

Web DB와 Store를 사용한 HTML5 캘린더 구현

최 민

충북대학교 정보통신공학부
e-mail:mchoi@cbnu.ac.kr

HTML5 Calendar Implementation using Web DB and Store

Min Choi

Dept of Information and Communication Engineering,
Chungbuk National University

요 약

기존의 애플리케이션 개발방식은 각각의 스마트폰 플랫폼에 종속되어 있기 때문에 플랫폼별로 매번 따로 개발하여야 하므로 비효율적이다. 반면, HTML5 기반 스마트 애플리케이션은 사용자의 스마트 폰이 안드로이드/iOS 여부에 관계없이 동일하게 동작한다. 본 연구에서는 Web DB와 Store를 활용한 캘린더를 구현하였으며, HTML5 UI Framework인 Sencha Touch를 사용하여 HTML5 기반 하이브리드 애플리케이션으로 제작하였다. 특히, Web DB와 Store를 활용하여 날짜 데이터를 다루는 방법, 날짜 기반 검색방법을 소개한다.

1. 서론

최근스마트 폰 사용자들이 급증하면서 스마트 폰 안의 필수 아이템인 모바일 애플리케이션은 이제 새로운 통신 트렌드로 자리 잡았다. 획기적이고 실속 있는 다양한 애플리케이션이 실제로 생활에 많은 편의와 이득을 주고 있다. 네이티브 앱은 iOS나 안드로이드처럼 특정 플랫폼에서만 작동되는 애플리케이션이다. 웹 앱은 웹페이지를 작업할 때 사용하는 HTML/CSS 또는 JQuery등을 활용하여 작성된 브라우저에서 동작되는 화면이다. 그러므로 하이브리드 앱은 네이티브 앱과 웹 앱의 장, 단점을 적절히 활용하여 개발한 형태이다. 기존의 개발 방식은 각 플랫폼별로 개발자가 존재해야하고, 플랫폼별로 업데이트가 되는 시기도 다르다는 한계점이 있다. 그렇기 때문에 한 번의 업데이트를 하려고 할 때, 기업에서는 플랫폼별 비용이 소요되고, 개발자는 유지보수에 더 많은 노력을 쓰게 된다. 또 사용자는 각 플랫폼별로 지원하는 기능의 차이가 있을 수 있다. 따라서 기업이나 개발자, 사용자 모두 불편할 수밖에 없다. 하지만 하이브리드 앱 방식의 개발을 하게 되면 기업에서는 비용을 줄일 수 있고, 개발자는 유지보수에 들일 노력을 다른 부분에 들일 수 있다. 그리고 사용자들에게는 똑같은 수준의 기능을 보장받을 수 있다는 장점이 있다.

하이브리드 앱 개발 방식은 개발자에게는 유지보수의 편리와 사용자에게는 스마트 폰에 종속되지 않는 편의를 주는 면이 있지만, 하이브리드 앱 개발 방식에 대한 진입 장벽이 있다. 이것은 센차터치를 이용한 개발이 다른 방식보다 약간 더 높은 수준이다. 그 이유는 센차터치는 모든

기능을 제공하고 있는데 개발자가 자신의 원하는 것을 직접 찾아서 적용해야 하기 때문이다. 하지만 개발을 진행하면 할수록 자신이 원하는 기능을 어떻게 찾아서 사용하는지 알게 될 것이다. 또 다른 제한점은 아직은 기존의 개별 플랫폼에 맞춘 개발 방식에 비해서 구동속도가 느리다는 제한점이 있다. 이것은 아직도 하이브리드 앱 개발 방식의 기술이 진보하는 중이기 때문이다. 그렇기 때문에 본 논문에서는 애플리케이션의 구동속도에 관한 부분은 기술 진보에 따라 해결될 부분으로 남겨두고자 한다.

