এবং নেটওয়ার্ক কানেকশন এটাকে সমর্থন করে, new DownloadXmlTask().execute(url) আহবান করে। এটা একটি নতুন DownloadXmlTask অবজেক্ট (AsyncTask সাবক্লাস) ইনস্ট্যান্সিয়েট করে এবং এর execute() পদ্ধতি রান করে, যা ফিড ডাউনলোড এবং পাস করে এবং ইউআইতে পর্দা পরিবর্তন করতে একটি সিস্টারিং রেজাল্ট ফেরত দেয়।

```
public class NetworkActivity extends Activity {
```

```
    public static final String WIFI = "Wi-Fi";
    public static final String ANY = "Any";
    private static final String URL = "http://stackoverflow.com/feeds/tag?tagnames=android&sort=newest";

    // Whether there is a Wi-Fi connection.
    private static boolean wifiConnected = false;
    // Whether there is a mobile connection.
    private static boolean mobileConnected = false;
    // Whether the display should be refreshed.
    public static boolean refreshDisplay = true;
    public static String sPref = null;

    ...

    // Uses AsyncTask to download the XML feed from stackoverflow.com.
    public void loadPage() {

        if((sPref.equals(ANY)) && (wifiConnected || mobileConnected)) {
            new DownloadXmlTask().execute(URL);
        }
        else if ((sPref.equals(WIFI)) && (wifiConnected)) {
            new DownloadXmlTask().execute(URL);
        }
        else {
            // show error
        }
    }
}
```

AsyncTask সাবক্লাস নীচে দেখানো হলে, DownloadXmlTask নীচের AsyncTask

পদ্ধতিগুলো বাস্তবায়ন করে:

- doInBackground() পদ্ধতি loadXmlFromNetwork() বাস্তবায়ন করে। এটা ফিডকে একটি পর্যায়ক্রমিক হিসাবে পাস করে। পদ্ধতি loadXmlFromNetwork() ফিড নিয়ে আসে এবং পরিকল্পনা করে। যখন এটা শেষ করে, একটি রেজাল্ট সিস্টারিং ফেরত দেয়।
- onPostExecute() ফেরত আসা সিস্টারিং পরিকল্পনা এবং ইউজার ইন্টারফেসে পর্দা পরিবর্তন করে।

```
// Implementation of AsyncTask used to download XML feed from stackoverflow.com.
private class DownloadXmlTask extends AsyncTask<String, Void, String> {
    @Override
    protected String doInBackground(String... urls) {
        try {
            return loadXmlFromNetwork(urls[0]);
        } catch (IOException e) {
            return getResources().getString(R.string.connection_error);
        } catch (XmlPullParserException e) {
            return getResources().getString(R.string.xml_error);
        }
    }

    @Override
    protected void onPostExecute(String result) {
        setContentView(R.layout.main);
        // Displays the HTML string in the UI via a WebView
        WebView myWebView = (WebView) findViewById(R.id.webview);
        myWebView.loadData(result, "text/html", null);
    }
}
```

নিচের টা হে চ্ ছ প দ্ ধ তি loadXmlFromNetwork() যা DownloadXmlTask তে থকে আহবায়িত হয়। এটানীচের বিষয়গুলো লাকরে থাকে:

1. একটি StackOverflowXmlParser ইনস্ট্যান্সে য়ট করে। এটা এনিম্ টর্ অবেজেক্টর (এনিম্ টর্ স), এব title, url এব summary এর একটি তালিকার (List) জনয্ ভেরিয়েবলও তৈরী করে, ঐ ফিল্ডগুলো লার জনয্ এক সমএল ফিড তে থকে বের করে নিয়ে আসা ভয়্ ালু ধারণ করতে।
2. downloadUrl কল করে, যা ফিড নিয়ে আসে এব একটি InputStream. িহসাবে এটা ফেরত দেয়।
3. InputStream কে পাসর্ করতে StackOverflowXmlParser বয়্ বহার করে। StackOverflowXmlParser ফিড তে থকে পাওয়া ডাটা সহকারে entries এর একটি List অধুয্ িষত করে।
4. entriesList পরে সস করে, HTML মাকর্ আপি দিয়ে ফিড ডাটা একটি তর্ত করে।
5. একটি HTML িস্ টর্ ফেরত আনে যা AsyncTask পদ্ ধতি onPostExecute() দবারা পরধান একটি টিভি টর মধ্যে পর্দশর্ন করে।

```

// Uploads XML from stackoverflow.com, parses it, and combines it with
// HTML markup. Returns HTML string.
private String loadXmlFromNetwork(String urlString) throws XmlPullParserException, IOException {
    InputStream stream = null;
    // Instantiate the parser
    StackOverflowXmlParser stackOverflowXmlParser = new StackOverflowXmlParser();
    List<Entry> entries = null;
    String title = null;
    String url = null;
    String summary = null;
    Calendar rightNow = Calendar.getInstance();
    DateFormat formatter = new SimpleDateFormat("MMM dd h:mmaa");

    // Checks whether the user set the preference to include summary text
    SharedPreferences sharedPrefs = PreferenceManager.getDefaultSharedPreferences(this);
    boolean pref = sharedPrefs.getBoolean("summaryPref", false);

    StringBuilder htmlString = new StringBuilder();
    htmlString.append("<h3>" + getResources().getString(R.string.page_title) + "</h3>");
    htmlString.append("<em>" + getResources().getString(R.string.updated) + " " +
        formatter.format(rightNow.getTime()) + "</em>");

    try {
        stream = downloadUrl(urlString);
        entries = stackOverflowXmlParser.parse(stream);
        // Makes sure that the InputStream is closed after the app is
        // finished using it.
    } finally {
        if (stream != null) {
            stream.close();
        }
    }

    // StackOverflowXmlParser returns a List (called "entries") of Entry objects.
    // Each Entry object represents a single post in the XML feed.
    // This section processes the entries list to combine each entry with HTML markup.
    // Each entry is displayed in the UI as a link that optionally includes
    // a text summary.
    for (Entry entry : entries) {
        htmlString.append("<p><a href=\"");
        htmlString.append(entry.link);
        htmlString.append("\">" + entry.title + "</a></p>");
        // If the user set the preference to include summary text,
        // adds it to the display.
        if (pref) {
            htmlString.append(entry.summary);
        }
    }
    return htmlString.toString();
}

// Given a string representation of a URL, sets up a connection and gets
// an input stream.
private InputStream downloadUrl(String urlString) throws IOException {
    URL url = new URL(urlString);
    HttpURLConnection conn = (HttpURLConnection) url.openConnection();
    conn.setReadTimeout(10000 /* milliseconds */);
    conn.setConnectTimeout(15000 /* milliseconds */);
    conn.setRequestMethod("GET");
    conn.setDoInput(true);
    // Starts the query
    conn.connect();
    return conn.getInputStream();
}

```

বয়্‌াট্‌াির শেয ন্‌া কেরই ডাট্‌া টর্‌ান্‌ সফার

(<http://developer.android.com/training/efficient-downloads/index.html>)

এই ক্‌ল্‌াসে আপনি শিখ্‌ে বন কীভ্‌াে ব ডাইনে ল্‌াড এব্‌ং নেটওয়ার্ক
কানেকশনের বয়্‌াট্‌াির লাইফ পর্‌ভাব ক্‌িমে য়ে ফেলা যায়, িবশেষত ওয়ারলেস
রেডিও এর সাথে সম্পর্কিত। এই ক্‌ল্‌াস কয়্‌াি শিখ্‌ে, পলিগ্‌ এব্‌ং পির্‌ফেচিগ্‌
এর ম্‌েত্‌াে কীশল বয়্‌বহার কেরে ডাউনে ল্‌াড এর পিরকল্‌ পনা এব্‌ং সম্পাদন
করার সের্‌াত্‌তম চর্‌াও িবষয়টি টেদখা য়ে। আপনি শিখ্‌ে বন বয়্‌াট্‌াির
লাইফের উপর পর্‌ভাব ক্‌িমে আনার জনয়্‌ কখন, িক এব্‌ং কীভ্‌াে ব ডাট্‌া
টর্‌ান্‌ সফার করতে হবে ত্‌াে ওয়ারলেস রেডিওর পাওয়ার-বয়্‌বহার
পের্‌াফাইল কীভ্‌াে ব পর্‌ভাি বত কেরে।

অনুশীলনীসমূহ

কাযরু করী নেটওয়ার্কে করু পরবেশ করার জন্য ডাউনলেড অপটিমাইজ করা

এই অনুশীলনী ওয়ারলেস রেডিও সেস্টটমেশিনকে পরিচিৎ করে, বয়স্ক যথা কের কীভাবে আপনার অর্থে পরকানেকটি ভিডিও ডেল এটার সাথে পারস্পরিক করু যা করে থাকে, এবং কীভাবে আপনি আপনার ডাটা কানেকশন কমিয়ে আনতে পারেন এবং আপনার ডাটা ট্রান্সফার করার সাথে সমরুপ করুত বয়স্ক এটারি নিঃশেষ করা কমিয়ে আনতে পরিচিৎ এবং বান্ ডিল এবং বয়স্ক বহার করা।

নিয়মিত আপডেটের পরে ভাব কমিয়ে আনা

পর্যায়মিক ওয়ারলেস রেডিও সেস্টটমেশিনে বয়স্ক পরে ডাউনড আপডেট সেরে তাত্তমভাবে উপশম করেত কীভাবে আপনার রিফরেশ ফিরকে কয়েনিংস ভিন্ হতে পাওতে তা এই অনুশীলনী পরীকৃষা করবে।

অনাবশ্যক (রেডুন্ডেন্ট) ডাউনলেড হচ্ছে অপরুয়োজনীয়

আপনার ডাউনলেড কমানোর সবচেয়ে মৌলিক উপায় হচ্ছে আপনার যেটা পরুয়োজন শুধুমাত্র সেটাই ডাউনলেড করা। এই অনুশীলনী অনাবশ্যক ডাউনলেড পরিহার করার কিছু উকৃষ্ট চর্চার সাথে পরিচিৎ করিয়ে দেয়।

কানেকটিভিটি টাইপের উপর ভিত্তি করে আপনার ডাউনলেড ধরণ পরিবর্তন করুন

যখন এটা বয়স্ক এটারির লাইফের উপরে পরে ভাবিহসাবে আসে, কানেকশনের সকল ধরন সমানভাবে তৈরী হয় না। ওয়াইফাই রেডিও শুধুমাত্র সুনিদিষ্ট ব্রুটভাবে এর অনুরূপ ওয়ারলেস রেডিও এর চেয়ে কম বয়স্ক এটারি বয়স্ক বহারই করে না বরুণে বিভিন্ ওয়ারলেস রেডিও পরুযুক্তি তর মধ্যে বয়স্ক বহার করে রেডিওর বিভিন্ বয়স্ক এটারি ইমিপি লেকশন আছে।

কাযর করী নে ট ও য় া কে কর প র ে ব শ কর ার জন য় ড াউনে ল াড অপ টি ম াই জ কর া

(<http://developer.android.com/training/efficient-downloads/efficient-network-access.html>)

ডাটা ট্রান্সফার করতে ওয়ারলেস ব্যবহার করা হচ্ছে আপনার অ্যাপের
ব্যবহারী নিশ্চেষ্ট করার সবচেয়ে সুনির্দিষ্ট উপায়
সম্ভাবনাময় কোন একটি। নেটওয়ার্ক কানেক্টিভিটির সাথে সম্পর্কিত
ব্যবহারি ব্যবহার মিনিমাইজ করতে, আপনার পক্ষে এটা বেঝা মৌলিক
যেকোনো আপনার কানেক্টিভিটি টিম ডেলিভারি রেডিও হার্ডওয়্যারকে
পরিচালিত করে।

এই অনুশীলনী ওয়ারলেস রেডিও সেটটো মিশনকে পরিচালিত করে এবং
ব্যবহারী করে কীভাবে আপনার অ্যাপের কানেক্টিভিটি টিম ডেলিভারি সাথে
পারস্পরিক করা করে থাকে। এই অধ্যায় দেখায় কীভাবে আপনি আপনার
ডাটা কানেকশন ক্রমেয় আনতে পারেন এবং আপনার ডাটা ট্রান্সফার করার
সাথে সম্পর্কিত ব্যবহারী নিশ্চেষ্ট করা ক্রমেয় আনতে পরিচালিত এবং
বান্ধিল ব্যবহার করা।

রেডিও স্টেট মেশিন

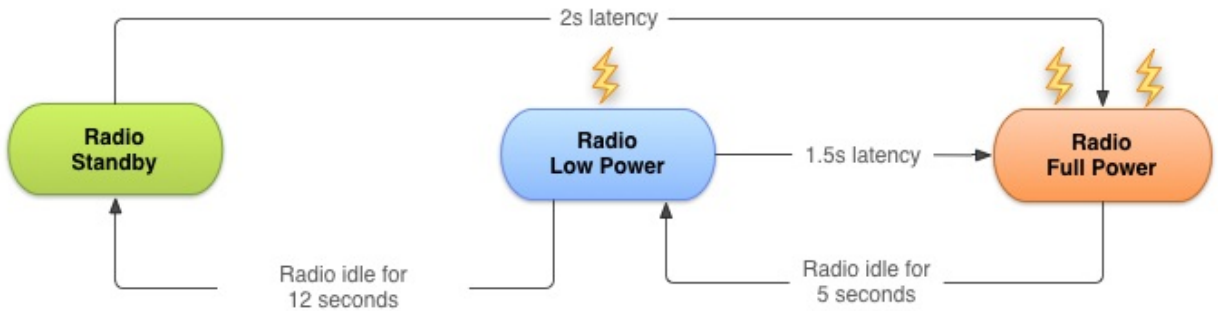
একটি পূর্ণরঙা গভাব সিকর্য ওয়ারলেস রেডিও নিদির্ষ টপাওয়ার বয়বহার করে, সুতরাং এটা যখন কোন বয়বহারের সাথে সম্পৃক্ত না থাকে তখন পাওয়ার রক্ষা (সেভ) করে তি বিভিন্ন এনাঞ্জির্ অবস্থার মধ্যে পরিবর্তিত হতে থাকে, পর্যেয়াজনের সময় রেডিও এর "পাওয়া আপের" (শুরু করা) এর সাথে সম্পিকর্য তলয় াটে টিনিং সেক কমিয়ে আনে তে চেষ্টা করে।

একটি 3G নেটওয়ার্ক রেডিওর জন্য স্টেট মেশিনে তিনটা এনারজি অবস্থা থাকে:

1. সম্পূর্ণ পাওয়ার (Full power): একিকানে কখন যখন সিকর্য থাকে তখন বয়বহৃত হয়, ডিভাইসকে সেরে রাচ চ সম্ভাবয় হারের এরাটা টারান সফার করতে দেয়।
2. কম পাওয়ার (Low power): একটি মাঝামাঝি অবস্থা যা পূর্ণ স্টেটের ৫০% ভাগ বয়টারি বয়বহার করে।
3. স্ট্যান্ডবাই: সবর্য নিম্নে া এনাঞ্জির্ স্টেট (অবস্থা) যে সময়ে কোন নেটওয়ার্ক কানে কখন সিকর্য থাকে না বা পর্যেয়াজন হয় না।

যখন কম এং নিস্কর্য স্টেট (অবস্থা) উল্লখযে যগয় ভাবে কম বয়টারি নিঃশেষ করে, তারানে নেটওয়ার্ক িরেকায়ে স্ট উল্লখযে যগয় লেটে টিনিং সও নিয়ে আসে। অলপ স্টেটে থেক পূর্ণ পাওয়ারে িফের আসতে এটা ১.৫ সেকেন্ড সময় নেয়, নিস্কর্য অবস্থা থেকে পূর্ণ পাওয়ারে িফের আসতে এটা ২ সেকেন্ড সময় নেয়।

লেটে টিনিং স কমিয়ে আনে, স্টেট মেশিন লেয়ার (কম) এনাঞ্জির্ অবস্থায় পরিবর্তন স্খিগত করতে একটি ডিলে (বিলমব) বয়বহার করে। িগার ১ একটি পর্চলিত 3G রেডিওর জন্য AT&T এর টাইমিং বয়বহার করে।



িগার ১: একটি পর্চলিত ওয়ারলেস রেডিও স্টেট মেশিন

পর্যিট া ডিভাইস রেডিও স্টেট মেশিন, নিদির্ষ টভাবে যুক্ পরিবর্তন লেটে টিনিং বিলমব ("tail time") এং শুরু করে, বয়বহৃত ওয়ারলেস রেডিও পর্যুক্ তর (2G, 3G, LTE ইতয় াদী) উপর িভিত্ত করে িভিন্ন হতে পারে এং কেরিয়ার নেটওয়ার্ক যার উপরে ডিভাইস পরিচািরত হয় তার দব্বারা নিধর ািরত এং কনিগার শন করা হয়।

এই অনুশীলনী একটি পর্চলিত 3G ওয়ারলেস রেডিওর জন্য পর্যি তিনিধ স্টেট মেশিনে ক নিয়ে আলাচনা করে, data provided by AT&T (http://www.research.att.com/articles/featured_stories/2011_03/201102_Energy_efficient?fbid=z12VzO4Xy_R) এর িভিত্ত করে। কিন্ত সাধারণ নিয়মকানুন এং ফেলে সেরে াত্ তম চচর া হয় তা সকল ওয়ারলেস রেডিও বাস্ তবায়নের জন্য পর্যেয়াজয়।

এই পদ্ধতি বিেষভাবে সাধারণ ওয়েব বর্ াউজিং এর জন্য কার্যকরী

যেহেতু এটাই উজ্জ্বলতার কণিকা ও যৈব বর্ষা উজ্জ্বলতার সময় অনাহুত লেটিন্সক
পরিহার করে। তুলনামূলকভাবে কম টাইল-টাইমও নিশ্চিত করে যাতে যখনই
একটি বর্ষা উজ্জ্বল সময় শেষ হয়ে যায় তখন একটি কম এনার্জির অবস্থা ঘটে
যেতে পারে।

দুর্ভাগ্যজনকভাবে, এই পদ্ধতি অ্যান্ডারয়েডের মতো আধুনিক
স্মার্টফোনে অকার্যকর অ্যাপ নিয়ে আসতে পারে, যখন অ্যাপ
ফোরগরান্ড (যেখানে লেটিন্স গুরুত্বপূর্ণ) এবং
ব্যাকগরান্ড (যেখানে ব্যাটারি লাইফ গুরুত্বপূর্ণ) উভয়ই রক্ষণ করে।

অয্যাপ কীভাবে রেডিও স্টেশন পরিচালিত করে

একটি নতুন নেটওয়ার্ক ক্যাম্পেইন তৈরী করার পরে প্রতিটি সময়, রেডিও পুনরু
পাওয়ার স্টেটে পরিবর্তন করে। উপরে আলোচিত সাধারণ 3G রেডিও স্টেট
মেশিনের ক্ষেত্রে, এটা আপনার ট্রান্সমিটার সময়কালের জন্য পুনরু
পাওয়ারে থাকবে-এর সাথে টাইল-টাইমের অতিরিক্ত আরও ৫ সেকেন্ড-কম
এনালিজিস স্টেটে ১২ সেকেন্ডকে অনুসরণ করার মাধ্যমে। সুতরাং একটি
সাধারণ 3G ডিভাইসের জন্য, পরে টাট্টা ট্রান্সমিটার সেশন রেডিওকে
পর্যায় ২০ সেকেন্ডের জন্য এনালিজিস ড্র করার কারন ঘটায়।

বাস্তবে, এটা বোঝায় একটি অয্যাপ যা ১ সেকেন্ডের জন্য আনবান্ডল
টাট্টা ট্রান্সমিটার করে পরে ১৮ সেকেন্ড ওয়ারলেস রেডিও টিকে
অবিরাম সিকার্য রাখে, এটাকে উচ্চ পাওয়ারে ফিরিয়ে নিয়ে আসে যেহেতু
এটা নিস্কর্য হয়েছিল। ফলশ্রুতিতে পরে টিমিনেট উচ্চ পাওয়ার
অবস্থায় এটা ১৮ সেকেন্ড বয় টাটারি খরচ করে, এবং বাকী ৪২ সেকেন্ড কম
পাওয়ার অবস্থায় খরচ করে।

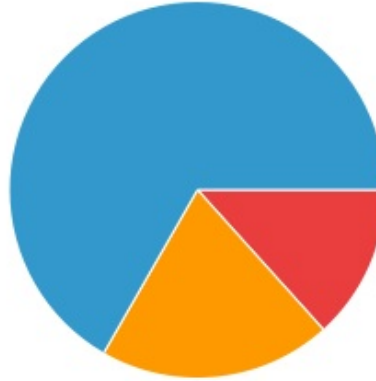
তুলনামূলকভাবে, একই অয্যাপ পরে টিমিনেটের ৩ সেকেন্ডের বান্ডল
ট্রান্সমিটার রেডিওকে ৮ সেকেন্ডের জন্য উচ্চ পাওয়ার স্টেটে রাখে
এবং কম পাওয়ারে মাত্র অতিরিক্ত ১২ সেকেন্ডের জন্য রাখে।

দিব্য উদাহরণ পরে টিমিনেটের অতিরিক্ত ৪০ সেকেন্ডের জন্য
রেডিওকে নিস্কর্য রাখতে দেয়, যা বয় টাটারি খরচের একটি বিরাট ফল
এনে দেয়।

Unbundled Transfers



Bundled Transfers



High Power
Low Power
Idle

ফিগার ২. বান্ডল এবং আনবান্ডল ট্রান্সমিটারের মধ্যে পার্থক্য
ওয়ারলেস রেডিও পাওয়ার ব্যবহার

ডাটা পিপিআর (পরিধান মেমরী থেকে অস্থায়ী স্টোরেজ পরিবর্তী ব্যবহারের জন্য) করা

ডাটা পিপিআর সর্বাধীন ডাটা ট্রান্সফার সেশন সংখ্যা কমিয়ে আনার একটি কার্যকর উপায়। পিপিআর আপনাকে পূর্ণরক্‌ষমতায়, একটি একক কানেকশনের জুরে, একটি একক বাস্ট্র এর মধ্যে একটি পর্দত সময়ের জন্য আপনার জন্য পর্য়েয়াজনীয় ডাটা ডাউনলেড করতে দেয়।

আপনার ট্রান্সফার সম্মুখে লেড করার দব্বারা, আপনি ডাটা ডাউনলেড করতে পর্য়েয়াজনীয় রেডিও একটি ডিভিশনের সংখ্যা কমিয়ে দেন। ফলশ্রুতিতে আপনি শ্রুধুমাতর ব্যব্াটটির লাইফ সেটই করে বননা বর আপনিন লেটেটিন্‌স উন্নতও করতে পারেন, পর্য়েয়াজনীয় ব্যব্ান্ড উইথড কমিয়ে দিতে পারেন এবং ডাউনলেড সময় কমিয়ে দিতে পারেন।

পিপিআর একটি একশন বা ডাটা ভিউ সম্পাদন করার পূর্বের ডাউনলেড সম্পূর্ণরক্‌ষ করার জন্য অপেক্ষার কারনে ইন-অয়্যাপলেটেটিন্‌স সরদব্বারা একটি উন্নত ইউজার এক্সপেরিয়েন্‌স সরবরাহও করে।

কিন্তু, আকর্মনাত্মকভাবে ব্যব্হৃত হওয়া, পিপিআর ব্যব্াটটির নিশেষ হওয়া এবং ব্যব্ান্ড উইথ ব্যব্হারের বৃদ্ধির ঝুঁকিকে পরিবর্তন করে- একইভাবে ডাউনলেড কেটা-ডাটা যেটা ব্যব্হার হচ্ছে নাস্টা ডাউনলেড করার মাধ্যমে। এটানিশ্চত করার জন্য ও গুরুতব্বপূর্ণরক্‌ষে পিপিআর অয়্যাপলিকেশনের শুরুর কার্টাকে বিলম্ব করায় না যখন অয়্যাপ সম্পূর্ণরক্‌ষ করার জন্য পিপিআর জনয়্য অপেক্ষা করে। ব্যব্হারিক ভাষায় যা পর্গতিশীলভাবে ডাটা পর্কিরয়্যাকরনকে বোঝাতে পারে, বাধারাবাহিক ট্রান্সফার আরম্ভ করে অয়্যাপলিকেশনের শুরুর করার জন্য পর্য়েয়াজনীয় ডাটার কানটা ডাউনলেড হবে এবং কান পর্‌সেস করা হবে সেটার গুরুতব্ব অনুরসারে।

আপনি কতটা আকর্মনাত্মকভাবে পিপিআর করেছেন সেটানিভর্ত্ত করে ডাউনলেড হওয়া ডাটার আকারের উপর এবং এটাকীভাবে ব্যব্হৃত হচ্ছে তার উপর। একটি রাফ গাইডিহসাবে, উপরের বিনির্ত সেটটি মেশিনের উপর ভিত্তিত করে, ডাটার জনয়্য যার চলতি ইউজার সেশনের মধ্যে থেকে ৫০% ব্যব্হৃত হওয়ার সুযোগ আছে, আপনি পর্চলিতভাবে পর্ায় ৬ সেকেন্ডের জনয়্য পিপিআর করতে পারেন (পর্ায় ১-২ MB) সম্ভাবয়্য ডাউনলেড করা অবয়্য ব্যব্হৃত ডাটার খরচ, ডাউনলেড না হওয়া সেট করা ও ইউটা ময়্যচ করে শুরুর করার পূর্বের।

সাধারনভাবে বলতে, ঐ ধরনের ডাটা পিপিআর করা একটি ভালো চর্া এমন যে পর্তিত ২ থেকে ৫ মিনিটে আপনার অনয়্য ডাটা ডাউনলেড শুরুর করার পর্য়েয়াজন হবে এবং ১ থেকে ৫ মেগাবাইটস অনুরয়্যায়ী।

এই নিয়ম অনুরসারে, বড় ডাউনলেড-যেমন ভিডিও ফাইল-নিয়মিত বিবর্তিতে একসাথে ডাউনলেড হওয়া উচিত (পর্তিত ২ থেকে ৫ মিনিট), কার্যকরীভাবে কয়েক মিনিটের মধ্যে ইদেখা হবে শ্রুধুম এমন ভিডিও ডাটা পিপিআর করা।

উল্লেখ্য পরিবর্তী ডাউনলেড বান্ডলড হওয়া উচিত, যা পরিবর্তী অয়্যয়ে আলেচনাকরা হয়েছে, Batch Transfers and Connections, এবং এই অনুরমান কানেকশনের ধরণ এবং গতির উপর নিভর্ত্ত করে ভিন্‌ন হয়, যা Modify your Download Patterns Based on the Connectivity Type আলেচনাকরা হয়েছে।

আসুন কিছু বাস্‌তব উদাহরণ দেখা যাক:

একটি মিউজিকে প্লয়ার

আপনি একটি সম্ভূত এ্যালবামি পরফেচ করার জন্য পছন্দ করতে পারেন, কিন্ত উজার পরথম গানেশানার পরেশানা বন্ধ করে দিতে পারে, সেক্ষেত্রে আপনি একটি টেবশিকহু বয়ান্ডউইথড এর পরিমান এবং বয়টিরিবরাইফ অপচয় করবেন।

এচেচেয়েভালোপদধতিহতেপারেএকটিগানহেচহতারসাথেএকটিগানেরবাফাররাখা।মিউজিকিস্ট্রিমিংএরজনয়,একটিঅিবরতিস্ট্রিমযাসবসময়েরিডওএকটিভধারনকরেতারবদলে,বাস্ট্র(bursts)অিডওিস্ট্রিমিটিপেররণকরতেHTTPলাইভিস্ট্রিমিংবয়বহারকরাটাবিবেচনাকরুন,উপরেআলাচনাকরাপিপরফেচিপদধতিঅনুসরণকরুন।

একটি নিউজিরিডার

একটি কয়টিাগিরিনিবরাচিত(িসিলেক্ট)করারপরশুধুহেডলাইনডাউনলেডকরারমাধ্যমেঅেনকনিউজঅয়াপবয়ান্ডউইথডকিমিয়েআনারচেচষ্টাকরে,পূনরআিটরকেলডাউনলেডকরেযখনইউজারএটাপড়তেচায়এখামবনেইল(ছিবি)দেকানেনাহবেশুধুমাত্রযখনতারািভউয়েস্করলকরেটুকবে।

এইঅয়াপেরাচবয়বহারকরে,ইউজারেরিনিউজিরিডিংসেশনেরজনয়রেিডওকেসিকরয়থাকতেবাধ্যকরে,যেহতুতারাহেডলাইনস্করলকরে,কয়টিাগিরিপরিবতরনকরে,আিটরকেলপড়ে।শুধুতাইনয়,পরিতিনিয়তএনািজরসেস্টেটেরমধ্যবদলহওয়াউলেলখযোগয়লেটয়ানিস্তৈরীকরবেযখনকয়টিাগিরিবারিডিংআিটরকেলপরিবতরনহয়।

শুরুরসময়েএকটিযৌক্তিকপরিমানডাটাপিপরফেচকরাএকটিভালোপদধতিহতেপারে,নিউজহেডলাইনএবছিবি(খামবনেইল)পরথমসেটিদিয়েশুরুরকরুন-একটিকমলয়টেিনিংসশুরুরসময়িনিশ্চতকরুন-এববাকীহেডলাইনএবখামবনেইলজারিরাখুনএকইভাবেিনিদেনপক্ষেপরার্থমিকেহেডলাইনতালিকাথেকেবিবদয়মানপরিতটাআিটরকেলেরজনয়আিটরকেলটেকসট।

পরিতটািশেরানাম,খামবনেইল,আিটরকেলটেকসটএবএমনকিহয়েতাপূনরআিটরকেলছিবিপিপরফেকসকরতেঅনয়বিবকলপহেচহ-সাধারণতএকটিপূবরমিধরািরতসুচিতেবয়াকগরাউনেডরমধ্য।এইপদধতিযেকনটেন্টকখনইবয়বহৃতহয়নাইতাডাউনলেডকরেউলেলখযোগয়বয়ান্ডউইথডএববয়টিরিবরাইফবয়করারঝুকিনয়,তাইএটাসতকরতারসাথেবয়বহারকরাউচিত।

ডাউনলেডকরারপূনরিশিডিউলতৈরীকরারএকটিউপায়হেচহশুধুমাত্রতখনযখনওয়াই-ফাইয়েযুকৃতহয়,এবসম্ভবতিভতাইসপরিবতরনহেচহশুধুমাত্রতখন।এটাModify your Download Patterns Based on the Connectivity Type অধ্যয়য়েআেরািবস্তারিতআলাচনাকরাহয়েছে।

বয়্াচ টর্ান্ সফার এব ং স ংযে াগ

একটি স ংযে াগ শুরুর করার পর্িত টা সময়-িনিবর্শে শেষ সম্্পক্ ত ডাটা টর্ান্ সফার এর সাইজ-আপনি রেডিওকে ২০ সেকেন্ ডর কাচাকাছি পাওয়ার ডর্ করার কারন ঘট াতে পারে যখন সাধ ারন 3G ওয়ারের রেডিও বয়্ বহার করেন।

একটি অয্াপ যা পর্িত ২০ সেকেন্ ড পরপর অনুসন্ ধান করে শুধু মাত্র জান ার জনয্ যে অয্াপটি র ান করে ছে এবং ইউজ ারের কাছে দৃশ্যমান আছে, অনিদিদর্ষ্ টক াল ধেরে রেডিও পাওয়ার চ ালু করে রাখে, ফলশর্ুতিতে একটি উল্লেখ্য াগ বয়্ াটারি খরচ করে পর্ ায়কে ান ডাটা স্থ ান ান তরি ত না করেই।

এটার সাথে মেনে রাখ্েব আপনার ডাটা স্থ ান ান তর বান্ ডল করা এবং একটি অিমম া াসি ত টর্ান্ সফার িকউ তৈরী করা ট াগুর তব্ পূনর্। সঠিক ভাবে করতে, আপনি কাযর্ করী ভাবে ফজ-িশফট টর্ান্ সফার করতে পারেন যা একটি অনুর্ প সময় উইনে ার মধ্যে ঘট ার কারনে, তােদর একই সময়ে স ংযি ট ত হতে-িনিশ্চ ত করতে যে যত কম সময়ে সম্ ভব রেডিও পাওয়ার ডর্ করতে পারে।

এই পদ্ধতি তর্মৌলিক দর্শন হেচ্ছ পর্িত টা টর্ান্ সফার সময়ক ালে যতটু কম সম্ ভব ডাটা টর্ান্ সফার করা যা আপনার জনয্ পর্েয াজনীয় সেশনের স ংখ্য া কমিয়ে আনে ত পারে।

