



# TravelDream Management System

Manuale di Installazione e d'Uso

Omar Maltoni, Enrico Ghirardi

Documento di Design in merito al progetto del corso di Ingegneria del  
Software 2 - A.A. 2013/2014  
Docente di Riferimento: Mirandola Raffaella

# 1. Introduzione

## 1.1 Obiettivo del documento

Il presente testo rappresenta il manuale di installazione e d'uso per TDMS. Verrà perciò descritta la procedura per installare l'applicazione, importare il progetto nell'ambiente di sviluppo desiderato e come utilizzare l'applicazione stessa.

Il documento è rivolto al team che dovrà occuparsi del testing dell'applicazione, agli utenti e agli sviluppatori che avranno il compito di mantenere e aggiornare TDMS.

## 1.2 Riferimenti

1. TravelDream - Requirements Analysis and Specification Document:  
<https://is2-travel-dream.googlecode.com/git/Deliveries/RASD%20-%20TravelDream.pdf>
2. TravelDream - Design Document:  
<https://is2-travel-dream.googlecode.com/git/Deliveries/DD - TravelDream.pdf>

## 1.3 Struttura del documento

Il presente testo si divide nelle seguenti quattro sezioni:

1. **Introduzione:** è la sezione introduttiva del documento, illustra lo scopo del documento e i riferimenti necessari per comprenderlo ;
2. **Installazione Applicazione:** questa sezione descriverà molto brevemente la procedura di installazione dell'applicazione;
3. **Importazione del Progetto:** tutti i passaggi necessari da seguire per importare il progetto nell'ambiente di sviluppo scelto saranno descritti in questa sezione ;
4. **Utilizzo TDMS:** qui verrà descritto come utilizzare correttamente l'applicazione, in particolare ci soffermeremo sulla descrizione delle funzioni disponibili all'AU;
5. **Appendice:** questa sezione raccoglierà informazioni utili come le note di versione, per seguire i cambiamenti applicati al documento durante le fasi successive del progetto.

## 2. Installazione Applicazione

### 2.1 Requisiti installazione

Per poter installare TDMS è necessario che sulla macchina preposta allo scopo siano installati la Java Virtual Machine e un Application Server. L'unico AS compatibile al momento risulta essere Glassfish. Sarà inoltre necessario l'installazione di MySQL server che è utilizzato per la persistenza dei dati.

### 2.2 Preparazione Database

Dopo aver verificato che tutti i requisiti indicati siano soddisfatti, il primo passo per installare TDMS è la creazione di un database.

L'utente dovrà quindi creare un database e un profilo utente che possieda tutti i privilegi sul server MySQL. Il nome del database e dell'utente non hanno importanza, sarà solo necessario ricordare i dati scelti per poter configurare correttamente Glassfish.

### 2.3 Configurazione Glassfish

Completata la creazione del database sarà necessario accedere alla pagina web di configurazione di Glassfish per impostare la Pool di connessione al database e il Realm da utilizzare per la gestione dell'autenticazione degli utenti.

L'URL a cui collegarsi in un'installazione standard di Glassfish è <http://localhost:4848/>.

Una volta collegati alla console di Glassfish bisogna raggiungere la sezione JDBC Connection Pools.



Qui è necessario creare una nuova Connection Pool al database; il nome scelto non ha importanza. E' importante però specificare il tipo di Resource, cioè **javax.sql.DataSource**.

Durante questa fase sarà richiesto di inserire il nome del DB creato in precedenza con MySQL, il nome utente e la password necessari per accedervi.

Completata la creazione di una Connection Pool ci si dovrà spostare nella sezione **JDBC Resources** per crearne una nuova.

## Edit JDBC Resource

Edit an existing JDBC data source.

**JNDI Name:** jdbc/travelDream\_db\_connection

**Pool Name:**

Use the [JDBC Connection Pools](#) page to create new pools

**Deployment Order:**

Specifies the loading order of the resource at server startup. Lower numbers are loaded first.

**Description:**

**Status:** ☒ **Enabled**

La resource creata dovrà avere esattamente i dati specificati nell'immagine di cui sopra, ad esclusione del Pool Name che ovviamente sarà quello che scelto nel precedente paragrafo durante la sua creazione.

Il passo successivo nella configurazione di Glassfish è la creazione di un Realm per la gestione dell'autenticazione degli utenti. Per fare questo bisogna recarsi nella sezione server-config -> Security -> Realms e aggiungere un nuovo reame.

La configurazione di questo dovrà essere uguale a quella riportata nell'immagine seguente:

jdbc/demo\_app

jdbc/travelDream\_db\_conne

JDBC Connection Pools

DerbyPool

\_\_TimerPool

demo\_poll

travelDream\_db\_connection

JMS Resources

JNDI

JavaMail Sessions

Resource Adapter Configs

Configurations

default-config

server-config

Admin Service

Connector Service

EJB Container

HTTP Service

JVM Settings

Java Message Service

Logger Settings

Monitoring

Network Config

ORB

Security

Realms

admin-realm

certificate

file

travelDreamJdbcRealm

Audit Modules

JACC Providers

Message Security

System Properties

Thread Pools

Transaction Service

Virtual Servers

Web Container

Update Tool

Configuration Name: server-config

Realm Name: travelDreamJdbcRealm

Class Name: com.sun.enterprise.security.ee.auth.realm.jdbc.JDBCRealm

Properties specific to this Class

JAAS Context: \*

jdbcRealm

Identifier for the login module to use for this realm

JNDI: \*

jdbc/travelDream\_db\_connection

JNDI name of the JDBC resource used by this realm

User Table: \*

USERS

Name of the database table that contains the list of authorized users for this realm

User Name Column: \*

EMAIL

Name of the column in the user table that contains the list of user names

Password Column: \*

PASSWORD

Name of the column in the user table that contains the user passwords

Group Table: \*

USERS\_GROUPS

Name of the database table that contains the list of groups for this realm

Group Table User Name Column:

Name of the column in the user group table that contains the list of groups for this realm

Group Name Column: \*

group\_name

Name of the column in the group table that contains the list of group names

Password Encryption Algorithm: \*

MD5

This denotes the algorithm for encrypting the passwords in the database. It is a security risk to leave this field empty.

Assign Groups:

Comma-separated list of group names

Database User:

Specify the database user name in the realm instead of the JDBC connection pool

Database Password:

Specify the database password in the realm instead of the JDBC connection pool

Digest Algorithm:

SHA-512

Digest algorithm (default is SHA-256); note that the default was MD5 in GlassFish versions prior to 3.1

Encoding:

Encoding (allowed values are Hex and Base64)

Charset:

Character set for the digest algorithm

## 2.3 Deploy dell'Applicazione

Terminata la configurazione di Glassfish, per effettuare il deploy dell'applicazione occorrerà recarsi nella sezione Applications, selezionare l'opzione deploy e caricare il file ear incluso con il manuale.

Total # of available updates : 1

Common Tasks

Domain

server (Admin Server)

Clusters

Standalone Instances

Nodes

Applications

TravelDream-Ear-1.0-SNAPSHO

Lifecycle Modules

Monitoring Data

Resources

Concurrent Resources

Connectors

Deploy Applications or Modules

Specify the location of the application or module to deploy. An application can be in a ps

Location: ☒ Packaged File to Be Uploaded to the Server

Scegli file

nessuno selezionato

☐ Local Packaged File or Directory That Is Accessible from GlassF

Type: \*

Concluso il deploy l'applicazione sarà disponibile all'indirizzo <http://localhost:8080/TDMS/>

L'applicazione così installata è pronta all'uso, però è priva di contenuti già impostati che dovranno perciò essere inseriti dai Testers di TDMS.

Gli unici dati necessari e presenti nel database sono le credenziali dell'admin:

**EMAIL:** admin@td.it

**PASSWORD:** balotelli

## 3. Importazione del Progetto

### 3.1 Importazione dal repository

Per importare il progetto nel proprio ambiente di sviluppo basterà semplicemente effettuare il cloning del repository da Google Code.

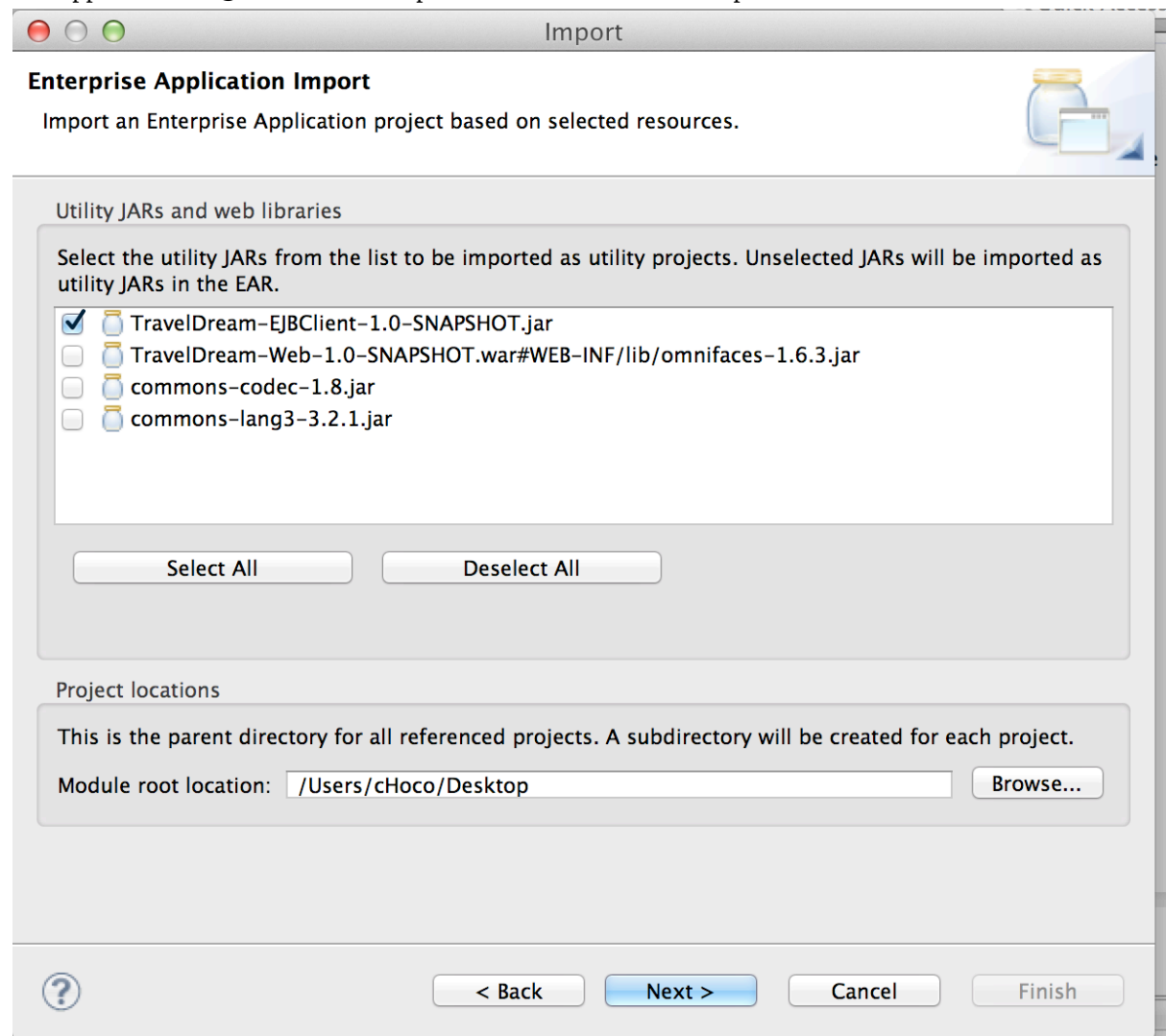
Per far ciò occorre digitare i seguenti comandi su una console (ipotizzando di aver installato una versione recente di Git)

```
git clone https://code.google.com/p/is2-travel-dream/
```

Una volta clonato basterà selezionare l'opzione importa nel proprio IDE e scegliere il tipo di progetto Maven che si occuperà dell'importazione delle varie dipendenze necessarie all'applicazione. (ATTENZIONE: l'importazione su Eclipse va effettuata due volte a causa di un bug di Eclipse stesso).

