**JFinal旅程**

版本号：v1.0.0

**版本控制信息**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **修改时间** | **主要内容** | **作者** |
| v1.0.0 | 2018年05月14日 | JFinal服务端根基 | 陈华 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 止步在

1. 公司项目：接口创建，接口调试post、get？
2. 视频课程：D:\51\\_DeveloperWorkspace\JFinal\视频课程——0~13

# 资源

### 本地

D:\51\\_DeveloperWorkspace\JFinal

### 开源 D:\51\\_DeveloperWorkspace\JFinal\Demo

<https://gitee.com/javen205/weixin_guide>

<https://github.com/Javen205/JPay>

### 借鉴

<http://keeganlee.me/post/practice/20160807>

### 代码参考

https://my.oschina.net/dongcb678/blog/272036?p=4

# 搭建代理

### Localtunnel

Nodejs安装。

可以debug代码。自动将本地ip设置成外网。

Lt –subdomain hlzskkk -p 9999

自动获取外网地址，如：https://hlzskkk.localtunnel.me

### nat123

配置：D:\51\\_Utility\Tomcat

类似本地架设一个服务器，无法调试。

### Natapp

付费：9元/月

D:\51\\_DeveloperWorkspace\代理

## 云服务器

### 阿里云

### 七牛云

### VPS

[VPS](https://zh.wikipedia.org/wiki/%E8%99%9A%E6%8B%9F%E4%B8%93%E7%94%A8%E6%9C%8D%E5%8A%A1%E5%99%A8) 全称 **Virtual private server，虚拟专用服务器，**是将一台服务器分区成多个虚拟专享服务器的服务。每个VPS可配置独立IP、内存、CPU资源、操作系统。

<https://my.vultr.com> g\_na1215/Ke…123!

### ftp

本地链接

外网链接？

# 接口测试工具

### Postman

### MarkdownPad 2 + awesomium\_v1.6.6\_sdk\_win

### 爬虫

# 打包

### War

Eclipse打包成war，并存到tomcat中的webapp下，会自动解压。

# 项目构建

### 借鉴视频：D:\51\\_DeveloperWorkspace\JFinal\视频课程

创建Maven——maven-archetype-webapp——配置jfinal.jar——MainConfig——web.xml

<filter>

<filter-name>jfinal</filter-name>

<filter-class>com.jfinal.core.JFinalFilter</filter-class>

<init-param>

<param-name>configClass</param-name>

<param-value>自己config文件路径</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>jfinal</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

——extends Controller——配置route——配置main

JFinal.start("src/main/webapp", 8080, "/");

