

# ☺ 纯洁的微笑 (/)

Home (/)    Spring-Boot (/spring-boot.html)

web开发

Spring-Cloud (/spring-cloud.html)    Archives (/archives.html)

- 自定义Filter

Link (/link.html)    About (/about.html)

- 配置在application.properties中

- log配置

数据库操作

## springboot(二): web综合开发

📅 2016/02/03

- 2、添加配置文件
- 3、添加实体类和Dao
- 4、测试

thymeleaf模板

- Thymeleaf 介绍
- URL

上篇文章介绍了Spring boot初级教程: spring boot(一): 入门篇

(<http://www.ityouknow.com/springboot/2016/01/06/spring-boot-quick-start.html>), 方便大家快速入门, 了解实践Spring boot特性; 本篇文章接着上篇内容继续为大家介绍spring boot的其它特性(有些未必是spring boot体系栈的功能, 但是是spring特别推荐的一些开源技术本文也会介绍), 对于这里只是一个大概的介绍, 特别详细的使用我们会在其它的文章中来展开说明。

Gradle 构建工具

- 什么是WebJars

web开发

- 如何使用

springboot web开发非常的简单, 其中包括常用的json输出、filters、property、log等

json 接口开发

在以前的spring 开发的时候需要我们提供json接口的时候需要做那些配置呢

1. 添加 jackjson 等相关jar包
2. 配置spring Spring-Boot (/spring-boot.html)
3. 对接的方法添加@ResponseBody
4. 配置spring Spring-Cloud (/spring-cloud.html)
5. Archives (/archives.html)

就这样我们会经常由于配置错误, 导致406错误等等, spring boot如何做呢, 只需要类添加

Link (/link.html) 默认类的方法都会以json格式返回

- 配置在application.properties中

@RestController

数据库操作 HelloWorldController {

- 1、添加jar包 @RequestMapping("/getUser")

public User getUser() {

- 2、添加配置文件 User user=new User();

- 3、添加实体类和Dao user.setName("小明");

- 4、测试 user.setPassWord("xxxx");

return user;

} thymeleaf模板

- Thymeleaf 介绍

- URL

- 条件求值

如果我们需要使用页面开发只要使用 @Controller , 下面会结合模板来说明

- for循环

- 页面即原型

自定义Filter

Gradle 构建工具

WebJars

我们常常在项目中会使用filters用于记录调用日志、排除有XSS威胁的字符、执行权限验证等等。Spring

Boot自动添加了OrderedCharacterEncodingFilter和HiddenHttpMethodFilter, 并且我们可以自定义Filter。

- 如何使用

两个步骤:

1. 实现Filter接口, 实现Filter方法

2. 添加 @Configuration 注解, 将自定义Filter加入过滤链

好吧, 直接上代码 [Spring-Boot \(/spring-boot.html\)](#)

#### web开发

[Spring-Cloud \(/spring-cloud.html\)](#) [Archives \(/archives.html\)](#)

- 自定义Filter

[Link \(/link.html\)](#) [About \(/about.html\)](#)

