

## ADF로 iPhone 용 웹 페이지 개발하기

최근 들어 IT 업계의 가장 큰 화두는 Mobile이라는 단어가 아닐까 생각한다. Application Development Framework(ADF)는 10여 년 전 300% JAVA라는 슬로건을 내걸고 당시 최고의 JAVA IDE였던 볼랜드사의 JBuilder를 라이선싱해서 만든 JDeveloper 환경에서 오랜 발전을 거듭한 결과물이다. 10여 년 전 오라클은 기업용 소프트웨어만을 전문적으로 만드는 회사이기에 각종 업무 솔루션이 하나의 패키지로 되어 있는 e-Business Suite와 같은 덩치 큰 패키지 소프트웨어를 만들고 싶었다. 하지만 수많은 개발자들이 협업하면서 제품을 개발하기 위해서는 견고하고 효율적인 Framework이 있어야만 가능한 일이었다.

이런 이유로 오라클은 자체적인 자바 Framework 개발에 착수하였고, Business Component for Java(BC4J)라는 모델 레이어 컴포넌트의 개발부터 시작해서 controller 및 UI Component를 개발하기에 이르렀다. 과거의 XML기반 UI프로젝트 산출물인 UIX는 지금의 JSF 표준이 탄생하게 된 계기를 마련하였다고 볼 수 있다. 또한, 오라클이 개발한 XML기반의 UI component들은 오라클의 Open Source 지원 정책(<http://oss.oracle.com>)에 따라 Open Source 그룹에 기증되어 Apache Faces의 근간을 이루게 되었다.

이번 호에서는 ADF의 UI를 담당하고 있는 ADF Faces의 모바일 지원 사항을 두 개의 기사로 나누어 “(1) ADF 웹 애플리케이션”에서는 iPhone을 지원하는 UI구현 중심으로, “(2) ADF 소핑몰 예제”에서는 ADF의 예제를 살펴봄으로써 ADF Faces의 모바일 기기 지원내용을 살펴보기로 하겠다. 실제 업무에 적용이 가능한 실습 내용을 위주로 구성하였으므로 이 내용이 독자의 주어진 업무를 해결하는데 큰 도움이 되기를 바라는 바이다.

### (1) ADF 웹 애플리케이션

✎ 저자 - Joe Huang

✎ 번역감수/보완 - 박지웅 컨설턴트, 한국오라클 Embedded Global Business Unit (jiwoong.park@oracle.com)

#### 개요

Apple사의 iPhone은 2007년 7월 북미에서 최초로 출시된 이후 18개월 만에 가장 빠른 속도로 판매된 스마트폰이 되었으며 터치스크린에 기반한 발전된 UX 경험과 앱스토어라는 새로운 시장 모델 등, 모바일 컴퓨팅의 미래상을 제시하는 리딩 플레이어의 면모를 과시하고 있다. 아이폰의 폭발적인 성공과 기록들은 경쟁사들인 폰 제조업체들로 하여금 각종 스마트폰의 출시를 앞다투게 하고 있으며 각국의 주요 통신사들도 유행처럼 자체적인 애플리케이션 스토어를 오픈하여 이 새로운 시장에서 승리자의 자리를 차지하기 위해 노력을 아끼지 않고 있다. 이런 폭발적인 모바일 컴퓨팅의 붐은 비단 저렴한 기술 및 네트워크 비용에만 기인하는 것은 아니며 이런 기술적 바탕 위에 사용자에게 가장 중요한 모티베이션인 인터넷 경험이 데스크탑 환경과 동일하게 모바일 환경에서 제공될 수 있게 되었다는 점이 큰 촉매의 작용을 한 것으로 생각할 수 있다. iPhone 성공의 핵심 요인 중 하나는 바로 Safari 웹 브라우저로서, 사용자는 데스크톱 브라우저 수준에서만 가능했던 JavaScript 및 AJAX 기술을 완벽하게 지원하여 웹 상의 대부분의 페이지를

iPhone 상에서 사용할 수 있게 되었다. 완벽한 인터넷 경험의 제공은 곧 SNS, 맵 등 인터넷 서비스의 직접적인 연계를 가능하게 함으로서 사용자들을 자연스럽게 데스크탑을 떠나 모바일 장치들로 컴퓨팅 환경의 공간이 옮겨가도록 만들고 있다. 본 기사에서는 Oracle JDeveloper 및 Oracle Application Development Framework(ADF) Mobile을 사용하여 ADF 기반의 기존 비즈니스 애플리케이션을 최소한의 개발 노력만으로 iPhone에 최적화된 모바일 웹 애플리케이션으로 개발하는 방법에 대해 간단히 소개해 보고자 한다.

#### 모바일 환경의 고려

3.5인치 iPhone 스크린을 이용하여 웹에 액세스하는 모바일 사용자를 위해 제공되어야 하는 웹 애플리케이션의 디자인은 일반적인 22인치 LCD 모니터를 통해 제공되는 웹 애플리케이션과는 전혀 다르다. 오늘날 시장에 출시되어 있는 많은 웹 개발 툴들이 다음과 같은 2개의 카테고리 중 하나에 맞게 설계되어 있으며 이로 인해 개발자들은 선택을 요구 받고 있다.

- 데스크톱 웹 브라우저를 대상으로 하는 툴: 이들은 데스크톱 브라우저가 제공하는 AJAX/JavaScript 기능을 완벽하게 활용한다. 또한 17인치 화면 크기 또는 1024x768 이상의 해상도를 위한 유저 인터페이스를 개발하는 데 초점을 맞추고 있다.
- HTML 모바일 웹 브라우저를 대상으로 하는 툴: 이들은 일반적으로 240x320 이하의 작은 해상도를 위한 유저 인터페이스를 제공하는 데 초점을 맞추고 있다. 또한 많은 종류가 존재하는 모바일 브라우저들에서 모두 구동 가능한 페이지의 구현이 될 수 있도록 가장 최소한이면서 공통적인 기능 집합만을 제공하는 접근 방식을 취하고 있다.

이들 두 가지 툴들은 결국 개발자에게 최대 기능이 제공되는 유저 인터페이스와 대량의 페이지 다운로드를 제공하는 웹 사이트를 개발하도록 하거나 Safari 브라우저처럼 고급 AJAX 기능 같은 데스크탑 브라우저 수준의 기능을 제공할 수 있는 최신 브라우저들이 있음에도 불구하고 가장 베이직한 HTML 만으로 구현된 웹사이트 개발만이 가능하도록 한정시켜 버린다.

하지만, Oracle ADF Mobile Browser 클라이언트는 개발자들이 iPhone Safari 브라우저의 AJAX 기능을 최대한 활용할 수 있게 함과 동시에 베이직한 HTML 이나 WAP 2.0 콘텐츠만이 지원되는 모바일 브라우저와도 완벽한 호환성을 유지할 수 있도록 지원한다. 뿐만 아니라, 개발자들이 웹 애플리케이션을 네이티브한 iPhone 애플리케이션과 동일한 룩앤필을 가지게 함으로써 iPhone에 최적화된 유저 경험을 제공할 수 있도록 한다.

### Oracle ADF를 이용한 모바일 개발의 기초 지식

Oracle ADF는 Oracle ADF Mobile을 통한 모바일 웹 애플리케이션 개발을 지원한다. 모바일 지원의 핵심은 ADF Faces의 근간이 되는 MyFaces Trinidad 구성 요소를 통해 제공되는 JavaServerFaces(JSF) 모바일 렌더 키트(Render Kits)와 Oracle JDeveloper 디자인 타임의 두 가지 지원으로 구성된다. ADF Mobile을 통해 제공되는 주요 특징들은 다음과 같다.

