開発メモ：各種手順篇 2013.07

本資料は nextserve7 をベースとする開発を行う際に必要となる標準的な手順をまとめるものです。実際の開発にあたっては各プロジェクトで適宜変更して運用されることを想定しています。

なお、資料内の手順等は現時点のものであり、修正／変更する場合がありますのでご了承ください。

目　次

[1 開発環境構築 3](#_Toc409430189)

[1.1 Java 3](#_Toc409430190)

[1.2 Tomcat 3](#_Toc409430191)

[1.3 Eclipse 3](#_Toc409430192)

[1.4 MySQL（オプション） 3](#_Toc409430193)

[1.5 Eclipse プロジェクト作成 3](#_Toc409430194)

[1.6 HTML／Javascript チェックの外し方 3](#_Toc409430195)

[1.7 サーバ起動時の警告ログ 5](#_Toc409430196)

[2 開発環境（オプション） 5](#_Toc409430197)

[2.1 Subversion クライアント【Eclipse プラグイン】 5](#_Toc409430198)

[2.2 プロパティエディタ【Eclipse プラグイン】 5](#_Toc409430199)

[2.3 Java 文法チェッカー【Eclipse プラグイン】 5](#_Toc409430200)

[2.4 JavaScript チェッカー【Eclipse プラグイン】 5](#_Toc409430201)

[2.4.1 プラグイン情報 5](#_Toc409430202)

[2.4.2 オプション設定 5](#_Toc409430203)

[3 自動生成 8](#_Toc409430204)

[3.1 準備：\_ag プロジェクト作成 8](#_Toc409430205)

[3.2 準備：\_ag プロジェクト取り込み 8](#_Toc409430206)

[3.3 自動生成（参考） 9](#_Toc409430207)

[3.3.1 エクセルファイルのロックを取得 9](#_Toc409430208)

[3.3.2 自動生成（\_ag） 9](#_Toc409430209)

[3.3.3 コミット／通知 9](#_Toc409430210)

[3.3.4 補足 9](#_Toc409430211)

[3.3.5 TIPS 9](#_Toc409430212)

[3.4 エクセルファイル作成 10](#_Toc409430213)

[3.5 TIPS 10](#_Toc409430214)

[4 新機能の登録 11](#_Toc409430215)

[4.1 サーバ 11](#_Toc409430216)

[4.1.1 iBatis 設定ファイル 11](#_Toc409430217)

[4.1.2 Spring ビーンロード範囲指定 11](#_Toc409430218)

[4.2 画面 11](#_Toc409430219)

[4.2.1 war/ui/common/js/data/urlMap.js 11](#_Toc409430220)

[4.2.2 war/ui/common/js/data/label\_ja.js 11](#_Toc409430221)

[4.2.3 war/ui/common/template/leftbar.tmpl 11](#_Toc409430222)

[5 サンプル削除 12](#_Toc409430223)

[5.1 サーバ側 12](#_Toc409430224)

[5.1.1 ソースコード 12](#_Toc409430225)

[5.1.2 applicationi-base.context.xml 12](#_Toc409430226)

[5.2 画面側 12](#_Toc409430227)

[5.2.1 HTML ファイル 12](#_Toc409430228)

[5.2.2 ソースコード 12](#_Toc409430229)

[5.2.3 war/ui/common/js/data/label\_ja.js 12](#_Toc409430230)

[5.2.4 war/ui/common/js/data/urlMap.js 12](#_Toc409430231)

[5.2.5 war/ui/common/template/leftbar.tmpl 12](#_Toc409430232)

[6 API の修正 13](#_Toc409430233)

[6.1 認証方式 13](#_Toc409430234)

[6.2 アクセス制限 13](#_Toc409430235)

[6.3 アプリケーションログ 13](#_Toc409430236)

[6.3.1 ログファイル 13](#_Toc409430237)

[6.3.2 固定の出力項目 13](#_Toc409430238)

[6.4 CSRF 対策フィルタ 13](#_Toc409430239)

[7 画面の修正 13](#_Toc409430240)

