ERP实现记录

2014年6月3日

目录

[ERP实现记录 1](#_Toc391499347)

[1 问题记录 3](#_Toc391499348)

[1.1 js中字符串换行连接 3](#_Toc391499349)

[1.2 js中如何将JSON字符串转换为对象 3](#_Toc391499350)

[1.2.1 使用eval函数 3](#_Toc391499351)

[1.2.2 使用JSON.parse和JSON.stringfy函数 4](#_Toc391499352)

# 问题记录

## js中字符串换行连接

JScript中的续行符是  "\"     反斜

var json = '[ \

{"id":1,"pid":-1,"name":"parent title 1", "url":"#"}, \

{"id":11,"pid":1,"name":"child title 11", "url":"#"}, \

{"id":12,"pid":1,"name":"child title 12", "url":"#"} \

]';

## js中如何将JSON字符串转换为对象

### 使用eval函数

注意，必须在json字符串两侧加上括号。因为为了让最外层的{}不要被eval理解成为Javascript的代码块的开始.而是理解成为一个对象字面量。

两种写法：

var json = '[ \

{"id":1,"pid":-1,"name":"parent title 1", "url":"#"}, \

{"id":11,"pid":1,"name":"child title 11", "url":"#"}, \

{"id":12,"pid":1,"name":"child title 12", "url":"#"} \

]';

var data = eval("("+json+")");

var jsonStr = '{"a":1}';

eval('var jsonObj = '+jsonStr);

alert(jsonObj.a);

### 使用JSON.parse和JSON.stringfy函数

eval本质是执行脚本，容易被注入攻击（用户构造恶意脚本数据），服务端一定不能使用。

当然，可以使用parse代替，前提是浏览器支持。

* JSON.parse(data)：将json字符串转换为js对象
* JSON.stringfy(data)：将js对象转换为json字符串

## springmvc与前台交互json数据

### 使用@ResponseBody和@RequestBody

配置文件spring-servlet中增加<mvc: annotation-driven />

注意，增加了上述配置后，DefaultAnnotationHandlerMapping和AnnotationMethodHandlerAdapter两项配置可以不用写了。

### 使用ObjectMapper

推荐使用第一种，比较简单。

## jdbc无法注入

web工程初始化，没有初始化jdbc数据源。

从日志看pre init 中已经初始化了对应的dao，该dao继承了JdbcDaoSupport，但是获取不到jdbc数据源。

原因：

拷贝其它项目的配置，在applicationContext.xml的开头配置了default-lazy-init="true"，

导致

spring在启动的时候，会默认加载会默认加载整个对象实例图，从初始化ACTION配置、到 service配置到dao配置、乃至到数据库连接、事务等等。这样可以减少web服务器在运行时的负担，但是对于开发者来说无疑是效率极低的一个设置了。

    还好，spring提供了default-lazy-init属性，其配置形式如下，applicationContext.xml中：

< beans  default-lazy-init ="true" >     
      < bean  class ="org.xxxx.bean" >

      。。。。。。

</beans>

    spring配置默认default-lazy-init为false，当配置为true时sping不会再去加载整个对象实例图，大大减少了初始化的时间，减少了spring的启动速度。

    这样做只是为了在开发过程中节约启动时间，在部署到实际环境中，倒是没必要设置default-lazy-init为true。毕竟部署到实际环境中不是经常的事，每次启动1分钟倒不是大问题，而且可以提高服务器效率。

删掉这行配置即可。

## junit总是报NullException

网上的配置

@RunWith(SpringJUnit4ClassRunner.**class**)

@ContextConfiguration(locations = {"classpath:spring-mvc.xml", "classpath:applicationContext.xml"})

少了一行配置@WebAppConfiguration

@RunWith(SpringJUnit4ClassRunner.**class**)

@WebAppConfiguration

@ContextConfiguration(locations = {"classpath:spring-mvc.xml", "classpath:applicationContext.xml"})