তার ম ানে আপনার উচি ত িবলমব্ সহনশীল স্থ ান ান ততর িকউ করার মাধয্ মে আপনার ডাটা বয়্ াচ করা এবং িশি ডউল আপে ডিট এবং িপর্েফিচ াকে িবরত রাখ া, যাতে এটার সবগু লেই সম্ পাদিত হয় যখন সময় স্পষর্ ক াতর টর্ান্ সফার পর্েয াজন হয়। একই ভাবে, আপনার িশি ডউল করা আপে ডিট এবং িনয়িম ত িপর্েফিচ া এর উচি ত আপনার অসম াপ ত টর্ান্ সফার িকউয়ের সম্ পাদন শুরুর করা।

একটি বয়্ বহার িক উদ াহরনের জনয্ Prefetch Data াকে পূবর্ বতর্ াী উদ াহরণে িফেরে য ান।

একটি িনউজ অয্াপ িলেকশন িনন যা উপের িবর্ ত িপর্েফিচ া রুটিন বয়্ বহার করে। িনউজ িরড ার এই ইউজ ারের িরিডি ায়ের ধরণ বুঝতে এবং সবচেয়ে জন িপর্ য গ ল্প র্ য া িক া করতে িবশ্ ল্ষন াত ম ক তথ্য স ংগর্ হ করে। খবর্কে ফের্ শ রাখতে এটা পর্িত যন্ ট ায় আপে ডিট চক করে। বয়্ ান্ ডউইথড স ংরক্ষণ করতে, পর্িত টা আি টর্কে লের জনয্ পূনর্ ফেটা ডাউনলে াড না করে, এটা শুধু মাত্র থ ামব্ নেইল িপর্েফিচ করে এবং ফেটা ডাউনলে াড করে তখনই যখন এটা িসে লক্ ট করা হয়।

এই উদ াহরনে, অয্াপের মধ্যে াকে স ংগর্ িহ ত সকল িবশ্ ল্ষণ ধর্ম াী তথ্যর্ ডাউনলে াড হওয়ার জনয্ একসাথে বান্ ডল হওয়া এবং িকউ হওয়া উচি ত, স্থ ান ান তরি ত হওয়ার বদলে যে হেতু এটা স ংগর্ িহ ত হেয়ে ছে। ফলশর্ুতিতে যে বান্ ডল ডাওয়া যায তার স্থ ান ান তরি ত হওয়া উচি ত যখন হয় একটি পূর্ণর্ সাইজের ফেটা ডাউনলে াড হেত াকে অথবা যখন পর্িত যন্ ট ার আপে ডিট তার কাযর্ কর্ ম করতে াকে।

ক ান সময়-স্পষর্ ক াতর বা চ াহ দ া িভিত ত ক টর্ান্ সফার-যেমন একটি পূনর্ সাইজের ইমেজ ডাউনলে াড করা- িনয়িম ত িশি ডউল করা আপে ডিট করা াকে িবরত রাখ া উচি ত। একটি িপরি ক িল্প ত আপে ডিট িশি ডউল একই সময়ে িবর্ হ হওয়া উচি ত যে হেতু চ াহ দ া িভিত ত ক টর্ান্ সফার, ইনটারভল (অন্ তরব্ তর্ াী ক াল) সেট করার পর পরবতর্ াী আপে ডিট এর সাথে য াতে িন ধর্ ারি ত হেয়ে াকে। এই পদ্ধতি একটি িনয়িম ত আপে ডিট সম্ পাদনের বয়্ ক িমিয়ে আনে, পর্েয াজনীয় সময়-স্পষর্ ক াতর ফেটা ডাউনলে াডে িপিগ-বয়্ াক া করার মাধয্ মে।

কানেকশন (সংযোগ) কিময়ে আনা

এটা সাধারণভাবে নতুন একটা শুরু করার চেয়ে বিদ্যমান সংযোগ পুনর্যবস্থাপনা করার চেয়ে আরও কঠোর করা। সংযোগের পুনর্যবস্থাপনা নেটওয়ার্কের জন্য ঋণাত্মক ও কার্জযাম হওয়ার এবং এই ধরনের নেটওয়ার্ক ডাটাইস্যুও কেসে তরু আরও বুদ্ধিমানতার সাথে পরীক্ষিত করা করতে অনুমোদনও করে।

ডাটাইনডাউনলেড করতে মাল্টিপল যুগে পায়োগী সংযোগ, বা বহুমুখি উপযুক্ত পরি GET রিকোয়েস্ট তৈরী করার বদলে, যেখানে আপনার উচিত ঐ রিকোয়েস্ট একটি একক GET এর মধ্যে বান্ডল করা।

উদাহরণস্বরূপ, কিছু নিউজ ধরনের জন্য বহুমুখি অনুসন্ধান করার চেয়ে একটি একক রিকোয়েস্ট/রেসপন্স এর মধ্যে ফেরত হওয়া পরীক্ষিত ঐ নিউজের জন্য একটি একক রিকোয়েস্ট তৈরী করতে এটা আরও কঠোর করা হতে পারে। সম্ভবতার এবং ক্লায়েন্ট টাইমআউটের সাথে সম্পৃক্ত টারমিনেশন/টারমিনেশন সবারীকোরাকিত প্যাকেট স্থানান্তর করার জন্য ওয়ারলেস রেডিওর সিকিউরিটি হওয়ার পরে যোজন, সুতরাং ঐ টাইমআউটের জন্য অপেক্ষা করার চেয়ে এটা আপনার সংযোগ যখন তাকে কান বস্থাপনার নাথেকে তা বন্ধ করার আরও একটি ভালো চর্চা।

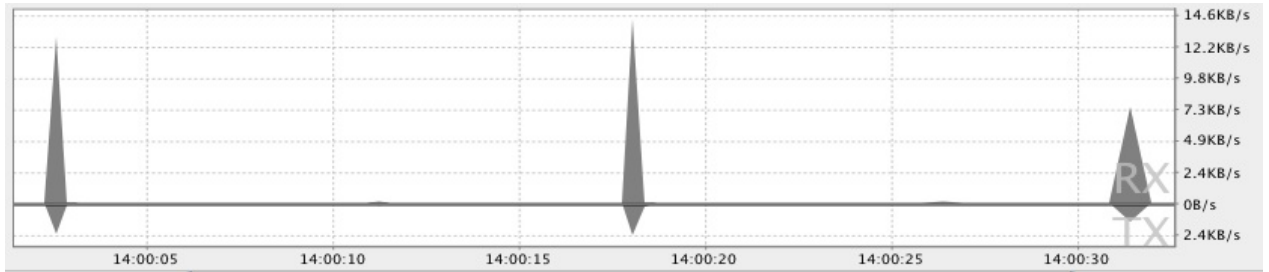
যেটা বলে, একটি সংযোগ তারাতারি বন্ধ করা এটাকে পুনর্যবস্থাপনা হওয়ার হাতে থেকে রক্ষা করে, যা পরবর্তীতে একটি নতুন সংযোগ পরীক্ষিত ঐ করার জন্য অতিরিক্ত ওভারহেড আকাঙ্ক্ষা করে। একটি উপকারী বোঝাপড়া হচ্ছে তা ক্রমিকভাবে সংযোগ বন্ধ না করা, কিন্ত অনুত্তিমরীতি হত টাইমআউটে ময়াদে শেষের পূর্বের এর খুব কাছাকাছি থাকে।

চিহ্নিতার জায়গা চিহ্নিত করতে DDMS নেটওয়ার্ক ট্রাফিক টুল ব্যবহার করে

অ্যান্ড্রয়েড ডিডিউএমএস (Dalvik Debug Monitor Server)

<http://developer.android.com/tools/debugging/ddms.html>) একটি টিবিবিস্তারিত করান নেটওয়ার্ক
ব্যবহারের টয় এবং অন্তর্ভুক্ত করে যা এটাকে অনুসরণ করতে সম্ভব
করে যখন আপনার অ্যাপলিকেশন নেটওয়ার্ক রিকোয়েস্ট তৈরি করেছে। এই
টুল ব্যবহার করে, আপনি মনিটর করতে পারেন কীভাবে এবং কখন আপনার
অ্যাপ ডাটাবেজ সফার করে এবং যথার্থভাবে মৌলিক কোড অপটিমাইজ
করে।

ফিগার ৩ দেখায় পর্যায় ১৫ সেকেন্ড পরপর অল্প পরিমাণ ডাটা
ট্রান্সফার করার একটি টপগ্রাফ, যা পরামর্শ দেয় যে, প্রতিটা
রিকোয়েস্ট বা আপলোড বান্ধুল করাপি পরফেচ করতে কায়রুকারিতা
নাটকীয়ভাবে উন্নত হতে পারে।



ফিগার ৩. DDMS দিয়ে নেটওয়ার্ক ব্যবহার ট্র্যাক করা

আপনার ডাটাবেজ সফারের ফিরকে যেনি (পুনরাবৃত্তি) এবং
প্রতিটা স্যেগে পর সময় ট্রান্সফার হওয়া ডাটার পরিমাণ মনিটর করার
মাধ্যমে, আপনি আপনার অ্যাপলিকেশনের এলাকা চিহ্নিত করতে পারেন যা
আরও ব্যাটারি-কায়রুকারিতা তৈরি করতে পারে। সাধারণভাবে আপনি
সঙ্কীর্ণ স্পাইক জুড়ে পারেন যা বিলম্ব হতে পারে, অথবা যা একটি
পরবর্তী ট্রান্সফার বিরত হওয়ার কারণ ঘটায়।

স্পাইক ট্রান্সফারের কারণ আরও ভালভাবে চিহ্নিত করতে, ট্রাফিক
স্টেট এপিআই পদ্ধতি `TrafficStats.setThreadStatsTag()` ব্যবহার করে একটি থ্রেডের
মধ্যে থেকে যিষ্ঠ ডাটাবেজ সফার টয়াগ করতে আপনাকে অনুমোদন
করে, ম্যানুয়ালি `tagSocket()` এবং `untagSocket()` ব্যবহার করে একক সেক্ট টয়াগ
করা অনুসরণ করার মাধ্যমে। উদাহরণস্বরূপ:

```
TrafficStats.setThreadStatsTag(0xF00D);  
TrafficStats.tagSocket(outputSocket);  
// Transfer data using socket  
TrafficStats.untagSocket(outputSocket);
```

চলতি `getThreadStatsTag()` ভ্যালুর উপর ভিত্তি করে অ্যাপাশ `HttpClient` এবং
`URLConnection` লাইব্রেরী সর্বোচ্চ যত্নে সেক্ট টয়াগ করে। এই
লাইব্রেরীগুলো সেক্ট টয়াগ এবং আনটয়াগ করে যখন জীবন-রাখা
পুল এর মাধ্যমে রিসাইকেল করা হয়।

```
TrafficStats.setThreadStatsTag(0xF00D);
try {
    // Make network request using HttpClient.execute()
} finally {
    TrafficStats.clearThreadStatsTag();
}
```

অ্যান্ডরয়েড ৪.০ সেকিট ট্যাগিংকে সাপোর্ট করে, কিন্তু রিয়েল-টাইম স্ট্যাটস শুধুমাত্র অ্যান্ডরয়েড ৪.০.৩ বা এর চেয়ে উপরের সংস্করণ দ্বারা রান করা ডিভাইসে পর্দাশীর্ণ হবে।

নিয়মিত আপডেটের পরে ভাব কমানো

(http://developer.android.com/training/efficient-downloads/regular_updates.html)

নিয়মিত আপডেটের অপটিমাইজেশন ফির্কে কয়েকটি সডিভাইসের অবস্থান, নেটওয়ার্ক কানেকটিভিটি, ইউজারের আচরণ এবং ইউজারের পছন্দ উপর নির্ভর করে।

Optimizing Battery Life কীভাবে ব্যাটারি-কার্যকর অ্যাপ তৈরী করতে হয় তা আলোচনা করে যা ধারক ডিভাইসের অবস্থার উপর নির্ভর করে তাদের রিফরেশ ফির্কে কয়েকটি সপরিবর্তন করে। এগুলির মধ্যে যথেষ্ট নিষিদ্ধ করা ব্যাকগ্রাউন্ড সার্ভিস অ্যাপেট যখন আপনি আপনার কানেকটিভিটি হারিয়ে ফেলেন এবং আপডেটের হার কমিয়ে ফেল যখন ব্যাটারি লেভেল কম যায়।

এই অনুশীলনী কীভাবে আপনার রিফরেশ ফির্কে কয়েকটি সমূল ওয়ারলেস রেডিও স্টেটমেনে ব্যাকগ্রাউন্ড আপডেটের পরে ভাব সবেচেয়ে ভালোভাবে কমিয়ে আনতে ভিন্ন হতে পারে তা পরীক্ষা করে।

পেপালিং এবিকল্‌প হিসাবে গুগল ক্লাউড মেসাজিং ব্যবহার

সবসময় আপনার অ্যাপ আপনার সাভার্‌র পেপাল করে যদি কোন আপডেটের
পর্যেয়াজন হয় তা পরীক্ষা করে, যদি একটি সাধারণ ওজিকানে কশনে পর্যায়
২০ সেকেন্ডের জন্য আপনি ওয়ারলেসেরিডিও সিকর্য করেন,
অপর্যেয়াজনীয়ভাবে পাওয়ার ডর্ করেন।

Google Cloud Messaging for Android (GCM) (দেখুন <http://developer.android.com/google/gcm/index.html>)
হচ্ছে একটি হালকা ধরনের কৌশল যা সাভার্‌র থেকে একটি নিদির্ষ্ট
অ্যাপ ইনসেটেনেস ডাটাস্থানান্তরের কাজে ব্যবহৃত হয়। GCM
ব্যবহার করে, আপনার সাভার্‌র একটি নিদির্ষ্ট ডিভাইসেরান করা
আপনার অ্যাপকে জানাতে পারে যে এর জন্য নতুন ডাটারেছে।

পেপালিং এর সাথে তুলনাকরে, যেখানে আপনার অ্যাপকে অবশ্যই নতুন
ডাটার জন্য অনুসন্ধান করে তিনয়িমিত সাভার্‌রকে পিৎ করতে হবে, এই
ইভেন্ট চািলত মডেল আপনার অ্যাপকে একটি নতুন কানে কশন তৈরী করতে
দেয় শুধু তখন যখন এটা জানে ডাউনলেড করা জন্য ডাটা আছে।

ফলাফল হচ্ছে একটি অপর্যেয়াজনীয় কানে কশন কমানো এবং আপনার
অ্যাপলিকেশনের মধ্যে আপডেট ডাটার জন্য একটি কমিয়ে আনা লেটিন্স।

GCM একটি স্থায়ী TCP/IP কানে কশন ব্যবহার করে বাস্‌তবায়িত হয়। যখন
আপনার নিজস্ব পুশ সাভার্‌স বাস্‌তবায়ন করা সম্ভব, এটাই সবচেয়ে
ভালে যে এক্ষেত্রে GCM ব্যবহার করা। এটা স্থির কানে কশনের সঞ্ছা
কমিয়ে আনে এবং প্লাটফর্মের কব্যান্ডউইথড অপটিমাইজ করতে এবং
ব্যব্‌টারি লাইফের সাথে সম্পৃক্ত পর্ভাব কমিয়ে আনতে দেয়।

ইন এক্সসফেক্টিরিপিটিং এয়ালামর্ এবং এক্সপোনেন্ট বয়াকঅফিডিয়েপোলিং অপটিমাইজ করা

যেখানে পোলিং পর্যেয়াজন, ইউজার এক্সপিরিয়েন্স থেকে হর্সাসনা করেই আপনার অয়্যেপরিডফল্ট ডাটারিফেরশিফরকে ায়েয়িন্স যতদূর সম্ভব কমিয়ে সেট করাটাই ভালো।

একটি সরল পদ্ধতি হচ্ছে ইউজারকে তাদের পর্যেয়াজনীয় আপডেটের টিনিজ পছন্দমতো সুষ্পষ্টভাবে সেট করতে দেয়া, তাদেরকে ডাটারিফেরশেনেস এবং বয়্যাটারির লাইফের মধ্যে যেকার নিজস্ব ভারসাম্য নিধারণ করতে দেয়া।

যখন আপডেটিশিডিউল করা হয়, ইন এক্সসফেক্টিরিপিটিং এয়ালামর্ বয়্যবহার করুন যািসেস্টমকে পর্তিতটা এয়ালামর্ টিটর্গর্ ায়ের সময় "ফেজ শিফট" ("phase shift") করতে দেয়।

```
int alarmType = AlarmManager.ELAPSED_REALTIME;  
long interval = AlarmManager.INTERVAL_HOUR;  
long start = System.currentTimeMillis() + interval;  
  
alarmManager.setInexactRepeating(alarmType, start, interval, pi);
```

যদি কতিপয় এয়ালামর্ একই সময়ে টিটর্গর্ ার করার জন্য শিডিউল করা হয়ে থাকে, এই ফেজ শিডিউল একই সাথে টিটর্গর্ ার হতে দিতে পারে, পর্তিতটা আপডেটকে একটি একক সিকর্যেরিডিও সেস্টটিপরিবর্তনের উপর piggyback করতে দেয়।

যেখানেই সম্ভব, সমগেগাতর্ীয়_WAKEUP এনাকের ELAPSED_REALTIME বা RTC এ আপনার এয়ালামর্ টাইপ সেট করুন। এটা অপেক্ষাকার মধ্যমে বয়্যাটারির পর্তাব অধিক হার কমিয়ে আন, ফোনটি এয়ালামর্ টিটর্গর্ ার হওয়ার পূবর্ পযর্ন্ত ততযতক্ষননা স্টয়ান্ডবাইমে ডেনা থাকে।

আপনি আপনার অয়্যাপকত সম্পর্তিত বয়্যবহার করেছেন তার উপর ভিত্তি করে তাদের রিফরকে ায়েয়িন্স কমিয়ে দেয়ার মধ্যমে এই শিডিউল করা এয়ালামর্ গুলোর পর্তাব কমিয়ে আনতে পারেন।

একটি উপায় হচ্ছে আপনার আপডেটের রিফরকে ায়েয়িন্স (এবং/বা পির্ফেচিং এর ডিগর্ আপনি সম্পাদন করেন) কমাতে একটি এক্সপোনেনশিয়াল বয়্যাক-অফ ধরন বাস্তুবায়ন করা, যদি অয়্যাপটি পূবর্বতর্ী আপডেটে থেকে বয়্যবহৃত হয়ে না আসে। এটা কখনও কখনও একটি কমপেক্ষ আপডেট রিফরকে ায়েয়িন্স জানাতে এবং যখনই অয়্যাপ বয়্যবহৃত হয় তখন রিফরকে ায়েয়িন্স পূর্গর্ ায়ে সেট করতে উপকারী। উদাহরণস্বরূপ:

```

SharedPreferences sp =
    context.getSharedPreferences(PREFS, Context.MODE_WORLD_READABLE);

boolean appUsed = sp.getBoolean(PREFS_APPUSED, false);
long updateInterval = sp.getLong(PREFS_INTERVAL, DEFAULT_REFRESH_INTERVAL);

if (!appUsed)
    if ((updateInterval * 2) > MAX_REFRESH_INTERVAL)
        updateInterval = MAX_REFRESH_INTERVAL;

Editor spEdit = sp.edit();
spEdit.putBoolean(PREFS_APPUSED, false);
spEdit.putLong(PREFS_INTERVAL, updateInterval);
spEdit.apply();

rescheduleUpdates(updateInterval);
executeUpdateOrPrefetch();

```

আপনি বস্তু ক্রমিক ক্রম এবং ডাউনলোড এরের পরে ভাব ক্রমিক আনতে একই ধরনের এক সপ্তেনশিয়াল বস্তু ক্রমিক-অফ ধরন বস্তু বহর করতে পারেন।

আপনি আপনার সাধারণের সাথে যোগে যোগ করতে সামর্থ্যে হান এবং ডাটা ডাউনলোড করুন বানান করুন একটি টেনেটওয়াকর ক্রমিক ক্রম আর্মভ করার খরচ একই। সময়-সংবেদী ট্রান্সফার যেখানে সফল সমাপিত গুরুত্বপূর্ণ, সম্প্রদায় বস্তু ট্রান্সফার পরে ভাব ক্রমিক আনার জন্য পূর্ণরায় চেষ্টা করার ফিরকে ক্রমিক সক্রমতে একটি এক সপ্তেনশিয়াল বস্তু ক্রমিক-অফ এ য়ালেগরিদম বস্তু বহর করা যেতে পারে, উদাহরণস্বরূপ:

```

private void retryIn(long interval) {
    boolean success = attemptTransfer();

    if (!success) {
        retryIn(interval*2 < MAX_RETRY_INTERVAL ?
            interval*2 : MAX_RETRY_INTERVAL);
    }
}

```

বিবর্তপভাবে, বস্তু ত্রুটি সহ্যে ট্রান্সফারের জন্য (যেমন, নিয়মিত আপডেট), আপনি শুধু অসফল ক্রমিক ক্রম এবং ট্রান্সফার চেষ্টা এড়িয়ে যেতে পারেন।

রিডানডেন্ট ডাউনলেড অনাবশ্যক

(http://developer.android.com/training/efficient-downloads/redundant_redundant.html)

আপনার ডাউনলেড কমান্ডের সবচেয়ে মৌলিক উপায় হচ্ছে যা আপনার পরে যাজন শুধু সেটাই ডাউনলেড করুন। ডাটার নিরিখে, যার মানে REST APIs বাস্তবায়ন করা যা আপনার কমান্ড অনুসন্ধান নিবন্ধকে সুনির্দিষ্ট করতে দেয় যা প্যারামিটার বর্ধার করে ডাটা ফেরত আসাকে সুনির্দিষ্ট করে,, যেমন আপনার শেষ আপডেটের সময়টি।

একইভাবে, যখন ইমেজ ডাউনলেড করা হয়, ক্লায়েন্ট কমান্ড পূর্ণরূপে সাইজের ইমেজ ডাউনলেড না করে সম্ভার-সাইডের ইমেজ সাইজ কমান্ড টাই ভালে।

স্থানীয়ভাবে ফাইল ক্যাশে (Cache) করা

আমেরকিটি গুরুত্বপূর্ণরকৌশলহচ্ছেএকই/অনুরূপডাটাদাউনলোডকরা
পরিহারকরা।দৃঢ়ভাবেক্যাশে(caching)করারমাধ্যমেআপনিএটাকরতে
পারেন।সবসময়স্ট্যাটিকরিসেসারক্যাশেকরুন,যারমধ্যেযেহে
চাহিদাকৃত(অন-ডিমান্ড)ডাউনলোডযেমনপূনরুসাইজেরইমেজ,
যেটিক্তকভাবেযতটুকুদীর্ঘরক্ষণসম্ভবততক্ষণপযর্ন্ত।আপনার
অন-ডিমান্ডক্যাশেরসাইজব্যবস্থাকরতেআপনাকেএটানিয়মিত
ফ্লয়শকরতেসিকর্যকরতেচাহিদাকৃত(অন-ডিমান্ড)রিসেসার
পৃথকভাবেসেটআরকরাউচিত।

এটানিশ্চিতকরতেযেআপনারক্যাশেআপনারঅ্যাগেপরমধ্যেপূরাতন
ডাটাপরদশরুনকরারকারণনাহয়,সময়বেরকরারব্যাপারেনিশ্চিত
হোন,যারমধ্যেযেটিক্তকয়েস্টকরাকনটেন্টশেষবারআপডেটহয়েছিলএবং
যখনযেদশেষহয়েছিল,যারমধ্যেযেথেকেHTMLক্রেসপন্সকরাহেডারস।
এটাআপনাকেিস্থরকরতেদেয়যখনসম্পৃক্তকনটেন্টরিফরশডহওয়া
উচিত।

```
long currentTime = System.currentTimeMillis();

URLConnection conn = (URLConnection) url.openConnection();

long expires = conn.getHeaderFieldDate("Expires", currentTime);
long lastModified = conn.getHeaderFieldDate("Last-Modified", currentTime);

setDataExpirationDate(expires);

if (lastModified < lastUpdateTime) {
    // Skip update
} else {
    // Parse update
}
```

এইপদ্ধতিব্যবহারকরে,আপনিকায়রকরীভাবেডায়নামিককনটেন্ট
ক্যাশেকরতেপারেনযখননিশ্চিতকরেএটাআপনারঅ্যাগেপরমধ্যে
পূরাতনডাটাপরদশরুনকরেন।

আপনিকোনব্যবস্থাপনাছাড়াবহিরাগতক্যাশেডিভেরকটরীতে
অসংবেদনশীলডাটাক্যাশেকরতেপারেন:

```
Context.getExternalCacheDir();
```

ব্যতিক্রমভাবে,আপনিসুষ্ঠব্যবস্থাপিত/নিরাপদঅ্যাপলিকেশন
ব্যবহারকরতেপারেন।উল্লেখ্যএইঅন্তর্গতক্যাশেফ্লয়শহেত
পারেযখনসিস্টেমসহজপর্যাপ্যসেটআরেজকমভাবে(Low)রানকরেছে।

```
Context.getCache();
```

ক্যাশেলেকেশনেসেটারহওয়াফাইলমুছেযেতেপারেযখন
অ্যাপলিকেশনআন-ইনস্টলকরাহয়।

HttpURL কানেকশন রেসপন্স ক্যাশে ব্যবহার করুন

অ্যান্ড্রয়েড ৪.০ HttpURLConnection এ একটি রেসপন্স ক্যাশে যুক্ত করে।
আপনি নিচের মতো পদ্ধতি ফলন ব্যবহার করে সাপোর্ট করা ডিভাইসে HTTP
কে রেসপন্স করা ক্যাশে সচিবায় করতে পারেন:

```
private void enableHttpResponseBodyCache() {  
    try {  
        long httpCacheSize = 10 * 1024 * 1024; // 10 MiB  
        File httpCacheDir = new File(getCacheDir(), "http");  
        Class.forName("android.net.http.HttpResponseCache")  
            .getMethod("install", File.class, long.class)  
            .invoke(null, httpCacheDir, httpCacheSize);  
    } catch (Exception httpResponseBodyCacheNotAvailable) {  
        Log.d(TAG, "HTTP response cache is unavailable.");  
    }  
}
```

এই উদাহরণ কোড পরামর্শ দিককারিরিলজকে পরম্ভাবিত না করেই
অ্যান্ড্রয়েড ৪.০+ ডিভাইসে রেসপন্স ক্যাশে চালু করতে পারেন।

একটি নেটওয়ার্ক কানেকশন ওপেন করার পরে যোজনীয়তাকে বাদ দিয়ে
ক্যাশে ইনস্টল ডিডেই, সম্ভবপর ক্যাশে ড HTTP এরিকোয়েস্ট স্থানিক
সেটোরজ থেকে সরাসরি পরিবেশিত হতে পারে, ডাউনলোডের সাথে
সম্পৃক্তত ব্যবান্ডউইথ বাদ দিয়ে শতরং সাপেক্ষভাবে ক্যাশে ড
রেসপন্স সাঁরভার থেকে তাদর পরিচ্ছন্নতাকে বৈধ করতে পারে।

আনক্যাশে ড রেসপন্স ভবিষ্যতেরিকোয়েস্টরজন্য রেসপন্স ক্যাশের
মেধ্যে সেটার হয়।

কানেকটিভিটি ধরনের উপর ভিত্তি করে ডাউনলেডের ধরন পরিবর্তন করুন

(http://developer.android.com/training/efficient-downloads/connectivity_patterns.html)

যখন এটা বয়্‌টাটির লাইফ পর্ভাব ফেলেত আসে, সকল কানেকশন ধরন সমানভাবে তৈরী হয় না। ওয়াই-ফাই রেডিও এর সমরূপ ওয়ারলেস রেডিওর চাইতে উল্লেখ্যে যোগ্য কম বয়্‌টাটির বয়্‌বহারই করেনা, বিভিন্ন ন ওয়ারলেস রেডিও পর্যুক্ষিত তে বয়্‌বহৃত হওয়া রেডিওতে বিভিন্ন ন বয়্‌টাটির ইমপি প্লে কশনও থাকে।

ওয়াই-ফাই ব্যবহার

অধিকাংশ ক্ষেত্রে একটি ওয়াই-ফাই রেডিও উল্লেখযোগ্য পরিমাণ কম ব্যাটারি খরচে অধিকতর ডাটা ব্যান্ডউইথ দেয়ার পরিস্কার করে থাকে। ফলশ্রুতিতে, আপনার উচিত যখনই সম্ভব ডাটার ব্যান্ড সফল কার্যকর সম্পাদনের চেষ্টা করা যখন ওয়াই-ফাইয়ে সংযুক্ত হোন।

কোনেকিটি ভিটি পরিবর্তন শোনার জন্য আপনি একটি বর্ডকাস্ট রিসিভার ব্যবহার করতে পারেন যখন একটি ওয়াই-ফাই কোনেকশন নির্দিষ্ট পরিমাণ ডাউনলোড, শিডিউল করা আপডেট অর্থাৎ, এবং Optimizing Battery Life অনুশীলনের Determining and Monitoring the Connectivity Status তে আলোচিত নিয়মিত আপডেটের ফ্রিকোয়েন্স অস্থায়ীভাবে বর্ধিত করে সংস্থাপিত হয়।

আরও বেশী বেশী ডাউনলেড করতে ব্যান্ডউইথ ব্যবহার করুন

যখন একটি ওয়ারলেস রেডিওতে সংযুক্ত হয়, উচ্চ পযর্ষের
ব্যান্ডউইথ সাধারণত উচ্চ ব্যাটারি খরচ হৈয়ে থাকে। এর মানে LTE
সাধারণভাবে 3G এর চাইতে বেশী শক্ত খরচ করে, যেটা যথাক্রমে 2G এর
চাইতেও ব্যয়বহুল।

এটাে বোঝায় যে যখন রেডিও পর্যাযুক্তের উপর ভিত্তি করে মূল রেডিও
স্টেট মিশন ভিন্ন হৈয়ে থাকে, সাধারণভাবে বলতে গেলে সেস্টেটে চেক
টাইল-টাইম এর সম্পর্কিত ব্যাটারি পরে ভাব উচ্চমাত্রার
ব্যান্ডউইথ রেডিওর জন্যে বেশী।

একই সময়ে, উচ্চমাত্রার ব্যান্ডউইথ বেঝায় আপনি আরও দৃঢ়ভাবে
পরিচালনা করতে পারেন, ওই সময় জুড়ে আরও ডাটা ডাউনলেড করতে পারেন।
সম্ভবত কম সঞ্চে, কারণ টাইল-টাইম ব্যাটারি খরচ অপেক্ষাকৃত
বেশী, এটা আপডেটের ফ্রিকোয়েন্সি কমতে পরে তটাই টার্মান সফার
সেশনের দীর্ঘ সময়ের জন্যে রেডিওকে সক্রিয় হিসাবে রাখতে আরও
কায়র করীও।

উদাহরণস্বরূপ, যদি একটি LTE রেডিওর ব্যান্ডউইথ দিবং গুণ এবং 3G এর
শক্তির ব্যয় দিবং গুন করে, আপনার উচিত পরে তটাই সেশনে ৪ বার যতটুকু
সম্ভব ডাটা ডাউনলেড করা-বা ১০mb এর কাছাকাছি যতটুকু সম্ভব। যখন
এত ডাটা ডাউনলেড করে বন, বিদ্যমান স্থানীয় সেস্টারে জে আপনার
পরিচালনা পরে ভাবে বিবেচনা করা এবং নিয়মিতভাবে আপনার কয়শে
ফলাশ করাটা গুরুত্বপূর্ণ।

সক্রিয় ওয়ারলেস রেডিও নিধরারণ করতে আপনিন কানে কটি ভিটি
ময়ানে জার ব্যবহার করত পারেন এবং আপনার পরিচালনা রেডিও সূচী
পরিবর্তন করতেও পারেন:

```
ConnectivityManager cm =  
(ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);  
  
TelephonyManager tm =  
(TelephonyManager) getSystemService(Context.TELEPHONY_SERVICE);  
  
NetworkInfo activeNetwork = cm.getActiveNetworkInfo();  
  
int PrefetchCacheSize = DEFAULT_PREFETCH_CACHE;  
  
switch (activeNetwork.getType()) {  
case (ConnectivityManager.TYPE_WIFI):  
    PrefetchCacheSize = MAX_PREFETCH_CACHE; break;  
case (ConnectivityManager.TYPE_MOBILE): {  
    switch (tm.getNetworkType()) {  
        case (TelephonyManager.NETWORK_TYPE_LTE |  
            TelephonyManager.NETWORK_TYPE_HSPAP):  
            PrefetchCacheSize *= 4;  
            break;  
        case (TelephonyManager.NETWORK_TYPE_EDGE |  
            TelephonyManager.NETWORK_TYPE_GPRS):  
            PrefetchCacheSize /= 2;  
            break;  
        default: break;  
    }  
    break;  
}  
default: break;  
}
```

