

上下文

本章内容：共2小节，6个知识点

- 第1节：ServletContext接口
- 第2节：数据作用域



本章目标



- 理解上下文的含义及作用；
- 掌握ServletContext接口实例的获取方法以及接口中常用方法；
- 理解请求、会话、上下文作用域的区别；

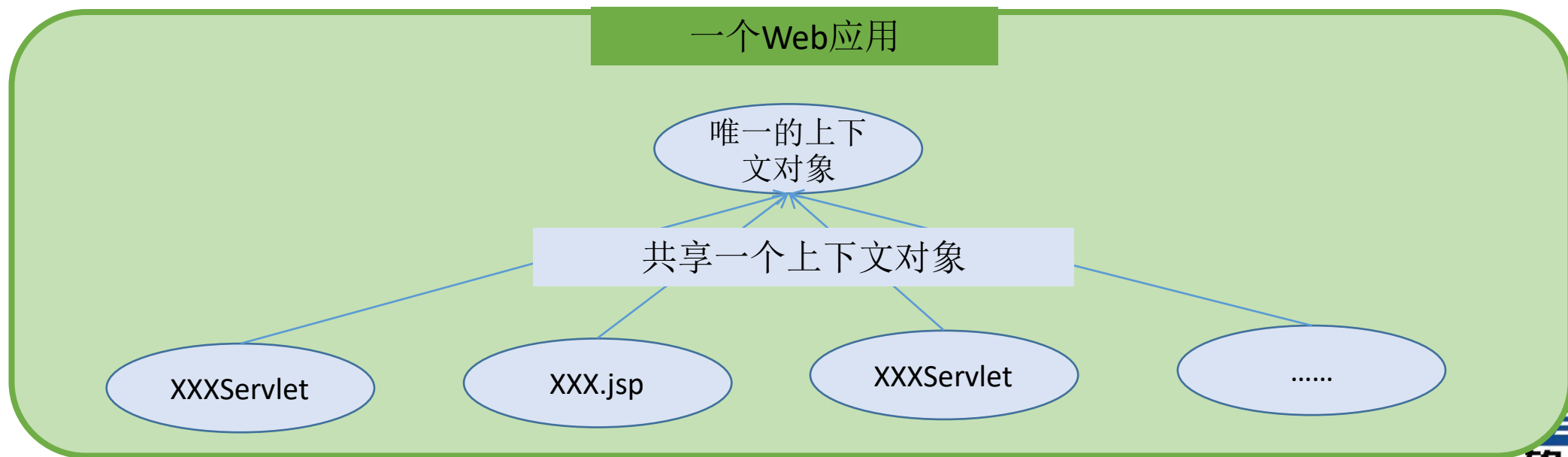
第1节 【ServletContext接口】



- 知识点1：上下文对象的概念、作用
- 知识点2：上下文获取方法
- 知识点3：上下文参数

知识点1: 上下文对象的概念、作用-1

- 上下文对象是用来存储全局范围信息的对象；换句话说，一个Web应用只有唯一的一个上下文对象；
- 当服务器启动的时候，就会为每一个应用创建一个上下文对象；
- 当服务器关闭的时候，上下文对象就销毁；



知识点1：上下文对象的概念、作用-2



- Servlet规范中定义了ServletContext接口，表示上下文对象；
- 该接口中定义了一系列的方法，例如：【与属性有关的方法下节学习】：

方法声明	方法描述
<code>java.io.InputStream getResourceAsStream(java.lang.String path)</code>	将path所代表的资源以输入流返回，可以进一步进行读操作；可以用来读取服务器端的文件；
<code>RequestDispatcher getRequestDispatcher(java.lang.String path)</code>	返回RequestDispatcher 对象，路径是相对于上下文路径的；

