Esecuzione Web ServiceRestfull allegato

1--il file war (formato di compressione applicazioni JEE)va messo in tomcat:

cartella

C:\xampp\tomcat\webapp\

Importare il database segreteria in mysql(mediante phpmyadmin)

2-accendere xampp(Tomcat+Mysql)

3-su un browser indicare URL

[**http://localhost:8080/RestProject/Rest/courseService/courses**](http://localhost:8080/RestProject/Rest/courseService/courses)

Vedremo visualizzati i dati del db

**Dentro xamp**

4-in tomcat:

nella cartella c:\xampp\tomcat\webapp\

vedremo deployata la cartella RestProject-contenente

* il file (file web.xml) , Il quale andrà a richiamare la servlet contenuta nelle librerie di jersey (il nostro provider di servizi).

quindi sara la servlet contenuta in jersey a caricare il servizio ,contenuto in

C:\xampp\Tomcat\webapps\RestProject\WEB-INF\classes\webCourses\

Jersey è il nostro Provider

Nella cartella

-C:\xampp\Tomcat\webapps\RestProject\WEB-INF\lib

troveremo

* Le librerie di jersey: framework strutturale per web services(questo estende le applicazioni web dinamiche e ha librerie per comprendere il formato json,formato trasporto dati)
* libreria gson :serializza i dati trasformandolo gli oggetti di tipo course in Stringhe
* libreria CONNECTOR J: drive mysql per java. (per la connessione diretta al RDBMS MYsql)

1. Presentation Tier(front-end)

Questo è il nostro client web

il programma utilizza a livello presentation il Patern mvc(A LIVELLO concettule), in questo modo i dati ottenuti dal business e a sua volta dal data. sono ulteriormente suddivisi in 3 parti concettualmente, cosi da poter successivamente modificare e ampliare il software

* La view è implementata nel Presentation Tier(pagina php+web server)

**il VIEW:è la parte che permette di usare il controller , E’ il modo in cui si usa il controller e quindi è la parte *"View" del Model* .Il request e il response dei dati notiamo che è definita nel presentacion insieme al web server(es applicazione clientRestful in php).**

**Per cui nel presentaion potrò definire varie view,indipendentemente dal controller e dal model,**

**basterà indicare l’uri secondo una logica Service oriented e tenere presente i dati che andrò a ricevere(dati persistenti)**

* CONTROLLER: class Controller il nostro web services

costituisce la logica di controllo

Logica di controllo:

il CONTROLLER : sara il nostra classe courseService, (l’utente richiede i datI al servizio tramite URI).ll controller le richiede al MODEL

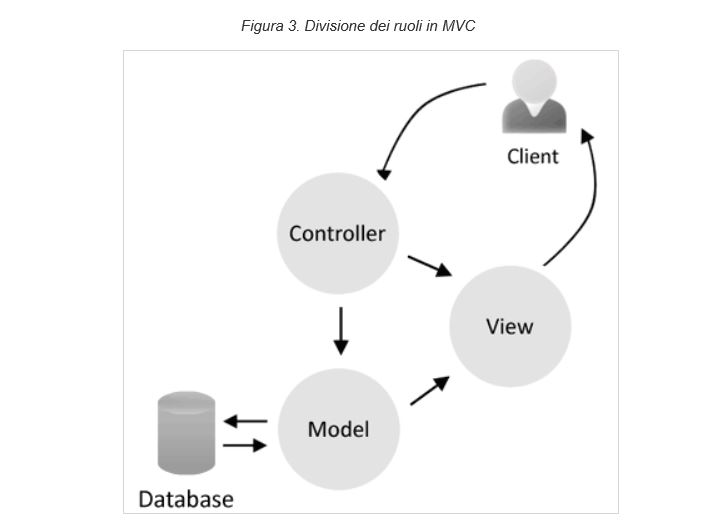
* MODEL:è LA MIA INTERFACCIA VERSO I LIVELLI INFERIORI:

bussnes logic layer()

data logic layer

Diagramma mvc:

Strato di Presentazione



Nota:Tutto questo risiede sul Application Server,il quale viene richiamato dal view mediante l’urI

Notiamo che il BUSINESS LOGIC LAYER è ulteriolmete stratificato in 3 parti:

Domain Class(dto)-OGGETTI BUSINESS

Service(Model:logica bussinesS

persistenza(dao)

i

il programma implementa a livello APPLICATION LOGIC tier la:

* Logica di servizio:bussiness

La classe model: è la nostra classe servizio, sa cosa bisogna fare ma no come, in esso sono contenute le operazioni da compiere .Rappresenta la logica business:

La classe AccessManager (é il nostro

model)contiene un arraylist di tipo Course(In questo modo definisco la struttura del db tramite gli oggetti business ).

Andando a creare un ORM(OBJECT RELATIONAL MAPPING) DEL DB.

Nel nostro caso della tabella corsi tramite un EJB

in sostanza viene caricata la tabella del db in una struttura dati dinamica dell’ applicazione java mediante il DAO(il quale a sua volta utilizza il jdbc per il collegamento con il db con la libreria java:” pure connector j”)

iNOLTRE la classe Access Manager(Model) funge da DOMAIN CONTROL SERVICES che contiene tutte le funzioni dell’applicazione le quali saranno implementate nel DAO

* Doman class (definisce le classi)

si occupa solo delle classi di dominio senza preoccuparsi del tipo di database ne dell’interfaccia.E’ la struttura dati(che dovra essere relazionato alla tabella db).

da questa classe verrano creati i dto:

DTO: Oggetto java semplice, rappresenta il singolo record della tabella e viene riempito con un record della tabella corsi e trasferito al model. Il quale ne ha fatto richiesta mediante il dao ,che conosce le operazioni da compiere per estrarre i dati e utilizzare il dto come contenitore.Naturalmente dovendo trasferire piu record dalla tabella al model istanziamo piu oggetti Courses(dto) in quale saranno raccolti in una lista, contenuta in AccessManager.

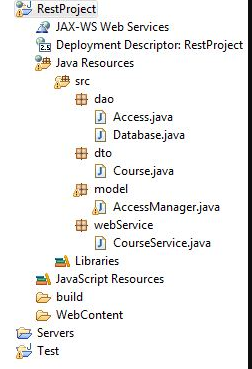
DATA LAYER TIER:

Logica accesso ai dati

* Logica accesso ai dati:

Patern dao per l’accesso ai dati, per interfacciarsi con i dati, separando cosi la logica business da quella dati

DAO:oggetto di accesso a i dati il quale sa cosa si deve fare ma no il perché(fa operazioni di crud), in esso ci sono le classi per comunicare con il db le quali utilizzano il collegamento jdbc e il connector j



****

**del modello.**