- 配置在application.properties中

- log配置

#### 数据库操作

- 1、添加jar包

- 2、添加配置文件

- 3、添加实体类和Dao

- 4、测试

#### thymeleaf模板

- Thymeleaf 介绍

- URL

- 条件求值

- for循环

- 页面即原型

#### Gradle 构建工具

#### WebJars

- 什么是WebJars

- 为什么使用

- 如何使用

#### 参考:

**@Configuration**

```
public class WebConfiguration {
```

```
    @Bean
```

```
    public RemoteIpFilter() {
        return new RemoteIpFilter();
    }
}
```

```
web开发
```

```
Spring-Cloud (/spring-cloud.html)    Archives (/archives.html)
```

```
- @Bean
```

```
LinkMyFilterRegistrationBean() {
    // 配置在application.properties中
    FilterRegistrationBean registration = new FilterRegistrationBean();
    registration.setFilter(new MyFilter());
    registration.addUrlPatterns("/");
    registration.addInitParameter("paramName", "paramValue");
    registration.setName("MyFilter");
    registration.setOrder(1);
    return registration;
}
```

```
数据库操作
```

```
1、添加jar包
2、添加配置文件
3、添加实体类和Dao
```

```
4、测试
```

```
thymeleaf模板
```

```
public class MyFilter implements Filter {
```

```
    @Override
```

```
    public void destroy() {
```

```
        // TODO Auto-generated method stub
    }
```

```
    @Override
```

```
    public void doFilter(ServletRequest srequest, ServletResponse sresponse, FilterChain
```

```
    throws IOException, ServletException {
```

```
        // TODO Auto-generated method stub
        HttpServletRequest request = (HttpServletRequest) srequest;
        System.out.println("this is MyFilter,url :"+request.getRequestURI());
        filterChain.doFilter(srequest, sresponse);
    }
```

```
参考:
```

```
@Override
```

```
public void init(FilterConfig arg0) throws ServletException {
```

```
// TODO Auto-generated method stub
```

```
}
}
}
```

Home (/)      Spring-Boot (/spring-boot.html)

web开发

Spring-Cloud (/spring-cloud.html)      Archives (/archives.html)

自定义Property

- 自定义Filter

Link (/link.html)      About (/about.html)

在web开发的过程中，我经常需要自定义一些配置文件，如何使用呢

- 配置在application.properties中

配置在application.properties中

数据库操作

- 1、添加jar包

com.neo.title=纯洁的微笑

- 2、添加配置文件

com.neo.description=分享生活和技术

- 3、添加实体类和Dao

- 4、测试

自定义配置类

thymeleaf模板

- Thymeleaf 介绍

@Component

public class NeoProperties {

- 条件求值

@Value("\${com.neo.title}")

- for循环

private String title;

- 页面即原型

@Value("\${com.neo.description}")

private String description;

Gradle 构建工具

WebJars

//省略getter settet方法

- 什么是WebJars

- 为什么使用

- 如何使用

参考:

log配置

## 配置输出的地址和输出级别

```
logging.path=/user/local/log
logging.level.org.springframework=ERROR
```

```
logging.level.org.springframework.web=INFO
```

```
logging.level.org.hibernate=ERROR
```

Spring Cloud (/spring-cloud.html) Archives (/archives.html)

- 自定义Filter

path=/api/... logging.level.org.springframework.web=INFO

- 配置在application.properties中

## 数据库操作

### 数据库操作

在这里我重点描述mysql、spring data jpa的使用，其中mysql就不用说了大家很熟悉，jpa是利用Hibernate生成各种自动化的sql，如果只是简单的增删改查，基本上不用手写了，spring内部已经帮大家封装实现了。

- 4、测试

下面简单介绍一下如何在spring boot中使用

### thymeleaf模板

- Thymeleaf 介绍

## 1、添加jar包

- 条件求值

- for循环

- 页面即原型

```
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-data-jpa</artifactId>
```

### Gradle 构建工具

- 什么是WebJars

```
<groupId>mysql</groupId>
```

```
<artifactId>mysql-connector-java</artifactId>
```

```
</dependency>
```

- 如何使用

参考：

## 2、添加配置文件

```
spring.datasource.url=jdbc:mysql://localhost:3306/test
spring.datasource.username=root
spring.datasource.password=root
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
```

Home (/) Spring-Boot (/spring-boot.html)

web开发

```
spring.jpa.properties.hibernate.hbm2ddl.auto=update
```

```
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect
```

```
spring.jpa.show-sql= true
```

自定义Site

Link (/link.html)

自定义Property

About (/about.html)

Archives (/archives.html)

其实这个hibernate.hbm2ddl.auto参数的作用主要用于：自动创建|更新|验证数据库表结构,有四个直：

- log配置

### 数据库操作

- 1. create：每次加载hibernate时都会删除上一次的生成的表，然后根据你的model类再重新生成新表，哪怕两次没有任何改变也要这样执行，这就是导致数据库表数据丢失的一个重要原因。
- 2. create-drop：每次加载hibernate时根据model类生成表，但是sessionFactory一关闭，表就自动删除。
- 3. update：最常用的属性，第一次加载hibernate时根据model类会自动建立起表的结构（前提是先建立好数据库），以后加载hibernate时根据 model类自动更新表结构，即使表结构改变了但表中的行仍然存在不会删除以前的行。要注意的是当部署到服务器后，表结构是不会被马上建立起来的，是要等 应用第一次运行起来后才

