

서버 설정 가이드

17.0.0.1000



이 문서에 잘못된 정보가 있을 수 있습니다. 투비소프트는 이 문서가 제공하는 정보의 정확성을 유지하기 위해 노력하고 특별한 언급 없이 이 문서를 지속적으로 변경하고 보완할 것입니다. 그러나 이 문서에 잘못된 정보가 포함되어 있지 않다는 것을 보증하지 않습니다. 이 문서에 기술된 정보로 인해 발생할 수 있는 직접적인 또는 간접적인 손해, 데이터, 프로그램, 기타 무형의 재산에 관한 손실, 사용 이익의 손실 등에 대해 비록 이와 같은 손해 가능성에 대해 사전에 알고 있었다고 해도 손해 배상 등 기타 책임을 지지 않습니다.

사용자는 본 문서를 구입하거나, 전자 문서로 내려 받거나, 사용을 시작함으로써, 여기에 명시된 내용을 이해하며, 이에 동의하는 것으로 간주합니다.

각 회사의 제품명을 포함한 각 상표는 각 개발사의 등록 상표이며 특허법과 저작권법 등에 의해 보호를 받고 있습니다. 따라서 본 문서에 포함된 기타 모든 제품들과 회사 이름은 각각 해당 소유주의 상표로서 참조용으로만 사용됩니다.

발행처 | (주)투비소프트

발행일 | 2018/07/23

주소 | (06083) 서울시 강남구 봉은사로 617 인탑스빌딩 2-5층

전화 | 02-2140-7700

홈페이지 | www.tobesoft.com

고객지원센터 | support.tobesoft.co.kr

제품기술문의 | 1588-7895 (오전 10시부터 오후 5시까지)

변경 이력

버전	변경일	내용
17.0.0.100	2017-10-13	17.0.0.100 공개로 전환
17.0.0.500	2018-03-05	Nexacro Mobile Library 관리 화면이 변경되었습니다.
17.0.0.1000	2018-07-23	Nexacro Mobile Library 관리 화면이 변경되었습니다. nexacro-xeni 설치 시 배포되는 파일 목록을 갱신했습니다.

차례

저작권 및 면책조항	ii
변경 이력	iii
차례	iv

파트 I. 서버 API 1

1. 넥사크로플랫폼 X-API 설치	2
1.1 설치	2
1.2 라이선스 인증	2
1.3 설치 확인	3
2. nexacro-xeni 설치	5
2.1 설치	5
2.2 설치 확인	7
2.3 주요 설정	8
2.4 예제	8
2.4.1 Export	9
2.4.2 Import	9

파트 II. App Builder 10

3. App Builder 설치	11
3.1 App Builder 설치 전 필요한 작업	11
3.1.1 App Builder 설치 공통	11
3.1.2 Android	11
3.1.3 iOS, macOS	12
3.2 App Builder 설치	12
4. Server Configuration 설정	15
4.1 General	16
4.2 iOS & macOS Configuration	16

4.3	In-house Distribution	17
4.4	Android Configuration	18
5.	Signing 설정	20
5.1	Android	22
5.2	iOS	23
5.3	macOS	24
6.	User Library 관리	25
7.	Nexacro Mobile Library 관리	27
8.	User 관리	29

파트 I.

서버 API

1.

넥사크로플랫폼 X-API 설치

넥사크로플랫폼 X-API는 서버모듈로 제공되며 서버와 클라이언트 간에 데이터 처리를 위해 필요한 기능을 제공합니다. 기본적인 데이터 송신, 수신 기능과 간단하게 필요한 데이터를 가공하는 기능을 제공해 데이터 처리 과정을 단순화합니다.

1.1 설치

X-API는 자바 기반 서버모듈로 제공되며 JDK 또는 JRE 1.4 이상 버전이 필요합니다.

설치 CD 내 nexacro-xapi/lib 디렉터리에 포함된 jar 파일을 WAS의 /WEB-INF/lib 디렉토리 또는 정의된 클래스 경로에 복사합니다. 제공되는 jar 파일은 아래와 같습니다.

파일명	필수여부	설명	참조
nexacro-xapi-x.x.x.jar	Y	X-API	
commons-logging-x.x.x.jar	Y	X-API 내부 로깅	Apache Commons Logging

1.2 라이선스 인증

라이선스 파일(nexacro17_server_license.xml)을 jar 파일과 같은 디렉토리 또는 정의된 클래스 경로에 복사합니다.



1개 이상의 라이선스 파일이 다른 경로로 복사된 경우에는 jar 파일과 같은 디렉토리에 있는 라이선스 파일을 먼저 적용합니다.



서버 환경 설정에 따라 jar 파일과 라이선스 파일을 복사하고 WAS 재시작이 필요할 수도 있습니다.

1.3 설치 확인

X-API 정상 설치 여부는 간단하게 아래 샘플 JSP 파일을 작성해 확인할 수 있습니다.

```
<%@ page contentType="text/html; charset=UTF-8" %>

<html>
  <head>
    <title>JarInfo</title>
    <style>
      * { font-family: Verdana }
    </style>
  </head>
  <body>
    <pre>
<%
  new com.nexacro17.xapi.util.JarInfo().info(out);
%>
      </pre>
    </body>
  </html>
```

정상적으로 설치가 되었다면 X-API 버전 정보와 제품 라이선스 정보가 출력됩니다.


```
--- Manifest ---
Manifest-Version: 1.0
Built-By: Server Team
Created-By: 1.5.0_22-b03 (Sun Microsystems Inc.)
Ant-Version: Apache Ant 1.7.1
Main-Class: com.nexacro17.xapi.util.JarInfo
Built-Date: April 5 2017

Name: com.nexacro17.xapi
Specification-Title: X-API
Implementation-Title: X-API
Specification-Version: 1.0
Specification-Vendor: FUJITSU CO., LTD.
Implementation-Vendor-Id: com.nexacro17
Implementation-Version: 1.0.0.6271
Implementation-Vendor: FUJITSU CO., LTD.

--- Log ---
INFO Loaded license file in JAR dir: path=C:\Program Files\Apache Software Foundation\Tomcat

--- License ---
product.name: nexacro platform
product.version: 17
product.function: Windows, WindowsRT, MAC, iOS, Android
customer.name: FUJITSU
server.cpuCoreCount: 0
server.ipAddress: 0.0.0.0
date.activation: 2017-04-01
date.term: 2
date.term.unit: month
```