## 忽略文件

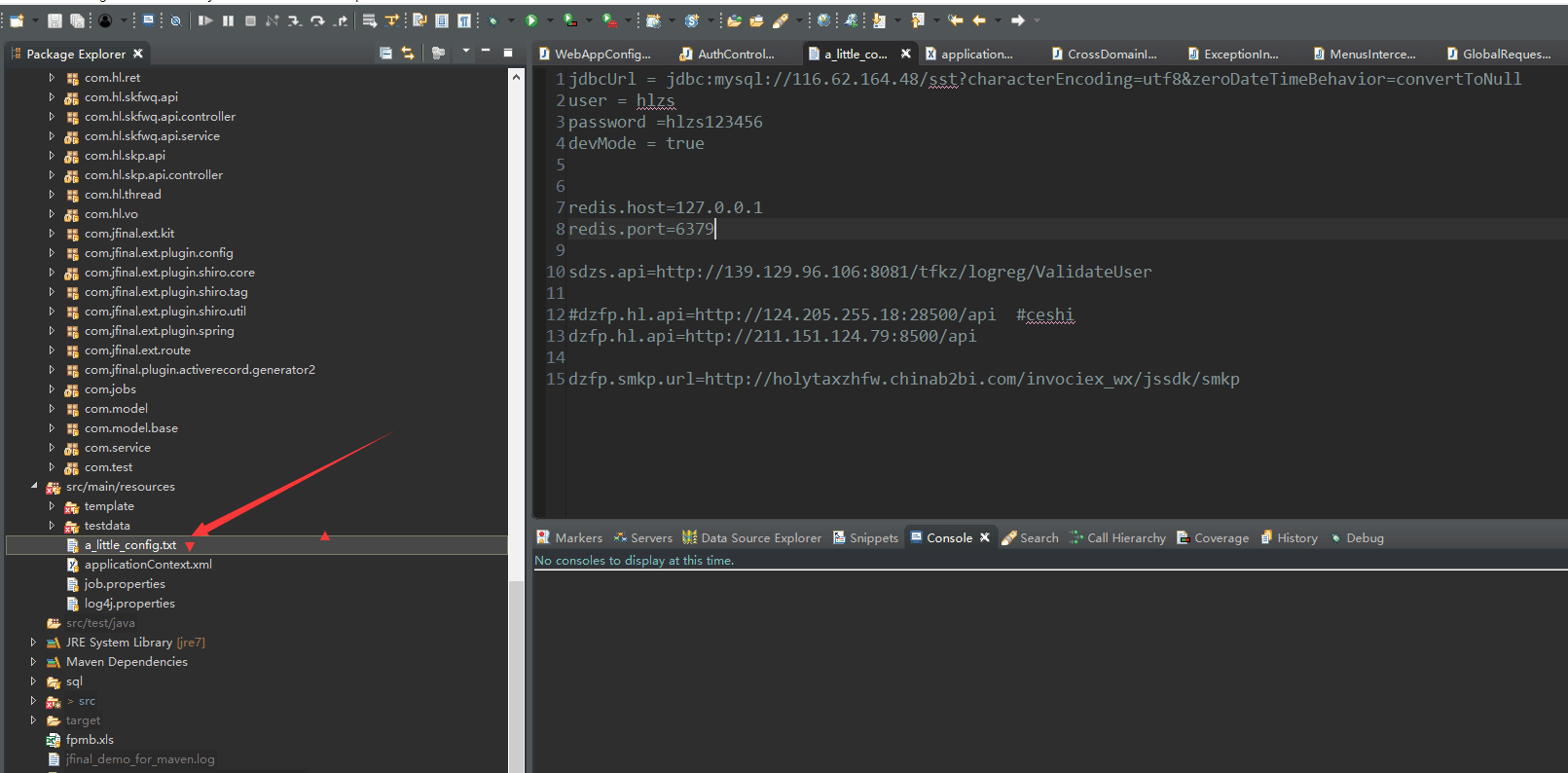
.git

.classpath

jfinal\_demo\_for\_maven.log\*

## 配置

### Redis



# 附加知识

### Maven

### Tomcat

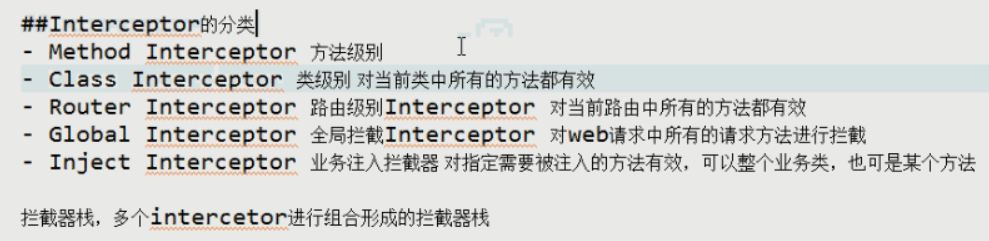
## 发布

1. 阿里云服务器
2. 搭建tomcat
3. 安装mysql

# 文档说明

## 概念

### Interceptor分类



## JFinalConfig

### Configconstant

常量配置

### configRoute

所有路由配置

路由分包

### ConfigPlugin

配置外部插件

### configEngine

3.0新加的，使我们更好的使用模板引擎

### configInterceptor

配置全局拦截器，拦截请求。如

路由级别的拦截器，有些路由上不做一些事情。

### configHandler

接管所有web请求

### afterJFinalStart

框架启动以后运行，框架启动之前启动其他功能什么

### BeforJFinalStop

## 接口回调方式

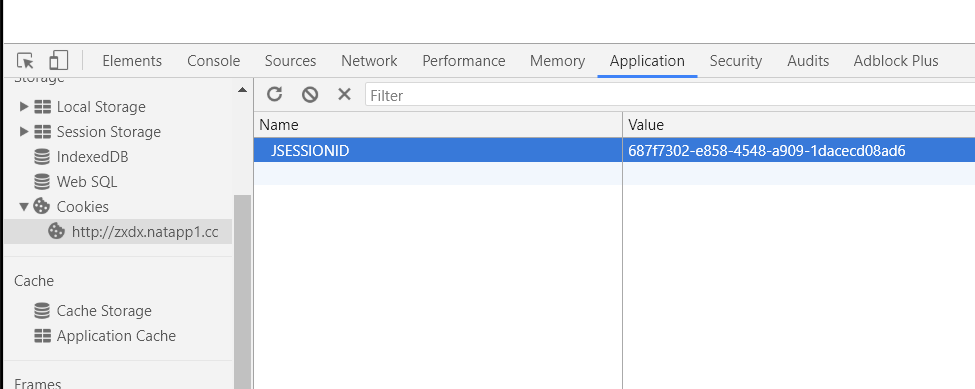
Post直接传报文，get传url地址

@ControllerBind(controllerKey = "/wx", viewPath = "/wx")

