Mobile Design Pattern: Live Scrolling

Description

The **Live Scrolling** Design Pattern, also known as **Continuous Scrolling** or **Infinite Scrolling**, is a way of displaying and loading data in incremental manner, alternative to **Pagination** (Mobile Design Pattern: Paging).

With Live Scrolling, **a first subset of data is initially loaded and shown** to the user. Then, **when the user reaches the bottom** of this data, **a new set of data is automatically loaded and displayed**. So, there's **no need of explicit user interaction** to load the next available items, contrary to **Pagination**, where the user has to click on a button or link to go to another page.

When **new items** are loaded, they're **appended to the items already displayed** to the user, so giving him the impression of a never-ending page, contrary to **Pagination**, where only a subset of data is always visible to the user.

An example of **Live Scrolling** is visible in the Gmail mobile client:

Advantages

- **User interaction is more intuitive and fast**, since the user does not need to press any buttons to see the next available items
- **While a new set of data is loading, the user can continue to view and browse** the data already displayed, so improving the overall user experience
- **All loaded data is visible** to the user, not only the last loaded one

Disadvantages

- User has **typically no explicit information about the total number of items**
- User has **no way to go directly to a specific set of data**, but has to load all the items until the desired one

Use when

The Live Scrolling Design Pattern is useful when there is the need of loading and displaying large sets of data.

Use how

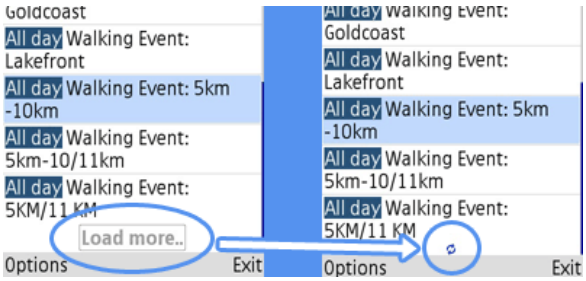
The typical implementation of this Design Pattern is structured as follows:

- a **first set of data is loaded**
- the user can freely scroll and view the loaded data
- when the **user reaches the bottom of this data**, **new data is automatically loaded** in background
- as soon as the new data is available, it is **appended to the existing one**

While a new set of data is loading in background, the user is free to continue reading and scrolling the already available items: so, it is important that the user interface remains usable

Alternative implementations use a **button or link**, placed at the bottom of the available data, to **allow the user to explicitly ask for new items**.

(C) Copyright Nokia 2013. All rights reserved.



This approach gives to the user more control over the loading of data, but loses the intuitiveness of the automatic loading of data, typical of **Live Scrolling**. An example of this alternative approach is visible in the Facebook website:

Design Tips

- **Load relatively small sets of data** for each step: this will increase loading speed, and give more responsiveness to applications
- Give the user a **clear indication when new data is loading**, to avoid the impression of blocked user interface
- If there is an **error while loading** the next set of data, it is **good practice to automatically retry to load it**, for a predefined number of times

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.