- 브라우저 유저 에이전트 탐지 및 기능 제공 - Trinidad 구성 요소의 모바일 렌더 키트는 사용자 에이전트를 탐지하여 클라이언트 브라우저 유형을 확인한 다음, 이를 바탕으로 지원 가능한 웹페이지 형태를 제공한다. 예를 들어, 전체 AJAX 기능을 지원하지 않는 모바일 브라우저로 전송될 때는 부분 페이지 리프렌시 지원이 자동으로 비활성화 된다. 이 메커니즘은 현재 많은 종류의 모바일 브라우저를 지원하며 대략 다음과 같은 카테고리 분류할 수 있다.

- iPhone Safari 브라우저
- BlackBerry 브라우저

- Nokia S60 브라우저
- Windows Mobile 브라우저
- Basic HTML 브라우저

- 모바일 브라우저 렌더링 최적화 - 각 모바일 브라우저는 지원하는 UI 구성 요소가 각기 다르며 또한 각 모바일 장치 역시 서로 다른 외형과 해상도를 가지고 있다. Trinidad 구성 요소는 이들 UI 요소를 차별적으로 전송하여 사용자 인터페이스의 디스플레이와 레이아웃에 맞게 최적화하므로 개발자들은 각각의 대상 모바일 장치의 다양성을 고민해야 하는 부담에서 해방될 수 있다.

- ADF Model에 대한 완벽한 지원 - 모든 ADF Model 및 Business Logic 구성 요소들은 일반적인 데스크탑 ADF 애플리케이션과 동일하게 JDeveloper 상에서 드래그 앤 드롭 지원을 통한 모바일 개발 형태를 지원한다.

- 모바일 뷰 생성 및 구성 요소 팔레트(Component Palette) - JDeveloper의 JSF Page 생성 시점에서 개발자들은 모바일 장치를 페이지의 대상으로 선택할 수 있다. 일단 설정이 되면, 구성 요소 팔레트는 모바일 브라우저상에서 지원되는 구성 요소만을 나열하여 지원되는 기능 집합만을 사용할 수 있도록 한다.

- 뷰 에디터(View Editor)의 화면 해상도 지원 - 개발자들은 JDeveloper의 뷰 에디터 크기를 모바일 장치 화면의 사이즈로 설정할 수 있다. 이러한 기능을 이용하여 개발자들은 실제 타겟인 모바일 장치에서 페이지가 최종적으로 보이게 될 개략적인 형태를 빠르게 확인할 수 있다.

ADF 모바일에는 다음과 같은 몇 가지 제약 사항이 있다.

- 오직 Trinidad 구성 요소만이 모바일 장치를 위한 렌더 키트를 포함하고 있다. Oracle ADF Faces 구성 요소는 2009년 7월 현재 아직 모바일 용 렌더 키트를 포함하고 있지 않다.
- 모바일 디바이스 상에서는 전체 Trinidad 구성 요소 중 지원되지 않는 일부 요소가 있다. 또한, 지원되는 Trinidad 구성요소 중 몇 가지에 대해서도 특수한 제약 사항이 존재한다. 보다 자세한 내용은 OTN(Oracle Technology Network) 웹 사이트의 ADF Mobile Component Tags Summary 페이지에서 확인할 수 있다.

### iPhone에 최적화된 Oracle ADF 모바일 애플리케이션 개발하기

이제 Oracle ADF 및 JDeveloper를 이용하여 iPhone에 최적화된 간단한 모바일 애플리케이션을 개발하는 방법을 소개하고자 한다. 본 기사에서는 오라클 데이터베이스의 예제로 제공되는 HR 스키마를 사용하는 ADF 예제를 바탕으로 하며 ADF 예제에서 생성된 데이터 컨트롤을 사용하여 iPhone의 룩앤필을 제공하는 간단한 데이터 조회 페이지를 작성하는 과정을 간략히 소개한다. 본 예제에서 사용되는 iPhone 용 스타일 시트 파일과 이미지 파일들을 참조 섹션의 URL에서 다운로드 받을 수 있다.

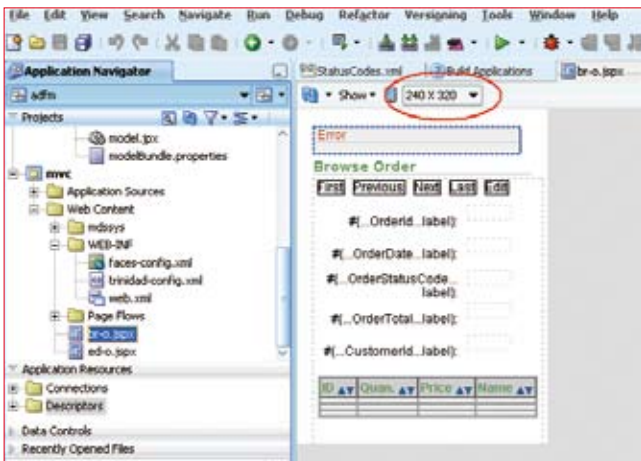
iPhone 용 모바일 웹 애플리케이션의 개발은 대략적으로 다음과 같은 단계로 구성된다.

- 1 기존 비즈니스 로직 구성 요소의 재사용 또는 개발 : ADF Mobile은 ADF Model 및 비즈니스 구성 요소와 완벽하게 호환되기 때문에 개발자들은 다른 프로젝트의 모든 기존 비즈니스 로직 구성 요소를 손쉽게 재사용할 수 있다. 또한, 페이지의 뷰 에디터에서 Data Controls의 드래그 앤 드롭을 지원하는 개발 형태도 완벽하게 지원한다.
- 2 모바일 애플리케이션을 위한 페이지 플로우 및 뷰 개발 : 모바일 애플리케이션을 위한 페이지 플로우 및 각 뷰를 개발하는 프로세스는 기본적으로 JSF 애플리케이션의 개발과 동일하다. 페이지를 개발하는 뷰 에디터의 유용한 기능 중 하나는 디자인 모드에서의 캔버스 크기를 지원하고자 하는 모바일 장치의 해상도로 설정할 수 있는 점이다. JDeveloper는 일반적인 몇 가지 모바일 해상도를 지원하고 있으며 iPhone 해상도는 수동적으로 추가할 수 있다.
- 3 iPhone Safari 브라우저를 위한 최적화 : iPhone 고유의 외관과 느낌을 살리고 iPhone 서비스와 상호 작용하며 네이티브한 iPhone 애플리케이션과 유사한 사용자 경험을 제공하기 위해 몇 가지 기법들이 사용될 수 있다.

아직 국내에 iPhone이 정식 출시되지 않은 관계로 세 번째의 iPhone 서비스와의 연동과 같은 본격적인 구현 및 테스트는 어렵다. 그러나 Safari 브라우저와 몇몇 iPhone 에뮬레이터를 통해 간단한 테스트는 가능하다. 하루라도 빨리 iPhone이 정식 출시되어 국내에서도 iPhone 특유의 UX를 경험해 볼 수 있기를 기대한다.

### 뷰 에디터에 iPhone 해상도 추가하기

뷰 에디터 내에 나열된 스크린 해상도는 개발자들이 뷰를 디자인할 때 시각적인 지원을 제공한다. JDeveloper는 기본적으로 240x240, 240x260, 240x320 및 320x240 등과 같은 해상도를 지원하지만 iPhone은 이와 달리 320x480 해상도를 사용하므로 지원 해상도를 추가해야 한다.



### 뷰 에디터에 iPhone 해상도 추가하기

- 1 다음의 파일을 찾아 임시 위치에 복사한다. 이는 JDeveloper에서의 모바일 개발 기능을 제공하는 JDeveloper Extension 파일이다.