## Session

Session 对象最常见的一个用法就是存储用户的首选项

后台生成的唯一识别号



## 工具类

### PropKit

用来操作外部配置文件，读取之后是键值对应。

使用：

PropKit.get()直接获取，PropKit.getBoolean("devMode", false)

PropKit.use()在加载配置文件内容以后会将数据缓存在内存之中。

PropKit.Useless()将缓存的内容进行清除。

### HttpKit

接受下发数据

String xmlMsg = HttpKit.readData(getRequest());

## SQL管理

### 查询sql语句是否正确

打开Navicat—查询—查询编辑器

### 查询

注意加引号，全是数字可以不加，有字符必须加，所以最好加上

PayJhzf.dao.findFirst("select \* from pay\_jhzf where trade\_no = '" + tradeNo + "'");

### 类似函数的区别

**（1）xxx与xxxFirst**

这个是返回值的区别

①xxx返回List

②xxxFisrt直接查询内容

**（2）query与find**

这个也是返回值的区别

①query返回查询属性的类型

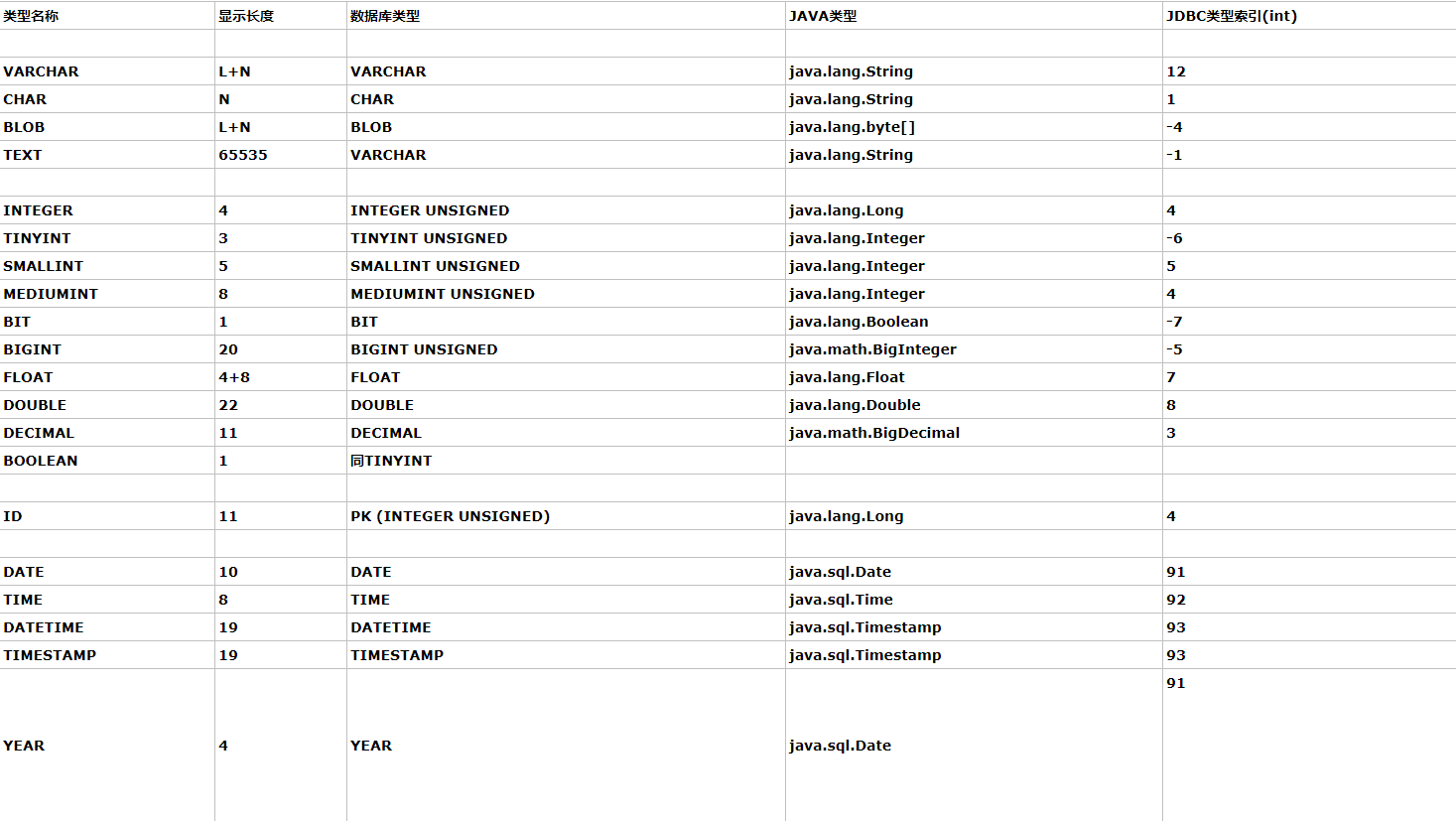
②find返回Record

**（3）set和put的区别**

①set会检测数据库是否有对应字段，没有会抛出错误

②put不会检测

### Java数据类型与mysql数据类型对应表



## Selvet

## 配置

jdbcUrl = jdbc:mysql://localhost:3306/corperation?characterEncoding=utf8&zeroDateTimeBehavior=convertToNull