2. 관련연구

센차터치는 모바일 웹 애플리케이션 개발을 위한 자바스크립트 프레임 워크이다. 기존 데스크톱 기반 자바스크립트 프레임워크인 ExtJS 개발사에서 모바일 시대에 부응하고자 새롭게 내놓은 제품이라고 할 수 있다. 즉, 자바스크립트, CSS, HTML과 같은 순수 웹 기술만으로 모바일 페이지를 작성할 수 있게 해주는 것이다. 센차터치는 ExtJS를 기반으로 하고 있어서 그 문법 체계를 그대로 따르고 있으며, 여기에 모바일에 최적화된 기능들이 추가된 형태이다. 다른 프레임워크와 비교했을 때, 센차터치의 가장 큰 특징은 기존의 데스크톱의 문법 체계를 그대로 가져왔기 때문에 방대하고 체계적인 API가 있다는 것이다. 그 외에도 스크립트 기반, HTML5와 CSS 지원, 향상된 터치 이벤트, 휴대폰과 태블릿 기기를 위한 프레임 워크, 높은 자유도, 모바일에 최적화된 기능이 있다.

종류	특징
jQuery Touch	<ul style="list-style-type: none"> - 가장 최초의 프레임워크(현재 Sencha사에 흡수) - jQuery 플러그인 형태, Markup기반의 개발스타일 - 모바일에 최적화된 UI, Ajax통신, 애니메이션, 자동 네비게이션 제공
jQuery Mobile	<ul style="list-style-type: none"> - jQuery진영에서 직접 개발, 가장 많은 모바일 기기를 지원 - Markup기반의 개발스타일, 최신 웹 표준 기술(HTML5, CSS)적극 활용 - 모바일에 최적화된 UI, Ajax통신, 애니메이션, 자동 네비게이션, 각종 터치 이벤트 등 제공
Sencha Touch	<ul style="list-style-type: none"> - ExtJS진영에서 개발한 가장 방대하고 체계적인 API를 보유 - Script기반의 개발 스타일 - 객체지향기법, 고급스트립트 개념이 API에 적용되어 프로그래밍적임 - 타 프레임워크에 비해 자유로운 표현과 방대한 기능 활용 가능

표 1 jQuery와 Sencha Touch의 비교

그림 1은 센차터치의 패널 구조를 나타내고 있다. 센차터치에서는 패널에도 특정기능을 부여해서 다양한 지원을 한다. 그러므로 본 애플리케이션에서 구현하기를 원하는 기능은 각 패널의 API를 참조해서 구현할 수 있었다.

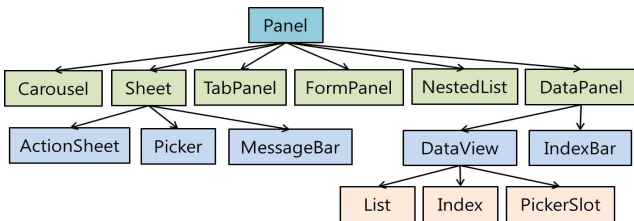


그림 1 센차터치 패널구조

폰갭은 하이브리드 애플리케이션을 개발할 수 있게 도와주는 프레임워크이다. 폰갭을 통해서 one-source multi-use가 가능해지는 것이다. 현재 폰갭 외에도 앱스프레소, 앱셀레이터 티타늄이 있지만 안드로이드와 iOS만을 지원하기 때문에 플랫폼의 제한이 있다.

아래의 그림 3.2를 통해서 자바스크립트로 작성된 소스코드에 폰갭을 사용해서 각 스마트 폰의 운영체제에 맞는 플랫폼에 적합하게 변형이 되는 것을 볼 수 있다. 소스코드에 폰갭을 사용하면 폰갭빌더가 각각의 운영체제에서 동작을 할 수 있도록 자바스크립트로 된 소스코드를 변형시킨다. 즉, 모든 플랫폼이 공통으로 적용되는 패키징 시스템이 있는 것이 아니라, 스마트 폰 플랫폼에 따라 폰갭 라이브러리를 개별적으로 패키징하는 구조이다. 따라서 개

발자가 작성한 하나의 소스코드로 다수의 사용자가 요구하는 환경에 맞게 사용할 수 있는 one-source multi-use가 가능해지는 것이다.

