

# 《Java EE 7精粹》

## 图书基本信息

书名：《Java EE 7精粹》

13位ISBN编号：9787115375488

出版时间：2015-2

作者：Arun Gupta

页数：314

译者：韩陆

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：[www.tushu111.com](http://www.tushu111.com)

## 作者简介

Arun Gupta是甲骨文的Java布道者。作为Java EE团队的创始成员之一，他的工作是创建和培育Java EE、GlassFish和WebLogic社区。他带领一个跨职能的团队通过战略、规划和内容、营销活动和方案的执行推动Java EE 7平台的全球启动。他非常热衷于开发和接洽合作伙伴、客户、JUG、Java champions以及其他传播于世界各地的Java的精华。Arun拥有丰富的演讲经验，他在30多个国家演讲过无数的话题。作为一名作者，Arun有一个多产的博客：[blogs.oracle.com/arungupta](http://blogs.oracle.com/arungupta)，同时他是一位跑步爱好者和环球旅行者，请在twitter关注他：[@arungupta](https://twitter.com/arungupta)，他很平易近人。

## 译者简介

韩陆，北京航空航天大学软件工程硕士。热爱编程，热衷于开源社区的技术交流和分享，将持续学习作为一种生活习惯。曾混迹于用友总部、新浪、Avaya和Technicolor从事研发工作。现就职于阿里巴巴总部，从事敏捷和自动化测试的研发工作。《JSF和Richfaces使用指南》、《Java Restful Web Services使用指南》作者。

## 书籍目录

### 目录

#### 第1章 Java EE

1

##### 1.1 简介

1

###### 1.1.1 拥抱HTML5

2

###### 1.1.2 更高的生产力

2

###### 1.1.3 企业级需求

3

##### 1.2 可交付成果

3

###### 1.2.1 规范 ( Specification )

3

###### 1.2.2 参考实现 ( RI )

3

###### 1.2.3 技术兼容包 ( TCK )

3

###### 1.2.4 Web技术

3

###### 1.2.5 企业技术

4

###### 1.2.6 平台1.1

4

###### 1.2.7 Web服务技术

5

###### 1.2.8 管理和安全技术

5

##### 1.3 Java EE 7有什么新功能

7

###### 1.3.1 新的规范

7

###### 1.3.2 更新的规范

8

#### 第2章 Servlets

12

##### 2.1 WebServlet

12

##### 2.2 Servlet过滤器

17

##### 2.3 事件监听器

18

##### 2.4 异步支持

21

##### 2.5 非阻塞I/O

23

2.6 Web Fragment	24
2.7 安全	26
2.8 资源打包	29
2.9 错误映射	30
2.10 Handling Multipart Requests	31
2.11 Upgrade Processing	32
第3章 JSF	34
3.1 Facelets	35
3.2 资源处理	38
3.3 复合组件	39
3.4 请求处理生命周期阶段	42
3.4.1 恢复视图	42
3.4.2 接受请求值	43
3.4.3 处理验证	43
3.4.4 更新模型值	43
3.4.5 调用应用	43
3.4.6 渲染响应	43
3.5 Ajax	44
3.6 HTTP GET	47
3.7 服务器和客户端扩展点	48
3.7.1 转换器	48
3.7.2 校验器	49
3.7.3 监听器	50
3.7.4 行为	50
3.8 验证数据	

51	
3.9 导航规则	
52	
3.10 Faces Flow	
52	
3.11 资源库契约	
58	
3.12 Passthrough属性和HTML5友好的标记	
60	
3.13 组件标签	
61	
第4章 REST式的Web服务	
74	
4.1 资源	
74	
4.2 绑定HTTP方法	
78	
4.3 多种资源表述	
80	
4.4 绑定请求到资源	
81	
4.5 实体Provider	
83	
4.6 客户端API	
85	
4.7 映射异常	
88	
4.8 过滤器和实体拦截器	
89	
4.8.1 过滤器	
89	
4.8.2 实体拦截器	
93	
4.9 资源验证	
94	
第5章 基于SOAP的Web服务	
97	
5.1 Web服务端点	
98	
5.2 基于Provider的动态端点	
101	
5.3 基于端点的端点	
102	
5.4 Web服务客户端	
103	
5.5 基于Dispatch的动态客户端	
105	
5.6 Handler	
106	