2.

nexacro-xeni 설치

nexacro-xeni는 서버모듈로 제공되며 클라이언트에서 파일 형태로 데이터 처리를 할 때 필요한 기능을 제공합니다. 파일에 포함된 데이터를 그리드 컴포넌트로 가져오거나 그리드에 연결된 데이터를 파일로 내보내는 기능을 처리합니다.

nexacro-xeni는 아래와 같은 파일 형태와 기능을 지원합니다.

파일 형태	확장자	기능
Microsoft Excel 97-2003	xls	export / import
Microsoft Excel 2007/2010	xlsx	export / import
CSV format file	csv	import

2.1 설치

nexacro-xeni는 자바 기반 서버모듈로 제공되며 JDK 또는 JRE 1.4 이상 버전이 필요합니다.

제공된 war(Web application ARchive) 파일을 직접 배치하거나 war 파일의 압축을 풀어 필요한 파일을 WAS의 / WEB-INF/lib 디렉터리 또는 정의된 클래스 경로에 복사해 사용할 수 있습니다. 제공되는 주요 jar(Java ARchive) 파일은 아래와 같습니다.

```
commons-codec-1.5.jar
commons-fileupload-1.3.1.jar
commons-io-2.2.jar
commons-logging-1.1.jar
dom4j-1.6.1.jar
log4j-1.2.13.jar
nexacro17-xapi-1.0.jar
nexacro17-xeni-1.0.jar
poi-3.10-FINAL-20140208.jar
poi-ooxml-3.10-FINAL-20140208.jar
```

```
poi-ooxml-schemas-3.10-FINAL-20140208.jar
poi-scratchpad-3.10-FINAL-20140208.jar
stax-api-1.0.1.jar
xmlbeans-2.3.0.jar
```



war 파일을 직접 배치하지 않고 압축을 풀어 복사하는 경우에는 라이브러리 버전 차이로 문제가 발생할 수 있습니다.



nexacro-xeni는 Apache POI 라이브러리를 사용합니다. 자세한 Apache POI 설명은 아래 링크를 참고해주세요.

<https://poi.apache.org/>



nexacro-xeni가 동작하려면 X-API가 설치되어 있어야 합니다. [넥사크로플랫폼 X-API 설치](#) 항목을 참고해주세요.

기존 컨텍스트에 파일을 복사한 경우에는 아래 내용을 web.xml 파일에 추가해주어야 합니다.

```
<servlet>
  <servlet-name>XExportImport</servlet-name>
  <servlet-class>com.nexacro.xeni.services.GridExportImportServlet</servlet-class>
</servlet>
<servlet>
  <servlet-name>XImport</servlet-name>
  <servlet-class>com.nexacro.xeni.services.GridExportImportServlet</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>XExportImport</servlet-name>
  <url-pattern>/XExportImport</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>XImport</servlet-name>
  <url-pattern>/XImport</url-pattern>
</servlet-mapping>

<context-param>
  <param-name>export-path</param-name>
  <param-value>/export</param-value>
```

```

</context-param>
<context-param>
  <param-name>import-path</param-name>
  <param-value>/import</param-value>
</context-param>
<context-param>
  <param-name>monitor-enabled</param-name>
  <param-value>true</param-value>
</context-param>
<context-param>
  <param-name>monitor-cycle-time</param-name>
  <param-value>30</param-value>
</context-param>
<context-param>
  <param-name>file-storage-time</param-name>
  <param-value>10</param-value>
</context-param>

```

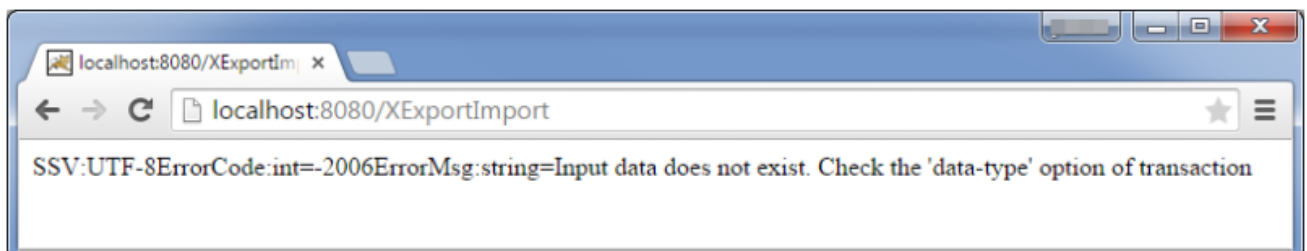


jar 파일 목록과 web.xml에 추가되는 내용은 nexacro17-xeni_20180531_1 버전 기준입니다. 업데이트에 따라 목록이 변경될 수 있습니다.

2.2 설치 확인

정상적으로 배치 또는 복사되었다면 아래 URL을 웹브라우저에서 열어 메시지가 정상적으로 나타나는지 확인합니다.

<http://127.0.0.1:8080/XExportImport>





URL은 설치 경로에 따라 달라질 수 있습니다. war 파일을 직접 배치한 경우 URL은 아래와 같습니다.
http://127.0.0.1:8080/nexacro-xeni/XExportImport

2.3 주요 설정

서블릿 컨텍스트에 파라미터로 설정된 주요 항목은 아래와 같습니다.