user = root

password = 123456

### 自动化生成Base字段文件

DbGenerator.java

### 增删改查

数据库增删改查必须post或get请求中修改

## 请求方式

### renderData

单：直接传值

多：JSONObject json = new JSONObject();

json.put("state", state);

json.put("fwyxq", fwyxq);

renderData(json);

### renderError

404

## 功能

### 生成model

安装mysql

### 登录

# Cmd命令

### Netstat -na|more

# 项目结构

## 公司项目

### 第一层

common：用到的工具类，再分包说明相应的工具包

controllerapi：接口

service：业务

model：数据库模板

jfinal：对框架的扩展

视频项目

# 通信

## 发送

## 接受

HttpKit.readData(getRequest());

Map<String, String> rel = getParaToMap();

Long fpmainid = getParaToLong("fpmainid");

# 加密技术

## Bouncycastle

### 配置

<dependency>

<groupId>org.bouncycastle</groupId>

<artifactId>bcprov-jdk15on</artifactId>

<version>1.57</version>

</dependency>

## commons-codec

### 配置

<dependency>

<groupId>commons-codec</groupId>

<artifactId>commons-codec</artifactId>

<version>1.10</version>

</dependency>

## URLEncoder

### Java自带

URLEncoder.encode(s2tr, "utf-8");

# 快捷键

### 生成方法注释 alt + shift + j

### 格式化代码 ctrl + shift + f

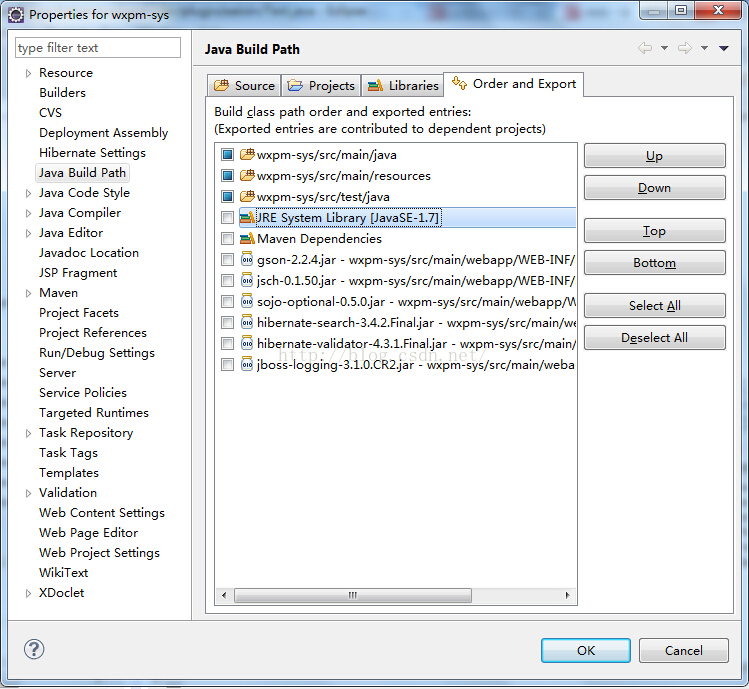
# 错误收集

### init datasource error, url: jdbc:mysql://localhost........

JFinal.start("WebContent", 80, "/", 5); 的最后一个参数去掉，在最新版的eclipse

### jar包冲突导致的

由于JDK本身和xml-apis-1.1.jar包  都有相同路径的类，但xml-apis-1.1.jar包中没有getTextContent()方法，此处我们用的是jdk  jar包中的方法



### 编码格式修改

Windows—preferences—general—workspace

### Console乱码问题

eclipse-->Run-->Run Configuration-->Common--> Console Encoding

### 支付宝—电脑网站支付

Demo错误：the superclass javax.servlet.http.httpservlet was not found on the java buil

解决方案：

1、安装tomcat

2、右击web工程-》属性或Build Path-》Java Build Path->Libraries-> Add Libray...->Server Runtime -》Tomcat Server

3、切换到Java Build Path界面中的Orader and Export，选择Tomcat。