[7.1 トップ画面の URL 変更方法 13](#_Toc409430241)

# 開発環境構築

開発に使用している環境は「\\yokok\DPFG\public\99\_作業用\itsukaha\public\nextserve7\02 開発環境」に格納してあります。

## Java

Java7 以上の SDK が必要です。 2013.12 Java6 から 7 に変更

* Eclipse 付属の JRE を利用するとコンパイルまたは実行時にエラー（「非推奨のメソッドを使用しています」）が発生する場合があるようです。その場合は、単独のディストリビューションを使用してください。 2014.06 追記

## Tomcat

Tomcat7 系（Servlet3.0）で動作確認しています。 2014.02 デフォルトを Tomcat6 から 7 に変更

* Eclipse 付属の 7.0.26 を利用すると起動時にエラー（「フィルターマッピングには Servlet または url の指定が必要です」）が発生する場合があるようです。その場合は、単独のディストリビューションを使用してください（7.0.32 で動作することは確認済みです）。 2014.06 追記

## Eclipse

「動的 Web プロジェクト」を扱えるバージョンが必要です。Eclipse3.4～4.3 については動作の実績があります。

* 3.4 未満では自動生成の Ant 実行が失敗するという事例報告があります。

2013.07 現在のメインの開発環境は 4.2.2 juno JEE 版です／2013.12 Kepler に移行しました  
 2015.01 Luna に移行しました

## MySQL（オプション）

動作確認には 5.5（4 バイトの Unicode を扱えます）を使用しています。

MySQL の旧バージョン利用者向けの注意事項を以下の場所に格納してあります（ご参考）。

* \\yokok\DPFG\public\99\_作業用\itsukaha\public\nextserve7\02 開発環境\MySQL 5.5\MySQL 5.5 補足.txt
* デフォルトの状態では、HSQL DB を使用するように設定されているため、特定の DBMS のインストールは行わなくても動作に支障はありません。

## Eclipse プロジェクト作成

SVN からプロジェクトを取得して Eclipse に取り込む場合は、「動的 Web プロジェクト」として作成してください。  
プロジェクト作成ウィザード中に注意していただく項目は以下の通りです。

* ターゲット・ランタイム ： Apache Tomcat v7.0
* 動的 web モジュールバージョン ： 3.0
* デフォルト出力フォルダー ： war/WEB-INF/classes（デフォルト：build\classes）
* コンテンツ・ディレクトリー ： war（デフォルト：WebContent）

この状態でサーバを起動することで、プリインストール済みの機能（handsOn／ユーザ管理）を利用可能です。

なお、war/m ディレクトリは開発途中のモバイル版のため、開発開始前に削除してください。

## HTML／Javascript チェックの外し方

HTML／xml／Javascript に関するエラーや警告が大量に表示される場合は、以下の手順でチェックを行わないようにします。（環境によって操作結果が異なるようです。どの方法でエラーが出なくなったかお知らせ頂ければ助かります・・・。）

【方法その１】

1. プロジェクトのプロパティを表示して、左ペインで「JavaScript」＞「インクルード・パス」を選択
2. 「ソース」タブを選択してソースフォルダを「除去」してウィンドウを閉じる（「OK」ボタン）

【方法その２】

1. プロジェクトのプロパティを表示して、左ペインで「ビルダー」を選択
2. 「JavaScript バリデーター」のチェックを外してウィンドウを閉じる

【方法その３】

1. プロジェクトのプロパティを表示して、左ペインで「検証」を選択（下から 2 番目辺り）
2. 「プロジェクト固有の設定を可能にする」をチェックして「すべて使用不可にする」を押下してウィンドウを閉じる（「OK」ボタン）

## サーバ起動時の警告ログ

Tomcat 起動時に以下のような警告ログが表示されます。（新しい namespase を使用せよ、ということのようなのですが、修正してしまうと自動生成（旧 HibernateTools で動作している）がエラーになってしまうため、対処していません。）

[WARN ] [main] [] [org.hibernate.internal.util.xml.DTDEntityResolver] - HHH000223: Recognized obsolete hibernate namespace http://hibernate.sourceforge.net/. Use namespace http://www.hibernate.org/dtd/ instead. Refer to Hibernate 3.6 Migration Guide!

