# 第5章 CHAPTER 5 **彻联网应用层**

# 5.1 物联网服务器

# 5.1.1 基于 Java 的 Web 服务器搭建

本书中采用 Java Web 技术,开发物联网服务器端软件。为实训项目提供门户展示、信息管理功能,同时为硬件端和移动端提供接口。基于 Java 的 Web 服务器搭建需要几个步骤,下面依次介绍。

#### 1. JDK 的安装

JDK 的安装文件可以从 Oracle 公司的网站 https://www.oracle.com/下载。 JDK 的安装步骤如下。

(1) 双击运行安装文件 jdk-9.0.4\_windows-x64\_bin.exe,会出现 Java SE 开发工 具包安装向导,如图 5-1 所示。



图 5-1 Java SE 开发工具包安装向导

(2) 单击图 5-1 中的"下一步"按钮,会出现 JDK 安装目录的选择界面,指定 JDK 安装目录为 C:\Program Files\Java\jdk-9.0.4,然后单击"下一步"按钮,如图 5-2 所示。

Java(TM) SE Development	Kit 9.0.4 (64-bit) - 定制安装	×
从下面的列表中选择要安装的可选。 所选择的功能	功能。您可以在安装后使用控制面板 功 9 - C 器	中的"添加/删除程序"实用程序更改 能说明 wa(TM) SE Development Kit .0.4 (64-bit), 包括 JavaFX SDK, 一く毎用 JRE 以及 Java Mission ontrol 工具委件。它要求硬盘驱动 計有 180MB 空间。
安装到: C:\Program Files\Java\jdk-9.0.4\		更改( <b>C</b> )
	<上一步(B) []	步( <u>N</u> ) > 取消

图 5-2 选择 JDK 安装目录

(3) 出现 JDK 安装进度条,如图 5-3 所示。

i Java(TM) SE Development Kit 9.0.4 (64-bit) - 进度	-	×
状态: 正在更新组件注册	_	

图 5-3 JDK 安装进度条

(4) 进度条结束后,出现指定 JRE 安装目录的界面。此处指定 JRE 的安装目录为 C:\Program Files\Java\jre-9.0.4,然后单击"下一步"按钮,如图 5-4 所示。

(5) 选择好 JRE 路径,会出现 Java 安装进度条,如图 5-5 所示。

Java 安装 - 定制安装	-		×
Java Java			
定制安装			
单击 "更改" 以将 Java 安装到其他文件夹。			
安装到: C:\Program Files\Java\jre-9.0.4 ☑ 启用浏览器中的 Java 内容	更	友(C)	
		下—步(N	)>

图 5-4 选择 JRE 安装路径

downle 状态:安装 Java			
ATMs, Smartcards, POS Terminals, Blu-ra Set Top (2015), Martin (2015), Se	ay Play	ers, PC Switche	s
Automot Sark Billio	en la	Device Lotte	is is
Devices Run Ja	ava	odules.	
Java #1 Development Platform		ILE"	

图 5-5 Java 安装进度条界面

(6)显示 JDK 安装成功界面,如图 5-6 所示。

(7) JDK 安裝完成后,配置环境变量。新建 JAVA\_HOME 环境变量 JAVA\_ HOME=C:\Program Files\Java\jdk-9.0.4,修改 path 环境变量,在 path 变量尾部添 加%JAVA\_HOME% \ bin。新建 classpath 环境变量 classpath =.;% JAVA\_ HOME% \lib;%JAVA\_HOME% \lib\dt.jar;%JAVA\_HOME% \tools.jar。

JDK 需要配置以上 3 个环境变量 JAVA\_HOME、path 和 classpath; JDK1.5 版本之后可以不再设置 classpath,但建议保留 classpath 设置。

👹 Java(TM) SE Development Kit 9.0.4 (64-bit) - 完成	×
Java(TM) SE Development Kit 9.0.4 (64-bit) 已成功安装	
单击"后续步骤"访问教程, API 文档, 开发人员指南, 发布说明及更多内容, 帮助您开始使用 JDK 。	
后续步骤(\))	
关闭 <b>(C)</b>	

图 5-6 JDK 安装成功界面

### 2. Tomcat 的安装和使用

Tomcat 的安装文件可以从 Apache 公司的官方网站 http://tomcat.apache.org/ index.html 下载,采用绿色版本 apache-tomcat-9.0.0.M9,解压到指定目录下即可。 启动和关闭 Tomcat 服务器的文件位于 Tomcat 主目录下的 bin 文件夹下,文件名分别 为 startup. bat 和 shutdown. bat。

(1) 执行 startup. bat, 启动 Tomcat, 如图 5-7 所示。



图 5-7 启动 Tomcat

(2) Tomcat 正常启动后,打开浏览器,在地址栏输入 URL: http://127.0.0.1: 8080,将看到如图 5-8 所示界面。

(3) Tomcat 提供服务的默认端口号是 8080,当与其他应用程序的端口号发生冲



图 5-8 Tomcat 系统首页

突时,将无法正常启动。此时,可修改 Tomcat 的端口号为其他未被占用的端口号。修改 Tomcat 端口号的位置在 Tomcat 主目录下的 conf 文件夹下,文件名为 server. xml。可以用记事本等文本编辑工具将其打开,重新改写端口号,如图 5-9 所示。



图 5-9 改写 Tomcat 端口号

(4) Tomcat 的主目录下有若干目录,其用途如表 5-1 所示。

目录	用 途
\bin	放置启动和关闭 Tomcat 的可执行文件和批处理文件
\lib	放置 Tomcat 运行所需要加载的 jar 包
\conf	放置 Tomcat 主要的配置文件
\logs	放置 Tomcat 日志文件
\temp	放置 Tomcat 运行时产生的临时文件
\webapps	放置 Web 应用的目录
\work	放置 JSP 页面转换成对应的 Servlet 的目录

表 5-1 Tomcat 目录结构及用途

#### 3. Eclipse 的安装和使用

Eclipse 的安装文件 eclipse-inst-win64. exe 可以从 http://www.eclipse.org/downloads/下载,用户可以下 载解压版使用。

(1) 将 Eclipse 的解压版文件 eclipse-jee-oxygen-2win32-x86\_64. zip 解压之后,双击 eclipse. exe 图标,运行 Eclipse 集成开发环境,如图 5-10 所示。

(2) 打开 Eclipse Launcher 对话框,提示用户选择 Eclipse 的工作空间,如图 5-11 所示。

> 新加卷 (D:) > ecl	ipse
名称	^
Configuration	n
I dropins	
features	
l p2	
plugins	
I readme	
.eclipseprod	uct
artifacts.xml	
eclipse.exe	
🔂 eclipse.ini	
eclipsec.exe	

图 5-10 解压并运行 Eclipse 集成开发环境

elect a dir	ectory as workspace	references and	doualonmont arti	facto
colpse uses	the workspace directory to store its p	references and	development arti	liacts.
Workspace:	D:\webworkspace	~	Browse	
Use this a	is the default and do not ask again			

图 5-11 选择 Eclipse 的工作空间

(3) 打开 Eclipse 后,可在 Servers 选项卡中,配置应用服务器 Tomcat,如图 5-12 所示。

② Markers □ Properties ▲ Servers ※ 難Data Source Explorer Shippets □ Console No servers are available. Click this link to create a new server...

图 5-12 在 Eclipse 中配置应用服务器

(4) 单击创建新服务器链接,会出现如图 5-13 所示的对话框,在对话框中选择解

压或安装过的 Tomcat 应用服务器。

New Server				×
Define a New Server				
Choose the type of server to	create			
272-1252-12				
Select the server type:				
type filter text				
Publishes and runs J2EE and	Java EE Web projects and serv	ver configurat	tions to a	a local
Tomcat server.				
Tomcat server. Server's host name:	localhost			]
Tomcat server. Server's host name: Server name:	localhost Torncat v9.0 Server at localho	ost		]
Tomcat server. Server's host name: Server name: Server runtime environment:	localhost Tomcat v9.0 Server at localh Apache Tomcat v9.0	ost	~	] ] 
Tomcat server. Server's host name: Server name: Server runtime environment:	localhost Tomcat v9.0 Server at localh Apache Tomcat v9.0 Confi	ost gure runtime	environ	Add.
Tomcat server. Server's host name: Server name: Server runtime environment:	localhost Tomcat v9.0 Server at localh Apache Tomcat v9.0 Confi	ost gure runtime	• environ	] Add. ments.
Tomcat server. Server's host name: Server name: Server runtime environment:	localhost Tomcat v9.0 Server at localh Apache Tomcat v9.0 Confi	ost gure runtime	• environ	] Add. ments.

