



TF Team

# eXbuilder6 Cordova

## 연동 가이드라인



## 목차

1.	개발환경 준비.....	3
1.1.	필요 도구 요약.....	3
1.2.	도구 설치 상세.....	3
	JDK, NodeJS 및 eXbuilder 6 설치.....	3
	Cordova.....	3
	Gradle.....	3
	안드로이드 커맨드 라인 도구.....	3
2.	Cordova 연동 준비.....	5
2.1.	코도바 프로젝트 준비.....	5
2.2.	첫 UI 프로젝트 연동해 보기.....	7
	코도바 템플릿 프로젝트를 이클립스 워크스페이스에 가져오기.....	7
	eXbuilder6 UI 프로젝트와 연결하기.....	8
2.3.	자동화 빌드 및 빌드 구성.....	10
	CI Tool, Runtime Library Export 방법.....	10
	eXbuilder 6 프로젝트 빌드 방법.....	10
	eXbuilder 런타임 복사.....	10
	안드로이드 애플리케이션 빌드 방법.....	10
	안드로이드 애플리케이션 실행.....	10
3.	테스트 전략.....	11
3.1.	개발 지속 테스트.....	11
3.2.	에뮬레이터 테스트.....	11
3.3.	실기기 테스트.....	11
4.	기타.....	12
4.1.	인증서.....	12

## 개발환경 준비

## 1.1. 필요 도구 요약

항목	역할
JDK8	스튜디오 실행 런타임. 안드로이드 빌드 툴 체인. 그래들 런타임. (안드로이드 애플리케이션 빌드 자동화 플랫폼)
eXbuilder 6	UI 애플리케이션 개발.
Node.JS	Cordova 설치 및 관리 도구.
Gradle	안드로이드 빌드 자동화 플랫폼.
Andorid 커맨드 라인 도구.	안드로이드 앱 빌드 자동화. 실 기기 테스트 및 에뮬레이터 테스트.
Cordova	eXbuilder 6 앱을 안드로이드 앱으로 래핑, 네이티브 API 추가 제공.

## 1.2. 도구 설치 상세

## JDK, NodeJS 및 eXbuilder 6 설치.

이 항목에 대한 설치 가이드는 생략합니다.

## Cordova

셸에서 다음과 같이 입력합니다.

```
$ npm install cordova -g
```

## Gradle

<https://gradle.org/> 에서 지침에 따라 설치합니다. 각 운영체제 별 패키지 매니저를 이용하여 설치하거나 수동으로 설치 할 수 있습니다.

## 안드로이드 커맨드 라인 도구.

<https://developer.android.com/studio/index.html> 페이지 하단의 “get just the command line tools” 항목에서 운영체제에 적합한 도구를 다운로드 받아 압축을 풉니다..

안드로이드 SDK 홈으로 사용할 폴더가 ANDROID\_HOME 이라고 할 경우, 위에서 다운로드 받은 파일압축을 풀어 생성된 tools 폴더를 ANDROID\_HOME 에 넣습니다. 그후 시스템 환경변수 ANDROID\_HOME 으로 ANDROID\_HOME(SDK 홈으로 사용할 폴더)를 가리키도록 합니다.

시스템 환경 변수 PATH 에 다음 항목을 추가합니다. 일부 명령어 도구 이름이 중복되는 것들이 있으므로, 아래의 순서는 중요합니다.

- ANDROID\_HOME/emulator - 이 항목은 아직 존재하지 않지만 추가 해 두십시오.
- ANDROID\_HOME/tools

- ANDROID\_HOME/tools/bin
- ANDROID\_HOME/platform-tools - 이 항목은 아직 존재하지 않지만 추가 해 두십시오.