# 開発環境（オプション）

## Subversion クライアント【Eclipse プラグイン】

【subversive】（Eclipse プラグイン更新サイト）

http://download.eclipse.org/technology/subversive/1.0/update-site-1.0.1/

## プロパティエディタ【Eclipse プラグイン】

【PropertyEditor】（Eclipse プラグイン更新サイト）

http://propedit.sourceforge.jp/eclipse/updates/

## Java 文法チェッカー【Eclipse プラグイン】

【FindBugs】（Eclipse プラグイン更新サイト）

http://findbugs.cs.umd.edu/eclipse

## JavaScript チェッカー【Eclipse プラグイン】

nextserve7 の JavaScript は jsHint で文法チェックを行いました。  
以下に、Eclipse プラグインの導入手順と、チェック時に使用したオプションを記述します。

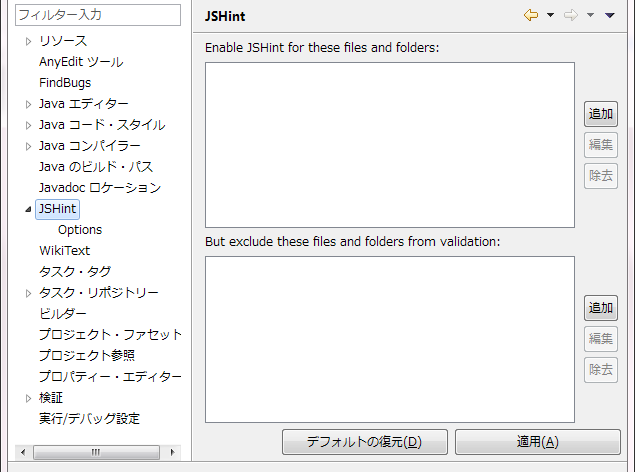
### プラグイン情報

Eclipse プラグイン更新サイト：http://github.eclipsesource.com/jshint-eclipse/updates/

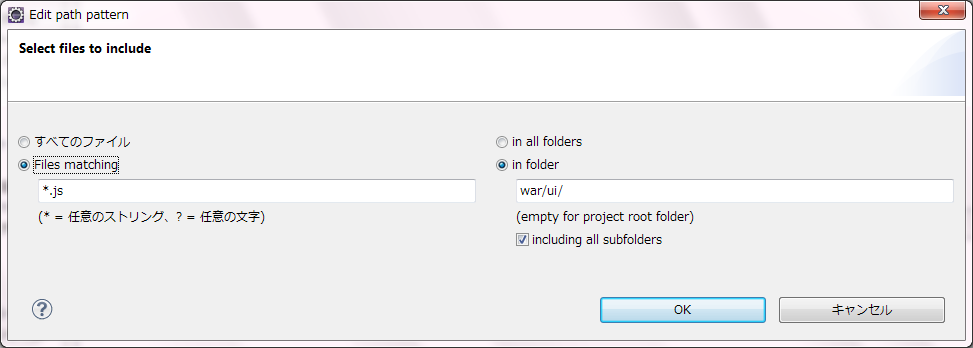
オプションの解説サイト：http://www.jshint.com/docs/

### オプション設定

1. プロジェクトのプロパティを表示し、左のペインから「JSHint」を選択します。それぞれの「追加」ボタンを押下して、処理対象ファイルと除外ファイルを設定します。