<JDeveloper Install Home>jdev/extensions/oracle.wireless.dt.jar

- 2 압축 해제 프로그램을 이용하여 jar파일을 열고 dc.xml 파일을 찾아 수정한다. dc.xml 파일에는 뷰 에디터에 다양한 장치 해상도 항목을 정의하는 라인이 포함되어 있으며 다음과 같은 형태로 정의된다.

```
reference-device name="320 X 240" displayable-name="320 X 240" width="320" height="240" TablesCapable="true" />
```

- 3 240x240 해상도 항목 바로 위에 다음 행을 삽입한다.

```
<reference-device name="320 X 480" displayable-name="320 X 480" width="320" height="480" TablesCapable="true" />
```

- 4 수정된 dc.xml 파일을 다시 oracle.wireless.dt.jar 파일 내의 동일한 위치에 저장한다.
- 5 일단 만약을 위해 jdev/extensions 디렉토리 내에 원본 oracle.wireless.dt.jar 파일의 백업 사본을 만든다. 그런 다음, 업데이트된 oracle.wireless.dt.jar를 jdev/extensions 디렉토리로 다시 복사한다.
- 6 만약 JDeveloper가 열린 상태였다면 재시작 한다. 뷰 에디터 해상도 목록 상자에 추가된 해상도를 확인할 수 있다.

### iPhone 고유의 룩앤필 구현

스킨 구조를 사용하여 Oracle ADF Mobile 애플리케이션이 iPhone 고유의 룩앤필을 가질 수 있도록 한다. 또한 이러한 스킨 작업을 통해 여러 다른 장치를 위한 같은 웹 페이지의 여러 버전을 일일이 개발하지 않고 스킨 인식만을 통해 각 UI 구성 요소가 자동적으로 플랫폼 고유의 룩앤필을 가질 수 있도록 한다.

iPhone의 스킨 생성을 위한 첫 단계는 JSF 애플리케이션이 런타임 시에 여러 다른 모바일 브라우저를 위한 각각의 스타일 시트를 적용할 수 있도록 구성하는 것이다. 이를 위해 먼저 각 애플리케이션 페이지 플로우에서 사용될 Managed Bean을 생성하고 여기에서 해당 플랫폼의 브라우저를 탐지하여 적절한 스킨 패밀리 명을 리턴하는 메소드를 구현해야 한다. 이 메소드는 Trinidad-config.xml 파일에서 호출되며 다음과 같이 대괄호로 묶어서 메소드 호출의 리턴값이 애플리케이션이 사용할 스킨 패밀리 명이 되도록 정의한다.

```
<skin-family>#{AgentUtil.phoneFamily}
```

AgentUtil은 구현한 Managed Bean의 클래스명이며 phoneFamily는 스킨 패

밀리명을 리턴하도록 구현된 메소드이다. 다양한 스킨들이 Trinidad-skins.xml 파일에서 정의될 수 있으며 이러한 정의에는 스킨 명 및 실제로 스타일 시트가 포함 될 물리적인 CSS 파일의 위치 등이 포함된다. 예를 들어 iPhone 용 스킨 군은 다음과 같이 정의할 수 있다.

```
<skin>
  <id>iphone</id>
  <family>iphoneFamily</family>
  <render-kit-id>org.apache.myfaces.trinidad.desktop</render-kit-id>
  <style-sheet-name> styles/iPhone.css</style-sheet-name>
</skin>
```

예제의 AgentUtil.phoneFamily 메소드는 요청을 보낸 브라우저가 iPhone의 Safari 브라우저로 판단되는 경우 iphoneFamily 문자열 반환하며 결과적으로 렌더링 된 페이지가 styles/iPhone.css 스타일시트 파일을 사용하게 한다. 다음은 AgentUtil.phoneFamily 메소드의 코드 일부로서 iPhone, Symbian 그리고 기타의 세 가지 케이스에 대해 각기 적용할 스킨 패밀리 명을 리턴하도록 구현되어 있다.

```
private String IPHONE_SKIN = "iphoneFamily";
private String SYMBIAN_SKIN = "symbianFamily";
private String DEFAULT_SKIN = "minimalFamily";

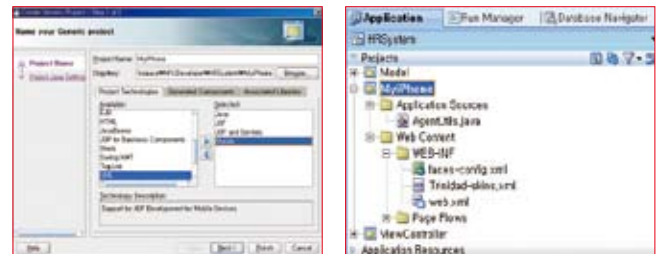
public String getPhoneFamily() {
    if (phoneFamily.isEmpty()) {
        FacesContext fc = FacesContext.getCurrentInstance();
        HttpServletRequest req = (HttpServletRequest)fc.getExternalContext().
        getRequest();
        String agent = req.getHeader("User-Agent");

        if (agent != null && agent.indexOf("iPhone") > -1) {
            phoneFamily = IPHONE_SKIN;
        } else if (agent != null && agent.indexOf("Symbian") > -1) {
            phoneFamily = SYMBIAN_SKIN;
        } else {
            phoneFamily = DEFAULT_SKIN;
        }
    }
    return phoneFamily;
}
```

코드 중간 정도에 있는 agent.indexOf() 부분을 보면 HttpServletRequest 객체에 포함된 헤더정보 중 User-Agent 정보를 분석하여 어떠한 브라우저로부터 HTTP Request가 이루어졌는지를 판단하도록 되어있다. 만약 agent.indexOf("iPhone")을 통해 Header 정보에 iPhone 임을 알리는 정보가 포함되어 있는 것이 인식된 경우 iPhone이 접속한 것으로 판단된다. 보다 정확한 판단을 위해 좀더 상세한 룰을 구현할 수도 있지만, 여기서는 가장 일반적으로 알려진 문자열 패턴만을 사용하였다.

- **iPhone** - 모든 iPhone Safari 브라우저를 식별하는 데 사용된다.
- **Symbian** - 모든 Nokia S60 브라우저를 식별하는 데 사용된다.
- **BlackBerry9000** - 모든 BlackBerry 9000 시리즈 브라우저를 인식하는 데 사용된다.
- **BlackBerry88** - BlackBerry 8800 시리즈 브라우저를 식별하는 데 사용된다.
- **BlackBerry83** - BlackBerry 8300 시리즈 브라우저를 식별하는 데 사용된다.
- **BlackBerry** - 기타 모든 BlackBerry 브라우저를 식별하는 데 사용된다.
- **Windows CE** - 모든 Windows Mobile 브라우저를 식별하는 데 사용된다.
- **Android** - Android의 WebKit 기반 브라우저를 식별하는 데 사용된다.

이제 ADF 예제에서 사용한 HRSystem 애플리케이션에 iPhone 프로젝트를 추가한다. 먼저 JDeveloper를 열고 새로운 프로젝트를 Generic Project로 생성한다. 먼저 앞서 설명된 내용에 바탕하여 AgentUtil 클래스를 구현 한 후 해당 위치를 참조하도록 WEB-INF의 Trinidad-config.xml 파일의 내용을 수정한다.



Trinidad-skins.xml 파일을 WEB-INF 폴더에 생성하고 예제처럼 iPhone 스킨 패밀리 정의를 추가해 준다. 앞서 다운로드 받은 iPhone.css 파일이 WEB-INF/styles 디렉토리에 위치하도록 복사한다. 언급했듯이 iPhone 용 스타일 시트 파일과 이미지 파일들을 참조 섹션의 URL에서 다운 받을 수 있다.