### Gradle构建工具

#### WebJars

- 4. validate：每次加载hibernate时，验证创建数据库表结构，只会和数据库中的表进行比较，不会创建新表，但是会插入新值。
- 为什么使用

如何使用 dialect 主要是指指定生成表名的存储引擎为InnoDB

参考：show-sql 是否打印出自动生产的SQL，方便调试的时候查看

## 3、添加实体类和Dao

**@Entity**

```
public class User implements Serializable {
```

```
    private static final long serialVersionUID = 1L;
```

```
    @Id
```

```
    @GeneratedValue
```

```
    private Long id;
```

```
    @Column(nullable = false, unique = true)
```

```
    private String userName;
```

```
    @Column(nullable = false)
```

```
    private String password;
```

```
    @Column(nullable = false, unique = true)
```

```
    private String email;
```

```
    @Column(nullable = true, unique = true)
```

```
    private String nickName;
```

```
    @Column(nullable = false)
```

```
    private String regTime;
```

```
    //省略getter setter方法、构造方法
```

```
    //省略getter setter方法、构造方法
```

```
    } - Thymeleaf 介绍
```

```
    - URL
```

```
    - 条件求值
```

dao只要继承JpaRepository类就可以，几乎可以不用写方法，还有一个特别有尿性的功能非常赞，就是可以根据方法名来自动的生产SQL，比如 findByUserName 会自动生产一个以 userName 为参数的查询方法，比如 findAll 会自动会查询表里面的所有数据，比如自动分页等等。。

```
    - for循环
```

```
    - 负面即原型
```

查询方法，比如 findAll 会自动会查询表里面的所有数据，比如自动分页等等。。

**Gradle 构建工具**

WebJars  
Entity中不映射成列的字段得加@Transient 注解，不加分解也会映射成列

```
    - 什么是WebJars
```

```
    - 为什么使用
```

```
public interface UserRepository extends JpaRepository<User, Long> {
```

```
    - 如何使用
```

```
    User findByUserName(String userName);
```

```
    User findByUserNameOrEmail(String username, String email);
```

```
    参考
```



## 4、测试

```

@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations = {"classpath:applicationContext.xml", "classpath:spring-boot.html"})
public class UserRepositoryTests {
    @Autowired
    private UserRepository userRepository;

    @Test
    public void test() throws Exception {
        Date date = new Date();
        DateFormat dateFormat = DateFormat.getDateInstance(DateFormat.LONG, DateF
        String formattedDate = dateFormat.format(date);

        userRepository.save(new User("aa1", "aa@126.com", "aa", "aa123456",formattedDate))
        userRepository.save(new User("bb2", "bb@126.com", "bb", "bb123456",formattedDate)
        userRepository.save(new User("cc3", "cc@126.com", "cc", "cc123456",formattedDate));

        Assert.assertEquals(9, userRepository.findAll().size());
        Assert.assertEquals("bb", userRepository.findByUserNameOrEmail("bb", "cc@126.com").
        userRepository.delete(userRepository.findByUserName("aa1"));
    }
}

```

**thymeleaf模板**

- Thymeleaf 介绍
- URL
- 条件求值
- for循环
- 页面即原型

**Gradle 构建工具**

**WebJars**

当让 spring data jpa 还有很多功能，比如封装好的分页，可以自己定义SQL，主从分离等等，这里就不详细讲了

- 什么是WebJars
- 为什么使用
- 如何使用

**thymeleaf模板**

Spring boot 推荐使用来代替jsp,thymeleaf模板到底是什么来头呢,让spring大哥来推荐,下面我们来聊聊

## Thymeleaf介绍 Spring-Boot (/spring-boot.html)

### web开发

Thymeleaf是一款用于渲染XML/XHTML/HTML5内容的模板引擎。类似JSP, Velocity, FreeMarker等, 它也可以轻易的与Spring MVC等Web框架进行集成作为Web应用的模板引擎。与其它模板引擎相比, Thymeleaf最大的特点是能够在浏览器中打开并正确显示模板页面, 而不需要启动整个Web应用。

- 配置在application.properties中

好了, log配置, 我们说了我们已经习惯使用了什么 velocity, FreMaker, beetle之类的模版, 那么到底好在哪里呢?

