# 永和大王门店管理系统

**内参保密不得外传**

**达内Java培优方向**

**陈子枢 主讲**

**QQ: 1920-9520-78**

**http://dobetter.tedu.cn**

**2017**

# 第一天：业务需求+开发环境

## 软件开发

拿到一个项目如何开发呢？从哪里入手呢？具体的技术点如何实现呢？这就是我们这次要让大家体会的，不为你理解每个技术细节，这些后期课程中都会详细讲到。只为你有大的格局，你有清晰的目标，你有真实的体验。有所了解后踏实学习java培优的课程，它能支持你在这个日益竞争激烈的世界里，安身立命，实现梦想，过上好日子！

### 开发岗位

* 初级程序员
* 中级程序员
* 高级程序员
* 系统分析师
* 架构师
* 技术总监
* 技术副总CTO

### 怎样是一个合格的程序员？

Do more，Do better，Do share！

* 逻辑思维能力，把需求能描述清楚
* 找重点的能力，从琐碎的细节中找到核心，什么是重要的，什么是可以暂时忽略
* 总结能力 3w1h，Why? Where? When? How?
* 解决问题能力，面对未知能快速解决，分担领导的工作
* 汇报能力，给经理、高层、客户等汇报工作，word、ppt，用数据说话
* 团队精神，乐于分享

### 软件开发流程

也称为软件开发生命周期：

* 意向
* 软件项目立项，立项报告，立项审批
* 需求调研，需求评审
* 概要设计，用例图，页面原型设计
* 数据库设计，表设计
* 详细设计，类设计
* 开发编码
* 需求变更
* 测试
* 试运行，bug满天飞（加班）
* 正式上线
* 维护期
* 项目完成

## 学到的技术点

### Day01

* Jdk的使用
* Eclipse的使用
* 创建java工程
* 创建class类

### Day02

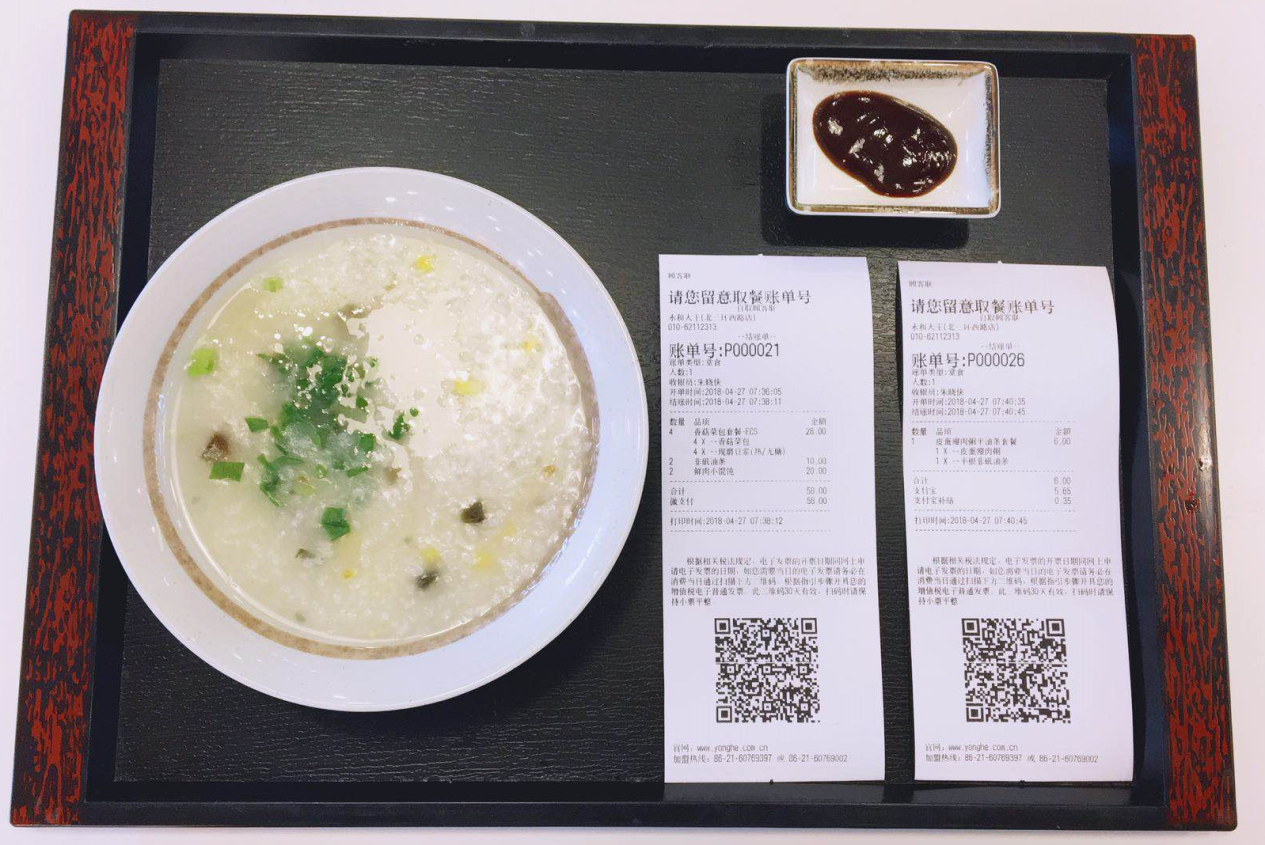
* 创建Web工程
* HTML静态网页
* CSS样式表

## 永和大王门店管理系统

### 项目背景



《永和大王门店管理系统》实现顾客进店后选择购买的套餐，系统生成订单，包括：是哪个门店的，产生订单流水号，并记录下订单类型及人数和收银员是谁，及下单时间和结账时间，并记录下订单详情，顾客点了哪些食物，及支付总金额。系统打印订单小票。同时，在订单下方展示本店的二维码，方便顾客扫描自助填写税务发票信息。



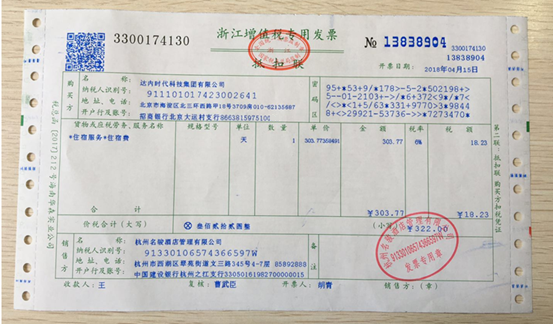
### 还能实现什么？

票据打印

#### 医药管理系统



#### 税务发票系统



#### 餐厅结账单



#### 德克士



#### 加油站

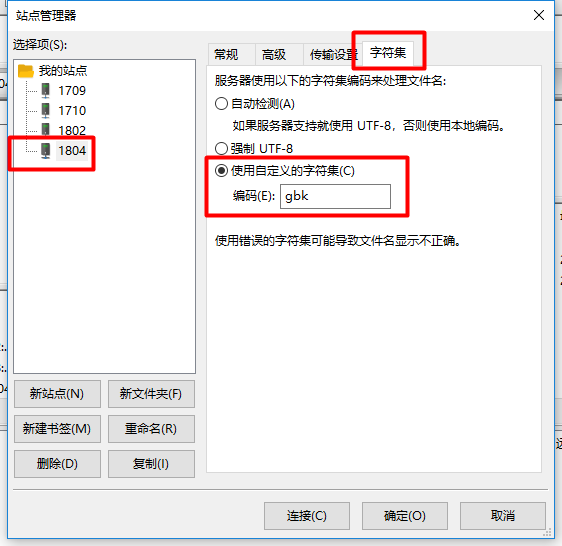


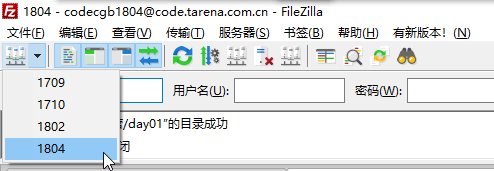
## 开发工具

### 必备的开发软件

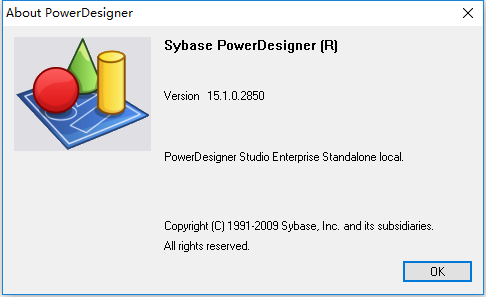
* PowerDesinger v15.1 数据库表设计&UML设计
* mysql-5.5.27-winx64.msi mysql数据库
* SQLyogEnt v8.2 数据库客户端
* 安装双jdk1.7 & jdk1.8 java环境
* Eclipse Mars.2 Release (4.5.2) IDE 开发工具，配置jdk、tomcat
* 安装Tomcat Web中间件，运行web程序

