

# 基于 Spring + JPA 框架的电子政务基础平台

周杨川, 孙淑霞, 丁照宇

(成都理工大学, 四川 成都 610059)

**摘 要:** 电子政务基础平台主要是利用 Internet 技术实现网络行政办公的一体化, 以核心 Spring API 作为系统后台管理, 对各个窗口的数据接口进行统一管理、分配; 以持久化对象 JPA(部分是用 Hibernate)作为 ORM 对象去处理数据存储, 其所引入的数据持久化编程模型利用 Java 5 中的注释(Annotation)和对象/关系映射, 为数据持久化提供了高效、易用的编程模式。结合电子政务自身的特点, 对表现层和业务逻辑层进行严格的分离, 减少功能耦合, 整体上构建高效、可靠、移植性高的基础系统平台。

**关键词:** Spring; ORM; IoC; Hibernate; JPA

**中图分类号:** TP393

**文献标识码:** A

**文章编号:** 1673-629X(2008)04-0098-03

## E - Government Foundation Platform Based on Spring + JPA

ZHOU Yang-chuan, SUN Shu-xia, DING Zhao-yu

(Chengdu University of Technology, Chengdu 610059, China)

**Abstract:** The E-government foundation platform uses Internet technique to realize the incorporating of network government work. It is based on Spring core API and the API manages every data interface. The system uses JPA or Hibernate to persist in object relational mapping for database processing. The annotation and ORM of Java 5 provide efficient and convenient programme mode for data layer. So, view layer and logic layer will be separated for E-government's characteristic. In balance, it is an efficient and credible foundation platform.

**Key words:** Spring; ORM; IoC; Hibernate; JPA

### 0 引言

由于电子政务应用开发往往牵扯到方方面面, 很多事情都难以控制, 而且开发周期也很难把握(除非它的确是一个重量级的复杂应用)。在目前的 J2EE 框架中, EJB 框架<sup>[1]</sup>涉及太多的系统事务、安全和生命周期等方面的问题, 而在创建一个具体应用的时候需要管理多个对象实例和很多宝贵的系统资源, 占用服务器 CPU 的处理能力和内存, 在其提供强大的功能的同时也付出了性能的代价。如何实现性能高, 功耗低, 具有可伸缩性、可扩展性、跨平台性的框架体系, 是值得研究的问题。

基于轻量级的 J2EE 框架逐渐成为电子政务基础平台建设的主流技术, 像现在的 Spring, Struts, WebWork, JSF 等在各个方向都有较好的应用, 电子政务主要是利用 Internet 技术实现行政办公的网络一体化。

在众多的 J2EE 框架中, 如何选取适合电子政务的高可靠性的框架, Spring 之所以与 Struts, Webwork, Hibernate 等单层框架不同, 是因为 Spring 致力于提供一个以整体的、高效的方式构造整个应用。文中将用 Spring 作为其核心框架, 集成各种先进的技术, 开发出灵活的、易扩展的和易测试的特性, 使基础系统具有可伸缩性、可动态扩展、可维护的、可控制的以及跨平台可移植性的良性系统。

### 1 JPA 概述

JPA(Java Persistence API)是 Java EE 5.0 平台标准的 ORM 规范, 将得到 Java EE 服务器的支持。在吸收现有 ORM 框架的基础上, JPA 是一个易于使用、伸缩性强的 ORM 规范。JPA 是新引入的数据持久化编程模型, 它利用 Java 5 中的注释(Annotation)<sup>[2]</sup>和对象/关系映射, 为数据持久化提供了更简单、易用的编程方式。

JPA 由 EJB 3.0 软件专家组开发, 作为 JSR-220 实现的一部分。但它不囿于 EJB 3.0, 可以在 Web 应用、甚至桌面应用中使用。JPA 的宗旨是为 POJO 提

收稿日期: 2007-07-10

作者简介: 周杨川(1981-), 男, 重庆人, 硕士研究生, 研究方向为网络技术及其应用; 孙淑霞, 硕士生导师, 教授, 研究方向为计算机图形图像处理; 丁照宇, 硕士生导师, 教授, 研究方向为数据库应用。

供持久化标准规范。目前 Hibernate 3.2、TopLink 0.1.3 以及 OpenJpa 都提供了 JPA 的实现。JPA 的总体思想和 Hibernate, TopLink, JDO 等 ORM 框架大体一致。总的来说, JPA 包括以下 3 方面的技术:

(1) ORM 映射元数据。JPA 支持 XML 和 JDK 5.0 注解两种元数据的形式, 元数据描述对象和表之间的映射关系, 框架据此将实体对象持久化到数据库表中。

(2) JPA 的 API。用来操作实体对象, 执行 CRUD 操作, 框架在后台替人们完成所有的事情, 开发者从繁琐的 JDBC 和 SQL 代码中解脱出来。

(3) 查询语言。这是持久化操作中很重要的一个方面, 通过面向对象而非面向数据库的查询语言查询数据, 避免程序的 SQL 语句紧密耦合。

只要声称符合 JPA 标准的框架 JPA 都提供相同的访问 API, 这保证了基于 JPA 开发的企业应用能够经过少量的修改就能够在不同的 JPA 框架下运行。JPA 框架支持大数据集、事务、并发等容器级事务, 这使得 JPA 超越了简单持久化框架的局限, 在企业应用发挥更大的作用。JPA 的目标之一就是提供更加简单的编程模型, 只要使用 javax.persistence.Entity 进行注释, JPA 基于非侵入式原则设计, 因此可以很容易地和其它框架或者容器集成。JPA 自身定义了 JPQL (Java Persistence Query Language), 能够支持面向对象的高级特性, 比如类之间的继承、多态和类之间的复杂关系, 这样的支持能够让开发者最大限度地使用面向对象的模型设计企业应用, 而不需要自行处理这些特性在关系数据库的持久化。

## 2 Spring 框架简介

Spring 框架包含许多特性, 它是一个基于 IoC 和 AOP 构架的多层 J2EE 系统的框架, 并被很好地组织在以下的模块中 (如图 1 所示)。

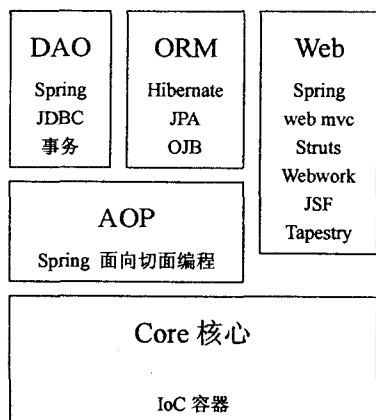


图 1 Spring 系统框架略图

Core 封装包是框架的最基础部分, 提供 IoC 和依赖注入特性。这里的基础概念是 BeanFactory, 它提供对 Factory 模式的经典实现来消除对程序性单例模式的需要, 并真正地允许你从程序逻辑中分离出依赖关系和配置。

DAO 提供了 JDBC 的抽象层, 可消除冗长的 JDBC 编码和解析数据库厂商特有的错误代码。并且, JDBC 封装包还提供了一种比编程性更好的声明性事务管理方法, 不仅仅是实现了特定接口, 而且对所有的 POJOs<sup>[2]</sup> (plain old Java objects) 都适用。

ORM 封装包提供了常用的“O/R”映射 APIs 的集成层。其中包括 JPA, JDO, Hibernate 和 iBatis。利用 ORM 封装包, 可以混合使用所有 Spring 提供的特性进行“O/R”映射。

## 3 在电子政务基础平台的应用

### 3.1 电子政务基础平台系统设计

本系统采用 Spring + Hibernate + JPA + Velocity, 其客户端、应用服务、Web 服务、数据库服务以及前置服务关系如下: 应用服务器调用操作数据库、封装业务并对外提供 WebService 服务; Web 服务器调用应用服务, 提供基于 Web 瘦客户端界面的网上政务大厅和多媒体查询系统; 各客户端调用应用服务器, 提供胖客户端界面; 前置服务调用本系统应用服务和其他异构系统服务, 成为应用桥梁; 对外接口由应用服务器提供, 通过客户端 API 调用; 系统各部分跨越政务外网、互联网和各业务部门业务网, 实现逻辑隔离和物理隔离情况下的应用信息通讯和交换。其系统框图如图 2 所示。

通过用 Spring 的声明事务管理特性, Web 应用可以做到完全事务性, 就像使用 EJB 提供的那种容器管理的事务<sup>[3]</sup>一样。业务逻辑可以通过简单的 POJO 来实现, 并利用 Spring 的 IoC 容器进行管理。对于其他的服务, Spring 的 ORM 支持可以和 JPA、Hibernate 集成起来, 可复用已经存在的映射文件与标准的 Hibernate SessionFactory 配置<sup>[4]</sup>。用控制器去无缝整合 Web 层和领域型, 消除对 ActionForms 的依赖, 避免了其他 class 为领域模型转换 HTTP 参数的需要。

