基于Netty的web控制器框架

# 背景介绍

目前我们web服务使用Tomcat + Java Servlet + SpringMVC框架，存在以下问题：

* Tomcat配置复杂，学习成本较高，难以掌握和性能优化。
* 目前我们的web业务主要是提供各种REST接口，重点在于controller的编写与提供。Spring MVC显得较重。

Netty作为高性能的网络框架，很简单就可以实现一个优秀的web容器。对于我们的业务需求，可以设想基于Netty的web容器，直接使用SpringMVC的controller机制。但是由于SpringMVC与Servlet耦合，因此难以直接将Netty与SpringMVC做整合。不过我们可以利用Spring框架的IoC特性，单单实现一个controller框架，简单易用。

基于Spring框架松耦合的特性，具体每个controller的业务实现，仍可沿用现有的DAO层不变。

# Happor原理与概念

## 框架原理说明



如上图所示，加入Netty pipeline的只有一个HttpRootController对象。该对象保存了一份由基类HttpController派生的HTTP处理对象链表。所有进来的HTTP请求依次在这些HttpController对象中走一遍，生成HTTP响应后再依次返回。

## 主要概念

### Handler

Handler表示HTTP处理器。一旦进入Handler之后，HTTP请求就不会在走到链表的下一个节点了，由当前的Handler负责生成HTTP响应。

Handler包含以下几种类型：

* HttpNormalHandler：接收请求后同步生成响应并立即返回。
* HttpAsyncHandler：接收请求后不立即返回响应，而是异步返回响应。
* HttpTransitHandler：将接收到的请求转发到其他地址，在回调中得到响应。

### Filter

Filter表示HTTP过滤器。一个Filter可以对HTTP请求和HTTP响应进行加工，然后传递给链表中的下一个节点。

Filter包含以下几种类型：

* HttpNormalFilter：正常过滤HTTP请求和HTTP响应。

# Happor使用说明

## 基本使用

详见TestWithoutSpring.java。

## 与Spring结合

详见Test.java。

## 利用注解简化使用

详见TestAnnotation.java。