### FTP工具FileZilla下载资料

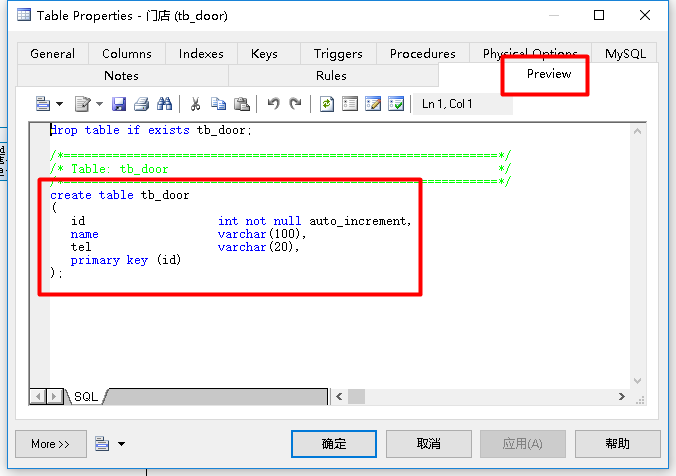




### PowerDesinger设计工具安装

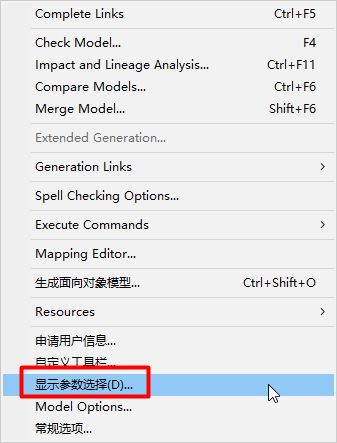


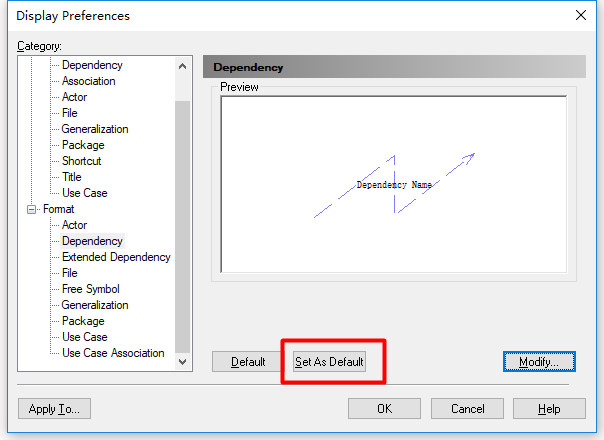
语言选择中文RPC，同意协议，其他一路next即可。

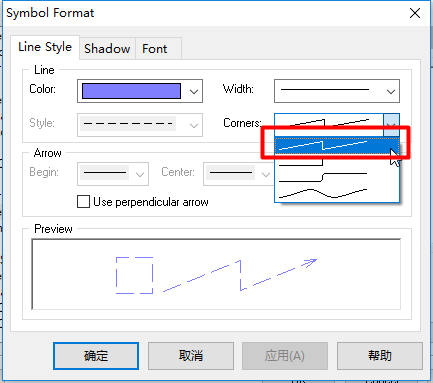


### PD配置直线

新版本默认拐角线，老版使用直线。先创建对应的模型，然后改其所有线类型。







## Java开发环境安装

### Dos命令基本操作

cmd 进入DOS窗口 #快捷方式，浏览目录，直接输入CMD跳转

cd java 进入java目录

cd .. 退到上层目录

cd / 退到根目录

dir 列出当前目录下的目录和文件

### 两个版本的jdk

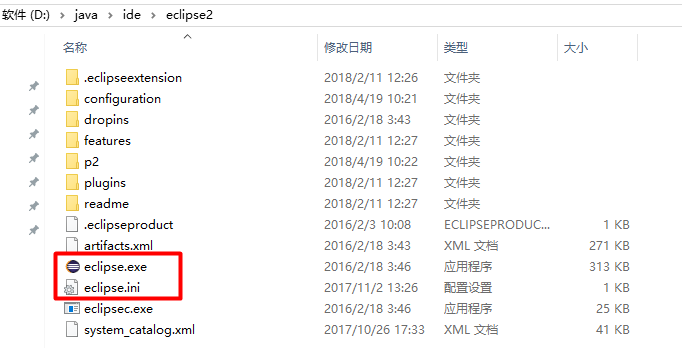
安装双jdk1.7 & jdk1.8

jdk-8u121-windows-x64.exe 安装解决dll注册问题

jdk1.7.0\_72.zip 绿色解压版

### 安装Eclipse开发工具

D:\java\ide\eclipse2



eclipse.ini

-startup

plugins/org.eclipse.equinox.launcher\_1.3.100.v20150511-1540.jar

--launcher.library

plugins/org.eclipse.equinox.launcher.win32.win32.x86\_64\_1.1.300.v20150602-1417

-product

org.eclipse.epp.package.jee.product

--launcher.defaultAction

openFile

--launcher.XXMaxPermSize

256M

-showsplash

org.eclipse.platform

--launcher.XXMaxPermSize

256m

--launcher.defaultAction

openFile

--launcher.appendVmargs

-vm

D:\java\env\jdk1.7.0\_72\bin\javaw.exe

-vmargs

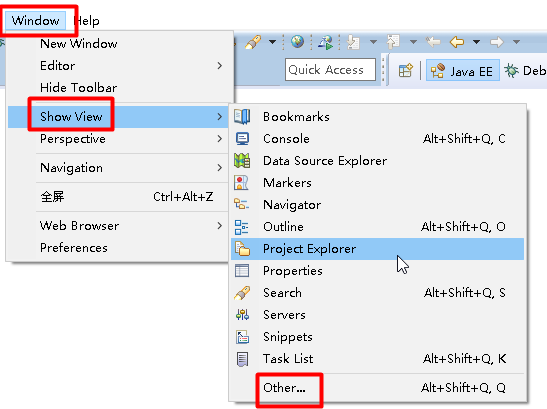
-Dosgi.requiredJavaVersion=1.7

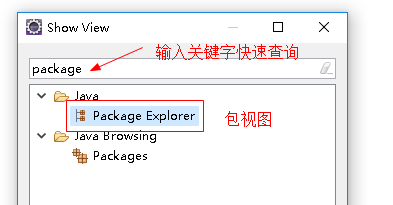
-Xms256m

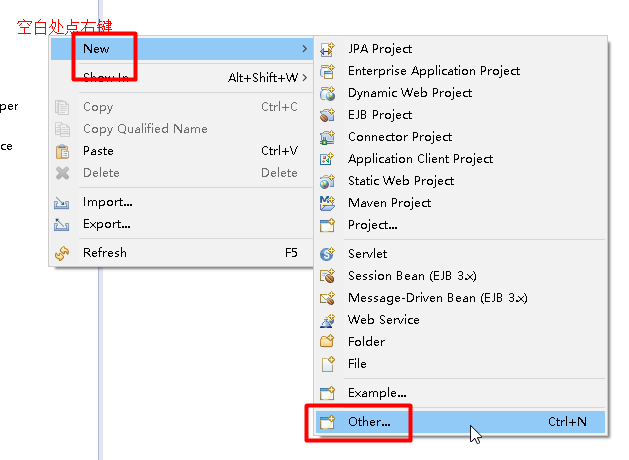
-Xmx1024m

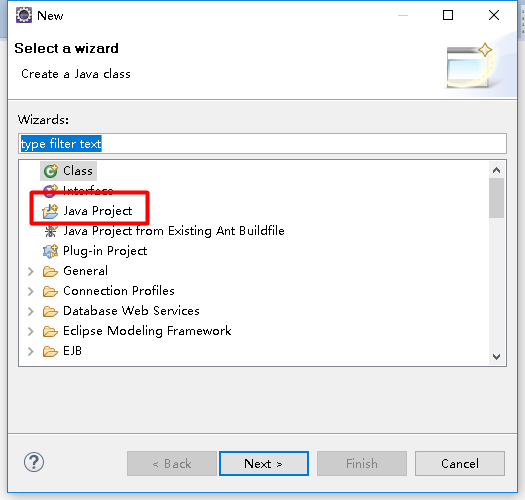
## 第一个java程序HelloWorld

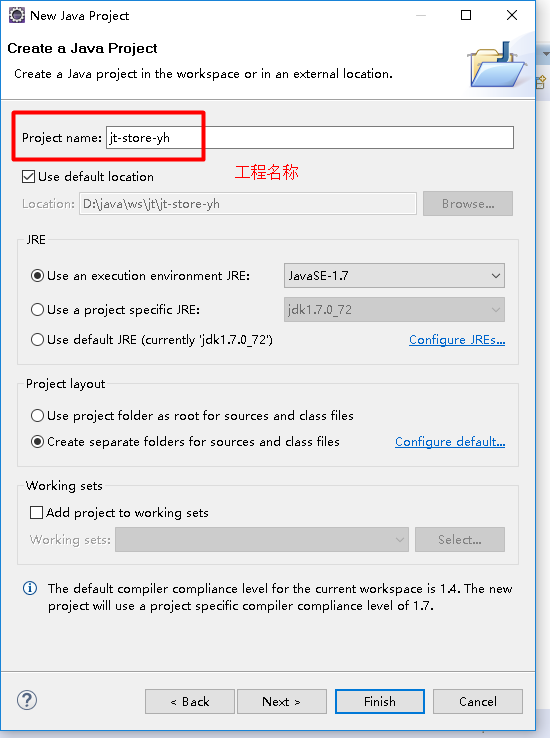
### 创建java工程



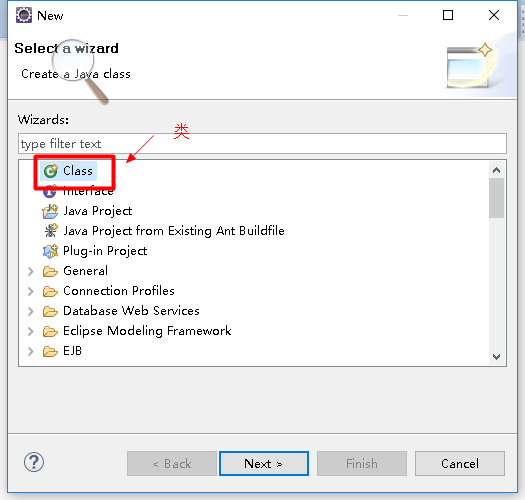


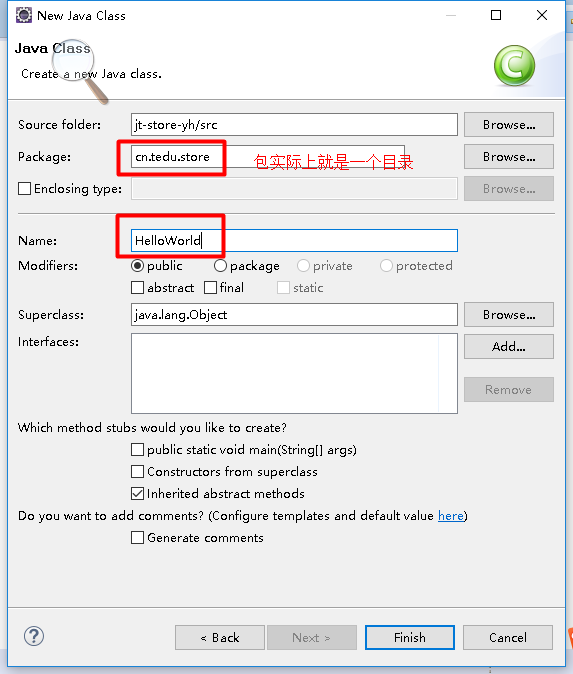




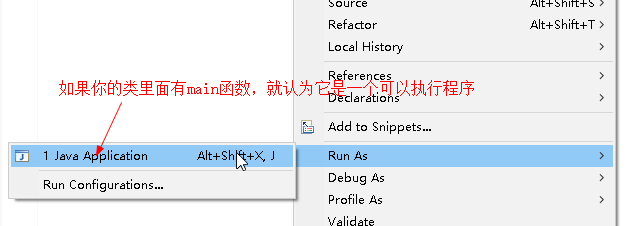


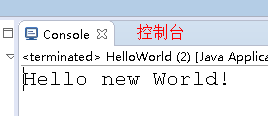
### 创建第一个类

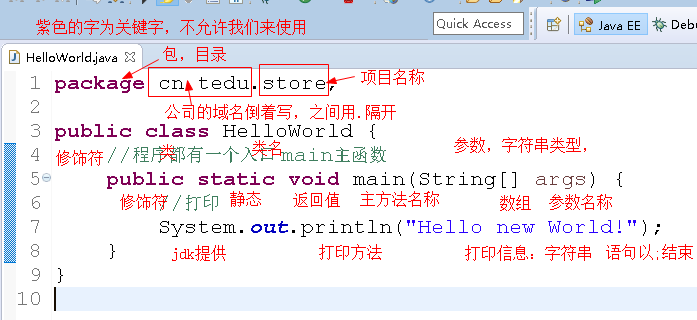


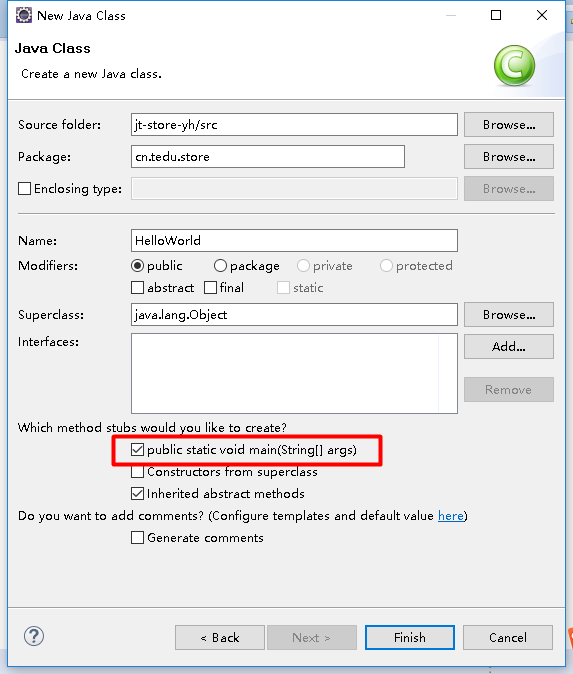












### 开发步骤

1. 创建java工程
2. 创建java类，设置包路径，类名称
3. 创建main主函数，才能运行
4. System.out.println(“Hello World”)；

### HelloWorld.java

package cn.tedu.yh.print; //包路径，注释

//类，public修饰符，class类，唯一代表，在包路径范围内

public class HelloWorld {

//程序运行入口

/\*

\* public 修饰符

\* static 静态

\* void 没有返回值

\* main 程序入口，写死的

\* String[] args 参数