图 5-13 在 Eclipse 中关联 Tomcat 应用服务器

(5) 在 Eclipse 中关联 Tomcat 后,需要进行服务器名、主机名、运行时环境、服务器位置、部署路径等配置,如图 5-14 所示。

General Inform	nation		
Specify the hos	t name ar	nd other common settings.	
Server name:		Tomcat v9.0 Server at localhost	
Host name:		localhost	
Runtime Envir	onment:	Apache Tomcat v9.0	~
Configuration	path:	/Servers/Tomcat v9.0 Server at localho	Browse
Open launch o	onfigura	tion	
- Server Locati	ons		
Specify the serv published with	er path (i no modu	i.e. catalina.base) and deploy path. Serve les present to make changes.	er must be
OUse worksp	ace meta	data (does not modify Tomcat installatio	on)
Use Torncat	installati	ion (takes control of Tomcat installation)	10
⊖Use custom	location	(does not modify Tomcat installation)	
	5110 30	-2\ENIV10\apache_tomcat_0.0.22	Browse
Server path:	E:/18-50	v.s.triate i 2 Japacine - connear-2-overe	DIVITISCO
Server path: Set deploy pa	th to the	default value	Diotisea

图 5-14 Tomcat 服务器配置

关于 Eclipse 中集成的 JDK 版本和安装位置,可以在如图 5-15 所示的对话框中进行查看和修改。

可在如图 5-16 所示的对话框中查看和编辑 Eclipse 中的运行时环境关联情况,即 Tomcat 应用服务器的版本选择。

单击图 5-16 中的 Edit 按钮,即可进入图 5-17 所示的对话框,编辑 Eclipse 的运行时环境配置,包括 Tomcat 的版本、位置和 JDK 版本。

type filter text		Installed IRFs		0.40.4
> Data Management Gradle > Help > Install/Update	^	Add, remove or edit to the build path of Installed JREs:	JRE definitions. By default, the check newly created Java projects.	ed JRE is added
🗸 Java		Name	Location	Add
> Appearance > Build Path		⊠ mijdk1.8.0_19	E:\19-20-2\ENV19\jdk18\bin	Edit
Code Coverage		-		Duplicate_
> Code Style				Remove
Somptime     Somptime				Search
<ul> <li>Server Audio</li> </ul>		٢	>	
Launching Overlays Profilers	÷			Apply

图 5-15 Eclipse 中的 JDK 版本和安装位置

ape inter text		Server Runtime Environ	nments	0.0.0.
> Code Style > Compiler > Debug	^	Add, remove, or edit serve Server runtime environme	er runtime environments. ents:	
<ul> <li>&gt; Editor</li> <li>&gt; Editor</li> <li>&gt; Editor</li> <li>&gt; Editor</li> <li>&gt; Editor</li> <li>&gt; Invalled JREs</li> <li>JUnit</li> <li>Properties Files Editor</li> <li>&gt; Java Persistence</li> <li>&gt; Java Persistence</li> <li>&gt; Java Persistence</li> <li>&gt; Java Script</li> <li>&gt; SGN</li> <li>&gt; Mayon</li> <li>&gt; Oomph</li> <li>&gt; Plag- in Development</li> <li>&gt; Plag- in Development</li> <li>&gt; RamObebug</li> <li>&gt; Server</li> <li>- Audio</li> <li>Laurching</li> <li>Overlays</li> <li>&gt; Terminal</li> <li>&gt; Terminal</li> <li>&gt; Vatiliation</li> <li>&gt; Web</li> <li>&gt; Web</li> <li>&gt; Web</li> </ul>		Name 3 Apache Tomcat v9.0	Type Apache Tomcat v9.0	Add Edit Remove Search Columns

图 5-16 Eclipse 中的运行时环境设置

### 4. MySQL 的安装和配置

本书的实训项目"智能停车场",将车位信息数据、用户数据等存储于 MySQL 数据库。

(1)从 MySQL 数据库的官网 http://www.mysql.com 下载与操作系统匹配的 MySQL 安装文件,单击后进入的首页如图 5-18 所示。

Runtime Environmen	t				×
r					
stallation directory					inter a
at v9.0					
ation directory:					
IV19\apache-tomcat-	9.0.22		B	rowse	
			Downloa	ad and I	install.
		~	Insta	alled JRI	t.s
<b>D</b>	Martin	field		Con	
	Runtime Environmen stallation directory at v9.0 ation directory: IV19\apache-tomcat-	Runtime Environment stallation directory at v9.0 ation directory: IV19\apache-tomcat-9.0.22	Runtime Environment	Runtime Environment stallation directory at v9.0 ation directory: IV19\apache-tomcat-9.0.22 B Downloa V Insta	Runtime Environment

图 5-17 编辑服务器运行时环境

2	winkth ment paputer-sper-inservicilations = 0,	Cantact MySQL   Login   Register
MySQL M	SQLCOM DOWNLOADS DOCUMENTATION DEVELOPER ZONE	f ⊮ in ≟i
MySQL Ent	erprise Edition	
MySQL Enterprise Edition inc	udes the most comprehensive set of advanced features and management tools for MySQL	
MyGQL Database     MyGQL Database     MyGQL Contexting Lifter,	NdR MySMA etc.) DBC: AvE. etc.) Here Downised That Downised Market All States MySMA Strangthere Transported Data Ecocycleon (TDR) A MySMA Strangthere Transported Transported Data Ecocycleon (TDR) A MySMA Strangthere Transported Transported Transported Data Ecocycleon (TDR) A MySMA Strangthere Transported	
Contact Sales	MvSQL Cluster CGE	
USA: <u>x1.806.221.0634</u> Canada: <u>x1.806.221.0634</u> Germany: <u>x81.85.143.01280</u>	MyGQL Clutter is a real-time open source transactional database designed for fast, always on access to data under high throughp MyGQL Clutter MyGQL Clutter MyGQL Clutter	ut conditions.

图 5-18 MySQL 官网首页

(2) 单击 DOWNLOADS→Community,选择 MySQL Community Server,如 图 5-19 所示。

M-COLM	C A Di Olivera e eleficanti	
MySQL full Repository	<ul> <li>C API (iibmysdicient)</li> </ul>	
MySQL APT Repository	<ul> <li>Connector/C++</li> </ul>	
MySQL SUSE Repository	- Connector/]	
	- Connector/NET	
MySQL Community Server	- Connector/Node.js	
MySQL Cluster	<ul> <li>Connector/ODBC</li> </ul>	
MySQL Router	- Connector/Python	
MySQL Shell	MySQL Native Driver for PHP	
MySQL Workbench		
	MySQL Benchmark Tool	
MySQL Installer for Windows	<ul> <li>Time zone description tables</li> </ul>	
MySQL for Excel	- Download Archives	
MySQL for Visual Studio		
MySQL Notifier		
© 2020, Oracle Corporation and/or its affiliates		

图 5-19 选择下载 MySQL Community Server

- General Availability (GA) Releases Archives MySQL Community Server 8.0.21 Select Operating System: Looking for previous GA ۲ Microsoft Windows Recommended Download: MySQL Installer for Windows All MySQL Products. For All Windo In One Package. ws Platfe ows (x86, 32 & 64-bit), MySQL Installer MSI Go to Download Page > Other Downloads Windows (x86, 64-bit), ZIP Archive 8.0.22 191.4M (mysql-8.0.22-winx64.zip) MD5: 478454426687651664606690977 Windows (x86, 64-bit), ZIP Archive 8.0.22 434.4M De Debug Binaries & Test Suite (mysql-8.0.22-winx64-debug-test.zip) MD5: 9e162d102s9692f8e76970b074480567 | Signature We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.
- (3) 找到 Recommended Download,单击 Go to Download Page,如图 5-20 所示。

图 5-20 下载 Windows 版本 MySQL 安装文件

(4)进入下载界面,如图 5-21 所示;选择 No thanks, just start my download 开始下载。

An Oracle Web Account provides you with the following advantages:	ogin	Now or Sign Up for a free account.
Fast access to MySQL software downloads     Download technical White Papers and Presentations     Post messages in the MySQL Discussion Forums     Report and track bugs in the MySQL bug system     Using my Oracle Web account     MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account,     click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link     and following the instructions.	n Orac	le Web Account provides you with the following advantages:
Download technical White Papers and Presentations     Post messages in the MySQL Discussion Forums     Report and track bugs in the MySQL bug system     Using my Oracle Web account     MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account,     click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link     and following the instructions.	- Fast	access to MySQL software downloads
Post messages in the MySQL Discussion Forums     Report and track bugs in the MySQL bug system      Login >>     using my Oracle Web account      MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account,     click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link     and following the instructions.	. Dow	nload technical White Papers and Presentations
Report and track bugs in the MySQL bug system      Login >     using my Oracle Web account      MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions.	. Post	messages in the MySQL Discussion Forums
Login » using my Oracle Web account MySQLcom is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions.	- Rep	ort and track bugs in the MySQL bug system
	My clia an	SQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, k the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link d following the instructions.
	No tha	nks, just start my download.
No thanks, just start my download.		
No thanks, just start my download.		
No thanks, just start my download.	ORA	CLCE © 2020, Oracle Corporation and/or its affiliates
Corporation and/or its affiliates		

图 5-21 MySQL 下载界面

(5) 在图 5-22 所示的界面中,选择 MySQL 安装类型为 Developer Default,单击 Next 按钮,进入下一步。



图 5-22 选择 MySQL 安装类型

(6) 在如图 5-23 所示的界面中,检查 MySQL 的安装条件。如需要 Microsoft Visual C++包的支持,则选择补充安装。



图 5-23 检查 MySQL 安装条件

(7) 在图 5-24 所示的界面中,单击 Execute 按钮执行安装,在后续的界面中,均可默认单击 Next 按钮,如图 5-25 所示。

MySQL Installer			-		×
MySQL. Installer Adding Community	Installation The following products will be installe	d.			
Choosing a Setup Type Check Requirements	Product	Status Ready to download	Progress	Notes	
Installation					
Product Configuration					
	Click [Execute] to install the following	packages.			
		< Back	Execute	Canc	el

图 5-24 Installation 界面

MySQL Installer		- 🗆 X
MySQL. Installer Adding Community	Product Configuration	wizard for each of the following products.
Choosing a Setup Type	You can cancel at any point if you wish to products.	to leave this wizard without configuring all the
Check Requirements	Product MySQL Server 8.0.21	Status Configuration complete.
Product Configuration		
Installation Complete	¢	>
		Next > Cancel