이름	기본값	설명
export-path	/export	내려받을 파일을 임시로 저장하는 경로를 지정합니다.
import-path	/import	그리드 컴포넌트로 가져올 파일을 임시로 저장하는 경로를 지정합니다. monitor-enabled 설정값이 true인 경우에는 데이터 처리 후 바로 삭제합니다.
monitor-enabled	true	파일 매니저 실행 여부를 지정합니다. false로 지정한 경우에는 파일 매니저가 동작하지 않고 임시 파일을 삭제하지 않습니다.
monitor-cycle-time	30	임시 저장된 내려받을 파일을 삭제하는 주기를 '분' 단위로 설정합니다. 해당 주기로 파일 매니저를 실행해 삭제할 파일을 체크하고 삭제 처리합니다.
file-stroage-time	10	임시로 저장한 내려받을 파일을 보관하는 시간을 '분' 단위로 설정합니다. 파일 매니저 실행 시 생성 후 지정된 시간이 지난 파일은 삭제합니다.



자세한 내용은 배포되는 war 파일에 포함된 readme.txt 파일을 참고하세요.

2.4 예제

간단한 데이터 처리 예제를 설명합니다. 세부적인 속성이나 메소드, 이벤트는 도움말에서 아래 항목을 참고하세요.

Objects > Misc. Objects > ExcelExportObject

2.4.1 Export

그리드 컴포넌트에 바인딩된 데이터를 서버로 전송해 파일로 생성하고 생성된 파일을 내려받습니다. 아래 예제에서는 Grid00 컴포넌트에 바인딩 된 데이터를 파일로 생성하고 내려받습니다.

```
this.Button00_onclick = function(obj:Button, e:nexacro.ClickEventInfo)
{
    this.exportObj = new ExcelExportObject();
    var ret = this.exportObj.addExportItem(nexacro.ExportItemTypes.GRID,
        this.Grid00, "Sheet1!A1");
    this.exportObj.set_exporturl("http://localhost:8080/XExportImport");
    this.exportObj.exportData();
}
```

2.4.2 Import

사용자가 가지고 있는 파일을 서버로 전송해 데이터셋으로 변환합니다. 사용자는 변환된 데이터셋을 내려받아 업무를 처리합니다.

```
this.Button00_onclick = function(obj:Button, e:nexacro.ClickEventInfo)
{
    this.importObj = new ExcelImportObject("Import00",this);
    this.importObj.addEventHandler("onsuccess", this.Import_onsuccess, this);
    this.importObj.set_importurl("http://localhost:8080/XExportImport");
    this.importObj.importData("", "output=ds", "Dataset00=ds");
}

this.Import_onsuccess = function(obj,e)
{
    this.Grid00.createFormat();
}
```

파트 II.

App Builder

3.

App Builder 설치

3.1 App Builder 설치 전 필요한 작업

App Builder를 서버에 설치하기 전에 필요한 작업입니다. App Builder를 설치하기 위한 공통 작업과 배포 대상에 따라 추가 작업이 필요합니다.

3.1.1 App Builder 설치 공통

JDK 설치 후 Tomcat(WAS)을 설치합니다.



본문에서는 JDK 1.6, Tomcat 7 버전을 사용했습니다.

설치파일은 아래 링크에서 내려받을 수 있습니다.

<http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html>

<http://tomcat.apache.org/download-70.cgi>

3.1.2 Android

Android SDK를 설치합니다. 안드로이드 스튜디오 설치 후 SDK Manager를 실행하거나 명령행 도구를 내려받아 SDK를 설치할 수 있습니다.



본문에서는 Android 7.0(API 24) 버전을 사용했습니다.

설치파일은 아래 링크에서 내려받을 수 있습니다.

<https://developer.android.com/studio/index.html#downloads>

3.1.3 iOS, macOS

Xcode를 설치하고 필요한 SDK를 설치합니다. Xcode는 macOS 이외의 운영체제에는 설치할 수 없으므로 macOS 운영체제를 사용할 수 있는 장비가 필요합니다.



본문에서는 Xcode(Build 9C40b), iOS SDK(11.2), OS X SDK(10.13) 버전을 사용했습니다. 설치파일은 Mac App Store에서 내려받을 수 있습니다.

3.2 App Builder 설치

App Builder는 WAR(Web application ARchive) 파일로 제공됩니다. 제공된 WAR 파일(appbuilder.war)을 각 WAS 환경에 따라 배포합니다.



본문에서는 Tomcat 7 버전을 사용했습니다. 해당 버전 기준으로 설명합니다.

① WAR 파일 배포

[webapps] 폴더에 appbuilder.war 파일을 복사해놓고 Tomcat 서버를 시작합니다. webapps 경로에 appbuilder 폴더가 생성된 것을 확인하고 Tomcat 서버를 중지합니다.

② 관리 콘솔 설정

넥사크로플랫폼 14 라이선스 파일을 [webapps/appbuilder/WEB-INF/lib] 폴더에 복사해놓습니다.



App Builder 관리 콘솔은 넥사크로플랫폼 14 애플리케이션으로 만들어졌기 때문에 14 버전 라이선스 파일이 필요합니다. 해당 설정은 이후 변경될 수 있습니다.

[webapps/appbuilder/pbuilder_ui] 폴더에서 pbuilder_ui.xadl.js 파일을 텍스트 편집기로 열고 "host:port"로 표기된 부분을 사용할 URL로 수정합니다.

