



TERASOLUNA Server Framework for Java(Web 版)

TERASOLUNA トラベル

ツアー予約システム

構築手順書

第 1.0 版

本ドキュメントを使用するにあたり、以下の規約に同意していただく必要があります。同意いただけない場合は、本ドキュメント及びその複製物の全てを直ちに消去又は破棄してください。

- (1)本ドキュメントの著作権及びその他一切の権利は、NTT データあるいは NTT データに権利を許諾する第三者に帰属します。
- (2)本ドキュメントの一部または全部を、自らが使用する目的において、複製、翻訳、翻案することができます。ただし本ページの規約全文、および NTT データの著作権表示を削除することはできません。
- (3)本ドキュメントの一部または全部を、自らが使用する目的において改変したり、本ドキュメントを用いた二次的著作物を作成することができます。ただし、「TERASOLUNA Server Framework for Java(Web 版) TERASOLUNA トラベルツアー予約システム構築手順書」あるいは同等の表現を、作成したドキュメント及びその複製物に記載するものとします。
- (4)前2項によって作成したドキュメント及びその複製物を、無償の場合に限り、第三者へ提供することができます。
- (5)NTT データの書面による承諾を得ることなく、本規約に定められる条件を超えて、本ドキュメント及びその複製物を使用したり、本規約上の権利の全部又は一部を第三者に譲渡したりすることはできません。
- (6)NTT データは、本ドキュメントの内容の正確性、使用目的への適合性の保証、使用結果についての確性や信頼性の保証、及び瑕疵担保義務も含め、直接、間接に被ったいかなる損害に対しても一切の責任を負いません。
- (7)NTT データは、本ドキュメントが第三者の著作権、その他如何なる権利も侵害しないことを保証しません。また、著作権、その他の権利侵害を直接又は間接の原因としてなされる如何なる請求(第三者との間の紛争を理由になされる請求を含む。)に関しても、NTT データは一切の責任を負いません。

本ドキュメントで使用されている各社の会社名及びサービス名、商品名に関する登録商標および商標は、以下の通りです。

- ・「Terasoluna/テラソルナ」は、株式会社 NTT データの登録商標です。
- ・その他の会社名、製品名は、各社の登録商標または商標です。

本書は、TERASOLUNA Server Framework for Java ver2.0.3.0 に対応しています。

変更履歴

| バージョン | 日付 | 改訂箇所 | 改訂内容 |
|-------|----------|------|------|
| 1.0 | 2010/7/1 | | 新規作成 |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

| | |
|------------------------------|---|
| 第 1 章 概要..... | 1 |
| 1.1 はじめに..... | 1 |
| 1.2 本書の目的 | 1 |
| 第 2 章 環境設定..... | 2 |
| 2.1 想定環境..... | 2 |
| 2.2 プロジェクトの実行..... | 2 |
| A) プロジェクトの実行（共通） | 2 |
| B) プロジェクトの実行（WTP 環境） | 3 |
| C) プロジェクトの実行（非 WTP 環境） | 4 |
| D) 動作確認 | 4 |

第 1 章 概要

1.1 はじめに

本資料は、TERASOLUNA Server Framework for Java (Web 版) を用いて作成した TERASOLUNA トラベルツアー予約システム(以下、ツアー予約システムと略す)を構築する手順書である。

動作環境として想定している AP サーバは、オープンソースで最もよく利用されている Apache Tomcat (以下、Tomcat と略す)を利用する。開発環境である Eclipse 上で Tomcat を動作させ、デプロイからデバッグまで実施できる WTP¹を利用した環境設定方法と WTP を利用できない環境での環境設定方法について解説していく。

1.2 本書の目的

本資料は、TERASOLUNA フレームワークを利用して作成したツアー予約システムの動作確認をする際に必要な開発環境の構築(AP サーバ等のプラットフォーム及びツアー予約システムに必要な設定)を記述したものである。当システム及び構築手順はあくまでサンプルであるため、実運用に向けた環境構築については、AP サーバ等のプラットフォームを提供している各ベンダの関連ドキュメントを参照すること。

¹ WTP とは Web Tools Platform といい、サーバサイドで開発を行うための Eclipse プラグインのセットである。サーバ管理プラグインや JavaEE アプリケーションを作成するためのプラグインなど、一通りの機能がそろっている。

第 2 章 環境設定

2.1 想定環境

本資料では以下の環境を例にして解説していく。他の環境で設定する際は本書をベースに適宜読み替えて設定していくこと。

- OS: Microsoft Windows XP Professional SP3
- JDK: SunJDK 6.0
- DB サーバ: PostgreSQL 8.4 もしくは Oracle Database 11g
- AP サーバ: Apache Tomcat 6.0.20
- 統合開発環境: Eclipse SDK 3.4.1
- Eclipse plugin: WTP 3.0.4, Apache Ant 1.7.0
- Web ブラウザ: Internet Explorer 7.0

なお、DB サーバにて使用する DB ユーザ、パスワード、接続文字列等の設定は「⑤設定ファイルの編集」に従って実施すること。

2.2 プロジェクトの実行

A) プロジェクトの実行(共通)

① ZIP ファイルの展開

toursample-javaweb.zip を適当な場所に展開する。

例) 「C:/toursample-javaweb/」

※指定されたディレクトリは固定ではないため、適宜読み替えて実行すること。

② Eclipse へのインポート

1. Eclipse 画面にて「ファイル」>「インポート」を選択する。
2. 「既存プロジェクトをワークスペースへ」を選択し「次へ」をクリックする。
3. 「ルート・ディレクトリーの選択」にチェックが入った状態で「参照」をクリックし、ブランクプロジェクトを展開した親ディレクトリを指定する。
4. 「toursample-javaweb」にチェックが入っていることを確認し、「終了」をクリックする。

※エラーが出る場合は「2.3 備考」を参照し、設定を変更すること。

③ JDBC ドライバの配置

Tomcat がデータベースに接続できるように JDBC ドライバを「<TOMCAT_HOME>/lib」にコピーする。

JDBC ドライバは付属されていないので、利用者各自で別途入手する必要がある。

- PostgreSQL: <http://jdbc.postgresql.org/download.html>
- Oracle Database: http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html

④ データベース環境の設定

1. 「setup.bat」の編集

PostgreSQL を利用する場合は「/sqls/postgres/setup.bat」、Oracle Database を利用する場合は「/sqls/oracle/setup.bat」の DB ユーザ、パスワード、接続文字列等を自端末の環境に合った値に書き換える。

※詳細は「setup.bat」を参照すること。

2. テーブル、サンプルデータの作成

「/sqls/postgres/setup.bat」もしくは「/sqls/oracle/setup.bat」を実行する(Eclipse から実行不可)。「SQL>」が表示されたら exit と入力して終了する。

※実行時パスの整合性をとるため、必ず当該ファイルが存在するディレクトリで実行すること。

⑤ 設定ファイルの編集

1. 「context.xml」の編集

「/webapps/META-INF/context.xml」の DB ユーザ、パスワード、接続文字列等を自端末の環境に合った値に書き換える。

※詳細は「context.xml」を参照すること。

2. 「sqlMapConfig.xml」の編集

PostgreSQL を利用するか、Oracle Database を利用するかで「/webapps/WEB-INF/sqlMapConfig.xml」のコメントアウト部を切り替える。

※詳細は「sqlMapConfig.xml」を参照すること。

B) プロジェクトの実行(WTP 環境)

WTP プラグインを利用する場合、以下の手順でプロジェクトを実行する。

① サーバの追加

1. Eclipse 画面にて「ウィンドウ」>「ビューの表示」>「その他」を実行し、「サーバー」を選択し「OK」をクリックする。
2. サーバービューで右クリック「新規」>「サーバー」を実行する。
3. 「Apache」>「Tomcat v6.0 サーバー」を選択し、「次へ」をクリックする。
4. 「Tomcat」を選択し、「次へ」をクリックする。
5. 自端末の環境に合った「Tomcat インストール・ディレクトリ」を選択する。
6. 「終了」をクリックする。

② プロジェクトのサーバへの追加

サーバービューで「toursample-javaweb」のプロジェクトを追加する。

③ Tomcat の起動および、実行確認

1. サーバを始動する。
2. http://localhost:8080/toursample-javaweb/にアクセスし、メニュー画面が表示されることを確認する。

C) プロジェクトの実行(非 WTP 環境)

WTP プラグインを利用しない場合、以下の手順でプロジェクトを実行する。

① Ant タスクの確認

プロジェクトの「/ant」ディレクトリには、Ant タスクが記述された「build.xml」が存在する。以下が Ant タスクの一覧である。

- clean
作成した war ファイル、コンパイルされたクラスファイルの削除を行う。
- compile
コンパイルを行う。
- native2ascii
native2ascii にてプロパティファイルのコード変換を行う。
- deploy
プロジェクトのデプロイを行う。
- createJavaDoc
JavaDoc の生成を行う。

② Ant タスクの実行

1. プロジェクトの「/ant/build.properties」を自端末の環境に合った値に書き換える。

※詳細は build.properties を参照すること。特にパス関連に関しては十分確認すること。

2. 「build.properties」の修正後、「/ant/build.xml」を右クリックし Ant タスクの「deploy」を選択して実行する。

③ Tomcat の起動および、実行確認

1. デプロイ先の Tomcat を起動する。
2. `http://localhost:8080/<context.name>/` にアクセスする。

※「<context.name>」は、build.properties 内に記述した値であり、デフォルトは toursample-javaweb となっている。

3. メニュー画面が表示されることを確認する。

D) 動作確認

① ツアー予約システムへのログイン

ログインする方法は2つある。一つは会員登録機能を利用し、新しくログインユーザを作成しそれを使用する方法、もう一つは以下の登録済みログインユーザを使用する方法である。全てのログインユーザについては<プロジェクトディレクトリ>/sqls/.../insert_customer.sql を参照のこと。

※予約履歴が存在するユーザ

ユーザ ID:00000001

パスワード:password

② ツアーの検索

ツアー検索画面にて検索条件を入力する必要があるが、以下のツアー情報が存在している。

- ・ 出発地： 北海道(検索条件一番目)
- ・ 目的地： 北海道、青森県、岩手県、宮城県、秋田県、山形県、福島県、茨城県、栃木県、群馬県
(検索条件一～十番目)
- ・ 出発日： SQL 文実行日の六日後から実行日の一年後

例) SQL 文実行日が 5/10 の場合は 2010/5/16～2011/5/15 のツアー情報が存在している。