#### Skinning 및 Trinidad 구성 요소를 이용한 일반적인 iPhone UI 요소 구현

iPhone 고유의 룩앤필을 달성하기 위한 다음 단계는 iPhone UI의 공통 구성 요소들의 형태를 지원하도록 적절한 스타일 클래스를 스타일 시트에서 정의하고 이

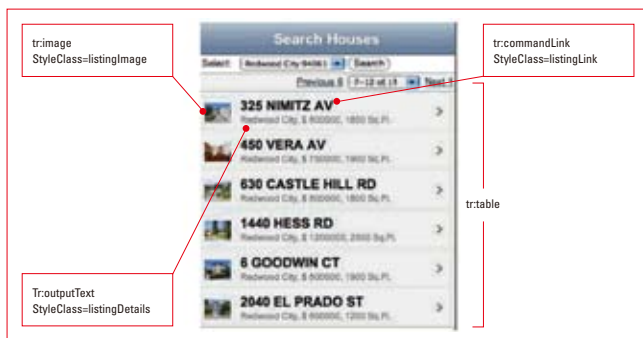


를 사용하여 실제 UI를 작성하는 것이다. iPhone 네이티브 애플리케이션에는 기본적으로 Navigation Panel Lists, Field Set Panel 및 Navigation Bar와 같은 공통 UI 요소가 제공 된다. 이들 UI 요소들을 구성하기 위한 세부적인 스타일 클래스들은 앞서 언급된 iPhone 용 스타일 시트 파일인 iPhone.css에 모두 정의되어 있으므로 상세한 정의를 원한다면 파일 내용을 참고하도록 하자. 이제 언급된 세 가지 UI 요소에 각각에 대한 세부 사항을 살펴 보도록 하겠다.

### Navigation Panel Lists

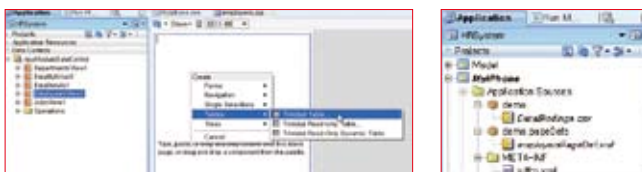
Navigation Panel Lists는 동일한 종류의 데이터 목록을 나열하는 테이블 구성 요소로서 나열된 각 행은 여러 종류의 데이터를 포함하는 형태를 띠고 있다. 이는 사용자가 테이블을 통해 요약된 형태의 데이터 목록을 탐색하고 관심 있는 행을 터치함으로써 보다 상세한 추가 데이터 사항을 조회해 보는 사용자 시나리오를 지원한다.

다음은 판매 중인 주택 목록을 나열하는 Panel List 예제 화면이다.



Panel List의 주요 구성요소는 전체를 감싸는 테이블과 각 행을 구성하는 한 개의 이미지, 상세 보기로 링크되는 메인 타이틀, 서브 타이틀의 형태로 구성되며 각각의 형태를 정의하는 스타일 클래스가 iPhone.css 파일에 정의되어 있다.

그럼 HR 예제에서 간단한 Employer 조회 테이블을 구현해 보겠다. JDeveloper를 열고 New 메뉴에서 Web Tier 카테고리 중 JSF Page를 선택하여 모바일 브라우저용 빈 페이지를 생성한다. 이렇게 생성된 페이지에 기존 ADF 예제에서 구현된 Employee1View1 데이터 컨트롤을 드래그하여 드롭하고 Trinidad Table을 지정하여 기본 테이블이 생성되도록 한다.



이 작업의 결과로 pageDef 패키지에 각 바인드의 명세가 포함된 xml 정의와 DataBindings.cpx 파일이 적절히 구성된 것을 확인할 수 있다. 다음으로 뷰 에디터에서 Trinidad 테이블소스의 태그들을 수정하여 Panel List를 구현한다.

```
<tr:table value="#{bindings.EmployeesView1.collectionModel}"
    var="row"
    rows="7"
    width="100%"
    styleClass="iphoneTable"
    emptyText="#{bindings.EmployeesView1.viewable ?
    'No rows yet.' :
    id="mainTable" horizontalGridVisible="false" >
```

```
<tr:column >
```

```
<tr:panelGroupLayout layout="vertical" styleClass="listing">
```

```
<tr:outputText value="#{row.bindings.PhoneNumber.inputValue}"
    styleClass="listingDetails"/>
```

```
<tr:commandLink text="#{row.bindings.LastName.inputValue} ,
    #{row.bindings.FirstName.inputValue} ,
    styleClass="listingLink"
    partialSubmit="true"
    actionListener="#{agentUtil.gotoPage2}"
    id="myLink1"
    disabled="#{!bindings.Execute.enabled}"
    onclick="iPhone.slideFragments('page2',
    'page1')"/>
```

```
</tr:commandLink>
```

```
<tr:image styleClass="listingImage"
    source="/images/#{row.bindings.EmployerId.inputValue}.
    png"/>
```

```
</tr:panelGroupLayout>
```

```
</tr:column>
```

```
</tr:table>
```

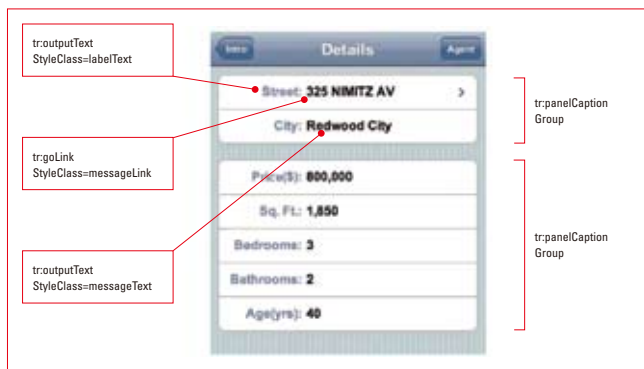
UI 구성요소는 먼저 데이터 바인딩을 위해 tr:table을 명세하고 실제 레이아웃은 tr:panelGroupLayout을 스타일 클래스 listing을 사용하여 정의한다. 각 행의 내용을 이루는 세 항목들은 이미지, 메인타이틀, 서브타이틀이다. 이들은 각각 image, commandLink, outputText 요소로서 각각 listingImage, listingLink, listingDetails 클래스로 정의한다. CommandLink의 경우 setActionListener를

사용하여 상세 정보 페이지로의 링크를 명세할 수 있다.

항상 테이블 구현 시에는 JDeveloper를 통한 생산성을 최대화하기 위해 데이터 컨트롤을 드래그하여 Trinidad 테이블로 생성하고 난 이후에 만들어진 테이블 소스를 수정하는 순으로 UI 구성 요소를 개발하는 것이 편리하다. 이는 다른 UI 구성 요소의 개발 때에도 동일하게 적용될 수 있다.

### Field Set Panel

Field Set Panel은 일반적으로 데이터에 대한 상세 정보를 표시하는 데 사용되며 관련 상세 정보를 표시하는 다른 스크린 또는 뷰를 탐색하는 링크를 제공한다. 앞서 보았던 테이블 예제 화면에서 특정 주택 정보를 보기 위해 사용자가 행을 터치하게 되면 다음과 같은 뷰 형식으로 주택에 대한 상세 정보 및 추가 상세 정보에 대한 링크가 제공되도록 할 수 있다.



Field Set Panel에서는 제공하고자 하는 정보의 형태에 따라 적절한 panelCaptionGroup 수를 구성하고 원하는 속성 각각을 명세 하는 적절한 outputText를 labelText와 messageText 스타일을 써서 배치하도록 디자인 한다. 모바일 장치 특성에 맞도록 너무 많은 필드를 한 화면에 보여주지 않고 적절히 링크를 통해 페이지를 나누어 구성해 주는 것이 중요하다.

