

WEM 에서 네이티브 모바일 앱 프로젝트의 특성.

WEM 은 API Level 21 이상의 Android 를 지원합니다. Android Lollipop 5.0 이 가장 낮습니다.

표준 웹 포털과 네이티브 앱 간의 차이점을 이해합니다.

- 웹포털은 고가용성, 많은 디스크 공간, CPU 성능 및 인터넷 대역폭을 갖춘 무거운 서버의 클라우드 단일 위치에서 실행되는 중앙 집중식입니다.
- 네이티브 앱은 여러 알 수 없는 위치, 제한된 디스크 공간, 제한된 CPU 전원 및 제한된 또는 전혀 연결되지 않은 작은 개인 장치에서 실행됩니다.
- 웹 포털은 많은 사용자에게 서비스를 제공하는 하나의 웹사이트이므로(일반적으로) 특정 사용자는 어떤 방식으로든 인증될 수 있으며, 반드시 필요합니다(기본 인증, ADFS 또는 OAuth2 를 사용한 Single Sign-On).
- 네이티브 앱은 개인 장치에서 실행되고 있으며, 이는 그 자체로 신뢰할 수 있는 단일 사용자 상황으로 간주될 수 있습니다. 사용자를 인증하는 다른 방법은 온라인/외부 인증 공급자를 위한 일회성 인증과 같이, 선택적으로 제한된 유효성 기간이 있는 특정 키가 장치에 저장되므로 다음에 앱이 시작될 때 장치 및 로컬 저장소가 신뢰할 수 있기 때문에 인증을 건너뛸 수 있습니다.
- 웹 포털은 WEM 에서 게시할 수 있으며 최종 사용자가 즉시 사용할 수 있습니다.
- 네이티브 앱은 WEM 에서 생성될 수 있지만 배포(앱을 장치에 가져오는 것)는 여전히 수동 작업이 필요한 별도의 프로세스입니다. WEM 에서 생성되는 결과는 설치 관리자 파일과 동일한 네이티브 모바일입니다. 파일을 단일 장치로 가져오거나 Android 및 iOS 에 따라 여러 장치에 배포하는 방법에는 여러 가지가 있습니다.
- 웹 포털에는 인터넷 연결이 있습니다. 현재 WEM 에 있는 많은 위젯(예: 차트 및 Google 지도)은 웹 및 연결을 염두에 두고 만들어집니다. 이러한 위젯 중 상당수는 네이티브 모바일 장치에서 작동하지 않거나 데이터 연결이 가능한 경우에만 작동할 수 있습니다.

네이티브 모바일 장치에서는 작동하지 않는 WEM 기능:

- OData 는 작동하지 않으며 모바일에서 지원되지 않으며 모바일 장치에서 앱 간에 데이터를 교환하는 유일한 방법은 REST 또는 SOAP 서비스를 사용하는 것입니다.
- 기업 SSO(Single Sign-On)용 ADFS 와 같은 SAML 기반 인증 공급자는 (쉽게) 작동하지 않습니다. 연결 가능하고 알려진 대상에서 신뢰할 수 있는 끝점이 필요하기 때문입니다. 모바일 장치는 그렇지 않습니다. Google, Facebook 및 기타 개인 인증자를 사용하는 OAuth 사용할 수 있습니다.
- 실시간 메시징을 위한 Comet 은 작동하지 않습니다(연결 및 알려진 끝점 필요). 네이티브 모바일에는 Firebase 메시징이 필요합니다. WEM 에서는 아직 지원이 제공되지 않습니다.
- 데이터 연결이 필요한 위젯(예: 차트, Google 지도, ReCaptcha)은 앱 자체의 일부가 아닌 외부 리소스(예: 온라인에서 사용할 수 있는 Google 자바스크립트 라이브러리의 지도 데이터)가 필요하기 때문에 데이터 연결이 가능한 경우에만 작동합니다. 특정 위젯은 웹을 염두에 두고 개발될 수 있으므로 사용하는 위젯에 주의하십시오. WEM Modeler 는 특정 위젯이 지원하는 플랫폼(웹, 안드로이드, iOS)에 대한 정보를 추가하는 수단을 제공합니다. 이것이 실제로 어떻게 작동할지 아직 명확하지 않습니다...
- 수신 전화, 요청, 서비스가 작동하지 않습니다(모바일 장치의 주소, 위치 및 연결을 알 수 없기 때문에). 따라서 모바일 앱은 웹 서비스를 사용하고 HTTP 요청을 사용하는 기능만 할 수 있습니다.
- 참조가 많은 매우 깊이 정규화된 데이터 구조는 피해야 합니다.
 - SQL Server 에서는 웹 포털을 백업하고 참조가 있는 정규화된 데이터 구조는 문제가 되지 않으며, 데이터 스토어는 SQL 조인 구조 및 최적화 기능을 사용하고 고급 데이터베이스 서버의 힘에 의존할 수 있음을 쿼리합니다.
 - 로컬 장치 저장소에는 결합 및 쿼리 최적화를 위한 SQL 기능이 없으므로 모바일 장치의 기능을 대상으로 하는 데이터를 가능한 한 "평평하게" 유지하는 것이 좋습니다. 이 경우 데이터를 비정규화해야 합니다(중복될 수 있음) 쿼리를 더 빠르게 만들기 위해 특정 WEM 모델 및 흐름에서 지원합니다.
- 장치 파일 저장소의 로컬 파일: 제한적이지만 예, 로컬 파일에 액세스 할 수 있습니다 (파일 입력을 사용할 때 일부 폴더의 사진 및 문서와 같이 공용 로컬 파일을 선택할 수 있습니다).

모바일 앱 프로젝트를 작게 유지

WEM의 프로젝트에는 모든 종류의 자산이 포함되어 있습니다. 순서도, 템플릿, 온톨로지 항목, 파일 라이브러리의 항목, 아이콘 등. 이러한 모든 자산은 네이티브 모바일 앱으로 생성된 파일에 포함됩니다.

모바일 프로젝트를 가능한 한 깨끗하게 유지하고 앱에 필요하지 않은 모든 자산을 제거하십시오.

또한 모바일 앱을 별도의 WEM 프로젝트로 사용하는 이유 중 하나는 네이티브 모바일 앱에서 사용되지 않는 많은 자산이 포함된 많은 웹 포털이 포함된 대규모 프로젝트의 일부가 아닙니다(현재 프로젝트 수준에서 엄격한 분리가 있습니다) 프로젝트를 만들 때 웹 또는 모바일 프로젝트가 될지 여부를 선택해야 합니다.

한 프로젝트에서 모든 종류의 포털 유형(웹 및 모바일 모두)을 갖도록 경로를 따라 가면 결과 파일 크기의 영향을 알고 있어야 합니다.

마지막으로 Android 용 APK 파일이 있는 경우 파일을 로컬로 변경하여 파일을 .zip 파일로 변경하여 해당 내용을 추출하여 파일의 규모가 얼마나 크고 특정 리소스가 얼마나 되는지 확인할 수 있습니다. 추출된 폴더에는 폴더 Asset 이 포함되어 있으며 포털 폴더가 있습니다. 이 폴더에는 모든 특정 WEM 프로젝트/포털 리소스가 포함되어 있습니다 (APK의 다른 모든 리소스는 다른 모든 빌드에 대해 동일하게 고정된 것으로 간주해야 합니다).

데이터 동기화

일반적인 상황은 다음과 같은 상황이 될 수 있습니다.

- 사용자 관리, 인증, 데이터 관리, 관리자 기능을 포함한 모든 기능에 대한 웹 포털/프로젝트가 있습니다.
- 기본적으로 모든 관리 및 관리자 기능 없이 웹의 최종 사용자 기능의 일부가 있는 네이티브 모바일 앱을 갖고 싶어합니다. 이 응용 프로그램은 오프라인으로 작동할 수 있어야 하지만 여전히 필요한 데이터가 있어야 합니다. 장치에서 변경한 일부(전부는 아님)도 중앙 집중식 웹에서 사용할 수 있어야 합니다.

이 작업을 수행하려면 웹 포털(SOAP 및/또는 REST)에서 웹 서비스를 노출하고 네이티브 앱에서 데이터를 소비하여 데이터 동기화 기능을 만들어야 합니다.

앱이 지원해야 하는 데이터와 기능에 따라 버전 및 마지막으로 업데이트된 타임스탬프에 대한 추가 필드를 추가하고, 어떤 데이터 스토어가 선호하고 있는지(일반적으로 응용 프로그램

데이터에 대한 중앙 웹 및 앱 작업 시 개인 데이터 및 환경 설정에 대한 로컬 장치)를 결정하고 동기화할 방법을 결정해야 합니다.

장치의 모든 응용 프로그램 데이터를 지우고 웹의 모든 데이터로 대체하는 것이 더 쉽습니다 (사용자가 완전한 업데이트를 강제로 요청하거나 장치에서 아직 사용할 수 없는 웹에서 사용할 수 있는 변경 사항이 있음을 확인할 수 있는 경우에만 이를 수행하십시오.)

최상의 시나리오를 해결하고 여러 업데이트 호출을 사용합니다. 작은 데이터 집합은 REST 를 사용하여 가장 쉽게 전송할 수 있습니다(JSON 가져오기/내보내기 노드를 사용하여 WEM 에서 모델링하는 것이 가장 쉽습니다). 큰 데이터 집합은 SOAP 를 사용하여 전송할 수 있으며, 이는 Modeler 에서 더 많은 작업이지만 진행률 막대를 표시하여 처리된 데이터의 양을 표시하고 필요한 경우 일부 데이터 조작 및 변환을 수행하는 등 간에 몇 가지 추가 작업을 수행할 수 있는 옵션을 제공합니다(그러나 명심하십시오. 기기 소비자에게는 가능한 한 적게, 웹 공급자에게는 가능한 한 많이)

마법 같은 자동 동기화는 없으며 웹 포털(또는 다른 소스)과 기본 모바일 앱 간의 모든 데이터 교환은 WEM 에서 사용자 정의해야 한다는 점을 이해하십시오.

유용할 수 있는 몇 가지 하이라이트

WEM Keywords

<https://my.wem.io/forum?threadid=145> about useful keywords for mobile:

- ConnectivityTypes
- Platform
- BatteryLife
- DeviceId
- DeviceName

사용자에게 데이터 연결(기본 모바일 장치에서는 사용할 수 없음)이 있는지 확인하려면 다음을 확인할 수 있습니다:

ConnectivityTypes contains "wifi" or

ConnectivityTypes contains "mobile" or

ConnectivityTypes contains "ethernet"

그리고 이것을 계산된 필드에서 사용하십시오.

UI

- 사용 가능한 모바일 Android 또는 모바일 iOS 마스터 템플릿을 사용하여 기본 모양과 느낌에 가깝게 만듭니다.
- 오버레이 사용을 제한하고 대부분 전체 페이지를 사용하며 응답성을 위해 사용 가능한 옵션(템플릿 편집기에서 xs 및 sm 을 사용하여 확인)을 사용합니다.
- 특히 앱 설정, 아이콘, 스플래시 화면에서 아이콘, 로고 등에 최대한 png 이미지를 사용하십시오.
- 메뉴에서 탐색 항목의 사용을 제한합니다.

데이터 동기화

기본 모바일 앱은 사용 가능한 데이터 연결이 없을 때 유용할 수 있는 로컬 데이터 저장소를 사용할 수 있습니다. 기기에 새로 설치된 앱에는 데이터가 포함되어 있지 않으므로 앱에서 원하는 작업과 수행해야 하는 작업에 따라 흐름에서 관리해야 합니다.

데이터 연결이 가능할 때 웹 서비스를 사용하십시오:

- REST/json 은 가장 낮은 오버헤드와 풋프린트로 가장 빠릅니다.
- SOAP/Xml 은 데이터 양이 많고 진행 정보를 사용하여 일괄 처리해야 하는 경우 더 유용할 수 있습니다.
- 데이터 소스인 WEM 웹 포털에서 모바일 전용 웹 서비스를 생성하여 클라이언트 풋프린트를 최대한 작게 유지합니다. WEM 에 노출된 모든 서비스 방법은 SOAP 및 REST 를 통해 사용할 수 있으며 앱에서 가장 적합하다고 생각하는 대로 선택할 수 있습니다. (둘 모두 사용 시, SOAP 를 사용하는 하나의 흐름이 실패하면 REST 로 다른 시도가 완료되었습니다...)
- 마지막 수정 날짜와 마지막 동기화 날짜를 사용하여 로컬 데이터의 일부를 동기화해야 하는지 여부를 간단히 확인합니다. 사용 가능한 데이터 연결이 있는 경우 모바일 앱 시작 시 이 확인을 수행한 다음 데이터를 동기화해야 하는 경우 사용자 정보를 표시합니다(테스트에서 시간이 걸린다는 것을 알게 되면 사용자가 동기화를 수행하도록 결정).
- 오프라인 상태에서 데이터에 액세스할 필요가 없고 테스트에서 웹서비스 과 활성 데이터 연결을 통한 직접 사용이 가능한 것으로 확인되면 이를 사용하십시오.
- 사용자 데이터가 저장되고 사용자 계정에 연결되면 사용자가 장치뿐만 아니라 웹 포털에서도 로컬 데이터와 프로필을 완전히 지울 수 있는 방법을 제공합니다(장치를 분실한 경우). 이를 위해 DeviceID, 선택적으로 유효한 기간이 있는 특정 토큰 값을 사용할 수 있으며, 사용자가 지울

수 있는 데이터 원본 웹 포털의 사용자 계정과 함께 저장할 수 있습니다. 이 DeviceId 및 Token 은 여전히 유효합니다. 또는 사용자가 여러 장치를 연결할 수 있는 경우 장치 및 토큰 목록이 있습니다.

애플 개발 iOS 앱 배포.

개발자 포털에서 인증서 및 프로비저닝 프로파일 생성: <https://developer.apple.com/>

이 포털에 대한 링크는 iOS 프로비저닝 및 인증서 오버레이에서도 사용할 수 있습니다(포털 설정을 통해 액세스 가능).

iOS 앱의 배포를 위해 Apple 과 개발자 계정이 적어도 있어야 하며, 앱 ID 를 만들고, 인증서를 만들고, 해당 앱 아이디어에 연결할 수 있는 프로파일을 프로비저닝하고, 테스트 목적으로 앱을 설치할 수 있는 소유 및 신뢰할 수 있는 장치를 추가해야 합니다.

이 모든 부분(앱 ID, 인증서, 프로비저닝 프로파일, 연결된 장치)은 모두 연결되어 있으며 Apple 개발자 환경에서 올바르게 설정해야 합니다.

WEM 에서 App ID 는 Apple 개발자 환경에서 정의한 것과 동일해야 합니다.

WEM 에서는 인증서 요청을 시작해야 하고(적절한 인증서가 아직 없는 경우) 인증서 생성 시 요청을 Apple 에 업로드해야 합니다. 다양한 유형의 인증서가 있으며 적어도 하나는 개발용으로, 다른 하나는 배포용으로 필요합니다.

그런 다음 프로비저닝 프로파일을 만들 수 있습니다. 하나는 개발을 위한 것이고 하나는 배포용입니다.

개발 프로비저닝 프로파일은 개발 계정에 연결된 하나 또는 몇 개의 기존 iOS 장치를 연결할 수 있는 곳입니다. 이러한 장치는 WEM iOS 빌드 오버레이에서 OTA 를 통해 앱을 테스트하는 데 사용할 수 있습니다.

배포 프로비저닝 프로파일은 공개 Apple App Store(Apple 의 승인 후)에 대해 프로비저닝 하거나 제한된 등록된 장치 집합을 사용하는 경우 프로비저닝할 수 있는 위치입니다.

몇 가지 링크:

<https://support.magplus.com/hc/en-us/articles/204270188-iOS-Creating-an-Ad-Hoc-DistributionProvisioning-Profile>

<https://developer.apple.com/programs/>

Apple 개발자 계정 만들기, 개인의 (또는 회사) AppleID 와 함께, 애플 개발자 프로그램에 가입.

개발자 포털에서 인증서 및 프로비전 프로파일 만들기: <https://developer.apple.com/>

여기에서 WEM iOS 프로젝트에서도 사용되는 bundleid 를 정의합니다. 이들은 bundleid 와 연결된 인증서 및 프로비저닝 프로파일에서 일치해야 합니다.

와일드카드가 가능합니다. 동일한 프로비저닝 프로파일 및 인증서 내에서 더 많은 앱을 가질 계획이라면, io.mycompanyname.myiOSapps 와 같은 공통 시작으로 번들 ID 를 정의하기만 하면 됩니다.* 이러한 인증서를 사용하고 프로파일을 프로비저닝 합니다(개발자 포털에서 다운로드하고 iOS 앱 프로젝트와 함께 WEM 으로 가져오기).

올바른 번들 ID 를 사용하는지 확인합니다(또는 와일드카드 번들과 함께 적절한 접두사를 사용하여 이 번들에 대해 인증서 및 프로비저닝이 유효함).

WEM 에서 앱 빌드(게시와 유사하지만 약간 다름, 현재 WEM v4 의 설정 페이지에 있는 설치 프로그램 파일 [appname.ipa] 생성)

WEM OTA 옵션을 사용하여 앱 개발자 계정(developer.apple.com 포털)에 연결된 iOS 테스트 장치에 로드 합니다.

앱 스토어 커넥트 문서:

앱스토어 연결 포털에서 필요한 모든 계약, 연락처, 은행 및 세금 정보를 관리합니다.

<https://help.apple.com/app-store-connect>

<https://appstoreconnect.apple.com/>

AppStore Connect 환경에 앱을 추가하고,

<https://help.apple.com/app-store-connect/#/dev2cd126805>

모든 필수 정보를 입력하십시오.

계속해서 앱 세부정보를 입력하세요.

<https://help.apple.com/app-store-connect/#/dev97865727c>

<https://developer.apple.com/business/custom-apps/>

<https://help.apple.com/app-store-connect/#/dev275598f16>

<https://help.apple.com/businessmanager/>

권한 목적 설명

승인을 위해 Apple Store 로 보내기 전에 WEM 은 다음을 수동으로 설정해야 합니다.

카메라 및 위치에 대한 권한의 목적에 관한 정보.

(Modeler 에서 관리하는 페이지 작업 중)

이러한 텍스트는 Apple 에서 면밀히 조사할 것이며 표준 콘텐츠가 아니라 구체적이고 정확해야 합니다.

```
{
  PortalPermissions:[
    {
      PortalId: .....,
      NSCameraUsageDescription: "This app does not use your camera.",
      NSPhotoLibraryUsageDescription: "This app does not access your photo library.",
      NSLocationWhenInUseUsageDescription: "Your location is used to show our organizations in
your vicinity and to calculate the distance to these organizations from your location.",
      NSLocationAlwaysUsageDescription: "Your location is only used when you are actively
looking for organizations in your vicinity."
    }
  ]
}
```