ক্লাউডে ডিসঙ্ক (Syncing) করা

(<http://developer.android.com/training/cloudsync/index.html>)

ইন্টারনেট কানেক্টিভিটির জনয্ শক্তিশালী এপিআই সরবরাহ করে, অ্যান্ড্রয়েড ফর্মওয়ার্ক সমৃদ্ধ ক্লাউড-সিঙ্কিং অ্যাপ তৈরী করতে সহায়তা করে যা তাদের ডাটাবেজ এক্টিভিটি মেটাডেটা সঞ্চিত করে, নিশ্চিত করে আপনার সকল ডিভাইস সবসময় সিঙ্ক থাকে এবং আপনার মূল্যবান ডাটাসবসময় ক্লাউডে একটি বয়াকআপ হিসাবে থাকে।

এই ক্লাস ক্লাউড সিঙ্কিং অ্যাপলিকেশনের জনয্ বিভিন্ন কৌশল আলোচনা করে। আপনার নিজস্ব বয়াক-এন্ডেড অ্যাপলিকেশন ব্যবহার করে এটাক্লাউড দিয়ে ডাটাসিঙ্ক করার বিষয়ে আলোচনা করে যাতে ইউজার তাদের ডাটার স্টোর করতে পারেন যখন একটি নতুন ডিভাইসে আপনার অ্যাপলিকেশন ইনস্টল করা হয়।

অনুশীলনী

বয়স্ক আপ এপিআই বয়স্ক বহার

শিক্ষানুষ্ঠান কীভাবে আপনার অর্থান্ডারয়েড অর্থাপিলিকেশনের মধ্যে বয়স্ক আপ এপিআই একীভূত করা হয়, যাতে ইউজার ডাটাইপারফরেন্স, নেটিভ এবং গেম হাইস্কোর আপডেটিনখুঁতভাবে ইউজারের ডিভাইস জুড়ে রাখতে পারে।

অধিকাংশ গুগল ক্লাউড মেসেজিং তৈরী করা

শিক্ষানুষ্ঠান কীভাবে কায়রু করীভাবে মাল্টিটাস্টমেসেজ সেন্ড করা যায়, বুদ্ধিধর্মতার সাথে গুগল ক্লাউড মেসেজিং (GCM) মেসেজ পরীতিকর্যাজানোনা হয়, এবং সম্ভারার দিয়ে কায়রু করীভাবে সিঙ্ক করতে এঙ্গ মেসেজ বয়স্ক বহার করা হয়।

বয়স্ক আপ এপিআই ব্যবহার

(<http://developer.android.com/training/cloudsync/backupapi.html>)

যখন একজন ইউজার একটি নতুন ডিভাইস কর্তৃক ব্যবহার করা শুরু করেন, তখন তারা আশা করেন যে পুরনো যন্ত্রে থাকা গুলি পুরনো যন্ত্র থেকে সেটিআপের সময় আপনার অ্যাপের ডেটা ডিভাইসে রিস্টোর করা হবে, একই সাথে অ্যাপের সমস্ত পুরনো ডেটাও রিস্টোর করা হবে। বাইন্ডিং ফল্ট এটা হয় না, এবং আপনার অ্যাপ ইউজারের সকল অর্জনের সঠিক হাতিয়ে যাবে।

যখন ডেটার আয়তন অপেক্ষাকৃত কম (মেগাবাইটের কম) সে অবস্থায়, ইউজারের পিওরফরেন্স, নোটস, মেসেজ ইত্যাদি স্কার অথবা অন্য পিসিস্টেম, বয়স্ক আপ এপিআই একটি হালকা সমাধান দিয়ে থাকে। এই অনুশীলনী আপনার অ্যাপের মধ্যে বয়স্ক আপ এপিআই একীভূত করার এবং বয়স্ক আপ এপিআই ব্যবহার করার নতুন ডিভাইসে ডেটা রিস্টোর করার বিষয়ে আলোচনা করবে।

অ্যান্ড্রয়েড ব্যাকআপ সার্ভিসের জনস্টিয়ার

এই অনুশীলনী Android Backup Service (<https://developer.android.com/google/backup/index.html?csw=1>) এর ব্যবহার চায় যার জন্য পরে যাজন রেজিষ্টেশন। সামনে এগিয়ে যান এবং এখানে (<https://developer.android.com/google/backup/signup.html?csw=1>) রেজিস্টার করুন। যখনই এটা করা হবে, সার্ভিসটি একটি এক্সএমএল ট্যাগ আগে থেকেই রেখে দিবে আপনার অ্যান্ড্রয়েড মনিফেস্টের মধ্যে যুক্ত করতে, যেটা দেখতে অনেকটা এরকম হবে:

```
<meta-data android:name="com.google.android.backup.api_key"  
  android:value="ABcDe1FGHij2KlMn3oPQRs4TuvW5xYZ" />
```

উল্লিখ্য পরতিটা ব্যাকআপিক একটি সেপারিসিফিক প্যাকেজ নামের সাথে কাজ করে। যদি আপনার ভিন্ন ভিন্ন অ্যাপলিকেশন থাকে, পরতিটার জন্য পৃথককি রেজিস্টার করতে হবে।

আপনার মেনিফেস্ট কনফিগার করুন

অ্যান্ড্রয়েড বয়াকআপ সার্ভিস বহার করার জন্য আপনার অ্যাপলিকেশন মেনিফেস্ট দুইটা এডিশন পরে যোজন। প্রথমত, ক্লাসের নাম ডিক্লেয়ার করুন যা আপনার বয়াকআপ এজেন্ট হিসাবে কাজ করে, এবং তারপর উপরের চিত্রটি অ্যাপলিকেশন ট্রাগার একটি চাইল্ড এলিমেন্ট হিসাবে যুক্ত করুন। আপনার বয়াকআপ এজেন্টকে TheBackupAgent হিসাবে আখ্যায়িত করা হবে, এখানে একটি উদাহরণ আছে এই ট্যাগ অন্তর্ভুক্ত করে মেনিফেস্টে দেখতে কেমন হবে:

```
<application android:label="MyApp"
    android:backupAgent="TheBackupAgent">
    ...
    <meta-data android:name="com.google.android.backup.api_key"
        android:value="ABcDe1FGHij2KImN3oPQRs4TUVW5xYZ" />
    ...
</application>
```

আপনার ব্যাকআপ এজেন্ট লিখুন (রাইট)

আপনার ব্যাকআপ এজেন্ট তৈরী করার সহজ উপায় হচ্ছে `BackupAgentHelper` পরিসারিত করা। এই সাহায্যকারী ক্লাস তৈরী করাটা একটি সহজ পরিকল্পনা। পূর্ববর্তী ধাপে পরমেনিফেস্ট (এই উদাহরনে, `TheBackupAgent`) ব্যবহার করার মতো করে একটি নাম দিয়ে শুধু ক্লাস তৈরী করুন, এবং `BackupAgentHelper` পরিসারিত করুন। তারপর `onCreate()` ওভাররাইড করুন।

`onCreate()` পদ্ধতির মধ্যে একটি `BackupHelper` তৈরী করুন। নির্দিষ্ট টিকিছু ডাটার ব্যাকআপ করতে এই হেল্পারসঙ্গে লেগিবেশ ক্লাস। অ্যান্ডরয়েড ফ্রেমওয়ার্কের বর্তমানে এইরকম দুইটাকা ক্লাস অন্তর্ভুক্ত করে: `BackupAgentHelper` এবং `SharedPreferencesBackupHelper`। আপনি হেল্পার তৈরী করার পর এবং আপনি যে ডাটা ব্যাকআপ করতে চান তার প্রতি নিবেদন করার পর, `addHelper()` পদ্ধতি ব্যবহার করে শুধু এটাকে ব্যাকআপ এজেন্ট হেল্পারের যুক্ত করে যুক্ত করে দিন, একটি কিস্যুক্ত করে দিন যা পরবর্তীতে ডাটা পূরণরূদ্ধার করতে ব্যবহৃত হবে। অধিকাংশ ক্ষেত্রে সম্ভ্রণর বাসন্তবায়ন সম্ভবত কেডের ১০ লাইন।

এখানে একটি উদাহরণ আছে যা একটি হাইস্কারফাইলকে ব্যাকআপ করে:

```
import android.app.backup.BackupAgentHelper;
import android.app.backup.FileBackupHelper;

public class TheBackupAgent extends BackupAgentHelper {
    // The name of the SharedPreferences file
    static final String HIGH_SCORES_FILENAME = "scores";

    // A key to uniquely identify the set of backup data
    static final String FILES_BACKUP_KEY = "myfiles";

    // Allocate a helper and add it to the backup agent
    @Override
    void onCreate() {
        FileBackupHelper helper = new FileBackupHelper(this, HIGH_SCORES_FILENAME);
        addHelper(FILES_BACKUP_KEY, helper);
    }
}
```

নমনীয়তা যুক্ত করার জন্য, `FileBackupHelper` এর কনস্ট্রাক্টর ফাইলনেমের একটি ভেরিয়েবল নামবার নিতে পারে। আপনি শুধু একটি হাইস্কারফাইল এবং একটি টেমপ্লেটের সংফাইল সহজে ব্যাকআপ রাখতে নীচের মতো করে শুধু একটি অতিরিক্ত প্যারামিটার যুক্ত করে দিতে পারেন:

```
@Override
void onCreate() {
    FileBackupHelper helper = new FileBackupHelper(this, HIGH_SCORES_FILENAME, PROGRESS_FILENAME);
    addHelper(FILES_BACKUP_KEY, helper);
}
```

পিপারফোরেন্সর ব্যাকআপ করা একইভাবে সহজ। একটি `FileBackupHelper` আপনি যেভাবে করেছেন ঠিক সেভাবেই একটি `SharedPreferencesBackupHelper` তৈরী করুন। এই ক্ষেত্রে, কনস্ট্রাক্টরের ফাইলনেম যুক্ত করার বদলে আপনার অ্যাপলিকেশন দবার ব্যবহৃত শেয়ারপিপারফোরেন্সগ্রেপের নাম যুক্ত করুন। এখানে যদি একটি ফ্লাটফাইলের বদলে হাইস্কারফাইল পিপারফোরেন্স বাসন্তবায়ন করে তাহলে আপনার ব্যাকআপ এজেন্ট হেল্পার দেখতে কেমন হবে তার একটি উদাহরণ আছে:

```

import android.app.backup.BackupAgentHelper;
import android.app.backup.SharedPreferencesBackupHelper;

public class TheBackupAgent extends BackupAgentHelper {
    // The names of the SharedPreferences groups that the application maintains. These
    // are the same strings that are passed to getSharedPreferences(String, int).
    static final String PREFS_DISPLAY = "displayprefs";
    static final String PREFS_SCORES = "highscores";

    // An arbitrary string used within the BackupAgentHelper implementation to
    // identify the SharedPreferencesBackupHelper's data.
    static final String MY_PREFS_BACKUP_KEY = "myprefs";

    // Simply allocate a helper and install it
    void onCreate() {
        SharedPreferencesBackupHelper helper =
            new SharedPreferencesBackupHelper(this, PREFS_DISPLAY, PREFS_SCORES);
        addHelper(MY_PREFS_BACKUP_KEY, helper);
    }
}

```

আপনার যত ইচ্ছা ততগুলো বয়স্ক আপ হেল্পার ইনস্টল করুন আপনার বয়স্ক আপ এজেন্টে হেল্পার যুক্ত করতে পারবেন, কিন্তুমেন রাখবেন যে প্রতিটার মধ্যে মাত্র একটি আপনার পেরে যাজন হবে। একটি FileBackupHelper সকল ফাইলকে চািলত করতে পারবে যা আপনার বয়স্ক আপের পেরে যাজন, এবং একটি SharedPreferencesBackupHelper সকল শেয়ারিং পেরফারেন্স পেরুপকে চািলত করে যা আপনার বয়স্ক আপ করাপেরে যাজন।

বয়স্ক আপডেইট কয়েস্ট

একটি বয়স্ক আপডেইট কয়েস্ট করার জন্য, শুধুমাত্র BackupManager এর একটি ইনস্ট্যান্স তৈরি করুন, এবং এর dataChanged() পদ্ধতি কল করুন।

```
import android.app.backup.BackupManager;
...

public void requestBackup() {
    BackupManager bm = new BackupManager(this);
    bm.dataChanged();
}
```

এই কল বয়স্ক আপডেইট ম্যানজারকে জানান যে কল উইথ বয়স্ক আপডেইট করার জন্য সেখানে একটি ডাটা আছে। ভবিষ্যতে তাকিছু বিষয়ে কল বয়স্ক আপডেইট ম্যানজার তখন আপনার বয়স্ক আপডেইটের onBackup() পদ্ধতি কল করে। অতিরিক্ত নেটওয়ার্ক একটি ভিউটির ঘটনার কারণে চিন্তিত না হয়ে যখনই আপনার ডাটা পরিবর্তন হবে আপনি কলটি করতে পারেন। আপনি যদি একটি বয়স্ক আপডেইট ঘটনার পূর্বে একটি বয়স্ক আপডেইট দুইবার রিকয়েস্ট করেন, তাহলে বয়স্ক আপডেইট একবারই ঘটবে।

একটি বয়স্ক আপডেটের স্টার করুন

সাধারণভাবে আপনাকে কখনই ম্যানুয়ালি একটি রিস্টোরের
রিকোয়েস্ট করতে হবে না, যেহেতু এটি স্বয়ংক্রিয়ভাবে হবে যখন আপনার
অ্যাপলিকেশন একটি ডিভাইসে ইনস্টল করা হবে। কিন্ত যদি একটি
ম্যানুয়ালি রিস্টোর সিকিউরিটি করার পরেয়োজন হয়, শুধু
`requestRestore()` পদ্ধতিকে কল করুন।

অধিকাংশ গুগল ক্লাউড মেসেজিং তৈরী করা

(<http://developer.android.com/training/cloudsync/gcm.html>)

অধিকাংশ গুগল ক্লাউড মেসেজিং (GCM) অ্যান্ডরয়েড ডিভাইসে বিনামূলীয়ে মেসেজ সেন্ড করার একটি সার্ভিস। GCM মেসেজিং ভালোভাবে ইউজার এক্সপেরিয়েন্সকে বৃদ্ধি দ্বন্দ্ব করতে পারে। রিডিউক জাগাতে কোন ব্যাটারি পাওয়ার খরচ না করেই এবং যখন সেখানে কোন আপডেট না থাকে তখন সার্ভার পাওলা করে আপনি আপনার অ্যাপলিকেশন সময়েপায়ে যাগী করতে পারেন। এছাড়াও GCM আপনাকে একটি একক মেসেজ ১০০০ পর্যন্ত রিসিপিয়েন্ট যুক্ত করতে দেয়, আপনাকে বহু বৈজ্ঞানিক তথ্য যাগ যাগ করতে দেয় যখন পরে যাঁজন হয়, যখন আপনার সার্ভারের কাজের চাপ কমিয়ে আন।

এই অনুশীলনী আপনার অ্যাপলিকেশনের মধ্যে GCM একীভূত করার কিছু ভালো চর্চা নিয়ে আলোচনা করে এবং মনে করে আপনি ইতিমধ্যে এই সার্ভিসের মৌলিক বাস্তবায়নের সাথে পরিচিত। যদি এক্ষেত্রে তানা হয় আপনি GCM demo app tutorial (<http://developer.android.com/google/gcm/demo.html>) পড়তে পারেন।

কায়র করীভাে বমালি ল্টকাস্ট মেসেজ সেন্ড (পাঠােনা) করুন

GCM এর একটি কায়র করীভাে বিশেষ টাইপ হচ্ছে একটি একক মেসেজের জন্য ১০০০ রিসিপিএন্ট (গ্রাহক) পর্যন্ত সাপোর্ট করে। এই সাধারণ আপনার পুনরাবৃত্তি ইউজার বেজিগুরুতব পুনর মেসেজ পাঠােনার কাজটি সহজ করে দিয়েছে। উদাহরণস্বরূপ, মনে করুন আপনার ১,০০০,০০০ ইউজারকে একটি মেসেজ পাঠাতে হবে এবং আপনার সার্ভার ও পরিত সেকেন্ড ৫০০ পাঠাতে পারে। আপনি যদি পরিত মেসেজ মাত্র একটি একক রিসিপিএন্ট দিয়ে পাঠান, সেকেন্ডের সময় লাগবে $1,000,000/500 = 2,000$ সেকেন্ড বা প্রায় আধা ঘন্টা। কিন্তু তু পরিত মেসেজ যদি ১০০০ রিসিপিএন্ট যুক্ত (অ্যাট্যাচ) করা যায় তাহলে $1,000,000$ জনকে একটি মেসেজ পাঠাতে মোট সময় কেমেগিয়ে হবে $(1,000,000/1000)/500 = 2$ সেকেন্ড। এটা শুধু কায়র করই নয়, যথাসময়ে ডাটাপাঠােনার জন্য গুরুতব পূর্ণর, যেমন পরাকৃতক বিপর্যয়ের সতর্কতা বা থেলারার সেকার যেখানে ৩০ মিনিটের বিরতি তথ্যকে অপরেয়োজনীয় করে ফলেত পারে।

এই কমরপদ্ধতির সুবিধা নিয়াসহজ। আপনি যদি Java'র জন্য GCM helper library (<http://developer.android.com/google/gcm/gs.html#libs>) ব্যবহার করে থাকেন, তাহলে একটি একক রেজিস্ট্রেশন আইডি এর পরিবর্তে send বা sendNoRetry পদ্ধতিতে শুধু রেজিস্ট্রেশন আইডির একটি খরসংসংগ্রহ সরবরাহ করুন।

```
// This method name is completely fabricated, but you get the idea.
List regIds = whoShouldISendThisTo(message);

// If you want the SDK to automatically retry a certain number of times, use the
// standard send method.
MulticastResult result = sender.send(message, regIds, 5);

// Otherwise, use sendNoRetry.
MulticastResult result = sender.sendNoRetry(message, regIds);
```

Java'র বদলে একটি ল্যাঙ্গুয়েজের মধ্যে ওই GCM সাপোর্ট বাস্তবায়ন করার জন্য, নিম্নলিখিত হেডারগুলো দিয়ে একটি HTTP POST রিকোয়েস্ট তৈরী করুন:

- Authorization: key=YOUR_API_KEY
- Content-type: application/json

তারপর যে প্যারামিটার আপনি একটি অবজেক্টের মধ্যে চান তা এনকোড করুন, সকল রেজিস্ট্রেশন আইডির 'registration_ids' অধীনে তালিকা কর। নিম্নলিখিত কোড চিত্রটি উদাহরণ হিসাবে দেয়া আছে। registration_ids ব্যতীত আর সকল প্যারামিটার ঐচ্ছিক, এবং data এর মধ্যে থাকা আইটেমগুলো ইউজার নির্ধারণের পলেডকে পরিতিনিধিত্ব করে, GCM নির্ধারণের প্যারামিটার নয়। এই HTTP POST মেসেজের জন্য শেখিবেন্দু হবে <https://android.googleapis.com/gcm/send>।

```
{ "collapse_key": "score_update",
  "time_to_live": 108,
  "delay_while_idle": true,
  "data": {
    "score": "4 x 8",
    "time": "15:16.2342"
  },
  "registration_ids":["4", "8", "15", "16", "23", "42"]
}
```

মাল্টিপল টাস্কিং GCM মেসেজের ফরমেট সম্পর্কে আরও বিস্তারিত জানতে
GCM গাইডের অধ্যায় Sending Messages (<http://developer.android.com/google/gcm/gcm.html#send-msg>)
দেখুন।

যে মেসেজটি পরিত্যক্ত স্থাপিত হতে পারে সেটাবন্ধকরে দিন

GCM মেসেজ কখনও কখনও টিকল হতে পারে, ফোর্শ ডাটার জনয সাভর এর সাথে যে গাগাগ করতে তে মা বাইল অয্যাপিল কেশনকে বল। GCM এ এই পরিসি স্থিত জনয বন্ধ হতে সক্ষম মেসেজে তৈরী করা সম্ভব, যার মধ্যে পুরাতনের স্থানে নতুন মেসেজ পরিত্যক্ত স্থাপিত হয়। এখানে একটি থেলার সেস্কোরের উদাহরণ দেয়া হলে। যদি আপনি আপডেটেস্কোর সহকারে একটি নিদির্ষ্ট থেলা অননুসারে সকল ইউজারকে একটি মেসেজ পাঠান, এবং ১৫ মিনিট পরে সেস্কোরের আপডেটে মেসেজ যায়, পর্থমে মেসেজটির আর গুরুত্ব থাকবে না। যে কোন ইউজারের জনয যে এখনও পর্থমে মেসেজটি গ্রহণ করেনা, তাকে দুইটা মেসেজ ইনাপাঠানোর কানকারন নেই এবং ডিভাইসকে দুইবার পরিত্যক্ত করা করতে (এবং ইউজারকে সম্ভাব্য সতর্ক করা) বাধ্য করা যখন মেসেজের মাত্র একটিই তখনও গুরুত্ব পূর্ণ।

যখন আপনি একটি বন্ধকিন ধারণ করেন, যখন একই ইউজারের জনয GCM সাভর এর মধ্যে মালি টপল মেসেজকে সারিবদ্ধ করে, শুধুমাত্র শেষেরটা সাথে যে কোন পরদত্ত কলাপসিক ডিলভারি করা হয়। থেলার সেস্কোরের মধ্যে তা অবস্থার জনয, এটা ডিভাইসকে অপর্য়েয়োজনীয় কাজ করা এবং সম্ভাব্য ওভারনেটিং ফাইকরা থেকে রক্ষা করে। এই অবস্থায় এটা একটি সাভরার সিঙ্ককে সম্পৃক্ত করে (মেইল চেক করার মধ্যে), এটা ডিভাইসকে করতে হবে এমন সিঙ্ক কর সংখ্যা কমিয়ে আনতে পারে। উদাহরণ হিসাবে, যদি সাভর এর ১০টি ইমেইল অপেক্ষা করে থাকে এবং দশটি "নতুন ইমেইল" GCM টিকলস পাঠানো হয়ে থাকে, এটার একটাই পর্যোজন, যেহেতু এটা একবার সিঙ্ক করে।

এই বিশেষ ট্যব ব্যবহার করার জনয, আপনার বিহরাগমন মেসেজে একটি কলাপসিক যুক্ত করুন। আপনি যদি GCM হেল্পার লাইব্রেরী ব্যবহার করতে থাকেন, মেসেজ ক্লাসের collapseKey(String key) পদ্ধতি ব্যবহার করুন।

```
Message message = new Message.Builder(regId)
    .collapseKey("game4_scores") // The key for game 4.
    .ttl(600) // Time in seconds to keep message queued if device offline.
    .delayWhileIdle(true) // Wait for device to become active before sending.
    .addPayload("key1", "value1")
    .addPayload("key2", "value2")
    .build();
```

যদি হেল্পার লাইব্রেরী ব্যবহার না করে থাকেন, শুধু আপনি তৈরী করেছেন এমন POST হেডার একটি ভেরিফেবল যুক্ত করুন, সাথে ফিল্ডনেম হিসাবে collapse_key, এবং ভ্যালু হিসাবে ওই আপডেটের সেট জনয ব্যবহার করা সিটর।

সরাসরি GCM মেসেজের মধ্যে ডাটা বসানো

কখনও, GCM মেসেজ একটি টিটকল হওয়া কেবল ঠিক, বা ডিভাইসে সিনক্রোনাইজ করা না হলে, সান্ডার্সের ক্যাথোড একটি ফ্রেশ ডাটা অপেক্ষাকৃত। কিন্ত, একটি GCM মেসেজ 4kb সাইজ পর্যন্ত হতে পারে তাই কখনও কখনও এটা শুধু GCM মেসেজের মধ্যে ডাটা সেন্ড করে নিজেই অর্থপূর্ণ হয়, যাতে ডিভাইসকে মেসেজটি সান্ডার্সের সাথে যোগাযোগ করার পরে যাজন না হয়। যে অবস্থায় নীচের বিবৃতিগুলি সঠিক সেই অবস্থায় এই পদ্ধতি বিবেচনা করুন:

- 4kb সীমার মধ্যে সকল ডাটা ফিট হবে
- প্রতিটা মেসেজই গুরুত্বপূর্ণ এবং সংরক্ষণ করা উচিত
- একটি একক "সান্ডার্সের নতুন ডাটা" টিটকল এর মধ্যে মাল্টিপল GCM মেসেজ কলাপস করানো কান মানে তৈরী করেনা।

উদাহরণস্বরূপ, একটি টানার-বেজড নেটওয়ার্ক গেমের মধ্যে শটর মেসেজ বা এনেকাউন্ট প্লয়ার হচ্ছে একটি GCM মেসেজের মধ্যে সরাসরি ডাটা বসানোর জন্য ভালো উদাহরণ। ইমেইল হচ্ছে খারাপ উদাহরণ, সেহেতু মেসেজ কখনও কখনও 4kb এর চাইতে বড় হতে পারে, এবং সান্ডার্সের তাদের জন্য অপেক্ষাকৃত ইমেইলের প্রতিটার জন্য ইউজারের একটি GCM মেসেজের পরে যাজন নেই।

আপনি এই পদ্ধতিও বিবেচনা করতে পারেন যখন মাল্টিটাস্ট মেসেজ সেন্ড করা হয়, সুতরাং আপনাকে ইউজার বেজ জুড়ে প্রতিটা ডিভাইসকে তা ক্রমিকভাবে আপডেটের জন্য আপনার সান্ডার্সের ক্রমিক হট করতে বলতে হবে না। এই ক্রমিক অধিক পরিমাণে ডাটা সেন্ড করার জন্য কয়েকটি কারণে উপযুক্ত নয়:

- মেসেজ সহকারে একটি একক ডিভাইসকে স্পর্শমি করাকে বাজে বা দ্রুত লোড করা অ্যাপের বিরত রাখতে রেস সীমানা নির্ধারণ স্থানে আছে।
- মেসেজ নির্ধারণ টিকের মান সারে পেছা যে তার কোন নিশ্চয়তাই
- আপনি যত দ্রুত মেসেজ বাইরে পাঠিয়েছেন তা সেভাবেই পেছা যে তার নিশ্চয়তাই। এমন কি যদি ডিভাইস একটি GCM মেসেজ একসেসে নেড গ্রহণ করে, ম্যাক্স. 1k এ যাচ্ছে 8kbps অথবা ১৯৯০'র দশকের পরথম দিকে হাম ডায়াল-আপ ইন্টারনেটের গতি সম্পর্কিত। গুলে প্লতে আপনার অ্যাপের টি ইউজারের কাছে এই সব কাজগুলোকে প্রতিফলিত করে। যখন যথার্থভাবে ব্যবহৃত হয়, সরাসরি GCM মেসেজ স্থাপিত ডাটা আপনার অ্যাপলিকেশনের অজিত গতিতে আরও বৃদ্ধি করতে পারে, এটাকে সান্ডার্সের দিকে রাউন্ড ট্রিপের পরিহার করতে দিয়ে।

GCM মেসেজের পরিত Intelligently রিয়েক্ট করা

আপনার অ্যাপ্লিকেশনের শুধুমাত্র ইনকামিং GCM মেসেজের পরিত
রিয়েক্ট করা উচিত নয়, *intelligently* ও রিয়েক্ট করা উচিত। কীভাবে রিয়েক্ট
করতে হয় তা নিভর করে এর পেরক্ষাপটের উপর।

বিবরিক্তকর করে বননা

যখন এটা আপনার ইউজারকে ফর্শ ডাটার সতর্কতা নিয়ে আসে, "উপকারী" ("useful") অবস্থাতে থাকে "বিবরিক্তকর" ("annoying") চলে আসাটা সহজ। যদি আপনার অ্যাপলিকেশন স্ট্যাটাসবারে নোটিফিকেশন ব্যবহার করে থাকেন, নির্দিষ্ট তথ্য তৈরী না করে আপনার যেটা আছে সেটাকেই আপডেট করুন (<http://developer.android.com/guide/topics/ui/notifiers/notifications.html#Updating>)। আপনি যদি ইউজারকে সতর্ক করতে বিপবাহাই বর্ট করতে চান তাহলে একটি টাইমার সেট করার কথাটি বিবেচনা করতে পারেন। অ্যাপলিকেশনকে একমিনিটে একবারের বেশী সতর্ক করতে দিবে বননা, এটা ইউজারকে আপনার অ্যাপলিকেশনকে আন-ইনস্টল করতে, ডিভাইস বন্ধ করতে বা ডিভাইস ছুড়ে ফেল দিতে পরেরা চিত করতে পারে। বিবরিক্তকর হবে বননা

স্মার্ট ভাবে সিস্টেম করুন কঠিন ভাবে নয়

যখন ডিভাইসে একটি ইনডিউক্টর (মানদণ্ড) হিসাবে GCM ব্যবহার করা যা
সাবর্যে তে ডাউনলোড করা পরে যাজন, এ বিষয়ে আপনার
অ্যাপলিকেশনকে স্মার্ট হতে সহায়তা করতে মনে রাখবেন আপনার 4kb
মেটাডাটা আছে। উদাহরণস্বরূপ, যদি আপনার একটি ফিডরিডি অ্যাপ
থাকে এবং আপনার ইউজারের ১০০ ফিডস আছে যা তার অনুসরণ করে, ডিভাইস
সাবর্যে তে ক্যাডাউনলোড করে সে সম্বন্ধে একটুকু স্মার্ট হতে
সহায়তা করে! নীচের উদাহরণটি দেখুন পেলে ডের মধ্যে আপনার
অ্যাপলিকেশনকে কানমেটাডেউটপাঠানো হয়েছে এবং কীভাবে
অ্যাপলিকেশন পরিত্যক্ত করা করতে পারে: স্মার্ট ভাবে সিস্টেম করুন
কঠিন ভাবে নয়

- refresh — আপনার অ্যাপলিকেশনকে মূলত এটার অনুসরণকৃত পরিত্যা
ফিডের একটি ডাম্পেরিরেকায়েস্ট করতে বলা হয়। আপনার অ্যাপের হয়
১০০ ভিন্ন সাবর্যের ফিডরিেকায়েস্ট পাঠানোর পরে যাজন, নয়তো
যদি আপনার সাবর্যের আপনার একটি স্যোগকারী থাকে, পুনরুদ্ধার
করতে একটি রিরেকায়েস্ট পাঠান, ১০০ ভিন্ন ফিড থেকে সাম্প্রতিক
ডাটাবান্ডল এবং স্থানান্তরিত করা, পরিত্যা সময় একটি আপডেট।
- refresh, feedID — Better: আপনার অ্যাপ আপডেটের জন্য একটি নির্দিষ্ট স্ট্রিফিডকে
চেক করতে জানে।
- refresh, feedID, timestamp — Best: যদি ইউজার GCM মেসেজ পেছার পূর্বেই
ম্যানুয়েলিরিরেফরশ ঘটায়, অ্যাপলিকেশন অতি সাম্প্রতিক
পেস্টরটাইমস্ট্যাম্প তুলনাকরতে পারে এবং নির্দিষ্টারন করতে
পারে যে এটার কানিকছু করা পরে যাজন নেই।

ক্লাউডে ডে স্ত ক রা স ম্প ক র ত সা ংয িষ র ক বিষ য়ে র স ম াধ ান ক রা

(<http://developer.android.com/training/cloudsave/conflict-res.html>)

এই আর্টিকেলটি বর্ণনা করবে কীভাবে অ্যাপের জন্য একটি রেবাস্ট কনফ্লিক্ট রেজালুয়শন কৌশল ডিজাইন করতে হয় যা Cloud Save service (<https://developers.google.com/games/services/common/concepts/cloudsave>) ব্যবহার করে ক্লাউডে ডাটাসেভ করে। ক্লাউডে স্ত সাঁভ র্ স আপন াকে গুগলের সাঁভ র্ স একটি অ্যাপলিকেশনের পর্তিটাইউজ ারে র জনয্ অ্যাপলিকেশন ডাটাসেভ করতে দেয়। আপন ার অ্যাপলিকেশন ক্লাউডে স্ত APIs ব্যবহার করে অ্যান্ডরয়েড সাঁভ র্ স, iOS ডিভাইস, বা ওয়েব অ্যাপলিকেশন থেকে এই ইউজার ডাটাপুনরুদ্ধ ার এব ং আপডেট করতে প ারে।

