LẬP TRÌNH TRÊN ANDROID

Part 1

Làm việc với XML trên Android

Xây dựng các ứng dụng Java cho các thiết bị di động

Michael Galpin, Kiến trúc sư phần mềm, Ludi Labs

Tóm tắt: Android là một hệ điều hành nguồn mở, hiện đại và là SDK cho các thiết bị di động. Với hệ điều hành này, bạn có thể tạo ra các ứng dụng di động rất mạnh. Điều này thậm chí còn trở nên hấp dẫn hơn nữa khi các ứng dụng của bạn có thể truy cập các dịch vụ Web, có nghĩa là bạn cần sử dụng ngôn ngữ của Web là: XML. Trong bài viết này, bạn sẽ thấy nhiều lựa chọn khác nhau để làm việc với XML trên Android và cách sử dụng chúng để xây dựng các ứng dụng Android của chính bạn.

Trong bài viết này, bạn học cách xây dựng các ứng dụng Android có thể làm việc với XML từ Internet. Các ứng dụng Android được viết bằng ngôn ngữ lập trình Java™, do vậy mà kinh nghiệm làm việc với công nghệ Java là điều cần phải có. Để phát triển cho Android, bạn sẽ cần đến Android SDK. Toàn bộ mã trình được trình bày trong bài viết này sẽ làm việc với bất kỳ phiên bản nào của Android SDK, nhưng phiên bản SDK 1.5_pre đã được sử dụng để phát triển mã trình. Bạn có thể phát triển các ứng dụng Android chỉ với SDK và một trình biên tập văn bản là đủ, nhưng sẽ dễ dàng hơn nhiều nếu sử dụng Android Developer Tools (ADT), là một trình bổ sung Eclipse. Đối với bài viết này, phiên bản 0.9 của ADT đã được dùng với Eclipse 3.4.2, một phiên bản Java. Xem Tài nguyên để lấy các liên kết dẫn đến tất cả các cộng cụ này.

XML trên Android

Nền tảng Android là một nền tảng phát triển di động mã nguồn mở. Nó giúp bạn truy cập vào tất cả các khía cạnh của thiết bị di động mà nó chạy trên đó, từ các đồ họa cấp thấp, đến phần cứng như là thiết bị camera trên điện thoại. Với rất nhiều thứ có thể sử dụng Android, có thể bạn sẽ tự hỏi tại sao bạn cần phiền đến XML. Đó không phải vì làm việc với XML rất thú vị; mà là nó đang làm việc với những

thứ mà nó kích hoạt. XML thường được dùng như là một định dạng dữ liệu trên Internet. Nếu bạn muốn truy cập dữ liệu từ Internet, các khả năng có thể là dữ liệu sẽ ở dạng XML. Nếu bạn muốn gửi dữ liệu đến một dịch vụ Web, có thể bạn cũng cần gửi cả dữ liệu XML. Nói ngắn gọn là nếu ứng dụng Android của bạn thúc đẩy Internet, thì có thể bạn sẽ cần phải làm việc với XML. Thật may mắn là bạn có rất nhiều lựa chọn có sẵn để làm việc với XML trên Android.

Các trình phân tích XML

Các từ viết tắt thông dụng

- API: Application programming interface (Giao diện lập trình ứng dụng)
- RSS: Really Simple Syndication (Giao thức tập hợp thông tin đơn giản)
- SDK: Software Developers Kit (Bộ dụng cụ cho nhà phát triển phầm mềm)
- UI: User interface (Giao diện người dùng)
- URL: Universal Resource Locator (Địa chỉ tài nguyên)
- XML: Extensible Markup Language (Ngôn ngữ đánh dấu mở rộng)