### 数据库操作

Thymeleaf是与众不同的, 因为它使用了自然的模板技术。这意味着Thymeleaf的模板语法并不会破坏文档的结构, 模板依旧是有效的XML文档。模板还可以用作工作原型, Thymeleaf会在运行期替换掉静态值。

Velocity与FreeMarker则是连续的文本处理器。下面的代码示例分别使用Velocity、FreeMarker与Thymeleaf打印出一条消息:

1. 添加jar包
2. 添加配置文件
3. 添加实体类和Dao

- 4、测试

### thymeleaf模板

Velocity: `<p>${message}</p>`

FreeMarker: `<p>${message}</p>`

Thymeleaf: `<p th:text="${message}">Hello World!</p>`

- 条件求值

### for循环

\* 注意, 由于Thymeleaf使用了XML DOM解析器, 因此它并不适合于处理大规模的XML文件。\*

- 页面即原型

### Gradle 构建工具

### WebJars

JRL在Web应用模板中占据着十分重要的地位, 需要特别注意的是Thymeleaf对于URL的处理是通过

语法@{...}来处理的。Thymeleaf支持绝对路径URL:

- 如何使用

### 参考:

`<a th:href="@{http://www.thymeleaf.org}">Thymeleaf</a>`

## 条件求值

```
<a th:href="@{/login}" th:unless=${session.user != null}>Login</a>
Home (/) Spring-Boot (/spring-boot.html)
```

### web开发

for循环 循环接口开发

```
for循环 Cloud (/spring-cloud.html) Archives (/archives.html)
```

- 自定义Filter

Link (/link.html) About (/about.html)

```
<tr th:each="prod, status" th:parent="tr">
```

- 配置在application.properties中

```
<td th:text="${prod.name}">Onions</td>
```

- log配置

```
<td th:text="${prod.price}">2.41</td>
<td th:text="${prod.inStock}? #{true} : #{false}">yes</td>
```

### 数据库操作

- 1、添加jar包

- 2、添加配置文件

就列出这几个吧

- 3、添加实体类和Dao

- 4、测试

## 页面即原型

thymeleaf模板

在Web开发过程中一个绕不开的话题就是前端工程师与后端工程师的写作，在传统Java Web开发过程中，前端工程师和后端工程师一样，也需要安装一套完整的开发环境，然后在各类Java IDE中修改模板、静态资源文件，启动/重启/重新加载应用服务器，刷新页面查看最终效果。

- for循环

但实际上前端工程师的职责更多应该关注于页面本身而非后端，使用JSP，Velocity等传统的Java模板

引擎很难做到这一点，因为它们必须在应用服务器中渲染完成后才能在浏览器中看到结果，而

Thymeleaf从根本上颠覆了这一过程，通过属性进行模板渲染不会引入任何新的浏览器不能识别的标

签，例如JSP中的`<%= %>`不会在Tag内部写表达式。整个页面直接作为HTML文件用浏览器打开，几乎就可

以看到最终的效果，这大大解放了前端工程师的生产力，它们的最终交付物就是纯的

HTML如何使用JavaScript文件。

参考：

## Gradle 构建工具

spring 项目建议使用Gradle进行构建项目，相比maven来讲 Gradle更简洁，而且gradle更时候大型复杂项目的构建。gradle吸收了maven和ant的特点而来，不过目前maven仍然是Java界的主流，大家可以先了解了解。

[Home \(/\)](#) [Spring-Boot \(/spring-boot.html\)](#)

一个使用gradle配置的项目  
web开发

[Spring-Cloud \(/spring-cloud.html\)](#) [Archives \(/archives.html\)](#)

- 自定义Filter

[Link \(/link.html\)](#) [About \(/about.html\)](#)