다시 HR 예제로 돌아와서, 테이블 예제 때와 마찬가지로 Employer1View1 구성 요소를 빈 페이지 상에 Trinidad Form으로서 드래그 앤 드롭하여 데이터 바인딩을 정의한다. 이후 생성된 뷰의 소스를 열고 상기에 언급된 스타일 클래스들을 사용하여 iPhone 록앤필이 느껴지도록 페이지를 수정한다. 다음은 Field Set Panel을 구현한 소스의 예제 일부이다.

```
<div class="panelBase">
  <tr:panelCaptionGroup>
    <div class="row">
      <tr:outputText styleClass="labelText" value="#{agentUtil.name}"
```

```
      truncateAt="0"/>
      <tr:outputText styleClass="messageText"
        value="#{sessionScope.FirstName}" />
    </div>
  <div class="row">
    <tr:outputText styleClass="labelText" value="Last Name"/>
    <tr:commandLink text="#{sessionScope.LastName}"
      styleClass="messageLink"
      partialSubmit="true"
      id="myLink2"
      actionListener="#{agentUtil.gotoPage3}"
      onclick='iPhone.slideFragments("page3",
        "page2");'
    />
  </div>
</tr:panelCaptionGroup>
<tr:panelCaptionGroup>
  <div class="row">
    <tr:outputText styleClass="labelText" value="Email"/>
    <tr:outputText styleClass="messageText"
      value="#{bindings.LastName}@oracle.com"/>
  </div>
  <div class="row">
    <tr:outputText styleClass="labelText" value="Salary"/>
    <tr:outputText styleClass="messageText" />
      value="#{sessionScope.Salary}"/>
  </div>
  .....
</tr:panelCaptionGroup>
</div>
```

전체적인 백그라운드는 panelBase 스타일로 정의되며 여기에서 핀 스트립(pin-strip) 이미지의 배경이 제공된다. 두 개의 panelCaptionGroup 컨트롤이 데이터 디스플레이를 실제 두 개의 그룹으로 나누고 있으며 위쪽의 패널에는 세션에 저장된 FirstName과 LastName을 텍스트로 출력하고 상세정보 페이지로의 링크를 제공하기 위한 commandLink를 정의하고 있다. 아래쪽 패널에서는 Employer 레코드의 나머지 속성들을 필드 형태로 나열하고 있다. 각 필드는 labelText 스타일로 정의된 레이블과 messageText 스타일로 정의된 실제 문자열 값을 출력하도록 구성한다.

### Navigation Bar

Navigation Bar는 iPhone 애플리케이션을 위한 기본 탐색 메커니즘을 제공하는

대표적인 컴포넌트이다. 앞서 언급된 테이블 및 상세보기 구성요소 예제상에서 항상 내비게이션 바가 화면 상단에 위치함을 확인할 수 있다. 내비게이션 바를 통해 이미 방문된 이전 페이지로의 복귀나 사전 정의된 사용자 페이지 플로우를 버튼 터치를 통해 보다 쉽게 진행해 갈 수 있도록 한다.



내비게이션 바 전체는 toolbar 스타일 클래스를 바탕으로 하는 panelHeader 클래스를 사용하여 각 버튼들은 backButton 스타일이 지정된 commandLink를 사용하여 각각을 명세하도록 한다.

```
<tr:panelHeader text="Details" styleClass="toolbar" rendered="true">
  <tr:commandLink text="Agent" styleClass="button"
    disabled="false"
    rendered="true"
    action="toAgent"/>
  <tr:commandLink text="Intro" styleClass="backButton"
    disabled="false"
    rendered="true"
    action="backtoIntro"/>
</tr:panelHeader>
```

화면이 작은 모바일 장치의 특성 상 많은 정보를 보여주기 위해서는 잘 짜여진 뷰들과 내비게이션 시나리오가 필요하다. JDeveloper의 JSF Flow 디자인 기능을 사용하여 많은 뷰가 참여하는 복잡한 시나리오를 시각적으로 모델링하고 구현할 수 있게 함으로써 높은 개발 생산성 향상을 기대할 수 있다.

### iPhone 서비스와의 상호 작용

일반적으로 모바일 장치를 위한 브라우저는 전화 기능 및 e-메일 기능 등과 같은 장치 서비스와의 기본 상호 작용 기능을 제공한다. iPhone은 iPhone Google Maps 애플리케이션과의 상호 작용을 추가하여 이를 한 단계 더 발전시켰다. 이러한 상호작용 기능들은 Oracle ADF 및 JDeveloper를 이용하여 개발할 경우 간단히 몇 줄의 코드를 추가 하여 손쉽게 연계하는 것이 가능하다. JDeveloper에서의 애플리케이션 구현은 페이지에 실제 전화 번호와 주소 등과 같은 데이터가 저장된 데이터 컨트롤 계층으로의 연계가 쉽고 간단하게 이루어질 수 있기 때문에 페이지 상에서 곧바로 전화를 걸거나 이메일을 전송하는 등의 기능을 애플리케이션에 간단히 구현할 수 있다.

### 전화 서비스

iPhone Safari 브라우저는 페이지에서 전화 번호를 인식하는 기능을 기본적으로 내장하고 있으며 사용자들이 해당 번호를 클릭하여 전화를 걸 수 있도록 한다. 하지만 페이지 상에 노출된 전화 번호에 대해 항상 이 기능이 실행되는 것은 아니다. 만약 전화 번호가 단순히 outputText 필드에 디스플레이 된 경우, Safari 브라우저가 이를 전화 번호로 인식할 것이라고 보장할 수 없다. iPhone이 전화 번호를 명확하게 인식하도록 보장하기 위해서는 개발자들이 전화 번호를 tr:goLink 구성 요소에 표시하고 goLink의 목적지 속성에 tel: 키워드를 추가해야 한다. 다음의 예제 코드에서 동일 세션 내에 존재하는 AgentPhone 값에 저장된 전화 번호로 전화를 거는 구현을 볼 수 있다.

```
<tr:goLink styleClass="messageLink"
  destination="tel:#{sessionScope.houseDetails.AgentPhone}"
  text="#{sessionScope.houseDetails.AgentPhone}"/>
```

또한, iPhone은 RFC 2086 프로토콜에 대한 부분적인 지원을 제공하고 있다. 예를 들어 전화 번호로 전화를 걸고, 몇 초 동안 일시 정지한 다음, 내선 번호를 누르는 것과 같은 형태의 사용자 시나리오가 가능하다. 다음의 예제 코드에서는 02-999-6666 번호에 전화를 걸고 2초 간 기다린 후 (pp) 8을 누르게 된다.

```
<tr:goLink styleClass="messageLink"
  destination="tel:01-999-6666;pp8"
  text="1-800-use-jdev"/>
```

### Email 클라이언트

iPhone Safari 브라우저는 iPhone Email 클라이언트의 호출에 대한 인터페이스도 제공한다. Email 목적지를 포함한 Email 클라이언트를 호출하기 위해 개발자는 tr:goLink 구성 요소를 이용하여 다음과 같이 e-필드를 표시하고 구성 요소의 목적지 속성에 mailto:를 추가할 수 있다.

```
<tr:goLink styleClass="messageLink"
  destination="mailto:#{sessionScope.houseDetails.AgentEmail}"
  text="#{sessionScope.houseDetails.AgentEmail}"/>
```

사용자가 goLink를 터치하면, iPhone Email 클라이언트가 시작되고 To: 필드는 AgentEmail 변수의 값으로 채워지게 된다.

iPhone은 이외에 일반적인 e-메일 필드도 지원한다. 개발자들은 간단히 goLink 구성 요소의 목적지 속성을 다음과 같이 추가하면 된다.

