

Об авторе

Дэвид Хеффельфингер (David R. Heffelfinger) является техническим директором Ensode Technology LLC – консалтинговой компании, специализирующейся на разработке программного обеспечения, расположенной в районе большого Вашингтона, округ Колумбия. Дэвид – профессиональный архитектор, проектировщик и разработчик программного обеспечения с 1995 года. Он использует Java в качестве основного языка программирования с 1996 года. Ему довелось работать во многих крупномасштабных проектах для ряда клиентов, в числе которых были департамент США по Национальной безопасности, Freddie Mac, Fannie Mae и Министерство обороны США. Дэвид имеет степень магистра в области разработки программного обеспечения Южного методического университета. Также Дэвид является главным редактором Ensode.net (<http://www.ensode.net>), веб-сайта, посвященного Java, Linux и другим технологиям.

Я хотел бы поблагодарить всех, кто помогал мне в создании этой книги. Выражаю благодарность редакторам Мехулу Шетти (Mehul Shetty) и Дхираджу Чандирамани (Dhiraj Chandiramani), а также координаторам проекта Шубханджану Чаттерджи (Shubhanjan Chatterjee) и Паллаби Чаттерджи (Pallabi Chatterjee).

Отдельное спасибо техническим рецензентам Аллану Бонду (Allan Bond) и Аруну Гупте (Arun Gupta) за их важные замечания и предложения.

Кроме того, хочу поблагодарить группу GlassFish в Oracle (ранее Sun Microsystems) за разработку такого замечательного сервера приложений с открытым исходным кодом.

И, наконец, я признателен моей жене и дочери за их терпеливое отношение к моей дополнительной работе, в силу которой я был вынужден уделять меньше времени семье.

О рецензентах

Аллан Бонд (Allan Bond) – разработчик программного обеспечения, более 10 лет работающий в сфере ИТ. Его основное внимание сосредоточено на разработке систем с использованием Java и связанных с ней технологий. В процессе работы Аллану довелось консультировать множество организаций, от малых предприятий до компаний, входящих в список Fortune 500, а также государственных учреждений. Он имеет степень магистра в области управления информационными системами Университета Бригама Янга.

Я хотел бы поблагодарить свою жену и детей за их терпение в ночные часы (а иногда и выходные дни), когда мне нужно было завершить рецензирование этой книги.

Арун Гупта (Arun Gupta) – апологет Java EE и GlassFish, работающий в компании Oracle. Арун имеет более чем 14-летний опыт в индустрии программного обеспечения и работает с платформой JavaTM и несколькими связанными с ней интернет-технологиями. В настоящий момент его деятельность направлена на создание и укрепление сообщества пользователей Java EE 6 и GlassFish. Арун сотрудничал с несколькими организациями по стандартизации и осуществлял взаимодействие с представителями других компаний. Он входил в команду Java EE начиная с момента ее создания и в той или иной степени внес свой вклад во все релизы Java EE. Имеет обширный опыт обсуждения множества тем в международном сообществе и обожает широкомасштабные проекты.

Арун является активным блогером (<https://blogs.oracle.com/arungupta/>). В этом блоге более 1000 записей, и его часто читают посетители со всего мира; посещаемость блога достигает 25 000 посещений в день.

Содержание

Об авторе	5
О рецензентах	6
Содержание	7
Предисловие	16
Темы, освещаемые в книге	16
Что нужно для чтения этой книги	19
Для кого эта книга	19
Соглашения	19
Поддержка клиентов	20
Сообщения об ошибках	20
Незаконное воспроизведение содержимого книги	21
Вопросы	21
1. Знакомство с сервером GlassFish	22
Общий обзор Java EE и GlassFish	22
Новые возможности Java EE	23
JavaServer Faces (JSF) 2.0	23
Enterprise JavaBeans (EJB) 3.1	23
API Персистентности Java (JPA) 2.0	24
Контексты и инъекция зависимости для Java (Web Beans 1.0)	24
API Java Сервлета 3.0	24
API Java для веб-сервисов RESTful (JAX-RS) 1.1	25
API Java для веб-сервисов XML (JAX-WS) 2.2	25
Архитектура Java для связывания с XML (JAXB) 2.2	25
Новые возможности GlassFish v.3	25
Преимущества GlassFish	26