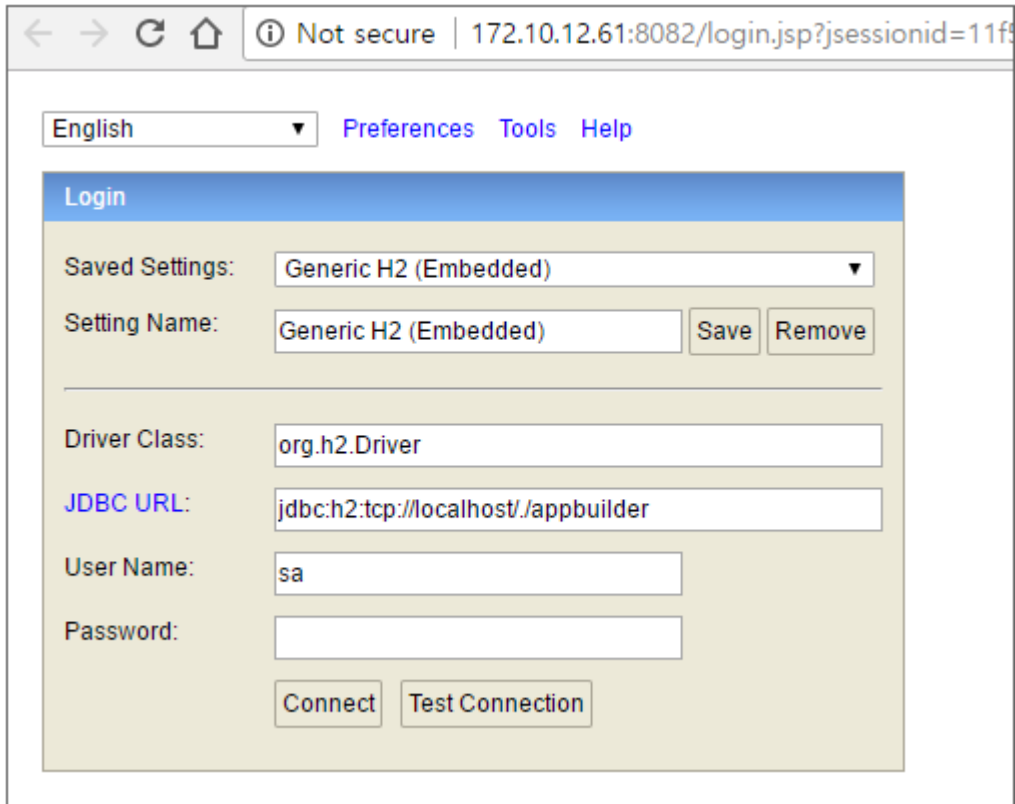
```
this._addService("default_typedef.xml", "ServiceURL", "JSP", "http://host:port/appbuilder", "", null, "", "0", "0");

this._addService("default_typedef.xml", "ServiceURL", "JSP", "http://xxx.xxx.xxx.xxx:8080/
```

```
appbuilder", "", null, "", "0", "0");
```

3 H2 Database 시작

[webapps/appbuilder/WEB-INF/package/h2/bin] 폴더에서 runH2DB 배치 파일을 실행하면 H2 Database가 시작되고 관리 콘솔이 표시됩니다. 관리 콘솔창은 사용하지 않습니다. 열린 웹브라우저 창을 닫습니다.



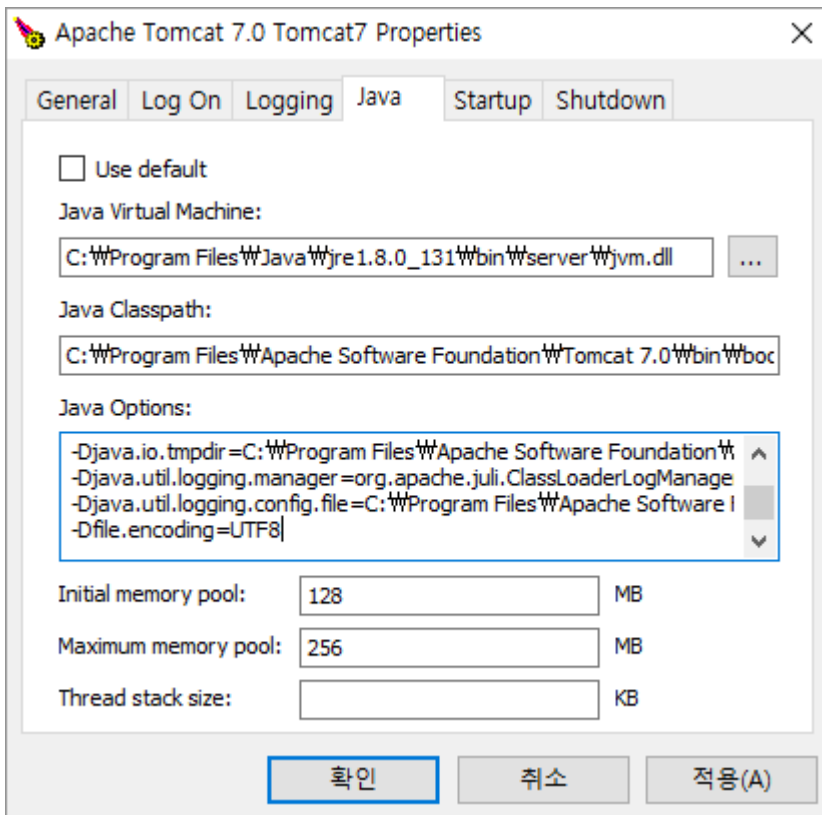
H2 Database를 처음 시작했을 때 해당 폴더에 필요한 파일을 생성하는데 윈도우 운영체제를 사용하는 경우에는 Tomcat 설치 경로에 따라 관리자 권한이 없으면 오류가 발생하는 경우가 있습니다. 오류가 발생하는 경우에는 관리자 권한으로 배치 파일을 실행해야 합니다.