- 다수의 수신자 추가 - 콤마(,)로 구분하며 e-메일 주소를 추가한다.
- e-메일 제목 - subject=<subject text>
- Cc 수신자 - c=<Email Addresses>
- Bcc 수신자 - bcc=<Email Addresses>
- 메시지 본문 - body=<Message Text>

각 필드는 “?” 문자로 분리해야 한다.

다음의 코드는 제목, cc 수신자 및 메시지 본문을 가진 e-메일로 Email 클라이언트를 사전에 채운 후에 띄워지게 한다.

```
<tr:goLink styleClass="messageLink"
  destination="mailto:john.doe@oracle.com,
  jane.doe@oracle.com?subject=Questions?
  cc=mymame@myhome.com?
  body=I have questions for you"
  text="#{sessionScope.houseDetails.AgentEmail}"/>
```

### iPhone Google Maps

iPhone Safari 브라우저는 <http://maps.google.com>에 대한 요청을 인터셉터하고, 대신 iPhone 상의 Google Maps 애플리케이션을 호출한다. Google Maps 애플리케이션에서는 특정 주소에 핀 아이콘을 표시하는 기능을 일부 지원한다. 일단 핀이 표시되면, iPhone 사용자는 GPS와 “길 찾기(getting directions)” 기능을 사용하여 해당 주소로 향하는 길을 안내받거나 해당 주소의 위치를 확인할 수 있다. 이 기능에 접근하기 위해서는 개발자들이 tr:goLink 구성 요소를 사용하여 goLink 구성 요소의 목적지 속성을 Google Maps 및 해당 주소에 대한 URL로 채워야 한다. Google Maps URL은 다음과 같은 포맷이 되어야 한다.

```
http://maps.google.com/maps?q=<Address\_Field>
```

예를 들어 아래 코드는 세션 변수 houseDetail.Street, houseDetail.City 및 houseDetail.Zip에 따라 지정된 위치를 표시할 것이다.

```
<tr:goLink styleClass="messageLink"
  destination="http://maps.google.com/maps?q=#{sessionScope.houseDetails.
  Street},
  +#{sessionScope.houseDetails.City},
  +#{sessionScope.houseDetails.Zip}"
  text="#{sessionScope.houseDetails.Street}"/>
```

이 기능을 응용하여 차량용 내비게이션 기능을 웹페이지에 추가 할 수도 있다.


이를 위한 Google Maps URL의 포맷은 다음과 같다.

```
http://maps.google.com/maps?saddr=<Start\_Address>&daddr=
```

구글 맵과의 연동은 매우 강력한 기능으로서 ADF와 JDeveloper 개발자들이 간단한 코드 몇 줄 추가만으로 차량용 내비게이션 기능이 제공되도록 지원한다. 사용자들은 ADF Mobile 뷰에서 연결되는 링크를 클릭하여 Google Maps 기능을 통해 위치를 확인하거나 차량용 길 안내 서비스 등을 받을 수 있다.

### 결론

iPhone은 모바일 장치를 위한 UX 개념을 새롭게 정의했다. 많은 iPhone 유저들이 특별한 학습 없이 감각적으로 손쉽게 다양한 애플리케이션을 이용한다. 이런 낮은 학습곡선의 UX가 가능해진 이유는 iPhone의 코어 애플리케이션의 유저 인터페이스가 iPhone 하드웨어와 완벽하게 연동된다는 점에서 비롯된다. 즉 iPhone 상에서 실행되는 여타 모든 애플리케이션들은 이러한 사용성의 유지를 위해 유저 인터페이스 표준을 준수해야 하며, 만약 이를 준수하지 않는 경우 사용자들은 익숙하지 않은 UX에 불편함을 느끼게 될 것이다. iPhone의 앱스토어 모델에서는 사용자가 서비스 업체에서 제공해준 애플리케이션에 귀속되지 않으며 애플리케이션 선택의 주도권이 사용자에게 있다. 홍수처럼 쏟아지는 모바일 애플리케이션에서 사용자는 더욱 매력적인 애플리케이션을 찾으며 손쉽게 애플리케이션을 바꾼다. 낮은 사용성, 불편한 UX의 애플리케이션은 결과적으로 사용자의 손길을 받지 못하게 되고 최종적으로는 버림 받게 될 것이다.

본 기사에서는 Oracle ADF 및 JDeveloper를 이용하여 아주 손쉽게 iPhone의 네이티브 애플리케이션과 동일한 룩앤필을 갖는 웹 애플리케이션을 개발하는 방법을 설명하였다. 2009년 7월 현재 iPhone은 아직 국내에 도입이 이루어지고 있는 상태는 아니며, 각 통신사와 애플 간의 논의가 이루어지고 있다는 소식만이 간간히 들리고 있다. 그러나 많은 사람들이 모바일 혁신의 아이콘과 같은 위치를 차지하고 있는 iPhone의 사용을 바라고 있기 때문에 조만간 국내에서도 사용해 볼 수 있을 것이라 조심스럽게 전망해 본다. 

#### 참조

1. iPhone.css 스타일시트 다운로드 링크:  
<http://www.oracle.com/technology/tech/wireless/samples/iPhone.css> 또는  
<http://pds15.egloos.com/pds/200907/27/94/iPhoneStyleFiles.zip> (이미지포함)
  2. Developing for the Apple iPhone with Oracle JDeveloper and ADF:  
<http://www.oracle.com/technology/pub/articles/luang-iphone.html>
  3. HR 스키마를 사용하는 ADF 튜토리얼:  
[http://www.oracle.com/technology/obe/11jdev/bulldog/ria\\_application/developriaapplication\\_long.htm](http://www.oracle.com/technology/obe/11jdev/bulldog/ria_application/developriaapplication_long.htm)
  4. Apache MyFaces Trinidad 공식 홈:  
<http://myfaces.apache.org/trinidad/>
- Joe Huang은 Oracle ADF Mobile 담당 프로젝트 매니저다

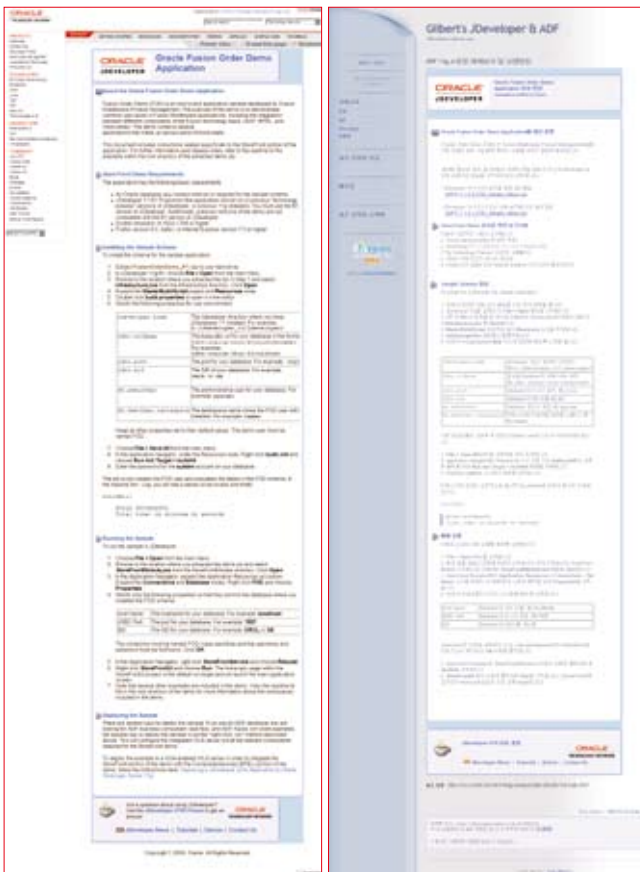


## (2) ADF 쇼핑물 예제

▶ 박길원 Principal Technologist, PTS(Gilbert.Park@oracle.com)

### 개요

본 기사에서는 ADF로 작성된 JDeveloper 11g 쇼핑물예제를 활용하여 쇼핑물 관리자가 모바일 단말기를 통해 손쉽게 주문처리를 할 수 있도록 모바일 기기 화면에 꼭 맞는 모바일 전용 페이지를 추가한 것에 대해 살펴보도록 하겠다.



