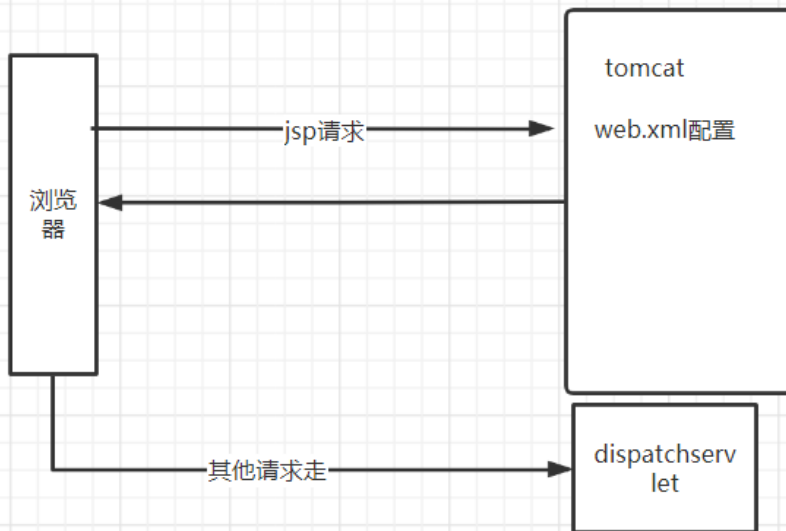


blog_product_经验

1. 静态资源场景分析
2. 路径加/的作用
3. servletContext的对象
4. 网页代码开发小技巧
 - 4.1 关于jsp的和遍历的
 - 4.2 idea 配置前端支持语法:
 - 4.3 注意js在传递参数的注意点:
 - 4.4 jsp通过url传递参数可以直接这么获取 `${param.uid}`
5. spring注解失效的应用
6. 解决延迟加载导致的序列化异常
7. jquery小知识
8. 业务开发小经验多表联查==>转换单表
9. 表单提交的几种小技巧
10. 页面跳转技巧与页面行元素移除
 - 10.1 页面跳转
 - 10.2 行元素删除(删除元素组)
11. ajax的华丽胡稍请求 (get也可这么干)
12. 做删除类别业务的注意点

1. 静态资源场景分析

- 不加静态资源处理器配置

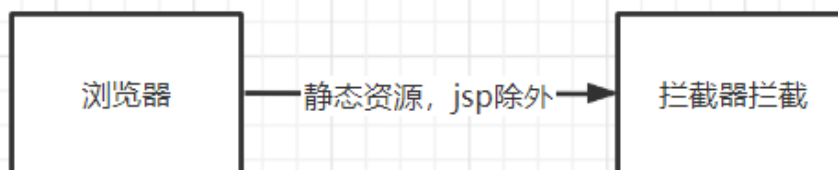


- 加上静态资源处理器配置



- 加上静态资源并加上拦截器

除了jsp请求都会被拦截器拦截



2. 路径加/的作用

- 浏览器解析：就是http://localhost:8080; 服务器解析：就是http://localhost:8080/项目名称
- 前端浏览器发送请求的所有，重定向----->浏览器解析
- 转发，获取上下文路径，url-patterns，----->服务器解析, 代表项目路径;


```

<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
<head>
<title>Title</title>
<base href="http://aaa/bbb/">
<script src="js/index.js"></script>
</head>
<body>
<form action="user/login" method="post">

<base href="${pageContext.request.scheme}://"
<script src="js/index.js"></script>
</head>
<body>

    I

    ${pageContext.request.scheme} <br>
    ${pageContext.request.serverName} <br>
    ${pageContext.request.serverPort} <br>
    ${pageContext.request.contextPath} <br>
    <form action="user/login" method="post">

```

http://localhost:8080/day02_springmvc01_war_exploded/

http

localhost

8080

/day02_springmvc01_war_exploded

username:

password:

可以通过它们动态拼接全路径请求。

4. 网页代码开发小技巧

4.1 关于jsp的和遍历的

//result.list:获取当前页的数据进行遍历

动态字符串中使用这个`${}会导致被认为从`

$\{ \}$ 里面可以是

json, 也可是字符串, 可以是方法参数, 也可以是动态字符串

4.2 idea 配置前端支持语法:



4.3 注意js在传递参数的注意点:

1 只有字符串才会加

4.4 jsp通过url传递参数可以直接这么获取 \${param.uid}

```

</style>
<script>
    $(function () {
        let uid=${param.uid};
        alert(uid);
    })
</script>

```

1 jsp获取id还可以这么直接获取url传递的参数

5.spring注解失效的应用

```

@Transactional
@Override
public void deleteBlogByBid(Integer bid) {
    //1.删除评论信息
    EvaluateExample evaluateExample = new EvaluateExample();
    evaluateExample.createCriteria().andBFkEqualTo(bid);
    evaluateMapper.deleteByExample(evaluateExample);
    //2.删除博客信息
    blogMapper.deleteByPrimaryKey(bid);
}