\* String[] 字符串数组

\* args 参数名称

\*

\* 运行，右键，执行Run As，run application

\*/

public static void main(String[] args){

//打印Hello World字符串

System.out.println("Hello World！");

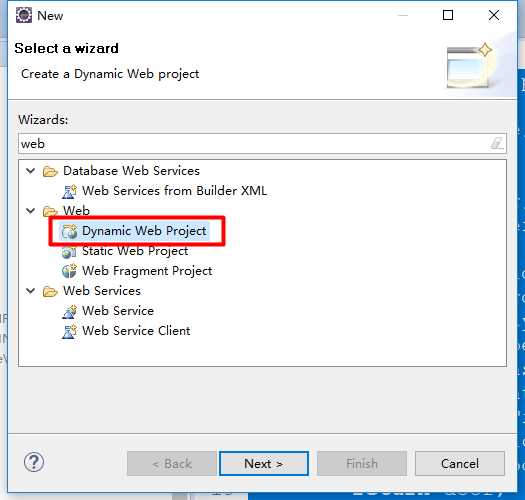
}

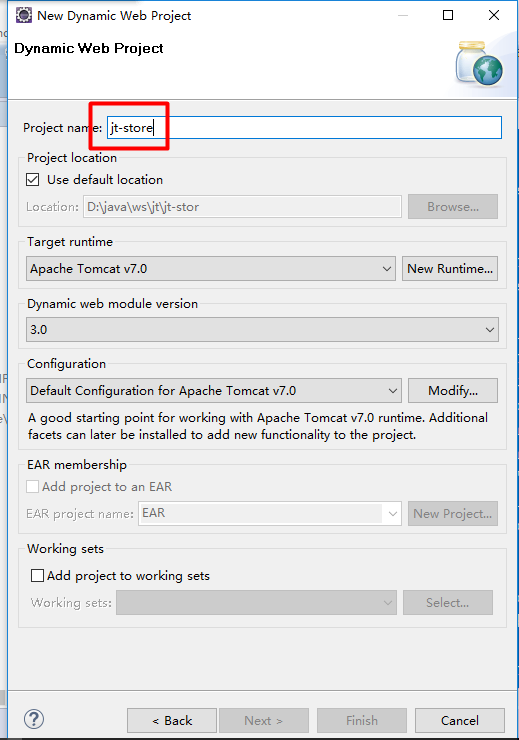
}

# 第二天：静态页面HTML+样式表+二维码

## 创建WebProject

### 创建web工程

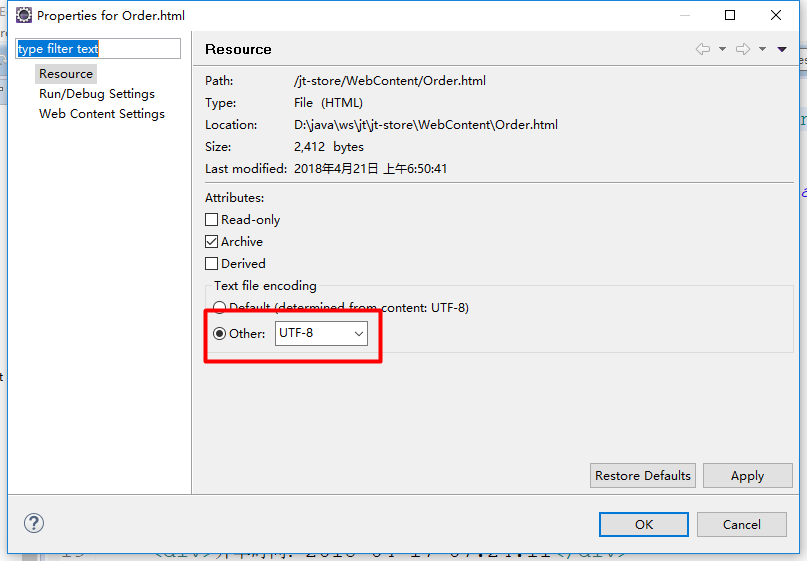




## 页面原型

### 第一个HTML静态页面

创建后修改文件的字符集，文件字符集



WebContent/Order.html

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<link rel="stylesheet" href="css/base.css">

<title>永和大王账单</title>

</head>

<body>

<div>顾客联</div>

<div class="please">请您留意取餐账单号</div>

<div class="me">自取顾客联</div>

<div class="doorName">永和大王（北三环西路店）</div>

<div class="doorName">010-62112313</div>

<div class="account">--结账单--</div>

<div class="please">账单号：P000009</div>

<div>账单类型：食堂</div>

<div>人数：1</div>

<div>收银员：张静</div>

<div>开单时间：2018-04-17 07:24:11</div>

<div>结账时间：2018-04-17 07:24:22</div>

<hr style="border : 1px dashed ;" />

<div>

<table>

<tr><td width="40">数量</td><td width="130">品项</td><td>金额</td></tr>

<tr>

<td>1</td><td>油条豆浆套餐</td><td>7.00</td>

</tr>

<tr>

<td>&nbsp;</td><td>1 X --非矾油条</td><td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td><td>1 X --现磨豆浆（热/甜）</td><td>&nbsp;</td>

</tr>

</table>

</div>

<hr style="border : 1px dashed ;" />

<div>

<table>

<tr>

<td width="140">支付宝花呗一元早餐</td><td width="30">1</td><td width="30" align="right">-3.00</td>

</tr>

<tr>

<td>合计</td><td>&nbsp;</td><td width="30" align="right">4.00</td>

</tr>

<tr>

<td>支付宝</td><td>&nbsp;</td><td width="30" align="right">1.00</td>

</tr>

<tr>

<td>支付宝补贴</td><td>&nbsp;</td><td width="30" align="right">3.00</td>

</tr>

</table>

</div>

<hr style="border : 1px dashed ;" />

<div>打印时间：2018-04-17 07:24:23</div>

<hr style="border : 1px dashed ;" />

<div class="note">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;根据相关税法规定，电子发票的开票日期同网上申请电子发票的日期，如您需要当日的电子发票请务必在消费当日通过扫描下方二维码，根据指引步骤开具您的增值税电子普通发票。此二维码30天有效，扫描时请保持小票平整。</div>