图 5-25 Product Configuration 界面

(8) 在如图 5-26 所示的 High Availability 界面中,选中 Standalone MySQL Server/Classic MySQL Replication 单选按钮。



图 5-26 High Availability 界面

(9) 在如图 5-27 所示的界面中,设置 MySQL 的服务配置类型和网络设置。

MySQL Installer	· 列			-	
MySQL. Installer	Type and Networ	king			
MySQL Server 8.0.21	Server Configuration Type				
	Choose the correct server of define how much system r	configuration type for esources are assigne	or this MySQL Se d to the MySQL	rver installation. This Server instance.	setting will
High Availability	Config Type: Developme	nt Computer			~
Type and Networking	Connectivity				
Authentication Method	Use the following controls	to select how you w	ould like to conr	nect to this server.	
	TCP/IP	Port:	3306	X Protocol Port:	33060
Accounts and Roles	🗹 Open Window	s Firewall ports for n	etwork access		
Windows Service	Named Pipe	Pipe Name:	MYSQL		
Apply Configuration	Shared Memory	Memory Name:	MYSQL		
	Advanced Configuration				
	Select the check box below and logging options for th	v to get additional co is server instance.	onfiguration pag	es where you can set	advanced
	Show Advanced a	nd Logging Options			
			< Back	Next >	Cancel

图 5-27 网络设置对话框

(10) 在如图 5-28 所示的界面中,设置 MySQL 的 root 用户密码,然后单击 Next 按钮,进入下一步。

MySQL Installer				- 0	×
MySQL. Installer	Accounts and Ro	oles			
	Enter the password for the place.	root account. Pleas	e remember to store this p	bassword in a secure	
High Availability	MySQL Root Password:	••••			
Type and Networking	Repeat Password:	••••			
		Password streng	th: Weak		
Windows Service Apply Configuration	MySQL User Accounts Create MySQL user accounts	ints for your users a	nd applications. Assign a r	ole to the user that	
	Consists of a set of privile MySQL User Name	ges. Host	User Role	Add Use	r
					-
				Delete	

图 5-28 root 密码设置

(11) 在图 5-29 所示的界面中,将 MySQL 服务配置为 Windows 服务,并配置在 系统启动时自动启动服务,启动服务时使用标准系统账户等信息。

MySQL Installer			-		×
MySQL. Installer MySQL Server 8.0.21	Windows Servic Configure MySQL Ser Windows Service Details	e ver as a Windows Service			
High Availability	Please specify a Windows A unique name is require	Service name to be used for this MySQL Server inst d for each instance.	ance.		
Type and Networking	Windows Service Name:	MySQL80			
Authentication Method	Start the MySQL S	ierver at System Startup			
Accounts and Roles					
Windows Service	Run Windows Service as a The MySQL Server needs	 to run under a given user account. Based on the sec tern you need to nick one of the ontions below.	urity		
Apply Configuration	Standard System	Account			
	Recommended for	or most scenarios.			
	O Custom User				
	An existing user a	account can be selected for advanced scenarios.			
		< Back Next >		Canc	el

图 5-29 Windows 服务设置

(12) 在图 5-30 所示的界面中,单击 Execute 按钮,即可由系统进行关闭现有服

务、写配置文件、更新防火墙、启动服务、应用安全设置等安装进程了。全部执行完毕 后会出现图 5-31 所示的界面。

S MySQL Installer		- 🗆 X
MySQL. Installer MySQL Server 8.0.21	Apply Configuration Click [Execute] to apply the changes	
High Availability Type and Networking Authentication Method Accounts and Roles Windows Service Apply Configuration	Configuration Steps Log Writing configuration file Updating Windows Firewall rules Adjusting Windows service Initializing database (may take a long time) Starting the server Applying security settings Updating the Start menu link	
	< Back	Execute Cancel

图 5-30 执行安装过程

MySQL Installer		-		х
MySQL. Installer MySQL Server 8.0.21	Apply Configuration The configuration operation has completed. Configuration Steps Log			_
High Availability Type and Networking Authentication Method Accounts and Roles Windows Service Apply Configuration	<ul> <li>Writing configuration file</li> <li>Updating Windows Firewall rules</li> <li>Adjusting Windows service</li> <li>Initializing database (may take a long time)</li> <li>Starting the server</li> <li>Applying security settings</li> <li>Updating the Start menu link</li> </ul>			
	The configuration for MySQL Server 8.0.21 was successful. Click Finish to continue.		Finis	h

图 5-31 完成应用配置

MySQL Installer -□ X MySQL. Installer **Product Configuration** Adding Community We'll now walk through a configuration wizard for each of the following products. You can cancel at any point if you wish to leave this wizard without configuring all the products. Choosing a Setup Type Product Status MySQL Server 8.0.21 Configuration complete. < Next > Cancel

(13) 进入产品配置界面,如图 5-32 所示,安装完成界面如图 5-33 所示。

图 5-32 Product Configuration 界面

MySQL Installer		-		×
MySQL. Installer Adding Community	Installation Complete The installation procedure has been completed.			
Choosing a Setup Type Check Requirements Installation	Copy Log to Clipboard			
Product Configuration				
			Finis	h

(14) 在图 5-34 所示的界面中,通过 Windows 的服务组件管理 MySQL 服务的启

图 5-33 MySQL 安装完成界面

品 服务						- 0	×
文件(F) 操作(A)	查看(V) 帮助(H)						
		₽					
。服务(本地)	<ol> <li>服务(本地)</li> </ol>						
	MySQL80	名称 ^	描述	状态	启动类型	登录为	^
		AmessagingService_1082de	支持		手动(触发	本地系统	
	停止此服务	《 Microsoft (R) 诊断中心标	诊断		手动	本地系统	
	<u>智停</u> 此服务 第中动业服务	Alicrosoft Account Sign-i	支持		手动(触发	本地系统	
	<u>mi/mi/////////////////////////////////</u>	A Microsoft App-V Client	Man		慧用	本地系统	
		Alicrosoft Defender Antiv	帮助	正在	手动	本地服务	
		Alicrosoft Defender Antiv	帮助	正在	自动	本地系统	- 1
		Alicrosoft iSCSI Initiator	管理		手动	本地系统	
		A Microsoft Passport	为用		手动(触发	本地系统	- 1
		Alicrosoft Passport Cont	管理		手动(触发	本地服务	
		Alicrosoft Software Shad	管理		手动	本地系统	
		A Microsoft Storage Space	Micr		手动	网络服务	
		④ Microsoft Store 安装服务	为 M	正在	手动	本地系统	
		Alicrosoft Windows SMS	根据		手动(触发	本地服务	
		MySQL80		正在	自动	网络服务	
		Ret.Tcp Port Sharing Ser	提供		禁用	本地服务	
		A Netlogon	为用		手动	本地系统	
		Retwork Connected Devi	网络		手动(触发	本地服务	
		Retwork Connection Bro	允许	正在	手动(触发	本地系统	
		Retwork Connections	管理		手动	本地系统	
		10 Notwork Connectivity Acc	10./11		ar an/én4è	*****	

动方式,可设置为自动或手动,可设置为本地系统账户登录。

图 5-34 测试连接 MySQL 服务

(15) 安装完成后进入 MySQL 的安装目录,进入 MySQL Sever,其目录下的文件 如图 5-35 所示。

□ > 此电脑 >	Windows (C:) > Program Files > MySQL > MySQL Server 8.0 >
项目训练- ^	📗 bin
	docs
	I include
	📜 lib
	I share
	COPYING
	README

图 5-35 MySQL Server 8.0 目录结构

(16) bin 目录下保存了 MySQL 常用的命令工具以及管理工具、data 目录是 MySQL 默认用来保存数据文件以及日志文件的地方(我的因刚安装还没有 data 文件 夹)、docs 目录下是 MySQL 的帮助文档、include 目录和 lib 目录是 MySQL 所依赖的 头文件以及库文件、share 目录下保存目录文件以及日志文件。

进入 bin 目录,按住 Shift 键,然后单击鼠标右键可以选择在该目录下打开命令窗口,或者在地址栏中输入 cmd 进入命令窗口,输入 mysql-u root-p 后按 Enter 键,然后 会提示输入密码,输入密码后就会进入 MySQL 的操作管理界面。

#### 5. Navicat 的安装和使用

Navicat 是一套数据库开发工具,用户可以利用该工具同时连接 MySQL、 MariaDB、MongoDB、SQL Server、Oracle、PostgreSQL 和 SQLite 数据库。它与 Amazon RDS、Amazon Aurora、Amazon Redshift、Microsoft Azure、Oracle Cloud、 MongoDB Atlas、阿里云、腾讯云、华为云等云数据库兼容,通过 Navicat 可以在可视化 界面下快速轻松地创建、管理和维护数据库,下载网址是 https://www.navicat.com. cn/,其安装过程比较简单,在此不再赘述。

### 5.1.2 数据库访问工具类的封装

服务器采用 JDBC 技术与数据库连接,并访问数据库内容。JDBC(Java Database Connectivity, Java 数据库连接)是一种用于数据库访问的 Java API(Application Programming Interface,应用程序设计接口),由一组用 Java 语言编写的类和接口组成。有了 JDBC,就可以用纯 Java 语言和标准的 SQL 语句编写完整的数据库应用程序,并且真正实现软件的跨平台性。

简单地说,JDBC 能完成下列三件事。

- (1) 为同一个数据库建立连接。
- (2) 向数据库发送 SQL 语句。
- (3) 处理数据库返回的结果。

为便于项目开发的层次化和模块化,将服务器访问数据库的流程封装为工具类 DBTool,将数据库的连接、关闭、查询、更新、分页显示方法封装于 DBTool 类,其类图 如图 5-36 所示。