### 3.2 配置文件

在 Spring 中的使用中, 配置文件必不可少, 是其 BeanFactory<sup>[5]</sup>实例化时必需的对象, 所以在 Spring 的使用中, 配置文件相当重要。在配置好数据源连接池以及所用到的类以后, 还需要配置可持久化对象映射文件, 其配置 applicationContext.xml 如下:

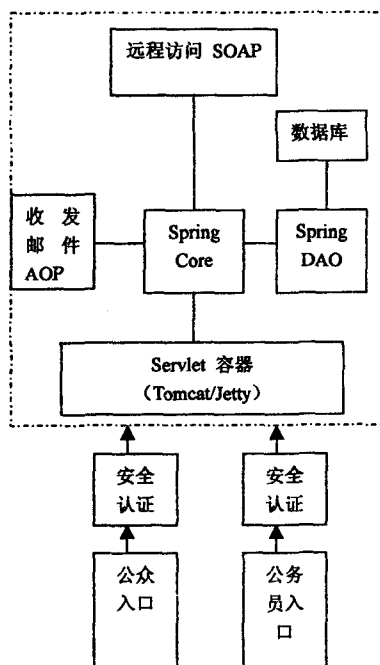


图 2 基础平台系统框图

```

<bean id="entityManagerFactory" class="org.springframework.
orm.jpa.LocalEntityManagerFactoryBean">
<property name="persistenceUnitName"
value="myJPAPersistence"/>
</bean>
<jndi:lookup id="entityManagerFactory"
jndi-name="jpa/myJPAPersistence"/>
<bean id="sessionFactory"
class="org.springframework.orm.hibernate3.LocalSession-
FactoryBean">
<property name="mappingResources">
<list>
<value>com/egov/persistence/Adviceandrefer.hbm.xml</val-
ue>
.....
<value>com/egov/persistence/user.hbm.xml</value>
</list></property></bean>

```

上面的映射文件配置完毕以后,还需要对 Spring MVC 的 Servlet 进行配置<sup>[6]</sup>,在应用的 web.xml 里配置如下:

```

<servlet>
<servlet-name>spring</servlet-name>
<servlet-class>
org.springframework.web.servlet.DispatcherServlet </servlet-
class>

```

```

<load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
<servlet-name>spring</servlet-name>
<url-pattern>*.work</url-pattern>
</servlet-mapping>

```

上面的意思是说只要是扩展名为 .work 的就交给 DispatcherServlet 进行控制处理,另外还要自己在 web-inf 目录下建立一个 spring-servlet.xml,其具体配置如下:

```

<bean id="hallMapping"
class="org.springframework.web.servlet.handler.SimpleUrlHan-
dlerMapping">
<property name="mappings">
<props> <prop key="/net/LoginBsAction.work">LoginBsAc-
tion</prop>
.....
<prop key="/net/DisplayProceedingAction.work">DisplayPro-
ceedingAction</prop>
</props></property></bean>

```

这样配置系统就可以运行了。

## 4 结 语

此电子政务基础平台在实际使用中作了压力测试,当 200 个并发请求时,Current thread busy 值在 40 个左右,说明了其高可靠性和高稳定性。由于 Spring 得到了行业的很好支持,其在 J2EE 的框架中将会发挥重要的作用。

## 参考文献:

- [1] Johnson R, Hoeller J. Expert One-on-One J2EE Development without EJB[M]. America: Wrox, 2004.
- [2] Johnson R, Arendsen A. Core Technologies[EB/OL]. 2007. <http://static.springframework.org/spring/docs/2.0.x/reference/spring-core.html>.
- [3] Walls G, Breidenbach R. Spring in action[M]. America: Manning, 2005.
- [4] 夏 昕. 深入浅出 Hibernate[M]. 北京:电子工业出版社, 2005.
- [5] Harrop R, Machacek J. Pro Spring[M]. America: Apress, 2005.
- [6] Hoeller J. Professional Java development with the Spring Framework[M]. America: Wiley, 2005.

《计算机技术与发展》欢迎投稿,欢迎订阅。