自定义Property  
- 配置在application.properties中

- log配置

### 数据库操作

- 1、添加jar包
- 2、添加配置文件
- 3、添加实体类和Dao
- 4、测试

### thymeleaf模板

- Thymeleaf 介绍
- URL
- 条件求值
- for循环
- 页面即原型

### Gradle 构建工具

#### WebJars

- 什么是WebJars
- 为什么使用
- 如何使用

#### 参考:

```

buildscript {
    repositories {
        maven { url "http://repo.spring.io/libs-snapshot" }
        mavenLocal()
    }
}

dependencies {
    compile("org.springframework.boot:spring-boot-gradle-plugin:1.3.6.RELEASE")
}

- 自定义Filter
} Link (/link.html)    About (/about.html)
- 自定义Property

- 配置在application.properties中
apply plugin: 'java' //添加 Java 插件, 表明这是一个 Java 项目
apply plugin: 'spring-boot' //添加 Spring-boot支持
apply plugin: 'war' //添加 War 插件, 可以导出 War 包
- 数据库操作
apply plugin: 'eclipse' //添加 Eclipse 插件, 添加 Eclipse IDE 支持, IntelliJ Idea 为 "idea"
- 添加jar包

2. 添加配置文件
war {
    - 3.baseName = 'Favorites'
    - 4.version = '0.1.0'
}

thymeleaf模板
- Thymeleaf 介绍
sourceCompatibility = 1.7 //最低兼容版本 JDK1.7
targetCompatibility = 1.7 //目标兼容版本 JDK1.7
- 条件求值
repositories { // Maven 仓库
    for循环
    mavenLocal() //使用本地仓库
    - 页面即原型
    mavenCentral() //使用中央仓库
}

Gradle构建工具 "http://repo.spring.io/libs-snapshot" } //使用远程仓库

WebJars
- 什么是WebJars
dependencies { // 各种 依赖的jar包
    - 为什么使用
    compile("org.springframework.boot:spring-boot-starter-web:1.3.6.RELEASE")
    - 如何使用
    compile("org.springframework.boot:spring-boot-starter-thymeleaf:1.3.6.RELEASE")
    compile("org.springframework.boot:spring-boot-starter-data-jpa:1.3.6.RELEASE")
}

参考:
compile group: 'mysql', name: 'mysql-connector-java', version: '5.1.6'
compile group: 'org.apache.commons', name: 'commons-lang3', version: '3.4'
compile("org.springframework.boot:spring-boot-devtools:1.3.6.RELEASE")

```

```
compile("org.springframework.boot:spring-boot-starter-test:1.3.6.RELEASE")
compile 'org.webjars.bower:bootstrap:3.3.6'
    compile 'org.webjars.bower:jquery:2.2.4'
compile("org.webjars:vue:1.0.24")
```

Home (/) Spring-Boot (/spring-boot.html)

web开发

Spring-Cloud (/spring-cloud.html) Archives (/archives.html)

自定义Filter

Link (/link.html) About (/about.html)

- 配置在application.properties中

- log配置

数据库操作

WebJars

- 1、添加jar包

- 2、添加配置文件

WebJars是一个很神奇的东西，可以让大家以jar包的形式来使用前端的各种框架、组件。

- 3、添加实体类和Dao

- 4、测试

什么是WebJars

thymeleaf模板

什么是WebJars?

WebJars是将客户端（浏览器）资源（JavaScript, Css等）打成jar包文件，以对资源进行统一依赖管理。WebJars的jar包部署在Maven中央仓库上。

- 条件求值

为什么使用

- 页面即原型

我们在开发Web项目的时候会使用像Maven, Gradle等构建工具以实现对jar包版本依赖管理，以及项目的自动化管理，但是对于JavaScript, Css等前端资源包，我们只能采用拷贝到webapp下的方式，这样做就无法对这些资源进行依赖管理。那么WebJars就提供给我们这些前端资源的jar包形势，我们就可以进行依赖管理。

如何使用

WebJars

我们可以在开发Web项目的时候使用像Maven, Gradle等构建工具以实现对jar包版本依赖管理，以及项目的自动化管理，但是对于JavaScript, Css等前端资源包，我们只能采用拷贝到webapp下的方式，这样做就无法对这些资源进行依赖管理。那么WebJars就提供给我们这些前端资源的jar包形势，我们就可以进行依赖管理。

- 如何使用

如何使用

1、 WebJars主官网 (<http://www.webjars.org/bower>) 查找对于的组件，比如Vuejs

```
<dependency>
  <groupId>org.webjars.bower</groupId>
  <artifactId>vue</artifactId>
```

version>1.0.2</version> Spring Boot (/spring-boot.html)

```
</dependency>
```

web开发

Spring Cloud (/spring-cloud.html) Archives (/archives.html)

端口开发

2、页面引入Filter

Link (/link.html) About (/about.html)

自定义Property

```
<link href="@{webjars.bootstrap/3.3.6/dist/css/bootstrap.css}" rel="stylesheet"></link>
```

- log配置

数据库操作

就可以正常使用了!