DBTool(Constructor)
-String driver(Set)
-String url(Set)
-String username(Set)
-String password(Set)
-Connection con
-PreparedStatement pstmt
+static final long PAGE_REC_NUM
-void init( )
-void close( )
-void setParams(String[] params)
+int update(String sql,String[] params)
+int update(String sql)
+List <map<string,string>&gt; getList(String sql,String[] params)</map<string,string>
+List <map<string,string>&gt; getList(String sql)</map<string,string>
+Map <string,string> getMap(String sql,String[] params)+</string,string>
Map <string,string> getMap(String sql)</string,string>
<ul> <li>List<map<string,string>&gt; getListFromRS(ResultSet rs)throws SQLException</map<string,string></li> </ul>
+Map <string,object> getPage(String sql,String[] params,String curPage)</string,object>

图 5-36 DBTool 类图

该工具类将访问数据库的连接关闭、参数设置,结果集到 Java 集合类对象的转换 都定义为私有方法,仅将查询方法、更新方法、和分页显示方法设置为公有方法,供其 他层次的开发者调用,从而提高了代码复用率,进而提高了项目开发效率。

DBTool 代码如下。

```
package util;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
public class DBTool {
    private String driver;
    private String url;
    private String username;
    private String password;
    private Connection con;
    private PreparedStatement pstmt;
    public static final long PAGE REC NUM = 8;
    public void setDriver(String driver) {
        this.driver = driver;
    public void setUrl(String url) {
        this.url = url;
    public void setUsername(String username) {
        this.username = username;
    }
    public void setPassword(String password) {
        this.password = password;
    public DBTool() {
        driver = "com.mysql.cj.jdbc.Driver";
        url = "jdbc:mysql://localhost:3306/meal?serverTimezone = UTC";
        username = "root";
        password = "root";
    }
```

```
private void init() {
    try {
        Class.forName(driver);
        con = DriverManager.getConnection(url, username, password);
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
private void close() {
    if(pstmt!= null)
        try {
             pstmt.close();
         } catch (SQLException e) {
             e.printStackTrace();
    if(con!= null)
        try {
             con.close();
         } catch (SQLException e) {
             // TODO Auto - generated catch block
             e.printStackTrace();
         }
}
private void setParams(String[] params) {
    if(params!= null) {
        for(int i = 0; i < params.length; i++) {</pre>
             try {
                 pstmt.setString(i+1, params[i]);
             } catch (SQLException e) {
                 // TODO Auto - generated catch block
                 e.printStackTrace();
             }
        }
    }
}
public int update(String sql,String[] params) {
    int result = 0;
    init();
    try {
        pstmt = con.prepareStatement(sql);
        setParams(params);
        result = pstmt.executeUpdate();
```

```
} catch (SQLException e) {
        // TODO Auto - generated catch block
        e.printStackTrace();
    }finally {
        close();
    return result;
}
public int update(String sql) {
    return update(sql, null);
}
public List < Map < String, String >> getList(String sql, String[] params){
    List < Map < String, String >> list = null;
    init();
    try {
        pstmt = con.prepareStatement(sql);
        setParams(params);
        ResultSet rs = pstmt.executeQuery();
        list = getListFromRS(rs);
        rs.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }finally {
        close();
    }
    return list;
3
private List < Map < String, String >> getListFromRS(ResultSet rs) throws SQLException {
    List < Map < String, String >> list = new ArrayList < Map < String, String >>();
    ResultSetMetaData rsmd = rs.getMetaData();
    while(rs.next()) {
        Map < String, String > m = new HashMap < String, String >();
        for(int i = 1; i <= rsmd.getColumnCount(); i++) {</pre>
             String colName = rsmd.getColumnLabel(i);
             String value = rs.getString(colName);
             if(value != null) {
                 m.put(colName, value);
        list.add(m);
    }
    return list ;
}
public List < Map < String, String >> getList(String sql){
    return getList(sql, null);
```

```
3
    public Map < String, String > getMap(String sql, String[] params){
        Map < String, String > m = new HashMap < String, String >();
        List < Map < String, String >> list = getList(sql, params);
        if(list!= null&&list.size()!= 0) {
            m = list.get(0);
        }
        return m;
    }
    public Map < String, String > getMap(String sql){
        return getMap(sql, null);
    public Map < String, Object > getPage(String sql, String[] params, String curPage){
        Map < String, Object > page = new HashMap < String, Object >();
        String newSql = sql + " limit " + (Long.parseLong(curPage) - 1) * PAGE REC NUM
+ "," + PAGE REC NUM;
        List < Map < String, String >> pageList = getList(newSql, params);
        sql = sql.toLowerCase();
        String countSql = "";
        if(sql.indexOf("group")>= 0) {
            countSql = "select count( * ) as tempNum from(" + sql + ") as temp";
        }
        else {
             countSql = "select count(*) as tempNum " + sql. substring(sql. indexOf
("from"));
        String count_s = (String)getMap(countSql, params).get("tempNum");
        long count = Long.parseLong(count_s);
        long totalPage = 0;
        if(count % PAGE_REC_NUM == 0)
             totalPage = count/PAGE REC NUM;
        else
        page.put("list", pageList);
        page.put("totalPage", totalPage);
        page.put("recNum", PAGE_REC NUM);
        return page;
    }
}
```

# 5.1.3 服务器端数据管理功能

服务器端的数据管理功能涉及项目的具体业务逻辑。例如,智能停车场系统,其 主要业务逻辑包括车辆进出场的 RFID 刷卡识别、进出场时长统计、计费管理等。如 果是环境监测系统,则主要的业务逻辑包括环境数据的显示、实时更新和报表统计。 无论项目的具体业务逻辑如何,在服务器端对项目数据的管理,基本上涵盖增加、删除、修改、查询这些典型的业务逻辑。下面以用户管理为例,阐述服务器端的数据管理功能。

(1) 在 Eclipse 中新建动态 Web 工程, 如图 5-37 所示。

eb project or add it	to a new or existi	ng Enterprise Ap	plication.
\IOTtraining			Browse
		. •	New Runtime.
ne Tomcat v9.0		~	Modify
ng with Apache Ton ty to the project.	ncat v9.0 runtime.	Additional facet	s can later be
		~	New Project.
			New
	eb project or add it \/OTtraining ne Tomcat v9.0 ng with Apache Ton ty to the project.	eb project or add it to a new or existin \/OTtraining he Tomcat v9.0 ng with Apache Tomcat v9.0 runtime. ty to the project.	eb project or add it to a new or existing Enterprise Ap \/OTtraining

图 5-37 在 Eclipse 中新建动态 Web 工程

(2) 单击 Next 按钮,将输出文件夹改为 WebContent\WEB-INF\classes,如图 5-38 所示。

(3) 单击 Finish 按钮,即可新建 Web 工程。在 WEB-INF 的 lib 文件夹下复制 MySQL 驱动 jar 包,如图 5-39 所示。

(4) 在工程名上右击,在弹出的快捷菜单中选择 build path→cofigure build path 选项,进入如图 5-40 所示的界面。

(5) 单击 Add JARs 按钮,进入 jar 包选择和添加界面,如图 5-41 所示。

(6) 导入 jar 包之后,单击 OK 按钮,完成数据库驱动包的导入。

下面即可新建工具类 DBTool,在工程目录树的 src 处右击,新建 class,指定包名 和类名,如图 5-42 所示。

编写 DBTool 的代码,如 5.1.2 小节数据库访问工具类的封装。下面就可以开始 用户管理功能的实现了。用户管理功能基于数据库中的 user 表。在 MySQL 中的数 据库名为 parking。

New Dynamic Web Pro	ject			
Java				12h
Configure project for buil	ding a Java application.			
Source folders on build pa	th:			
@ src				Add Folder
				Edit
				Remove
Default output folder:				
WebContent\WEB-INF\cla	sses			
	. Death	Nexts	Flatab	Consul
(J)	< Back	Next >	Finish	Cancel

图 5-38 修改 Web 工程输出文件夹







图 5-40 configure build path 界面

terre Olter trut	n.	
type inter text		_
> ≌ ch02		
> 🖻 ch03		
> 🖻 ch05		
⇒ ≅ ch4		
✓ ☞ IOT training		
> le .settings		
<ul> <li>WebContent</li> </ul>		
> B MEIA-INF		
✓ ▷ WEB-INF		
v ⊵ lib		
iii mysql-connector-j	ava-8.0.19.jar	
.classpath		
i .project		
> We Servers		
		-

图 5-41 选择并添加 jar 包

		-
Java Class		C
Create a new Java	class.	
Source folder:	IOTtraining/src	Browse
Package:	tools	Browse
Enclosing type:		Browse
Name:	DBTool	
Modifiers:	public O package O private O protected     abstract I final I static	
Superclass:	java.lang.Object	Browse
Interfaces:		Add_
		Remove
Which method stu	bs would you like to create?	
	public static void main(String[] args)	
	Constructors from superclass	
	Inherited abstract methods	
Do you want to ad	d comments? (Configure templates and default value he	ere)
	Generate comments	

图 5-42 创建 DBTool 类

在 parking 数据库中的 user 表,定义了智能停车场的用户信息,用户在停车场中 凭 RFID 刷卡出入,如图 5-43 所示。

	cardid	name	count	license	phone	opendate
,	2B2F	raindy	5	翼A1234	987654321	2017-10-25 15:28:03
	3F5N	123	5	京A0000	1234567890	2017-10-24 16:19:08

图 5-43 parking 数据库中的 user 表内容

首先创建 UserService 类。封装停车场用户管理的业务逻辑,包括用户开卡、注销、修改停车卡信息和查询停车卡信息 4 个业务逻辑。UserService 位于 model 包下, 其类图如图 5-44 所示。

```
UserService (Constructor)
```

```
-DBTool db
+List<Map<String,String>> getUsers(String name)
+int delUser(String cardid)
+Map<String,String> getUser(String cardid)
+int updateUser(String cardid,String name,String count,String license,String phone)
+int addUser(String cardid,String name,String count,String license,String phone)
```

图 5-44 UserService 类图

UserService 代码如下所示。

```
package model;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import tools.DBTool;
public class UserService {
    private DBTool db;
    public UserService() {
        db = new DBTool();
    public List < Map < String, String >> getUsers(String name){
        List < Map < String, String >> list = new ArrayList < Map < String, String >>();
        String sql = "select * from user";
        String[] params = null;
        if(name!= null) {
             sql = sql + " where name like ?";
             params = new String[] {"%" + name + "%"};
        list = db.getList(sql, params);
        return list;
    )
    public int delUser(String cardid) {
        String[] params = {cardid};
        String sql = "delete from user where cardid = ?";
        return db.update(sql, params);
    }
```

```
public int addUser(String cardid, String name, String count, String license, String
phone) {
         String sql = "select * from user where cardid = ?";
        List < Map < String, String >> list = db.getList(sql,new String[] {cardid});
         if(list.size()!= 0)
             return 0;
         else {
             sql = "insert into user values(?,?,?,?,now())";
             return db.update(sql, new String[] {cardid, name, count, license, phone});
         }
    }
    public Map < String, String > getUser(String cardid){
         Map < String, String > user = new HashMap < String, String >();
         String sql = "select * from user where cardid = ?";
        user = db.getMap(sql, new String[] {cardid});
         return user;
    }
    public int updateUser(String cardid, String name, String count, String license, String
phone) {
         String sql = "update user set name = ?, count = ?, license = ?, phone = ?, opendate
= now() where cardid = ?";
         return db.update(sql, new String[] {name, count, license, phone, cardid});
    }
}
```

查询用户信息的界面为 list\_user. jsp,代码如下。

```
<% @page import = "java.util.Map" %>
<% @page import = "java.util.List" %>
<% @page import = "model.UserService" %>
<% @ page language = "java" contentType = "text/html; charset = UTF - 8"
    pageEncoding = "UTF - 8" % >
<! DOCTYPE html >
< html >
< head >
< meta charset = "UTF - 8">
<title>Insert title here </title>
</head>
< body >
<% String s_un = request.getParameter("name");
UserService us = new UserService();
List < Map < String, String >> users = us.getUsers(s_un);
%>
<div align = "center">
< form action = "list users.jsp" method = "post">
<input type = "text" name = "s_un" placeholder = "请输入用户名查询">
```

```
<input type = "submit" value = "搜索">
</form>
<a href = "add_user.html">添加停车用户卡</a>
序号
卡号
用户名
停车次数
车牌号
电话
>办卡时间
操作
< %
int num = 0;
for(Map < String, String > user:users) {
  num++;
  %>
  >
  << = num %>
  < % = user.get("count") %>
  <% = user.get("license") %>
  <a href="del_user.jsp?cardid=<% = user.get("cardid")%>">删除</a>
  <a href = "edit_user.jsp?cardid = <% = user.get("cardid")%>">修改</a>
  < %
}
%>
</div>
</body>
</html>
```

删除用户信息的界面为 del\_user. jsp,代码如下。

```
< % @ page import = "java.util.Map" % >
< % @ page import = "java.util.List" % >
< % @ page import = "model.UserService" % >
< % @ page language = "java" contentType = "text/html; charset = UTF - 8"
    pageEncoding = "UTF - 8" % >
<! DOCTYPE html >
```

```
< html >
< head >
< meta charset = "UTF - 8">
<title>删除用户</title>
</head>
< body >
< %
String cardid = request.getParameter("cardid");
UserService us = new UserService();
int r = us.delUser(cardid);
if(r == 1)
   out.println("删除用户成功!");
else
    out.println("删除用户失败");
8>
<a href = "list users.jsp">返回用户列表</a>
</body>
</html>
```

用户的增加和修改功能,用户可以参照以上的用户列表和用户删除功能,自行实现,在此不再赘述。

### 5.1.4 为硬件端和移动端提供服务

作为物联网服务器,除了门户展示和信息管理功能之外,另一个重要功能是为物 联网系统中的硬件端和移动端提供服务,使得硬件端实时采集的数据可以上传至服务 器持久化存储,并由服务器进行后续的数据分析处理。同时使得移动端可以通过服务 器实时查看数据变化,或发送指令给服务器,硬件端查询后,执行器件做出相应的 改变。

下面以智能停车场的车位泊车信息为例,实现硬件端接口和移动端接口。智能停车场的车位信息对应数据库中的 psd 表,其表结构如图 5-45 所示。





其中,node 表示停车位节点,status 表示停车位状态。每个停车位上方部署有超 声测距传感器,测量传感器与地面距离。没有车辆泊车的情况下,距离约为 3.5m,当 有车辆泊车的情况下,测距结果小于 2.5m。依据这个原理进行车位空闲和占用的判 断。硬件端会将测距结果发送给服务器,服务器端则修改 psd 表中的 status 状态,从 而存储车位的泊车状态。 移动端通过网络通信框架访问服务器端,查询 psd 表中对应的车位节点的 status 状态,从而实现用户通过 Android APP 远程查看车位占用情况。

因此,服务器提供两套接口,分别为硬件端和移动端。硬件端组成 ZigBee 网络后,终端节点部署于每个车位上方,I/O 端口连接超声测距传感器,检测车位状态,发送给协调器,协调器将信息发送至服务器。协调器与服务器端的接口采用 Servlet 编写,代码如下。

Coord. java

package parkinglot;

```
import java. io. IOException;
import java. sql. Date;
import java. sql. Timestamp;
import java. text. DateFormat;
import java. text. SimpleDateFormat;
import java. util. Iterator;
import java. util. List;
import java. util. Locale;
import java. util. Map;
```

```
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 utils.DBUtil;
```

```
/**
* Servlet implementation class Coord
* /
@WebServlet("/Coord")
public class Coord extends HttpServlet {
    private static final long serialVersionUID = 1L;
```

```
/ **
 * @ see HttpServlet # HttpServlet()
 * /
public Coord() {
    super();
    // TODO Auto - generated constructor stub
}
/ **
```

\* @ see HttpServlet # doGet(HttpServletRequest request, HttpServletResponse response)

```
* /
    protected void doGet (HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        DBUtil db;
        db = new DBUtil();
        String sql;
        String cardId = null;
        cardId = request.getParameter("card");
        String role = null;
        role = request.getParameter("role");
        String dis = null;
        dis = request.getParameter("dis");
        String temp = null;
        String hu = null;
        temp = request.getParameter("temp");
        hu = request.getParameter("hu");
        System.out.println("yes");
        if(cardId != null) {
            if(role != null) {
                 if(role.equals("enter")) {
                   sql = "insert into card(cardid, starttime, status) VALUES('" + cardId
+ "',now(),'泊车');";
                     db.update(sql);
                     sql = "select * from card where cardid = '" + cardId + "';";
                     List < Map < String, String >> list = db.getList(sql);
                     Iterator < Map < String, String >> iter = list.iterator();
                     //inverse control relay to control machine
                     if (list.isEmpty()){
                         response.getWriter().append("0");
                     } else {
                          response.getWriter().append("1");
                     }
                 } else if (role.equals("exit")) {
                     //update endtime
                     sql = "update card set endtime = now() where cardid = '" + cardId + "';";
                     db.update(sql);
                     //acquire charge rule -- fee
                     sql = "select fee from chargerule;";
                     Map < String, String > res = db.getMap(sql);
```

```
String feeString = res.get("fee");
                     int fee = Integer.parseInt(feeString);
                     int charge = 0;
                     //acquire timestamp through cardid
                     sql = "select * from card where cardid = '" + cardId + "';";
                     res = db.getMap(sql);
                     //calculate min
                     try {
                         Timestamp times = string2Time(res.get("starttime"));
                         Timestamp timee = string2Time(res.get("endtime"));
                         long nd = 1000 * 24 * 60 * 60;
                         long nh = 1000 * 60 * 60;
                         long nm = 1000 * 60;
                         // ms
                         long diff = timee.getTime() - times.getTime();
                         // min
                         long min = diff % nd % nh / nm;
                                  charge = (int) ((min/15) * fee);
                     } catch (Exception e) {
                         e.printStackTrace();
                     }
                     //update status and charge
                     sql = "update card set status = '离开', charge = '" + charge + "'
where cardid = '" + cardId + "';";
                     db.update(sql);
                      request.getRequestDispatcher("UpdateUser?cardid = " + cardId).
forward(request, response);
                }
            }
        }
        if(dis != null && role != null) {
            int disint = Integer.parseInt(dis);
            //no car
            if(disint > = 100) {
                 sql = "update psd set status = '空闲' where node = '" + role + "';";
             } else {
                 sql = "update psd set status = '泊车' where node = '" + role + "';";
             }
            db.update(sql);
        }
```

```
if(temp != null && hu != null && role != null) {
            System.out.println("温湿度");
            sql = "insert into dht11(temp, hu, time, node) VALUES('" + temp + "','" + hu
+ "',now(),'" + role + "');";
            db.update(sql);
        }
    }
    / **
     * @ see HttpServlet # doPost (HttpServletRequest request, HttpServletResponse
response)
     * /
    protected void doPost (HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        doGet(request, response);
    }
    / **
    * method 将字符串类型的日期转换为一个 Date(java.sql.Date)
    * @param dateString 需要转换为 Date 的字符串
    * @return dataTime Date
    * /
    public final static java.sql.Date string2Date(String dateString)
    throws java.lang.Exception {
    DateFormat dateFormat;
    dateFormat = new SimpleDateFormat("yyyy - MM - dd", Locale.ENGLISH);
    dateFormat.setLenient(false);
    java.util.Date timeDate = dateFormat.parse(dateString);
                                                                    //util 类型
    java.sql.Date dateTime = new java.sql.Date(timeDate.getTime());
                                                                    //sql 类型
    return dateTime;
    }
    / * *
     * method 将字符串类型的日期转换为一个 timestamp(时间戳记 java. sql. Timestamp)
     * @param dateString 需要转换为 timestamp 的字符串
     * @return dataTime timestamp
     * /
    public final static java.sql.Timestamp string2Time(String dateString)
            throws java.text.ParseException {
            DateFormat dateFormat;
            dateFormat = new SimpleDateFormat("yyyy - MM - dd kk:mm:ss.SSS", Locale.
ENGLISH);
               //设定格式
```

移动端采用 Android 技术实现,用户通过移动端远程查看车位状态。移动端通过 网络请求,访问服务器。服务器端为移动端开发提供的接口代码如下。

QueryAllPSDData. java

```
package parkinglot;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import java.util.Map;
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 bean.PSD;
import utils.DBUtil;
/ * *
* Servlet implementation class QueryAllUsers
* /
@WebServlet("/QueryPSDData")
public class QueryPSDData extends HttpServlet {
    private static final long serialVersionUID = 1L;
    / **
      * @see HttpServlet # HttpServlet()
     * /
    public QueryPSDData() {
        super();
        // TODO Auto - generated constructor stub
    }
```

```
/ * *
       * @ see HttpServlet # doGet (HttpServletRequest request, HttpServletResponse
response)
     * /
     protected void doGet (HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        doPost(request, response);
    }
    / * *
      * @ see HttpServlet # doPost (HttpServletRequest request, HttpServletResponse
response)
     * /
     protected void doPost (HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        request.setCharacterEncoding("utf - 8");
        response.setContentType("text/html;charset = utf - 8");
        response.setCharacterEncoding("utf - 8");
        List < PSD > psdList = new ArrayList < PSD >();
        DBUtil db = new DBUtil();
        String sql;
        sql = "select * from psd;";
        List < Map < String, String >> list = db.getList(sql);
        Iterator < Map < String, String >> iter = list.iterator();
        if (list.isEmpty()){
            System.out.println("query nothing!");
        } else {
             while (iter.hasNext()) {
                 Map < String, String > tempmap = (Map < String, String >) iter.next();
                 PSD psd = new PSD();
                 psd.setNode(tempmap.get("node"));
                 psd.setStatus(tempmap.get("status"));
                 psdList.add(psd);
             }
         }
        request.setAttribute("psdList",psdList);
        request.getRequestDispatcher("psdlist.jsp").forward(request, response);
    }
}
```

# 5.2 物联网移动端

# 5.2.1 Android 平台

Android 平台是我们生活中接触较多的平台,许多手机厂商是基于 Android 定制的系统,很多开发商开发基于 Android 平台的 App,在物联网中使用 Android 开发嵌入式控制程序。可以说,Android 是人们工作和生活中不可缺少的平台。因此,本书所研究的项目对于 Android 平台专门开发了一款 App。图 5-46 是一个简化的 Android 软件层次结构。



图 5-46 Android 软件层次结构

### 5.2.2 Android Studio 环境的安装

IDE 是 Intelligent Development Environment 的简称,即智能开发环境。Android IDE 是为 Android 应用开发提供支持的开发软件,有关 Android 的项目和代码将在 Android IDE 中管理。Android IDE 是一个集成开发环境,常用的 Android IDE 有 Eclipse+ADT、ADT-Bundle 和 Android Studio。本书采用 Android Studio 作为移动 端程序的集成开发环境。下面介绍 Android Studio 的安装配置过程。

(1) 从 Android Studio 官方网址 http://www.android-studio.org/下载 Android Studio,界面如图 5-47 所示。

(2)下载好安装包之后,单击 Next 按钮进行安装,如图 5-48 所示。



图 5-47 Android Studio 官方网站



图 5-48 Android 安装开始界面

(3) 在选择组件对话框中,选择安装 Android Studio 虚拟机,如图 5-49 所示。

(4) 在如图 5-50 所示的界面中,选择 Android Studio 的安装路径。

(5) 在如图 5-51 所示的界面中,为 Android Studio 选择一个开始菜单文件夹,以 便创建编程中使用到的快捷键。

(6) 进入安装界面及安装完成界面如图 5-52~图 5-54 所示。

Android Studio Setup		- 0
4	Choose Components	
R.	Choose which features of Androi	d Studio you want to install.
Check the components you v install. Click Next to continue	vant to install and uncheck the con	nponents you don't want to
Select components to install:	<ul> <li>✓ Android Studio</li> <li>✓ Android Virtual Device</li> </ul>	Description Position your mouse over a component to see its description.
Space required: 2.3GB		

图 5-49 选择组件界面

ndroid Stud	lio Setup	-	
×	Configuration Settings Install Locations		
Android Studio	D Installation Location		
The location Click Browse	specified must have at least 500MB of free space. e to customize:		
C:\Program	n Files\Android\Android Studio	Bro	wse

图 5-50 选择 Android Studio 安装路径

Android Studio Setup			-		×
	Choose Start I	Menu Folder			
R.	Choose a Start	Menu folder for	the Android Stud	io shortcuts	u -
Select the Start Menu fold can also enter a name to c	er in which you wo reate a new folder	uld like to create	the program's sh	ortcuts. Yo	u
Android Studio					
7-Zip Accessories Administrative Tools Android Studio CCleaner Everything FortiClient Git IAR Systems Maintenance					~
Do not create shortcuts					
		< Back	Install	Can	:el

图 5-51 选择 Android Studio 开始菜单文件夹



图 5-52 Android Studio 安装进度

Android Studio Setup	Installation Complete Setup was completed successfully.	-		×
Completed Show details				
	< Back Ne	xt >	Can	e

图 5-53 Android Studio 安装完成界面 1



图 5-54 Android Studio 安装完成界面 2

(7) 在导入 Android Studio 设置选项中,选择 Do not import settings,如图 5-55 所示。

🛎 Import Android Studio Settings From	×
O Config or installation folder:	
Do not import settings	
	OK

图 5-55 暂不导入 Android Studio 设置

(8) 在如图 5-56 所示的界面中,提示无法找到 Android SDK,此时单击 Cancel 按钮。

And And	roid Studio First Run	×
×	Unable to access Android SDP Setup Proxy Cancel	K add-on list
		$\square$

图 5-56 取消 Android SDK 获取

(9) 进入 Android Studio 的安装设置欢迎界面,如图 5-57 所示。



图 5-57 Android Studio 设置欢迎界面



(10) 在安装类型中,选择标准安装,如图 5-58 所示。

图 5-58 选择安装类型

(11) 选择 Android Studio 的界面风格界面,可以选择自己熟悉的编程界面风格, 如图 5-59 所示。



图 5-59 选择 Android Studio 界面风格

SDK是 Software Development Kit 的简称,即软件开发工具包,一般是被软件工程师 用于为特定的软件包、软件框架、硬件平台、操作系统等建立应用软件的开发工具的 集合。

在 Android 中, Android SDK 为开发者提供了库文件以及其他开发所用到的工具。简单理解为 Android 开发工具包集合, 是整体开发中所用到的工具包。这里需要指定 SDK 的本地路径, 如果之前计算机中已经存在 SDK, 可以指定该路径, 后续就可以不用下载 SDK; 这里暂时可以指定一个后续将保存 SDK 的路径, 如图 5-60 所示。随后进入下载组件过程, 如图 5-61 和图 5-62 所示。

					-		
							_
tion settings,	lick Previou	s.					
52.6 MB							
7.87 MB							
2.63 MB							
	Demin		Blant	Cane	1	et	
	tion settings, o 52.6 MB 7.87 MB 2.63 MB	tion settings, click Previou 52.6 MB 2.63 MB	tion settings, click Previous. 52.6 MB 7.87 MB 2.63 MB	tion settings, click Previous. 52.6 MB 7.87 MB 2.63 MB	tion settings, click Previous. 52.6 MB 7.87 MB 2.63 MB	tion settings, click Previous. 52.6 M8 7.87 M8 2.63 M8	tion settings, click Previous. 52.6 MB 7.87 MB 2.63 MB

