

Assignment 1

VU Entwurfsmethoden für Verteilte Systeme

SS2010

A Restful Service Framework

Dominik Dorn, 0626165, 033 534

Table of Contents

Verwendete Patterns.....	3
Basic Remoting Patterns.....	3
Server Request Handler.....	3
Invoker.....	3
Marshaller.....	3
Remoting Error.....	3
Identification Patterns.....	3
Absolute Object Reference.....	3
Lookup.....	3
Extension Patterns.....	4
Invocation Interceptor.....	4
Gang of Four Patterns.....	4
Chain of Responsibility.....	4
Proxy.....	4
Strategy.....	4
Data Access Object (DAO).....	4
Facade.....	4
J2EE Patterns.....	4
Abstract DAO.....	4
Sequenz-Diagramm.....	6
Klassendiagramm.....	7

Verwendete Patterns

In dieser Aufgabe wurden die folgenden Patterns verwendet.

Basic Remoting Patterns

Server Request Handler

Der Server Request Handler wird durch ein Servlet abgebildet. Das Servlet nimmt Anfragen ueber die Methoden GET, POST, PUT und DELETE an, konvertiert die Eingabedaten in ein einheitliches Format und uebergibt diese an den Invoker.

Bevor das Servlet aufgerufen wird, werden die Daten von 3 ServletFiltern untersucht und verarbeitet. Sollte dabei ein Fehler auftreten, wird der Request abgebrochen.

Invoker

Der Invoker nimmt die Daten des Server Request Handlers entgegen, uebergibt diese dem Marshaller und uebergibt die erhaltenen Objekte an das eigentliche Service, in unserem Fall das RestService. Das Ergebnis des RestService wird wiederum dem Marshaller uebergeben, um eine passende Representation der Daten fuer den Client zu erhalten.

Marshaller

Der Marshaller konvertiert Objekte in Representationen und umgekehrt. In unserem Fall haelt der Marshaller eine Liste von MarshallingStrategies um die entsprechenden Representationen umzuwandeln.

Remoting Error

Das Remoting Error Pattern wurde umgesetzt, um auftretende Fehler dem Client zurueck zu melden.

Identification Patterns

Absolute Object Reference

Das Absolute Object Reference Pattern ist indirekt durch die Verwendung von REST gegeben.

Lookup

Das Lookup Pattern wird durch den RegistryFilter und die Absolute Object Reference implementiert.

Extension Patterns

Invocation Interceptor

Unsere „interceptoren“ entsprechen dem Invocation Interceptor Pattern.

Gang of Four Patterns

Chain of Responsibility

Die Chain of Responsibility findet bei den verwendeten ServletFiltern Verwendung. Jeder ServletFilter uebernimmt eine kleine Ueberpruefung der Daten und stellt den nachfolgenden korrigierte+spezifizierte Werte zur Verfuegung.

Proxy

Das Proxy Pattern wird bei den geforderten „Interceptors“ verwendet. Es wird auf zwei verschiedene Arten umgesetzt:

- **Proxy mittels `Java.lang.reflect.Proxy`:** Hier wird ein Proxy generiert, der zur Laufzeit jeden Method-Call annehmen kann und entsprechende Logik ausfuehrt. Verwendet wurde dieser, auf reflection basierende Proxy zum Hinzufuegen von Log-Meldungen und zur Berechnung des Laufzeitaufwandes.
- **Proxy mittels gleichem Interface:** Hier implementiert einfach eine Klasse das gleiche Interface wie das Objekt, dessen Zugriffe abgefangen werden sollen. Verwendet wurde diese Art des Proxy bei der Klasse `JpaTransactionalDao` um DAO-Aufrufe in eine Transaktion zu packen.

Strategy

Das Strategy Pattern wurde zur Realisierung von verschiedenen Marshalling-Strategien verwendet (JSON/XML, etc.).

Data Access Object (DAO)

Das DAO-Pattern wurde zum Zugriff auf die Datenbank verwendet.

Facade

Der Invoker, der Marshaller und das RestService sind im eigentlichen Sinne Facaden, da sie die dahinterliegende Komplexitaet geschickt verstecken.

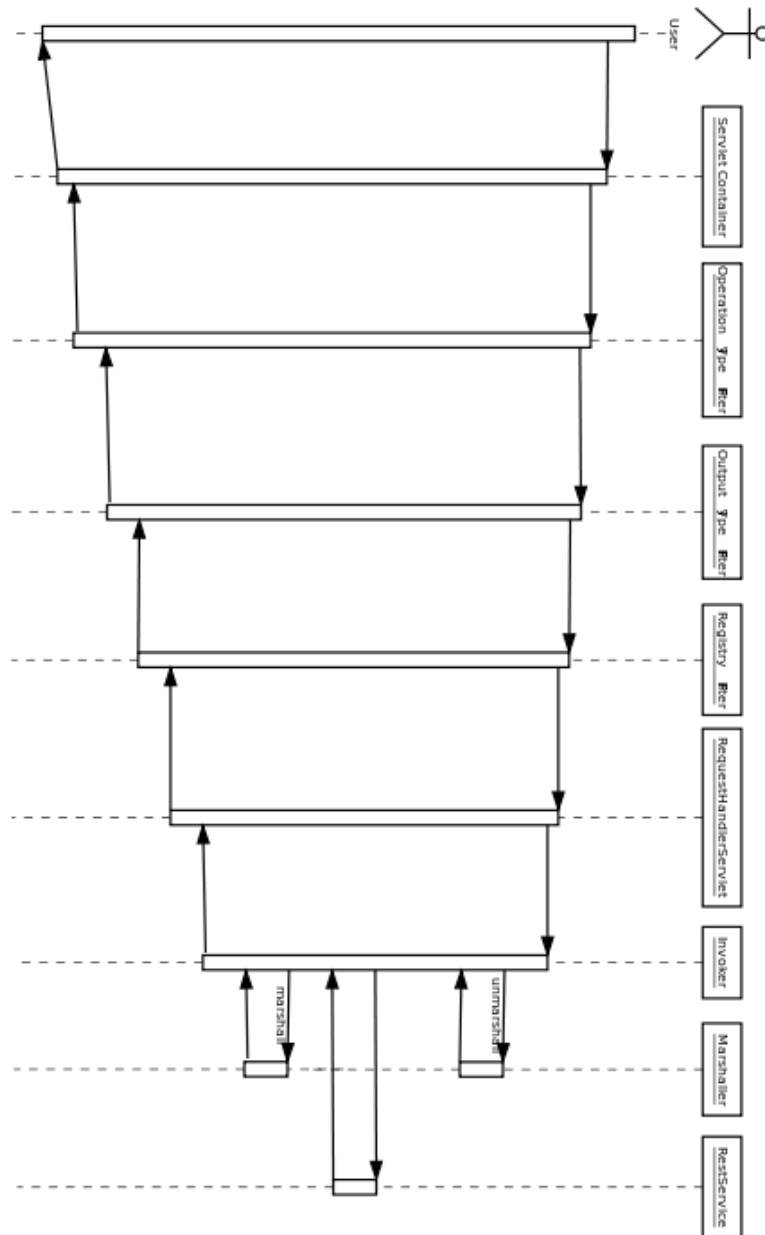
J2EE Patterns

Abstract DAO

Das aus der JavaEE Welt bekannte Abstract DAO Pattern wurde verwendet, um einen generischen Zugriff mittels JPA anzubieten. Es implementiert ein generelles DAO-Interface um so von den

anderen Komponenten verwendet werden zu koennen.

Sequenz-Diagramm



Dominik Dorn