ক্লাউডে স্তে স্ত এব ং লোড করার পর্তিকর্য াখুবই সহজ: এটা বাইট বিনয্ াস থেকে এব ং বাইট বিনয্ াসের পর্তি তে প্লয়ারের ডাটা ধ ারাব াহিকভাবে ক র্ম ানু স ারে স াজ ানে ার বিষ য় এব ং ঐ বিনয্ াসটি ক্লাউডে স্তে স্ত ার ক রা। কিন্তু, যখন আপন ার ইউজ ারে র ম াল্ টিপল (বহু মুখি) ডিভাইস থেকে এব ং এর মধ্যে দুই বা তে াধিক ক্লাউডে স্ত ডাটা সেভ করার পর্য াস নেয়, সেভ করার ক রাটা সাঁয িষ র ক হতে প ারে, এব ং আপন াকে অবশ্য ই িঠ ক করতে হবে কীভাবে এর স ম াধ ান ক রা য ায়। আপন ার ক্লাউডে স্ত ডাটার গঠন ব্যবহ ার কভাবে দেখ ায় আপন ার কনফ্লিক্ট রেজালুয়শন কতটা শক্তিশালী হতে প ারে তাই আপন াকে অবশ্য ই আপন ার কনফ্লিক্ট রেজালুয়শন লজিককে পর্তিট াকে স ি িঠ কভাবে চ ালিত করতে দেয়ার জন্য সতকর্ত ার সাঁথে আপন ার ডাটা ডিজাইন করতে হবে।

আর্টিকেলটি শুরুর কের িকছু তরু িটপূর্ণ পদ্ধতি আলাচনা দি়ে এব ং ব্যবহ ার কের ক াথ ায় তাে দর য াটি তি আেছ। তারপর এটা সাঁয িষ র বন্ধ করতে একটি স ম াধ ান উপস্ থ াপন করে। এই আলাচন ায় গে মকে ফেকাস করা হ়ে়েছ, কিন্তু আপনি একই নিয়ম অন্য অ্যাপের য়ে াগ করতে প ারেন যা ক্লাউডে স্ত ডাটা সেভ করে।

কনফ্লিক্ট সম্ভব করে দেওয়া

`OnStateLoadedListener` পদ্ধতি গুলোর সাহায্যে একটি অ্যাপ্লিকেশনের স্টেট ডাটাবেজের জন্য দায়িত্ব পূরণ করা যায়। কলব্যাক `OnStateLoadedListener.onStateConflict` ইউজারের ডিভাইসে লোকাল স্টেট এবং ক্লাউডের মধ্যে স্টোর করা স্টেটের মধ্যে পার্থক্যের সনাক্তকরণ এবং সমাধান করতে আপনার অ্যাপ্লিকেশনের জন্য একটি মেসেজিং সিস্টেম সরবরাহ করে:

```
@Override
public void onStateConflict(int stateKey, String resolvedVersion,
    byte[] localData, byte[] serverData) {
    // resolve conflict, then call mAppStateClient.resolveConflict()
    ...
}
```

এই ক্ষেত্রে আপনার অ্যাপ্লিকেশনকে অবশ্যই ইচ্ছা করে দেওয়া উচিত যে স্টেটকে রাখা উচিত, অথবা এটা একটানতুন ডাটাবেজের মাধ্যমে পুনরায় একটির (মাজর) করা ডাটাবেজের প্রতিস্থাপন করে। সনাক্তকরণের জায়গায় লজিক বাস্তবায়ন করা আপনার উপর নির্ভর করে।

এটা অনুধাবন করা জরুরী যে ক্লাউডে স্টোরেড সার্ভিস বাকগরান্ডের মধ্যে ডাটাবেজের নাইজ করে। অতএব, আপনার নিশ্চিত করা উচিত যে আপনার অ্যাপ এই কলব্যাক কনফ্লিক্টের বাইরে রিসিভ করার জন্য পুরোপুরি আচ্ছন্ন আপন শুরুর ডাটাবেজের রিইনিয়েল। বিবেচনা করে যদি গুলে পল সার্ভিস অ্যাপ্লিকেশন বাকগরান্ডে একটি সনাক্তকরণের বিষয় সনাক্ত করে, কলব্যাক পরবর্তী সময়ে ডাটাবেজের চেঞ্জের সময় কল করে, যা তৎক্ষণাত পুনরায় হবেনা যতক্ষণ পরবর্তী সময়ে ইউজার অ্যাপ শুরুর না করে।

অতএব, ক্লাউডে স্টোরেড ডাটাবেজের ইউজারের ইউজার-নির্ভর (context-independent): পুরোপুরি দুইটা কনফ্লিক্টের স্টেট, ডাটাবেজের স্টেটের মধ্যে আছে শুধুমাত্র এমন ডাটাবেজের বাকগরান্ডের আপনাকে অবশ্যই সনাক্তকরণের সমাধান করতে পারবে, বহিরাগত কোন কনফ্লিক্টের সাথে পরামর্শ না করেই।

সাধারণ কেসগুলো পরিচালিত করা

এখানে কনফ্লিক্ট (সংঘাত) রেজুয়লেশনের কিছু সাধারণ কেস দেয়া হলো। অনেক অ্যাপের জন্য, এই সকল কেসগুলোর কোন একটির বিকল্প গ্রহণ করা যেতে পারে:

- পূরাতনের চেয়ে নতুন ভালো। কিছু কেসে তবু, পুরাতন ডাটার স্থানে নতুন ডাটা পরিস্থাপন হওয়া জরুরী। উদাহরণস্বরূপ, যদি ডাটা একটি চিরিতের শাটের রঙের পল্যাের পছন্দে বিষয়টার পরিস্থিতি নিধিতব করে, তখন চলতি পছন্দটি পুরাতন পছন্দকে খারিজ করে দেওয়া উচিত। একে কেসে তবু আপন ক্লাউড সেভ ডাটাতে টাইমস্ট্যাম্প সেটার করার বিষয়টা বেছে নিতে পারেন। যখন কনফ্লিক্ট (সংঘাত) নিরসন হবে তখন অতি সাম্প্রতিক টাইমস্ট্যাম্প দিয়ে ডাটা সেট বেছে নিতে হবে (একটি গ্রহণযোগ্য কলকব্যবহার করার বিষয়টা সর্বগণ্য রাখবেন এবং টাইমজেনের ভিন্ন মতের বিষয়ে সতর্ক থাকবেন)।
- ডাটার একটি সেট পরিষ্কারভাবে আরেকটির চেয়ে ভালো। অন্য কেসে তবু, কোন ডাটা সেবের ক্যুশ্ট হিসাবে বিনধরার হতে পারে সে বিষয়ে এটা সব সময় পরিষ্কার। উদাহরণস্বরূপ, যদি ডাটা একটি রেসিগ্গে মের মধ্যে পল্যাের বেস্ট টাইমকে পরিস্থিতি নিধিতব করে, তখন এটা পরিষ্কার যে, কনফ্লিক্টের (সংঘাতে) কেসে তবু, আপনার উচিত বেস্ট টাইম (ক্যুদরতর) ধারণ করা।
- ইউনিয়ন দ্বারা একটি তরুত করা (মাজরু করা)। দুইটা সাংঘিষরু কেসের একটি সমনব্য (ইউনিয়ন) গণনা করার মাধ্যমে সাংঘিষরু অবস্থার সমাধান করার সম্ভব হতে পারে। উদাহরণস্বরূপ, যদি আপনার ডাটা লেভেলের সেট পরিস্থিতি নিধিতব করে যা পল্যার আনলক করে, তখন সমাধান করা ডাটা সাধারণভাবে দুইটা কনফ্লিক্ট সেটের ইউনিয়ন হয়। এই উপায়ে, পল্যার রায়ে লেভেল গুলো আনলক করেছে তা হারিয়ে ফেলবেন না। CollectAllTheStars নমুনা গেমেটি এই কেসগুলোর একটি ভেরিয়েন্ট (বিকল্প) ব্যবহার করে।

আরও জি টি লকেসের জনস্বার্থে একটি কৌশল ডিডাইন করা

আরও জি টি লকেস যেটি যখন আপনার গেমপ্লেয়ারকে পরিবর্তনশীল
(ফ্রাঞ্জবল) আইটেম বা ইউনিট সংগ্রহ করতে দেয় যেমন সোনার কয়েন
(মুদ্রা) বা এক্সপেরিয়েন্স পয়েন্ট। একটি অনুমেয়
(হাইপোথেটিক্যাল) গেম বিবেচনা করা যাক, যার নাম কয়েন রান, একটি
অসমরান্য যথার্থ লক্ষ্য হেট্‌ছ কয়েন সংগ্রহ করা এবং খুব ধিনি হওয়া।
পর্যাপ্ত হলেই সংগ্রহ প্লেয়ারের পিগি বয়সেও জমা হবে।

নিচের অধ্যায় মাল্টিপল ডিডাইনসের মধ্যে সিস্টেম কনফ্লিক্ট
সমাধানের জনস্বার্থে তিনটুকো কৌশল আলোচনা করে: তার মধ্যে দুইটা ডাউনলাই
কিন্তু চুড়ান্তভাবে সফলতার সাথে সকল পরিস্থিতি সমাধান করতে
বর্থ হয় এবং একটি চুড়ান্ত সমাধান যা যে কোন ডিডাইনসের সংখ্যার
মধ্যে যেকোন সংঘাত (কনফ্লিক্ট) বয়স স্থাপনা করতে পারে।

পৰ্ৱৰ্তী পৰ্ৱৰ্তী চেষ্টা: শুদ্ধ সবৰ্ণ মেটিটি (টেটাল) সেট কৰুন

পৰ্ৱৰ্তী চিন্তাত, এটা মনে হৈছে পৰে যেকো ডাউডে সন্তোষিত
শুদ্ধ মাতৰ বৰ্ণাঙ্ক কয়েনৰ সংখ্যা হওয়া উচিত। কিন্তু যদি
যেটুকুৰি বদ্যমান আছে সেটুকুৰি হয়, কনফ্লিক্টে জুয়েলেশন
জোড়ালে ভাবে সীমিত হবে। সবচেয়ে ভালো হৈছে একটি কনফ্লিক্ট
কেষ্টেৰ দুইটা সংখ্যার বড়টা বেছে নেয়া।

টেবিল ১. এব্যাক্থ্যাকৰাচিতৰ্টিবিবেচনাকৰুন। হৈছে পৰে
প্লেয়াৰেৰ পৰাখমিকভাবে ২০টি কয়েন আছে এবং তারপৰি ডিভাইস A থেকে
১০টি কয়েন এবং ডিভাইস B থেকে ১৫ কয়েন সংগ্রহ কৰে। তখন ডিভাইস B
ক্লাউডে ডাটাসেটটি সেভ কৰে। যখন ডিভাইস A সেভ কৰাৰ চেষ্টা কৰে,
একটি কনফ্লিক্ট চিহ্নিত হয়। "Store Only the Total"/ "শুদ্ধ সবৰ্ণ মেটিটি সেট
কৰুন" কনফ্লিক্টে জুয়েলেশন এলগরিদম ৩৫ (দুইটা সংখ্যার মধ্যে
বড়টা) লৈখ্য মাধ্যমে কনফ্লিক্ট সমাধান কৰে পৰে।

টেবিল ১. শুদ্ধ কয়েনৰ সবৰ্ণ মেটিটি (টেটাল) সেট কৰা (অসফল কৌশল):

Event	Data on Device A	Data on Device B	Data on Cloud	Actual Total
Starting conditions	20	20	20	20
Player collects 10 coins on device A	30	20	20	30
Player collects 15 coins on device B	30	35	20	45
Device B saves state to cloud	30	35	35	45
Device A tries to save state to cloud. Conflict detected.	30	35	35	45
Device A resolves conflict by picking largest of the two numbers.	30	35	35	45

এই কৌশল অসফল হৈছে পৰে-প্লেয়াৰেৰ বৰ্ণাঙ্ক ২০ থেকে ৩৫ চলে যেত
পৰে, যখন ইউজার আসলেই সবৰ্ণ মেটি ২৫ কয়েন সংগ্রহ কৰে (ডিভাইস A তে ১০
টি কয়েন এবং ডিভাইস B তে ১৫ কয়েন)। সুতরাং ১০টি কয়েন হাৰিয়ে যায়।
ক্লাউডে সেভৰ মধ্যে শুদ্ধ কয়েনৰ সবৰ্ণ মেটিটি (টেটাল) সেট কৰা
একটি রেবাস্ট কনফ্লিক্টে জুয়েলেশন এলগরিদম বাস্তুবায়নের জন্য
যথেষ্ট নয়।

দিব্ৰতীয় পৰ্চেষ্টা: সৰৱ্ৰম্যে (টোটাৰল) এবাং ডেল্টা সেট্টাৰ কৰুন

একটি ভিত্তি পদ্ধতি হৈছে যে সমস্ত ডাটাত একেটি অতিরিক্ত ফিল্ড যুক্ত কৰুন: কয়েনৰ সংখ্যা যোগ হৈছে (ডেল্টা) সৰৱ্ৰম্যে বার্তাৰে। এই পদ্ধতিতে সমস্ত ডাটা একটি (T, d) কত্বৰূপে পৰিচালিত হৈছে। যিহেতু T হৈছে কয়েনৰ ম্যেট সংখ্যা এবাং d হৈছে মাতৰ ইয়াক হওয়া কয়েনৰ সংখ্যা।

এই কাঠামো দিয়া আপনৰ কনফ্লিক্ট ৰেজলুয়শ্যন এলগরিদমৰ আৰও বিলম্ব (ৱেবাস্ট) জায়গা থাকে, যেভাবে নিচে বখাখ্যাকৰা হৈছে। কিন্তু এই পদ্ধতি এখনও আপনৰ অখাপকে পল্যাৱেৰ সাবিতৰক অবস্থার গৰহণযোগ্য চিত্তৰ দিতে পোৱেনা।

এখানে ডাটা অন্তত দুৰূপে কৰিবৰ জনয় কনফ্লিক্ট ৰেজলুয়শ্যন এলগরিদম আছে:

- Local data: (T, d)
- Cloud data: (T', d')
- Resolved data: $(T' + d, d)$

উদাহরণস্বৰূপে, যখন আপনি লোকাল সেট্টা (T, d) এবাং ক্লাউড সেট্টা (T', d') মধ্য একটি কনফ্লিক্ট পান, আপনি এটাকে $(T+d, d)$ হিসাবে সমাধান কৰিতে পোৱেন। এটা যাবে বাৰ্তা হৈছে আপনি আপনৰ লোকাল ডাটাত ডেল্টা নিয়েছে এবাং এটাকে ক্লাউড ডাটাৰ মধ্য একটিৰূপে কৰে, আশা কৰে যে, এটা সঠিকভাবে যে কোনে সবর্ণ কয়েনৰ জনয় বখাখ্যাকৰে বা অনয় ডিভাইসে সংগৃহিত হৈছে।

এই পদ্ধতি আশাপৰ্দ লাগতে পোৱে, কিন্তু এটা একটি ডাইনামিক মেবাইল এনভায়ৰনমেন্টৰ মধ্য ভেঙে পড়ে:

- ইউজাৰ সেট্টা (অবস্থা) সেত কৰিতে পোৱে যখন ডিভাইস অফলাইনে থাকে। এই পৰিবৰ্তন জমা দেয়াৰ জনয় শেৰুনী বদ্ধ হয় যখন ডিভাইস অনলাইনে ফিৰে আসে।
- সিঙকে যেভাবে কাজ কৰে সে উপায় হৈছে যে সবেচেয়ে সমসাময়িক পৰিবৰ্তন যে কোনে পূবৰ বতী পৰিবৰ্তন কেনে নতুন ভাবে লেখে (ওভাৱরাইট)। অথৰা দিব্ৰতীয় রাইট একমাতৰ যাক ক্লাউডে পাঠােনা হৈছে থাকে (এটা যতে থাকে যখন ডিভাইস চুড়ান্ততাবে অনলাইনে আসে), এবাং পৰ্থম রাইটেৰ মধ্য কাৰ ডেল্টা অবহেলিত হয়।

বখাখ্যাকৰাৰ জনয়, টেবিল ২. তে বখাখ্যাকৰা চিত্তৰ টিবিবেচনা কৰুন। ধাৰাবাহিক অপাৰেশ্যনগুলেৰ পৰবৰ্তীতে টেবিলটাত দেখােনা হৈছে, ক্লাউড সেট্টা $(130, +5)$ হৈছে পোৱে। এটাৰ মানে নিধৰাৱিত সেট্টা $(180, +10)$ হৈছে পোৱে। এটা সঠিক নয় কাৰন সৰৱ্ৰম্যে সেট্টাৰ মধ্য ইউজাৰ ডিভাইস A তে ১১০ টি কয়েন এবাং ডিভাইস B তে ১২০ টি কয়েন সংগৰহ কৰেছে। মেট হওয়া উচিত ২৫০ কয়েন।

টেবিল ২, মেট+ডেল্টা কৌশলেৰ জনয় অসফল কেস।

Event	Data on Device A	Data on Device B	Data on Cloud	Actual Total
Starting conditions	(20, x)	(20, x)	(20, x)	20
Player collects 100 coins on device A	(120, +100)	(20, x)	(20, x)	120
Device A saves state to cloud	(120, +100)	(20, x)	(20, x)	120

Player collects 10 more coins on device A	(130, +110)	(20, x)	(120, +100)	130
Player collects 1 coin on device B	(130, +110)	(21, +1)	(120, +100)	131
Device B attempts to save state to cloud. Conflict detected.	(130, +110)	(21, +1)	(120, +100)	131
Device B solves conflict by applying local delta to cloud total.	(130, +110)	(121, +1)	(121, +1)	131
Device A tries to upload its data to the cloud. Conflict detected.	(130, +110)	(121, +1)	(121, +1)	131
Device A resolves the conflict by applying the local delta to the cloud total.	(131, +110)	(121, +1)	(231, +110)	131

(*): X ডাটা পরিবেশন করে যা আমাদের এই পেরেক্ষাপটে অপর্াসিঙ্গক।

এখন আপনার বিপরীত সমস্যা থাকবে: আপনি প্লেয়ারকে অনেক বেশী কয়েন দিচ্ছেন। এই প্লেয়ার ২১১ কয়েন অর্জন করেছে, যখন বস্তুত সে মাত্র ১১১ কয়েন সংগ্রহ করেছে।

সমাধান: ডিভাইস পরীক্ষিত সাব-টেস্টার করুন

পূর্ববর্তী পরীক্ষিত ট্যাবলেটগুলি পরীক্ষিত করে, মনে হয় যে ওই কীলেকান কয়েনটি ইতিমধ্যে ধারণ করা হয়েছে এবং কয়েনটি এখনও গণনা করা হয় নাই তা জানার যে সামর্থ্য তামিস করে, বিশেষ করে বিভিন্ন ডিভাইস থেকে আসা বহুমুখি ঘটমানতার উপস্থিতিতে।

সমস্যাটির সমাধান হচ্ছে একটি ডিকশনারী হওয়া আপনার ক্লাউড সেভার কাঠামো পরিবর্তন করা যা ইনটিজার সিস্টার মধ্যাপ করে। এই ডিকশনারীর মধ্যে পরীক্ষিত ট্যাবলেট-ভ্যালু পেয়ার একটি "drawer" পরিবেশন করে যা কয়েন ধারণ করে, এবং সেভার মধ্যে কয়েনের মোট সংখ্যা হচ্ছে সকল এন্ট্রির ভ্যালুর যোগফল। এই ডিভাইসের মূলনীতি হচ্ছে যে পরীক্ষিত ট্যাবলেটের সারানিভসব্ ডায়ার থাকা, এবং শুধুমাত্র ডিভাইস নিজে ডায়ারের মধ্যে কয়েন রাখতে পারবে।

ডিকশনারীর কাঠামো হচ্ছে $(A:a, B:b, C:c, \dots)$ যেখানে a হচ্ছে ডায়ার A এর মধ্যে রাখা কয়েনের মোট সংখ্যা, b হচ্ছে ডায়ার B এর মধ্যে রাখা কয়েনের মোট সংখ্যা এবং এরকম আরও অন্যান্য গুলো।

"drawer" সমাধানের জন্য নতুন কনফ্লিক্ট রেজলুয়েশন এলগরিদম নিচের মতো:

- Local data: (A:a, B:b, C:c, ...)
- Cloud data: (A:a', B:b', C:c', ...)
- Resolved data: (A:max(a,a'), B:max(b,b'), C:max(c,c'), ...)

উদাহরণস্বরূপ, যদি লোকাল ডাটা (A:20, B:4, C:7) হয় এবং ক্লাউড ডাটা (B:10, C:2, D:14) হয় তখন রিজলভড ডাটা (A:20, B:10, C:7, D:14) হবে। উল্লেখ্য যে আপনি কীভাবে এই ডিকশনারিতে কনফ্লিক্ট রেজলুয়েশন পর্যালোচনা করে বনসেটা আপনার অ্যাপের উপর নির্ভর করে। উদাহরণের জন্য বলা যায় কিছু অ্যাপের জন্য আপনি লোয়ার ভ্যালু যুক্ত করে নেবেন। এই নতুন এলগরিদম পরীক্ষা করে, উপরে বিনরুত পরীক্ষামূলক চিত্রগুলোর কোন একটিতে এটা পর্যালোচনা করুন। আপনি দেখতে পারবেন যে এটা একটি সঠিক ফলাফলে পৌঁছেছে। টেবিল ৩. এর উপর ভিত্তি করে টেবিল ৪. নিম্নলিখিত দর্শকটবে বখাখ্যাকরে:

- শুরুতে প্লয়ারের ২০ কয়েন ছিল। এটা পরীক্ষিত ট্যাবলেট এবং ক্লাউডে সঠিকভাবে পরীক্ষিত ফলিত হয়। এই ভ্যালু একটি ডিকশনারি (X:20) হিসাবে পরিবেশিত হয় যেখানে X এর ভ্যালু যোগুরুরতবং পূর্ণ নয়- এই পরীক্ষিত ডাটাকে কাঠামো থেকে আসলে সেবিষয়ে আমার পেরোয়াকরি না।
- যখন প্লয়ার ডিভাইস Aতে ১০০ কয়েন সংগ্রহ করে, এই পরিবর্তন একটি ডিকশনারি হিসাবে প্যাকেজ হয় এবং ক্লাউডে সেভ হয়। ডিকশনারির ভ্যালু হচ্ছে ১০০ কারণ ওটাকে কয়েনের সংখ্যা যা ইউজার ডিভাইস Aতে সংগ্রহ করেছে। এই পয়েন্ট ডাটাকে কোন গণনা সংঘটিত হয় নাই- ডিভাইস A সাদাভাবে এটাতে ইউজার কত কয়েন সংগ্রহিত কয়েনের সংখ্যা রিপোর্ট করে।
- কয়েনের পরীক্ষিত নতুন জমা ডিভাইসের সাথে সম্পৃক্ত ডিকশনারিতে প্যাকেজ হয় যা এটাকে ক্লাউডে সেভ করে। যখন প্লয়ার ডিভাইস Aতে আরও ১০ কয়েন সংগ্রহ করে, উদাহরণস্বরূপ, ডিভাইস A এর ভ্যালু ১১০ এ উন্নত হয়।
- নেট ফলাফল হচ্ছে যে পরীক্ষিত ট্যাবলেট ইউজার কত কয়েন সংগ্রহিত

কয়েকের সংখ্যা যা অ্যাপস জানে। এইভাবে এটা সহজ ভাবে সবর্মেটা গণনা করে।

টেবিল ৪. সফলিক-ভয়ালু য়ে পয়ার কৌশল পর্যেগ

Event	Data on Device A	Data on Device B	Data on Cloud	Actual Total
Starting conditions	(X:20, x)	(X:20, x)	(X:20, x)	20
Player collects 100 coins on device A	(X:20, A:100)	(X:20)	(X:20)	120
Device A saves state to cloud	(X:20, A:100)	(X:20)	(X:20, A:100)	120
Player collects 10 more coins on device A	(X:20, A:110)	(X:20)	(X:20, A:100)	130
Player collects 1 coin on device B	(X:20, A:110)	(X:20, B:1)	(X:20, A:100)	131
Device B attempts to save state to cloud. Conflict detected.	(X:20, A:110)	(X:20, B:1)	(X:20, A:100)	131
Device B solves conflict	(X:20, A:110)	(X:20, A:100, B:1)	(X:20, A:100, B:1)	131
Device A tries to upload its data to the cloud. Conflict detected.	(X:20, A:110)	(X:20, A:100, B:1)	(X:20, A:100, B:1)	131
Device A resolves the conflict	(X:20, A:110, B:1)	(X:20, A:100, B:1)	(X:20, A:110, B:1) total 131	131

আপনার ডাটা প্ৰিয়কৰণ কৰুন

ক্লাউড সেৱা ডাটাৰ সাইজ সীমাবদ্ধতা আছে, তাই এই আৰ্টিকলে যেকোনো অন্তৰ্ভুক্ত কৰা হৈছে, ইচ্ছামত বড় ডাটাৰ তৈৰী নকৰাৰ পৰিৱৰ্তিত থকাৰ লক্ষ্য হৈছে। প্ৰথম দৰ্শনে এটা মনে হৈছে প্ৰায় ডাটাৰিটৰি ডাটাৰিটৰি শুল্ক এক টি এনিং টৰ্ণ থকাৰে, এৰ এমনি অতি উ সৰ্হ ইউজাৰেৰ এগুলাৰ হাজাৰ টা থকাৰে না। অন্যৰিদে এক টি ডাটাৰিট আইডি অজৰ্ণ কৰা কঠিন এৰ এক টি থাৰাপ চৰ্চাৰি হাৰে বিবেচনা কৰা হয়, সুতৰাং পৰিবৰ্তৰ আপনাৰ উচিত এক টি ইনস্টলেশন আইডি বৰ্হাৰ কৰা উচিত যা অজৰ্ণ কৰা সহজ এৰ এটা থুব ইনিভৰ্ৰে যাগৰ্। এৰ মানে হৈছে যি পৰিৱৰ্তিত টা ডাটাৰিট ইউজাৰ কতৰ্ৰক অৰ্থাপলিকেশন ডাউনলোড কৰাৰ পৰিৱৰ্তিত টা সময় ডাটাৰিটৰ এক টি এনিং টৰ্ণ থকাৰে। ধাৰণ কৰা পৰিৱৰ্তিত টা ক-ভাৰ্গাল ১২ বাইট সেনয়, এৰ যিহেতু এক টি এক ক্লাউড সেৱা বাৰ্ফাৰ 128 K পৰ্ণৰ্ণত হৈছে পৰে, আপনি নিৰাপদ যদি আপনাৰ 4,096 পৰ্ণৰ্ণত এনিং টৰ্ণ থকাৰে।

বাস্তবিক জীৱনে আপনাৰ ডাটা কৰ্ণেৰ এক টি সৰ্থৰ্ৰাৰে চেয়ে সম্ভবত আৰু জিটল হৰে। এই কৰ্ণেৰে, এই ডাটাৰিটৰি এনিং টৰ্ণৰ সৰ্থৰ্ৰা আৰু সীমিত হৰে। আপনাৰ বাস্তবায়নেৰি ভিত্তিত কৰে, টাইমস্ টয়াম্ পৰে টাৰ কৰাৰ এক টি মানে থকাৰে যখন ডাটাৰিটৰি পৰিৱৰ্তিত টা এনিং টৰ্ণ পৰিবৰ্তৰ্ণ হৰে। যখন আপনি চিহ্নিত কৰে বনযে এক টি পৰ্ণদত্ত এনিং টৰ্ণ গত সপ্ৰাহগুলাত বা মাৰে পৰিবৰ্তৰ্ণ হয় নাই, কৰ্ণেৰি টি কৰ্ণেৰি টি এনিং টৰ্ণে বৰ্ণিল কৰা টা পৰ্ণাতন এনিং টৰ্ণে কৰ্ণিল টি কৰা টা সম্ভবত নিৰাপদ।

সিস্টেমিক অ্যান্ডাউটপটের ব্যবহার করে ডাটাস্টোরান্সফার

(<http://developer.android.com/training/sync-adapters/index.html>)

একটি অ্যান্ডরয়েড ডিভাইস এবং ওয়েব সার্ভারের মধ্যে ডাটাসিস্টেমিকরানাইজকরাট আপনার অ্যান্ডাউটপটলিকশনকে ইউজারেরজন্য আরওনিদিষ্টরূপে টাভে উপকারী এবং পরে ভাবিবস্তুারকারীকরে তুলবে।
উদাহরণ হিসাবে বলা যায় ওয়েব সার্ভারের ডাটাস্টোরান্সফার করা একটি কায়রুরকরীব্যবহাৎকআপটৈতরীকরে, এবং সার্ভারথেকে ডাটাস্টোরান্সফার করাটা এটাতে ইউজারেরকাছে সহজপরাপয় করে তৈরীকরে এমনকি যখন এটা অফলাইনে থাকে তখনও। কিছু কিছু ক্ষেত্রে ইউজার দেখবে ওয়েব ইন্টারফেসে পরে বশ করা এবং ডাটা এডিট করা খুব সহজ এবং তখন ওই ডাটাতাদের ডিভাইসে সহজপরাপয় হবে, অথবা তার সময়জুড়ে ডাটাসংগরহ করে তৈরীকরে এবং এগুলো একটি কেন্দ্রীয় স্টোরে এলাকায় আপলোড করে তৈরীকরে।

যদিও আপনি আপনার নিজস্ব অ্যান্ডাউটপটের ডাটাস্থানান্তরেরজন্য আপনার নিজস্ব ব্যবস্থার ডিভাইস করে তৈরীকরেন, কিন্তু আপনার উচিত অ্যান্ডরয়েডে ডাটাসিস্টেমিক অ্যান্ডাউটপটের ফের্মওয়াকর ব্যবহার করার বিষয়টাবিবেচনাকরা। এই ফের্মওয়াকর ডাটাস্থানান্তরকে পরিচালিত এবং সর্বাংকিয় করে তৈরীকরে এবং বিবিধ নতুন অ্যান্ডাউটপটের যেসিস্টেমিকরানাইজকায়রকলাপ হয়ে থাকে তা সমন্বয় করে। যখন আপনি এই ফের্মওয়াকর ব্যবহার করে বন আপনি কিছু বিশেষতবের সুবিধা নিতে পারবেন যা আপনার নিজের তৈরীকরা ডাটাস্টোরান্সফার সিস্টেমে পাওয়া যাবে না:

Plug-in architecture / প্লাগ-ইন আর্কিটেকচার

সিস্টেমে ক্যালাবল উপাদানরূপে ডাটাস্টোরান্সফারকে কাডযুক্ত করে দেয়।

Automated execution/ সর্বাংকিয় কায়রু সম্পাদন

কিছু মানদেন্ডের যেন ডাটার পরিবর্তন, অতিবাহিত সময় বা অতিবাহিত দিনের উপর ভিত্তি করে আপনাকে ডাটাস্টোরান্সফারকে সর্বাংকিয় করে দেয়। উপরন্তু সিস্টেমিক কিছু স্থানান্তরকে যুক্ত করে যা একটি ক্রিয়েরান করে তৈরীকরা এবং তাতে যখন সম্ভব তখন রান করেনা হয়।

Automated network checking / সর্বাংকিয় নেটওয়ার্ক পরিদর্শন

এই সিস্টেম শূন্য মাত্র তখনই আপনার ডাটাস্থানান্তর করে যখন ডিভাইসে সরে নেটওয়ার্ক কোনকিটিবিটি থাকে।

Improved battery performance/ ব্যাটারির কায়রু সম্পাদনের উন্নয়ন

আপনাকে আপনার অ্যান্ডাউটপটের সকল ডাটাস্থানান্তরের কাজগুলো একটি স্থানে কেন্দ্রীভূত করে দেয়, যাতে সেগুলো একসাথে রান করে তৈরীকরে। আপনার ডাটাস্টোরান্সফার অন্য অ্যান্ডাউটপটের স্থানান্তরের সাথে একতরুনিধরারিত। এই বিষয়গুলো নেটওয়ার্কের বারবার সুইচ করাটাকে কমিয়ে আন যা ব্যাটারির ব্যবহার কমিয়ে আনে।

Account management and authentication / একাউন্ট ব্যবস্থাপনা এবং পরামর্শন (অনলাইন লিকশন) একীভূত করা

যদি আপনার অ্যান্ডাউটপট ইউজারের পরিচয় বা সার্ভার প্লাগ-ইন চায়, আপনি ঐচ্ছিকভাবে আপনার ডাটাস্থানান্তরের মধ্যে একাউন্ট

ব্যবস্থাপনা এবং পর্যালোচনা (অথেনটিকেশন) একীভূত করে তৈরি করেন।

এই ক্লাস আপনাকে দেখাবে কীভাবে একটি সিস্টেম অ্যাডাপ্টর তৈরি করতে হয় এবং এটাকে র‍্যাপ (wraps) করা Service কে বাউন্ড করে, অন্য উপাদান যা ফর্মওয়ারকে র‍্যাপের মধ্যে সিস্টেম অ্যাডাপ্টর প্লাগ করতে সহায়তা করে তা কীভাবে সরবরাহ করা যায় এবং কীভাবে ভিন্ন উপায়ে সিস্টেম অ্যাডাপ্টরকে র‍্যাপ করা যায়।