<div class="qrimg"><img src="img/qr.png"></div>

<div>官网：www.yonghe.com.cn</div>

<div>加盟热线：86-21-60769397 或 86-21-60799002</div>

<br/>

</body>

</html>

### 引入样式表

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<link rel="stylesheet" href="css/base.css">

<title>永和大王账单</title>

</head>

### 第一个样式表css

WebContent/css/base.css

body{ margin: 5; font-size:8px; width:280px;}

.please{ font-size:22px; }

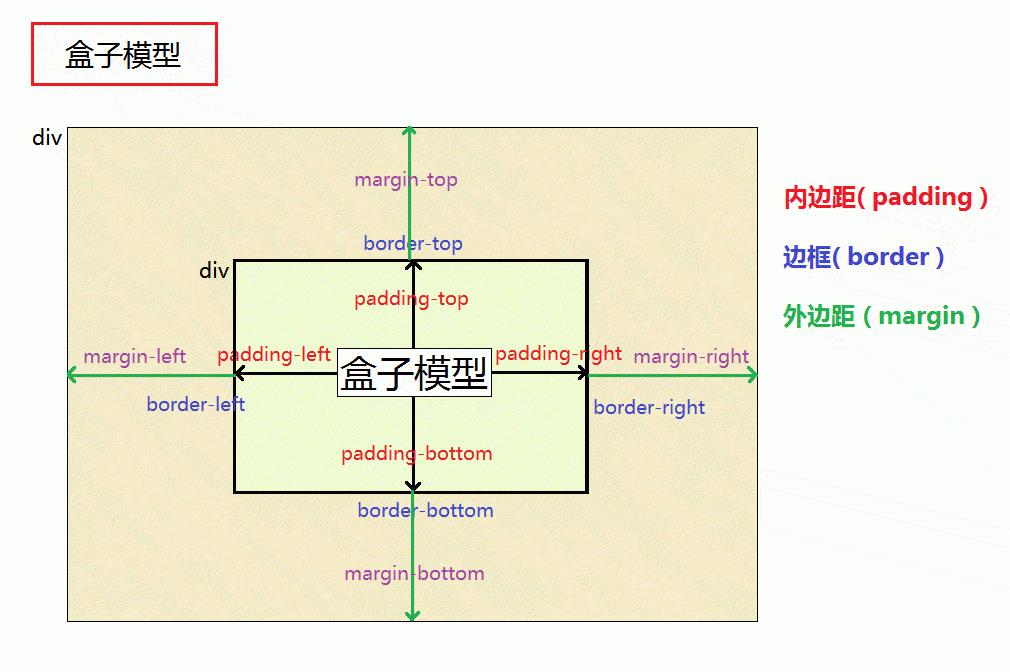
.me{ padding-left:56px; }

.doorName{ font-size:10px; }

.account{ padding-left:60px; }

.note{ padding-top:5px;padding-bottom:5px; }

.qrimg{ padding-left:60px; }



## 创建二维码

导入google提供的jar包：

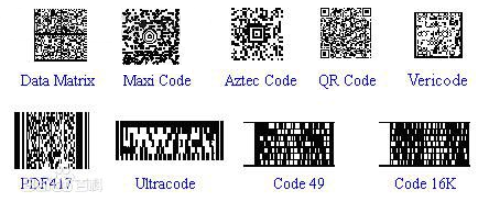
D:\java\ws\jt\jt-store\WebContent\WEB-INF\lib\core-3.3.2.jar

D:\java\ws\jt\jt-store\WebContent\WEB-INF\lib\javase-3.3.2.jar

### 一维码和二维码



一维条码（条形码）的宽度记载着数据，而其长度没有记载数据。



二维条码是指在一维条码的基础上扩展出另一维具有可读性的条码，使用黑白矩形图案表示二进制数据，被设备扫描后可获取其中所包含的信息。

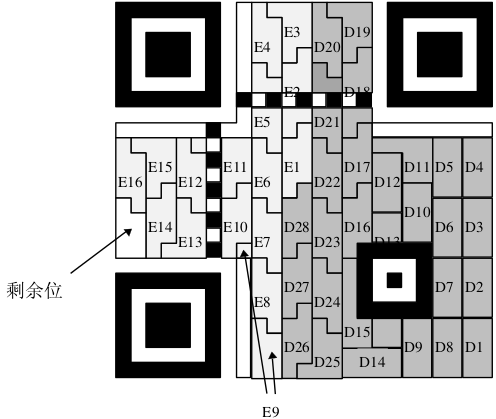
二维条码的长度、宽度均记载着数据。二维条码有一维条码没有的“定位点”和“容错机制”。容错机制在即使没有辨识到全部的条码、或是说条码有污损时，也可以正确地还原条码上的信息。二维条码的种类很多，不同的机构开发出的二维条码具有不同的结构以及编写、读取方法。

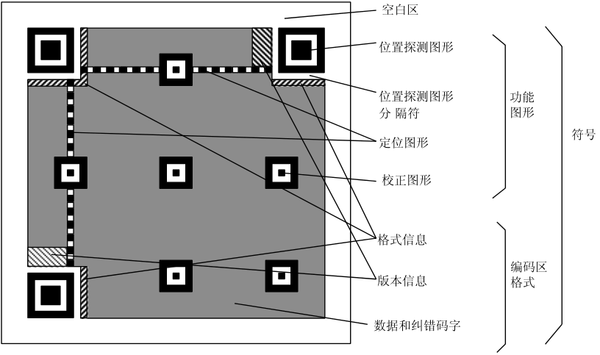
二维码虽然比起一维条码具有更强大的信息记载能力，但也是有容量限制，通过下面这个表格小草带你了解二维码的容量到底有多大。



### 什么是二维码

二维码又称二维条码，是1994年由日本DW公司发明，常见的二维码为QR Code，QR全称Quick Response，是一个近几年来移动设备上超流行的一种编码方式。





* 位置探测图形、位置探测图形分隔符、定位图形：用于对二维码的定位，对每个QR码来说，位置都是固定存在的，只是大小规格会有所差异；
* 校正图形：规格确定，校正图形的数量和位置也就确定了；
* 格式信息：表示改二维码的纠错级别，分为L、M、Q、H；
* 版本信息：即二维码的规格，QR码符号共有40种规格的矩阵（一般为黑白色），从21×21（版本1），到177×177（版本40），每一版本符号比前一版本 每边增加4个模块。
* 数据和纠错码字：实际保存的二维码信息，和纠错码字（用于修正二维码损坏带来的错误）。
* 了解了二维码的基本结构后，将探测图形、分隔符、定位图形、校正图形和码字模块放入矩阵中，并把上面的完整序列填充到相应规格的二维码矩阵的区域中。

### CreateQR.java

package qr;

import java.io.File;

import java.nio.file.Path;

import java.util.HashMap;

import org.junit.Test;

import com.google.zxing.BarcodeFormat;

import com.google.zxing.EncodeHintType;

import com.google.zxing.MultiFormatWriter;

import com.google.zxing.client.j2se.MatrixToImageWriter;

import com.google.zxing.common.BitMatrix;

import com.google.zxing.qrcode.decoder.ErrorCorrectionLevel;

/\*

\* 二维码又称二维条码，常见的二维码为QR Code，QR全称Quick Response，

\* 是一个近几年来移动设备上超流行的一种编码方式，

\* 它比传统的Bar Code条形码能存更多的信息，也能表示更多的数据类型。

\*/

public class CreateQR {

@Test

public void createQR(){

int width = 150; //宽

int height = 150; //高

String url = "http://www.tedu.cn"; //内容

String fileName = "D:\\qr.png";

create(width, height, url, fileName);

System.out.println("二维码创建成功！");

}

private void create(Integer width, Integer height, String url, String fileName) {

HashMap<EncodeHintType, Comparable> hm = new HashMap<EncodeHintType, Comparable>();

hm.put(EncodeHintType.CHARACTER\_SET, "UTF-8"); //1设置编码格式

hm.put(EncodeHintType.MARGIN, 2); //2设置四周的边：0-7个等级

hm.put(EncodeHintType.ERROR\_CORRECTION, ErrorCorrectionLevel.M); //纠错等级（容错率）：Q、Ｍ、Ｌ、Ｈ

//产生二维码

try {

BitMatrix bm = new MultiFormatWriter().encode(url, BarcodeFormat.QR\_CODE, width, height, hm);

Path file = new File(fileName).toPath();

MatrixToImageWriter.writeToPath(bm, "png", file);

} catch (Exception e) {

e.printStackTrace();

}

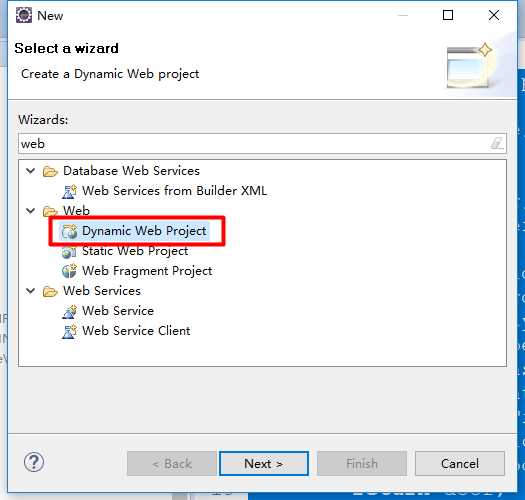
}

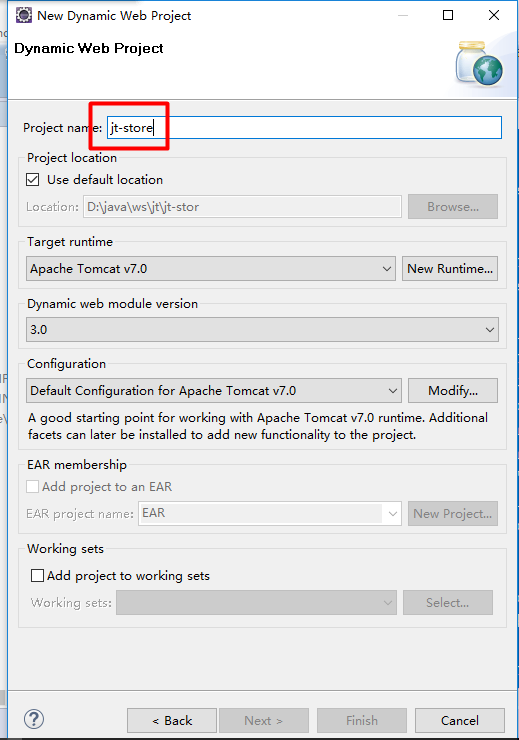
}

# 第三天：动态页面JSP

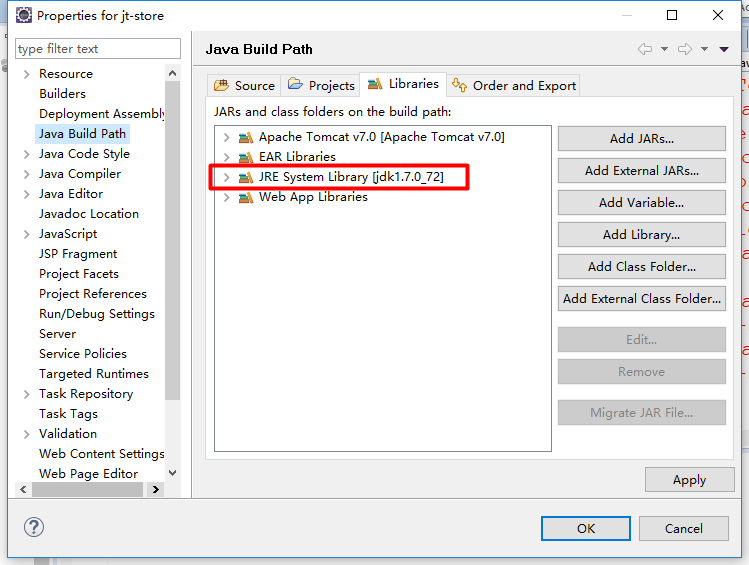
## 创建WebProject

### 创建web工程

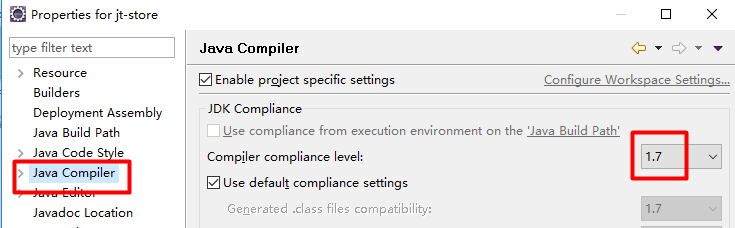




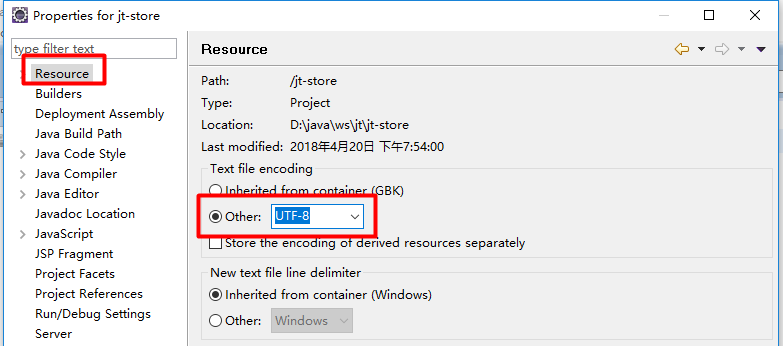
### 配置jdk



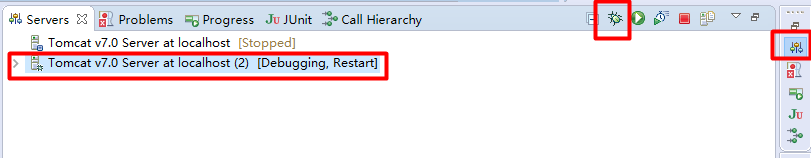
### 配置编译版本



### 导入jar包



### 配置tomcat



## 动态页面

### 第一个Servlet3.0程序

package test;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

@WebServlet("/FirstServlet")

public class FirstServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

public FirstServlet() {

super();

}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

response.getWriter().append("Hello JSP. Served at: ").append(request.getContextPath());

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

doGet(request, response);

}

}

访问地址：http://localhost:8080/jt-store/OrderServlet

### 第一个JSP页面

hello.jsp

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

Hello Jsp;

</body>

</html>

## 属性文件存放数据

### 属性文件

jdbc.properties

jdbc.driver=com.mysql.jdbc.Driver

jdbc.url=jdbc:mysql://127.0.0.1:3306/jtdb?useUnicode=true&characterEncoding=utf8&autoReconnect=true&allowMultiQueries=true

jdbc.username=root

jdbc.password=root

### 读取属性文件

package qr;

import java.io.IOException;

import java.io.InputStream;

import java.util.Properties;

import java.util.ResourceBundle;

import org.junit.Test;

public class PropertiesUtil {

@Test

public void getValue() throws IOException{

String filename = "jdbc.properties";

InputStream is = ClassLoader.getSystemResourceAsStream(filename);

Properties prop = new Properties();

prop.load(is);

System.out.println(prop.getProperty("jdbc.driver"));

}

@Test

public void get(){

ResourceBundle rb = ResourceBundle.getBundle("jdbc");

System.out.println(rb.getString("jdbc.url"));

}

}

### Order.jsp

创建完成后修改字符集ISO8859-1修改为utf-8

<%@ page language="java" contentType="text/html; charset=utf-8"

pageEncoding="utf-8"%>

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<link rel="stylesheet" href="css/base.css">

<title>永和大王账单</title>

</head>

<body>

<div>顾客联</div>

<div class="please">请您留意取餐账单号</div>

<div class="me">自取顾客联</div>

<div class="doorName">${order.door.name}</div>

<div class="doorName">${order.door.tel}</div>

<div class="account">--结账单--</div>

<div class="please">账单号：${order.orderNo}</div>

<div>账单类型：${order.ctype}</div>

<div>人数：${order.personNum}</div>

<div>收银员：${order.cashier}</div>

<div>开单时间：${order.createTime}</div>

<div>结账时间：${order.endTime}</div>

<hr style="border : 1px dashed ;" />

<div>

<table>

<tr><td width="40">数量</td><td width="160">品项</td><td>金额</td></tr>

<c:forEach var="o" items="${order.orderDetails}" >

<tr>

<td>${o.num}&nbsp;</td><td>${o.item}</td><td>${o.price}</td>

</tr>

</c:forEach>

</table>

</div>

<hr style="border : 1px dashed ;" />

<div>打印时间：${order.printTime}</div>

<hr style="border : 1px dashed ;" />

<div class="note">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;根据相关税法规定，电子发票的开票日期同网上申请电子发票的日期，如您需要当日的电子发票请务必在消费当日通过扫描下方二维码，根据指引步骤开具您的增值税电子普通发票。此二维码30天有效，扫描时请保持小票平整。</div>

<div class="qrimg"><img src="img/qr.png"></div>

<div>官网：www.yonghe.com.cn</div>

<div>加盟热线：86-21-60769397 或 86-21-60799002</div>

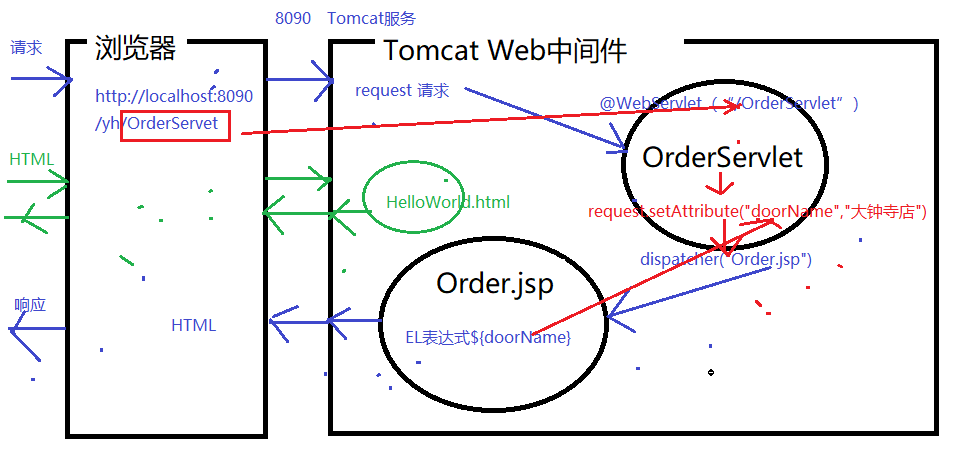
<br/>

</body>

</html>

## 拓展

### 访问流程图



### 实现jsp文件安全访问

讲jsp文件放入到WEB-INF目录中，j2ee规范中，网络用户是无法直接访问WEB-INF目录下的内容的，但服务端的Servlet是可以访问这些资源文件的。

于是将Order.jsp放入到WEB-INF/views目录中，通过Servlet中转就实现了jsp文件的安全访问。

开发步骤：

1. 将Order.jsp放入到WEB-INF/views目录中
2. OrderServlet.java中写法如下

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

//转向jsp,Dispatcher分发，名称就是转向的jsp名称

request

.getRequestDispatcher("/WEB-INF/views/Order.jsp")

.forward(request, response);