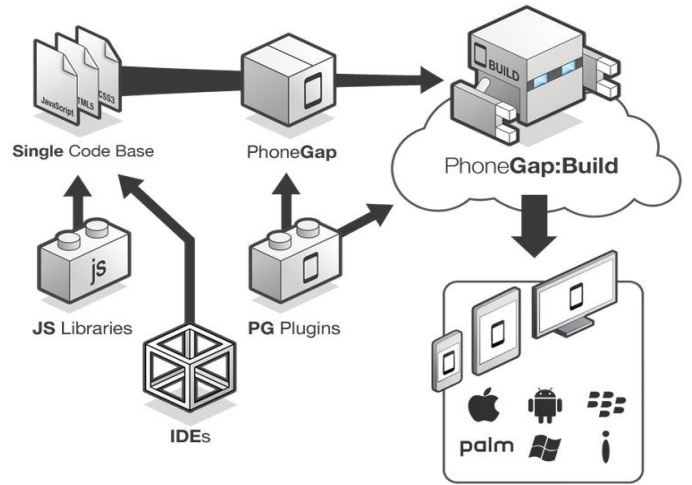


그림 2 폰갭 패키징 단계

그림 2는 HTML5 소스코드에 PhoneGap에서 제공하는 라이브러리와 플러그인을 활용하여 폰갭 빌드 단계를 거치면, 안드로이드/아이폰 등에서 활용가능한 바이너리 패키지를 얻을 수 있음을 나타낸다.

지원 API 종류	기능
Accelerometer	디바이스의 모션센서 처리
Camera	디바이스의 카메라 모듈을 이용한 사진 캡처 처리
Capture	오디오, 이미지, 비디오 캡처 기능 처리
Compass	디바이스 포인팅 방향 정보 처리
Connection	휴대폰과 와이파이 접속 정보 제공
Contacts	디바이스의 연락처 데이터베이스 접근 제어
Device	디바이스의 하드웨어 및 소프트웨어 정보 표현
Events	폰갭 라이프사이클 이벤트 처리
File	파일 읽기, 쓰기 기능 제어
Geolocation	GPS 정보 처리
Media	오디오 파일에 대한 녹음, 재생 기능 처리
Network	네트워크 접속 정보 제공
Notification	alert, confirm, beep, vibrate에 대한 알림 기능 제공
Storage	디바이스 스토리지 접근 처리

표 2 폰갭이 지원하는 디바이스 API

3. WebDB와 Store를 활용한 HTML5 캘린더

웹 SQL 데이터베이스는 SQL을 사용하여 관계형 데이터

베이스를 이용할 수 있도록 해주는 것이다. 그러므로 웹 SQL 데이터베이스를 적용한 브라우저에서는 자유롭게 데이터베이스를 작성하거나 이용할 수 있다. 일반적으로 서버와 통신을 하면서 데이터를 저장하는 웹 애플리케이션이 웹 SQL 데이터베이스를 이용하면, 로컬 데이터베이스에 먼저 저장을 해두고 나중에 서버와 통신을 하게 되었을 때 동기화를 한다. 오프라인일 때는 이런 점을 이용해서 웹 애플리케이션에 사용을 한다. 본 논문에서의 애플리케이션은 서버와 통신을 하지는 않는다. 하지만 오프라인일 때의 웹 애플리케이션과 같은 동작을 구현해야하므로 이 방법을 통해서 구현했다.



그림 3 Web DB 활용과정

트랜잭션이란 데이터베이스를 조작할 때, 관련된 여러 처리를 하나의 처리단위로 묶은 것이다. 이렇게 트랜잭션을 만드는 이유는 데이터의 무결성을 위한 것이다. 무결성이란 어떤 처리가 시작되고 완료되는 과정에서 예기치 못한 상황이 발생해서 완료가 되지 못했을 때, 진행 중이던 처리를 처음부터 다시 시작해서 데이터 처리의 신뢰성을 가지기 위한 것이다. 본 애플리케이션에서의 트랜잭션은 사용자가 선택한 프레임의 모양, 배치되는 이미지와 글로 볼 수 있다.

```
var DB=openDatabase(name, version, display DB name, estimated Size, callBack);
DB.transaction(callBack, errorCallback, successCallBack);
tx.executeSql(sql, args, callback, errorCallback);
```

