

Jfinal 入门介绍

Kid Zhou

2012-11-19

Outline

- 1 简介
- 2 和传统框架的对比
- 3 官方示例代码
- 4 jfinal-ext
- 5 我们的系统架构方式和开发规范
- 6 相关资源以及广告

- 1 简介
- 2 和传统框架的对比
- 3 官方示例代码
- 4 jfinal-ext
- 5 我们的系统架构方式和开发规范
- 6 相关资源以及广告

JFinal 是基于 Java 语言的极速 web 开发框架，其核心设计目标是开发迅速、代码量少、学习简单、功能强大、轻量级、易扩展、Restful。在拥有 Java 语言所有优势的同时再拥有 ruby、python 等动态语言的开发效率。

特点

- MVC 架构，设计精巧，使用简单
- 遵循 COC 原则，零配置，无 xml
- ActiveRecord 支持，使数据库开发极致快速
- 自动加载修改后的 java 文件，开发过程中 无需重启 web server
- AOP 支持，拦截器配置灵活，功能强大
- Plugin 体系结构，扩展性强
- 多视图支持，支持 FreeMarker、JSP、Velocity
- 强大的 Validator 后端校验功能
- 功能齐全，拥有 struts2 的绝大部分功能
- 体积小仅 **225K** (编译打开 debug 参数)，且无第三方依赖
- 详细的 debug 信息,eclipse 源代码定位

先简单再容易!

- 框架足够简单, 可控性很强, 源码仅 1w 多行
- 不过度设计, 类调用层次扁平, 源码可读性强
- 仅对 `servlet` 做了很薄的包装.
- 提倡用原生 `sql` 语句描述业务

所以

你一直积累的都是 `javaee` 平台的最底层和核心的技术, 万变不离其宗

- 1 简介
- 2 和传统框架的对比
- 3 官方示例代码
- 4 jfinal-ext
- 5 我们的系统架构方式和开发规范
- 6 相关资源以及广告

JFinal WEB MVC 和 Struts 简要对比

- ① JFinal 遵循 COC 原则，零配置，无 xml，而 struts 需要配置来支持 action、result、interceptor 配置与使用。
- ② JFinal 开发效率非常之高，相对 Struts 开发效率能提升五到十倍。
- ③ JFinal 代码量非常省，相对 Struts 开发能省 50% 到 70% 代码量。
- ④ JFinal 遵循 Restful 规范，而 struts 自身未提供 Restful 支持。Struts 可以通过插件来支持 restful，但支持不彻底使用不方便。
- ⑤ JFinal 提供数据库支持，属于一站式解决方案，而 struts 仅为 WEB MVC 框架并未提供数据库支持。
- ⑥ JFinal 学习成本极低，只需两个小时学习即可上手开发，而 Struts 学习成本相对较高。
- ⑦ JFinal 相对 Struts 来说更加轻量级，JFinal 打包 jar 文件仅 225KB，而 struts 则为 1390kb。

JFinal ORM 和 Hibernate 简要对比

- ① JFinal 采用 ActiveRecord 实现数据库操作支持，较 Hibernaet 开发效率提升六到十倍。
- ② JFinal ActiveRecord 较 Hibernate 学习成本低，一小时内能上手开发。
- ③ JFinal 零配置，对数据库支持五个无特点：无 xml、无 annotation、无 getter、无 setter、无 attribute，极大降低了代码量，统计证实代码量节省 70% 到 95%。
- ④ JFinal 数据库操作完全采用原生 sql，相对 Hibernate 采用的 HQL 学习成本低，功能更强大，性能更高，稳定性好。

Jfinal 和 Spring

- ① 在 jfinal 推荐的开发方式里面, 是不采用 ioc 这样的设计模式的,jfinal 也提供了 controller 层的 aop 机制, 如果是 web 项目则可以完全不引入 spring. 其他我们习惯用 spring 来做的事务管理和以及一些第三方的 jar 整合也有相应更简单的方式来处理.
- ② 如果真要引入. 那么可以使用 jfinal 的 SpringPlugin,spring 该怎么用还是怎么用

- 1 简介
- 2 和传统框架的对比
- 3 官方示例代码
- 4 jfinal-ext
- 5 我们的系统架构方式和开发规范
- 6 相关资源以及广告

控制器

支持 FreeMarker、JSP、Velocity、JSON 等等以及自定义视图渲染

```
@Before(BlogInterceptor.class)
public class BlogController extends Controller {
    public void index() {
        setAttr("blogList",
            Blog.dao.find("select * from blog"));
    }
    public void add() {
    }
    @Before(BlogValidator.class)
    public void save() {
        getModel(Blog.class).save();
    }
}
```

控制器

```
public void edit() {  
    setAttr("blog", Blog.dao.findById(getParaToInt()  
}  
@Before(BlogValidator.class)  
public void update() {  
    getModel(Blog.class).update();  
}  
public void delete() {  
    Blog.dao.deleteById(getParaToInt());  
}  
}
```

Model

无 xml、无 annotaion、无 attribute、无 getter、无 setter、new Blog() 这行代码也不是必须

```
public class Blog extends Model {  
    public static final Blog dao = new Blog();  
}
```