}

# 第四天：MySQL数据库+SqlYog客户端

## 数据库基本操作

SQL语句分三类

* DML（Data Manipulation Language）数据操纵语言

如：insert，delete，update，select（插入、删除、修改、检索）简称CRUD操作新增Create、查询Retrieve、修改Update、删除Delete

* DDL（Data Definition Language）数据库定义语言

如：create table之类

* DCL（Data Control Language）数据库控制语言

如：grant、deny、revoke等，只有管理员才有相应的权限

注意：SQL不区分大小写

### 建库

CREATE DATABASE

jt-store-yh

DEFAULT CHARACTER SET utf8;

### 查看有哪些数据库

show databases;

### 删库

DROP DATABASE jt-store-yh;

### 建表

CREATE TABLE tb\_door123

(

id INT NOT NULL AUTO\_INCREMENT,

NAME VARCHAR(100),

tel VARCHAR(20),

PRIMARY KEY (id)

);

### 查看有哪些表

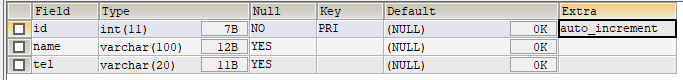
show tables;

### 查看建表语句

show create table tb\_door;

### 查看表结构

desc tb\_door;



### 删表

DROP TABLE tb\_door;

## 数据库常用操作

### 新增

INSERT INTO tb\_door

(id,NAME,tel)

VALUES

(1,'中关村车库咖啡店','010-38382718')

### 查询

SELECT \* FROM tb\_door

### 修改

UPDATE tb\_door

SET tel='010-33282718'

WHERE id=1

### 删除

DELETE FROM tb\_door123 WHERE id=1

### 排序

select \* from tb\_door order by door\_name;

### 记录总数

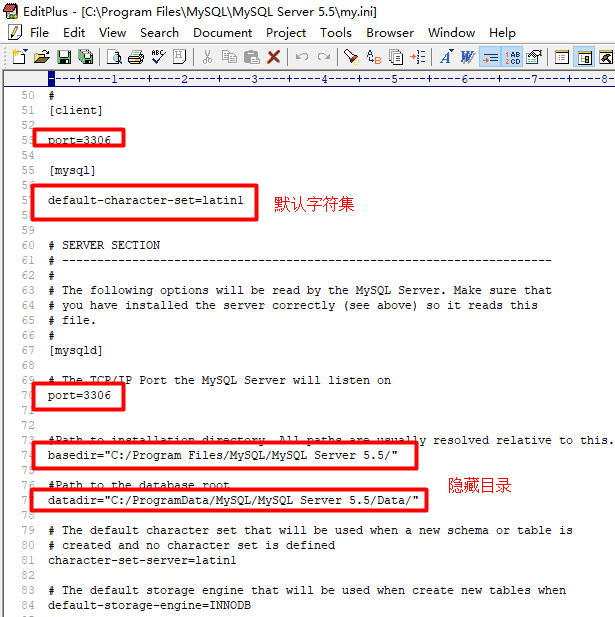
select count(\*) from tb\_door;

select count(id) from tb\_door;

## 数据库工具安装

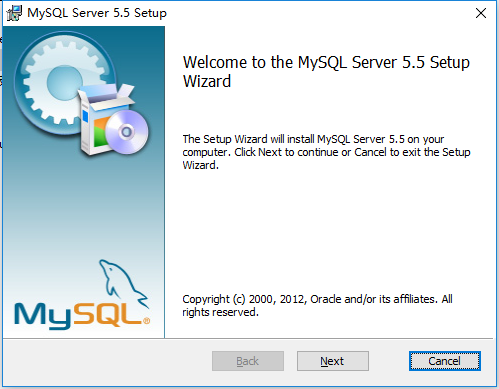
### MySQL数据存放在哪里？

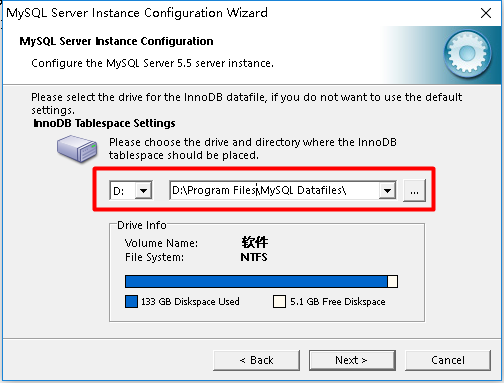
在MySQL的配置文件my.ini中会进行默认配置

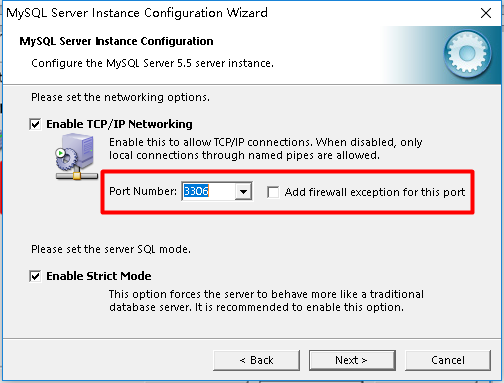


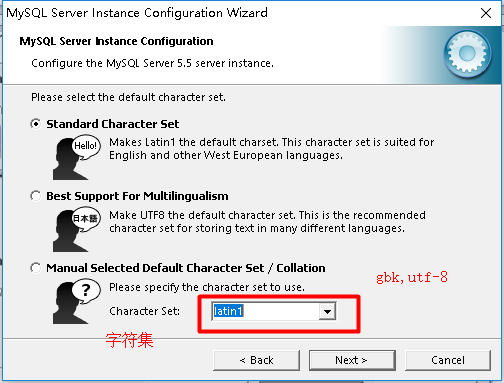
### MySQL服务端

mysql-5.5.27-winx64.msi

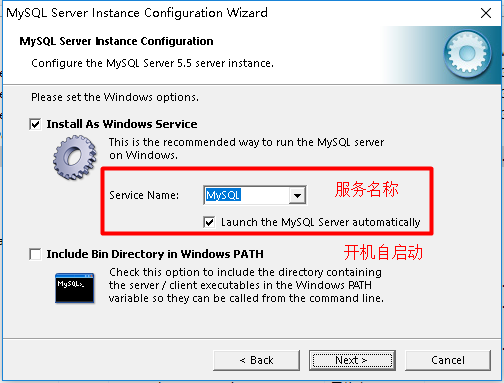


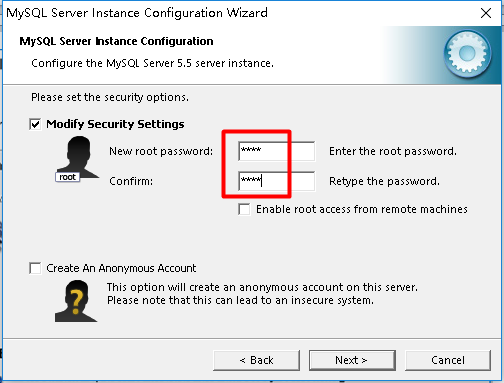






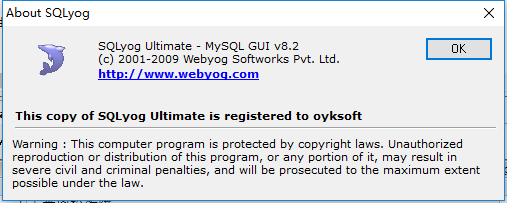
Mysql数据库默认的编码是latin1等价于iso-8859-1



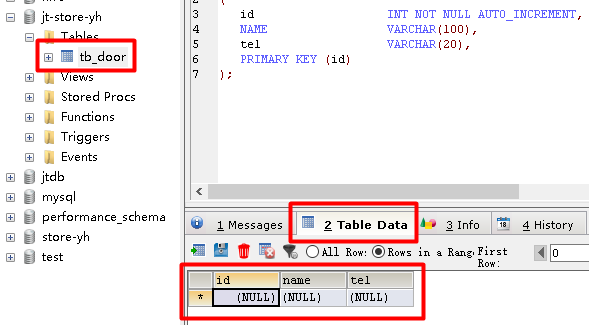


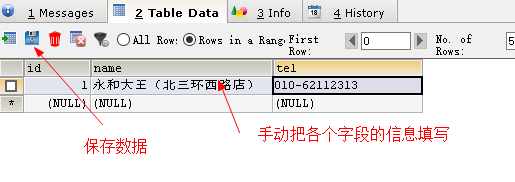
注意：配置完，mysql开始执行，最后一步出错有时仍可以使用，使用SQLyog工具测试，如不行，再执行安装程序，选择remove，删除，然后重新安装。同时注意必须是管理员权限。