### 3.2 Importazione da file .ear

Una seconda possibilità è l'importazione diretta del file ear. Esso infatti include anche il codice sorgente dell'applicazione. Questa soluzione è possibile solo utilizzando Eclipse.



## 4. Utilizzo TDMS

### 4.1 Sezione Admin

Subito dopo l'installazione gli unici dati inseriti nel database sono quelli dell'admin. Il primo passo nell'utilizzo dell'applicazione sarà perciò effettuare il login con essi.

Nella pagina dell'admin l'utente avrà possibilità di inserire nuove credenziali di accesso nel DB per gli AU.

### 4.2 Sezione AUs

Dopo aver creato le credenziali per almeno un AU sarà possibile effettuare il login con quest'ultimo. Gli AUs sono gli utenti che si occupano della gestione dei contenuti e pertanto della creazione di Pacchetti e della creazione e modifica di Servizi.

Per prima cosa sarà necessario creare dei servizi. Per fare ciò basta selezionare il tipo di servizio desiderato e selezionare l'opzione **Aggiungi Servizio**.

I servizi creati devono possedere un codice alfanumerico di 5 che li identifica. Questo codice deve essere unico per ogni tipologia (non possono esistere due servizi di tipo Trasporto con lo stesso codice). E' possibile modificare i servizi creati inserendo il codice del servizio stesso e selezionare l'opzione **Modifica Servizio**.

Dopo aver creato tutti i servizi desiderati è possibile passare alla Creazione di un Pacchetto viaggio. Dopo aver selezionato l'opzione **Crea Pacchetto** nella pagina proposta dovranno essere inseriti i dati necessari. Purtroppo non c'è stato tempo di implementare alcuni controlli quindi bisogna fare attenzione in questa fase e seguire le istruzioni accuratamente.

I servizi che vogliamo associare al pacchetto dovranno infatti rispettare i seguenti vincoli:

- I mezzi di trasporto devono avere come destinazione quella inserita nel campo località della pagina di creazione pacchetto
- Non devono essere inseriti due volte gli stessi mezzi di trasporto nello stesso pacchetto
- Nella sezione trasporti predefiniti deve esserci solo un Trasporto predefinito che parte da una città (Se volete associare due trasporti che partono da Milano e arrivano a Roma, uno solo dei due deve essere predefinito)
- Sempre nella sezione trasporti se un trasporto scelto è l'unico che parte da una certa località per raggiungere la destinazione del viaggio, questo deve essere tra i predefiniti.
- Deve essere presente solo un hotel predefinito

Seguendo queste semplici regole i pacchetti verranno creati in modo corretto.

**ATTENZIONE:** Non esiste la concezione di viaggio di andata e viaggio di ritorno, non è quindi necessario associare a un pacchetto per Roma un viaggio che parte da Roma per tornare alla località di partenza.

Purtroppo non è stata implementata la cancellazione di un servizio, quindi anche se visibile nell'interfaccia dell'AU, non è utilizzabile.



## 4.2 Sezione Users

La gestione degli errori nell'interfaccia user è molto più avanzata. Infatti seguendo le istruzioni dell'interfaccia e gli use case illustri nel Design Document dovrebbe essere possibile utilizzare l'applicazione con facilità.

L'unica eccezione che non viene gestita è il codice del pacchetto viaggio a cui un utente viene invitato. Infatti se viene inserito un codice non esistente il sistema darà errore.

## 5. Appendici

### 5.1 Note di versione

Versione	Note
1.0	Il tempo dedicato sia alla preparazione, che alla stesura della prima versione del presente documento è sulle 2 ore, suddivise in percentuale uguale tra i due autori Omar Maltoni ed Enrico Ghirardi