



Oracle ADF Mobile: 단일 아키텍처를 기반으로 엔터프라이즈 애플리케이션과 모바일 애플리케이션에 통합된 개발 환경 제공

스마트폰 보급과 함께 모바일 오피스에 대한 관심이 커지면서 기업들의 모바일 업무 혁신에 대한 기대 또한 높아지고 있다. 금융, 공공, 유통, 교육, 제조 등 우리나라 전 산업에 걸쳐 모바일 업무시스템 환경을 보다 신속하고 유연하게 구축하여 비즈니스 혁신을 이루고자 하고 있다. 기업들이 모바일 업무 혁신을 빠르게 이룰 수 있도록 도움을 주는 모바일 엔터프라이즈 애플리케이션인 오라클 ADF 모바일은 ADF(Application Development Framework), 자바 테크놀로지, HTML5, CSS3, 자바스크립트 등의 표준기술을 이용하여 한 번의 개발로 안드로이드와 iOS에 하이브리드 모바일 애플리케이션으로 배포가 가능한 혁신적인 솔루션이다.

저자 - 정보람 컨설턴트, 한국오라클 APAC Channel Enablement(boram.jeong@oracle.com)



<그림1> Oracle ADF - 멀티 채널 프레임워크

기존에 각 사용자 기기에 따라 애플리케이션을 개발해야 했던 번거로움을 없애고 같은 코드와 개발 방법을 사용하여 모바일 애플리케이션 개발을 가능하게 해주는 ADF 모바일과 오라클 JDeveloper를 이용하여 안드로이드와 iOS 애플리케이션을 만드는 방법을 살펴보고자 한다.

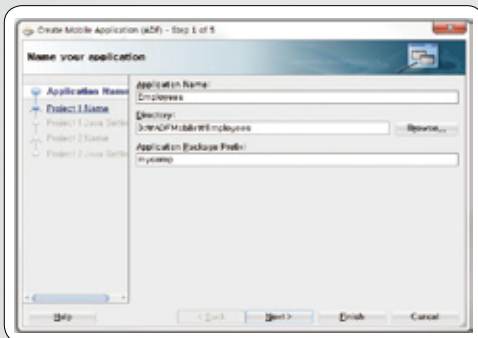
Oracle ADF Mobile을 이용한 Android & iOS 모바일 애플리케이션 생성

모바일 애플리케이션을 생성하기 위해서는 JDeveloper 11.1.1.2.3 버전과 ADF Mobile Extensions와 안드로이드 애플리케이션 개발에 필요한 Android SDK 버전 4.03 설치되어 있어야 한다.

STEP1. 애플리케이션 생성하기

모바일 애플리케이션을 생성하기 위한 방법은 엔터프라이즈 애플리케이션 생성 방법과 유사하다. 간단하게 마법사를 통해 “Mobile Application(ADF)”를 생성한다.

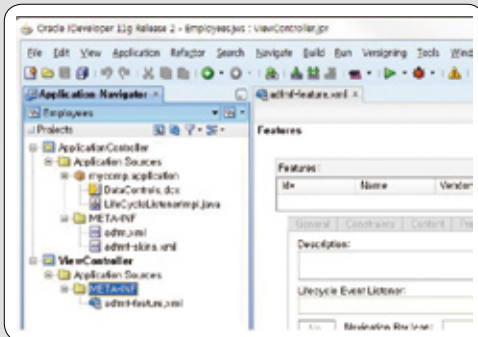
❶ JDeveloper 11.1.2.3을 이용하여 새로운 모바일 애플리케이션을 생성한다. 파일에서 New Application 클릭, General 클릭, Applications 클릭, Mobile Application (ADF) 클릭 한 후 OK를 선택하면 된다.



<그림2> 마법사를 통해 모바일 애플리케이션 생성

❷ 모바일 애플리케이션 생성 마법사가 나타나면, <그림2>처럼 입력한 후 Next를 클릭한다. 총 다섯 단계로 구성되어 있으며, 두 번째 단계부터는 기본 설정값 그대로 Next 버튼을 클릭하여 완료한다.