图 5-60 指定 SDK 路径



图 5-61 下载组件过程 1

ndroid Studio Setup Wizard		-		
Downloading Components				
Parsing C:\Users\TheJiaM\AppData\Local\Android\Sdk\build-tools\	28. 0. 3\package.xml			
Parsing C:\Users\TheJiaM\AppData\Local\Android\Sdk\build-tools\	29. 0. 2\package xml			
Parsing C:\Users\TheJiaN\AppData\Local\Android\Sdk\build-tools\	29. 0. 3\package.xml			
Parsing C:\Users\TheJiaN\AppData\Local\Android\Sdk\emulator\pac	kage. xml			
Parsing C:\Users\TheJiaN\AppData\Local\Android\Sdk\extras\intel	\Hardware_Accelerated_Execution_Manag	er \packag	ge. xml	
Parsing C:\Users\TheJiaN\AppData\Local\Android\Sdk\patcher\v4\p	ackage. xml			
Parsing C:\Users\TheJiaN\AppData\Local\Android\Sdk\platform-too	ls\package.xml			
Parsing C:\Users\TheJiaN\AppData\Local\Android\Sdk\platforms\an	droid-25\package.xml			
Parsing C:\Users\TheJiaN\AppData\Local\Android\Sdk\platforms\an	droid-28\package.xml			
Parsing C:\Users\TheJiaN\AppData\Local\Android\Sdk\platforms\an	droid-29\package.xml		- 1	1
Parsing C:\Users\TheJiaN\AppData\Local\Android\Sdk\sources\andr	oid-29\package.xml			
Parsing C:\Users\TheJiaH\AppData\Local\Android\Sdk\system-image	s\android-29\google_apis_playstore\x8	6\package	e. xml	
Parsing C:\Users\TheJiaN\AppData\Local\Android\Sdk\tools\packag	e. xml			
Android SDE is up to date.				
Running Intel® HAXM installer				
Intel HAXM updated successfully!				
	Previo Next	I lave	Fini	1
	PICTION IVEAL CO	10.01	2000	- 1

图 5-62 下载组件过程 2

组件下载完毕后, Android Studio 即安装成功, 如图 5-63 所示。

- Welcome to Android Studie	2	-		×
	2			
	Android Studio			
	+ Start a new Android Studio project			
	🖶 Open an existing Android Studio project			
	Check out project from Version Control +			
	Profile or debug APK			
	ピ Import project (Gradle, Eclipse ADT, etc.)			
	🛒 Import an Android code sample			
		🗘 Configure 🕶	Get He	rlp ≠

图 5-63 Android Studio 安装成功

单击图 5-63 中的 Start a new Android Studio project 新建一个工程,进入如图 5-64 所示界面,选择 Empty Activity 作为一个新的界面。

在如图 5-65 所示的界面中,新建一个新的工程,并加以配置。建立新工程后会出 现工程编辑界面,如图 5-66 所示。

	÷ 1	÷	¢ 1
Add No Activity			
	Basic Activity	Empty Activity	Bottom Navigation Activity
¢			
Fragment + ViewModel	Fullscreen Activity	Master/Detail Flow	Navigation Drawer Activity

图 5-64 新建一个 Activity

🛲 Create New Project		×
Configure your project		
	Name	
	FirstApp	
	Package name	
÷	com.example.firstapp	
	Save location	
	E:\project\FirstApp 🛸	
	Language	
	Java 👻	
	Minimum API level API 15: Android 4.0.3 (IceCreamSandwich)	
Empty Activity	• Your app will run on approximately 100% of devices.	
	Help me choose This project will support instant apps	
	Use androidx." artifacts	
Creates a new empty activity		
	Previous Next Cancel	Finish
	Erevious Hext Gancer	Funsu

图 5-65 新建一个新的 Android 工程



图 5-66 工程编辑界面

工程建立完成后,系统开始构建 gradle,如图 5-67 所示。



图 5-67 构建 gradle

gradle 构建成功之后,出现如图 5-68 所示的界面,代表一个 Android Studio 工程 已经建立成功。即可在编码区域开始工程的编码实现,如图 5-69 所示。

A Build Sync	¢ -
	38 m 22 4 608 m 3 8 1 355 m 3 2 1 395 m 3 3 4 231 m 3 2 1 21 m 4 3 4 231 m 4 1 1 666 m 1
IE TODO III Terminal 🔥 Build 🗈 § Logcat	Event Log
* daemon started successfully (2 minutes ago)	15:1 CRLF I UTF-8 I 4 spaces I 🚡 💆



CS Wind	lvity.java [app] - dow Help	Android Studio											-			2	ĸ
firstapp	)  MainActivit	у	4	≡ app ▼	No devices 🔻	► d	. E.	ŏ	G. 4	の前	 85	Þ	R.	<u>n</u>	0,	Q	F
activi	ity_main.xml $\times$	C MainActivity.java	×														
1 2 3 6 7 8 9 9 10 11 12 13 14 15	<pre>package com er import public class 1 @Override protected super </pre>	ample. firstapp: fainActivity extends void onCreate (Bundle onCreate (savedInstan atentView (E. layout. ac	AppCompatAc savedInst. ceState): tivity_main	itivity { unceState) { 0;												,	

图 5-69 Android Studio 工程代码编辑区域

Android 工程编码完成后,可通过单击 Build→Build Bundle(s)/APK(S)选项,生成 Android APK 文件,用于在 Android 模拟器或 Android 系统的真机上运行测试,如图 5-70 所示。

# 5.2.3 Android 网络通信

OkHttp 框架是一个处理网络请求的开源项目,是安卓端的轻量级框架,由 Square 公司开发,用于替代 HttpUrlConnection 和 Apache HttpClient。OkHttp 框架可以支



图 5-70 生成 Android APK

持 Android 2.3 及以上版本,需要 JDK 1.7 及以上版本。本书介绍利用 OkHttp 框架 实现安卓端与服务器网络通信的方法。

OkHttp 框架优点如下。

- (1) 允许连接到同一个主机地址的所有请求,提高请求效率。
- (2) 共享 Socket,减少对服务器的请求次数。
- (3) 通过连接池,减少了请求延迟。
- (4) 通过缓存响应数据来减少重复的网络请求。
- (5) 减少了对数据流量的消耗。
- (6) 自动处理 GZip 压缩。

使用 OkHttp 框架之前,需要在工程中引入 OkHttp 包,OkHttp 框架还依赖另一个 okio 包,同样需要引入,代码如下所示,也可以在相应的 model 中的 build.gradle 配置文件中填入,然后将工程同步,环境会自动下载需要的包。

```
compile 'com.squareup.okhttp3:okhttp:3.2.0'
compile 'com.squareup.okio:okio:1.7.0'
//可以修改版本号
```

需要在 Android 工程中设置网络权限:

<user - permission andriod:name = "andriod.premission.INTERNET"/> <!-- 用户连接网络权限-->

OkHttp框架的使用涉及 OkHttpClient、RequestBody、Request、Call、Response 等基本类。

(1) OkHttpClient: 对于该类创建对象实例化的方式有默认的标准形式和自定义 形式两种。

• 标准形式采用如下代码进行实例化。

OkHttpClient mOkHttpClient = new OkHttpClient();

以自定义形式实例化 OkHttpClient 对象时,可以设置网络连接的超时时长、读取超时时长和写入超时时长,可以调用 build()方法进行实例化,代码如下。

<pre>OkHttpClient mOkHttpClient = new OkHttpClient.Builder()</pre>	
.connectTimeout(10,TimeUtil.SECONDS)	//为新连接设置默认连接超时 时长,第一个参数是时长,第 二个参数是单位
.readTimeout(10,TimeUtil.SECONDS)	//设置新连接的默认读取超时 时长
.writeTimeout(10,TimeUtil.SECONDS)	//设置新连接的默认写入超时 时长
.cache(setCache)	//设置用于读取和写入缓存响 应的响应缓存[1]
.build();	

在如上代码的[1]处,需要一个参数为 Cache 的对象,如下代码定义了 Cache 对象,大小是 10×1024×1024,在访问 File 对象指定的路径 filePath 时,可以扩展缓存区 域大小。

```
//[1]:参数为 Cache 对象
File filePath = new File(getExternalCacheDir(),"netCache");
int cacheSize = 10 * 1024 * 1024;
Cache setCache = new Cache(filePath,cacheSize);
```

在 OkHttp2.x 版本中,设置以上超时时长的代码有所不同,如下所示。

```
mOkHttpClient.setConnectTimeout(10,TimeUtil.SECONDS);
mOkHttpCLient.setConnectTimeout(10,TimeUtil.SECONDS);
mOkHttpClient.setConnectTimeout(10,TimeUtil.SECONDS);
mOkHttpClient.setCache(setCache);
```

(2) RequestBody: 该类是用于封装 OkHttp 框架进行网络通信时的请求体,适用 于 post 请求方式,用于上传数据到服务器。其核心方法有如下 4 个。

public abstract MediaType contentType()
public long MediaType contentLength()
public abstract void writeTo(BufferedSink sink)
public static RequestBody create(MediaType contentType,String content) [2]

其中,MediaType可以是多种类型,该类的对象实例化方式如下。

//[2]:第二个参数可以是多种类型;该类存在 create 的多个重载方法 //该类的对象实例化方式: MediaType mMediaType = MediaType.parse("application/octet - stream"); //http://tool.oschina.net/commons File putFile = new File(Environment.getExternaStorageDirectory(),"文件名.扩展名"); //父路径和子路径两个参数拼接起来为文件地址绝对地址 RequestBody mRequestBody = RequestBody.create(mMediaType,putFile);

RequestBody 的 create()方法需要两个参数:指定要上传文件的类型和文件对象本身。如果用 create()方法上传键值对类型的数据,可使用 FormBody,代码如下。

(3) Request: 该类用于生成网络连接请求对象,包括请求参数、请求头、请求方式 等多种信息,常用方法有 url()、post()、method()、headers()等。

```
Request mRequest = new Request.Builder()
    .url("http://www.baidu.com")
    .post(mRequestBody) [4]
    .build();
```