নোট: সিস্টেম অ্যাডাপ্টর আসিঙ্কর‍্নাস ভাবে র‍্যাপ করে, সুতরাং আপনাকে সেই আকাঙ্ক্ষা নিয়ে তাদেরকে র‍্যাপ করার জন্যে উচিত যে তার নিয়মিতভাবে এবং দক্ষতার সাথে ডাটাস্ট্রান্সফার করতে পারে, কিন্ত
তা ক্রমিকভাবে নয়। যদি আপনার রিয়েল-টাইম ডাটাস্ট্রান্সফার করার পরয়োজন হয়, আপনি এটা একটি AsyncTask বা একটি IntentService তৈরি করে তৈরি করেন।

অনুশীলনীসমূহ

একটি স্টাব-অথেন্টিকেটর (পর্ম্যাণকারী) তৈরী করা

এখানে শিখুন কীভাবে একটি একাউন্ট-চালিত করা উপাদান যুক্ত করতে হয় যেটি সঙ্ক অ্যাডাপটর ফর্মওয়ার্ক আপনার অ্যাপের অংশ হওয়ার আকাঙ্ক্ষা করে। এই অনুশীলনী আপনাকে দেখাবে সহজ করার জন্য কীভাবে একটি একটি স্টাব-অথেন্টিকেটর (পর্ম্যাণকারী) উপাদান তৈরী করতে হয়।

একটি স্টাব-কনটেন্ট পর্ম্ভাইডার তৈরী করা

এখানে শিখুন কীভাবে একটি কনটেন্ট পর্ম্ভাইডার উপাদান যুক্ত করতে হয় যেটি সঙ্ক অ্যাডাপটর ফর্মওয়ার্ক আপনার অ্যাপের অংশ হওয়ার আকাঙ্ক্ষা করে। এই অনুশীলনী মনে করবে আপনার অ্যাপ একটি কনটেন্ট পর্ম্ভাইডার ব্যবহার করে না, তাই এটা আপনাকে দেখায় কীভাবে একটি স্টাব উপাদান যুক্ত করতে হয়। যদি আপনার অ্যাপের মধ্যে ইতিমধ্যে একটি কনটেন্ট পর্ম্ভাইডার থাকে, তাহলে এই অনুশীলনী বাদ দিয়ে পেরটোতে চলে যান।

একটি সঙ্ক অ্যাডাপটর তৈরী করুন

এখানে শিখুন একটি উপাদানের মধ্যে কীভাবে আপনার ডাটাবেজ সফার কেড পর্ম্বেশ করতে বন্যাতৈ সঙ্ক অ্যাডাপটর ফর্মওয়ার্ক সব যেকি যভাবে রান করতে পারে।

একটি সঙ্ক অ্যাডাপটর রান করা

এখানে শিখুন সঙ্ক অ্যাডাপটর ফর্মওয়ার্ক ব্যবহার করে কীভাবে ডাটাবেজ সফার সিকর্য এবং পরিকল্পনা করা যায়।

একটি স্টাব-অথেনটিকেটর (পৰ্মাণকারী) তৈরী করা

(<http://developer.android.com/training/sync-adapters/creating-authenticator.html>)

সিঙ্ক অ্যাডাপটর ফর্মওয়ার্ক মেনে করে যে আপনার সিঙ্ক অ্যাডাপটর একটি একাউন্ট এবং সাভার্সার স্টারেজ য়ালগ-ইন এক্সেস চায় তাদের সাথে সম্পৃক্ত ডিভাইসে স্টারেজের মধ্যে ডাটাতরান্সফার করে। এই জন্যে, ফর্মওয়ার্ক আপনার কাছে আপনার সিঙ্ক অ্যাডাপটরের অংশ হিসাবে অথেন্টিকেটর নামে একটি উপাদান আকাঙ্ক্ষা করে। এই উপাদান অ্যান্ডরয়েড একাউন্ট এবং অথেন্টিকেশন ফর্মওয়ার্কের মধ্যে প্লাগ করে এবং ইউজারের পরিচয় য়মন লগ-ইন তথ্য পরিচালনার জন্য একটি আদর্শ ইন্টারফেস প্রদান করে।

এমনকি যদি আপনার অ্যাডাপ একাউন্ট ব্যবহার না করে, আপনাকে তখনও একটি অথেন্টিকিটর উপাদান প্রদান করতে হবে। যদি আপনি একাউন্ট বা সাভার্সার লগ-ইন ব্যবহার করেন না থাকেন, অথেন্টিকেটর দ্বারা চালিত তথ্য হবে লিখিত হবে, তাই আপনি একটি অথেন্টিকেটর উপাদান সরবরাহ করতে পারেন যা স্টাব পদ্ধতি বাস্তবায়ন ধারণ করে। আপনার একটি বাউন্ড Service ও সরবরাহ করা উচিত যা সিঙ্ক অ্যাডাপটর ফর্মওয়ার্কের অথেন্টিকেটরের পদ্ধতি কল করতে দেয়।

এই অনুশীলনী আপনাকে দেখায় কীভাবে একটি স্টাব অথেন্টিকেটরের সকল অংশ নিধারণ করতে হয় যে আপনার সিঙ্ক অ্যাডাপটর ফর্মওয়ার্কের প্রয়োজনীয়তাকে সন্তুষ্ট করা প্রয়োজন। যদি আপনার একটি সত্যিকারের অথেন্টিকেটর প্রয়োজন হয় যা ইউজার একাউন্ট চালিত করে, AbstractAccountAuthenticator জন্য রেফারেন্স ডকুমেন্টশনটি পড়ুন।

একটি স্ট্যাব অথেন্টিকেটর উপাদান যুক্ত করুন

আপনার অ্যাপে একটি স্ট্যাব অথেন্টিকেটর উপাদান যুক্ত করতে, একটি ক্লাস তৈরি করুন যা `AbstractAccountAuthenticator` কে পর্সারিত করে, এবং তখন পর্যালোচনীয় পদ্ধতি লোপ করে, হয় `null` ফেরত দিয়ে বা একটি বয়্যিতকর্ম দেয়ার মাধ্যমে।

নিচের চিত্রটি একটি স্ট্যাব অথেন্টিকেটর ক্লাসের উদাহরণ দেখাচ্ছে:

```

/*
 * Implement AbstractAccountAuthenticator and stub out all
 * of its methods
 */
public class Authenticator extends AbstractAccountAuthenticator {
    // Simple constructor
    public Authenticator(Context context) {
        super(context);
    }
    // Editing properties is not supported
    @Override
    public Bundle editProperties(
        AccountAuthenticatorResponse r, String s) {
        throw new UnsupportedOperationException();
    }
    // Don't add additional accounts
    @Override
    public Bundle addAccount(
        AccountAuthenticatorResponse r,
        String s,
        String s2,
        String[] strings,
        Bundle bundle) throws NetworkErrorException {
        return null;
    }
    // Ignore attempts to confirm credentials
    @Override
    public Bundle confirmCredentials(
        AccountAuthenticatorResponse r,
        Account account,
        Bundle bundle) throws NetworkErrorException {
        return null;
    }
    // Getting an authentication token is not supported
    @Override
    public Bundle getAuthToken(
        AccountAuthenticatorResponse r,
        Account account,
        String s,
        Bundle bundle) throws NetworkErrorException {
        throw new UnsupportedOperationException();
    }
    // Getting a label for the auth token is not supported
    @Override
    public String getAuthTokenLabel(String s) {
        throw new UnsupportedOperationException();
    }
    // Updating user credentials is not supported
    @Override
    public Bundle updateCredentials(
        AccountAuthenticatorResponse r,
        Account account,
        String s, Bundle bundle) throws NetworkErrorException {
        throw new UnsupportedOperationException();
    }
    // Checking features for the account is not supported
    @Override
    public Bundle hasFeatures(
        AccountAuthenticatorResponse r,
        Account account, String[] strings) throws NetworkErrorException {
        throw new UnsupportedOperationException();
    }
}

```

ফের্ম ওয়াকের্ অথেন্টিকেটর সংযুক্ত করা

আপনার অথেন্টিকেটরের সিঙ্ক অফ্‌রাদাপটর ফের্ম ওয়াকের্ পর্বেশ
করানোর জন্য, আপনাকে অবশ্যই এটার জন্য একটি বাউন্ড সার্ভিস তৈরী
করতে হবে। এই সার্ভিস একটি অফ্‌রান্ডরয়েড বাইন্ডার অবজেক্ট
পর্দান করবে যা ফের্ম ওয়াকের্ ক আপনার অথেন্টিকেটরকে কল করতে দেয়
এবং অথেন্টিকেটর ও ফের্ম ওয়াকের্ এই দুইয়ের মধ্যে ডাটাপাস করে দেয়।

যেহেতু ফের্ম ওয়াকের্ পর্থমবার এই সার্ভিস শুরুর করে এটার
অথেন্টিকেটরের পর্বেশ করা পর্যেয়াজন হয়, সার্ভিসের `Service.onCreate()`
পদ্ধতিতে অথেন্টিকেটর কশনের কনস্ট্রাক্টরকে কল করা মাধ্যমে আপনি
অথেন্টিকেটর আরম্ভ করতে সার্ভিসটি ব্যবহার করতে পেরেন।

নিচের চিত্রটি দেখায় কীভাবে বাইন্ড `Service` নিধরারণ করতে হয়:

```
/**
 * A bound Service that instantiates the authenticator
 * when started.
 */
public class AuthenticatorService extends Service {
    ...
    // Instance field that stores the authenticator object
    private Authenticator mAuthenticator;
    @Override
    public void onCreate() {
        // Create a new authenticator object
        mAuthenticator = new Authenticator(this);
    }
    /**
     * When the system binds to this Service to make the RPC call
     * return the authenticator's IBinder.
     */
    @Override
    public IBinder onBind(Intent intent) {
        return mAuthenticator.getIBinder();
    }
}
```


অ্যেথনটি টেকেটর মেটাডেটা ফাইল যুক্ত করা

সিঙ্ক অ্যেডাপ্টর এবং একাউন্ট ফর্মওয়ার্কের মধ্যে আপনার অ্যেথনটি টেকেটর উপাদান যুক্ত করতে, আপনার পরে যাজন মেটাডেটা সহকারে এই ফর্মওয়ার্কের গুলে সরবরাহ করা যাউপাদান বর্ণনাকরে। এই মেটাডেটা একাউন্ট টাইপে যাষণাকরে যা আপনি আপনার সিঙ্ক একাউন্টের জন্য তৈরী করেছেন এবং ইউজার ইন্টারফেস ডিক্লয়ার করে যে যদি আপনি আপনার একাউন্ট টাইপ ইউজারের কাছে দৃশ্যমান করতে চান সিস্টেম এটা পর্দাশর্ন করে। এই মেটাডেটা আপনার অ্যেপ পর্জেক্টের `/res/xml/` ডিরেক্টরির মধ্যে স্টোর হওয়া একটি XML ফাইলে ডিক্লয়ার করুন। আপনি ফাইলটির যে কোন নাম দিতে পারেন, যদিও এটা সাধারণত `authenticator.xml` নামে পরিচিত।

এই XML ফাইল একটি একক এলিমেন্ট `<account-authenticator>` ধারণ করে যার নিচের গুণগুলো রয়েছে:

`android:accountType`

সিঙ্ক অ্যেডাপ্টর ফর্মওয়ার্কের প্রতিটা সিঙ্ক অ্যেডাপ্টরের একটি একাউন্ট টাইপ থাকুক, একটি ডেফোল্ট মইনেনম আকারে। ফর্মওয়ার্ক সিঙ্ক অ্যেডাপ্টরের অভ্যন্তরীণ পরিচিতির অংশ হিসাবে একাউন্ট টাইপ ব্যবহার করে। সাধারণতের জন্য লগ-ইন পরে যাজন, একটি ইউজার একাউন্টের সাথে একাউন্ট টাইপ লগ-ইন তথ্যের অংশ হিসাবে সাধারণতের পাঠানো হয়ে থাকে।

যদি আপনার সাধারণ লগ-ইন না চায়, আপনাকে তখনও একটি একাউন্ট টাইপ সরবরাহ করতে হবে। ভ্যালুর জন্য, একটি ডেফোল্ট মইনেনম ব্যবহার করুন যা আপনি নিয়ন্ত্রণ করেন। যখন ফর্মওয়ার্ক আপনার সিঙ্ক অ্যেডাপ্টর পরিচালনা করতে এটা ব্যবহার করে, ভ্যালু আপনার সাধারণতের পাঠানো হয় না।

`android:icon`

একটি আইকন ধারণ করা `Drawable` রিসোর্স এর প্রতি নির্দেশক। আপনি যদি `res/xml/syncadapter.xml` এর মধ্যে বিবেচিত `android:userVisible="true"` নির্দিষ্ট করার মাধ্যমে সিঙ্ক অ্যেডাপ্টর দৃশ্যমান করেন, তখন আপনাকে অবশ্যই এই আইকন রিসোর্স সরবরাহ করতে হবে। এটা সিস্টেমের স্টিং অ্যেপের একাউন্টস (Accounts) সেকশনে দেখা যায়।

`android:smallIcon`

একটি ছোট আকারের আইকন ধারণ করা `Drawable` রিসোর্স এর প্রতি নির্দেশক। স্ক্রিনের আকারের উপর ভিত্তি করে এই রিসোর্স সিস্টেমের স্টিং অ্যেপের একাউন্টস (Accounts) সেকশনের মধ্যে থাকা `android:icon` এর পরিবর্তে ব্যবহৃত হয়।

`android:label`

স্থানীয়করণে যাগযে স্ট্রিং যা ইউজারের একাউন্ট টাইপ পরীক্ষা করে। আপনি যদি `res/xml/syncadapter.xml` এর মধ্যে বিবেচিত `android:userVisible="true"` নির্দিষ্ট করার মাধ্যমে সিঙ্ক অ্যেডাপ্টর দৃশ্যমান করেন, তখন আপনাকে এই স্ট্রিং সরবরাহ করতে হবে। এটা সিস্টেমের স্টিং অ্যেপের একাউন্টস (Accounts) সেকশনে দেখা যায়, অ্যেথনটি টেকেটের জন্য যে আইকন নির্ধারণ করেছেন তার পাশে দেখা যায়।

নিচের চিত্রটি অ্যেথনটি টেকেটের জন্য XML ফাইল দেখায় যা আপনি পূর্বে

তৈরী করেছেন:

```
<?xml version="1.0" encoding="utf-8"?>
<account-authenticator
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:accountType="example.com"
  android:icon="@drawable/ic_launcher"
  android:smallIcon="@drawable/ic_launcher"
  android:label="@string/app_name"/>
```

মেনিফেস্টের মধ্যে অথেনটিকেটর ডিকে ক্লয়ার করা

পূর্ববর্তরী ধাপে, আপনি একটি বাউন্ড Service তৈরী করেছেন যা অথেনটিকেটরকে সিস্টেম অফ অ্যাপট্রফর্ম ওয়াকের সাথে যুক্ত করে। সিস্টেম এই সার্ভিস চিহ্নিত করতে, নিচের <service> এলিমেন্টটি <application> এর একটি চাইল্ড এলিমেন্ট হিসাবে যুক্ত করার মাধ্যমে আপনার অ্যাপ মেনিফেস্টের মধ্যে এটি ডিকে ক্লয়ার করতে হবে:

```
<service
  android:name="com.example.android.syncadapter.AuthenticatorService">
  <intent-filter>
    <action android:name="android.accounts.AccountAuthenticator"/>
  </intent-filter>
  <meta-data
    android:name="android.accounts.AccountAuthenticator"
    android:resource="@xml/authenticator" />
</service>
```

<intent-filter> এলিমেন্টটি একটি ফিল্টার সেট আপ করে যা ইন্টেন্ট একশন android.accounts.AccountAuthenticator দ্বারা সিকর্য হয়, যা অথেনটিকেটর রান করতে সিস্টেম দ্বারা পাঠানো হয়ে থাকে। যখন ফিল্টার সিকর্য হয়ে থাকে, সিস্টেমটি AuthenticatorService শুরুর করে, বাউন্ড Service সার্ভিস যা আপনি অথেনটিকেটর র্যাপ করার জন্য পর্দান করেছেন।

<meta-data> এলিমেন্টটি অথেনটিকেটরের জন্য মেটাডাটাই ডিকে ক্লয়ার করে। android:name এটির বিউট অথেনটিকেটর ফর্ম ওয়াকের মেটাডাটাই স্যোগ করে। android:resource এলিমেন্ট আপনি পূর্ববর্তী তৈরী করেছেন এমন অথেনটিকেটর মেটাডাটাই ফাইলের নাম নির্ধারণ করে।

একটি অথেনটিকেটর ছাড়াও, একটি সিস্টেম অফ অ্যাপট্রফর্ম কনটেন্ট পর্ভাইডার চায়। যদি আপনার অ্যাপ ইতমধ্যে কনটেন্ট পর্ভাইডার ব্যবহার না করে থাকে, তাহলে পর্বর্তী অনুষ্টীলনীতে চলে যান যেখানে একটি স্টাব কনটেন্ট পর্ভাইডার তৈরী করার শেখানো হয়েছে; অন্যথায় Creating a Sync Adapter অনুষ্টীলনীতে চলে যান।

একটি স্ট্রাব কন্টেইনার তৈরী করা

(<http://developer.android.com/training/sync-adapters/creating-stub-provider.html>)

সিঙ্ক অ্যাডাপটর ফর্মওয়ার্ক নমনীয় এবং অতিনিরাপদ কন্টেইনট প্ৰভাইডার ফর্মওয়ার্ক দ্বারা চালিত ডিভাইস ডাটাবেজের সাথে কাজ করার মতো করে ডিজাইন করা। এই কারণে, সিঙ্ক অ্যাডাপটর ফর্মওয়ার্ক আকাঙ্ক্ষা করে যে একটি অ্যাডাপ্টার ফর্মওয়ার্ক ব্যবহার করে সেটাই তৈরি হয়। এর লোকাল ডাটাবেজের জন্য একটি কন্টেইনট প্ৰভাইডার নিখরহা রাখা হয়েছে। যদি সিঙ্ক অ্যাডাপটর ফর্মওয়ার্ক আপনার সিঙ্ক অ্যাডাপটর রান করার চেষ্টা করে এবং আপনার অ্যাডাপ্টার কন্টেইনট প্ৰভাইডার না থাকে, আপনার সিঙ্ক অ্যাডাপটর কর্তৃক শাসিত হবে।

যদি আপনি একটি নতুন অ্যাডাপ্টার ডেভেলপ করতে থাকেন যা একটি সাধারণ থেকে ডিভাইস ডাটাবেজ স্থানান্তরিত করে, আপনার উচিত একটি কন্টেইনট প্ৰভাইডারের মধ্যে লোকাল ডাটাবেজ স্টোর করার বিষয়টি জিজ্ঞাসা করা। সিঙ্ক অ্যাডাপ্টারের জন্য তাদের গুরুত্ব হ্যাঁড়াও কন্টেইনট প্ৰভাইডার নান্দ্রনের নির্যাত্তার (সিকিউরিটির) সুবিধা দেয়ার পরে স্ট্রাব করে এবং বিশেষভাবে অ্যান্ড্রয়েড সিস্টেম ডাটাবেজ পেরিচালনাকার জন ডিজাইন করা হয়। একটি কন্টেইনট প্ৰভাইডার তৈরী করার বিষয়ে বিস্তারিত জানতে [Creating a Content Provider](#) দেখুন।

কিন্তু, যদি আপনি ইতিমধ্যে লোকাল ডাটাবেজ কানেক্ট করে থাকেন, আপনি এখনও ডাটাবেজের সফার পেরিচালিত করতে একটি সিঙ্ক অ্যাডাপ্টার ব্যবহার করতে পারেন। একটি কন্টেইনট প্ৰভাইডারের জন্য সিঙ্ক অ্যাডাপ্টার ফর্মওয়ার্ক পরে যোজনীয়তা তপ্ত করতে, আপনার অ্যাডাপ্টার একটি স্ট্রাব কন্টেইনট প্ৰভাইডার যুক্ত করুন। একটি স্ট্রাব প্ৰভাইডার কন্টেইনট প্ৰভাইডার ক্লাস বাস্তবায়ন করে, কিন্তু এর সকল পরে যোজনীয় পদ্ধতি null বা ০ ফেরত দেয়। আপনি যদি একটি স্ট্রাব প্ৰভাইডার যুক্ত করেন, আপনি যেটা পছন্দ করেন তার যে কোন স্ট্রাবেজ পদ্ধতি থেকে ডাটাবেজ স্থানান্তরিত করতে আপনি তখন একটি সিঙ্ক অ্যাডাপ্টার ব্যবহার করতে পারেন।

আপনার অ্যাডাপ্টার যদি ইতিমধ্যে একটি কন্টেইনট প্ৰভাইডার থেকে থাকে, আপনার স্ট্রাব কন্টেইনট প্ৰভাইডারের কোন পরে যোজন নেই। সেক্ষেত্রে আপনি এই অনুশীলনীতে ছেড় [Creating a Sync Adapter](#) অনুশীলনীতে চলে যান। যদি আপনার অ্যাডাপ্টার এখনও কন্টেইনট প্ৰভাইডার না থেকে থাকে, এই অনুশীলনী আপনাকে শেখাবে একটি স্ট্রাব কন্টেইনট প্ৰভাইডার যুক্ত করতে হয় যা আপনাকে ফর্মওয়ার্কের মধ্যে সিঙ্ক অ্যাডাপ্টার প্লাগ করতে দেয়।

একটি স্টাব কন্টেন্ট প্রভাইডার যুক্ত করা

আপনার অ্যাপের জন্য একটি স্টাব কন্টেন্ট প্রভাইডার তৈরী করতে, ক্লাস `ContentProvider` পর্সারিত করুন এবং এর পর্য়োজনীয় পদ্ধতিরৈাপ করে। নিচের চিত্রটি আপনাকে দেখায় কীভাবে স্টাব প্রভাইডার তৈরী করতে হয়:

```
/*
 * Define an implementation of ContentProvider that stubs out
 * all methods
 */
public class StubProvider extends ContentProvider {
    /*
     * Always return true, indicating that the
     * provider loaded correctly.
     */
    @Override
    public boolean onCreate() {
        return true;
    }
    /*
     * Return an empty String for MIME type
     */
    @Override
    public String getType() {
        return new String();
    }
    /*
     * query() always returns no results
     */
    @Override
    public Cursor query(
        Uri uri,
        String[] projection,
        String selection,
        String[] selectionArgs,
        String sortOrder) {
        return null;
    }
    /*
     * insert() always returns null (no URI)
     */
    @Override
    public Uri insert(Uri uri, ContentValues values) {
        return null;
    }
    /*
     * delete() always returns "no rows affected" (0)
     */
    @Override
    public int delete(Uri uri, String selection, String[] selectionArgs) {
        return 0;
    }
    /*
     * update() always returns "no rows affected" (0)
     */
    public int update(
        Uri uri,
        ContentValues values,
        String selection,
        String[] selectionArgs) {
        return 0;
    }
}
```

মেমিনেফেস্টর মধ্যে পর্ভাইডার ডিফেক্লয়ার করা

আপনার অ্যাপ তার মেমিনেফেস্টর মধ্যে একটি পর্ভাইডার ডিফেক্লয়ার করে ছে তা চেক করার মাধ্যমে সিস্ট্রক অ্যাপটার ফর্ম ওয়াকর্ যাচাই করে যে আপনার অ্যাপের একটি কনটেন্ট পর্ভাইডার আছে। মেমিনেফেস্টর মধ্যে স্টাব পর্ভাইডার ডিফেক্লয়ার করতে, একটি `<provider>` এলিমেন্ট যুক্ত করুন যার মধ্যে াকৃত গুণাগুন গেলার রয়েছে:

```
android:name="com.example.android.datasync.provider.StubProvider"
```

ক্লাসের পুনর-যোগ্যতা সম্পন্ন নাম নিদির্ষ্ট করে যা স্টাব কনটেন্ট পর্ভাইডার বাস্ তবায়ন করে।

```
android:authorities="com.example.android.datasync.provider"
```

একটি URI অথরিটি যা স্টাব কনটেন্ট পর্ভাইডার চিহ্নিত করে। এই ভ্যালু কে আপনার অ্যাপের পয্যাকেকজনাম করুন সিস্ট্রক এর সাথে স্যেযাজন করা "provider" দিয়ে। যদিও আপনি সিস্ট্রমে আপনার স্টাব পর্ভাইডার ডিফেক্লয়ার করে ছন, পর্ভাইডার নিজে ছাড়া আর কে কোন কিছুই পর্বেশ করতে চেষ্টা করেনা।

```
android:exported="false"
```

অন্য অ্যাপসও কনটেন্ট পর্ভাইডারের পর্বেশ করতে পারের কিনাতা নিধর্ারণ করে। আপনার স্টাব কনটেন্ট পর্ভাইডারের জন্য, false এ ভ্যালু সেট করুন, যেহেতু পর্ভাইডার দেখার জন্য অন্য অ্যাপকে অনুমোদন করার কোন পর্য়েযাজন নেই। এই ভ্যালু সিস্ট্রক অ্যাপটার এবকনটেন্ট পর্ভাইডারের মধ্যে ক্রমিখিস্কর্ য়কে কোন পর্ভাব ফেলেনা।

```
android:syncable="true"
```

একটি ফ্ল্যাগ সেট করে যানির্দর্শ করে যে পর্ভাইডার হচ্ছে সিস্ট্রক করার যোগ্য। আপনি যদি এই ফ্ল্যাগ true এ সেট করেন, আপনার কেডেড `setIsSyncable()` কল করার কোন পর্য়েযাজন নেই। ফ্ল্যাগটি সিস্ট্রক অ্যাপটার ফর্ম ওয়াকর্ কে কনটেন্ট পর্ভাইডার দিয়ে ডাটাস্থানান্তরিত করতে দেয়, কিন্তু ট্রান্সফার তখনই ঘটবে আপনি যদি সেজাসুজিভাবে তাকরে থাকেন।

নিচের চিত্রটি আপনাকে দেখায় কীভাবে অ্যাপ মেমিনেফেস্ট `<provider>` এলিমেন্ট যুক্ত করা য়:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.android.network.sync.BasicSyncAdapter"
    android:versionCode="1"
    android:versionName="1.0" >
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        ...
        <provider
            android:name="com.example.android.datasync.provider.StubProvider"
            android:authorities="com.example.android.datasync.provider"
            android:export="false"
            android:syncable="true"/>
        ...
    </application>
</manifest>
```

এখন আপনি যে সিস্টেম অফ্রাড আপটোর ফর্মওয়ার্ক দ্বারা কাঙ্ক্ষিত
নির্ভরতা (dependencies) তৈরী করেছেন, আপনি উপাদান তৈরী করতে পারেন যা
আপনার ডাটাবেজ সফটওয়্যারকে ডাটাবেজের বিস্তৃত করে। এই উপাদানকে সিস্টেম
অফ্রাড আপটোর বলাইয়া। পরবর্তী অনুশীলনীতে দেখাবে কীভাবে এই উপাদান
আপনার অফ্রাডে যুক্ত করা যায়।

একটি সিস্টেমিক অ্যাডাপ্টার তৈরী করা

(<http://developer.android.com/training/sync-adapters/creating-sync-adapter.html>)

আপনার অ্যাপের সিস্টেমিক অ্যাডাপ্টার উপাদান কাজের জন্য কোড পরিবেশিত করে রাখা যেখানে ডাটাইস এবং একটি সার্ভারের মধ্যে ডাটাত্রান্সফার করে। আপনার অ্যাপের পর্দান করা সুচি এবং ট্রিগার অনুযায়ী সিস্টেমিক অ্যাডাপ্টার ফের্মওয়াকর্ সিস্টেমিক অ্যাডাপ্টার উপাদানের মধ্যে কোড রাখা করে। আপনার অ্যাপের সিস্টেমিক অ্যাডাপ্টার উপাদান যুক্ত করতে আপনার নিম্নলিখিত অংশগুলো পর্যালোচনা করুন:

সিস্টেমিক অ্যাডাপ্টার ক্লাস

একটি ক্লাস যা সিস্টেমিক অ্যাডাপ্টার ফের্মওয়াকর্ সাথে যথেষ্ট যুক্ত একটি ইন্টারফেসের মধ্যে আপনার ডাটাত্রান্সফারকে কোড করা যাবে।

ব্যাউন্ড সার্ভিস

একটি উপাদান যা সিস্টেমিক অ্যাডাপ্টার ফের্মওয়াকর্কে আপনার সিস্টেমিক অ্যাডাপ্টার ক্লাসকে কোড রাখা করতে দেয়।

অ্যাডাপ্টার XML মেটাডেটাইফাইল সিস্টেমিক করুন

আপনার সিস্টেমিক অ্যাডাপ্টার সম্পর্কিত তথ্য ধারণ করা একটি ফাইল। কীভাবে আপনার ডাটাত্রান্সফার লোড হবে এবং পরিকল্পিত হবে খুঁজে বের করতে ফের্মওয়াকর্ এই ফাইল পড়ে।

অ্যাপ মেনিফেস্টের মধ্যে ডিক্লারেশন

XML বা ব্যাউন্ড সার্ভিস ডিক্লারেশন করে এবং সিস্টেমিক অ্যাডাপ্টার নির্দিষ্ট মেটাডেটাইনিয়েশন করে।

এই অনুশীলনী আপনাকে দেখাবে কীভাবে এই উপাদানগুলো নিধারণ করতে হয়।

সিস্টেমিক অ্যাডাপ্টার ক্লাস তৈরী করুন

অনুশীলনীর এই অংশে আপনি শিখবেন কীভাবে সিস্টেমিক অ্যাডাপ্টার ক্লাস তৈরী করতে হয় ডাটাত্রান্সফারকে কোড করে। ক্লাস তৈরী করুন সিস্টেমিক অ্যাডাপ্টার ইন্টারফেসের তথ্য ক্লাসের পর্দা করা, ক্লাসের জন্য কনস্ট্রাক্টর নিধারণ করা, এবং পদ্ধতি বাস্তবায়ন করে। অন্তর্ভুক্ত করে যেখানে আপনি ডাটাত্রান্সফার টাস্ক নিধারণ করেছেন।

বেজিসিক অ্যাডাপ্টার ক্লাস AbstractThreadedSyncAdapter পর্যালোচনা

সিঙ্ক অ্যাডাপ্টার উপাদান তৈরী করতে, AbstractThreadedSyncAdapter পর্যালোচনা করার এবং এর কনস্ট্রাক্টর লেখার মাধ্যমে শুরু করুন। সংক্রমণের ক্ষেত্রে প্রতিবার আপনার সিঙ্ক অ্যাডাপ্টার উপাদান তৈরী হওয়ার সময় টাস্ক (কমরকান্ড) সেটআপ করার সময় কনস্ট্রাক্টর ব্যবহার করুন, যেভাবে একটি একটি ভিডিও সেটআপ করতে আপনি Activity.onCreate() ব্যবহার করেছেন ঠিক সেভাবে। উদাহরণস্বরূপ, যদি আপনার অ্যাডাপ্টার সেট করার ক্ষেত্রে একটি কন্টেন্ট প্রভাইডার ব্যবহার করে থাকে, একটি ContentResolver ইনস্ট্যান্স পাওয়ার জন্য কনস্ট্রাক্টর ব্যবহার করুন। যেহেতু দুই ধরনের সমস্যা হতে পারে আগের মতো স্যুপার ক্লাসের জন্য অ্যান্ড্রয়েড প্ল্যাটফর্ম সংস্করণ ৩.০ এর মধ্যে কনস্ট্রাক্টরের বিদ্যমান (ফর্ম) যুক্ত করা হয়েছে, সবরকম ক্ষেত্রে উপযোগী তাজার রাখতে আপনাকে কনস্ট্রাক্টরের দুইটা ধরণ (ফর্ম) তৈরী করতে হবে।

নোট: সিঙ্ক অ্যাডাপ্টার ফর্ম ওয়াকার সিঙ্ক অ্যাডাপ্টার উপাদানের সাথে কাজ করার জন্য ডিজাইন করা হয়েছে যা হচ্ছে সিমুলেটেড ইনস্ট্যান্স। ইনস্ট্যান্স সেট করা সিঙ্ক অ্যাডাপ্টার উপাদান Bind the Sync Adapter to the Framework অধ্যায়ে আরও বিস্তারিতভাবে আলোচনায় এসেছে।

নীচের নকশা উদাহরণ আপনাকে দেখাবে কীভাবে AbstractThreadedSyncAdapter এবং এর কনস্ট্রাক্টর বাস্তবায়ন করতে হয়:

```
/**
 * Handle the transfer of data between a server and an
 * app, using the Android sync adapter framework.
 */
public class SyncAdapter extends AbstractThreadedSyncAdapter {
    ...
    // Global variables
    // Define a variable to contain a content resolver instance
    ContentResolver mContentResolver;
    /**
     * Set up the sync adapter
     */
    public SyncAdapter(Context context, boolean autoInitialize) {
        super(context, autoInitialize);
    }
    /**
     * If your app uses a content resolver, get an instance of it
     * from the incoming Context
     */
    mContentResolver = context.getContentResolver();
}
...
/**
 * Set up the sync adapter. This form of the
 * constructor maintains compatibility with Android 3.0
 * and later platform versions
 */
public SyncAdapter(
    Context context,
    boolean autoInitialize,
    boolean allowParallelSyncs) {
    super(context, autoInitialize, allowParallelSyncs);
}
/**
 * If your app uses a content resolver, get an instance of it
 * from the incoming Context
 */
mContentResolver = context.getContentResolver();
...
}
```

onPerformSync() এ ডাটাত্রান্ সফার কেড যুক্ত করুন

সিঙ্ক অ্যাডাপ্টর উপাদান সর্বত্রিকর্যভাবে ডাটাত্রান্ সফার করে
না। পরিবর্তে এটা আপনার ডাটাত্রান্ সফার কেড আবিরিত করে, যাতে
আপনার অ্যাডাপ্টর থেকে কোন ধরনের সম্পৃক্ততা ছাড়াই সিঙ্ক
অ্যাডাপ্টর ফের্মওয়াকর বয়াকগর আইনে ডাটাত্রান্ সফার রান
করতে পারে। যখন ফের্মওয়াকর আপনার অ্যাডাপ্টরকে কনের ডাটাসিঙ্ক
করতে পরস্তুত হয়, এটা পদ্ধতি onPerformSync() এর বাস্তুবায়ন আহবান করে।

সিঙ্ক অ্যাডাপ্টর উপাদানে আপনার মূল অ্যাডাপ্টর কেড থেকে ডাটার
ত্রান্ সফার সঞ্চালন করতে, সিঙ্ক অ্যাডাপ্টর ফের্মওয়াকর নিচের
আগুর্তে মেন্টিগুনেলা সহকারে onPerformSync() কল করে:

Account

ইভেনেটর সাথে সম্পৃক্ত একটি Account অবজেক্ট যা সিঙ্ক
অ্যাডাপ্টরকে সিকর্য করে। যি আপনার সাভরার অ্যাকাউন্ট
বয়বহার না করে, আপনার এই অবজেক্টর মধ্যে তথ্য বয়বহার করার
কোন পর্তেয়োজন নেই।

Extras

ইভেন্ট কতৃত্ব পাঠানে একটি Bundle ধারন করার্ফল্যাগস।

Authority

সিসেট্টেমের মধ্যে একটি কনটেন্ট পর্তাইডারের অথরিটি। আপনার
অ্যাডাপ্টর এই পর্তাইডারের পর্তেবশগম্যতা থাকতে হবে। সাধারণত, অথরিটি
আপনার নিজেস্বর অ্যাডাপ্টর মধ্যে একটি কনটেন্টেট্টেযোগাযোগ করে।

Content provider client

কনটেন্ট পর্তাইডারের জন্য একটি ContentProviderClient অথরিটি আগুর্তে মেন্টি
দবারানিদেদর্শ করে। একটি ContentProviderClient হচ্ছে একটি কনটেন্ট
পর্তাইডারের লাইটওয়েট পাবলিক ইন্টারফেস। একটি ContentResolver হিসাবে
এটার একই রকম মৌলিক কাযর্পরণালী রয়েছে। আপনি যদি আপনার
অ্যাডাপ্টর জন্য ডাটাসেট্টার করতে একটি কনটেন্ট পর্তাইডার
বয়বহার করতে থাকেন, আপনি এই অবজেক্ট দিয়ে কনটেন্ট পর্তাইডারের
সাথে সংযোগ স্থাপন করতে পারেন, অন্যথায় আপনি এটাকে অবহেলা করতে
পারেন।

Sync result

একটি SyncResult যা আপনি সিঙ্ক অ্যাডাপ্টর ফের্মওয়াকর তথ্য পঠাতে
বয়বহার করেন।

নিচের অংশিটি onPerformSync() এর সাবরক্ত কাঠামো তুলে ধরেছে:

```

/*
 * Specify the code you want to run in the sync adapter. The entire
 * sync adapter runs in a background thread, so you don't have to set
 * up your own background processing.
 */
@Override
public void onPerformSync(
    Account account,
    Bundle extras,
    String authority,
    ContentProviderClient provider,
    SyncResult syncResult) {
/*
 * Put the data transfer code here.
 */
...
}

```

যখন `onPerformSync()` এর আসল বাস্‌তবায়ন আপনার অয্যাপের ডাটা হিসেবে থরানাইজেশন আকাঙ্ক্ষায় এবং সাভরার কাকেশন পরেটাকলে নিদির্ষ্ট হয়, সেখানে নিকছু সাধারণ কাজ থাকে যা আপনার বাস্‌তবায়নকে অবশ্যই সম্‌পাদন করতে হবে।

সাভরার য়েক্ত করা

যদিও আপনি মনে করতে পারেন যে যখন আপনার ডাটা টরান্‌সফার শুরুর হয় তখন নেটওয়ার্ক থকে, সিঙ্ক অয্যাপটির ফের্মওয়ার্ক সব্‌য়্যিকর্য্যভাবে একটি সাভরার য়েক্ত হয় না।

ডাটা ডাউনেলাড এবং আপেলাড করা

একটি সিঙ্ক অয্যাপটির কোন ডাটা টরান্‌সফারের কাজ সব্‌য়্যিকর্য্য পদ্ধতি পরেয়াগ করেনা। আপনি যদি সাভরার থেকে ডাউনেলাড করতে চান এবং সেটাকনেটিন্‌ট পর্ভাইডারে সেটার করতে চান, আপনাকে কেড পর্‌দান করতে হবে যা ডাটাকে রিকয়েস্ট করে, এটাকে ডাউনেলাড করে, এবং পর্ভাইডারের মধ্যে এটাকে পর্বেশ করায়। একইভাবে, যদি আপনি সাভরার ডাটাকে সেন্ড করতে চান, আপনাকে এটা একটি ফাইল, ডাটাবেজ বা পর্ভাইডার থেকে পড়তে হবে এবং পরেয়াজনীয় আপেলাড রিকয়েস্ট পাঠাতে হবে। আপনাকে নেটওয়ার্ক এররকে ওচালিত করতে হবে যা আপনার ডাটা টরান্‌সফার হওয়ার সময় ঘটতে পারে।

ডাটাকনফ্লক্ট(সংঘাত/সংঘর্ষ) পরিচালনা বা ডাটাকতটা সাম্প্রতিকতানিধরার

একটি সিঙ্ক অয্যাপটির ডিভাইসের ডাটা এবং সাভরার ডাটার মধ্যে ক্রসিংয়ের ক্রিষয় সব্‌য়্যিকর্য্যভাবে পরিচালনা করেনা। যদি সাভরারের মধ্যে ক্রসিংয়ের ডাটা ডিভাইসের মধ্যে ক্রসিংয়ের ডাটার চেয়ে অধিকতর সাম্প্রতিক হয় বা এর বিপরীতভাবে ঘটলে এটা সেবিষয়টাও সব্‌য়্যিকর্য্যভাবে পরিচালনা করেনা। পরিবর্তে, এই অবস্থা চালিত করতে আপনাকে আপনার নিজস্ব এলগরিদম সরবরাহ করতে হবে।

পরিচ্ছন্ন করা

আপনার ডাটা টরান্‌সফার শেষে সবসময় একটি সাভরার সাথে সংযোগ বন্ধ করতে হবে এবং সেটস্‌ফাইল এবং কয্যশ ফাইল পরিচ্ছন্ন করতে হবে।

েনাট: সিঙ্ক অয্যাপটির ফের্মওয়ার্ক একটি বয্যাকগর্‌াউন্ড থের্‌ডে `onPerformSync()` রান করায়, তাই আপনাকে আপনার নিজস্ব বয্যাকগর্‌াউন্ড পর্‌সিস্যেকে সেটআপ করতে হবে না।

concentrating আপনার সিঙ্ক-সম্পর্কিত কাজ ছাড়াও, আপনার নিয়মিত
নেটওয়ার্ক-সম্পর্কিত কাজ একত্রে করা চেষ্টা করা উচিত এবং
তাদেরকে `onPerformSync()` এ যুক্ত করা উচিত। এই পদ্ধতিতে আপনার সকল
নেটওয়ার্ক কাজ কেন্দ্রীভূত করার মাধ্যমে আপনি ব্যাটারির পাওয়ার
সংরক্ষণ করতে পারেন যা নেটওয়ার্ক ইন্টারফেস শুরুর করতে এবং বন্ধ
করতে পরে যাবেন। নেটওয়ার্ক এক্সেসকে আরও কমরদক্ষ করার বিষয়ে
বিস্তারিত জানতে [Transferring Data Without Draining the Battery](#) পরিশীলন ক্লাসটি
দেখুন, যা বেশ কিছু নেটওয়ার্ক এক্সেস বিবরণ আলোচনা করে
যেটা আপনি আপনার ডাটাবেজের ক্ষেত্রে অন্বেষণ করতে পারেন।

ফের্ম ওয়াকে র্ সিসঙ্ক অয্ াড াপ টর একিতর্ত করা

এখন আপনার সিসঙ্ক অয্ াড াপ টর উপাদানের মধ্যে আবদ্ধ ডাটা টর ান সফার কে াড আছে, কিন্ত আপনাকে আপনার কে াড পর্বেশ গম্যতা সহ ফের্ম ওয়াকর সর্বরাহ করতে হবে। এটা করতে, আপনার একটি বাউন্ড Service তৈরী করা পর্বেযাজন যা সিসঙ্ক অয্ াড াপ টর থেকে ফের্ম ওয়াকর একটি বিবেশ অয্ ান ডর্বেড বাইন্ডার অবজেক্ট পাস করে। এই বাইন্ডার অবজেক্ট দিয়ে ফের্ম ওয়াকর `onPerformSync()` পদ্ধতি আহবান করতে পারে এবং ডাটা এটাতে পাস করে।

সাভিতর সের `onCreate()` পদ্ধতির মধ্যে একটি সিসঙ্ক গলেটান হিসাবে আপনার সিসঙ্ক অয্ াড াপ টর উপাদান ইনস্টেন্সিয়েট করুন। `onCreate()` এর মধ্যে উপাদান ইনস্টেন্সিয়েট করার মাধ্যমে, যতক্ষণ সাভিতর স শুরু না হয় আপনি এটাতৈরী করা মূলতবী করতে পারেন, যা ঘটবে যখন ফের্ম ওয়াকর আপনার ডাটা টর ান সফার পর্ধে মরান করার চেষ্টা করে। সিসঙ্ক অয্ াড াপ টর ফের্ম ওয়াকর সিকর্য করা বা পিরকল্পনা করার জবাবে আপনার সিসঙ্ক অয্ াড াপ টরের বহুমুখি কমর সম্পাদন সারিবদ্ধ করার কেষ্টের আপনার একটি থের্ড-পার্টি সফম্য ানারের মধ্যে উপাদান ইনস্টেন্সিয়েট করা পর্বেযাজন।

উদাহরণস্বরূপ, নীচে চিত্রিত আপনাকে দেখায় কীভাবে একটি ক্লাস তৈরী করতে হয় যা বাউন্ড সাভিতর স বাসতবায়ন করে, আপনার সিসঙ্ক অয্ াড াপ টর উপাদান ইনস্টেন্সিয়েট করে, এবং অয্ ান ডর্বেড বাইন্ডার অবজেক্ট লাভ করে:

```

package com.example.android.syncadapter;

/**
 * Define a Service that returns an IBinder for the
 * sync adapter class, allowing the sync adapter framework to call
 * onPerformSync().
 */
public class SyncService extends Service {
    // Storage for an instance of the sync adapter
    private static SyncAdapter sSyncAdapter = null;
    // Object to use as a thread-safe lock
    private static final Object sSyncAdapterLock = new Object();
    /**
     * Instantiate the sync adapter object.
     */
    @Override
    public void onCreate() {
        /**
         * Create the sync adapter as a singleton.
         * Set the sync adapter as syncable
         * Disallow parallel syncs
         */
        synchronized (sSyncAdapterLock) {
            if (sSyncAdapter == null) {
                sSyncAdapter = new SyncAdapter(getApplicationContext(), true);
            }
        }
    }
    /**
     * Return an object that allows the system to invoke
     * the sync adapter.
     */
    @Override
    public IBinder onBind(Intent intent) {
        /**
         * Get the object that allows external processes
         * to call onPerformSync(). The object is created
         * in the base class code when the SyncAdapter
         * constructors call super()
         */
        return sSyncAdapter.getSyncAdapterBinder();
    }
}

```

নোট: একটি টিসিঙ্ক অর্ডাপটের জনস্ব একটি টবাইন্ডের বিস্তারিত উদাহরণের জনস্ব উদাহরণ অর্ডাপ দেখুন।

ফের্ম ওয়াকর কতর ক চাওয়া একাউন্ট যুক্ত করুন

সিঙ্ক অ্যাডাপ্টার ফের্ম ওয়াকর চায় প্রতিটা সিঙ্ক অ্যাডাপ্টারের একটি একাউন্ট টাইপ থাক। আপনি Add the Authenticator Metadata File অধ্যায়ে আপনি একাউন্ট টাইপ ভ্যালু ডিক্লেয়ার করেছিলেন। এখন আপনাকে অ্যান্ড্রয়েড সিস্টেমের মধ্যে এই একাউন্ট টাইপ সেট আপ করতে হবে। একাউন্ট টাইপ সেট আপ করতে, একটি ডামি (পরিতকৃত) একাউন্ট যুক্ত করুন যা `addAccountExplicitly()` কল করার মাধ্যমে একাউন্ট টাইপ ব্যবহার করে।

পদ্ধতি কল করার সবচেয়ে ভালো জায়গা হচ্ছে আপনার অ্যাপের প্রারম্ভিক কায়রকর্মের `onCreate()` পদ্ধতির মধ্যে। নিচের কোড চিত্রটি এটাকীভাবে করতে হয় তা দেখায়:

```
public class MainActivity extends FragmentActivity {
    ...
    ...
    // Constants
    // The authority for the sync adapter's content provider
    public static final String AUTHORITY = "com.example.android.datasync.provider"
    // An account type, in the form of a domain name
    public static final String ACCOUNT_TYPE = "example.com";
    // The account name
    public static final String ACCOUNT = "dummyaccount";
    // Instance fields
    Account mAccount;
    ...
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        ...
        // Create the dummy account
        mAccount = CreateSyncAccount(this);
        ...
    }
    ...
    /**
     * Create a new dummy account for the sync adapter
     *
     * @param context The application context
     */
    public static Account CreateSyncAccount(Context context) {
        // Create the account type and default account
        Account newAccount = new Account(
            ACCOUNT, ACCOUNT_TYPE);
        // Get an instance of the Android account manager
        AccountManager accountManager =
            (AccountManager) context.getSystemService(
                ACCOUNT_SERVICE);

        /*
         * Add the account and account type, no password or user data
         * If successful, return the Account object, otherwise report an error.
         */
        if (accountManager.addAccountExplicitly(newAccount, null, null)) {
            /*
             * If you don't set android:syncable="true" in
             * in your <provider> element in the manifest,
             * then call context.setIsSyncable(account, AUTHORITY, 1)
             * here.
             */
        } else {
            /*
             * The account exists or some other error occurred. Log this, report it,
             * or handle it internally.
             */
        }
    }
    ...
}
```

সিঙ্ক অ্যাডাপটর মেটাডেটা ফাইল যুক্ত করুন

ফের্মওয়াকের মধ্যে আপনার সিঙ্ক অ্যাডাপটর উপাদান প্লাগ করতে আপনার মেটাডেটা সহকারে ফের্মওয়াকের সরবরাহ করা যাউপাদান আলাচনা করে এবং অতিরিক্ত ফ্ল্যাগস পরদান করে। আপনি আপনার সিঙ্ক অ্যাডাপটরের জন্য যে একাউন্ট টাইপ তৈরী করেছেন মেটাডেটা তামেটাডাটানিদিদর্শক করে, আপনার অ্যাডাপের সাথে সম্পৃক্ত কনটেন্ট পরভাইডার অথরিটি ডিকে লয়ার করে, সিঙ্ক অ্যাডাপটরের সাথে জিরিত সিস্টেম ইউজার ইন্টারফেসের একটি অংশে নিয়ন্ত্রণ করে এবং অন্য সিঙ্ক সম্প্রদায় ফ্ল্যাগ ডিকে লয়ার করে। আপনার অ্যাডাপ পরে জেক্টও মধ্যে কার/res/xml/ ডিরেক্টরির মধ্যে স্টোর করা একটি বিবেশ XML ফাইলের মধ্যে এই মেটাডেটা ডিকে লয়ার করুন। আপনি এই ফাইলের যেকোন নাম দিতে পারেন, যদি এটাকে syncaapter নামে ডাকা হয়।

এই XML ফাইল একটি একক XML এলিমেন্ট <sync-adapter> ধারণ করে যার নিম্নোক্ত বিবেশতব্গুত্ব লে রয়েছে:

android:contentAuthority

আপনার কনটেন্ট পরভাইডারের জন্য URI অথরিটি। পূর্ববর্তরী অনুশীলনীতে (Creating a Stub Content Provider) আপনি যদি আপনার অ্যাডাপের জন্য একটি স্টাব কনটেন্ট পরভাইডার তৈরী করে থাকেন, আপনি অ্যাডাপের মিনিফেস্টে <provider> এলিমেন্ট যুক্ত করেছেন তার মধ্যে এটির বিউট android:authorities জেনে যে ভয়ালু নিদিদর্শক করেছেন তা ব্যবহার করুন। এই এটির বিউট Declare the Provider in the Manifest অধ্যায়ে আরও বিস্তারিতভাবে আলাচনা করা হয়েছে। আপনি যদি সিঙ্ক অ্যাডাপটর দিয়ে একটি কনটেন্ট পরভাইডার থেকে সাভারের ডাটাবেস সফার করতে থাকেন, এই ভয়ালু একইরকম হওয়া উচিত যেমন আপনি ডাটাবেসের জন্য কনটেন্ট টজও ব্যবহার করেছেন সেরকম। এই ভয়ালু একটি অথরিটি ওয়া <provider> এলিমেন্টের android:authorities এটির বিউটের মধ্যে যা আপনি নিদিদর্শক করেছেন যা আপনার অ্যাডাপের মিনিফেস্টের মধ্যে আপনার পরভাইডারকে ডিকে লয়ার করে।

android:accountType

সিঙ্ক অ্যাডাপটর ফের্মওয়াকের দবারা চাওয়া একাউন্ট টাইপ। যখন আপনি অথেনটিটিকেটর মেটাডেটা ফাইল তৈরী করেছিলেন তখন যে একাউন্ট টাইপ ভয়ালু পরদান করেছিলেন ভয়ালু টিকে অবশ্যই সেইরকম হতে হবে, যেভাবে Add the Authenticator Metadata File অধ্যায়ে আলাচনা করা হয়েছে। এটা Add the Account Required by the Framework অধ্যায়ে মধ্যে ক্যাড চিত্রের (ক্যাড সিনপেট) মধ্যে কনসেন্ট ACCOUNT_TYPE এর জন্য নিদিদর্শক ভয়ালু।

Settings attributes

android:userVisible সিঙ্ক অ্যাডাপটরের একাউন্ট টাইপের দৃশ্যমানতা সেট করুন। বাইডিফল্ট একাউন্ট টাইপের সাথে সম্পৃক্ত একাউন্ট আইকন এবং লেভেল সিস্টেমের সিস্টেম অ্যাডাপের Accounts অধ্যায়ে মধ্যে দৃশ্যমান, আপনার উচিত আপনার সিঙ্ক অ্যাডাপটর অদৃশ্য করা যতকষণ না আপনার একটি একাউন্ট টাইপ বাডোমইন থাকে যা আপনার অ্যাডাপের সাথে সহজেই সম্পৃক্ত হয়। আপনি যদি আপনার একাউন্ট টাইপ অদৃশ্য করেন, আপনি তখনও ইউজারকে আপনার অ্যাডাপের একটি একটি ভিউটির মধ্যে একটি ইউজার ইন্টারফেস দিয়ে আপনার সিঙ্ক অ্যাডাপটর নিয়ন্ত্রণ করতে দিতে পারবেন।

android:supportsUploading

আপনাকে কল্লাউডে ডাটা আপলোড করতে দেয়। আপনার অ্যাপ যদি শুধু ডাটা ডাউনলোড করে এটাকে false এসেট করুন।

android:allowParallelSyncs

একই সময়ে রান করতে আপনার সিঙ্ক অ্যাপটির উপাদানের মাল্টিপল ইনস্টেন্স অনুমোদন। এটা ব্যবহার করুন যদি আপনার অ্যাপ মাল্টিপল (বহুবিধ) ইউজার একাউন্ট সাপোর্ট করে এবং সমান্তরালভাবে আপনি মাল্টিপল ইউজারকে ডাটা ট্রান্সফার করতে চান। ফলস্বরূপের কোন পরে ভাবেনই যদি আপনি কখনই মাল্টিপল ডাটা ট্রান্সফার না করেন।

android:isAlwaysSyncable

সিঙ্ক অ্যাপটির ফোর্স ওয়াকে কর্নিডের শকের যে এটা আপনার সিঙ্ক অ্যাপটিকে রান করে আপনি যখন ইনিদির স্ট করেন। আপনি যদি পেরাগর্মে টিকয্যালিনিয়ন তরণ করতে চান, যখন আপনার সিঙ্ক অ্যাপটির রান করতে পারবে, এই ফলস্বরূপ সতর্ক অব এসেট করুন, এবং তারপর সিঙ্ক অ্যাপটির রান করতে requestSync() কল করুন। একটি সিঙ্ক অ্যাপটির রান করার বিষয়ে বিস্তারিত জানতে Running a Sync Adapter দেখুন।

নিচের উদাহরণে দেখায়, একটি সিঙ্ক অ্যাপটির জন্য XML যা একটি একক পরিত্রুপ (ডামি) একাউন্ট ব্যবহার করে এবং শুধুমাত্র ডাউনলোড করে।

```
<?xml version="1.0" encoding="utf-8"?>
<sync-adapter
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:contentAuthority="com.example.android.datasync.provider"
  android:accountType="com.android.example.datasync"
  android:userVisible="false"
  android:supportsUploading="false"
  android:allowParallelSyncs="false"
  android:isAlwaysSyncable="true"/>
```

মেনিফেস্টের মধ্যে সিস্টেম অ্যান্ড্রয়েড পটের ডিকে ক্লয়ার করা

যখনই আপনি আপনার অ্যান্ড্রয়েড পিসিস্টেম অ্যান্ড্রয়েড পটের উপাদান যুক্ত করবেন, আপনার এক উপাদান ব্যবহার সম্পর্কিত পারমিশন রিকোয়েস্ট করতে হবে, আপনি যে বাউন্ড Service যুক্ত করেছেন তাকে ডিকে ক্লয়ার করতে হবে।

যেহেতু সিস্টেম অ্যান্ড্রয়েড পটের উপাদান কেডরান করে যানেনটওয়ার্ক এবং ডিভাইসের মধ্যে ডাটাতরান্সফার করে, আপনার ইন্টারনেট পর্বেশ করার জন্য পারমিশন রিকোয়েস্ট করা পরয়োজন। উপরন্তু সিস্টেম অ্যান্ড্রয়েড পটের সিস্টেম পডতে এবং লেখার জন্য আপনার অ্যান্ড্রয়েড পটের পারমিশন রিকোয়েস্ট করা পরয়োজন, সুতরাং আপনি আপনার অ্যান্ড্রয়েড পটের মধ্যে অন্যান্য উপাদান থেকে সিস্টেম অ্যান্ড্রয়েড পটকে পেরাগর্গে মটিক্যালীনিয়ন্তরণ করতে পারবেন। আপনার একটি বিশেষ পারমিশন রিকোয়েস্ট করতে হবে যা আপনি Creating a Stub Authenticator অনুশীলনীতে যে অথেনটিকেটর উপাদান তৈরী করেছিলেন তা আপনার অ্যান্ড্রয়েড পটকে ব্যবহার করতে দেয়।

এই পারমিশনগুলোকে রিকোয়েস্ট করার জন্য <manifest> এর চাইল্ড এলিমেন্ট হিসাবে নিচের বিষয়গুলো আপনার অ্যান্ড্রয়েড মেনিফেস্ট যুক্ত করুন:

android.permission.INTERNET

ইন্টারনেট পর্বেশ করার জন্য সিস্টেম অ্যান্ড্রয়েড পটকে ডানুমেদান করা যাতে এটা ডিভাইস থেকে সাভর্গের ডাটাদাউনলেড এবং আপলেড করতে পারে। আপনি যদি পূর্বে এই পারমিশন রিকোয়েস্ট করে থাকেন আপনার কনতুন করে এই এটাকে যুক্ত করতে হবে না।

android.permission.READ_SYNC_SETTINGS

চলতি সিস্টেম অ্যান্ড্রয়েড পটের সিস্টেম পডার জন্য আপনার অ্যান্ড্রয়েড পটকে ডানুমেদান করে। উদাহরণস্বরূপ, getIsSyncable() কল করার জন্য আপনার এই পারমিশন পরয়োজন।

android.permission.WRITE_SYNC_SETTINGS

সিস্টেম অ্যান্ড্রয়েড পটের সিস্টেম নিয়ন্ত্রণ করতে আপনার অ্যান্ড্রয়েড পটকে ডানুমেদান করে। addPeriodicSync() ব্যবহার করে মেয়াদকালীন সিস্টেম অ্যান্ড্রয়েড পটের রানসেটআপ করার জন্য আপনার এই পারমিশন পরয়োজন। requestSync() কল করা এই পারমিশনের জন্য পরয়োজনীয় নয়। সিস্টেম অ্যান্ড্রয়েড পটের রান করা বিষয়ক আরও তথ্যের জন্য Running A Sync Adapter দেখুন।

android.permission.AUTHENTICATE_ACCOUNTS

আপনি Creating a Stub Authenticator অনুশীলনীতে যে অথেনটিকেটর উপাদান তৈরী করেছিলেন তা আপনার ব্যবহার করতে দেয়।

নিচের চিত্রটি দেখায় কীভাবে পারমিশন যুক্ত করতে হয়:

```

<manifest>
...
<uses-permission
    android:name="android.permission.INTERNET"/>
<uses-permission
    android:name="android.permission.READ_SYNC_SETTINGS"/>
<uses-permission
    android:name="android.permission.WRITE_SYNC_SETTINGS"/>
<uses-permission
    android:name="android.permission.AUTHENTICATE_ACCOUNTS"/>
...
</manifest>

```

চুড়ান তভাবে, বাউন্ড Service ডিকে লয়ার করতে, যা ফের্মওয়াকর আপনার সিঙ্ক অফাডাপটের সাথে যোগাযোগ করতে ব্যবহার করে, <application> এর একটি চাইল্ড এলিমেন্ট হিসাবে আপনার অফাপমেনিফেস্ট টিনিচর XML যুক্ত করুন:

```

<service
    android:name="com.example.android.datasync.SyncService"
    android:exported="true"
    android:process=":sync">
    <intent-filter>com.example.android.datasync.provider
        <action android:name="android.content.SyncAdapter"/>
    </intent-filter>
    <meta-data android:name="android.content.SyncAdapter"
        android:resource="@xml/syncadapter" />
</service>

```

<intent-filter> এলিমেন্ট একটি ফিল্টার সেটআপ করে যা ইন্টেন্ট একশন android.content.SyncAdapter দ্বারা সিকর্য হয়, সিঙ্ক অফাডাপটর রান করানোর জন্য সিসেস্টম কতরূপ পাঠানো হয়। যখন ফিল্টার সিকর্য করানো হয়, আপনি যে বাউন্ড সার্ভিস তৈরী করেছেন সিসেস্টম তা শুরুর করে, এই উদাহরণে যেটা হচ্ছে SyncService। এটির বিউট android:exported="true" টি Service এ প্রবেশ করতে প্রসেসসে অনুরোধ মাদন করে আপনার অফাপকে (সিসেস্টম সহ) নয়। এটির বিউট android:process=":sync" টি সিসেস্টমকে একটি বৈশিষ্ট্য করার প্রসেসসে যার নাম sync তার মধ্যে Service কে রান করতে বলে। যদি আপনার যদি আপনার অফাপের মধ্যে মাল্টিপল সিঙ্ক অফাডাপটর থেকে থাকে যা এই প্রসেসসে করার করতে পারে, এটা ওভারহেড কমিয়ে দেয়।

<meta-data> এলিমেন্ট পূর্ববর্তী করেছেন এমন সিঙ্ক অফাডাপটর মেটাডেটা XML ফাইলের নাম প্রদান করে। android:name এটির বিউটিনে দর্শ করে যে এই মেটাডেটা সিঙ্ক অফাডাপটর ফের্মওয়াকর জনক। android:resource এলিমেন্ট মেটাডেটা ফাইলের নাম নির্দিষ্ট করে।

এখন আপনার কাছে সিঙ্ক অফাডাপটরের জন্য সকল উপাদান আছে। পরবর্তী অনুশীলনী আপনাকে দেখাবে কীভাবে আপনার সিঙ্ক অফাডাপটরকে রান করতে সিঙ্ক অফাডাপটর ফের্মওয়াকরকে বলতে হয়, একটি ইভেন্টের প্রতিতিকর্যায়ন নাটক বা একটি টিনিয়মিত সময়সূচী অনুযায়ী।

একটি সিস্টেমিক অ্যান্ড্রোইড অ্যাপটর রান করা

(<http://developer.android.com/training/sync-adapters/running-sync-adapter.html>)

এই কন্ট্রোলারের পূর্ববর্তকারী অনুশীলনীতে, আপনি শিখেছেন কীভাবে একটি সিস্টেমিক অ্যান্ড্রোইড অ্যাপটর উপাদান তৈরি করতে হয় যা ডাটাবেজের মধ্যে যথার্থভাবে, এবং কীভাবে অতিরিক্ত উপাদান যোগ করতে হয় যা সিস্টেমের মধ্যে আপনাকে সিস্টেমিক অ্যান্ড্রোইড অ্যাপটর প্লাগ করতে দেয়। একটি অ্যান্ড্রোইড ইনস্টল করার জন্য যা পরে যোজন এখন আপনার তাআছে যা একটি সিস্টেমিক অ্যান্ড্রোইড অ্যাপটর অন্তর্ভুক্ত করে, কিন্তু তু আপনি যেকোনো ডেভেলোপার দেখেছেন তার কোন টাইআসলে তা সিস্টেমিক অ্যান্ড্রোইড অ্যাপটর রান করতে পারেন না।

আপনার উচিত একটি সময়সূচির উপর ভিত্তি করে বার্ষিক ছুটি ইভেন্টের পরে একটি ফলাফল হিসাবে আপনার সিস্টেমিক অ্যান্ড্রোইড অ্যাপটর রান করার চেষ্টা করা। উদাহরণস্বরূপ, আপনি হয়তো হয় একটি নিম্নদির্ষ্ট সময় পর বা দিনের একটি নিম্নদির্ষ্ট সময়ে একটি নিয়মিত সময়সূচির (শিডিউল) উপর আপনার সিস্টেমিক অ্যান্ড্রোইড অ্যাপটর রান করতে চাইতে পারেন। আপনি হয়তো আপনার সিস্টেমিক অ্যান্ড্রোইড অ্যাপটর রান করতে চাইতে পারেন যখন ডিভাইসে ডাটাবেজের স্টোর করার ক্ষেত্রে কোন পরিবর্তন থাকে তখনও। আপনার উচিত সিস্টেমিক অ্যান্ড্রোইড অ্যাপটর একটি ইউজার একশনের পরে তৎক্ষণাত ফলাফল হিসাবে রান করাটা পরিহার করা, কারণ এটা করার মাধ্যমে আপনি সিস্টেমিক অ্যান্ড্রোইড অ্যাপটর ফের্মওয়াকে রিশিডিউল করার সাথে থের পুণর্ সূচিবধা পাবেন না। উদাহরণস্বরূপ বলা যায়, আপনার উচিত আপনার ইউজার ইন্টারফেসের মধ্যে একটি রিফ্রেশ বাটন যুক্ত করার বিষয়টা পরিহার করা।

আপনার সিস্টেমিক অ্যান্ড্রোইড অ্যাপটর রান করার জন্য নিম্নের কোন একটি বেছে নিতে পারেন:

যখন সান্তর্কারী ডাটাবেজ পরিবর্তন করে

একটি সান্তর্কারী থেকে আসা মেসেজের পরে তৎক্ষণাত ফলাফল হিসাবে সিস্টেমিক অ্যান্ড্রোইড অ্যাপটর রান করা, নিম্নের শর্তে যে সান্তর্কারী ভিত্তি করে ডাটাবেজ পরিবর্তন হয়েছে। এই পছন্দ আপনাকে কায়র্ সম্পাদনের কোন রূপ পরিবর্তন ছাড়াই বা সান্তর্কারী পোল করার মাধ্যমে বয়্যাটারির আয়ুষ্কাল অপচয় করা ছাড়াই সান্তর্কারী থেকে ডিভাইসে ডাটাবেজের রিফ্রেশ করতে দেয়

যখন ডিভাইসে ডাটাবেজ পরিবর্তন করে

একটি সিস্টেমিক অ্যান্ড্রোইড অ্যাপটর রান করা যখন ডিভাইসে ডাটাবেজ পরিবর্তন হয়। এই পছন্দ আপনাকে ডিভাইস থেকে পরিবর্তিত ডাটাবেজ সান্তর্কারী পঠাতে দেয়, এবং বিবেচনা করে উপকারী হয় যদি আপনার নিশ্চিত করার পরে যোজন হয়েছে সান্তর্কারীর সব সময় সবার্ষিক তথ্য রয়েছে। এই পছন্দ বাস্তবায়নের ক্ষেত্রে অকপট যদি আপনি আপনার কন্টেন্ট পরে ডাইভার যথার্থই ডাটাবেজে স্টোর করেন। যদি আপনি একটি স্টাব কন্টেন্ট পরে ডাইভার করতে থাকেন, ডাটাবেজ পরিবর্তন খুঁজে বের করাটা কঠিন হয়ে যেতে পারে।

যখন সিস্টেম একটি নটওয়াকর্ মেসেজ পাঠায়

একটি সিস্টেমিক অ্যান্ড্রোইড অ্যাপটর রান করা যখন অ্যান্ড্রয়েড সিস্টেম একটি নটওয়াকর্ মেসেজ পাঠায় যা ঐচ্ছিক/ওচ কানেকশন ওপেন ধারণ করে: এই মেসেজ হচ্ছে নটওয়াকর্ ফের্মওয়াকে একটি মেমোরিক অ্যাক্স। এই পছন্দ ব্যবহার করাটা সিস্টেমিক অ্যান্ড্রোইড অ্যাপটর সবার্ষিক ফলাফল হিসাবে রান করার একটা উপায় যা বিবর্তী (ইন্টারভ্যাল)-ভিত্তি সিস্টেমিক অ্যান্ড্রোইড অ্যাপটর রানের সাথে এটার একটি রত বয়্যে ব্যবহার বিবেচনা করুন।

নিয়মিত বিবর্তীতে (ইন্টারভ্যাল)

আপনি যে বিবর্তী (ইন্টারভ্যাল) পছন্দ করেছেন তার সমাপ্তির পরে একটি
সিঙ্ক অফ ডাউড আপটর রান করা বা এটাকে পরিত্যাগ করে একটি নির্দিষ্ট
সময়ে রান করুন।

পর্যায়াজন অনুসারে

একজন ইউজারের একশনের জবাবে সিঙ্ক অফ ডাউড আপটর রান করা। যাহোক,
সবচেয়ে ভালো ইউজার এক্সপেরিয়েন্স সরবরাহ করতে পার্থক্যমূলকভাবে
আপনার আরও সফলিকর্য (অটোমেটেড) পছন্দ উপর নির্ভর করা উচিত।
সফলিকর্য (অটোমেটেড) পছন্দ ব্যবহার করার মাধ্যমে আপনি
ব্যাকটিরি এবং নোটওয়ার্কের রিসোর্স সঞ্চয় করতে পারেন।

এই অনুশীলনের পরবর্তীতে পরিত্যাগ পছন্দ বিবেচনা করার উপায়
আলাচনা করা হবে।

একটি সিস্টেম অফ অ্যান্ড্রয়েড অ্যাপের রান করা যখন সার্ভারের ডাটাবেস পরিবর্তন করে

যদি আপনার অ্যাপ একটি সার্ভার থেকে ডাটাবেস রান সংফার করে এবং সার্ভারের ডাটাবেস পরিবর্তন করে, আপনি ডাটাবেস পরিবর্তনের পরে তির্যক য় ডাউনলেড করতে একটি সিস্টেম অফ অ্যান্ড্রয়েড অ্যাপের বয়বহার করুন। সিস্টেম অফ অ্যান্ড্রয়েড অ্যাপের রান করতে, সার্ভারের অ্যাপের মধ্যে একটি BroadcastReceiver এ একটি বিবেশ মেসেজ পাঠাতে হবে। এই মেসেজের জবাবে, আপনার সিস্টেম অফ অ্যান্ড্রয়েড অ্যাপের রান করতে সিস্টেম অফ অ্যান্ড্রয়েড অ্যাপের ফেরমওয়াকরকে কিসগনয়াল দিতে ContentResolver.requestSync() কল করুন।

গুগল ক্লাউড মেসেজিং (GCM) (<http://developer.android.com/google/gcm/index.html>) এই মেসেজ সিস্টেম কাজ তৈরী করার জন্য প্রয়োজনীয় সার্ভার এবং ডিভাইস উপাদান সরবরাহ করে। টরান্সফার সিকর্য করতে GCM বয়বহার স্ট্র্যাটাজী সরজনয় সার্ভারের কল করার চেয়ে আরও নিভর্রযোগয় এবং কমর্রদকষ। যখন পোলিং একটি Service আকাঙ্খাকরে যা সবসময় সিকর্য থাকে, GCM একটি বয়বহার BroadcastReceiver করে যা সিকর্য হয় যখন একটি মেসেজ পৌঁছায়। যখন পোলিং নিয়মিত বিবর্তীতে বয়বহারের পাওয়ার বয়বহার করে এমনকি যদিও আপডেট বিবদয়মান আছে, GCM শুধু মেসেজ পাঠায় যখন প্রয়োজন হয়।

নোট: যদি আপনার সিস্টেম অফ অ্যান্ড্রয়েড অ্যাপের সিকর্য করতে সকল ডিভাইসে বর্রদকাস্ট করার মাধ্যমে আপনি GCM বয়বহার করেন যেখানে অ্যাপ ইনস্টল করা হয়েছে, মেনে রাখবেন যে তারা আপনার মেসেজ মেটামুটিভাবে একই সময়ে গ্রহণ করে। এই অবস্থা একই সময়ে রান করতে আপনার সিস্টেম অফ অ্যান্ড্রয়েড অ্যাপের মাল্টিপল ইনস্ট্যান্স করতে পারে, সার্ভারের এবং নেটওয়ার্ক ওভারলেড করে। সকল ডিভাইসে বর্রদকাস্টের জন্য এই পিরিস্থিতি পিরহার করতে, আপনার উচিত একটি সময়কালের জন্য সিস্টেম অফ অ্যান্ড্রয়েড অ্যাপের শুরুর বিবেচনা করা যা পরে টািডিভাইসের জন্য ইউনিক (অদিবর্তীয়)।

নিম্নোক্ত কোড চিত্রটি আপনাকে দেখাবে কীভাবে একটি ইনকামিং GCM মেসেজের জবাবে requestSync() রান করা হয়:

```

public class GcmBroadcastReceiver extends BroadcastReceiver {
    ...
    // Constants
    // Content provider authority
    public static final String AUTHORITY = "com.example.android.datasync.provider"
    // Account type
    public static final String ACCOUNT_TYPE = "com.example.android.datasync";
    // Account
    public static final String ACCOUNT = "default_account";
    // Incoming Intent key for extended data
    public static final String KEY_SYNC_REQUEST =
        "com.example.android.datasync.KEY_SYNC_REQUEST";
    ...
    @Override
    public void onReceive(Context context, Intent intent) {
        // Get a GCM object instance
        GoogleCloudMessaging gcm =
            GoogleCloudMessaging.getInstance(context);
        // Get the type of GCM message
        String messageType = gcm.getMessageType(intent);
        /*
         * Test the message type and examine the message contents.
         * Since GCM is a general-purpose messaging system, you
         * may receive normal messages that don't require a sync
         * adapter run.
         * The following code tests for a a boolean flag indicating
         * that the message is requesting a transfer from the device.
         */
        if (GoogleCloudMessaging.MESSAGE_TYPE_MESSAGE.equals(messageType)
            &&
            intent.getBooleanExtra(KEY_SYNC_REQUEST)) {
            /*
             * Signal the framework to run your sync adapter. Assume that
             * app initialization has already created the account.
             */
            ContentResolver.requestSync(ACCOUNT, AUTHORITY, null);
            ...
        }
        ...
    }
    ...
}

```

সিঙ্ক অ্যাডাপটর রান করা যখন কন্টেন্ট পর্ভাইডার ডাটা পরিবর্তন করে

যদি আপনার অ্যাডাপ্টার একটি কন্টেন্ট পর্ভাইডারের মধ্যে ডাটা সংগ্রহ করে, এবং আপনি সাবস্ক্রাইব আপডেট করতে চান, যখনই আপনি পর্ভাইডার আপডেট করেন, আপনি আপনার সিঙ্ক অ্যাডাপটর সবেয়ালিকরয়ভাবে রান করতে আপনার অ্যাডাপ্টার সেটআপ করতে পারেন। এটা করতে কন্টেন্ট পর্ভাইডারের জন্য আপনি একটি অবজাভারার রেজিস্টার করেন। যখন আপনার কন্টেন্ট পর্ভাইডারের ডাটা পরিবর্তন করে, কন্টেন্ট পর্ভাইডার ফর্মওয়ার্ক অবজাভারার কল করে। অবজাভারারের মধ্যে ফর্মওয়ার্ক কে আপনার সিঙ্ক অ্যাডাপটর রান করতে বলতে `requestSync()` কল করুন।

নোট: যদি আপনি স্টাব কন্টেন্ট পর্ভাইডার ব্যবহার করেন, কন্টেন্ট পর্ভাইডারের আপনার কোন ডাটানাই এবং `onChange()` কে কখনই কল করা হয় না। এই ক্ষেত্রে, ডিভাইস ডাটাবেজ পরিবর্তন চিহ্নিত করতে আপনার নিজস্ব পেরিকরয় (মেকানিজম) পর্দাান করতে হবে। এই পেরিকরয় `requestSync()` কল করার জন্য দায়িতব পর্াপ্ত যখন ডাটা পরিবর্তন করে।

আপনার কন্টেন্ট পর্ভাইডারের জন্য একটি অবজাভারার তৈরী করতে, ক্লাস `ContentObserver` পর্সারিত করুন এবং এর `onChange()` পদ্ধতির উভয় রূপই বাস্তবায়ন করুন। `onChange()` এর মধ্যে, সিঙ্ক অ্যাডাপটর শুরুর করতে `requestSync()` কল করুন।

অবজাভারার রেজিস্টার করতে, `registerContentObserver()` একটি কলের মধ্যে একটি আগুরমেন্টিহসাবে এটাকে পাস করুন। এই কলে, আপনি যে ডাটাবেজ দেখতে চান তার জন্য একটি কন্টেন্ট URI এ আপনাকে পাস করতে হবে। কন্টেন্ট পর্ভাইডার ফর্মওয়ার্কিট এই ওয়াচ URI কে কন্টেন্ট URI এ তুলনাকরে যা `ContentResolver` পদ্ধতিতে আগুরমেন্টিহসাবে পাস হয়েছে যা আপনার পর্ভাইডার পরিমিত করে, যেমন, `ContentResolver.insert()`। যদি সেখানে কোন সমরূপ কিছুর থাকে, আপনার `ContentObserver.onChange()` এর বাস্তবায়ন কে কল করা হয়।

নিচের কোড চিত্রটি দেখাচ্ছে যে কীভাবে আপনি একটি `ContentObserver` নিধরারণ করবেন যা `requestSync()` কল করে যখন একটি টেবিবল পরিবর্তন করে:


```

public class MainActivity extends FragmentActivity {
    ...
    // Constants
    // Content provider scheme
    public static final String SCHEME = "content://";
    // Content provider authority
    public static final String AUTHORITY = "com.example.android.datasync.provider";
    // Path for the content provider table
    public static final String TABLE_PATH = "data_table";
    // Account
    public static final String ACCOUNT = "default_account";
    // Global variables
    // A content URI for the content provider's data table
    Uri mUri;
    // A content resolver for accessing the provider
    ContentResolver mResolver;
    ...
    public class TableObserver extends ContentObserver {
        /*
         * Define a method that's called when data in the
         * observed content provider changes.
         * This method signature is provided for compatibility with
         * older platforms.
         */
        @Override
        public void onChange(boolean selfChange) {
            /*
             * Invoke the method signature available as of
             * Android platform version 4.1, with a null URI.
             */
            onChange(selfChange, null);
        }
        /*
         * Define a method that's called when data in the
         * observed content provider changes.
         */
        @Override
        public void onChange(boolean selfChange, Uri changeUri) {
            /*
             * Ask the framework to run your sync adapter.
             * To maintain backward compatibility, assume that
             * changeUri is null.
             */
            ContentResolver.requestSync(ACCOUNT, AUTHORITY, null);
        }
        ...
    }
    ...
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        ...
        // Get the content resolver object for your app
        mResolver = getContentResolver();
        // Construct a URI that points to the content provider data table
        mUri = new Uri.Builder()
            .scheme(SCHEME)
            .authority(AUTHORITY)
            .path(TABLE_PATH)
            .build();

        /*
         * Create a content observer object.
         * Its code does not mutate the provider, so set
         * selfChange to "false"
         */
        TableObserver observer = new TableObserver(false);
        /*
         * Register the observer for the data table. The table's path
         * and any of its subpaths trigger the observer.
         */
        mResolver.registerContentObserver(mUri, true, observer);
        ...
    }
    ...
}

```

একটি টেনেটওয়াকার্ মেসেজের পর সিঙ্ক অ্যাডাপটর রান করা

যখন একটি টেনেটওয়াকার্ কানেকশন পাওয়া যায়, অ্যান্ডার্নেয়েড সিসেস্টম ডিভাইসে সর TCP/IP কানেকশন ওপেন ধরে রাখতে কিছু সেকেন্ড পরপরই একটি মেসেজ পাঠায়। এই মেসেজ পরিতটা অ্যাডাপ্টর ContentResolver এ যায়। setSyncAutomatically() কল করার মাধ্যমে, আপনি সিঙ্ক অ্যাডাপটর রান করতে পারবেন যখনই ContentResolver মেসেজ গ্রহণ করে।

যখন টেনেটওয়াকার্ মেসেজ পাঠানো হয় তখন আপনার সিঙ্ক অ্যাডাপটর রান করতে শিডিউল করে, আপনি নিশ্চিত করেন যে যখনই টেনেটওয়াকার্ সহজপরাপয হয় আপনার সিঙ্ক অ্যাডাপটর সবসময় রান করার জন্যে শিডিউল করা থাকে। ডাটাপরিবর্তনের জবাবে আপনাকে একটি ডাটাপরিবর্তন সফার বল পূবরক করতে হয় না, কিন্তু আপনি নিশ্চিত করতে চান যে আপনার ডাটানিয়মিতভাবে আপডেট হয় তখন এই পছন্দ বয্বহার করুন। একইভাবে, যদি আপনি আপনার সিঙ্ক অ্যাডাপটরের জন্যে কোন নির্দিষ্ট শিডিউল না চান, কিন্তু আপনি চান এটানিয়মিতভাবে রান করুক সেক্ষেত্রে এই পছন্দ বয্বহার করতে পারেন।

যেহেতু পদ্ধতি setSyncAutomatically() টি addPeriodicSync() কে নিস্কর্য করেনা, আপনার সিঙ্ক অ্যাডাপটর অল্প সময়ে মধে বারবার সিকর্য হতে পারে। যদি আপনি আপনার সিঙ্ক অ্যাডাপটর নিয়মিত শিডিউলে পযর্ায়কর্মে রান করতে চান, আপনার উচিত setSyncAutomatically() নিস্কর্য করা।

নিম্নের কড চিত্রটি আপনাকে দেখাচ্ছে যে একটি টেনেটওয়াকার্ মেসেজের জবাবে সিঙ্ক অ্যাডাপটর রান করতে কীভাবে আপনার ContentResolver কনফিগার করতে হয়:

```
public class MainActivity extends FragmentActivity {  
    ...  
    // Constants  
    // Content provider authority  
    public static final String AUTHORITY = "com.example.android.datasync.provider";  
    // Account  
    public static final String ACCOUNT = "default_account";  
    // Global variables  
    // A content resolver for accessing the provider  
    ContentResolver mResolver;  
    ...  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        ...  
        // Get the content resolver for your app  
        mResolver = getContentResolver();  
        // Turn on automatic syncing for the default account and authority  
        mResolver.setSyncAutomatically(ACCOUNT, AUTHORITY, true);  
        ...  
    }  
    ...  
}
```

পযৰ্ায়কৰ্মে সিসঙ্ৰক অয্ৰাডডাপটৰ ৰান কৰা যখন

ৰানেৰ মধেয্ৰ অপেক্ষাকৰ্তে বাৰিদিনেৰ একটি সময়ে এটাে কৰান কৰা বা
উভয় কৰাৰ জনয্ৰ একটি সময়কাল সেটি ঙ্ৰকৰাৰ মাধয্ৰমে আপনি পযৰ্ায়কৰ্মে
আপনাৰ সিসঙ্ৰক অয্ৰাডডাপটৰ ৰান কৰাতে পোৱেন। পযৰ্ায়কৰ্মে আপনাৰ
সিসঙ্ৰক অয্ৰাডডাপটৰ ৰান কৰাটো আপনাে আপনাৰ সাভৰ্ৰাৱেৰ হালনাগাদ
বিবৰিত (ইন্টাৰভয্ৰাল) মেটাটামুটিভােব ময্ৰাচ কৰতে দেয়।

একইভােব, আপনি ডিভাইস থেকে ডাটা আপলেড কৰতে পোৱেন যখন আপনাৰ
সাভৰ্ৰাৱেৰ অপেক্ষাকৃত নিস্কৰ্য় হয়, ৰাতেৰ্ ৰান কৰাতে আপনাৰ সিসঙ্ৰক
অয্ৰাডডাপটৰ শিডিউল কৰাৰ মাধয্ৰমে। অধিকাংশ ইউজাৰ ৰাতেৰে বলায়
পাওয়াৰ অন বা প্লাগ ইন কৰাে থেকে বিবৰত থাকে, সুতৰাে এই সময়টি
সাধাৰণভােব সহজলভয্ৰ। অধিকন্তু, আপনাৰ সিসঙ্ৰক অয্ৰাডডাপটৰেৰ মতে
ডিভাইস একই সময়ে অন্য কাৰ ৰান কৰেনা। যদি আপনি এই পদ্ধতি গৰ্হণ
কৰে থাকেন, আপনাে নিশ্চিত কৰতে হেবে যে পৰ্তিটো ডিভাইস একটু
ভিন্ৰন সময়ে একটি ডাটা ট্ৰান্সফাৰ কৰে থাকে। যদি সকল ডিভাইস আপনাৰ
সিসঙ্ৰক অয্ৰাডডাপটৰ একই সময়ে ৰান কৰে, আপনি সম্ভবত আপনাৰ সাভৰ্ৰাৱেৰ
এব্ৰেসল পৰ্ভাইডাৰ ডাটা নেটওয়ার্কৰ ওভাৰলেড এ আছেন।

সাধাৰণভােব, পযৰ্ায়কৰ্মিক ৰানেৰ অথৰ্ থাকে যদি আপনাৰ ইউজাৰেৰ
তাক্ষণিক আপডেটৰ পৰ্য়োজন না হয় কিন্তু নিয়মিত আপডেট থাকে
আকাঙ্ৰখাকে। পযৰ্ায়কৰ্মিক ৰানেৰে অথৰ্ও থাকে যদি আপনি
হালনাগাদ ডাটা এব্ৰক্ৰদুৰ্ সিসঙ্ৰক অয্ৰাডডাপটৰেৰ কমৰ্দ্দক্ৰতাৰ
মধেয্ৰ ভাৰসাময্ৰ কৰতে চান যাি ডিভাইস ৰিসেসাৰেৰ অতি-বয্ৰবহাৰ কৰে
না।

নিয়মিত বিবৰিতে আপনাৰ সিসঙ্ৰক অয্ৰাডডাপটৰ ৰান কৰাতে `addPeriodicSync()` কল
কৰুন। এটা একটি নিদিৰ্ৰ্ষ্ট সময় অতিকৰ্ানত হওয়ার পৰ ৰান কৰতে
আপনাৰ সিসঙ্ৰক অয্ৰাডডাপটৰেৰ শিডিউল কৰে। যেহেতু সিসঙ্ৰক অয্ৰাডডাপটৰ
ফেৰ্ৰমওয়ার্কৰ অন্য সিসঙ্ৰক অয্ৰাডডাপটৰ ফেৰ্ৰমওয়ার্কৰ
কমৰ্দ্দক্ৰতাপাদনেৰ জনয্ৰ দায়ী এব্ৰবয্ৰাটোৱিৰ দক্ৰ্ষতাবাড়ােতে চেষ্টা
কৰে, অতিকৰ্ানত সময় কয়েক সেকেণ্ডে ভিন্ৰন হতে পাৰে। এছাড়াও,
ফেৰ্ৰমওয়ার্কৰ আপনাৰ সিসঙ্ৰক অয্ৰাডডাপটৰ ৰান কৰেব না যদি নেটওয়ার্ক
না থাকে।

লক্ৰ্ষ কৰুন যে `addPeriodicSync()` দিনেৰ নিদিৰ্ৰ্ষ্ট সময়ে সিসঙ্ৰক অয্ৰাডডাপটৰ
ৰান কৰায় না। মেটাটামুটিভােব পৰ্তিদিনেৰ একই সময়ে সিসঙ্ৰক
অয্ৰাডডাপটৰ ৰান কৰাতে একটি টিটৰ্গাৰিহসােব একটি পূনাৰাব্ৰত্ৰত কৰা
এলামৰ্ৰবয্ৰবহাৰ কৰুন। `AlarmManager` এৰেফাৰেন্স ডকুমেণ্টশনে
পূনাৰাব্ৰত্ৰত কৰা এলামৰ্ৰবিষয়ে আৰও বিসৰ্তাৰিতভােব আলাচনা কৰা
হয়েছে। আপনি যদি দিনেৰ সময়কালীন টিটৰ্গাৰেসট কৰতে পদ্ধতি
`setInexactRepeating()` বয্ৰবহাৰ কৰেন যাৰ কিছু বৈচিতৰ্ৰৰেয়েছে, আপনাৰ উচিত
তখনও শূৰুৰ সময়ৈ দবচয়ন কৰা এটা নিশ্চিত কৰতে যে বিভিন্ৰন ডিভাইস
থেকে ৰান কৰা সিসঙ্ৰক অয্ৰাডডাপটৰ বিসৰ্ত্ৰত হয়।

পদ্ধতি `addPeriodicSync()`, `setSyncAutomatically()` কে নিস্কৰ্য় কৰেনা, তাই আপনি
অপেক্ষাকৃত কম সময়ে ৰান কৰা মািল্ৰটপল সিসঙ্ৰকে পেত পোৱেন। এছাড়াও,
শুধু কিছু সিসঙ্ৰক অয্ৰাডডাপটৰ নিয়ন্তৰ্গফলয্ৰাগস `addPeriodicSync()` এৰ
পৰ্তি কলেৰ মধেয্ৰ অনুমেদিদত; যে ফলয্ৰাগেৰ অনুমেদনেই তা
`addPeriodicSync()` এৰ জনয্ৰেফাৰেন্স ডকুমেণ্টশনে আলাচনা কৰা হয়েছে।

নিচেৰেকে ডিচিতৰ্ৰটি দেখায় কীভােব পযৰ্ায়কৰ্মিক সিসঙ্ৰক
অয্ৰাডডাপটৰ ৰান শিডিউল কৰতে হয়:

```

public class MainActivity extends FragmentActivity {
    ...
    // Constants
    // Content provider authority
    public static final String AUTHORITY = "com.example.android.datasync.provider";
    // Account
    public static final String ACCOUNT = "default_account";
    // Sync interval constants
    public static final long MILLISECONDS_PER_SECOND = 1000L;
    public static final long SECONDS_PER_MINUTE = 60L;
    public static final long SYNC_INTERVAL_IN_MINUTES = 60L;
    public static final long SYNC_INTERVAL =
        SYNC_INTERVAL_IN_MINUTES *
        SECONDS_PER_MINUTE *
        MILLISECONDS_PER_SECOND;
    // Global variables
    // A content resolver for accessing the provider
    ContentResolver mResolver;
    ...
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        ...
        // Get the content resolver for your app
        mResolver = getContentResolver();
        /*
        * Turn on periodic syncing
        */
        ContentResolver.addPeriodicSync(
            ACCOUNT,
            AUTHORITY,
            null,
            SYNC_INTERVAL);
        ...
    }
    ...
}

```

চাহিদা অনুযায়ী সঙ্ক অ্যাডডাপটর রান করা

ইউজারের অনুবোধের জবাবে আপনার সঙ্ক অ্যাডপটর রান করা হচ্ছে একটি সঙ্ক অ্যাডপটর রান করার সবচেয়ে কম ক্যাঙ্ক থত কৌশল। ফের্মওয়াকর্ বয্াট্াির পাওয়ার স্ারক্ষণ করার জন্য বি্েষভাবে তৈরী করা হয়েছ যখন এটা একটি শিডিউল অনুযায়ী সঙ্ক অ্যাডপটর রান করে। পছন্দ (অপশন) যা ডাটাপরিবতর্নের জবাবে একটি সঙ্ক রান করে তা বয্াট্াির পাওয়ার কায়র্ করীভাবে বয্বহার করে, যেহেতু পাওয়ার নতুন ডাটাপর্দান করার কাজে বয্বহৃত হয়।

তুলনামূলকভাবে, ইউজারকে তার চাহিদার উপর একটি সঙ্ক রান করতে দেয়ার মােন হচ্ছে যে সঙ্ক নিজে নিজেই রান করেছ, যােনটওয়ার্ এবং পাওয়ার রিসেসাসর্ বয্বহারে অদক্ষ। এছাড়াও, চাহিদার উপর একটি সঙ্ক রান করতে দেয়া ইউজারকে একটি সঙ্ক রিকেয়েস্ট রিদি কনিযে যায় এমন কিসেখানে কোন পর্মানেনই যে ডাটাপরিবতর্ন হয়েছ এবং একটি সঙ্ক রান করা যা ডাটাকে রিফর্শ করে না তা বয্াট্াির পাওয়ার অকায়র্ কর বয্বহার। সাধারণভাবে, আপনার অ্যােপর উচিত হয় একটি সঙ্ক সিকর্য় করতে অন্য সিসগনাল বয্বহার করা বা ইউজার ইনপুট ছাড়াই নিয়মিত বিবর্তীতে তাদের শিডিউল করা হয়।

কিন্তু, যদি আপনি এখনও সঙ্ক অ্যাডপটর চাহিদার উপর রান করতে চান, একটি ম্যানুয়েল সঙ্ক অ্যাডপটর রান এর জন্য সঙ্ক অ্যাডপটর ফ্লয়্াগ সেট করুন, তারপর `ContentResolver.requestSync()` কল করুন।

নিচের ফ্লয়্াগ গুলে াদিয়ে চাহিদাকৃত টর্ান সফারগুলাে রান করুন:

SYNC_EXTRAS_MANUAL/ সঙ্ক এক্ সটর্াস ম্যানুয়েল

একটি ম্যানুয়েল সঙ্ক ফেসর্ করে (বল পর্য়োগ)। সঙ্ক অ্যাডপটর ফের্মওয়াকর্ বিদয়্ মানসেটি াকে এড়িয়ে যায়, যেমন ফ্লয়্াগ `setSyncAutomatically()` কতর্ক সেট হয়।

SYNC_EXTRAS_EXPEDITED/ সঙ্ক এক্ সটর্াস এক্ সপেক্টেড

অবলম্বে শুরু করতে সঙ্ক কে ফেসর্ করে (বল পর্য়োগ)। যদি আপনি এটা সেট করে না থাকেন, সিসেস্টম সঙ্ক রিকেয়েস্ট রান করার জন্য কয়েক সেকেন্ড অপেক্ষাকরে, কারন এটা অতি অল্প সময়ে অনেক রিকেয়েস্ট শিডিউল করার মাধ্যমে বয্াট্াির বয্বহার অপটিমাইজ করার চেষ্টা করে।

নিচের কোড চিতর্টি আপনাকে দেখায় একটি বাটন ক্লক করার জবাবে কীভাবে `requestSync()` কল করতে হয়:

```

public class MainActivity extends FragmentActivity {
    ...
    // Constants
    // Content provider authority
    public static final String AUTHORITY =
        "com.example.android.datasync.provider"
    // Account type
    public static final String ACCOUNT_TYPE = "com.example.android.datasync";
    // Account
    public static final String ACCOUNT = "default_account";
    // Instance fields
    Account mAccount;
    ...
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        ...
        /*
         * Create the dummy account. The code for CreateSyncAccount
         * is listed in the lesson Creating a Sync Adapter
         */

        mAccount = CreateSyncAccount(this);
        ...
    }
    /**
     * Respond to a button click by calling requestSync(). This is an
     * asynchronous operation.
     *
     * This method is attached to the refresh button in the layout
     * XML file
     *
     * @param v The View associated with the method call,
     * in this case a Button
     */
    public void onRefreshButtonClick(View v) {
        ...
        // Pass the settings flags by inserting them in a bundle
        Bundle settingsBundle = new Bundle();
        settingsBundle.putBoolean(
            ContentResolver.SYNC_EXTRAS_MANUAL, true);
        settingsBundle.putBoolean(
            ContentResolver.SYNC_EXTRAS_EXPEDITED, true);
        /*
         * Request the sync for the default account, authority, and
         * manual sync settings
         */
        ContentResolver.requestSync(mAccount, AUTHORITY, settingsBundle);
    }
}

```

ইউজার ইনফো এবং লোকেশন (অবস্থান) সহকারে অ্যাপ তৈরী করা

(<http://developer.android.com/training/building-userinfo.html>)

এই ক্লাসগুলে আপনাকে শেখাবে কীভাবে আপনার অ্যাপে ইউজার পারসোনালাইজেশন() যুক্ত করতে হয়। এটা করার কিছু উপায়ের মধ্যে রয়েছে ইউজারকে চিহ্নিত করা, তাদের সম্পর্কে পর্দা স্কেচ তথ্য পর্দা নকশার মাধ্যমে এবং তাদের পরিপাশি বর্ক অবস্থাসম্পর্কে তথ্য পর্দা নকশার মাধ্যমে।

১. কনটেন্ট অ্যাক্টস ডাটাবেস পরবেশ করা

কীভাবে অ্যান্ড্রয়েডের কন্টেন্ট প্রদর্শন এবং ডেটাবেস, কনটেন্ট প্রদর্শন আইডার ব্যবহার করা যায়, কনটেন্ট প্রদর্শন এবং তাদের বিস্তারিত এবং কনটেন্ট তথ্য পরিবর্তন।

- কনটেন্ট অ্যাক্ট লিস্ট প্রদর্শন
- কনটেন্ট এর জন্য বিবরণ প্রদর্শন
- ইন্টেন্ট ব্যবহার করে কনটেন্ট পরিবর্তন
- কনটেন্ট অ্যাক্ট ব্যবহার করে প্রদর্শন

২. আপনার অ্যাপে লোকেশন-অ্যাওয়ারের তৈরী করা

কীভাবে ইউজারের বর্তমান অবস্থান পাওয়ার মাধ্যমে আপনার অ্যাপে অবস্থান-সচেতন বৈশিষ্ট্য (লোকেশন-অ্যাওয়ার ফিচার) যোগ করা যায়।

- বর্তমান লোকেশন (অবস্থান) প্রদর্শন
- লোকেশন (অবস্থান) আপডেট গ্রহণ
- লোকেশন (অবস্থান)
- এবডবল পয়েন্ট তৈরী করা এবং মিনিমিউম করা
- ইউজারের বর্তমান একটি ভিত্তি সনাক্ত করা
- ম্যাক লোকেশন ব্যবহার করে পরীক্ষা

Table of Contents

বাংলায় অর্থান্দেরয়েড সহায়িকা	2
মন্তব্য/সংশোধন/অবদান	2
অ্যাপিলেশন তৈরীর কাজ শুরু করুন	4
আপনার পরামর্শ অ্যাপ তৈরী করুন	4
অর্থান্দেরয়েড পরজেক্ট তৈরী করা	4
ইক্লিপস দিয়ে পরজেক্ট তৈরী করা	8
কমান্ড লাইন টুলস দিয়ে পরজেক্ট তৈরী করুন	8
অ্যাপ রান করা	8
একটাসিটিয়কারডিভাইস রান করুন	13
ইমুয়লিটের রান করুন	13
একটি সরল ইউজার ইন্টারফেস তৈরী করা	16
লিনিয়ার লেআউট তৈরী করুন	17
টেক্সট ফিল্ড এ্যাড (সংযোজন) করুন	18
সিস্টারিং রিসোর্স এ্যাড (সংযোজন) করুন	20
বাটন এ্যাড (সংযোজন) করুন	20
স্ক্রলিং পরামর্শ ভরাট করার মতো কন্ট্রোল ইনপুট বক্স তৈরী করুন	22
অন্য একটি ডিভিডিটি শুরু করা	22
সেন্ড বাটন কন্ট্রোল প্যানেল করা	22
ইন্টেন্ট তৈরী করা	26
দ্রব্যীয় কায়ার্কর্ম (একটি ডিভিডিটি) শুরু করুন	26
দ্রব্যীয় একটি ডিভিডিটি তৈরী করুন	29
টাইম টেলিস্টারিং এ্যাড করুন	31
এটাকে মিনিফেস্ট এ্যাড করুন	32
ইন্টেন্ট গরুহণ করুন	33
মেসেজ ডিসপেপ্ল করুন	34
একশন বার এ্যাড (সংযোজন) করা	34
এই অধ্যায়ে এর অনুষঙ্গীলনী সমূহ	34
একশন বার সেট আপ করা	34
শুধুমাত্র অর্থান্দেরয়েড ৩.০ এবং এরচেয়ে উন্নত সংস্করণকে সাপোর্ট করুন	34
অর্থান্দেরয়েড ২.১ এবং এরচেয়ে উন্নত সংস্করণকে সাপোর্ট করুন	39
একশন বাটন এ্যাড (সংযোজন) করা	40
টগথ এ একশনকে সুনির্দিষ্ট করুন	41
একশন বারের একশন এ্যাড করুন	41
একশন বাটন কন্ট্রোল প্যানেল করা	41
লে-লেভেল একটি ডিভিডিটির জন্যে আপ বাটন এ্যাড (সংযোজন) করা	45
একশন বার টিকে স্টাইল করা	45
অর্থান্দেরয়েড থীম ব্যবহার করুন	48
ব্যাকগ্রাউন্ড কাস্টমাইজ করুন	48
শুধুমাত্র অর্থান্দেরয়েড ৩.০ এবং এরচেয়ে উন্নত সংস্করণের জন্যে	48
অর্থান্দেরয়েড ২.১ এবং এরচেয়ে উন্নত সংস্করণের জন্যে	52
টেক্সট কালার কাস্টমাইজ	52
শুধুমাত্র অর্থান্দেরয়েড ৩.০ এবং এরচেয়ে উন্নত সংস্করণের জন্যে	52
অর্থান্দেরয়েড ২.১ এবং এরচেয়ে উন্নত সংস্করণের জন্যে	55
ট্যাব ইন্ডিকেটর কাস্টমাইজ করা	56
শুধুমাত্র অর্থান্দেরয়েড ৩.০ এবং এরচেয়ে উন্নত সংস্করণের জন্যে	56
অর্থান্দেরয়েড ২.১ এবং এরচেয়ে উন্নত সংস্করণের জন্যে	59
একশন বার ওভারলে করা	60
ওভারলে মেড স্ক্রলিং করুন	60
তর্ অর্থান্দেরয়েড ৩.০ এবং এরচেয়ে উন্নত সংস্করণের জন্যে	60
অর্থান্দেরয়েড ২.১ এবং এরচেয়ে উন্নত সংস্করণের জন্যে	60
লেআউট টপ মার্জিন নির্দিষ্ট করুন	60
বিভিন্ন ডিভাইসকে সাপোর্ট করুন	60

এই অধ্যয়নের অনুষঙ্গীলননী সমূহ	60
ভিন্‌ন ভিন্‌ন ভাষাকে সাপোর্ট করা	60
লোকালিডের কন্ট্রোল এবং স্ট্রিং ফাইল তৈরী করা	68
স্ট্রিং রিসোর্সের ব্যবহার করুন	68
ভিন্‌ন ভিন্‌ন সিস্টেমকে সাপোর্ট করা	71
ভিন্‌ন ভিন্‌ন লেআউট তৈরী করুন	72
বিভিন্ন ধরনের বিটম্যাপ তৈরী করুন	74
ভিন্‌ন প্ল্যাটফর্মের সংস্করণকে সাপোর্ট করা	75
মিনিমাম (ন্যূনতম) এবং ট্যাগের টিএপিআই লেভেল সূচিন্দিদর্শক করা	75
রানটাইমে সিস্টেম সংস্করণকে চেক করুন	75
প্ল্যাটফর্মের স্টাইল এবং থিম ব্যবহার	78
একটি টিভি টিলাইফসাইকেল ব্যবস্থাপনা	79
এই অধ্যয়নের অনুষঙ্গীলননী সমূহ	80
একটি টিভি টিশুরুর করা	80
লাইফসাইকেল কলব্যবহার সম্পর্কে ধারণা	82
আপনার অ্যাপের লঞ্চার একটি টিভি টি সূচিন্দিদর্শক করুন	84
একটি টিনতুন ইনস্টলেশন তৈরী করুন	85
একটি টিভি টি রিবিলেপ	87
একটি টি একটি টিভি টি পজ এবং রিজিউম করা	88
একটি টিভি টিতে পজ দেওয়া	88
আপনার একটি টিভি টি রিজিউম (পুনরায় শুরুর) করা	88
একটি টিভি টি স্টপ এবং রিস্টার্ট করা	91
আপনার একটি টিভি টি বন্ধ (স্টপ) করুন	93
আপনার একটি টিভি টি স্টার্ট/রিস্টার্ট করুন	93
একটি টিভি টি পুনরায় তৈরী করা	96
আপনার একটি টিভি টি সেস্টট (অবস্থান) সেভ করা	96
আপনার একটি টিভি টি সেস্টটকে (অবস্থান) রিসেট করার করুন	99
ফরম্যাগমেন্ট সহ ডাইনামিক ইউজার ইন্টারফেস তৈরী করা	99
এই অধ্যয়নের অনুষঙ্গীলননী সমূহ	99
একটি টি ফরম্যাগমেন্ট তৈরী করা	99
একটি টি ফরম্যাগমেন্ট ক্লাস তৈরী করুন	103
XML ব্যবহার করে একটি টিভি টিতে ফরম্যাগমেন্ট সংযোজন করুন	104
একটি টিনমনীয় (ফেলিক্সবল) ইউজার ইন্টারফেস তৈরী করুন	104
রানটাইমে একটি টি একটি টিভি টিতে ফরম্যাগমেন্ট সংযোজন (এ্যাড) করুন	107
একটি ফরম্যাগমেন্টের সাথে আরেকটি ফরম্যাগমেন্টের প্রতিস্থাপন (রিপ্লেস)	109
অন্য ফরম্যাগমেন্টের সাথে যোগাযোগ	110
একটি টি ইন্টারফেস নিধরারন করা	111
ইন্টারফেস বাস্তবায়ন করা	112
ফরম্যাগমেন্ট বাতর্ক (মেসেজ) পেয়েছি দি	113
ডাটাবেসেভ করা	114
এই অধ্যয়নের অনুষঙ্গীলননী সমূহ	114
কি-ভ্যালু সেটে সেভ করা	114
একটি Shared Preferences এহয়ানে ডললাভ করা	117
শেয়ারডিপি রফারেন্স লেখা (রাইট)	118
শেয়ারডিপি রফারেন্স থেকে পাঠ করা	118
ফাইলে সেভ করা	118
ইন্টারনাল বা এক্সটানরাল সেস্টারেজ পছন্দ করা	121
এক্সটানরাল সেস্টারেজের জন্য অনুষঙ্গিত গর্হণ	121
ইন্টারনাল সেস্টারেজ ফাইলে সেভ করা	123
এক্সটানরাল সেস্টারেজ ফাইলে সেভ করা	125
ফরম্যাগমেন্টের অনুষঙ্গীলন	125
ফাইল ডিলিট করা	125
SQL ডাটাবেজে ডাটাবেসেভ করা	125
স্কিম (Schema) এবং কন্ট্রোল টিনধরার	125

SQL হ্রস্বকর্তৃক ডাটাবেজের ক্ষমতা	131
ডাটাবেজ থেকে তথ্য পঠ করা	134
ডাটাবেজ থেকে তথ্য ডিলিট করা	135
ডাটাবেজ আপডেট করা	135
অন্য অ্যাপের সাথে সম্মিলিত স্থাপন	135
এই অধ্যায়ের অন্তর্ভুক্ত শীলন সমূহ	135
ইউজারকে অন্য অ্যাপে পাঠানো	135
একটি ইমপিট ইনস্ট্রাকশন তৈরি করা	140
ইনস্ট্রাকশনটি গুরুত্বপূর্ণ করে একটি অ্যাপ আফ্রিকা যাত্রা করা	142
ইনস্ট্রাকশনটি দিয়ে একটি টিভি শিরু করা	143
একটি অ্যাপ চুজার পর্দা শিরু করা	144
একটি একটি টিভি টেবিলের জাল টপাওয়া	144
একটি টিভি শিরু করা	144
রেজাল্টটি গুরুত্বপূর্ণ (রিসিভ) করা	147
বেবানাস: কনটেন্ট ডাটাপডা	148
অন্য অ্যাপকে আপনার অ্যাপ শিরু করতে দেয়া	149
ইনস্ট্রাকশনটি ফিল্টার সফ্টওয়্যার (এক্স) করা	150
আপনার একটি টিভি টেবিল ইনস্ট্রাকশনটি বস্ স্থাপন করা	152
একটি রেজাল্ট ফরম্যাট নিয়ে আসা	153
কনস্ট্রাকশনটি শিরু ও সহ অ্যাপ স্টেরিও করা	154
সাধারণ ডাটাবেজ শিরু করা	154
অনুশীলন সমূহ	154
অন্য অ্যাপে সাধারণ ডাটাবেজ সন্ড করা	154
টেক্সট কনস্ট্রাকশনটি সন্ড করা	158
বাইনারি কনস্ট্রাকশনটি সন্ড করা	160
কনস্ট্রাকশনটি এর মাঝে টপল অংশ সন্ড করা	161
অন্য অ্যাপ থেকে সাধারণ ডাটাবেজ রিসিভ করা	161
আপনার মেনিফেস্ট আপডেট করা	161
ইনকামিং কনস্ট্রাকশনটি বস্ স্থাপন করা	164
একটি সহজ শিরু একশন যুক্ত করা	164
মেনু ডিউক্লয় এর শিরু আপডেট করা	164
শিরু ইনস্ট্রাকশনটি স্ট করা	167
ফাইল শিরু করা	168
অনুশীলন সমূহ	168
ফাইল শিরু করতে অ্যাপ স্ট আপ করা	168
ফাইল পর্ভাইডার (পর্দা নকারী) নিদিষ্ট স্ট করা	171
শিরু যোগ্য ডিউক্লয় টি নিদিষ্ট স্ট করা	172
একটি ফাইল শিরু করা	172
ফাইল রিকোয়েস্ট রিসিভ করা	172
ফাইল সিলেকশন একটি টিভি টেবিল তৈরি করা	172
ফাইল সিলেকশন একটি টিভি টেবিল ডিউক্লয় করা	176
ফাইল সিলেকশনকে রেসপন্স করা	177
ফাইলের জন্য পারমিশন (অনুমতি) পর্দা	177
রিকোয়েস্ট করা অ্যাপের সাথে ফাইল শিরু	180
একটি শিরু করা ফাইল রিকোয়েস্ট করা	181
ফাইলের জন্য একটি রিকোয়েস্ট সন্ড (পাঠানো) করা	181
রিকোয়েস্ট করা ফাইল পর্ভেবশ	183
ফাইল তথ্য উদ্ধার	183
একটি ফাইলের MIME টাইপ উদ্ধার	183
একটি ফাইলের নাম ও সাইজ উদ্ধার	186
এনএফসি দিয়ে ফাইল শিরু করা	187
অনুশীলন সমূহ	187
মেনিফেস্ট স্ট্রিফচারগুলা (বিশেষ স্ট্রিফচারগুলা) ডিউক্লয় করা	187
পারমিশন (অনুমতি) রিকোয়েস্ট (অনুরোধ) করা	187
NFC বিশেষ স্ট্রিফচারগুলা (বিশেষ স্ট্রিফচারগুলা) নিদিষ্ট স্ট করা	187
অ্যান্ডার ডিউক্লয় ফাইল টরান্সফার নিদিষ্ট স্ট করা	187

অয্য্ামন্ডরংয়েডবিমফাইলটরন্‌ামন্সফারসােপােটিরর্জনয়্	193
ঐক্কীর্টিক্ষর্ষ্যাকমেখডতৈত্রীকরুনযাফাইলসরবরাহকের	193
েসন্ডকরতেফাইলিনিদির্ষ্‌ষ্টকরুন	193
অনয়্‌ডিভাইসথেকেফাইলিরিসভকরা	193
ডাটাপরদশর্নকরতেএকিটিরিকোেয়েস্টেরপর্তিত্রেসপন্সকরা	198
ফাইলপারিমিশনিরিকোেয়েস্ট	199
কপি কৃত ফাইলের জনয়্‌ ডিরেক্‌টরী পাওয়া	200
একটি ফাইল ইউআরআই থেকে ডিরেক্‌টরী লাভ	201
একটি কনটেইন্‌ ট ইউআরআই থেকে ডিরেক্‌টরী লাভ করা	201
কনটেইন্‌ ট পর্‌ ভাইডার নিন্নর্‌য় করা	203
মাণিল্‌ টিমিডিয়া সহ অয্য্াপ তৈত্রী	205
অডিও পে‌ লবয্য্‌াক বয্য্‌বস্‌ থাপনা	205
অনু শীলনীসমূহ	205
আপনার অয্য্‌াে পর ভলিউম এব‌ ং পে‌ লবয্য্‌াক নিয়ন্‌ তর্‌গ	205
েকান অডিও সিস্‌টর্‌ম বয্য্‌বহার করতে হবে তা চিহ্ন নত করা	205
আপনার অয্য্‌াে পর অডিও ভলিউম নিয়ন্‌ তর্‌গ করতে হাডর্‌ ওয়ার ভলিউম িক বয্য্‌বহার করুন	205
আপনার অয্য্‌াে পর অডিও পে‌ লবয্য্‌াক নিয়ন্‌ তর্‌গ করতে হাডর্‌ ওয়ার পে‌ লবয্য্‌াক নিয়ন্‌ তর্‌গ িক বয্য্‌বহার করুন	211
অডিও ফেকাস বয্য্‌বস্‌ থাপনা	211
অডিও ফেকাস রিকোেয়েস্ট	214
অডিও ফেকাে সর ক্‌ষতি/হারিয়ে ফেলা কে বয্য্‌বস্‌ থপনা করা	216
Duck! / ডাক!	216
অডিও আউটপুট হাডর্‌ ওয়ার বয্য্‌বস্‌ থা করা	216
েচক করুন কোন হাডর্‌ ওয়ার বয্য্‌বহৃ ত হচ্ছে	216
অডিও আউটপুট হাডর্‌ ওয়ার এর পরিবতর্‌ন সামলােনা	216
ফেটা কয্য্‌াপচার করা	216
অনু শীলনীসমূহ	216
ছবি নেয়া	216
কয্য্‌ােমরা পারিমিশনিরিকোেয়েস্ট করা	216
কয্য্‌ােমরা অয্য্‌াপ দি়ে ফেটা তুলুন	216
ফেটা ভিউ করা	216
ফেটা সেভ করা	216
ফাইল নেম সেট করুন	216
ইনেটেনে‌ টর উপর ফাইল নেম যুক্ত করুন	216
একটি গয্য্‌ালারিতে ফেটা সােযাজন করুন	216
একটি সেস্‌কলড ইমেজ ডিকেড করা	216
িভিডিও রেকডর্‌ করা	216
কয্য্‌ােমরা পারিমিশনিরিকোেয়েস্ট	216
কয্য্‌ােমরা অয্য্‌াপ দি়ে ভিভিডিও রেকডর্‌ করা	216
িভিডিও ভিউ	216
কয্য্‌ােমরা নিয়ন্‌ তর্‌গ করা	216
কয্য্‌ােমরা অবজেক্‌ ট ওপেন করা	216
কয্য্‌ােমরা পির্‌ ভিউ তৈত্রী করা	216
পির্‌ ভিউ ক্লাস	216
পির্‌ ভিউ সেট করুন এব‌ ং শুরুর করা	216
কয্য্‌ােমরা সেটি ং পরিবতর্‌ন করা	216
পির্‌ ভিউ ওরিয়েনে‌ টশন সেট করুন	216
একটি ছবি নেয়া	216
রিভিউ প্লগ রা় শুরুর করিনি (রিস্‌টাটর্‌)	216
পির্‌ ভিউ বন্‌ ধ করুন এব‌ ং কয্য্‌ােমরা রিলিজ (েছেড়ে দিয়া) করুন	245
কনটেইন্‌ ট পির্‌ ন্‌ ট করা	246
অনু শীলনীসমূহ	246
ফেটা পির্‌ ন্‌ ট	246
ইমেজ পির্‌ ন্‌ ট	249
HTML ডেক্‌ মেন্‌ ট পির্‌ ন্‌ ট	250
HTML ডেক্‌ মেন্‌ ট লোড করুন	251

কীপার্স্ট্রিউজবৈতরীকরণ	253
পার্স্ট্রিময়ানেজারেরসাথেযুক্তকরুন	255
একটিপার্স্ট্রিঅ্যাডাপটারবৈতরীকরুন	256
পার্স্ট্রিডকুমেন্টিইনফোক্যাংপচারকরুন	256
একটিপার্স্ট্রিডকুমেন্টিফাইলেলেখা(রাইট)	259
PDFপেজকন্টেন্টডর করা	259
গরুাফিক্সএবংঅ্যাংগিনেমেশনদিয়েঅ্যাংপসবৈতরী	262
দক্‌ষতারসাথেবিটম্যাংপপার্দশরন	262
অনুশীলনীসমূহ	264
বড়আকারেরবিটম্যাংপদক্‌ষতারসাথেলেডকরা	265
বিটম্যাংপেরমাতর(ডাইমেনশন)এবংটাইপপড়ুন	265
মেমরীতেএকটিটেস্টকলডাউন-সংস্করণলেডকরুন	267
বিটম্যাংপপার্কিরিয়াইউআইথেরডবন্ধকরে	267
Async Task ব্যবহারকরুন	270
কনকারেনিন্সচালনাকরা	271
বিটম্যাংপজিমিয়েরাখা(ক্যাংগি)	271
একটিমেমরীক্যাংগব্যবহার	274
ডিস্কক্যাংগব্যবহার	277
কনফিগারেশনপরিবর্তনব্যবস্থাপনা	277
বিটম্যাংপমেমরীব্যবস্থাপনা	277
অ্যাংগনডরয়েড ২.২.৩ এবংএরচেয়েনীচেরসংস্করণেমেমরীব্যবস্থাপনা	282
অ্যাংগনডরয়েড ৩.০ এবংএরচেয়েউপরেরসংস্করণেমেমরীব্যবস্থাপনা	282
পরিবর্তরীব্যবহারেরজন্যএকটিবিটম্যাংপসেভকরা	282
একটিবিটম্যাংপবিটম্যাংপব্যবহার	286
আপনারইউআইএকটিবিটম্যাংপপার্দশরন	286
ভিউপেজবাস্তবায়নেবিটম্যাংপলেডকরা	290
একটিগিগরুডিভিউবাস্তবায়নেবিটম্যাংপলেডকরা	293
OpenGL ES দিয়েগরুাফিক্সপার্দশরন	293
অনুশীলনীসমূহ	299
একটিOpenGL ES পরিবেশবৈতরীকরা	299
মেনিফেস্টOpenGL ES ব্যবহারডিউক্লয়ারকরুন	299
OpenGL ES গরুাফিক্সএরজন্যএকটিএকটিভিউবৈতরীকরুন	299
একটিGLSurfaceView অবজেক্টবৈতরীকরুন	303
রেন্ডারক্লাসবৈতরীকরা	303
গঠন/আকৃতিনির্ধারণকরা	303
একটিতিরুভুজ(টিরুয়াংগল)নির্ধারণ	306
একটিবগরুকেষতর(স্কয়ার)নির্ধারণ	306
শেপ/আকৃতিঅঙ্কনকরা	306
শেপ/আকৃতিআরম্ভকরা	306
শেপঅঙ্কন/ডর করা	311
পারুজেকশনএবংক্যাংগেরাভিউপারুয়াংগ	314
একটিপারুজেকশননির্ধারণ	315
একটিক্যাংগেরাভিউনির্ধারণ	315
পারুজেকশনএবংক্যাংগেরাটিরুান্‌সফেরশনপারুয়াংগ	317
গতিশীলতা(মেশন)যুক্তকরা	318
শেপেরাটটকরা	318
অবিরামরেন্ডারিংসিকরুয়করা	318
টাচইভেন্‌টেরেসপন্‌সকরা	318
টাচলিসেনারসেটআপ	322
েরাটশনঅ্যাংগলউনেমাচনকরুন	322
েরাটশনপারুয়াংগ	324
অ্যাংগিনেমেশনসংযোজনকরা	325
অনুশীলনীসমূহ	325
দুইটাভিউকেকরুসেফডকরা	325

জিভ্রুউপ্‌লিমেইশন বৈসিটআপ করণ	325
ভিভউ কর্‌সফেড	330
িস্‌কর্‌ন স্‌লাইড এর জনয্‌ ভিভউ পেজ বয্‌বহার	331
ভিভউ তৈরী করুন	331
ফর্‌গে মেন্‌ট তৈরী করুন	331
ভিভউ পেজ যুক্ত করণ	334
পেজ টর্‌ানন্‌ সফরমার দি দেয় অয্‌ ানিমেশন কাস্‌টমাইজ করণ	336
পেজ টর্‌ানন্‌ সফরমার জুম আউট করণ	337
ডেপথ পেজ টর্‌ানন্‌ সফরমার	338
কাডর্‌ ফিফ্‌লপ অয্‌ ানিমেশন পর্‌দশর্‌ন করণ	339
অয্‌ ানিমেশন টৈরী করুন	339
card_flip_left_in.xml	339
card_flip_left_out.xml	339
card_flip_right_in.xml	339
card_flip_right_out.xml	339
ভিভউ তৈরী করুন	339
ফর্‌গে মেন্‌ট তৈরী করণ	346
কাডর্‌ ফিফ্‌লপ অয্‌ ানিমেশন টৈ করণ	347
ভিভউ জুম করণ	347
ভিভউ তৈরী করণ	350
জুম অয্‌ ানিমেশন সেটআপ করণ	351
ভিভউ জুম করুন	352
লেআউট পিরিবতর্‌ন অয্‌ ানিমেশন টৈ করণ	352
একটি টৈলেআউট তৈরী করণ	352
লেআউট টৈথেকে আইটেম এয্‌ াড (যুক্ত), আপডেট বারিমুয্‌ভ (বিবেচনা জন)	352
কানেকটি টি ভিভিট এব্‌ ক্‌ল আউড দি দেয় অয্‌ াপস তৈরী	358
ডিভাইসে ওয়ারলেস সন্‌যোগ	358
অনুশীলনী সমুহ	358
নেটওয়ার্কর্‌ সাঁভির্‌স ডৈলভারি বয্‌বহার	358
নেটওয়ার্কর্‌ আপনার সাঁভির্‌স রেজিট্রেশন করণ	363
নেটওয়ার্কর্‌ সাঁভির্‌স উইন্‌মাচন করণ	366
নেটওয়ার্কর্‌ সাঁভির্‌সে যুক্ত করুন	366
অয্‌ াপিলেকশন বন্‌ধ করণের সময় আপনার সাঁভির্‌স আনরেজিস্ট্রার করুন	369
ওয়াই-ফাই সহকারে P2P কানেকশন তৈরী করণ	370
অয্‌ াপিলেকশন পারমিশন সেটআপ করণ	370
একটি বর্‌ডকাস্ট রিসিভার এব্‌ ওয়াই-ফাই পিয়ার টু পিয়ার (P2P)	372
ময্‌ ানেজার সেটআপ করুন	372
পিয়ার ডিসকভারি শুরূ করুন	372
পিয়ার লিস্ট নিম্নে আসণ	375
একটি পিয়ারের সাথে সন্‌যোগ	376
সাঁভির্‌স ডিসকভারি র জনয্‌ ওয়াই-ফাই P2P বয্‌বহার করণ	376
মিনিফেস্‌টে সেটআপ করুন	376
লোকাল সাঁভির্‌স যুক্ত করুন	380
নিকটস্থ সাঁভির্‌স উদয্‌টন (ডিসকভার)	381
নেটওয়ার্কর্‌ অপারেশন সম্‌পাদন করণ	381
অনুশীলনী সমুহ	381
নেটওয়ার্কর্‌ যুক্ত হওয়া	381
একটি HTTP ক্‌ল ায়েন্‌ট নিবর্‌াচন	381
নেটওয়ার্কর্‌ কানেকশন চেক করুন	381
একটি পৃথক থের্‌ডে নেটওয়ার্কর্‌ অপারেশন সম্‌পাদন করুন	389
ডাটাসন্‌যোগ এব্‌ ডাউনলেড	389
ইনপুটটি স্‌টর্‌মকে একটি টি স্‌টর্‌ এ পিরিবতর্‌ন (কনভার্টর্‌)	393
নেটওয়ার্কর্‌ বয্‌বহার বয্‌বস্‌থাপনা	393
একটি ডিভাইসে সরনেটওয়ার্কর্‌ কানেকশন চেক করণ	395
নেটওয়ার্কর্‌ বয্‌বহার বয্‌বস্‌থাপনা	397
একটি পির্‌ফারেন্স একটি টি ভিভিট বাস্‌তবায়ন	397

পরিবর্তনসম্পন্ন পরিবর্তন নৈসর্গিক করা	400
কোড কন্ট্রোল পরিবর্তন নৈসর্গিক করা	402
XML ডাটাপারিসংকরণ	402
একটি পাসওয়ার্ড নিবন্ধন করা	402
ফিড বিবেচনা	402
পাসওয়ার্ড ইনস্টলেশন সফটওয়্যার করা	406
ফিড পাঠ করা	406
XML পাসওয়ার্ড করা	408
আপনার কাছে গুরুত্বপূর্ণ নথি এমন ট্যাগ গুলো এডিট করে (সংকল্প)	410
XML ডাটাব্যবহার	412
ব্যাটারি শেখানো করেই ডাটাত্রান্সফার	415
অনুশীলনীসমূহ	415
কায়রুর ক্রীড়া নেটওয়ার্কিং পরবেশ করার জন্য ডাউনলোড	415
অপটিমাইজ করা	
রেডিও স্টেশন	418
অ্যাপ কীভাবে রেডিও স্টেশন পরেভর্তিত করে	418
ডাটাপরিবর্তন (পরিধান মেমরী থেকে অস্থায়ী স্টোরেজ পরবর্তী ব্যাবহারের জন্য) করা	421
ব্যাট্রান্সফার এবং সংযোগ	423
কোড কন্ট্রোল (সংযোগ) কমিয়ে আনা	423
চিন্তার জায়গা চিহ্নিত করতে DDMS নেটওয়ার্কিং ট্রাফিক টুল ব্যবহার করে	426
নিয়মিত আপডেট পরেভর্তিত কমানো	426
পোলিং এ বিকল্প পিছনে গুল ক্লাউড মেসেজিং ব্যবহার	426
ইন এক স্যাক্সট্রিপিটিং এয়ালামর এবং এক সপোনেনন্ট ব্যাকঅফি দিয়ে পোলিং অপটিমাইজ করা	430
রিডানমেডন্ট ডাউনলোড অনাবশ্যক	430
স্থানীয়ভাবে ফাইল ক্যাশ (Cache) করা	433
HttpURL কোড কন্ট্রোল রেসপন্স ক্যাশিং ব্যবহার করুন	433
কোড কন্ট্রোল টিউনিং উপরিত্ত করে ডাউনলোডের ধরন পরিবর্তন করুন	433
ওয়াই-ফাই ব্যবহার	433
আরও বেশী বেশী ডাউনলোড করতে ব্যান্ডউইথ ব্যবহার করুন	437
ক্লাউডে সিঙ্ক (Syncing) করা	437
অনুশীলনী	437
ব্যাকআপ এপিআই ব্যবহার	437
অ্যান্ডরয়েড ব্যাকআপ সার্ভিসের জন্য রেজিস্টার	437
আপনার মেনিফেস্ট কমিফগার করুন	437
আপনার ব্যাকআপ এজেন্ট লিখুন (রাইট)	443
ব্যাকআপ রিকোয়েস্ট	443
একটি ব্যাকআপ থেকে রিস্টোর করুন	443
অধিকাংশ গুল ক্লাউড মেসেজিং তৈরী করা	443
কায়রুর ক্রীড়া মাল্টিটাস্কিং মেসেজ সেন্ড (পাঠানো) করুন	448
মেসেজটি পরিত্যক্ত স্থাপিত হতে পারে সেটা বন্ধ করে দিন	448
সরাসরি GCM মেসেজের মধ্যে ডাটাবসানো	451
GCM মেসেজের পরিত Intelligently রিয়েক্ট করা	451
বিবর্তিত করে বনানো	451
স্মার্ট র্ভাবে সিঙ্ক করুন কঠিনভাবে নয়	451
ক্লাউডে সেভ করা সম্প্রতিকর্ত সাংঘর্ষিক বিষয়ের	455
সমাধান করা	
কন্ফ্লিক্ট সম্প্রতিকর্ত অবগত হতে	456
সাধারণ কেস গুলো পরিচালিত করা	457
আরও জটিল কেসের জন্য একটি কীল ডিভাইন করা	457
পার্থক্য পরে চেষ্টা: শুধু সবার মেসেজটি (টেস্টালা) স্টোর করুন	459
দিবর্তী পরে চেষ্টা: সবার মেসেজ (টেস্টালা) এবং ডেল্টা স্টোর করুন	460
সমাধান: ডিভাইন সপারিত সাব-টেস্টালা স্টোর করুন	462
আপনার ডাটাপরিবর্তন করার করুন	462

সিঙ্ক অ্যাডাপটর ব্যবহার করে ডাটাত্রান্সফার অনুশীলনীসমূহ	465
একটি স্টাব-অথেনটিকেটর (পরিমাণকারী) তৈরী করা	468
একটি স্টাব অথেনটিকেটর উপাদান যুক্ত করুন	469
ফর্মওয়ার্কের অথেনটিকেটর সংযুক্ত করা	471
অথেনটিকেটর মেটাডেটা ফাইল যুক্ত করা	472
মেনিফেস্টের মধ্যে অথেনটিকেটর ডিক্লেয়ার করা	472
একটি স্টাব কন্টেন্ট পর্ভাইডার তৈরী করা	475
একটি স্টাব কন্টেন্ট পর্ভাইডার যুক্ত করা	476
মেনিফেস্টের মধ্যে পর্ভাইডার ডিক্লেয়ার করা	476
একটি সিঙ্ক অ্যাডাপটর তৈরী করা	479
বেজিসিক অ্যাডাপটর ক্লাস AbstractThreadedSyncAdapter পরিসারিত করুন	480
onPerformSync() এ ডাটাত্রান্সফারকে ডায়ালেক্ট করুন	481
ফর্মওয়ার্কের সিঙ্ক অ্যাডাপটর একিত্ব করা	484
ফর্মওয়ার্কের কতৃক চাওয়া একাউন্ট যুক্ত করুন	486
সিঙ্ক অ্যাডাপটর মেটাডেটা ফাইল যুক্ত করুন	486
মেনিফেস্টের মধ্যে সিঙ্ক অ্যাডাপটর ডিক্লেয়ার করা	489
একটি সিঙ্ক অ্যাডাপটর রান করা	491
একটি সিঙ্ক অ্যাডাপটর রান করা যখন সার্ভার ডাটাপরিবর্তন করে	493
সিঙ্ক অ্যাডাপটর রান করা যখন কন্টেন্ট পর্ভাইডার ডাটাপরিবর্তন করে	495
একটি নেটওয়ার্ক মেসেজের পরিসিঙ্ক অ্যাডাপটর রান করা	495
পয়ারায়কর মেসিঙ্ক অ্যাডাপটর রান করা যখন	498
চাহিদা অনুযায়ী সিঙ্ক অ্যাডাপটর রান করা	500
ইউজার ইনফো এবং লোকেশন (অবস্থান) সহকারে অ্যাডাপটর তৈরী করা	502