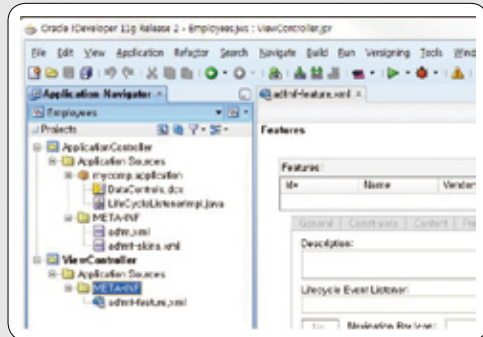
❸ 마법사가 완료되면 Employees라는 모바일 애플리케이션이 생성되고 프로젝트에 “ApplicationController”와 “ViewController”가 생성된 것을 확인할 수 있다.



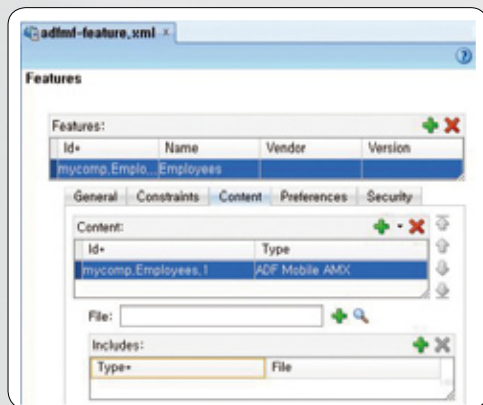
<그림3> 모바일 애플리케이션 기본 화면

STEP2. 애플리케이션에 Feature 생성하기

일반적으로 ADF Mobile 애플리케이션을 배포할 때, 모바일 디바이스에는 하나의 애플리케이션으로 등록되어 사용된다. 하지만 하나의 애플리케이션 안에서는 직원 검색하는 페이지로 이루어진 부분, 도움말과 같이 html 문서로 되어있는 부분 등과 같이 여러 개의 다른 파트로 구성이 가능하고 이는 모두 features에서 관리한다.



<그림 4> ViewController 폴더 구조



<그림 5> 모바일 애플리케이션에 피처(Feature) 등록 화면

❶ ViewController 프로젝트 안에 Application Sources를 확장해 보면, META-INF라는 폴더가 있고 그 안에 adfmf-feature.xml 이라는 파일을 볼 수 있다.

❷ Feature 옆에 있는 + 버튼을 클릭하여 Employees라는 새로운 feature를 등록할 수 있다.

❸ ADF Mobile에서는 UI 콘텐츠를 생성할 수 있는 3가지 타입이 존재한다. 첫 번째 타입은 ADF Mobile AMX이다. 이는 AMX 컴포넌트를 이용하여 Drag & Drop 방식으로 데이터와 바인딩 할 수 있다. 기존에 엔터프라이즈 애플리케이션에 사용한 Java Server Faces 컴포넌트와 유사하다. ADF Mobile AMX 파일은 디바이스에서 애플리케이션을 실행하는 시점에서 HTML이나 자바스크립트로 변환된다. 두 번째 타입은 Remote URL이다. Remote URL을 이용하여 서버 기반의 웹 애플리케이션을 ADF 모바일에 추가할 수 있다. 이러한 웹 애플리케이션은 폰갭

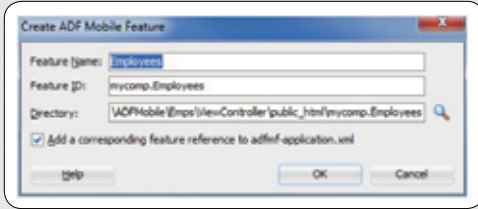
(PhoneGap)에서 제공하는 자바스크립트 API를 통해 디바이스 서비스(카메라, 주소록, 캘린더 등등)를 이용할 수 있다. 마지막으로 제공되는 타입은 Local HTML이다. HTML5를 이용하여 Local HTML 페이지를 생성할 수 있다.