1. 処理対象ファイルの指定



「in folder」を選択して

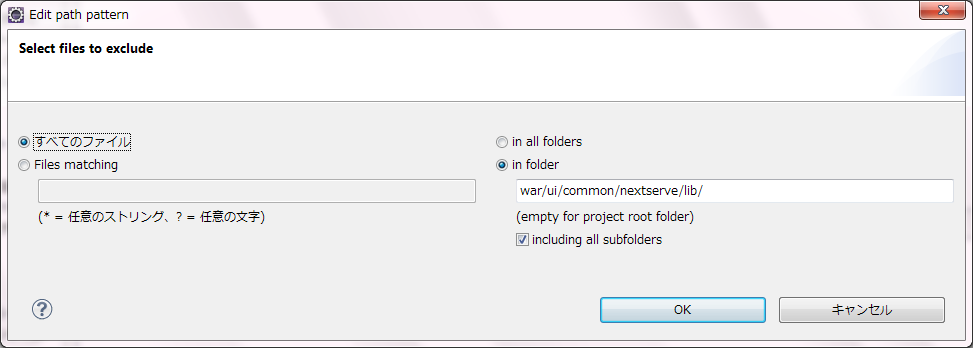
「war/ui/」を入力し

「including all subfolders」をチェック

「Files matching」を選択して

「\*.js」を入力

1. 除外ファイルの指定（2 つ設定）

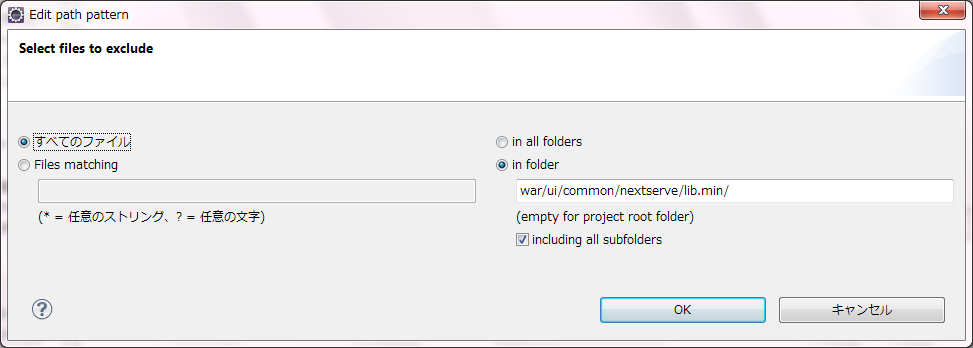


「in folder」選択して

「war/ui/common/nextserve/lib/」を入力し

「including all subfolders」をチェック

「すべてのファイル」



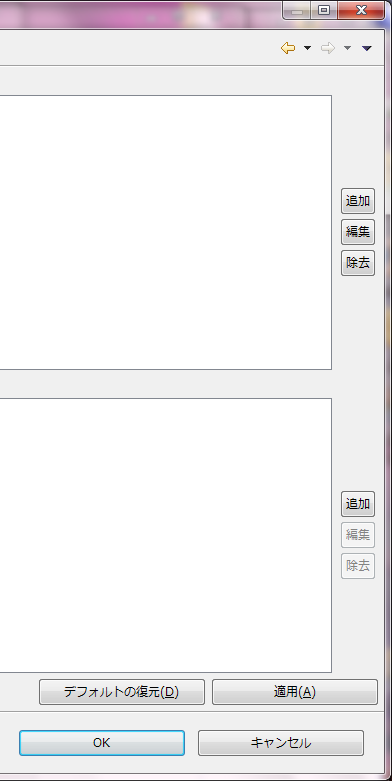
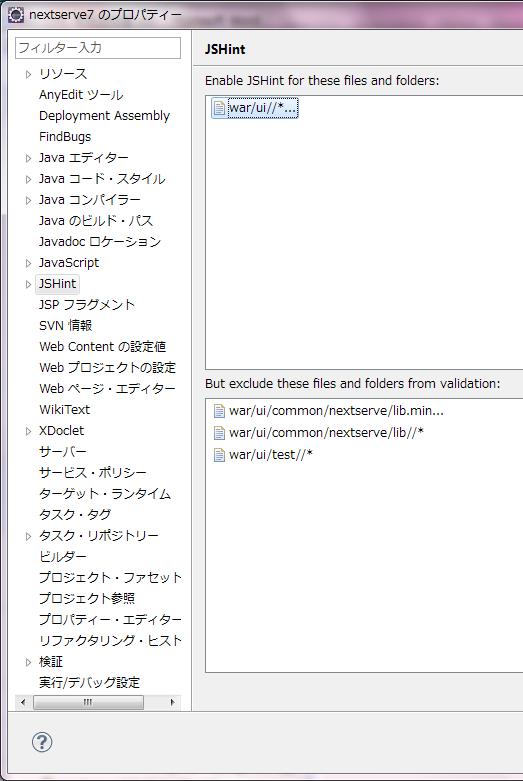
「in folder」選択して

「war/ui/common/nextserve/lib.min/」を入力し

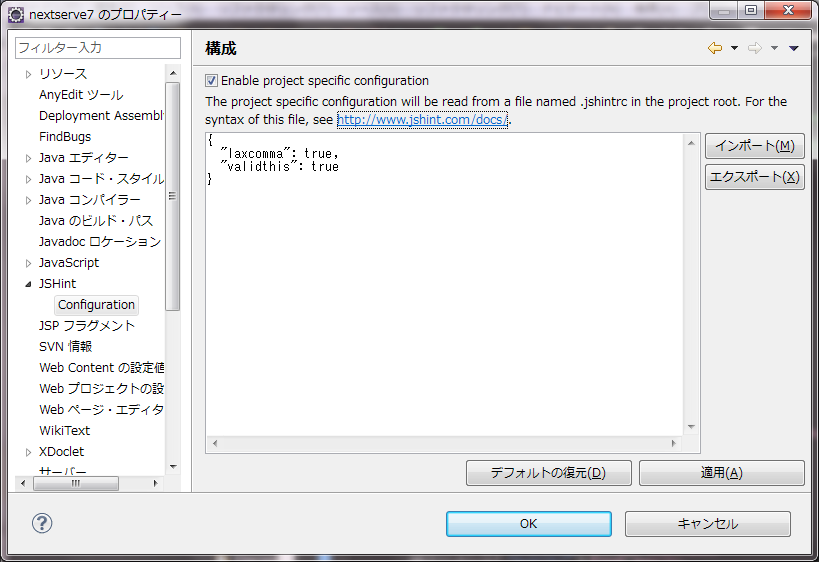
「including all subfolders」をチェック

「すべてのファイル」

1. 処理対象／除外対象ファイルの設定内容（設定後の表示内容）



1. プロジェクトのプロパティを表示し、左のペインから「JSHint」＞「Configuration」を選択します。「Enable project specific configuration」をチェックして「{ "laxcomma": true, "validthis": true }」を入力します。



# 自動生成

プロジェクト開始後にエンティティ仕様書の変更が見込まれる場合、自動生成実行用のプロジェクト（以下、\_ag）を個別に作成することをお薦めします。

エンティティ仕様書からの自動生成は既存のファイルを無条件で上書きします。DAO とエンティティはカスタマイズ不要のため、何度でも上書きでの再生成が可能ですが、app 層と ui 層については、開発が進んでからの再生成（上書き）は通常は行えません。

本章では、ソースの自動生成については以下のように管理することを想定して手順を解説します。

* \_ag プロジェクト上で自動生成を行う
* DAO とエンティティは SVN の外部参照機能を利用して実プロジェクトと共有する
* app 層と ui 層はカスタマイズ後のソースに必要な部分を手動で取り込む

## 準備：\_ag プロジェクト作成

この手順は、プロジェクト内で 1 度だけ実施します。

1. Eclipse 上で \_ag プロジェクトを新規作成（Java プロジェクト）
   * デフォルト出力フォルダー ：\_ag/bin　（「ソース」タブ）
   * ビルドパス上に必要なプロジェクト ：nextserve7 （「プロジェクト」タブ）　←実プロジェクトを指定
   * ライブラリー ：nextserve7 の Web アプリケーションライブラリー  
      「ライブラリー」タブで「ライブラリーの追加」ボタン押下  
      →「Web app ライブラリー」選択→nextserve7 プロジェクト選択
2. 自動生成に必要なファイルを準備
   * nextserve7/tools/nextserve を \_ag/tools/nextserve としてコピー
   * nextserve7/src/jdbc.properties を \_ag/src にコピー
   * \_ag/DDL ディレクトリを作成
3. \_ag/tools/nextserve/00\_xls2all.xml を ant 実行し、コンソールの末尾に「BUILD SUCCESSFUL」と表示されることを確認（自動生成完了）  
   ※エンティティ／DAO 等のファイルが指定した場所／パッケージに作成されていることを確認してください。
4. プロジェクトを「リフレッシュ」して「ビルド」し、コンパイルエラーが発生しないことを確認
5. 必要に応じて、\_ag プロジェクトを SVN に登録  
   このとき、以下のディレクトリ／ファイルは登録しないように注意してください。  
   .settings／bin／.classpath／.project
6. プロジェクト本体で自動生成を行わない場合は、nextserve7（実プロジェクト）下の tools および DDL ディレクトリを削除
7. 必要に応じて、nextserve7（実プロジェクト）側に外部参照（svn:externals）を設定

## 準備：\_ag プロジェクト取り込み

この手順は開発マシンごとに、最初に再生成を行うときに 1 度だけ実施します。

1. SVN から \_ag プロジェクトをチェックアウトします（\_ag は「Java プロジェクト」として作成しておきます）
2. Eclipse にインポートします。（「Java プロジェクト」）  
   プロジェクト作成ウィザード中に注意していただく項目は以下の通りです。
   * デフォルト出力フォルダー ： \_ag/bin　（「ソース」タブ）
   * ビルドパス上に必要なプロジェクト ： nextserve7 （「プロジェクト」タブ）　←実プロジェクトを指定
   * ライブラリー ： nextserve7 の Web アプリケーションライブラリー

「ライブラリー」タブで「ライブラリーの追加」ボタン押下  
→「Web app ライブラリー」選択→nextserve7 プロジェクト選択

1. （\_ag/tools/nextserve/00\_xls2all.xml ファイルを Ant ビューに登録しておくと便利です。）

## 自動生成（参考）

### エクセルファイルのロックを取得

複数の担当者による並行作業を回避するため、原則として、エンティティ仕様書の修正時に自動生成作業まで完了してください。

### 自動生成（\_ag）

1. \_ag/tools/nextserve/xls 下のエンティティ仕様書を修正（または最新版を取得）
2. \_ag/tools/nextserve/00\_xls2all.xml を ant 実行
3. コンソールの末尾に「BUILD SUCCESSFUL」と表示されることを確認
4. プロジェクトをリフレッシュして、ビルド（手動によるソース修正は絶対に行わないでください。次回生成時に元に戻ります）
5. コンパイルエラーが発生していないことを確認
6. \_ag プロジェクト全体について、リポジトリとの差分を確認し、意図しない箇所が変更されていないか確認

### コミット／通知

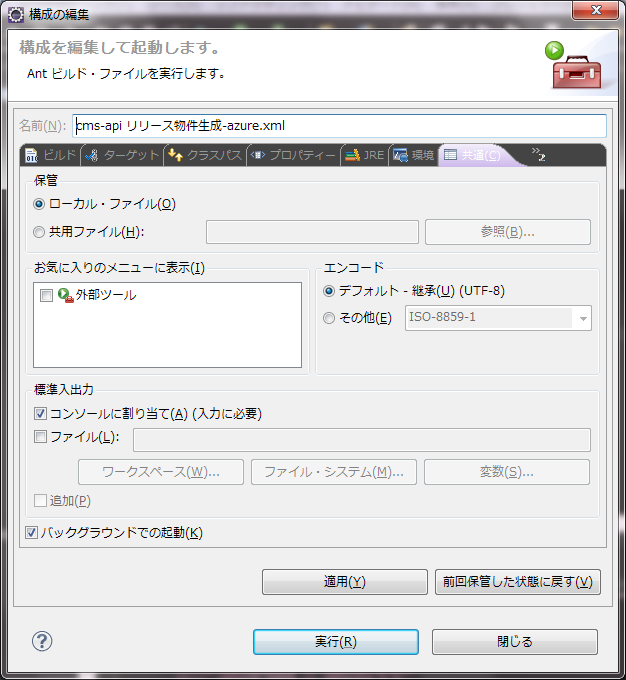
エンティティ仕様書／\_ag の修正内容をコミットして、担当者向けに修正内容を通知します。

### 補足

* 実プロジェクトの entity ディレクトリが \_ag への外部参照の場合、\_ag へのコミットと同時に実プロジェクト側にも反映されます。
* 自動生成元となるエクセルファイルは \_ag/tools/nextserve/xls に格納しておく必要があります。  
  →設計書への外部参照でも可

### TIPS

* 自動生成したファイルが文字化けする！  
  まずはテキストファイルのエンコード指定が正しく設定できているか確認します。（Java＝MS932／XML＝UTF-8）  
  Ant 実行時の実行構成を表示→右端の「共通」タブを選択  
  エンコードを変更すると改善されることが多いようです（UTF で OK な環境と MS932 で OK な環境があります）



## エクセルファイル作成

tools/nextserve/xls に配置するエクセルファイルは、必ず「HandsOn.xls」をコピーして使用してください。シートを増やす場合には、既存のシートをコピーして使用します。

ファイルが複数存在する場合は全てのファイルについて自動生成処理が実行されます。

1. 除外ファイル
   * 「User.xls」というファイル名のエクセルファイルは自動生成の対象外です（作成済みのユーザ管理機能を保護するため）
   * パッケージ名に「modules」が含まれるマッピングファイルは自動生成の対象外です（同上）。
2. 除外シート

K1 セルにジェネレータクラス名が記述されていないワークシートは自動生成の対象外になりますので、エンティティ仕様書ファイル内に ER 図やサンプルデータなど、エンティティ仕様書以外の情報を記述することが可能です。

（ジェネレータクラス名でない値が設定されている場合は自動生成エラーになりますが何も生成されないため実害はありません。）

1. 終端マーカー

A 列に「E」を入力すると、以降の行は自動生成の対象外になりますので、エンティティ仕様の下に関連データや図などを記述することが可能です。

1. カラム名とフィールド名

通常、DB 上のカラム名と Java エンティティクラスのフィールド名にはそれぞれ任意の名称を指定可能です。ただし、一覧画面上でソートのキーとして使用する項目については、以下の手順でフィールド名からカラム名への変換が行えるような名前になっている必要があります。

* + フィールド名に含まれる大文字アルファベットを小文字に変換して直前に「\_」を追加する

OK の例）カラム名＝field\_name1　フィールド名：fieldName1

NG の例）カラム名＝comment\_data　フィールド名：comment

* マルチプロジェクト対応が必要な場合は、「101 バッチ基盤簡易説明書.doc」の「4.1 エクセルファイル作成」および「8 【付録＞解説】マルチプロジェクト対応のまとめ」を参照してください。

## TIPS

# 新機能の登録

新たな機能（モジュール）を追加した場合、以下の手順が必要です。

## サーバ

### iBatis 設定ファイル

war/WEB-INF/conf/sqlmap-config.xml に作成した機能の sqlmap ファイルの設定を追加します。

### Spring ビーンロード範囲指定

生成したクラスが com.nextserve の外側に存在する場合、war/WEB-INF/conf/applicationi-base.context.xml について以下の修正を行います。

* <context:component-scan> 要素の base-package 属性に app 層のクラスが含まれるパッケージを追記

例）

<context:component-scan base-package="com.nextserve, sample, jp.xxx"/>

## 画面

新たに生成された画面は html ファイルにアクセスすることで、一通り動作しますが（例：http://localhost:8080/nextserve/ui/mySample.html）、そのままでは左メニューには表示されません。

左メニューからアクセスするためには、以下のファイルを修正してください。

### war/ui/common/js/data/urlMap.js

43 行目付近に左メニューに追加する機能（モジュール）の記述を追記します。

"redirectMap" : {

　　：

　"xxx" : { "url" : root + "/ui/xxx.html", 他モジュールの記述を参照して追記

　　"rule" : "（アクセス制限を記述）" },

},

### war/ui/common/js/data/label\_ja.js

26 行目付近に左メニューに表示するラベルを追記します。

"moduleMenu" : {

　　：

　"xxx" : "ＸＸＸ", エンティティ名：xxx、表示ラベル：ＸＸＸ

},

### war/ui/common/template/leftbar.tmpl

左メニューにエントリを追加します。

<!-- ＸＸＸ -->

<li

　name="xxx"

　data-ns-rule="${$prop.getRedirectMap( 'xxx' ).rule || ''}"

　data-ns-notrule="${$prop.getRedirectMap('xxx' ).notrule || ''}"

　class="module-menu-item titleMiddle"

>

　<div>${label.common.moduleMenu.xxx}</div>

</li>

# サンプル削除

リリース前に必ずサンプルの削除を行ってください。

## サーバ側

### ソースコード

src/sample をディレクトリごと削除します。

### applicationi-base.context.xml

war/WEB-INF/conf/applicationi-base.context.xml について以下の修正を行います。

* <context:component-scan> 要素の base-package 属性から「sample」を削除

<context:component-scan base-package="com.nextserve, sample "/>

## 画面側

### HTML ファイル

war/ui/handsOn.htmlを削除します。

### ソースコード

war/ui/handsOn をディレクトリごと削除します。

### war/ui/common/js/data/label\_ja.js

labe.common.moduleMenu 下の「handsOn」メンバを削除します。

### war/ui/common/js/data/urlMap.js

redirectMap 下の「handsOn」メンバを削除します。

### war/ui/common/template/leftbar.tmpl

name 属性に「handsOn」が設定されている <li> 要素を削除します。

# API の修正

## 認証方式

デフォルトの状態では、固定アカウント（admin、test）が有効になっています。

DB に権限／ユーザ情報を登録後は、war/WEB-INF/conf/application-security.context.xml の 118 行目付近を参照して、DB 情報での認証に切り替えてください。

## アクセス制限

デフォルトの状態では、新たに作成された API（URL＝/api/\*\*）は全てのログインユーザがアクセス可能です。必要に応じて API ごとのアクセス制限を war/WEB-INF/conf/application-security.context.xml に記述してください。

## アプリケーションログ

### ログファイル

デフォルトの状態では、アプリケーションログは C:\logs\nextserve.log に出力されます。  
src/log4j.xml を適宜修正してください。

### 固定の出力項目

デフォルト状態でのアプリケーションログの出力形式は以下の通りです。

[%d] [%-5p] [%t] [%X{ipAddress};%X{userId};%X{sessionId}] [%c] - %m%n

* %X{ipAddress}：アクセス元の IP アドレス
* %X{userId}：ログインユーザ ID
* %X{sessionId}：HTTP セッション ID

%X{...} の項目が不要の場合、web.xml 96 行目付近で定義している「MDCLogFilter」サーブレットフィルタの定義を削除可能です。

## CSRF 対策フィルタ

nextserve7 では、API の CSRF 対策として、HTTP サーブレットフィルタによる固有リクエストヘッダのチェックを行っているため、全ての API 呼び出し時には呼び出し側で、固有のリクエストヘッダを設定する必要があります。

ただし、web.xml 142 行目付近で定義している除外 URL に登録することでこのチェックを除外することが可能です。（除外すると、外部サイトから直接呼び出すことが可能になりますのでご注意ください。）

# 画面の修正

## トップ画面の URL 変更方法

デフォルトの状態では、ログイン後に表示されるトップ画面はハンズオン画面になっています。

war/ui/common/js/view.js 内の \_getTopUrl() 関数の戻り値をトップ画面の URL として使用しますので、プロジェクトごとにこの関数の処理内容を修正してください。