知识点2：上下文获取方法-1



- ServletContext对象是服务器自动创建的；
- 要使用该对象，需要获取该对象才行；
- Servlet规范中的多个接口中都定义了getServletContext方法获得上下文对象：

getServletContext occurrences:

```
javax.servlet.FilterConfig.getServletContext()  
javax.servlet.GenericServlet.getServletContext()  
javax.servlet.http.HttpSession.getServletContext()  
javax.servlet.jsp.PageContext.getServletContext()  
javax.servlet.ServletConfig.getServletContext()  
javax.servlet.ServletContextEvent.getServletContext()  
javax.servlet.ServletRequest.getServletContext()  
javax.servlet.ServletRequestEvent.getServletContext()  
javax.servlet.ServletRequestWrapper.getServletContext()
```

HttpServlet类继承了GenericServlet类，所以自定义的Servlet类中都可以直接使用。

知识点2：上下文获取方法-2

课堂案例：

[TestServletContext1.java](#)

[TestServletContext2.java](#)

- 在两个Servlet中使用getServletContext方法返回上下文对象；
- 打印输出两个对象的hashCode，可见值相同；
- 说明在一个应用下，上下文对象是唯一的；

```
//返回ServletContext对象
```

```
ServletContext ctxt=this.getServletContext();
```

```
//在控制台打印输出上下文对象的hashCode
```

```
System.out.println("TestServletContext01 中的上下文对象hashcode:"+ctxt.hashCode());
```

- 输出结果可见：两个Servlet中获得的上下文对象的hashcode相同，是同一个对象；

```
TestServletContext01中的上下文对象hashcode:31067154
```

```
TestServletContext02中的上下文对象hashcode:31067154
```


知识点3：上下文参数-1



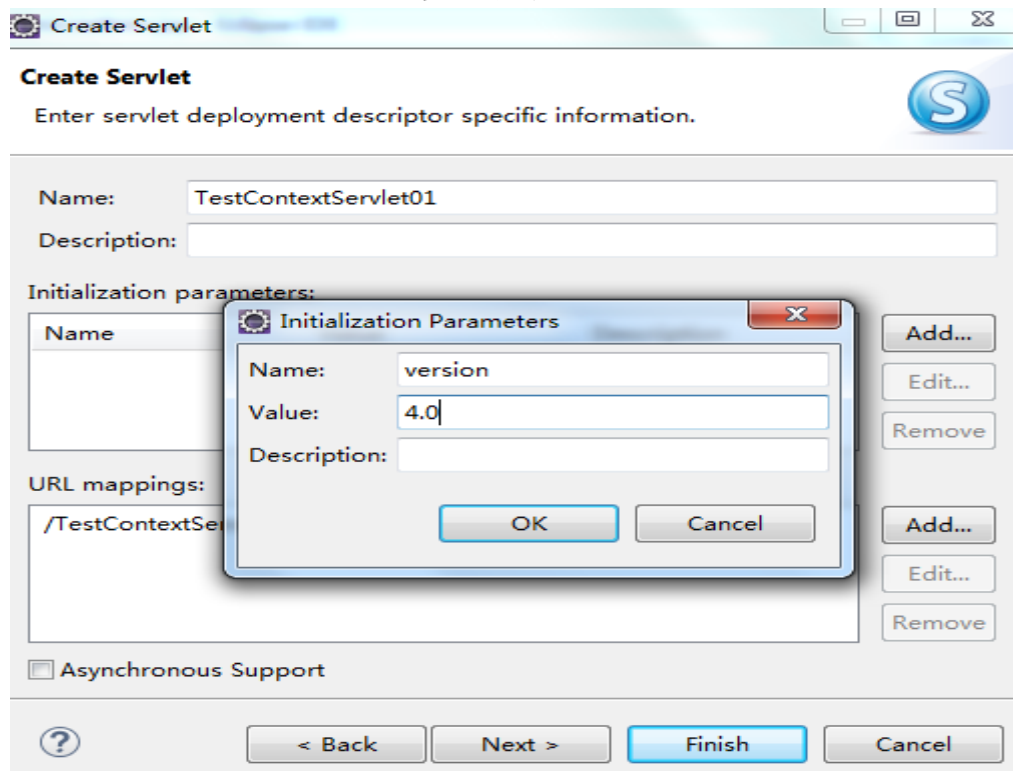
- 在web.xml中可以配置上下文参数，使用ServletContext中的getInitParameter方法可以获取该参数；【之前学习过的Servlet初始化参数，只能在当前Servlet中使用】
- 上下文参数存储在上下文对象，所以应用下所有组件都可以使用；

```
<context-param>
    <param-name>version</param-name>
    <param-value>2.0</param-value>
</context-param>
```