Получение GlassFish	27
Установка GlassFish	28
Зависимости GlassFish	28
Выполнение установки	29
Проверка установки	34
Развертывание нашего первого приложения Java EE	35
Развертывание приложения через веб-консоль	35
Отмена развертывания приложения через веб-консоль	37
Развертывание приложения с помощью командной строки	38
Каталог autodeploy	39
Утилита командной строки asadmin	40
Домены GlassFish	41
Создание доменов	41
Удаление доменов	43
Остановка домена	43
Настройка подключения к базе данных	43
Создание пулов соединений	44
Создание источников данных	48
Заключительные замечания	49
Резюме	49
2. Разработка и развертывание сервлета	50
Что такое сервлет?	50
Написание нашего первого сервлета	51
Компиляция сервлета	52
Конфигурирование сервлета	53
Упаковка веб-приложения	55
Развертывание веб-приложения	56
Тестирование веб-приложения	56
Обработка HTML-форм	58
Переадресация запросов и перенаправление откликов	64
Переадресация запроса	64
Перенаправление отклика	67

Сохранение данных приложения между запросами	69
Новые возможности, появившиеся в Сервлете 3.0	71
Необязательный дескриптор развертывания web.xml	71
Аннотация @WebServlet	71
Передача сервлету параметров инициализации через аннотации ..	73
Аннотация @WebFilter	74
Аннотация @WebListener	76
Подключаемость	78
Программное конфигурирование веб-приложений	80
Асинхронная обработка	82
Резюме	84
3. JavaServer Pages	85
Введение в JavaServer Pages	85
Разработка нашей первой JSP-страницы	86
Неявные объекты JSP	90
JSP и JavaBeans	97
Повторное использование JSP-контента	101
Пользовательские теги JSP	103
Расширение класса SimpleTagSupport	104
Использование файлов тегов для создания пользовательских тегов JSP	109
Унифицированный язык выражений	113
XML-синтаксис JSP	116
Резюме	118
4. Библиотека стандартных тегов JSP	119
JSTL-библиотека базовых тегов	119
JSTL-библиотека тегов форматирования	127
JSTL-библиотека SQL-тегов	131
JSTL-библиотека XML-тегов	136
Функции JSTL	140
Резюме	143

5. Подключение к базе данных	144
База данных CUSTOMERDB	144
JDBC	145
Извлечение данных из базы данных	146
Изменение информации в базе данных	152
API Персистентности Java	154
Отношения сущности	159
Отношения «один к одному»	159
Отношения «один ко многим»	164
Отношения «многие ко многим»	168
Составные первичные ключи	173
Язык запросов персистентности Java	177
Новые функции, введенные в JPA 2.0	181
API Критериев	181
Поддержка проверки допустимости со стороны бинов	184
Резюме	186
6. Java Server Faces	187
Введение в JSF 2.0	187
Фэйслеты	187
Необязательный файл faces-config.xml	188
Стандартное расположение ресурсов	188
Разработка нашего первого JSF 2.0-приложения	189
Фэйслеты	189
Этапы проекта	193
Проверка допустимости	195
Группировка компонентов	197
Отправка формы	197
Управляемые бины	198
Контексты управляемых бинов	199
Навигация	200
Пользовательская проверка допустимости данных	201
Создание нестандартных элементов верификации	202
Методы блока проверки допустимости	204