- 1、添加jar包

示例代码-github (https://github.com/ityouknow/spring-boot-examples)

- 3、添加实体类和Dao

示例代码-码云 (https://gitee.com/ityouknow/spring-boot-examples)

1、测试

thymeleaf模板

参考: thymeleaf 介绍

- URL

新一代的模板引擎Thymeleaf (http://www.tianmaying.com/tutorial/using-thymeleaf)

- for循环

Spring Boot参考指南-中文版 (https://qbgbook.gitbooks.io/spring-boot-reference-guide-

zh/content/)

Gradle 构建工具

WebJars

- 什么是WebJars

作者: 纯洁的微笑

出处: 如何使用

www.ityouknow.com (http://www.ityouknow.com)

版权所有, 欢迎保留原文链接进行转载: )



Home (/) Spring-Boot (/spring-boot.html)

## web开发

Spring-Cloud (/spring-cloud.html) Archives (/archives.html)

- 自定义Filter

Link (/link.html) About (/about.html)

- 配置在application.properties中

- log配置

## 数据库操作

- 1、添加jar包

- 2、添加配置文件

- 3、添加实体类和DAO (转载请注明出处 纯洁的微笑-ityouknow (<http://www.ityouknow.com>))

- 4、测试

Show Disqus Comments

## thymeleaf模板

- Thymeleaf 介绍

13 (<https://github.com/ityouknow/blog-comments/issues/153>) 条评论

未登录用户 ▾

- 条件求值

- for循环



- 页面即原型

Gradle 构建工具

## WebJars

- 什么是WebJars

- 支持 Markdown 语法 (<https://guides.github.com/features/mastering-markdown/>)

- 为什么使用

- 如何使用

使用 Github 登录

## 参考:



gaoKyle (<https://github.com/gaoKyle>) 发表于 3 个月前

很不错







Han-China (<https://github.com/Han-China>) 发表于 3 个月前



Home 再看一遍 Spring-Boot (/spring-boot.html)

## web开发



Spring-Cloud (/spring-cloud.html) Archives (/archives.html)

Jedeah (<https://github.com/Jedeah>) 发表于 3 个月前



自定义Filter

Link (/link.html) About (/about.html)

- 配置在application.properties中

- log配置

## 数据库操作

lessoon (<https://github.com/lessoon>) 发表于 2 个月前



- 1、添加jar包

- 2、添加配置文件

- 3、添加实体类和Dao

- 4、测试

## thymeleaf模板

- Thymeleaf 介绍

- URL



条件求值

for循环

ityouknow (<https://github.com/ityouknow>) 发表于 2 个月前



## Gradle 构建工具

## WebJars

- 什么是WebJars

- 为什么使用

- 如何使用

## 参考:

@lessoon (<https://github.com/lessoon>)

提个小BUG, 在该页面点【spring boot(一): 入门篇】连接会跳到【All articles are here :

[http://www.ityouknow.com/springboot/2016/01/06/springboot\(一\)-入门篇.html](http://www.ityouknow.com/springboot/2016/01/06/springboot(一)-入门篇.html)】页面

([http://www.ityouknow.com/springboot/2016/01/06/springboot\(%E4%B8%80\)-%E5%85%A5%E9%97%A8%E7%AF%87.html%E3%80%91%E9%A1%B5%E9%9D%A2](http://www.ityouknow.com/springboot/2016/01/06/springboot(%E4%B8%80)-%E5%85%A5%E9%97%A8%E7%AF%87.html%E3%80%91%E9%A1%B5%E9%9D%A2))

([http://www.ityouknow.com/springboot/2016/01/06/springboot\(%E4%B8%80\)-%E5%85%A5%E9%97%A8%E7%AF%87.html%E3%80%91%E9%A1%B5%E9%9D%A2](http://www.ityouknow.com/springboot/2016/01/06/springboot(%E4%B8%80)-%E5%85%A5%E9%97%A8%E7%AF%87.html%E3%80%91%E9%A1%B5%E9%9D%A2))