4 인코딩 설정

일부 환경에서 한국어가 정상적으로 표기되지 않을 수 있습니다. 그런 경우에는 인코딩 설정을 추가합니다. [bin] 폴더 아래 catalina.bat 파일을 열어 아래와 같이 수정합니다.

```
set "JAVA_OPTS=%JAVA_OPTS% %LOGGING_CONFIG% -Dfile.encoding=UTF8"
```

Tomcat 버전에 따라 catalina.bat 파일을 사용하지 않는 경우가 있습니다. 그런 경우에는 [bin] 폴더 아래에서 실행파일인 Tomcat7w.exe를 실행하면 Tomcat 속성을 설정할 수 있는 창이 나타납니다. 해당 창에서 [Java > Java Options] 항목에 "-Dfile.encoding=UTF8" 항목을 추가합니다.



5 서버 시작

[webapps] 폴더에 복사해놓은 appbuilder.war 파일을 삭제하고 Tomcat 서버를 다시 시작합니다.

6 관리 콘솔 접근

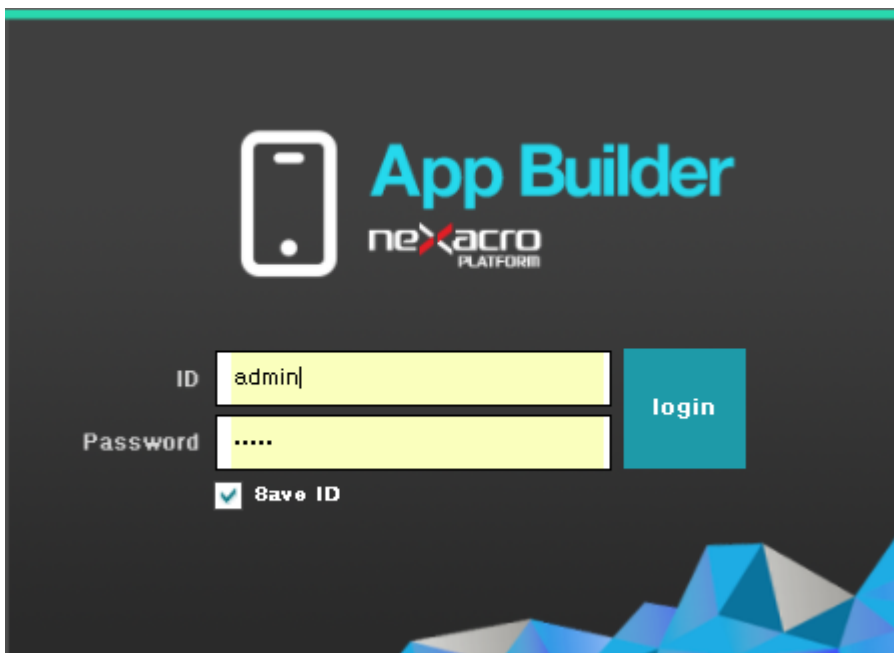
웹 브라우저를 열어 아래 URL을 입력 후 관리 콘솔에 접근할 수 있는지 확인합니다. 입력할 URL은 pbuilder_ui.xadl.js 파일에서 수정한 URL입니다.

http://xxx.xxx.xxx.xxx:8080/appbuilder

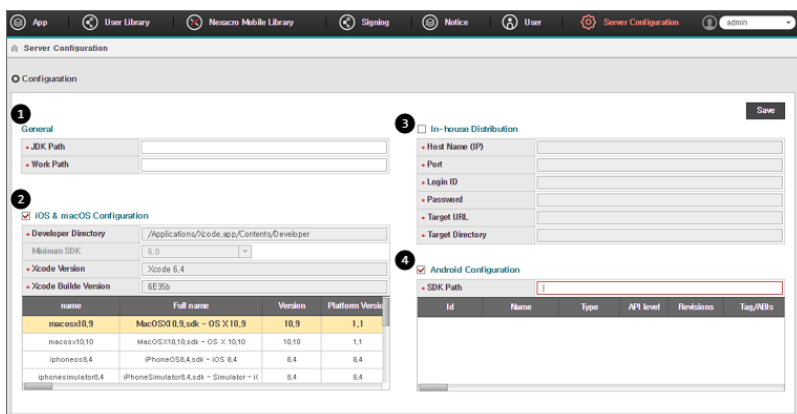
4.

Server Configuration 설정

App Builder 설치 이후 관리 콘솔에 접속해 App Builder 사용을 위한 설정을 진행합니다. 관리 콘솔에 접속하면 로그인 창이 표시됩니다. 최초 로그인 시 ID는 "admin", Password는 "admin"입니다. Password는 로그인 후 메뉴[User]에서 변경할 수 있습니다.



App Builder 환경 설정을 위해 관리 콘솔에서 메뉴[Server Configuration] 항목을 선택합니다. 해당 메뉴는 4가지로 구분되어 있습니다. General 항목을 제외한 나머지 항목은 체크박스에서 사용 여부를 선택할 수 있습니다.



	항목	설명
1	General	JDK 경로와 생성된 앱 프로젝트를 관리할 경로를 지정합니다.
2	iOS & macOS Configuration	iOS, macOS SDK 정보를 표시합니다.
3	In-house Distribution	In-house 배포 설정 시 관련 정보를 지정합니다.
4	Android Configuration	Android SDK 설치 경로를 지정합니다.

4.1 General

General 항목은 기본적으로 작성해야 하는 항목입니다.

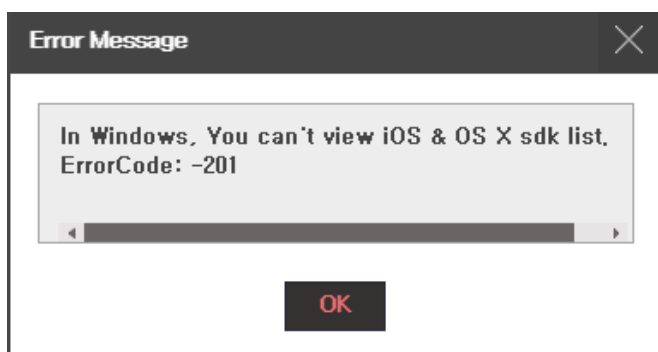
General

* JDK Path	/Library/Java/JavaVirtualMachines/jdk1.8.0_60.jdk/Contents/Home
* Work Path	/Users/User/Documents/appbuilder/output

	항목	설명
1	JDK Path	JDK 설치 경로를 지정합니다.
2	Work Path	앱 프로젝트를 생성하고 업로드한 리소스 파일을 저장할 작업 경로를 지정합니다.