그림 4 웹 SQL 데이터베이스를 사용하기 위한 단계별 함수

웹 SQL 데이터베이스를 사용하기 위한 함수를 아래 그림 3.5에 나타내었다. 첫 번째 함수를 사용해서 데이터베이스를 열게 된다. 데이터베이스를 열기 위해서는 생성되는 데이터베이스의 이름, 데이터베이스 버전, 보여지는 데이터베이스의 이름, 사용할 데이터베이스의 크기를 정해 주어야 한다. 그런 후에 열려진 데이터베이스의 트랜잭션을 지정한다. 트랜잭션을 정하는 함수에서는 첫 번째 인자만 지정하여도 무방하다. 트랜잭션을 지정하면 세 번째 함수를 이용해서 SQL문으로 데이터를 처리한다. 이 함수도 첫 번째 인자만을 사용하여도 무방하다.

4. 구현 및 실험

하이브리드 앱의 전체적인 구조는 웹 앱을 네이티브 형식으로 패키징한 형태이므로 먼저 모바일 웹 앱을 개발해야

한다. 따라서 웹 표준 기술을 활용하고 모바일에 최적화된 구현을 위해 모바일 UI 프레임워크를 사용했다. 본 연구에 사용할 프레임워크는 센차터치로 이 기술에 대한 연구를 통하여 네이티브 수준의 UI와 동작방식을 구현했다.

```
Ext.define('Schedule', {
    extend: "Ext.data.Model",
    config: {
        fields: [{name: 'event', type: 'string'},
                {name: 'location', type: 'string'},
                {name: 'start', type: 'date'},
                {name: 'weather', type: 'string'},
                {name: 'content', type: 'string'}]
    }
});
var calendar = Ext.create('Ext.ux.TouchCalendarView', {
    mode: 'month',
    weekStart: 0,
    value: new Date(),
    schedules: scheduleStore,
    diaries: diaryStore,
    plugins: [new Ext.ux.TouchCalendarEvents({eventBarTpl: '{event} - {location}'}),
             new Ext.ux.TouchCalendarSimpleEvents()],
    listeners: {
        'selectionchange': function(calendar, newDate, prevDate){
            scheduleStore.clearFilter();
            scheduleStore.filter('start', newDate);
            ViewPort.setActiveItem('DayView', {type: 'slide', direction: 'left'});
        },
        'eventtap': function(event){},
        'eventdragstart': function(draggable, eventRecord, e){},
        'beforeeventdrop': function(draggable, eventRecord, e){return true;},
        'eventdrop': function(draggable, eventRecord, e){},
        'eventdrag': function(draggable, eventRecord, e){}
    }
});
```

그림 5 날짜 검색을 구현한 소스코드

터치한 날짜에 해당하는 리스트를 보려면 센차터치가 제공하는 filter를 사용해서 해당하는 날짜를 필터링하면 된다. 아래의 그림 4.2에서 보듯이 필터링의 원리는 model의 fields의 name을 통해서 걸러내고자 하는 값을 찾아내는 것이다. 날짜를 터치하는 순간 동작하는 함수가 'selectionchange':function이다. 이 함수 안에서 데이터를 저장하고 있는 store에 filter를 사용해서, model의 fields가 가지는 name을 비교하고, 해당날짜와 일치하면 화면에 보이도록 하였다.

특히, 날짜를 검색하는 단계에서는 "Jan 12, 2013" 등의 형태로 들어오는 Date 변수에서 이를 다시 Store의 내용과 비교하는 경우, type mismatch로 잘 검색되지 않는 경우가 발생하였다. 이는 Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec가 Date 타입에서 String으로 변환되면서 Store에 검색하는 함수에서 이를 잘 인지하지 못하여 발생한다.