Một trong những ưu điểm lớn nhất của nền tảng Android chính là việc nó thúc đẩy ngôn ngữ lập trình Java. Android SDK không hoàn toàn cung cấp sẵn mọi thứ cho Môi trường Thời gian chạy Java (JRE) chuẩn của bạn, nhưng nó lại hỗ trợ một phần rất đáng kể cho nó. Nền tảng Java đã và đang hỗ trợ rất nhiều cách khác nhau để làm việc với XML trong thời gian nhất định, và hầu hết các API có liên quan đến XML của Java đều được hỗ trợ đầy đủ trên Android. Ví dụ, Simple API của Java cho XML (SAX) và Document Object Model (DOM) hiện đều có sẵn trên Android. Nhiều năm qua, cả hai API này là một phần của công nghệ Java. Sản phẩm Streaming API mới đây cho XML (StAX) hiện chưa có trong Android. Tuy nhiên, Android lại cung cấp một thư viện tương đương về mặt chức năng. Điều cuối cùng là Java XML Binding API cũng không có sẵn trong Android. Chắc chắn có thể thực hiện API này trong Android. Tuy nhiên, nó lại có xu hướng là một API nặng ký, với rất nhiều thể hiện khác nhau thuộc các lớp khác nhau thường cần việc trình bày một tài liệu XML. Do vậy mà nó không lý tưởng lắm cho một môi trường bị ràng buộc chẳng hạn như thiết bị cầm tay mà Android được thiết kế để chạy trên đó. Trong các phần tiếp theo, bạn sẽ lấy một nguồn XML đơn giản có sẵn trên Internet, và xem cách phân tích nguồn đó như thế nào trong phạm vi một ứng dụng Android sử dụng các API khác nhau được nhắc đến ở trên.Trước tiên, hãy xem các phần cần thiết của ứng dụng đơn giản sẽ sử dụng XML từ Internet.

Trình đọc tin Android

Ứng dụng sẽ lấy điểm tin RSS từ trang nhà phát triển Android phổ biến Androidster và phân tách nó thành một danh sách các đối tượng Java đơn giản mà bạn có thể sử dụng để quay lại Android ListView (xem Tải về để lấy mã nguồn). Đây là hoạt động đa hình thái cổ điển — tức là các thực thi khác nhau (các thuật toán phân tích XML khác nhau) cung cấp hoạt động giống nhau. Ví dụ 1 cho bạn thấy bạn có thể mô hình hóa điều này dễ dàng như thế nào trong mã trình Java sử dụng một giao diện.

Ví dụ 1. giao diện trình phân tích điểm tin XML

```
package org.developerworks.android;
import java.util.List;

public interface FeedParser {
    List<Message> parse();
}
```

Trong Ví dụ 2, lớp `Message` là một POJO (Plain Old Java Object) cổ điển miêu tả một cấu trúc dữ liệu.

Ví dụ 2. `Message` POJO

```
public class Message implements Comparable<Message>{
    static SimpleDateFormat FORMATTER =
        new SimpleDateFormat("EEE, dd MMM yyyy HH:mm:ss
Z");
    private String title;
    private URL link;
    private String description;
    private Date date;

      // getters and setters omitted for brevity
    public void setLink(String link) {
        try {
            this.link = new URL(link);
```

```
    } catch (MalformedURLException e) {
        throw new RuntimeException(e);
    }
}

public String getDate() {
    return FORMATTER.format(this.date);
}

public void setDate(String date) {
    // pad the date if necessary
    while (!date.endsWith("00")){
        date += "0";
    }
    try {
        this.date = FORMATTER.parse(date.trim());
    } catch (ParseException e) {
        throw new RuntimeException(e);
    }
}

@Override
public String toString() {
         // omitted for brevity
}

@Override
public int hashCode() {
        // omitted for brevity
}

@Override
public boolean equals(Object obj) {
        // omitted for brevity
}
  // sort by date
public int compareTo(Message another) {
    if (another == null) return 1;
    // sort descending, most recent first
    return another.date.compareTo(date);
```

```
        }
}
```

Message, trong Ví dụ 2, thường rất dễ làm. Nó ẩn đi một vài trạng thái bên trong của mình bằng cách cho phép truy cập ngày tháng và các liên kết như các chuỗi đơn giản, trong khi thể hiện chúng như các đối tượng được sắp xếp một cách rõ ràng (một `java.util.Date` và một `java.net.URL`). Nó là một Value Object (Đối tượng Giá trị) cổ điển, do vậy nó thực thi `equals()` và `hashCode()` dựa trên trạng thái bên trong của nó. Nó cũng thực hiện giao diện `Comparable` vì thế bạn có thể sử dụng nó để sắp xếp (theo ngày tháng). Thực tế, dữ liệu được phân loại từ điểm tin, do vậy mà điều này không cần thiết.

Mỗi thực thi trình phân tích sẽ cần đưa một URL đến điểm tin Androidster và sử dụng cái này để mở một kết nối HTTP đến trang Androidster. Hoạt động phổ biến này được mô hình hóa một cách tự nhiên trong mã trình Java sử dụng lớp cơ sở trừu tượng như trong Ví dụ 3.