4.2 iOS & macOS Configuration

iOS, macOS 앱을 빌드하기 위한 설정입니다. App Builder를 윈도우 운영체제에 설치하고 해당 항목의 체크박스를 체크했다면 아래와 같은 오류 메시지를 보여줍니다.



Xcode와 관련 SDK가 정상적으로 설치된 경우라면 관련 설정을 자동으로 확인하고 설정값을 표시합니다. 설정값이 표시되지 않는 경우에는 Xcode나 SDK가 정상적으로 설치되었는지 확인합니다.

☒ **iOS & macOS Configuration**

* Developer Directory

/Applications/Xcode.app/Contents/Developer

Minimum SDK

6.0

* Xcode Version

Xcode 9.1

* Xcode Build Version

9B55

name	Full name	Version	Platform Version
iphoneos11.1	iPhoneOS11.1.sdk - iOS 11.1	11.1	11.1
iphonesimulator11.1	iPhoneSimulator11.1.sdk - Simulator - iOS	11.1	11.1
macosx10.13	MacOSX10.13.sdk - macOS 10.13	10.13	1.1
appletvos11.1	AppleTVOS11.1.sdk - tvOS 11.1	11.1	11.1

	항목	설명
1	Developer Directory	Xcode developer directory를 표시합니다.
2	Minimum SDK	빌드를 위한 Deployment Target 정보를 표시합니다. 6.0으로 고정되어 있습니다.
2	Xcode Version	설치된 Xcode 버전을 표시합니다.
3	Xcode Build Version	Xcode 빌드 버전을 표시합니다.
4	SDK Details	설치된 관련 SDK 상세 정보를 표시합니다.

4.3 In-house Distribution

In-house Distribution이란 Apple App Store를 사용하지 않고 자체적으로 설치파일을 배포할 수 있는 서버를 사용하는 것입니다. 해당 기능을 사용하기 위해서는 HTTPS 설정이 되어 있어야 하며 공인된 도메인으로 발급받은 서버 인증서가 갖춰져야 합니다.

In-house Distribution 항목이 설정된 경우에는 앱 빌드 시 .ipa 파일과 .plist 파일을 설정된 서버로 전송합니다. 빌드된 앱을 내려받을 때 설정한 Host 정보를 적용합니다.

☒ **In-house Distribution**

* Host Name (IP)	cacao.naxacro.com
* Port	22
* Login ID	nexacro
* Password	*****
* Target URL	https://cacao.naxacro.com/~nexacro/appbuilder/tops
* Target Directory	/home/nexacro/public_html/appbuilder/tops

	항목	설명
1	Host Name(IP)	서버 호스트명 또는 IP 정보를 입력합니다.
2	Port	파일 전송을 위한 Port 정보를 입력합니다.
3	Login ID	서버 접속 시 로그인 ID를 입력합니다.
4	Password	서버 접속 시 로그인 Password를 입력합니다.
5	Target URL	복사된 .ipa , .plist 파일을 참조하게 될 URL을 입력합니다.
6	Target Directory	복사된 .ipa , .plist 파일이 위치하게 경로를 입력합니다.



Target URL 항목에 입력된 URL과 실제 파일을 내려받기 위해 접속하는 URL은 다릅니다. Target URL 항목은 .ipa, .plist 파일이 실제 저장된 경로이며 파일을 내려받을 때는 빌드 후 제공되는 URL에 접속해 설치 파일을 내려받습니다.



HTTPS 설정이 되어 있지 않거나 공인된 도메인으로 발급받은 서버 인증서가 없는 경우에는 "In-house Distribution" 기능을 통해 앱 설치를 할 수 없습니다. 이런 경우에는 "In-house Distribution" 기능을 사용하지 않고 .ipa 파일을 PC에 내려받은 후 Xcode 또는 애플 아이튠즈를 통해 앱을 설치해야 합니다.

4.4 Android Configuration

☒ **Android Configuration**

* SDK Path	/Users/tobesoft/appbuilder/android-sdk-macosx				
Id	Name	Type	API level	Revisions	Tag/ABIs
android-24	Android 7.0	Platform	24	2	no ABIs.

	항목	설명
1	SDK Path	Android SDK가 설치된 경로를 지정합니다.



SDK Path 항목 설정 후 설치된 Android SDK 목록이 표시되지 않는 경우에는 아래 사항을 확인합니다.

- * Android SDK가 정상적으로 설치되었는지 확인
- * 설치된 Android SDK Path를 정확하게 지정했는지 확인
- * Android SDK Manager 를 실행해 Android 7.0(API 24)가 설치되었는지 확인

5.

Signing 설정

앱 빌드 시 사용할 Signing 설정을 위해 관리 콘솔에서 메뉴[Signing] 항목을 선택합니다. Signing 파일을 등록하거나 생성(Android의 경우)하고 등록된 내용을 확인할 수 있습니다.