Настройка сообщений JSF по умолчанию	207
Настройка стилей сообщения	207
Настройка текста сообщения	209
Интеграция JSF и JPA	211
Включение Ajax в приложения JSF 2.0	218
Стандартные компоненты JSF	222
Базовые компоненты JSF	222
Тег <f:actionListener>	222
Тег <f:ajax>	222
Тег <f:attribute>	223
Тег <f:convertDateTime>	223
Тег <f:convertNumber>	223
Тег <f:converter>	224
Тег <f:event>	224
Тег <f:facet>	225
Тег <f:loadBundle>	225
Тег <f:metadata>	225
Тег <f:param>	226
Тег <f:phaseListener>	226
Тег <f:selectItem>	226
Тег <f:selectItems>	226
Тег <f:setPropertyActionListener>	227
Тег <f:subview>	227
Тег <f:validateBean>	227
Тег <f:validateDoubleRange>	228
Тег <f:validateLength>	228
Тег <f:validateLongRange>	228
Тег <f:validateRegex>	229
Тег <f:validateRequired>	229
Тег <f:validator>	229
Тег <f:valueChangeListener>	229
Тег <f:verbatim>	230
Тег <f:view>	230
Тег <f:viewParam>	230

HTML-компоненты JSF	231
Ter <h:body>	231
Ter <h:button>	231
Ter <h:column>	231
Ter <h:commandButton>	231
Ter <h:commandLink>	232
Ter <h:dataTable>	232
Ter <h:form>	232
Ter <h:graphicImage>	233
Ter <h:head>	233
Ter <h:inputHidden>	233
Ter <h:inputSecret>	233
Ter <h:inputText>	234
Ter <h:inputTextarea>	234
Ter <h:link>	234
Ter <h:message>	234
Ter <h:messages>	235
Ter <h:outputFormat>	235
Ter <h:outputLabel>	235
Ter <h:outputLink>	236
Ter <h:outputScript>	236
Ter <h:outputStylesheet>	236
Ter <h:outputText>	236
Ter <h:panelGrid>	236
Ter <h:panelGroup>	237
Ter <h:selectBooleanCheckbox>	238
Ter <h:selectManyCheckbox>	238
Ter <h:selectManyListbox>	238
Ter <h:selectManyMenu>	239
Ter <h:selectOneListbox>	239
Ter <h:selectOneMenu>	239
Ter <h:selectOneRadio>	239
Дополнительные библиотеки компонентов JSF	240
Резюме	240

7. Служба обмена сообщениями Java	241
Настройка GlassFish для использования JMS	241
Создание фабрики JMS-соединений	241
Создание очереди JMS-сообщений	243
Создание темы JMS-сообщений	244
Очереди сообщений	245
Отправка сообщений в очередь сообщений	245
Извлечение сообщений из очереди сообщений	249
Асинхронный прием сообщений из очереди сообщений	250
Просмотр очередей сообщений	253
Темы сообщений	254
Отправка сообщений теме сообщений	254
Получение сообщений от темы сообщений	255
Создание долговременных подписчиков	257
Резюме	260
8. Безопасность	261
Области безопасности	261
Предопределенные области безопасности	262
Область администратора	262
Область файла	264
Стандартная аутентификация через область файла	265
Область сертификата	276
Создание самоподписанных сертификатов	276
Конфигурирование приложений для использования области сертификата	280
Определение дополнительных областей	283
Определение дополнительных областей файла	283
Определение дополнительных областей сертификата	285
Определение области LDAP	286
Определение области Solaris	287
Определение области JDBC	288
Определение пользовательских областей	293
Резюме	298

9. Enterprise JavaBeans	299
Сеансовые бины	300
Простой сеансовый бин	300
Более реалистический пример	303
Вызов сеансовых бинов из веб-приложений	305
Одноэлементный сеансовый бин (Singleton)	306
Асинхронные вызовы метода	307
Управляемые сообщением бины	309
Транзакции в Enterprise JavaBeans	310
Транзакции, управляемые контейнером	311
Транзакции, управляемые бином	313
Жизненный цикл Enterprise JavaBeans	315
Жизненный цикл сеансового бина с сохранением состояния	316
Жизненный цикл сеансового бина, не сохраняющего состояние	319
Жизненный цикл управляемых сообщением бинов	321
Служба таймера EJB	322
Выражения таймера EJB на основе календаря	324
Безопасность EJB	326
Аутентификация клиента	329
Резюме	330
10. Контексты и инжекция зависимости	332
Именованные бины	332
Инжекция зависимости	334
Квалификаторы	335
Контексты именованных бинов	338
Резюме	345
11. Веб-сервисы JAX-WS	346
Разработка веб-сервисов JAX-WS	346
Разработка клиента веб-сервиса	351
Отправка вложений веб-сервисам	357

Представление EJB как веб-сервисов	359
Клиенты веб-сервиса EJB	360
Безопасность веб-сервисов	360
Безопасность веб-сервисов EJB	362
Резюме	364
12. RESTful веб-сервисы в Jersey и JAX-RS	365
Введение в веб-сервисы RESTful и JAX-RS	365
Разработка простого веб-сервиса RESTful	366
Конфигурирование пути к ресурсам REST для нашего приложения	368
Конфигурирование через web.xml	368
Конфигурирование через аннотацию @ApplicationPath	369
Тестирование нашего веб-сервиса	370
Преобразование данных между Java и XML с помощью JAXB ..	372
Разработка клиента веб-сервиса RESTful	375
Параметры запроса и пути	377
Параметры запроса	377
Отправка параметров запроса через клиентский API Jersey	378
Параметры пути	380
Отправка параметров пути через клиентский API Jersey	382
Приложение А: Отправка электронной почты из приложений Java EE	384
Конфигурирование сервера GlassFish	384
Реализация функциональности доставки электронной почты	387
Приложение Б: Интеграция с IDE	390
NetBeans	390
Eclipse	392
Алфавитный указатель	397

Предисловие

Изложение материала в данной книге начинается с установки сервера GlassFish v.3 и развертывания Java-приложений. Затем объясняется, как разработать, сконфигурировать, упаковать и развернуть сервлеты; помимо прочего, уделяется внимание изучению обработки HTML-форм. По мере продвижения в изучении материала мы будем разрабатывать Серверные страницы Java (Java Server Pages (JSP)) и узнаем о неявных объектах JSP. Также мы познакомимся со всеми Библиотеками стандартных тегов JSP (JSP Standard Tag Library (JSTL)). Эта книга позволит нам лучше понять, как управлять данными, хранящимися в базе данных, через API Подключения к базе данных Java (Java Database Connectivity (JDBC)) и через API Персистентности Java (Java Persistence API (JPA)). Кроме того, мы узнаем о новых функциях, введенных в JPA 2.0, и разработаем приложения JSF 2.0 для их изучения и настройки. Затем настроим сервер GlassFish для работы с API Системы обмена сообщениями Java (Java Messaging System (JMS)) и изучим работу тем и очередей сообщений. Позже мы будем использовать API Контекстов и инъекции зависимости (Context and Dependency Injection (CDI)) для интегрирования различных уровней приложения, а также изучим веб-сервис на основе SOAP, используя для его разработки спецификацию JAX-WS. Наконец, мы узнаем больше о разработке веб-сервиса RESTful с использованием спецификации JAX-RS.

Наконец, в книге обсуждаются различные соглашения и аннотации Java EE 6, которые помогут упростить разработку корпоративных приложений Java. Также рассматриваются самые последние версии спецификаций Сервлета (Servlet), JSF, JPA, EJB и JAX-WS, а также новые дополнения к спецификации, такие как JAX-RS и CDI.

Темы, освещаемые в книге

В главе 1, *«Знакомство с сервером GlassFish»*, объясняется, как загрузить и установить сервер GlassFish. Здесь мы рассмотрим несколько методов развертывания приложений Java EE: через веб-консоль GlassFish, с помощью утилиты командной строки `asadmin` и путем копирования файла приложения в каталог авторазвертывания. Мы рассмотрим основные задачи администрирования GlassFish, такие как настройка доменов и настройка соединений с базой данных, добавление пулов соединений и источников данных.

В главе 2, *«Разработка и развертывание сервлета»*, показано, как разработать, сконфигурировать, упаковать и развернуть сервлеты. Также мы рассмотрим, как обра-

богаты информацией HTML-формы, получая доступ к объекту HTTP-запроса. В дополнение будет объяснена переадресация HTTP-запросов от одного сервера к другому, наряду с перенаправлением HTTP-отклика на другой сервер. Мы обсудим, как сохранить объекты в памяти с помощью запросов, присоединяя их к контексту сервера и HTTP-сеанса. Наконец, мы рассмотрим все важные новые функции Сервлета 3.0, включая конфигурирование веб-приложений с помощью аннотаций, подключение с помощью `web-fragment.xml`, программное конфигурирование сервлета и асинхронную обработку.

Глава 3, «*JavaServer Pages*», содержит сведения о том, как разработать и развернуть простую JSP-страницу. Мы рассмотрим, как получить доступ к неявным объектам, таким как запрос, сеанс и т. д., из JSP-страницы. Кроме того, мы выясним, как установить и получить значения свойств JavaBean с помощью тега `<jsp:useBean>`. Поговорим и о том, как включить одну JSP-страницу в другую во время выполнения с помощью тега `<jsp:include>` и во время компиляции с помощью директивы JSP `include`. Мы обсудим, как написать пользовательские JSP-теги, расширяющие `javax.servlet.jsp.tagext.SimpleTagSupport` или TAG-файлы. Также будет показано, как получить доступ к JavaBeans и их свойствам с помощью Унифицированного языка выражений (Unified Expression Language). Наконец, мы рассмотрим XML-синтаксис JSP, который позволяет нам разрабатывать XML-совместимые JSP-страницы.

Глава 4, «*Библиотека стандартных тегов JSP*», познакомит нас со всеми Библиотеками стандартных тегов JSP (JSP Standard Tag Library (JSTL)), включая библиотеки базовых тегов, тегов форматирования, SQL- и XML-тегов. Кроме этого, будут объяснены функции JSTL. В главе приведены примеры, поясняющие использование наиболее распространенных тегов JSTL; также упоминаются и описываются дополнительные JSTL-теги.

В главе 5, «*Подключение к базе данных*», рассказывается о том, как получить доступ к данным в базе данных с помощью API Подключения к базе данных Java (Java Database Connectivity (JDBC)) и через API Персистентности Java (Java Persistence API (JPA)). Будут рассмотрены определения однонаправленных и двунаправленных отношений «один к одному», «один ко многим» и «многие ко многим» между JPA-сущностями. Помимо прочего, мы обсудим, как использовать первичные ключи составной JPA-сущности путем разработки пользовательских классов первичного ключа. Также будет показано, как получить объекты из базы данных путем использования Языка запросов персистентности Java (Java Persistence Query Language (JPQL)). Вы узнаете, как создавать программные запросы с помощью API Критериев JPA 2.0 (JPA 2.0 Criteria API) и автоматизировать проверку допустимости данных с помощью поддерживаемой JPA 2.0 Проверки допустимости со стороны бинов (Bean Validation Support).

Глава 6, «*JavaServer Faces*», посвящена разработке веб-приложений с использованием JavaServer Faces – стандартного каркаса компонентов для платформы Java EE 5. Мы поговорим о том, как написать простое приложение, создав JSP-страницу, содержащую JSF-теги и управляемые бины (managed beans). Обсудим, как проверить

данные, вводимые пользователем, путем использования стандартных блоков проверки допустимости JSF и создаваемых нами собственных блоков проверки допустимости или путем написания методов блока проверки допустимости. Рассмотрим, как настроить стандартные сообщения об ошибках JSF – и текст сообщения, и стиль сообщения (шрифт, цвет и т. д.). Наконец, узнаем, как написать приложения, интегрирующие технологии JSF и API Персистентности Java (JPA).

Глава 7, «*Служба обмена сообщениями Java*», повествует о том, как настроить в GlassFish фабрики соединений JMS, очереди и темы сообщений JMS, используя веб-консоль GlassFish. Мы рассмотрим, как отправлять и получать сообщения в очередь и из очереди сообщений JMS. Обсудим, как отправлять и получать сообщения в тему и из темы сообщений JMS. Узнаем, как просмотреть сообщения в очереди сообщений не удаляя их из очереди. В завершение будет показано, как создать и настроить темы JMS и взаимодействовать с долговременными подписчиками (durable subscriptions) на них.

Глава 8, «*Безопасность*», рассказывает о том, как использовать для аутентификации наших веб-приложений установленные по умолчанию области (realms) GlassFish. Мы рассмотрим область файла (file realm), которая хранит пользовательскую информацию в плоском файле, и область сертификата (certificate realm), которая требует клиентских сертификатов для пользовательской аутентификации. Кроме того, обсудим, как создать дополнительные области аутентификации, которые ведут себя точно так же, как области по умолчанию, при использовании классов области, включенных в GlassFish.

Глава 9, «*Enterprise JavaBeans*», объясняет, как реализовать бизнес-логику с помощью сохраняющих и не сохраняющих состояние сеансовых биннов (session beans). Кроме того мы разьясим понятия транзакций, управляемых контейнером, и транзакций, управляемых бином. Будут рассмотрены жизненные циклы для различных типов Enterprise JavaBeans. Мы поговорим о том, как периодически вызывать методы EJB – контейнером EJB, используя возможности службы таймера EJB. Наконец, мы расскажем, как сделать так, чтобы EJB методы могли быть вызваны только авторизованными пользователями.

Глава 10, «*Контексты и инъекция зависимости*», расскажет нам о том, как JSF страница, может получить доступ к именованным бинам (named beans) CDI, как будто они являются управляемыми бинами (managed beans) JSF. Мы объясним, как CDI облегчает внедрение (инъекцию) зависимостей в наш код. Мы обсудим, как можно использовать квалификаторы (qualifiers), чтобы определить, какие конкретно реализации зависимостей инжектировать в наш код. Наконец, мы рассмотрим все контексты, в которые может быть помещен бин CDI.

Глава 11, «*Веб-сервисы JAX-WS*», рассматривает, как разработать веб-сервисы и клиентов веб-сервисов с помощью API JAX-WS. Мы обсудим, как отправлять вложения веб-сервису, как представить методы EJB в качестве веб-сервисов и, наконец, как обеспечить безопасность веб-сервисов, чтобы они не были доступны неавторизованным клиентам.

В главе 12, «*RESTful веб-сервисы в Jersey и JAX-RS*», обсуждается, как легко и быстро разработать RESTful веб-сервисы, используя API JAX-RS – новое дополнение к спецификации Java EE. Мы объясним, как автоматически преобразовать данные между Java и XML, используя возможности API Java для связывания с XML (Java API for XML Binding (JAXB)). Наконец, мы рассмотрим, как передать параметры нашим RESTful веб-сервисам с помощью аннотаций `@PathParam` и `@QueryParam`.

Что нужно для чтения этой книги

Для чтения этой книги потребуется установить Комплект разработчика Java (Java Development Kit (JDK)) 1.5 или более новую версию, а также GlassFish v.3 или v.3.1. Настоятельно рекомендуется установить Maven 2, поскольку он используется во всех примерах кода, приведенного в книге. Наличие IDE Java, таких как NetBeans, Eclipse или IntelliJ IDEA, необязательно.

Для кого эта книга

Если Вы являетесь разработчиком Java и хотите стать специалистом по Java EE 6, эта книга для Вас. Чтобы чтение пошло Вам на пользу, необходимо иметь некоторый опыт работы с Java, а также опыт разработки и развертывания собственных приложений, однако никаких предварительных знаний о Java EE или J2EE не требуется. Вы также изучите, как использовать сервер приложений GlassFish v.3 для разработки и развертывания приложений.

Соглашения

В этой книге используется несколько стилей и пометок для выделения особо важной информации. Ниже мы поясним все эти типы выделений.

Элементы кода в тексте обозначаются моноширинным шрифтом, например: «XML-теги `<servlet>` и `<servlet-mapping>` используются для фактического конфигурирования нашего сервлета».

Блоки кода представлены следующим образом:

```
<servlet-mapping>
  <servlet>SimpleServlet</servlet>
  <url-pattern>*.foo</url-pattern>
</servlet-mapping>
```

Если мы хотим привлечь ваше внимание к определенной части блока кода, соответствующие фрагменты выделяются еще и жирным шрифтом:

```
<b>Application Menu</b>
<ul>
  <li> <a href="main.jsp">Main</a>
  <li> <a href="secondary.jsp">Secondary</a>
</ul>
Current page: <%= pageName %>
```

Ввод или вывод командной строки записывается так:

```
javac -cp /opt/sges-v3/glassfish/lib/javaee.jar
net/ensode/glassfishbook/simpleapp/SimpleServlet.java
```

Важные (ключевые) слова в тексте выделяются курсивом.

Элементы интерфейса программы, например пункты меню или поля в диалоговых окнах, отмечены жирным шрифтом: «Щелкните по кнопке **Далее** (Next)».

Для обозначения последовательно выполняемых действий используется символ | (вертикальная черта): «Щелкните по узлам **Ресурсы** (Resources) | **JDBC** | **Пулы соединений** (Connection Pools) в панели навигации».

Для удобства восприятия перечисляемые элементы оформлены в виде маркированных списков, например:

Элемент `<tag>` содержит несколько подэлементов:

- подэлемент `<name>`, который присваивает логическое имя пользователю тегу;
- подэлемент `<tag-class>`, который идентифицирует полностью определенное (квалифицированное) имя пользовательского тега;
- один или более подэлементов `<attribute>`, которые определяют атрибуты пользовательского тега.

Веб-адреса выделяются подчеркиванием и специальным шрифтом, например: «Мы сможем увидеть отображение файла `dataentry.html`, вводя в адресной строке обозревателя <http://localhost:8080/formhandling>».



Предупреждения или важные примечания отмечены в тексте таким образом.



Советы и рекомендации обозначены так.



Символ в правом нижнем углу нечетной страницы указывает что листинг кода имеет продолжение на следующем развороте страниц.

Поддержка клиентов

Теперь, когда Вы являетесь счастливым обладателем книги, у нас имеется возможность помочь Вам извлечь максимум пользы из Вашего приобретения.



Загрузите примеры кода для этой книги

Файлы с примерами кода для данной книги можно загрузить с сайта издательства :

<http://www.dmk-press.ru/>

Сообщения об ошибках

Хотя мы делаем все возможное для того, чтобы не допустить ошибок в наших изданиях, Вы можете встретить в тексте какие-либо неточности. В таком случае мы будем рады, если Вы сообщите нам об этом.

Сообщения об ошибках в русскоязычном издании этой книги можно оставить на сайте издательства «ДМК Пресс»: <http://www.dmk-press.ru/contacts1/contacts/>.

Незаконное воспроизведение содержимого книги

Пиратские копии – повсеместная проблема. Если Вам встретились незаконным образом растиражированные экземпляры данной книги (в любом формате), пожалуйста, сообщите источник публикации, написав письмо по адресу dm@dmk-press.ru или copyright@packtpub.com.

Вопросы

Вы можете присылать любые вопросы, касающиеся данной книги, по адресу dm@dmk-press.ru или questions@packtpub.com. Мы постараемся разрешить возникшие проблемы.

1

Знакомство с сервером GlassFish

В этой главе мы обсудим, как приступить к работе с сервером GlassFish. Вот некоторые из обсуждаемых тем:

- общий обзор Java EE и GlassFish;
- получение сервера приложений GlassFish;
- установка сервера приложений GlassFish;
- проверка установки сервера GlassFish;
- развертывание приложения Java EE;
- установка соединения с базой данных.

Общий обзор Java EE и GlassFish

Спецификация Java EE (ранее называемая J2EE) включает в себя стандартный набор технологий для разработки серверных приложений Java. Технологии Java EE включают, среди прочего, Сервлеты (Servlets), Серверные страницы Java (Java Server Pages (JSP)), Каркас стандартных компонентов Java Server Faces (JSF), Компоненты корпоративных приложений (Enterprise Java Beans (EJB)), Службу обмена сообщениями Java (Java Messaging Service (JMS)), API Персистентности Java (Java Persistence API (JPA)), используемый для сохранения сущностей в базе данных, API Java для веб-сервисов XML (Java API for XML Web Services (JAX-WS)) и API Java для веб-сервисов RESTful (Java API for RESTful Web Services (JAX-RS)). Существует несколько коммерческих вариантов серверов приложений и несколько вариантов серверов приложений с открытым исходным кодом. Серверы приложений Java EE позволяют разработчикам разрабатывать и развертывать Java EE-совместимые приложения; одним из таких серверов приложений является сервер GlassFish. В числе других серверов приложений Java EE с открытым исходным кодом – JBoss Red Hat, Apache Software Foundation’s Geronimo и ObjectWeb JOnAS. Коммерческие варианты сервера – Oracle (ранее BEA) Weblogic, IBM Websphere и Oracle Application Server.

GlassFish является сервером приложений Java EE с открытым исходным кодом, находящимся в свободном доступе. GlassFish лицензируется в соответствии с Общей лицензией разработки и распространения (Common Development and Distribution License (CDDL)).



Чтобы узнать больше о лицензиях GlassFish, обратитесь к сайту http://glassfish.java.net/public/CDDL+GPL_1_1.html.

Как полностью совместимый сервер приложений Java EE, GlassFish предоставляет необходимые библиотеки, позволяющие нам разрабатывать и развертывать Java-приложения, соответствующие спецификации Java EE.

Новые возможности Java EE

Java EE 6 на сегодняшний день является самой последней версией спецификации Java EE, включающей несколько усовершенствований и дополнений. В следующих разделах перечислены основные усовершенствования спецификации, которые представляют интерес для разработчиков корпоративных приложений.

JavaServer Faces (JSF) 2.0

Java EE 6 включает новую версию JSF. В каркасе стандартных компонентов JSF 2.0 появились следующие важные возможности:

- JSF 2.0 принял фэйслеты (Facelets) в качестве части официальной спецификации. Фэйслеты – технология представления, специально разработанная для JSF. В число преимуществ фэйслетов входят следующие возможности: возможность определения представления в XHTML, возможность упрощения создания шаблонов и возможность разработки компонентов JSF только с использованием разметки – без использования кода Java;
- JSF 2.0 также включает возможность конфигурирования приложения JSF с использованием аннотаций, значительно снижая, таким образом, а во многих случаях и полностью устраняя необходимость использования XML для конфигурирования.

Enterprise JavaBeans (EJB) 3.1

Ранние версии спецификации EJB приобрели репутацию сложных в практическом применении.

EJB 3.0 сделал большой шаг по пути к упрощению разработки EJB. EJB 3.1, в свою очередь, добавил новые возможности, которые еще больше упростили разработку EJB, а именно:

- локальные интерфейсы теперь являются не обязательными, поскольку фактический экземпляр бина может быть инжектирован (injected) в локального клиента;
- одиночный (Singleton) сеансовый бин может использоваться для управления состояниями приложения;

Конец ознакомительного фрагмента.
Приобрести книгу можно
в интернет-магазине
«Электронный универс»
e-Univers.ru