### SQLyog数据库访问客户端









执行单条sql语句，如果窗口内有多条SQL语句？只能选择一条来执行。

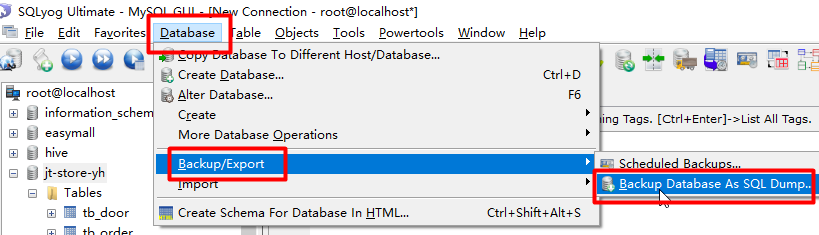
查询语句时，一次只能查询一个；

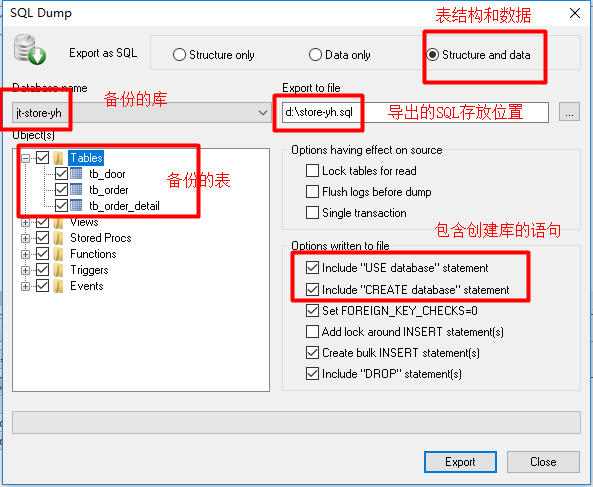
 可以执行多条；删除表，创建表，修改表结构

# 第五天：POJO+对象之间的关系

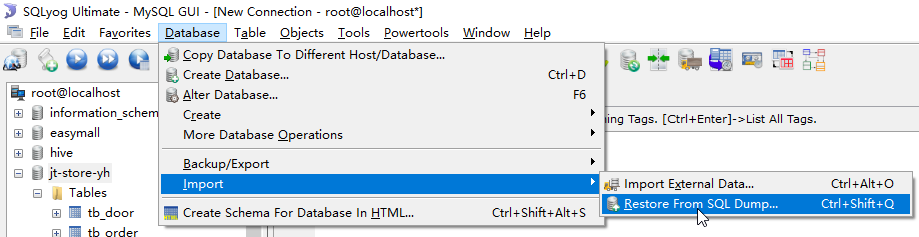
## 数据库增强

### 导出（备份）





### 导入（还原）



### 开启远程访问权限

语法：

grant [权限] on [数据库名].[表名] to ['用户名']@['web服务器的ip地址'] identified by ['密码'];

grant all on \*.\* to 'root'@'%' identified by 'root';

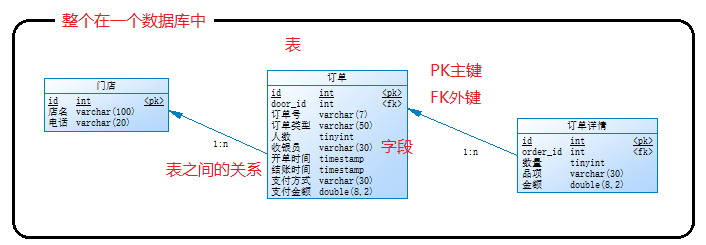
或者指定IP地址

grant all on \*.\* to 'root'@'192.168.1.103' identified by 'root';

## 详细设计

### 数据库概念

* 数据库 Database 不同项目之间的区分
* 表 Table 某个业务存放数据的抽象
* 字段 Column/Field 某个列
* 主键 Primary Key 唯一标识一条记录的字段
* 外键 Foreign Key 引用其它表的主键值



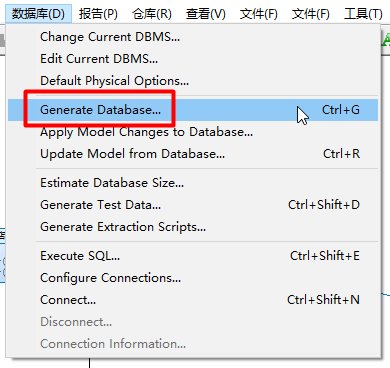
### 表设计

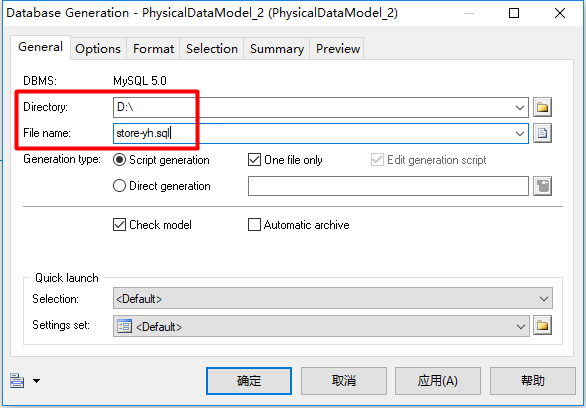
数据库建模 jt-store-yh

****

## 创建测试库

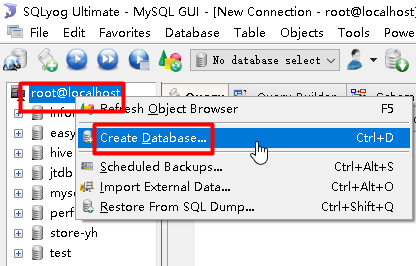
### 生成建表SQL语句

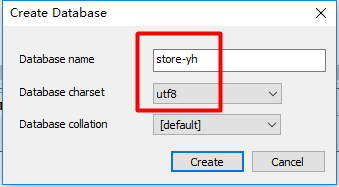




### 创建数据库

创建数据库，注意字符集设置为utf8，防止中文乱码。





### 执行建表脚本

store-yh.sql

/\*==============================================================\*/

/\* DBMS name: MySQL 5.0 \*/

/\* Created on: 2018/4/20 16:42:26 \*/

/\*==============================================================\*/

drop table if exists tb\_door;

drop table if exists tb\_order;

drop table if exists tb\_order\_detail;

/\*==============================================================\*/

/\* Table: tb\_door \*/

/\*==============================================================\*/

create table tb\_door

(

id int not null auto\_increment,

name varchar(100),

tel varchar(20),

primary key (id)

);

/\*==============================================================\*/

/\* Table: tb\_order \*/

/\*==============================================================\*/

create table tb\_order

(

id int not null,

door\_id int,

order\_no varchar(7),

ctype varchar(50),

person\_num tinyint,

cashier varchar(30),

create\_time timestamp,

end\_time timestamp,

micro\_payment double(8,2),

primary key (id)

);

/\*==============================================================\*/

/\* Table: tb\_order\_detail \*/

/\*==============================================================\*/

create table tb\_order\_detail

(

id int not null,

order\_id int,

num tinyint,

item varchar(30),

price double(8,2),

primary key (id)

);

alter table tb\_order add constraint FK\_Reference\_2 foreign key (door\_id)

references tb\_door (id) on delete restrict on update restrict;

alter table tb\_order\_detail add constraint FK\_Reference\_3 foreign key (order\_id)

references tb\_order (id) on delete restrict on update restrict;

注意创建表时，主流方式都不创建外键约束！SQL操作时性能高，方便维护不受约束，程序代码去维护外键关系。

### 第一个JDBC数据库程序

**1）导入jar：**

jt-store\WebContent\WEB-INF\lib\mysql-connector-java-5.1.10-bin.jar

**2）测试类：**

package test;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

public class TestJdbc {

public static void main(String[] args) throws Exception {

String url = "jdbc:mysql://localhost:3306/store-yh";

String user = "root";

String password = "root";

Connection conn = DriverManager.getConnection(url, user, password);

Statement stat = conn.createStatement();

String sql = "select \* from tb\_door";

ResultSet rs = stat.executeQuery(sql);

while (rs.next()){

int id = rs.getInt(1); //rs.getInt("id");

String name = rs.getString(2);

String tel = rs.getString(3);

System.out.println("id="+id+"，name："+name+"，tel："+tel);

}

conn.close();

stat.close();

rs.close();

}

}

## 开发步骤

### POJO

#### Door.java

package com.jt.store.pojo;

public class Door {

private Integer id;

private String name;

private String tel;

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getTel() {

return tel;

}

public void setTel(String tel) {

this.tel = tel;

}

}

#### OrderDetail.java

package com.jt.store.pojo;

public class OrderDetail {

private Integer id;

private Integer num;

private String item;

private Double price;

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public Integer getNum() {

return num;

}

public void setNum(Integer num) {

this.num = num;

}

public String getItem() {

return item;

}

public void setItem(String item) {

this.item = item;

}

public Double getPrice() {

return price;

}

public void setPrice(Double price) {

this.price = price;

}

}

#### 对象之间的关系

public class Order {

private Door door;

private List<OrderDetail> orderDetails;

#### Order.java

package com.jt.store.pojo;

import java.util.Date;

import java.util.List;

public class Order {

private Door door;

private List<OrderDetail> orderDetails;

private Integer id;

private String orderNo;

private String ctype;

private Integer personNum;

private String cashier;

private Date createTime;

private Date endTime;

private Double microPayment;

public Door getDoor() {

return door;

}

public void setDoor(Door door) {

this.door = door;

}

public List<OrderDetail> getOrderDetails() {

return orderDetails;

}

public void setOrderDetails(List<OrderDetail> orderDetails) {

this.orderDetails = orderDetails;

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getOrderNo() {

return orderNo;

}

public void setOrderNo(String orderNo) {

this.orderNo = orderNo;

}

public String getCtype() {

return ctype;

}

public void setCtype(String ctype) {

this.ctype = ctype;

}

public Integer getPersonNum() {

return personNum;

}

public void setPersonNum(Integer personNum) {

this.personNum = personNum;

}

public String getCashier() {

return cashier;

}

public void setCashier(String cashier) {

this.cashier = cashier;

}

public Date getCreateTime() {

return createTime;

}

public void setCreateTime(Date createTime) {

this.createTime = createTime;

}

public Date getEndTime() {

return endTime;

}

public void setEndTime(Date endTime) {

this.endTime = endTime;

}

public Double getMicroPayment() {

return microPayment;

}

public void setMicroPayment(Double microPayment) {

this.microPayment = microPayment;

}

}

## 向jsp传递数据

### EL表达式

在jsp页面中获取参数值

${name} 一般从request对象中获取name值

### JSTL标签库

**1）导入jar包：**

WebContent\WEB-INF\lib\jstl-1.2.jar

**2）jsp页面：引入taglib**

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

#### For循环

<c:forEach var="o" items="${order.orderDetails}">

<tr>

<td valign="top">${o.num}</td>

<td>${o.item}</td>

<td width="30" valign="top">${o.price}</td>

</tr>

</c:forEach>

#### 累计

<c:set var="total" value="0"/>

<c:forEach var="o" items="${order.orderDetails}">

<c:set var="total" value="${total+o.num\*o.price}"/>

<tr>

<td valign="top">${o.num}</td>

<td>${o.item}</td>

<td width="30" valign="top">${o.price}</td>

</tr>

</c:forEach>

<table>

<tr><td width="210">合计</td><td>${total}</td></tr>

<tr><td width="30">支付（${order.paymentType}）</td><td align="right">${total}</td></tr>