OS Type	Name	Owner	Access	Last Update Date
Android	android-signing	admin	public	2017-01-25 14:25:45
iOS	ios-inhouse	admin	public	2017-01-17 17:35:04
macOS	osx-signing	admin	public	2017-01-26 15:06:21

Details	
Name	android-signing
Owner	admin
Access	<input type="radio"/> share <input checked="" type="radio"/> public <input type="radio"/> private
Keystore Alias Name	android
Keystore Alias Password	*****
Keystore File	<input checked="" type="radio"/> Generate <input type="radio"/> Upload <input type="text" value="android-signing.keystore"/>
Keystore Password	*****

	항목	설명
1	Sign Name	등록된 항목의 Name 정보를 검색합니다.
2		항목을 추가하거나 등록된 항목을 삭제합니다.
3		등록된 항목의 목록을 표시합니다.
4	Details	추가할 항목의 세부 정보를 입력하거나 등록된 항목의 세부 정보를 표시합니다. Name, Owner, Access 항목은 배포할 대상과 상관없이 공통으로 적용되는 항목입니다.

* Name	<input type="text" value="android-signing"/>							
* Owner	<input type="text" value="admin"/>							
* Access	<input type="radio"/> share <input checked="" type="radio"/> public <input type="radio"/> private	<div style="text-align: right;">+ -</div> <table border="1"> <thead> <tr> <th></th> <th>ID</th> <th>Authority</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table>		ID	Authority			
	ID	Authority						

	항목	설명
1	Name	등록할 항목 이름을 입력합니다.
2	Owner	Keystore(또는 Signing)의 소유자를 입력합니다. (로그인 계정으로 자동 입력됩니다).
3	Access	Keystore(또는 Signing)의 접근 권한을 설정합니다. 권한 설정에 따라 사용자가 App 생성 시 사용할 수 있는 Signing 목록이 다르게 표시됩니다. - share: 지정된 사용자 공유 - public: 전체 사용자 공유 - private: Owner만 사용

Keystore 접근 권한 설정 시 지정된 사용자 공유(share)를 선택한 경우에는 등록된 사용자 중 권한을 부여할 대상을 선택할 수 있습니다.

The screenshot shows the 'User List To Share' dialog box. The dialog has a table with the following data:

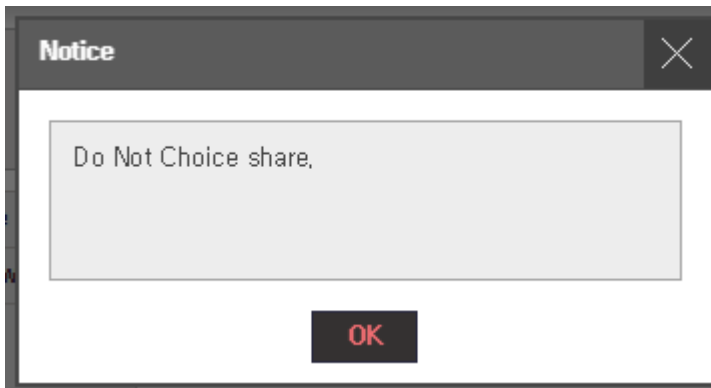
	id	name	part	authority	active
<input type="checkbox"/>	user1	user1		Developer	0

At the bottom of the dialog, there are 'Add' and 'Cancel' buttons. A red arrow points from the '+' button in the main form to the dialog box.

keystroe 접근 권한이 부여된 사용자를 확인할 수 있으며 해당 사용자를 삭제하면 접근 권한이 해제됩니다.

	ID	Authority
<input type="checkbox"/>	user1	Developer

public, private 항목을 선택한 경우에는 사용자를 선택 기능이 동작하지 않습니다.



5.1 Android

	항목	설명
1	Keystore Alias Name	Keystore Alias 이름을 입력합니다. Keystore Alias 이름은 6자 이상으로 입력해야 합니다.
2	Keystore Alias Password	Keystore Alias의 비밀번호를 입력합니다. Keystore Alias 비밀번호는 6자 이상으로 입력해야 합니다.
3	Keystore File	Keystore 파일을 생성하거나 가지고 있는 파일을 등록합니다. - Generate: App Builder에서 Keystore 파일을 생성합니다. - Upload: Keystore 파일을 등록하고 등록된 파일을 표시합니다. 등록된 파일은 링크 형태로 표시되며 링크를 클릭하면 해당 파일을

	항목	설명
		내려받을 수 있습니다.
4	Keystore Password	Keystore 비밀번호를 입력합니다.



Keystore에 대한 자세한 사항은 아래 URL을 참고하세요.

<http://developer.android.com/tools/publishing/app-signing.html>

5.2 iOS

★ Certificate File	<input type="text" value="TOBESOFT_In-House_Distribution_ServerTeam.p12"/>	Upload
★ Certificate Name	<input type="text" value="iPhone Distribution: TOBESOFT"/>	
★ Certificate Password	<input type="password" value="*****"/>	
★ Provisioning Profile	<input type="text" value="TOBESOFT_InHouse_Distribution_ServerTeam.mobileprov"/>	Upload
★ Login Keychain Password	<input type="password" value="*****"/>	

	항목	설명
1	Certificate File	iOS Certificate 파일을 등록합니다.
2	Certificate Name	iOS Certificate 이름을 입력합니다.
3	Certificate Password	등록한 iOS Certificate 파일 비밀번호를 입력합니다.
4	Provisioning Profile	Provisioning Profile 파일을 등록합니다.
5	Login Keychain Password	로그인 키체인 비밀번호를 입력합니다.



iOS Certificate 파일 발급 시 등록한 Package 이름과 앱 생성 시 입력한 Package 이름이 같아야 정상적으로 앱을 빌드할 수 있습니다.



Certificate 파일과 Provisioning Profile 파일에 대한 자세한 사항은 아래 URL을 참고하세요.