### JDeveloper 11g에서 쇼핑물 예제 설치 및 실행하기

JDeveloper 11g의 쇼핑물 예제를 설치하는 방법은 OTN에도 소개가 되어 있지만 JDeveloper 버전 별로 설치 파일에 차이가 있고, 영문으로 되어 있는 탓에 어려워하는 사용자들이 있어 필자의 블로그에 다시 정리해 놓았으니 참고하기 바란다.

· OTN 영문자료(11.1.1.0.0버전용) :

<http://www.oracle.com/technology/products/jdev/samples/fod/index.html>

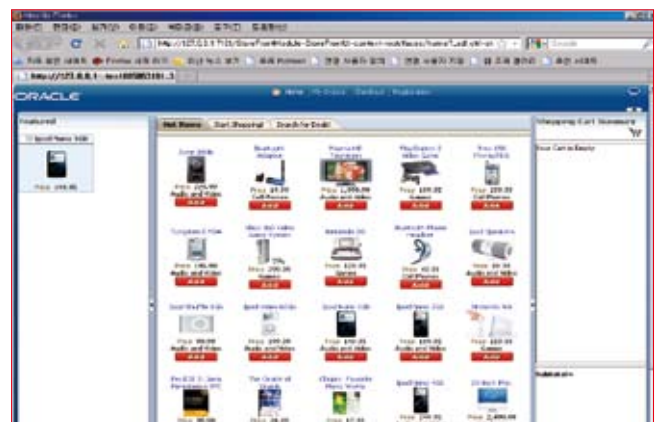
· Gilbert's 한글자료(11.1.1.x.x버전용) :

<http://jdeveloper.egloos.com/5056326>

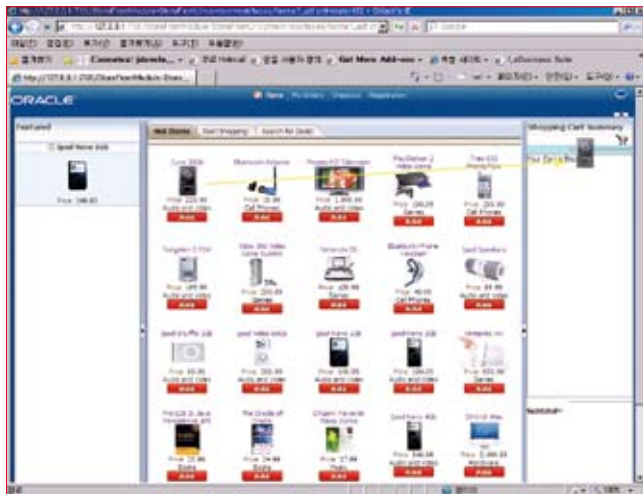
예제 설치를 완료하고 실행을 하게 되면 쇼핑물이 시작된다. 예제 쇼핑물을 이용하기 위해서는 Firefox, 2.0, Safari 또는 Internet Explorer 7.0 이상의 웹 브라우저가 필요하다. 쇼핑물 구현에 대한 기술적인 상세 내용을 알고 싶다면, Developer's Guide for Oracle ADF 11g Release 1(11.1.1) 매뉴얼에서 ADF 샘플 Application 소개([http://download.oracle.com/docs/cd/E12839\\_01/web.1111/b31974/intro\\_tour.html](http://download.oracle.com/docs/cd/E12839_01/web.1111/b31974/intro_tour.html))를 참조하면 된다.

### 쇼핑물 예제(Fusion Order Demo) 살펴 보기

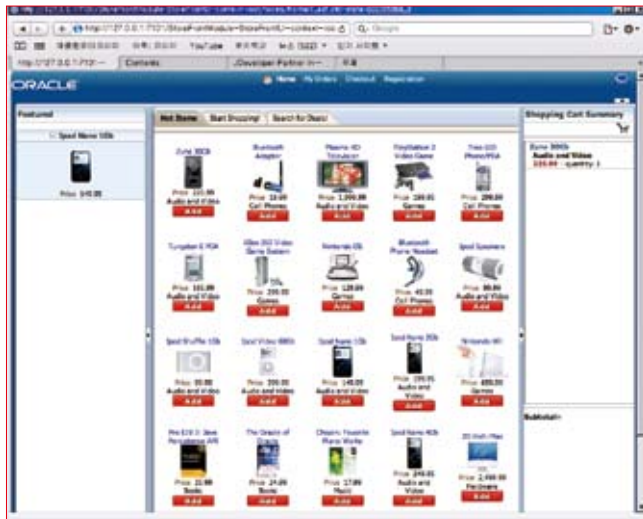
<그림 1> ~ <그림 3>은 각각 Firefox 3.5, Internet Explorer 8, Safari 3.0을 통해 예제 쇼핑물에 접속한 화면이다. 각각의 화면은 Hot Item 목록에서 Zune 30Gb를 선택하여 장바구니에 드래그 앤 드롭 하는 과정을 순서대로 캡처하였다.



<그림 1> Firefox 3.5로 접속한 접속화면(초기 접속 상태)



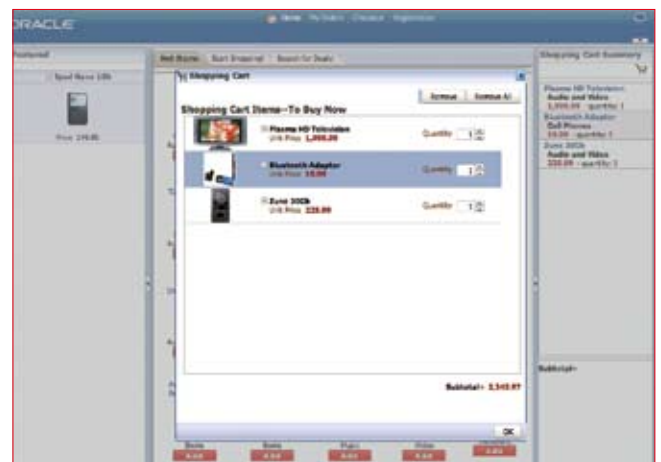
<그림 2> Internet Explorer 8으로 접속한 화면(물건을 장바구니로 드래그 앤드롭 하는 중)



<그림 3> Safari 3.0으로 접속한 화면(드래그 앤드롭으로 장바구니에 물건을 추가한 결과)

표준 기술 기반으로 구현된 ADF UI 컴포넌트로 구현되어 있기 때문에 추가적인 plug-in을 다운 받을 필요도 없고 서로 다른 브라우저에서도 동일한 룩앤필과 기능들을 유지하고 있다.

예제 쇼핑몰에 모바일 페이지들을 추가하기 전에 쇼핑몰의 몇 가지 기능들을 살펴 보도록 하겠다.



<그림 4> 장바구니를 열어서 수량의 변경 또는 삭제할 수 있다.



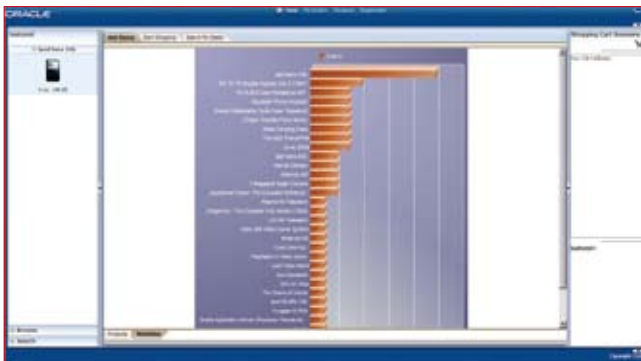
<그림 5> 다양한 형태의 상품 보기 및 팝업으로 좀더 상세한 정보를 볼 수도 있다.