```

1 不能写try catch将异常吞掉不然会导致注解失效

- 情况二:调用了当前类的方法，不是代理对象调用
- 情况三：没有注解驱动

6.解决延迟加载导致的序列化异常

```

.com.fasterxml.jackson.databind.exc.InvalidDefinitionException: No serializer found for class org.apache.ibatis.executor.loader.javassist.JavassistProxyFactory$EnhancedResultObjectProxyImpl and no properties discovered to create BeanSerializer (to avoid exception, disable SerializationFeature.FAIL_ON_EMPTY_BEANS) (through reference chain: com.github.pagehelper.PageInfo["list"]->com.github.pagehelper.Page[0]->com.edu.bean.Blog_$_jvstd82_0["handler"]) with root cause
at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:81)
at org.apache.catalina.valves.AbstractAccessLogValve.invoke(AbstractAccessLogValve.java:690)
at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:87)
at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:343)
at org.apache.coyote.http11.Http11Processor.service(Http11Processor.java:615)

```

延迟加载的时候为我们的查询blog对象添加了属性handler,导致序列化异常

```

@JsonIgnoreProperties(value = "handler")
public class Blog implements Serializable {
    private Integer bid;

    private String btitle;

    private Integer typeFk;

```

1 忽略被动添加的 handler 属性

7. jquery小知识

- 如何绑定渲染后的函数，页面\$(function(){ 想要绑定函数，绑定我们渲染的数据标签}): 直接使用\$.click()不起作用

```

<script>
    $(function () {
        getBlogList(1);
        $(document).on("click", $(".delBtn"), function () {
            alert("===")
        })
    })

```

这么绑定可以绑

定到

- jquery循环小知识

\$.each(arr,function(value,index){ })

有的人也这么写: \$(arr).each(function(index,value){ })

- javase小知识

```

public String test01(){
    try {
        int x=9/0;
        return "success";
    } catch (Exception e) {
        e.printStackTrace();
    }
    return "error";
}

```

① 这个会执行

8.业务开发小经验多表联查==>转换单表

```

@Override
public Blog getBlogByBid(Integer bid) {
    Blog blog = blogMapper.selectByPrimaryKey(bid);
    User user = userMapper.selectByPrimaryKey(blog.getuFk());
    blog.setUser(user);
    return blog;
}

```

- 直接这么分步走，整合封装即可

9.表单提交的几种小技巧

表单提交：

1. 普通提交
2. jquery提交\$("#id").submit()
3. jquery序列化提交
4. onclick="return tijiao()";


```
// 博客的添加
$("#subBtn").click(function () {
    var formData=$("#formId").serialize();
    $.ajax({
        type:"post",
        url:"blog/addBlog",
        data:formData,
        dataType: "json",
    })
})
})
```

1 表单序列化

```
});
$("#subBtn").click(function(){
    $("#subBtn").submit();
    window.location.href="blog_list.jsp";
});
```

屏幕影

10. 页面跳转技巧与页面行元素移除

10.1 页面跳转

```
se if(result.status==-2){
    Dreamer.error(result.message,1500);
    // 没有登录就跳转登录页面
    window.setTimeout(function () {
        window.open("login.jsp","_top")
        window.location.href=""
    },1500)
```

```
});
$("#subBtn").click(function(){
    $("#subBtn").submit();
    window.location.href="blog_list.jsp";
});
```

屏幕影

10.2 行元素删除(删除元素组)

```

<td>\${ev.econtent}</td>
<td>\${ev.etime}</td>
<td><button type="button" eid="\${ev.eid}"
        onclick="deleteEvaluate(\${ev.eid})">删除</button></td>
</tr>

```

```

function (result) {
    if(result.status==1){
        Dreamer.success(result.message);
        // 删除成功后，删除页面中的当前行评论数据
        $(`button[eid=\${eid}]`).parents("tr").remove();
    }else{
        Dreamer.error(result.message);
    }
}

```

11. ajax的华丽胡稍请求 (get也可这么干)

```

// 评论删除的方法
function deleteEvaluate(eid){
    if(confirm("您确定删除该评论吗")){
        $.post("evaluate/deleteEvaluate",`eid=\${eid}`,
            function (result) {
                if(result.status==1){
                    // 删除成功
                }
            }, "json")
    }
}

```

```

<script>
    $(function(){
        // 请求你的类型信息 不传查询一级分类，传查询二级分类
        function selectBtypeList(bid) {
            if(bid==undefined||bid==""){
                // 请求你的类型信息
            }
        }
    })
</script>

```

12. 做删除类别业务的注意点

- 我们做删除业务的时候一般不要级联删除，要提示用户的方式什么可以删，什么不可以删除
- 具体的业务删除也是改字段，并不是真正的删除