STEP3. ADF Task Flow 생성

하나의 큰 JSF 페이지 플로우로 애플리케이션의 흐름을 보여줄 수도 있지만 오라클 ADF에서는 재사용 가능한 태스크 플로우로 나타낸다. 각 태스크 플로우는 페이지 간에 인터랙션을 표현하거나 애플리케이션 로직을 실행하기도 하고, 다른 태스크 플로우를 호출하는 등의 논리적 오퍼레이션을 수행하는 액티비티 노드로 구성되어 있다. 각 액티비티 간에 트랜잭션은 컨트롤 플로우 케이스를 통해 수행된다.

- ❶ Content 탭을 선택한 후 + 버튼을 클릭하고 Task flow를 EmpsTaskFlow.xml 이름으로 생성한다.
- ❷ 생성된 EmpsTaskFlow.xml View와 Control Flow Case 컴포넌트를 이용하여 <그림 6>처럼 만들어 준





<그림 6> 새로운 피처 등록

다. 이때 showGraph 컨트롤 플로우 케이스 속성은 filpRight로 지정하여 모바일 상에서 오른쪽으로 넘길 경우 페이지간의 이동 이벤트가 발생되도록 한다.

STEP4. 자바 클래스와 데이터 컨트롤 생성하기

오라클 ADF는 개발 생산성을 극대화하는 Java EE 기반의 MVC 프레임워크다. ADF는 데이터를 관리하기 위한 모델 레이어를 제공하기 때문에 UI는 어디에 어떻게 데이터가 저장되는지 알 필요 없이 데이터를 사용할 수 있다. 이것을 데이터 컨트롤 레이어라고 부른다. ADF는 데이터베이스, 웹 서비스, 메인 프레임, 애플리케이션 등과 같은 다양한 데이터 소스로부터 데이터 컨트롤을 생성할 수 있다. ADF 모바일의 경우 Managed bean을 이용하여 데이터 컨트롤을 생성하게 되는데, 일단 데이터 컨트롤을 생성하면 Drag & Drop 방식으로 원하는 UI 컴포넌트와 데이터를 바인딩 할 수 있다.

```

import java.util.Date;

public class Emp {
    public Emp() {
        super();
    }

    // 다음 코드를 추가한다.
    private String name;
    private String email;
    private int salary;
    private Date hireDate;

    public Emp(String name, String email) {
        super();
        this.name = name;
        this.email = email;
    }

    public Emp(String name, String email, int salary, Date hireDate) {
        super();
        this.name = name;
        this.email = email;
        this.salary = salary;
        this.hireDate = hireDate;
    }
}
    
```

<그림 7> Emp.java 파일 생성

1 ViewController에서 New를 클릭한 후 General 카테고리에 있는 Java Class를 선택한다. 아래와 같이 코드를 작성한 후에 Generate accessors를 이용하면 필요한 코드가 자동으로 생성된다.

2 Emps라는 자바 파일도 위와 같은 방법으로 생성하고 <그림 8>과 같이 입력한다.

```

package orcomp.mobile;

import java.util.*;

public class Emps {
    private List<Emp> emp;

    public Emps() {
        super();
    }

    public Emps() {
        // This method gets a list of the employees and their email
        emp = new ArrayList();
        Emp e;
        e = new Emp("Scott", "scott@oracle.com", 4000, getDate(2011, 3, 24, 9, 0));
        emp.add(e);
        e = new Emp("Zoe", "zoe@oracle.com", 3000, getDate(2007, 2, 24, 9, 0));
        emp.add(e);
        e = new Emp("John", "john@oracle.com", 5000, getDate(2005, 2, 18, 9, 0));
        emp.add(e);
        e = new Emp("Shelley", "shelley@oracle.com", 4000, getDate(2005, 2, 18, 9, 0));
        emp.add(e);
        emp = (List<Emp>) emp.toArray(new Emp[emp.size()]);
        return emp;
    }

    private Date getDate(int y, int M, int d, int H, int M) {
        Calendar cal = Calendar.getInstance();
        cal.set(y, M, d, H, M);
        Date date = cal.getTime();
        return date;
    }
}
    
```

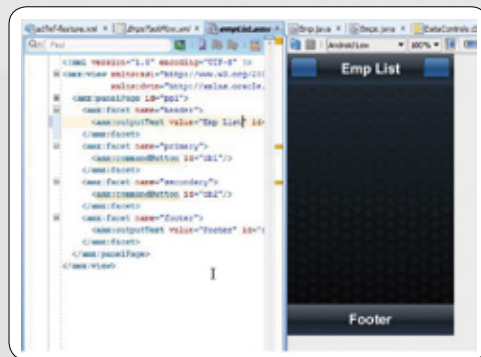
<그림 8> Emps.java 파일 생성

④ 완성된 자바파일을 이용해 데이터 컨트롤을 생성하기 위해서는 자바 파일 선택한 후에 마우스 우클릭을 이용하여 Create Data Control을 선택한다.

STEP5. UI 생성하고 테스트해보기

태스크 플로우에서 생성한 View 컴포넌트는 실제 페이지가 아니다. 어떤 페이지를 보여줄지에 대한 참조 정보를 가지고 있기 때문에 페이지를 생성해 주어야 한다.

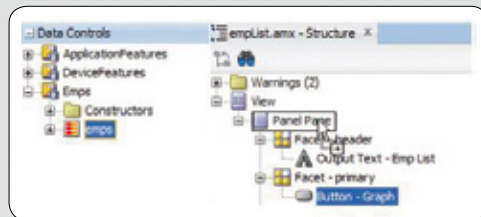
① EmpsTaskFlow.xml에서 empList를 더블클릭 하면 ADF Mobile AMX 페이지를 생성할 수 있는 창이 나



<그림 9> 모바일 페이지 - empList.amx 화면

타난다. 원하는 속성을 선택한 후에 OK를 클릭한다.

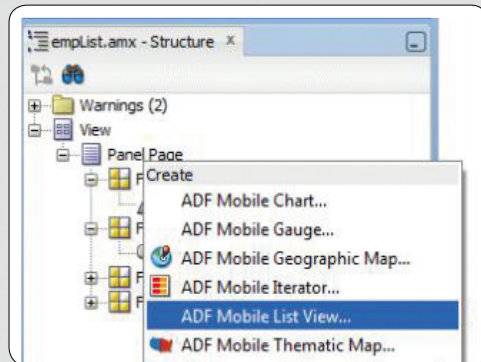
② <그림9>처럼 empList.amx로 만들어지며, Preview를 통해 내가 스스로 작성한 부분들이 어떻게 모바일 화면에서 보여질지 확인할 수가 있다. Header의 text value를 Emp List로 변경한다.



<그림 10> empList.amx의 구조(Structure)

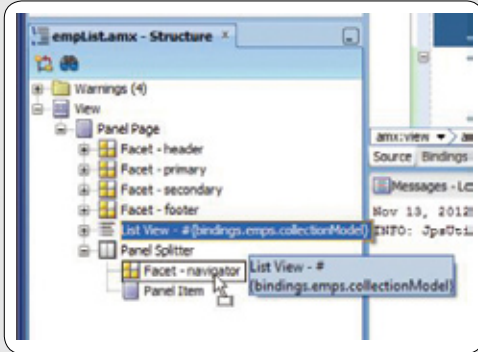
③ Primary 버튼 text를 Graph로 변경하고 이 버튼을 누르면 그래프로 정보를 볼 수 있는 페이지로 연결 될 수 있도록 Action showGraph를 선택한다.

④ STEP4에서 자바 클래스를 이용해 생성한 데이터 컨트롤 패널을 확장해보면 <그림10>처럼 Emps라는 데이터 컨트롤이 생성된 것을 확인할 수가 있다. Emps를 Drag하여 empList.amx의 Panel Page에 Drop한다.

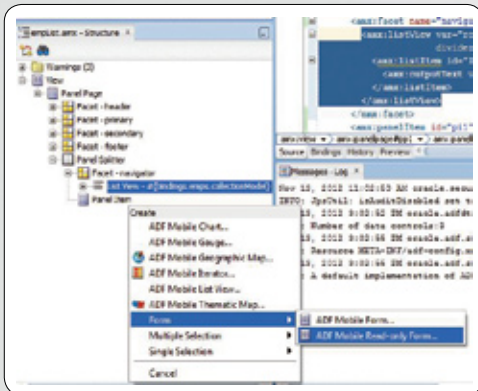


<그림 11> 모바일 애플리케이션에서 사용 가능한 ADF Mobile Faces Rich Client 컴포넌트

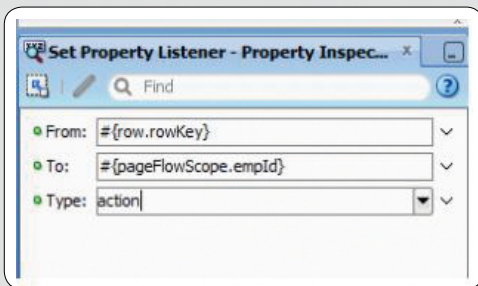
⑤ 그러면 <그림11>처럼 생성된 데이터를 다양한 UI 컴포넌트 형태로 생성할 수가 있다. 이번 랩에서는 List View 형태를 만들 것이므로 ADF Mobile List View를 클릭한다.



<그림 12> List View 컴포넌트 이동



<그림 13> 읽기 전용 폼(Form) 형태로 데이터 바인딩



<그림 14> Action Binding Property 설정

6 List View Gallery가 나타나면 사용자가 원하는 형태의 list를 <그림12>처럼 선택할 수가 있다.

그 다음에는 Edit 창이 나타나게 되는데 선택된 List View를 Divider Mode를 First Letter로 지정해주면 알파벳 이름으로 정렬된다.

7 Panel Splitter를 이용하여 <그림12>와 같은 구조로 만들어 준다.

8 Panel Item 안에는 데이터 컨트롤에 있는 Emps를 Drag하여 Form - ADF Mobile Read-only Form 형태로 생성한다. Form에 한글로 표시되길 원할 경우 Display Label을 수정해 주면 된다.

<그림13>

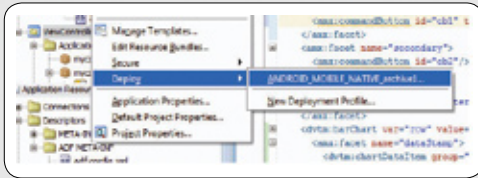
9 Panel splitter 안에 있는 List Item에 Set Property Listener를 추가해 준다.

<그림14>

10 데이터 컨트롤을 생성하게 되면 많이 사용되는 기능들이 자동으로 생성된다. setCurrentRowWithKey를 ADF Mobile Button UI 컴포넌트와 바인딩 해준다.

11 Edit Action Binding 창이 나타나면 RowKey 값을 EL로 다음과 같이 표현해 준다.

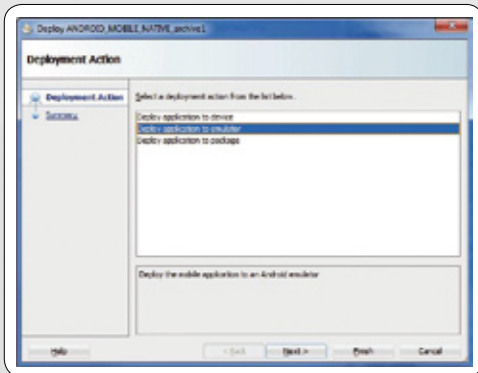
`{pageFlowScope.empId}`



<그림 18> ADF 모바일 애플리케이션 배포

20 한 번 Deployment Profile을 설정하면 Application -> Deploy에 사용자가 조금 전에 생성한 프로파일을 확인 할 수가 있다. Application -> Deploy -> ANDROID_MOBILE_NATIVE_archive1를 선택한다.

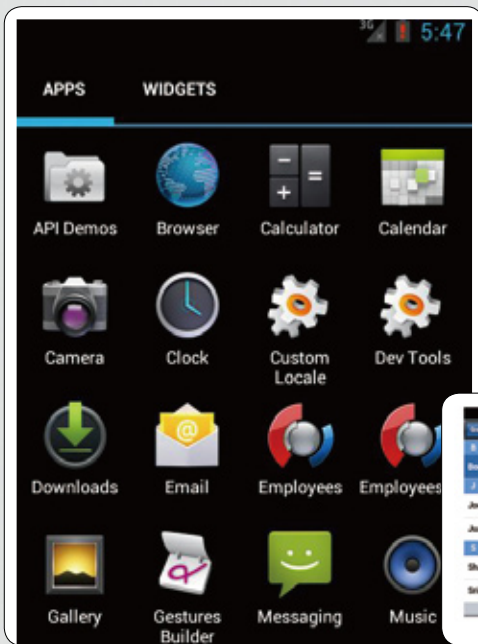
<그림18>



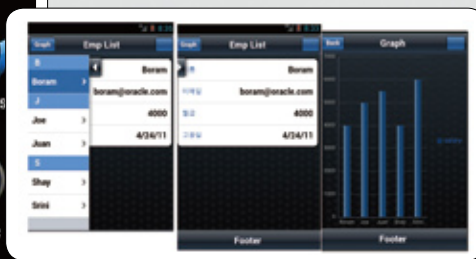
<그림 19> Deployment Actions 선택

21 Deploy Action에서 Deploy application to emulator를 선택한다. <그림19>

22 Emulator에 배포된 Employees를 확인하고 클릭한다. <그림20><그림21>



<그림 20> Android Emulator에 등록된 Employees 모바일 애플리케이션



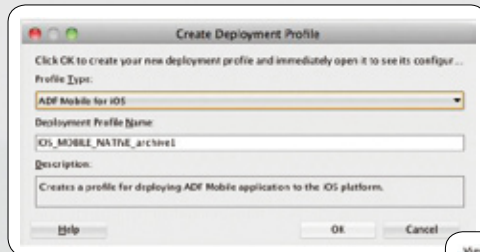
<그림 21> 완성된 모바일 화면

STEP6. iOS 애플리케이션으로 생성하기

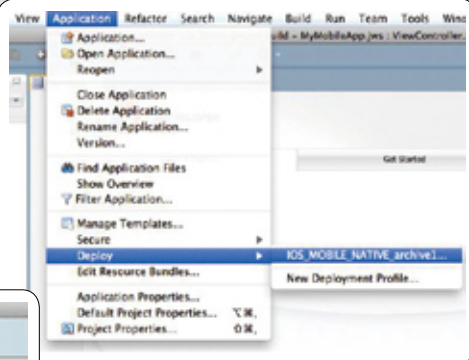
완성된 안드로이드 애플리케이션을 코드 수정 없이 iOS 애플리케이션으로 배포 할 수 있다.

- ① 애플리케이션에서 새로운 Deployment Profile을 생성한다. <그림22>
- ② 디플로이어에서 방금 생성한 Deployment Profile을 선택한다. <그림23>

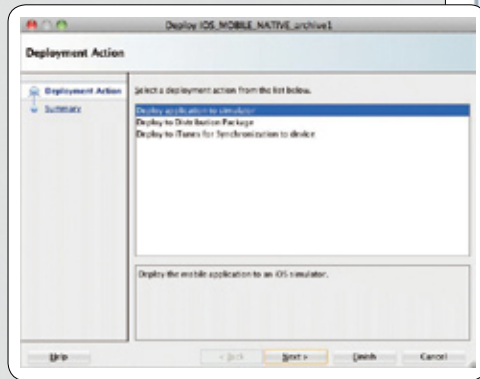
③ Deploy application to simulator를 선택하면 iOS 애플리케이션을 확인할 수 있다. <그림24>



<그림 22> 완성된 모바일 애플리케이션을 iOS로 배포하기 위한 deployment profile 설정



<그림 23> 새로 설정된 deployment profile로 배포



<그림 24> Deployment Actions 선택

오라클 ADF Mobile에 대한 더 많은 정보는 아래 한국 ADF 커뮤니티에서 찾아볼 수 있다.
 참고: 한국오라클 ADF 커뮤니티 <http://adf.oracle-pdm.com/xe>