<그림 6> 상품명으로 검색할 수도 있다.



<그림 7> Layout에서 각각의 Pane을 접고 펴거나 크기를 조정할 수 있다.



<그림 8> Hot Item의 서버탭 중에서 통계(Statistics)탭을 클릭하면 주문 현황을 그래프로 보여준다.

그 외 상품에 대한 상세한 정보들을 보는 화면과 사용자 등록이나 주문 상태 등을 볼 수 있는 화면들이 있다.

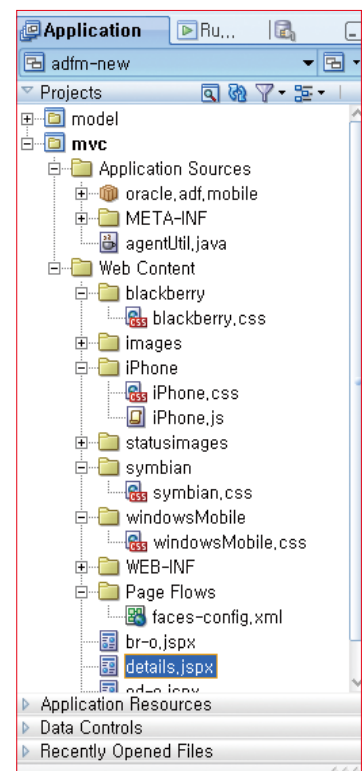
참고로, BPEL Process와의 통합 및 BPM Worklist를 활용한 주문 승인 프로세스 등에 대한 구현은 이번 모바일 주제와는 벗어나기 때문에 언급하지 않고 있지만 다음의 URL을 방문하면 해당 내용을 확인할 수 있다.

<http://www.oracle.com/technology/obe/obe11jdev/bulldog/deploysoaapp/deployment.htm>

### ADF 모바일 예제 설치하기 및 실행하기

앞의 기사 “(1) ADF 웹 애플리케이션”에서는 기존의 ADF 웹 애플리케이션을 iPhone UI로 변경하는 방법에 대해 살펴보았다. 이와 비슷한 방법을 통해 앞에서 설치한 예제의 모델들을 그대로 사용하여 구현된 모바일 예제를 살펴봄으로써 ADF를 활용한 iPhone 용 모바일 웹 애플리케이션 구현에 대한 아이디어를 제공한다. <http://jdeveloper.egloos.com/5059386>에 모바일 예제의 설치 및 실행 방법을 등록하여 놓았으니 해당 페이지의 안내에 따라 설치하기 바란다.

다음 그림은 모바일 예제에는 <그림 9>처럼 다양한 모바일 기기들의 샘플 스타일시트들이 들어 있으므로 향후 다양한 모바일 기기의 웹페이지를 개발하는데 유용하게 쓰일 수 있다.



<그림 9> ADF 모바일 예제에 포함된 Style 들

## Gilbert's JDeveloper & ADF

JDeveloper.esloos.com

### ADF 11g 쇼팔들에 Mobile 지원 예제설치 및 실행방법

Oracle Fusion Order Demo Mobile Application 설치 방법  
Written by Gilbert

#### Oracle Fusion Order Demo Mobile Application에 대한 설명

Fusion Order Demo Mobile Application(FOD)은 Fusion Middleware Product Management팀에 의해 개발한 ADF 11g 버전 중에서 쇼팔들 관리자의 기능을 모바일 기기에서 수행할 수 있도록 작성된 예제입니다.

실행에 필요한 데모 설치파일은 아래 파일을 다운로드 받으시면 됩니다.

\* JDeveloper 11g 버전을 데모 설치파일  
ADF\_Mobile\_Sample\_Gilbert.zip

#### ADF Mobile Demo 설치를 위한 요구사항

아래의 기본적인 사항이 요구됩니다.

- ADF 11g 쇼팔들 패키지(<http://jdeveloper.esloos.com/5056326>)
- JDeveloper 11.1.1.8.0 이상의 버전 ("11g Technology Preview" 버전은 사용불가)
- Firefox 2.0, Safari 또는 Internet Explorer 7.0 이상의 웹브라우저.

#### ADF Mobile Demo 설치

- 위에서 링크된 예제 설치 파일을 다운로드 받으신 후 압축을 풉니다.
- JDeveloper 11g를 실행한 뒤 File > Open 메뉴를 선택합니다.
- 1번 단계에서 압축을 톤 위치로 이동하여 adfm-new.jar를 Open합니다.
- View > Database Navigator 메뉴를 선택한 후 adfm-new application에 연결된 DBConn2드에서 오른쪽 버튼을 누른 후 properties를 선택합니다.
- 아래의 속성(properties)들을 자신의 환경에 맞게 수정해 줍니다.

Password	FOD 스키마의 암호 (예) fusion
Host Name	간혹할 Database의 서버 이름 (예) localhost
JDBC Port	Database의 리스너 포트 (예) 1521
SID 또는 Service Name	Database의 SID/서비스 이름 (예) xxx

다른 속성값들은 그대로 유지합니다(demo user는 반드시 FOD이어야 합니다).

6. File > Save All 메뉴를 선택하여 모두 저장합니다.

#### 실행 방법

아래의 순서에 의해 쇼팔들 모바일 예제를 실행합니다.

- mvc 프로젝트에서 오른쪽 클릭하여 Run을 선택합니다. (mvc 프로젝트의 home.jspx파일이 기본 실행 target이고, iPhone 스타일로 보여지도록 설정 되어 있습니다).
- 모바일 기기에서 어떻게 보여지는지 보려면 아래와 같은 형식의 iPhone 크기 및 해상도를 작성하여 해당 파일을 브라우저에서 열면 됩니다.

```
<script>
window.open("http://127.0.0.1:7101/adfm-new-mvc-context-vol/faces/home.jspx", "", "width=320, height=480, status=no, resizable=no, scrollbars=no, menubar=no")
</script>
```

3. Apple의 Safari 브라우저(<http://www.apple.com/safari/>)로 보시면 실행되는 iPhone Style의 페이지를 보실 수가 있습니다. 아래의 이미지는 Safari 브라우저의 결과 페이지를 iPhone의 웹 이미지와 형식한 결과입니다.



예시처럼 iPhone에 적합한 페이지를 구성하기 위해서는 화면 사이즈 및 록엔필이 상당히 중요하다. 아래는 모바일 예제에서 제공하고 있는 iPhone용 웹페이지들을 캡처한 그림들이다.



<그림 10> 주문 사항을 확인하는 관리자 화면

<그림 11> 주문 사항의 상세 정보를 확인하는 화면

중요한 사항은 ADF로 개발된 웹시스템들은 ADF의 구조상 Model-View-Controll의 형태로 나뉘어 지게 되므로, 향후 업무 확장성이 용이하고, 서로 다른 플랫폼이나 다양한 클라이언트의 지원이 필요한 경우 많은 노력 없이도 구현이 가능하다는 것이다. 비록 ADF가 범용적인 개발 틀은 아니지만 웹 기반의 업무 시스템을 구현하고 유지 보수를 위한 Framework으로는 막강한 기능들을 제공하고 있다.

이제 Web2.0 기반의 새로운 UI컴포넌트들로 다시 태어난 ADF 11g와 함께 모바일의 세계로 빠져보는 것은 어떨까? 