请求体可以通过 post()方法提交,如果不明确调用 post()方法,则可使用 get()方法请求网络。使用 post()方法上传数据比 get()方法更安全,数据大小不受限制,多个参数可以封装成请求体上传,对于长表单数据、上传文件,首选 post()方法请求网络。

(4) Call: 该类是网络请求执行的最后一个需要实例的对象,该类提供了网络请求的同步与异步连接方式,网络连接属于耗时操作,因此不能编写在 UI 线程中。如果在网络请求后,需要更新 UI 布局就需要调用 Handler,实现 Handler 的 runOnUiThread()方法如下。

```
Call mCall = mOkHttpClient.newCall(mRequest);
//使用异步网络连接
mCall.enqueue(new CallBack(){
    @override
    public void onFailure(Call call, IOException e){
        //网络连接失败
    }
    @override
    public void onResponse(Call call, Response response){
        if(response.isSuccessful()){
            //请求数据成功,返回 code 为 200~300 [5]
        }else{
            //请求数据失败
        }
    }
});
```

其中,请求数据成功,服务器将返回响应状态码。响应状态码是在程序中经常需要判断的一个值,响应状态码大致分为:

- 1××: 信息,表示请求收到,继续处理;
- 2××: 成功,表示请求成功;
- 3××:重定向,为完成请求客户需进一步细化请求;
- 4××:由客户端引发的错误;
- 5××:由服务器引发的错误。

因此,2××(以2开头的状态码)表示请求成功。判断请求是否成功,及判断是否 得到了请求成功的响应状态码,response.isSuccessful()的源码如下。

```
public boolean isSuccessful(){
    return code > = 200 && code < 300;
}</pre>
```

使用同步方式请求网络,在 Android 编程中被归类为耗时操作,耗时操作不允许 在 UI 主线程中运行,因此需要开启新的子线程,代码如下。

```
new Thread(new Runnable(){
    Call mCall = mOkHttpClient.newCall(mRequest);
    Response mResponse = mCall.execute(); //涉及的 Response 对象知识点
    if(mResponse.isSuccessful()){
        //网络同步请求成功,更新 UI 布局需要使用 runOnUiThread()方法
    }else{
    }
}
```

```
}).start(); //创建一个子线程后需要使用 start()方法启动该线程
注:runOnUiThread(new Runnable(){
    @override
    public void run(){
        //更新 UI 的逻辑代码
    }
});
//同步方法会阻塞当前线程的执行,异步方法不会阻塞当前线程的执行
```

(5) Response: 该类是网络请求后的响应信息对象,对于服务器返回的数据均存 放在该示例对象中,而且对于 Response 实例一次请求中只能有一次有效调用,如果调 用两次将会出现程序错误,这就使得在需要多次使用数据前要将 Response 实例中的 数据保存下来, Response 类提供了多种方法,如 body、code、protocol、request、 isSuccessful、headers(响应头对象)、toString 等。

```
Response mResponse = mCall.execute();
if(mResponse.isSuccessful()){
   String str = mResponse.body().string(); [6]
   int code = mResponse.code();
   Protocol protocol = mResponse.protocol();
   Request request = mResponse.request();
   String head = mResponse.toString();
   Log.i("Response 响应信息集:","" + str + code + protocol + request + head);
}
```

基于以上五个类,就可以写出基本的 get()、post()请求,进行同步或异步的网络 连接了。对于 OkHttpClient 表示 HTTP 请求的客户端类,大多数情况下推荐只使用 创建一个该对象的实例,全局使用。

下面是一个完整的关于 post()请求的异步方法示例。

```
public class PostAsyn(){
    private TextView tvShowNetInfo;
    public static 0kHttpClient okHttpClient;
    static(
        okHttpClient = new 0kHttpClient.Builder()
            .connectTimeout(15,TimeUnit.SECONDS)
            .readTimeout(15,TimeUnit.SECONDS)
            .writeTimeout(15,TimeUnit.SECOND)
            .build();
    );
    @ Overvide
    public void onCreate(Bundle saveInstanceState){
```

```
super.onCreate(saveInstanceState);
        setContentView(R.layout.activity_main);
        tvShowNetInfo = (TextView) findViewId(R.id.tv show net info);
        postAsynDate();
    }
    public void postAsynData(){
        RequestBody requestBody = new FormBody.Builder()
                        .add("key1","value1")
                        .add("key2","value2")
                        .build();
        Request request = new Request.Builder()
                    .url("http://www.baidu.com")
                    .post(requestBody)
                    .build();
        Call call = okHttpClient.newCall(request);
        call.enqueue(new Callback(){
                @Overvide
                public void onFailure(Call call, IOException e) {
                    goUiThread("网络连接失败");
                }
                @Overvide
                public void onResponse(Call call, Response response) {
                    if(response.isSuccessful()){
                        goUiThread("请求数据成功,响应码:" + response.code());
                    }else{
                         goUiThread("网络连接成功,但是没有响应数据,响应码:" +
response.code());
                    }
                }
            }
        );
    }
    private void goUiThread(final String str){
        runOnUiThread(new Runnable(){
            @Overvide
            public void run(){
                tvShowNetInfo.setText(str);
        });
    }
}
//对应的布局文件略
```

# 5.3 习题

1.	下列(	)选项不是 JSP ;	运行所必需的软件	牛玎	不境。
	A. 操作3	系统		В.	JavaJDK
	C. 支持 ]	JSP 的 Web 服务者	22	D.	数据库
2.	在 JDBC	中,用来描述结果	集的接口是(	)	0
	A. States	ment		В.	Connection
	C. Resul	tSet		D.	DriverManager
3.	在 JDBC	API中,下列(	)接口或类可	以	用来保存从数据库返回的查询
结果。					
	A. Resul	tSet		В.	Connection
	C. Stater	ment		D.	DriverManager
4.	在 JDBC	API中,下列(	)接口或类可以	用	来执行 SQL 语句。
	A. Resul	tSet		В.	Connection
	C. Stater	ment		D.	DriverManager
5.	下述选项	中不属于 JDBC 基	基本功能的是(		)。
	A. 与数打	居库建立连接		В.	提交 SQL 语句
	C. 处理3	查询结果		D.	数据库维护管理
6.	假设已创	建了语句对象名	为 sta,下列语句中	户辑	皆误的是( )。
	A. sta.e:	xecuteUpdate("de	elete from food v	vh	ere price $< 0"$ );
	B. sta.ex	kecuteQuery("del	ete from food wh	ner	The price $< 0"$ );
	C. sta.ez	xecute("delete fro	om food where pr	rice	e < 0");
	D. sta.e:	xecuteQuery("sel	ect * from food	W	here price $< > 0"$ );
7.	创建 JDB	C的数据库连接X	时象,下列语句中	Εł	确的是( )。
	A. Conn	ection conn = D	riverManager. ge	etC	Connection("jdbc:mysql://127.
	0.0.1	l:3306/mealsyste	em", "root", "ro	ot	");
	B. Conne	ection conn $=$ C	lass. forName("	'jd	bc: mysql://127. 0. 0. 1: 3306/
	meals	system", "root",	"root");		
	C. Conne	ection conn = D	river. getConnec	tio	n("jdbc:mysql://127.0.0.1:
	3306/	/mealsystem", "1	<pre>coot", "root");</pre>		
	D. Conn	ection conn = I	DriverManager. g	et(	Connection ( " com. mysql. jdbc.
	Drive	er", "root", "roo	t");		
8.	下列关于	JDBC 说法中错记	吴的是()。		
	A. JDBC	使得编程人员从	复杂的驱动器调	用	命令和函数中解脱出来,可以致
	力于原	应用程序中的关键	<b>地</b> 方		

B. JDBC 支持非关系数据库,如 NoSQL 等 C. 用户可以使用 IDBC-ODBC 桥驱动器将 IDBC 函数调用转换为 ODBC D. JDBC API 是面向对象的,可以让用户把常用的方法封装为一个类以备后用 9. 请阐述 MVC 设计模式的含义,并说明 M、V、C 各自代表什么层次。 10. 下列关于 MVC 的说法中错误的是( )。 A. 在 MVC 模式中,如果哪一层的需求发生了变化,只需要更改相应层的代 码而不会影响其他层中的代码 B. 在 MVC 模式中,所有的核心业务逻辑都应该放在控制层实现 C. 在 MVC 模式中,由于按层把系统分开,那么就能更好地实现开发中的分工 D. 使用 MVC 模式,有利于组件的重用 11. Android 应用程序需要打包成( )文件格式在手机上安装运行。 A. class B. . xml C. . apk D. dex 12. 在 Activity 的生命周期中,当 Activity 被某个 AlertDialog 覆盖一部分后,会 处于( )状态。 A. 暂停 B. 活动 C. 停止 D. 销毁 13. Android 项目启动时最先加载的是 AndroidManifest. xml 文件,如果有多个 Activity,以下()属性决定了该 Activity 最先被加载。 A. android. intent. action. LAUNCH B. android: intent. action. ACTIVITY C. android: intent. action. MAIN D. android: intent. action. VIEW 14. 下列关于 Handler 的说法中不正确的是( ) 。 A. 它是实现不同进程间通信的一种机制 B. 它采用队列的方式来存储 Message C. Handler 既是消息的发送者也是消息的处理者 D. 它是实现不同线程间通信的一种机制 15. 下列( )不是 Android 的存储方式。 A. File B. SharedPreferences C. SQLite D. ContentProvider 16. Android 支持的 4 大重要组件,分别是 Activity、 、 、Service 和 Content Provider.

17. Android 的事件处理机制有两种:一种是基于回调的处理机制;另一种 。