## 第6章 JSON-P

110

### 6.1 流式API

111

#### 6.1.1 使用流式API消费JSON

111

#### 6.1.2 使用流式API生产JSON

113

### 6.2 对象模型API

115

#### 6.2.1 使用对象模型API消费JSON

115

#### 6.2.2 使用对象模型API生产JSON

116

## 第7章 WebSocket

119

### 7.1 注解式服务器端点

120

### 7.2 编程式服务器端点

125

### 7.3 注释式客户端端点

130

### 7.4 编程式客户端端点

132

### 7.5 JavaScriptWebSocket客户端

134

### 7.6 编码器和解码器

136

### 7.7 与Java EE安全集成

139

## 第8章 EJB

141

### 8.1 有状态会话Bean

141

### 8.2 无状态会话Bean

144

### 8.3 单例会话Bean

146

### 8.4 生命周期事件回调

147

### 8.5 消息驱动Bean

150

### 8.6 可移植的全局JNDI名称

152

### 8.7 事务

153

### 8.8 异步调用

154

### 8.9 定时器

155
8.10 嵌入式API
159
8.11 EJB精简版
160
第9章 上下文和依赖注入
162
9.1 探测Bean
162
9.2 注入点
165
9.3 Qualifier和Alternative
166
9.4 生产者和处置者
168
9.5 拦截器
169
9.6 装饰器
173
9.7 作用域和上下文
174
9.8 构造型
175
9.9 事件
177
9.10 可移植的扩展
178
9.11 内置Bean
180
9.12 生命周期回调
181
第10章 并发工具包
182
10.1 异步任务
182
10.2 计划任务
187
10.3 托管线程
190
10.4 动态上下文对象
191
第11章 Bean验证
195
11.1 内置的约束
195
11.2 自定义的约束
199
11.3 验证组
202

11.4 方法和构造子约束	203
第12章 Java事务	206
12.1 用户管理的事务	206
12.2 容器管理事务	207
12.3 @TransactionScoped	209
第13章 Java持久化	210
13.1 实体	210
13.2 持久化单元、持久化上下文和实体管理器	213
13.3 生成Schema	217
13.4 创建、读取、更新和删除实体	220
13.5 实体监听器	223
13.6 存储过程	226
13.7 验证实体	227
13.8 事务和锁	230
13.9 缓存	232
第14章 消息服务	234
14.1 发送消息	237
14.2 同步接收消息	241
14.3 异步接收消息	243
14.4 服务的质量	244
14.5 临时Destination	245
第15章 批处理	246
15.1 面向块的处理	247
15.1.1 自定义检查点	252
15.1.2 异常处理	



253	
15.2 Batchlet处理	
253	
15.3 监听器	
254	
15.4 作业顺序	
256	
15.4.1 Flow	
257	
15.4.2 Split	
257	
15.4.3 Decision	
258	
15.5 作业分区	
259	
第16章 构建端到端的应用	
263	
16.1 介绍	
263	
16.2 问题陈述	
264	
16.3 示例走查	
266	
16.4 显示订票 (JSF)	
270	
16.5 聊天室 (WebSocket的Java API)	
277	
16.6 查看和删除影片 (JAX-RS)	
283	
16.7 添加电影 (JSON-P)	
288	
16.8 售票 (Java平台的批处理应用)	
294	
16.9 电影点 (Java消息服务2)	
301	
16.10 结论	
306	
16.11 故障排除	
308	
16.12 完整的解决方案	
308	
附录A 补充书目	
309	

## 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:[www.tushu111.com](http://www.tushu111.com)