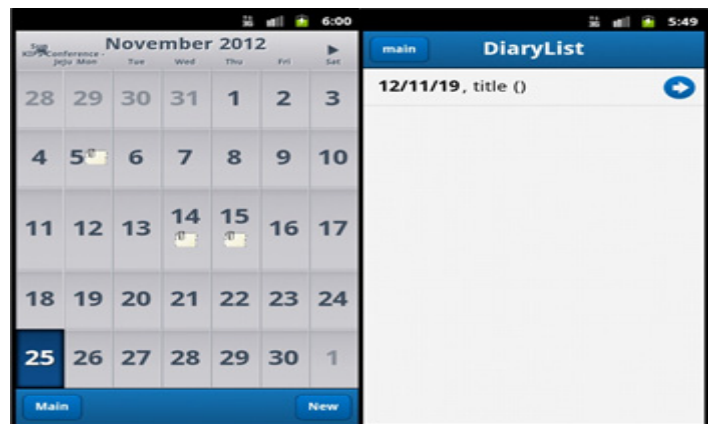


그림 6 Web DB와 Store를 활용한 캘린더 프로그램

따라서, 본 연구에서는 Date 타입의 내용을 String으로 변

환한 후에, Javascript 구문을 사용하여 이를 각각 파싱 (parsing)하여 날짜 데이터를 얻어낸 후, 이를 다시 Date 타입으로 만들어서 Store에 filter 검색함수의 인자로 제공하였다. 그림 6은 본 연구에서 개발한 Web DB와 Store를 활용한 캘린더 스마트 애플리케이션을 실행한 화면으로, 센차터치가 제공하는 API를 사용해서 구현하였다. 캘린더에 보이는 날짜들을 클릭할 수 있고 상단의 화살표를 터치해서 이전과 이후의 달력을 볼 수 있다. 사용자가 스케줄을 입력하면 화면에 메모 모양의 그림으로 표시되어 나타난다. 오른쪽의 리스트 화면은 사용자가 작성한 다이어리의 리스트를 보여준다. 리스트의 오른쪽에 보이는 화살표를 터치하면 작성한 다이어리를 보여주도록 되어있다.

5. 결 론

본 연구에서는 HTML5 기반 스마트폰 앱개발 방식을 사용함으로써 사용자의 스마트 폰이 안드로이드/iOS 여부에 관계없이 동일하게 동작하도록 하였다. 기존의 플랫폼별 개발방식에서는 각각의 스마트폰 플랫폼에 개발환경과 실행 바이너리가 모두 종속되어 있기 때문에 플랫폼별로 매번 따로 개발하여야 하므로 비효율적인 측면이 있었다. 본 연구에서 개발한 Web DB와 Store를 활용한 캘린더를 구현은 일반적으로 스마트폰 애플리케이션에서 많이 활용되는 캘린더를 HTML5 UI Framework인 Sencha Touch를 사용하여 HTML5 기반 하이브리드 애플리케이션으로 제작하였다. 특히, Web DB와 Store를 활용하여 날짜 데이터를 다루는 방법, 날짜 기반 검색방법을 독창적으로 제안하고, 이를 구현하였다.

참고문헌

- [1] <http://docs.sencha.com/touch/2-0-0/#!/api>
- [2]<http://obanadingyo.tistory.com/entry/9%EC%9D%BC%EC%B0%A8-Sencha-Touch-2-Floating-Components>
- [3] <http://coronasdk.tistory.com/160>
- [4]http://www.tutorialspoint.com/html5/html5_web_sql.htm
- [5]<http://try.sencha.com/touch/2.0.0/demos/Ext.data.Store.filters/>
- [6]<https://github.com/SwarmOnline/Ext.ux.TouchCalendar>
- [7]http://docs.phonegap.com/en/1.0.0/phonegap_camera_camera.md.html
- [8] <http://blog.daum.net/pcbookweb/1125>
- [9]<http://www.metaltoad.com/services/phonegap-application-development>
- [10]<http://blogs.technet.com/b/port25/archive/2011/12/19/full-support-for-phonegap-on-windows-phone-is-now-complete.aspx>