- 获取上下文参数：

```
//          返回ServletContext对象
ServletContext ctxt=this.getServletContext();
//          获取上下文参数
String version=ctxt.getInitParameter("version");
System.out.println("上下文参数version的值: "+version);
```

知识点3：上下文参数-2



注解方式获取参数

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
    // 获取上下文参数  
    ServletConfig config = getServletConfig();  
    String version = config.getInitParameter("version");  
    System.out.println("上下文参数version的值: "+version);  
}
```

本节总结提问【ServletContext接口】



- 上下文对象是如何创建的？有什么作用？
- 上下文对象如何获取？
- 什么是上下文参数？

本节总结 【 ServletContext接口】



- 上下文对象是服务器创建的，一个应用只有一个上下文对象；
- 上下文对象可以用来存储全局共享的数据；
- 很多接口提供了getServletContext方法获得上下文对象；Servlet中可以直接使用该方法；
- 在web.xml中可以定义上下文参数或生成Servlet时用注解的方式创建，使用ServletContext、ServletConfig接口中的getInitParameter方法可以获得该参数进行使用；

第2节：数据作用域



- 知识点1：利用ServletContext在应用中共享数据
- 知识点2：四大作用域范围
- 知识点3：请求、会话、上下文中存放、修改、删除数据方法

知识点1: 利用ServletContext在应用中共享数据

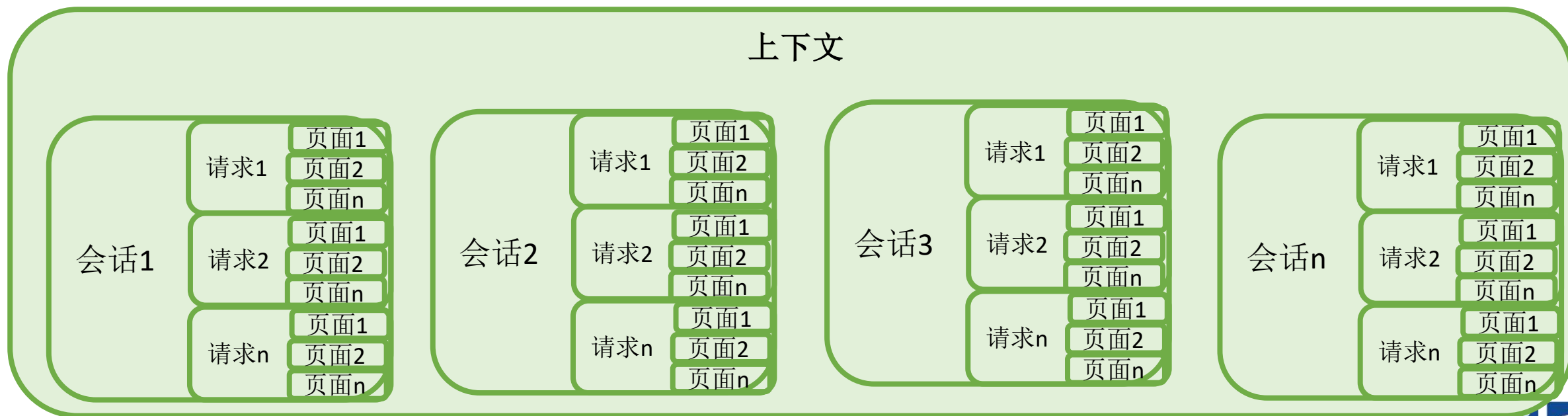


- 之前学习过，请求及会话接口都定义了与属性有关的方法；
- ServletContext接口中也定义了与属性有关的方法，上下文的属性可以在整个应用中共享；

方法声明	方法描述
<code>void setAttribute(java.lang.String name, java.lang.Object o)</code>	将任意类型对象设置为上下文属性，指定一个名字；
<code>java.lang.Object getAttribute(java.lang.String name)</code>	通过属性的名字，获取属性的值；
<code>void removeAttribute(java.lang.String name)</code>	通过属性的名字，删除属性；

知识点2：四大作用域范围

- 在Web应用中，有四大作用域范围
 - 页面范围：一个Servlet或JSP文件；
 - 请求范围：一次请求中可以访问多个Servlet或JSP； 访问的Servlet或JSP能够包含其他资源；
 - 会话范围：一次会话中可以包含多个请求；
 - 上下文范围：上下文包含所有会话；



知识点3：请求、会话、上下文中存放、修改、删除数据方法

- 请求接口、会话接口、上下文接口中都定义相同的方法，可以对数据进行存放、修改、删除：