<https://developer.apple.com/library/ios/documentation/IDEs/Conceptual/AppDistributionGuide/MaintainingCertificates/MaintainingCertificates.html>

5.3 macOS

★ Certificate File	<input type="text" value="developer-id-12345678.p12"/> <input type="button" value="Upload"/>
★ Certificate Name	Developer ID Application: TOBESOFT (P4W1N4G6T7Q)
★ Certificate Password	*****
★ Login Keychain Password	*****

	항목	설명
1	Certificate File	macOS Certificate 파일을 등록합니다.
2	Certificate Name	macOS Certificate 이름을 입력합니다.
3	Certificate Password	등록한 iOS Certificate 파일 비밀번호를 입력합니다.
5	Login Keychain Password	로그인 키체인 비밀번호를 입력합니다.



macOS Certificate 파일 발급 시 등록한 Package 이름과 앱 생성 시 입력한 Package 이름이 같아야 정상적으로 앱을 빌드할 수 있습니다.

6.

User Library 관리

넥사크로플랫폼은 모바일 운영체제에서 제공하는 기능을 넥사크로플랫폼 스크립트 환경에서 사용할 수 있는 기능을 개발할 수 있습니다. 이렇게 개발된 기능을 DeviceAPI라고 하며 라이브러리 형태로 생성해 앱 빌드 시 적용할 수 있습니다.

User Library Name

☒

Name

Description

Version

Device Type

Android Class

Last Upload Time

Owner

☐

Hello World

test

1

iOSlibHelloPlugin.a
Androidlibhelloplugin.jar

com.example.helloplugin.HelloPlugin

2016-06-21 18:42:12

admin

Details

Name

Version

Owner

Description

Hello World

1

admin

test

iOS☒

Android☒

Android Class

libHelloPlugin.a

libhelloplugin.jar

com.example.helloplugin.HelloPlugin

Upload

Upload

Save

목록에 표시된 User Library 항목을 선택하거나 신규로 추가하는 경우 하단 [Details] 항목에서 상세 내용을 확인하거나 등록, 수정할 수 있습니다.

Details

Name

Version

Owner

Description

admin

iOS☒

Android☒

Android Class

Upload

Upload

	항목	설명
1	Name	User Library 이름
2	Version	User Library 버전
3	Owner	등록한 사용자 정보
4	Description	User Library에 대한 설명
5	iOS	iOS User Library 파일 등록 .a 파일을 등록합니다.
6	Android	Android User Library 파일 등록 .jar 파일을 등록합니다.
7	Android Class	Android User Library 파일에서 사용하는 Class를 지정합니다. (패키지명 포함)

7.

Nexacro Mobile Library 관리

넥사크로플랫폼 버전별 모바일 라이브러리 파일을 관리합니다. 앱 빌드 시 애플리케이션 개발에 사용한 넥사크로플랫폼 버전과 모바일 라이브러리 버전이 같지 않은 경우 실행 시 오류가 발생할 수 있습니다.

Nexacro Mobile Library Management

	Name	Version	Owner	Date
<input type="checkbox"/>	lib_20180619	0		2018-07-12 15:10:13
<input type="checkbox"/>	lib_20180717	0		2018-07-12 14:45:51

Details

Save

Name	lib_20180619	Android Library File	nexacro17_Android_20180717_1.zip	Upload
Version	0	iOS Library File	nexacro17_iOS_20180717_1.zip	Upload
Owner		macOS Library File	nexacro17_macOS_20180717_1.zip	Upload
		Nexacro Framework File	nexacro17_frameworklibrary_20180717_1.zip	Upload
Description				



관리자 권한을 가진 사용자가 등록한 모바일 라이브러리는 사용자 모두가 확인하고 앱 빌드 시 사용할 수 있습니다. 하지만 개별 사용자가 등록한 모바일 라이브러리는 해당 사용자와 관리자만 확인하고 사용할 수 있습니다.

모바일 라이브러리 파일은 제품과 함께 배포됩니다. 지정된 파일 그대로 업로드한 후 사용합니다.

	항목	설명
1	Name	모바일 라이브러리 이름을 입력합니다.
2	Version	모바일 라이브러리 버전을 입력합니다.
3	Owner	소유자 정보를 표시합니다.
4	Description	라이브러리에 대한 설명을 입력합니다.
5	Mobile Library File	배포되는 모바일OS 에 받는 라이브러리 파일을 업로드합니다. 예) Android : nexacro17_Android_2018xxxx_1.zip iOS : nexacro17_iOS_2018xxxx_1.zip

	항목	설명
		macOS : nexacro17_macOS_2018xxxx_1.zip 업로드한 항목은 마우스로 클릭 시 파일을 내려받을 수 있습니다.
6	Nexacro Framework File	배포되는 nexacro17lib.zip 파일을 업로드합니다. 업로드한 항목은 마우스로 클릭 시 파일을 내려받을 수 있습니다.

8.

User 관리

App Builder를 사용할 사용자를 관리합니다. 등록된 사용자 정보는 관리 콘솔에 접근하거나 넥사크로 스튜디오에서 Deploy 시 App Builder를 사용하는 경우 로그인 시 사용됩니다.

User ID

Q Search

+ -

<input type="checkbox"/>	ID	Name	Department	Email	Authority	Active
<input type="checkbox"/>	admin	Administrator			Admin	0
<input type="checkbox"/>	user1	user1			Developer	0

Details

Save

User ID

admin

User Name

Administrator

E-mail

Active

0

Password

Department

Authority

Admin

	항목	설명
1	User ID	사용자 ID를 입력합니다.
2	User Name	사용자 이름을 입력합니다.
3	E-mail	사용자 이메일 주소를 입력합니다.
4	Password	사용자 비밀번호를 입력합니다.
5	Department	사용자 소속 부서를입력정합니다.
6	Authority	사용자 권한을 선택합니다. - Admin: 관리자 권한 - Developer: 개발자 권한 관리자 권한은 User, Server Configuration 메뉴를 사용할 수 있으며 모든 사용자가 생성한 "App", User Library, 모바일 라이브러리를 확인할 수 있습니다.
7	Active	사용 여부를 선택합니다. X로 지정한 경우에는 로그인할 수 없습니다.