Validator

API 引导式校验，比 xml 校验方便 N 倍，有代码检查不易出错

```
public class BlogValidator extends Validator {  
    protected void validate(Controller controller) {  
        validateRequiredString("blog.title",  
                                "titleMsg", " 请输入 Blog 标题!");  
        validateRequiredString("blog.content",  
                                "contentMsg", " 请输入 Blog 内容!");  
    }  
    protected void handleError(Controller controller) {  
        controller.keepModel(Blog.class);  
    }  
}
```

拦截器

在此 demo 中仅为示例，本 demo 不需要此拦截器

```
public class BlogInterceptor implements Interceptor {  
    public void intercept(ActionInvocation ai) {  
        System.out.println("Before invoking " +  
            ai.getActionKey());  
        ai.invoke();  
        System.out.println("After invoking " +  
            ai.getActionKey());  
    }  
}
```


- 1 简介
- 2 和传统框架的对比
- 3 官方示例代码
- 4 jfinal-ext**
- 5 我们的系统架构方式和开发规范
- 6 相关资源以及广告

Jfinal-ext 是对 jfinal 的一个扩充，主要利用 jfinal 的高扩展性集成第三方框架和做常用功能的封装，简化开发者的学习应用成本.

目前包含的功能

- AutoTableBindPlugin
- SqlInXmlPlugin
- JmsPlugin
- QuartzPlugin Cron4jPlugin
- ConfigPlugin
- ExcelRender
- AmchartsRender
- AutoControllerRegist
- FreeMarkerXMLRender
- CsvRender

- 分优先级加载配置文件

- 在 xml 中管理 sql 语句. 适合复杂的 sql 语句管理, 如统计分析等.
- 一般的 sql 语句尽量在 code 里面, 一目了然.

- 1 简介
- 2 和传统框架的对比
- 3 官方示例代码
- 4 jfinal-ext
- 5 我们的系统架构方式和开发规范**
- 6 相关资源以及广告

Jfinal 建议 MVC 模式

关于 Controller，Controller 主要干这几件事：

- 1: 使用 `get` 系列方法从请求中获取参数
- 2: 使用得到的参数，调用 Model 中的业务方法
- 3: 得到 Model 业务方法返回的结果并使用 `setAttr(...)` 方便 view 中使用
- 4: 根据情况 `render` 到不同的 view

关于 Model，JFinal 建议使用充血领域模型，所以 Model 中可以放入相关的业务方法。如果项目规模足够大，可以引入 Service 层来放业务。

关于 view，JFinal 建议尽量只输出数据，如果涉及到一些逻辑操作也仅仅针对于输出控制，绝不能涉及业务逻辑。

扁平的 MVC

将传统的 Action->service->dao->bo 压缩成 Controller->Model

- Controller
做参数的接收和转换, 以及视图跳转, **禁止** 出现 sql 语句
- Model
承载所有的业务逻辑 (sql 语句)

弱化类型

- ❶ 数据模型操作都采用 `model+record` 操作, **禁止** 使用 `javabean`. (虽然会牺牲一点 `ide` 提示优势, 但是更利于数据模型改变之后的维护)
- ❷ 全程键值对操作避免了数据模型改动带来的代码膨胀和维护的麻烦.
- ❸ 对外提供 `http` 服务的接口都输出 `json`.

配置文件

关键的可变配置项务必写入配置文件. 采用 **txt** 后缀名, **禁止** 使用 **properties** 文件, 避免中文乱码问题

提供数据接口的 action 统一使用 JsonRender

```
User user = new User();  
user.set("name", "zhoulei");  
user.set("age", "22");  
renderJson(user);
```

单元测试

数据接口 **必须** 编写并维护好单元测试, 保证代码的可测试性, 便于重构代码. 采用 **dbunit** 作为测试工具.

- 1 简介
- 2 和传统框架的对比
- 3 官方示例代码
- 4 jfinal-ext
- 5 我们的系统架构方式和开发规范
- 6 相关资源以及广告**

- oschina <http://www.oschina.net/p/jfinal>
- github <https://github.com/jfinal/jfinal>
- QQ 群:196337924(二号群)
- www.jfinal.com java 极速开发社区, 筹划开发中, 全站开源 (<https://github.com/jfinal/jfinal.com>), 春节左右能与大家见面.

- oschina <http://www.oschina.net/p/jfinal-ext>
- github <https://github.com/b1412/jfinal-ext>

2012 年中国最受欢迎的开源软件投票

http://www.oschina.net/project/top_cn_2012
目前依然在 top10 中. 排在 java 类框架之首.

jfinal 的开源不到 1 年时间 (之前仅属于作者所在公司网校网项目的后台技术架构), 便受到了广大 java 开发者的欢迎. 在这个被吐槽开发效率低下, 框架设计过度, 代码罗嗦的企业级 java 开发领域注入了一针兴奋剂, 它践行了大道至简的设计哲学 (现代的优秀框架难道不都这样么), 虽然他还很年轻, 可能也存在一些问题和不是太完美的地方和一些问题但是设计目标是正确的, 方向没有问题, 框架本身已经足够简单, 那有问题也是很容易解决的, 作为个人或者公司投资 jfinal, 选它作为 java 的 web 开发框架必定是值得的.