方法声明	方法描述
<code>void setAttribute(java.lang.String name, java.lang.Object o)</code>	将任意类型对象设置为属性，指定一个名字；
<code>java.lang.Object getAttribute(java.lang.String name)</code>	通过属性的名字，获取属性的值；
<code>void removeAttribute(java.lang.String name)</code>	通过属性的名字，删除属性；

- 多数服务器中，都是使用Map对象来实现不同范围的属性；

ServletConfig接口的简述：

- public interface ServletConfig 由servlet容器使用的servlet配置对象，用于在servlet初始化时传递信息。
- getServletName()方法概述： public java.lang.String getServletName()
- 该方法返回一个servlet实例的名称，该名称由服务器管理员提供。
- getServletContext()方法概述： public ServletContext getServletContext()
- 返回一个ServletContext对象的引用。
- getInitParameter()方法概述： public java.lang.String
getInitParameter(java.lang.String name)
- 返回一个由参数String name决定的初始化变量的值，如果该变量不存在，返回null。

ServletConfig接口的简述:



- getInitParameterNames()方法概述: `public java.util.Enumeration getInitParameterNames()`
- 返回一个存储所有初始化变量的枚举函数。如果servlet没有初始化变量, 返回一个空枚举函数。
- 最后, 该ServletConfig接口由GenericServlet类实现。
- ServletConfig与ServletContext的区别:
- ServletConfig是针对**只有一个**特定的servlet的局部变量,
- ServletContext是整个web application的**全局变量**ServletContext存放在ServletConfig中

本节总结提问【数据作用域】

- 有几个作用域？
- 不同作用域共享数据的方法是什么？



本节总结 【数据作用域】

- Web应用中有4个作用域，分别是页面范围、请求范围、会话范围、上下文范围；这四个作用域的范围从小到大排列；
- 页面范围指的是一个Servlet或JSP；请求范围可以访问多个Servlet/JSP；会话范围可以包含多个请求；上下文范围包含所有会话；
- 请求、会话、上下文接口都定义了与属性有关的方法，用来保存、获取、删除共享数据；

本章总结



- 上下文是一个全局对象，由服务器创建，一个应用只有唯一的一个；
- 可以用上下文对象存储全局共享数据；
- ServletContext接口中定义了相关方法，可以用来获取初始化参数、输入流、请求转发器等，也可以用来操作上下文范围的属性；
- Web应用中，有页面、请求、会话、上下文四大作用域；

本章作业

- 作业1：用XML方式配置上下文参数，并获得其值
- 作业2：用注解方式配置上下文参数，并获得其值
- 难度：中