또한 시스템 환경 변수

ANDROID\_HOME/tools/bin sdkmanager.bat 파일을 텍스트편집기로 열어 sdk\_root 를 추가합니다.

```
@rem Execute sdkmanager
"%JAVA_EXE%" %DEFAULT_JVM_OPTS% %JAVA_OPTS% %SDKMANAGER_OPTS% -classpath
"%CLASSPATH%" com.android.sdklib.tool.sdkmanager.SdkManagerCli %CMD_LINE_ARGS% --
sdk_root=%ANDROID_HOME%
```

안드로이드 명령어 도구를 이용하여 개발에 필수적인 추가 도구를 설치합니다:

```
# 빌드 도구 설치
$ sdkmanager "build-tools;28.0.3"

# 타겟 SDK 설치.
$ sdkmanager "platforms;android-28"
```

에뮬레이터를 통한 테스트를 원할 경우, 아래 항목도 추가 설치:

```
# 필수 플랫폼 도구 설치
$ sdkmanager "platform-tools"

# 에뮬레이터를 위한 안드로이드 이미지 설치
$ sdkmanager "system-images;android-27;google_apis;x86"

# 에뮬레이터 하드웨어 가속 가상화를 위한 도구 설치.
$ sdkmanager "extras;intel;Hardware_Accelerated_Execution_Manager"

# 안드로이드 SDK 폴더로 이동하면 extras/intel 디렉토리가 신규 추가되어 있습니다.
# 해당 디렉토리로 이동 하십시오.
$ cd ANDROID_HOME/extras/intel/Hardware_Accelerated_Execution_Manager

# 유닉스 계열의 운영체제라면 아래와 같이 쉘 스크립트를 실행하고
# 윈도우즈 계열의 경우라면 해당 폴더에 인스톨러가 존재하므로 실행 하십시오.
$ sudo sh silent_install.sh

# 안드로이드 버추얼 디바이스 추가
```

```
$ avdmanager create avd -n test -k "system-images;android-27;google_apis;x86" --abi
google_apis/x86

# window
$ avdmanager -v create avd -n test -k "system-images;android-27;google_apis;x86" -g
"google_apis;x86"
```

## 2. Cordova 연동 준비

### 2.1. 코도바 프로젝트 준비

코도바 프로젝트를 만듭니다.

```
$ cordova create cordova-sample
$ cd cordova-sample
```

config.xml 파일의 내용 중 아래 부분을

```
<content src="index.html" />
```

아래와 같이 수정하여 주십시오:

```
<content src="app/index.html" />
<!-- 필요한 경우, 해당 파일 내 앱 이름과 설명 등을 수정해도 좋습니다 -->
```

런타임으로 안드로이드 및 브라우저를 추가합니다.

```
$ cordova platform add android

# 아래의 명령어로 플랫폼에 브라우저를 추가하면 코도바를 웹 서버 형식으로 실행 해 볼
수 있습니다.
$ cordova platform add browser

# macOS 에 xcode 등이 준비된 경우 아래와 같이 ios 지원도 추가할 수 있습니다.
$ cordova platform add ios
```

필요한 네이티브 API 플러그인들을 설치합니다. 단순히 코도바에서 eXbuilder6 애플리케이션을 실행 해볼 테스트 목적이라면 이 과정을 건너 뛰어도 좋습니다. (단 제공되는 예제를 실행 해 보려면 카메라 플러그인은 설치해야 합니다.)

예시:

```
$ cordova plugin add cordova-plugin-camera #사진촬영
$ cordova plugin add cordova-plugin-file #파일 저장 접근
```

```
$ cordova plugin add cordova-plugin-geolocation #위치 정보 접근  
$ cordova plugin add cordova-sqlite-storage #내부 데이터 베이스 사용  
$ cordova plugin add ionic-plugin-keyboard #키보드가 input 을 가리지 않도록, 자동 스크롤
```

설치 후 “cordova plugin list” 를 통해 설치된 plugin 들을 확인할 수 있습니다.

cordova-sample/www 폴더의 내용을 비웁니다. 이 내용은 이후 설명될 자동화 스크립트 또는 빌드 미러링 기능에 의해 자동으로 채워 넣어지게 됩니다.

코도바 프로젝트 준비(이하 템플릿 프로젝트)가 완료되면, 이 템플릿 프로젝트를 별도 보관하거나, 형상 관리에 포함시켜 둡니다.