</table>

### 转向jsp页面

//转向jsp页面

String url = "Order.jsp";

request.getRequestDispatcher(url).forward(request, response);

### 单值

Servlet中

String name = "tony";

//为转向下一个请求中准备数据

request.setAttribute("name", name);

Jsp中

<div>${name}</div>

### 单对象

Servlet中

Door door = new Door();

door.setName("永和豆浆（北三环店）");

door.setTel("010-372349281");

request.setAttribute("door", door);

Jsp中

<div>${ door }</div>

<div>${ door.name }</div>

<div>${ door.tel }</div>

### 关联对象

Servlet中

Order order = new Order();

Door door = new Door();

door.setName("永和豆浆（北三环店）");

door.setTel("010-372349281");

//绑定数据

order.setDoor(door);

request.setAttribute("order", order);

Jsp中

<div>${ order.door.name }</div>

<div>${ order.door.tel }</div>

### 集合对象

//从数据库获取数据

String cnurl ="jdbc:mysql://localhost:3306/jt-store-yh";

String user = "root";

String password = "root";

//传递集合

List<Door> doorList = new ArrayList<Door>();

try {

new Driver();

Connection conn = DriverManager.getConnection(cnurl,user,password);

Statement stat = conn.createStatement();

ResultSet rs = stat.executeQuery("select \* from tb\_door");

while(rs.next()){

Door door = new Door();

door.setId(rs.getInt("id"));

door.setName(rs.getString("name"));

door.setTel(rs.getString("tel"));

//绑定，把数据放入

doorList.add(door);

}

request.setAttribute("doorList", doorList);

//把门店信息传入，对象变量，传值更多

Door door = new Door();

door.setName("北三环店");

door.setTel("119");

request.setAttribute("door", door);

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

# 第六天：永和大王门店系统开发

## 开发步骤

### POJO

#### Door.java

package com.jt.store.pojo;

public class Door {

private Integer id;

private String name;

private String tel;

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getTel() {

return tel;

}

public void setTel(String tel) {

this.tel = tel;

}

}

#### OrderDetail.java

package com.jt.store.pojo;

public class OrderDetail {

private Integer id;

private Integer num;

private String item;

private Double price;

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public Integer getNum() {

return num;

}

public void setNum(Integer num) {

this.num = num;

}

public String getItem() {

return item;

}

public void setItem(String item) {

this.item = item;

}

public Double getPrice() {

return price;

}

public void setPrice(Double price) {

this.price = price;

}

}

#### 对象之间的关系

public class Order {

private Door door;

private List<OrderDetail> orderDetails;

#### Order.java

package com.jt.store.pojo;

import java.util.Date;

import java.util.List;

public class Order {

private Door door;

private List<OrderDetail> orderDetails;

private Integer id;

private String orderNo;

private String ctype;

private Integer personNum;

private String cashier;

private Date createTime;

private Date endTime;

private Double microPayment;

public Door getDoor() {

return door;

}

public void setDoor(Door door) {

this.door = door;

}

public List<OrderDetail> getOrderDetails() {

return orderDetails;

}

public void setOrderDetails(List<OrderDetail> orderDetails) {

this.orderDetails = orderDetails;

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getOrderNo() {

return orderNo;

}

public void setOrderNo(String orderNo) {

this.orderNo = orderNo;

}

public String getCtype() {

return ctype;

}

public void setCtype(String ctype) {

this.ctype = ctype;

}

public Integer getPersonNum() {

return personNum;

}

public void setPersonNum(Integer personNum) {

this.personNum = personNum;

}

public String getCashier() {

return cashier;

}

public void setCashier(String cashier) {

this.cashier = cashier;

}

public Date getCreateTime() {

return createTime;

}

public void setCreateTime(Date createTime) {

this.createTime = createTime;

}

public Date getEndTime() {

return endTime;

}

public void setEndTime(Date endTime) {

this.endTime = endTime;

}

public Double getMicroPayment() {

return microPayment;

}

public void setMicroPayment(Double microPayment) {

this.microPayment = microPayment;

}

}

### Servlet3.0

#### OrderServlet.java

要解决的问题：

* 怎么把数据传递给jsp？
* 怎么构建Door对象？
* 怎么构建Order对象？
* 怎么构建OrderDetail对象？

package com.jt.store.servlet;

import java.io.IOException;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.List;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.jt.store.pojo.Door;

import com.jt.store.pojo.Order;

import com.jt.store.pojo.OrderDetail;

import com.mysql.jdbc.Driver;

@WebServlet("/OrderServlet")

public class OrderServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

public OrderServlet() {

super();

}

//获取门店信息

public Door getDoorById(Integer id) throws SQLException{

String url = "jdbc:mysql://localhost:3306/store-yh";

String user = "root";

String password = "root";

new Driver();

Connection conn = DriverManager.getConnection(url, user, password);

Statement stat = conn.createStatement();

String sql = "select \* from tb\_door where id = "+id;

ResultSet rs = stat.executeQuery(sql);

Door door = new Door();

while (rs.next()){

door.setId(rs.getInt("id"));

door.setName(rs.getString("name"));

door.setTel(rs.getString("tel"));

}

return door;

}

//获取订单信息

public Order getOrderById(Integer id) throws SQLException{

String url = "jdbc:mysql://localhost:3306/store-yh";

String user = "root";

String password = "root";

new Driver();

Connection conn = DriverManager.getConnection(url, user, password);

Statement stat = conn.createStatement();

String sql = "select \* from tb\_order where id = "+id;

ResultSet rs = stat.executeQuery(sql);

Order order = new Order();

Door door = new Door();

while (rs.next()){

door.setId(rs.getInt("door\_id"));

order.setId(rs.getInt("id"));

order.setOrderNo(rs.getString("order\_no"));

order.setCtype(rs.getString("ctype"));

order.setPersonNum(rs.getInt("person\_num"));

order.setCashier(rs.getString("cashier"));

order.setCreateTime(rs.getDate("create\_time"));

order.setEndTime(rs.getDate("end\_time"));

}

order.setDoor(this.getDoorById(door.getId()));

order.setOrderDetails(this.getOrderOrderDetail(id));

return order;

}

//获取订单详情信息

public List<OrderDetail> getOrderOrderDetail(Integer orderId) throws SQLException{

String url = "jdbc:mysql://localhost:3306/store-yh";

String user = "root";

String password = "root";

new Driver();

Connection conn = DriverManager.getConnection(url, user, password);

Statement stat = conn.createStatement();

String sql = "select \* from tb\_order\_detail where order\_id = "+orderId;

ResultSet rs = stat.executeQuery(sql);

List<OrderDetail> odList = new ArrayList<OrderDetail>();

while (rs.next()){

OrderDetail od = new OrderDetail();

od.setId(rs.getInt("id"));

od.setNum(rs.getInt("num"));

od.setItem(rs.getString("item"));

od.setPrice(rs.getDouble("price"));

odList.add(od);

}

return odList;

}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

try {

Order order = this.getOrderById(1);

request.setAttribute("order", order);

//response.sendRedirect("/jt-store/Order.jsp");

request.getRequestDispatcher("/Order.jsp").forward(request, response);

} catch (SQLException e) {

e.printStackTrace();

}

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

doGet(request, response);

}

}

#### 导入jstl标签支持

D:\java\ws\jt\jt-store\WebContent\WEB-INF\lib\jstl-1.2.jar

Jsp文件中增加

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

循环迭代

<c:forEach var="o" items="${order.orderDetails}" >

<tr>

<td>${o.num}&nbsp;</td><td>${o.item}</td><td>${o.price}</td>

</tr>

</c:forEach>

#### 增加打印日期

private String printTime;

public String getPrintTime() {

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

return sdf.format(new Date());

}

### Order.jsp

<%@ page language="java" contentType="text/html; charset=utf-8"

pageEncoding="utf-8"%>

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<link rel="stylesheet" href="css/base.css">

<title>永和大王账单</title>

</head>

<body>

<div>顾客联</div>

<div class="please">请您留意取餐账单号</div>

<div class="me">自取顾客联</div>

<div class="doorName">${order.door.name}</div>

<div class="doorName">${order.door.tel}</div>

<div class="account">--结账单--</div>

<div class="please">账单号：${order.orderNo}</div>

<div>账单类型：${order.ctype}</div>

<div>人数：${order.personNum}</div>

<div>收银员：${order.cashier}</div>

<div>开单时间：${order.createTime}</div>

<div>结账时间：${order.endTime}</div>

<hr style="border : 1px dashed ;" />

<div>

<table>

<tr><td width="40">数量</td><td width="160">品项</td><td>金额</td></tr>

<c:forEach var="o" items="${order.orderDetails}" >

<tr>

<td>${o.num}&nbsp;</td><td>${o.item}</td><td>${o.price}</td>

</tr>

</c:forEach>

</table>

</div>

<hr style="border : 1px dashed ;" />

<div>打印时间：${order.printTime}</div>

<hr style="border : 1px dashed ;" />

<div class="note">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;根据相关税法规定，电子发票的开票日期同网上申请电子发票的日期，如您需要当日的电子发票请务必在消费当日通过扫描下方二维码，根据指引步骤开具您的增值税电子普通发票。此二维码30天有效，扫描时请保持小票平整。</div>

<div class="qrimg"><img src="img/qr.png"></div>

<div>官网：www.yonghe.com.cn</div>

<div>加盟热线：86-21-60769397 或 86-21-60799002</div>

<br/>

</body>

</html>