([http://www.ityouknow.com/springboot/2016/01/06/springboot\(%E4%B8%80\)-%E5%85%A5%E9%97%A8%E7%AF%87.html%E3%80%91%E9%A1%B5%E9%9D%A2](http://www.ityouknow.com/springboot/2016/01/06/springboot(%E4%B8%80)-%E5%85%A5%E9%97%A8%E7%AF%87.html%E3%80%91%E9%A1%B5%E9%9D%A2))

感谢反馈，这个问题解决了。



Home (/)

Spring Boot (/spring-boot.html)

web开发

Spring 社区 (/spring-in-cn.html)

Spring 文档 (/spring-doc.html)

自定义Filter

Link (/link.html)

自定义Property

数据库操作

thymeleaf模板

Gradle 构建工具

WebJars

参考:

whatalittlebear (https://github.com/whatalittlebear) 发表于 大约 1 个月前

shandongcjjia (https://github.com/shandongcjjia) 发表于 大约 1 个月前

@whatalittlebear (https://github.com/whatalittlebear)

@xiaolu991954992 (https://github.com/xiaolu991954992)

您好, 请问一下配置信息通过@value (https://github.com/value)自动注入的时候, 中文信息会产生乱码, 怎么解决呀?

我也是这个问题, 希望大侠们指点下, 我的配置文件编码改成utf-8了, 但是断点跑的时候, 看到代码处获得的值的确乱码了, 好纠结。。我的是eclipse

我也是这个问题, 希望大侠们指点下, 我的配置文件编码改成utf-8了, 但是断点跑的时候, 看到代码处获得的值的确乱码了, 好纠结。。我的是eclipse

@whatalittlebear (<https://github.com/whatalittlebear>)

@xiaolu991954992 (<https://github.com/xiaolu991954992>)

Home (/) 您好, 请问Spring-Boot不配置信息通过@value(<https://github.com/value>)自

web开发

动注入的时候, 中文信息会产生乱码, 怎么解决呀?

Spring-Cloud (/spring-cloud.html) Archives (/archives.html)

- 自定义Filter

我也是这个问题, 希望大侠们指点下, 我的配置文件编码改成utf-8了, 但

Link (/link.html) 是断点跑的时候, 看到代码处获得的值的确乱码了, 好纠结。。我的是

- 配置在application.properties中

- log配置

加这个配置

数据库操作

spring.http.encoding.charset=UTF-8

- 1、添加jar包

- 2、添加配置文件

- 3、添加实体类和Dao

- 4、测试

Jedeiah (<https://github.com/Jedeiah>) 发表于 大约 1 个月前

thymeleaf模板

- Thymeleaf 介绍

- URL

- 条件求值

- for循环

- 页面即原型

Gradle 构建工具

WebJars

- 什么是WebJars

- 为什么使用

- 如何使用

参考:

自定义工具类 1 重写MySQL5Dialect的getTableTypeString方法, 修改创建数据库时

候的编码 (ta默认库的编码是iso8859-1) 2 插入数据时候, 需指定编码, mysql5.6需

在my.ini文件中指定。(在idea的spring boot框架的配置文件引入不管用-待思考)

我给你赋上工具类截图。发送自 Windows 10 版邮件应用 发件人: whatalittlebear 发

送时间: 2018年8月8日 11:24 收件人: ityouknow/blog-comments 抄送: Jedeiah;

Comment 主题: Re: [ityouknow/blog-comments] springboot(二): web综合开发 -

纯洁的微笑博客 (#153 ([https://github.com/ityouknow/blog-](https://github.com/ityouknow/blog-comments/issues/153)

[comments/issues/153](https://github.com/ityouknow/blog-comments/issues/153))) @xiaolu991954992

(<https://github.com/xiaolu991954992>) 您好, 请问一下配置信息通过@value

(<https://github.com/value>)自动注入的时候, 中文信息会产生乱码, 怎么解决呀? 我

也是这个问题, 希望大侠们指点下, 我的配置文件编码改成utf-8了, 但是断点跑的时候,

看到代码处获得的值的确乱码了, 好纠结。。我的是eclipse — You are receiving

this because you commented. Reply to this email directly, view it on GitHub, or

mute the thread.