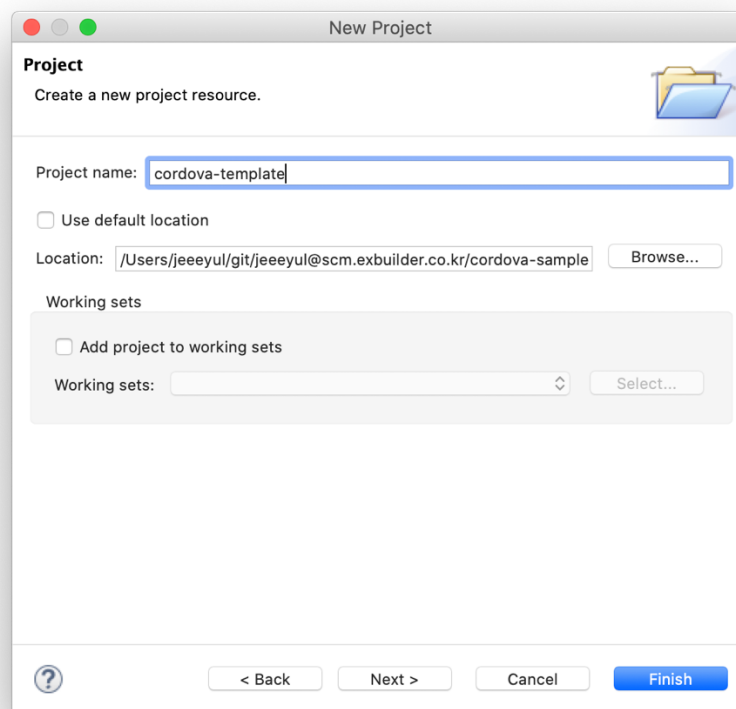
1. 이때 node\_modules 디렉토리는 개발자의 환경에 따라 다르므로 형상관리 대상에서 제외 시켜 주십시오. (.gitignore 등).
2. www/ 디렉토리의 내용은 exbuilder6 프로젝트에 의해 자동 생성 될 것이므로 형상관리 대상에서 제외 시켜 주십시오.

## 2.2. 첫 UI 프로젝트 연동해 보기

### 코도바 템플릿 프로젝트를 이클립스 워크스페이스에 가져오기

eXbuilder6 프로젝트와 코도바 템플릿 프로젝트를 연동시키기 위해서는 먼저 해당 템플릿 프로젝트를 워크스페이스에 연결 시켜야 합니다.

1. eXbuilder6 스튜디오의 메뉴의 File > New > Others... 를 선택하십시오.
2. General 섹션의 Project 항목을 선택 하십시오.
3. “Project name” 필드에 “cordova-template” 이라고 입력하십시오.
4. “[x] Use Default Location 필드”의 체크박스를 해제하고,  
앞서 만든 cordova-sample 프로젝트의 경로를 지정하십시오.



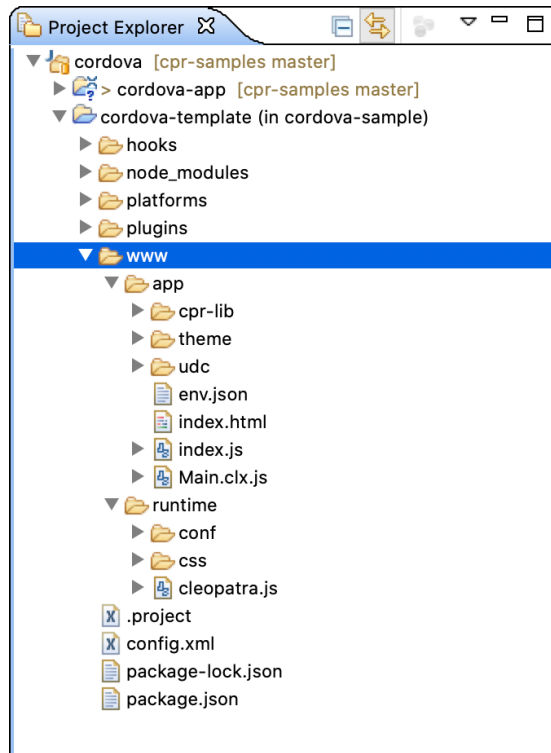
5. 생성된 템플릿 프로젝트 내 www 디렉토리에 다음 두개의 디렉토리를 만드십시오:
  - A. “app” – 이 폴더에는 eXbuilder6 UI 프로젝트의 빌드 결과물이 탑재될 것입니다.
  - B. “runtime” – 이 폴더에는 eXbuilder6 런타임 라이브러리가 탑재될 것입니다.

## eXbuilder6 UI 프로젝트와 연결하기

1. 샘플 UI 프로젝트 만들기
  - A. 별도 첨부 파일 사용
    - i. 별도 첨부된 cordova-app.zip 파일로부터 eXbuilder6 프로젝트를 임포트 하십시오.
    - ii. File > Import... 메뉴를 선택합니다.
    - iii. General 섹션의 “Existing Projects into Workspace”를 선택하고 [Next]를 클릭합니다.
    - iv. “Select archive file” 필드를 체크하고, 해당 zip 파일을 선택하십시오.
    - v. [Finish] 버튼을 눌러 프로젝트를 가져 옵니다.
    - vi. 워크스페이스에 “cordova-app” 프로젝트가 생성되었는지 확인 합니다.
  - B. 내장 예제 사용
    - i. 프로젝트 익스플로러를 우클릭 하여 “새 eXbuilder6 예제 프로젝트 선택”
    - ii. 프로젝트 이름을 “cordova-app”으로 지정합니다.
    - iii. 예제 선택에서 “cordova 예제”를 선택합니다.
    - iv. [Finish] 버튼을 클릭합니다.
2. “cordova-app” 프로젝트의 빌드 결과물을 템플릿 프로젝트의 “www/app” 디렉토리에 동기화 시킵니다.
  - A. “cordova-app” 프로젝트를 우 클릭하고, “Properties”를 선택합니다.
  - B. “eXbuilder6” 섹션의 “Compiler” 항목을 선택하고, “빌드 미러링” 탭을 선택합니다.
  - C. “추가” 버튼을 클릭하고, 코도바 템플릿 프로젝트의 “www/app” 디렉토리를 선택합니다.
  - D. [OK] 버튼을 클릭합니다.
  - E. 템플릿 프로젝트의 “www/app” 디렉토리에 빌드 결과물이 미러링 되고 있는지 확인하십시오.
3. 런타임 탑재하기
  - A. 스튜디오 메인 메뉴 > eXbuilder6 > 도구 내보내기를 선택합니다.
  - B. [ ] Runtime Library 항목만 체크합니다.
  - C. 보낼 경로로 임시 디렉토리를 선택하고, [Finish] 버튼을 클릭합니다.
  - D. 해당 디렉토리에 runtime 디렉토리가 만들어진 것을 확인 한 뒤, 이 내용을 템플릿 프로젝트의 www/runtime 디렉토리에 옮겨 넣으십시오.



이 과정을 정상적으로 마쳤다면, 템플릿 프로젝트의 내용은 다음과 같습니다:



이제 “cordova-app” eXbuilder6 프로젝트의 내용을 수정할 때마다, 이를 탑재 할 코도바 프로젝트(템플릿 프로젝트)의 www/app 디렉토리가 자동으로 갱신 됩니다.

이제 코도바 디렉토리에서 다음 명령을 이용하여, 직접 실행 해 볼 수 있습니다:

```
$ cordova build android
$ cordova run android # 실제 기기가 연결된 경우 실제 기기가, 그렇지 않은 경우
                        에뮬레이터가 실행 됩니다.
```

iOS 의 경우 다음과 같은 명령을 통해 Xcode 프로젝트를 구성할 수 있습니다.

```
$ cordova prepare ios
# 이후 platforms/ios 디렉토리 안에 Xcode 프로젝트를 찾을 수 있습니다.
# 이 프로젝트를 xcode 로 열어, 서명 정보를 설정하면 실 기기에서 실행할 수 있습니다.
```

### 2.3. 자동화 빌드 및 빌드 구성

빌드 서버를 별도로 운용하여 자동으로 빌드하려면, 이 장을 참고 하십시오. 빌드 서버는 eXbuilder6 스튜디오 없이 무인으로 작동하므로, 빌드 미러링을 이용한 코도바 템플릿 프로젝트가 불가능 하므로 다음의 절차를 수행하는 스크립트를 작성해야 합니다:

1. UI 프로젝트를 체크아웃 받습니다.
2. 체크아웃 받은 소스를 컴파일 하여, 코도바 템플릿 프로젝트의 www/app 디렉토리에 컴파일합니다.
3. 필요한 경우 최신 버전의 eXbuilder6의 런타임을 www/runtime 에 복사합니다.
4. 코도바의 빌드 스크립트를 호출하여 코도바 애플리케이션을 빌드합니다.

#### CI Tool, Runtime Library Export 방법

먼저 빌드를 자동화 하려면 eXbuilder6에 내장된 런타임 및 CI 도구가 필요하며, 이는 도구 내보내기 기능을 통해 확보할 수 있습니다.

1. 스튜디오에서 eXbuilder6 > Export Tools 를 선택합니다.
2. “CI Tool” 및 “Runtime Library”를 선택하고 내보낼 디렉토리를 지정 한 뒤, [Finish] 버튼을 클릭합니다.

#### eXbuilder 6 프로젝트 빌드 방법

eXbuilder 6 UI 프로젝트를 컴파일 하여 템플릿 프로젝트의 www 에 복사합니다.

```
$ java -jar e6-compiler.jar -lg -s UI_프로젝트_경로 -o 코도바_프로젝트_경로/www/app
```

#### eXbuilder 런타임 복사

```
$ cp -r runtime 코도바_프로젝트_경로/www/runtime  
#windows 의 경우 아래 명령어 사용  
$ xcopy /E runtime 코도바_프로젝트_경로/www/runtime
```

#### 안드로이드 애플리케이션 빌드 방법

```
$cordova build android  
# 이후 platforms/android 안에서 빌드된 apk 파일을 찾을 수 있습니다.
```

#### 안드로이드 애플리케이션 실행

```
$cordova run android
```

앞서 설명한 스크립트들을 포함하는 단일 스크립트를 만들어, 전체 소스를 빌드하고 안드로이드 앱을 빌드 하는 전 과정을 하나의 스크립트로 수행 할 수 있습니다.

### 3. 테스트 전략

#### 3.1. 개발 지속 테스트

1. 코도바 플러그인들의 기능을 이용하는 경우, 가급적 개발자들이 직접 코도바 플러그인 API 를 사용하게 하지 마시고, 공통 모듈을 통해 간접적으로 사용하도록 하십시오.
2. 이러한 공통 모듈들은 실행 환경이 브라우저인지 코도바 실제 기기 환경인지를 구분하여 작동하도록 하여, 가능한 대부분의 테스트가 일반 브라우저에서 이뤄질 수 있도록 하십시오.  
예제 프로젝트의 “camera.module.js”를 참고 하십시오.

#### 3.2. 에뮬레이터 테스트

1. 크롬 실행
2. 주소 표시줄에 chrome://inspect/#devices 입력
3. 에뮬레이터 기동

```
$ emulator -avd test
emulator-x86 링크 를 못 찾는 경우
linux 기반 : ln -s
window 기반 : mklink /d 이용 해서 심볼릭 링크 디렉토리를 변경시켜준다.
권한이 없으면 관리자권한으로 실행한다.
```

4. 템플릿 프로젝트에서 다음 명령어 입력

```
$ cordova run android --emulator
```

5. 앱이 빌드되고 에뮬레이터에 설치되고 자동 실행됨.
6. 크롬 인스펙터에서 에뮬레이터 장비가 잡히면, inspect 링크를 클릭하여 원격 디버깅 시작합니다.

#### 3.3. 실기기 테스트

에뮬레이터 테스트와 동일하나, 아래와 같이 안드로이드 앱을 설치하고 실행합니다..

```
$ cordova run android
```

기기 설정에서 개발자 옵션에 웹 인스펙터로 원격 디버깅을 허용해야 합니다.

## 4. 기타

### 4.1. 인증서

공인/비공인 인증서를 통해 안드로이드 APK 를 빌드 하려면 다음 명령어로 빌드해야 합니다.

```
$ cordova build android --release -- --keystore="..../android.keystore" --storePassword=android  
--alias=mykey
```

iOS 의 경우, xcode 로 해당 프로젝트를 열고, 서명 설정을 1 회 구성하고 나면 이후부터 다음 명령어로 빌드하거나 수동으로 빌드 할 수 있습니다.

```
$ cordova build ios --release
```