Etnen (<https://github.com/Etnen>) 发表于 23 天前



挺不错的, 那个你每个类展示的时候也把上面导入的包也展示一下。有时候导入时不知

道是哪个Spring Boot (Spring-boot.html)

web开发 import org.springframework.web.bind.annotation.RequestMapping;  
Spring Cloud (spring-cloud.html) Archives (archives.html)

- 自定义Filter

Link (/link.html) About (/about.html)

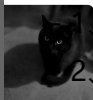
自定义Property daguang08 (<https://github.com/daguang08>) 发表于 19 天前



- 配置在application.properties中

- log配置

数据库操作



添加jar包  
bihell (<https://github.com/bihell>) 发表于 7 天前



2、添加配置文件

3、添加实体类和Dao  
综合开发的代码运行报错。。。需要帮助

- 4、测试

thymeleaf模板

with name 'enableRedisKeyspaceNotificationsInitializer' defined in class path  
- Thymeleaf 介绍  
resource

- URL

[org.springframework.session.data.redis/config/annotation/web/http/RedisHttpSessionConfiguration.class]: Invocation of init method failed; nested exception

- for循环

is org.springframework.data.redis.RedisConnectionFailureException:

- 页面即原型

java.net.SocketException: Broken pipe (Write failed); nested exception is

Gradle 构建工具

redis.clients.jedis.exceptions.JedisConnectionException:

WebJars

java.net.SocketException: Broken pipe (Write failed)

- 什么是WebJars

- 为什么使用

org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory  
y.initializeBean(AbstractAutowireCapableBeanFactory.java:1628) ~[spring-beans-

参考:

4.3.10.RELEASE.jar:4.3.10.RELEASE]

at

org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactor

y.doCreateBean(AbstractAutowireCapableBeanFactory.java:555) ~[spring-beans-4.3.10.RELEASE.jar:4.3.10.RELEASE]

at

Home (/) Spring Boot (/spring-boot.html) AbstractAutowireCapableBeanFactory

web开发 y.createBean(AbstractAutowireCapableBeanFactory.java:483) ~[spring-beans-

Spring-Cloud (/spring-cloud.html) Archives (/archives.html)

at

Link (/link.html) About (/about.html) AbstractBeanFactory\$1.getObject(AbstractBeanFactory.java:306) ~[spring-beans-4.3.10.RELEASE.jar:4.3.10.RELEASE]

at

- 自定义配置

org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:230) ~[spring-beans-

4.3.10.RELEASE.jar:4.3.1

- 1、添加依赖包

- 2、添加配置文件

- 3、添加实体类和Dao

- 4、测试



thymeleaf模板

bajie2 (https://github.com/bajie2) 发表于 大约 23 小时前



- Thymeleaf介绍

- URL 夹? 还要进行哪些配置, controller才能使用它?

- 条件求值

- for循环

- 页面即原型

## Post Directory

**Gradle 构建工具**

**WebJars**

- 什么是WebJars

- 为什么使用

- 如何使用 知乎 (https://www.zhihu.com/people/ityouknow) 微博 (http://weibo.com/ityouknow)

**参考:**

Github (https://github.com/ityouknow)

资源 (/share/2017/10/01/resource-sharing.html) 故事 (/life.html) 架构 (/arch.html) Jvm (/jvm.html) FastDFS (/fastdfs.html)

MongoDB (/mongodb.html) Docker (/docker.html) Code (/open-source.html)

Home (/) Spring-Boot (/spring-boot.html)

## web开发

Spring-Cloud (/spring-cloud.html) Archives (/archives.html)

- 自定义Filter

Link (/link.html) About (/about.html)

- 配置在application.properties中

- log配置

## 数据库操作

- 1、添加jar包

- 2、添加配置文件

- 3、添加实体类和Dao

- 4、测试

## thymeleaf模板

- Thymeleaf 介绍

- URL

- 条件求值

- for循环

- 页面即原型

## Gradle 构建工具

## WebJars

- 什么是WebJars

- 为什么使用

- 如何使用

## 参考: