ОГЛАВЛЕНИЕ

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. Использует Java в качестве основного языка программирования с 1996 года. Ему довелось работать во многих крупных проектах для ряда клиентов, в числе которых 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 является главным редактором Ensode.net (http://www.ensode.net), веб-сайта, посвященного Java, Linux и другим технологиям. Часто выступает на конференциях Java-разработчиков, таких как JavaOne. Вы можете следовать за Дэвидом в Твиттере, его учетная запись: @ensode.

О РЕЦЕНЗЕНТАХ

Штефан Хорошовец (Stefan Horochovec) из Бразилии. Имеет ученую степень в области проектирования программного обеспечения и в настоящее время работает как программный архитектор.

Последние 10 лет занимался разработкой корпоративных приложений на Java, с использованием серверов приложений, таких как GlassFish, JBoss, Weblogic и WildFly.

Имеет 4-летний опыт работы с такими технологиями создания пользовательских интерфейсов, как Apache Flex (три года подряд выступал с докладами на конференции FlexMania – небывалое событие в Apache Flex в Латинской Америке), Struts и JSF. В настоящее время занимается проектами на основе фреймворка JSF 2 и JavaScript, с сильным уклоном в сторону AngularJS.

Имеет опыт работы с мобильными технологиями в течение 6 лет, хорошо знает платформу Android. Был одним из первых преподавателей по платформе Android в Бразилии выступал с докладами на конференции, посвященной Android, в Бразилии. Два года занимался разработкой корпоративных веб-приложений на основе HTML для мобильных устройств, используя такие фреймворки, как PhoneGap.

В 2014 Штефана пригласили присоединиться к программе «BlackBerry Elite Member», объединившей около 100 специалистов по всему миру, осознающих важность развития мобильных технологий и использующих операционную систему BlackBerry на одноименных устройствах в качестве мобильной платформы.

Штефан преподает в университете дисциплины, связанные с разработкой мобильных и веб-приложений, а также ведет курсы на предприятиях, где преподает Java, HTML/JS/CSS3, PhoneGap, Git и серверы приложений для Java.

Тим Пинет (Tim Pinet) – практикующий программист и веб-разработчик. В настоящее время живет в Оттаве, Канада. С раннего возраста начал интересоваться всем, что связано с электроникой и в

свое время поступил на факультет проектирования и разработки программного обеспечения. Оттава – крупный столичный город с развитым промышленным производством, предоставляющим богатые возможности, поэтому Тим получил отличный шанс попрактиковаться в разработке программного обеспечения в частных (Computer Associates, Emergis, Telus, Nortel) и общественных компаниях (City of Ottawa) в самых разных областях, таких как транспорт, здравоохранение, связь и муниципальные услуги.

Как приверженец идее открытого программного обеспечения, Тим готов работать за невысокую плату, но с большой отдачей и в любых окружениях. Часто вносит свой вклад в развитие открытых проектов (таких как Apache и SourceForge) и делится своими знаниями в сообществах (таких как Stackoverflow и его персональный блог). В своей работе широко использует инструменты с открытым исходным кодом, чем экономит своим работодателям тысячи долларов, привносит передовые приемы ускоренной разработки и тестирования, не требующие финансовых затрат и не влекущих потерю качества.

Обожая все, что связано с программированием и Интернетом, Тим постоянно удовлетворяет свои желания использовать новейшие технологии для повышения качества обслуживания клиентов. Он обладает обширным опытом практического применения корпоративных технологий Java и веб-служб, разработки пользовательских интерфейсов и серверных компонентов приложений, управления базами данных и интеграции служб SOAP. Он очень ценный командный игрок и лучше всего проявляет себя в руководстве коллективами и принятии архитектурных решений.

Чираг Сангани (Chirag Sangani) – программист, живет в пригороде Сиэтла. Получил степень магистра в Стэнфордском университете (штат Калифорния), и бакалавра – в Индийском технологическом институте, в городе Каннур (Индия). В настоящее время работает инженером-программистом в Microsoft.

ПРЕДИСЛОВИЕ

Java Enterprise Edition 7, последняя версия Java EE, добавила несколько новых особенностей в спецификацию. Некоторые существующие Java EE API претерпели значительные усовершенствования в этой версии спецификации; другие, совершенно новые API, были добавлены в спецификацию Java EE. Эта книга охватывает самые последние версии наиболее популярных спецификаций Java EE, включая JavaServer Faces (JSF), Java Persistence API (JPA), Enterprise JavaBeans (EJB), Contexts and Dependency Injection (CDI), новый Java API for JSON Processing (JSON-P), WebSocket, полностью обновленный Java Messaging Service (JMS) API 2.0, Java API for XML Web Services (JAX-WS) и Java API for RESTful Web Services (JAX-RS), а также приемы обеспечения безопасности в приложениях Java EE.

Сервер приложений GlassFish является эталонной реализацией для Java EE; это первый сервер приложений для Java EE на рынке, обеспечивший поддержку Java EE 7. Эта книга охватывает GlassFish 4.0, последнюю версию мощного сервера приложений с открытым исходным кодом.

Темы, освещаемые в книге

В главе 1, «Знакомство с сервером GlassFish», объясняется, как установить и настроить сервер GlassFish и как развертывать приложения Java EE через веб-консоль GlassFish. Наконец, здесь рассматриваются основные задачи администрирования GlassFish, такие как настройка доменов и соединений с базой данных, добавление пулов соединений и источников данных.

В главе 2, «JavaServer Faces», рассказывается о разработке приложений с использованием фреймворка JSF и его новых особенностей, включая поддержку HTML5-подобной разметки и потоков Faces Flows. Кроме того, здесь рассказывается о приемах проверки ввода пользователя с применением стандартных валидаторов JSF, а также о том, как создавать собственные валидаторы и определять методы, осуществляющие проверку.

Глава 3, «Объектно-реляционное отображение в JPA», обсуждает вопросы взаимодействий с системами управления реляционными базами данных (СУРБД), такими как Oracle или MySQL с применением Java Persistence API.

Глава 4, «Enterprise JavaBeans», объясняет, как разрабатывать приложения с использованием сеансовых компонентов и компонентов, управляемых сообщениями. Описывает основные особенности EJB, такие как поддержка управления транзакциями, служба таймеров EJB и безопасность. Рассматривает жизненные циклы разных типов компонентов Enterprise JavaBeans и рассказывает, как обеспечить автоматический вызов методов компонентов контейнером EJB в определенные моменты жизненного цикла.

Глава 5, «Контексты и внедрение зависимостей», представляет собой введение в механизм поддержки контекстов и внедрения зависимостей (Contexts and Dependency Injection, CDI). Эта глава познакомит вас с именованными компонентами CDI, квалификаторами CDI и расскажет, как внедрять зависимости с помощью CDI.

Глава 6, «Обработка JSON с помощью JSON-P API», рассказывает, как создавать и обрабатывать данные в формате JSON с применением нового прикладного интерфейса JSON-P. Она охватывает оба API для обработки JSON: Model API и Streaming API.

Глава 7, «Веб-сокеты», рассказывает, как создавать веб-приложения, поддерживающие полноценные двусторонние взаимодействия между клиентом и сервером, в противоположность традиционным, опирающимся на традиционный цикл запрос/ответ.

Глава 8, «Служба обмена сообщениями Java», повествует о том, как настроить в GlassFish фабрики соединений JMS, очереди и темы сообщений JMS, используя веб-консоль GlassFish. Здесь также рассказывается, как организовать обмен сообщениями между приложениями с использованием полностью обновленного JMS 2.0 API.

Глава 9, «Безопасность приложений Java EE», рассказывает, как обезопасить приложения Java EE с применением стандартных областей безопасности (security realms), а также о том, как добавлять собственные области безопасности.

Глава 10, «Веб-службы JAX-WS», охватывает приемы разработки веб-служб и их клиентов с помощью JAX-WS API. Здесь также рассматривается возможность автоматического создания программного кода с использованием инструментов сборки ANT и Maven.

Глава 11, «Веб-службы RESTful JAX-RS», описывает приемы создания веб-служб RESTful с использованием нового прикладного ин-

терфейса Java API for RESTful Web Services, а также приемы создания клиентов веб-служб RESTful с применением совершенно нового стандартного клиентского JAX-RS API. Наконец, здесь рассказывается, как автоматически преобразовывать данные между Java и XML, используя возможности Java API для связывания с XML (Java API for XML Binding, JAXB).

Что нужно для чтения этой книги

Для чтения этой книги и опробования примеров из нее потребуется установить следующее программное обеспечение:

- комплект разработчика Java (Java Development Kit, JDK) версии 1.7 или выше;
- GlassFish 4.0;
- Maven версии 3 или выше (необходим для сборки примеров);
- среда разработки на Java, такая как NetBeans, Eclipse или IntelliJ IDEA (необязательно, но рекомендуется).

Кому адресована эта книга

Эта книга предполагает знакомство читателя с языком программирования Java, поэтому она адресована разработчикам приложений на Java, желающим освоить Java EE, а также разработчикам приложений Java EE, желающим освежить свои знания и познакомиться с последней версией спецификации Java EE.

Соглашения

В этой книге вы обнаружите несколько стилей оформления текста, которые разделяют различные виды информации. Ниже приводятся примеры этих стилей и поясняется их значение.

Элементы программного кода в тексте, имена таблиц в базах данных, имена папок и файлов, расширения файлов, пути к каталогам в файловой системе, фиктивные адреса URL, ввод пользователя и учетные записи в Twitter оформляются так: «Аннотация `@Named` на уровне класса указывает, что этот компонент является именованным компонентом CDI».

```
if (!emailValidator.isValid(email)) {
  FacesMessage facesMessage =
    new FacesMessage(htmlInputText.getLabel()
        + ": email format is not valid");
```

```
    throw new ValidatorException(facesMessage);
}
```

Чтобы привлечь ваше внимание к определенной части в блоке кода, соответствующие строки или элементы будут выделены жирным шрифтом:

```
<ejb>
  <ejb-name>CustomerDaoBean</ejb-name>
  <ior-security-config>
    <as-context>
      <auth-method>username_password</auth-method>
      <realm>file</realm>
      <required>true</required>
    </as-context>
  </ior-security-config>
</ejb>
```

Любой ввод или вывод в командной строке оформляется так:

```
$ ~/GlassFish/glassfish4/bin $ ./asadmin start-domain
Waiting for domain1 to start ........
```

Новые термины и **важные (ключевые) слова** в тексте выделяются жирным. Элементы интерфейса программ, например пункты меню или поля в диалогах выделяются так: «Щелкните на кнопке **Next** (Далее), чтобы перейти к следующему экрану».

Предупреждения или важные примечания отмечены в тексте таким образом.

Советы и рекомендации обозначены так.

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или может быть не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв прямо на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставить комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору

по адресу dmkpress@gmail.com, при этом напишите название книги в теме письма.

Если есть тема, в которой вы квалифицированы, и вы заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Загрузка исходного кода примеров

Загрузить файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте www.dmkpress.com или www.дмк.рф в разделе «Читателям – Файлы к книгам».

Список опечаток

Хотя мы приняли все возможные меры, чтобы удостовериться в качестве наших текстов, ошибки всё равно случаются. Если вы найдёте ошибку в одной из наших книг – возможно, ошибку в тексте или в коде – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от расстройств и поможете нам улучшить последующие версии этой книги.

Если вы найдёте какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу dmkpress@gmail.com, и мы исправим это в следующих тиражах.

Нарушение авторских прав

Пиратство в Интернете по-прежнему остается насущной проблемой. Издательства ДМК Пресс и Packt очень серьезно относится к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в Интернете с незаконно выполненной копией любой нашей книги, пожалуйста, сообщите нам адрес копии или веб-сайта, чтобы мы могли принять меры.

Пожалуйста, свяжитесь с нами по адресу электронной почты dmkpress@gmail.com со ссылкой на подозрительные материалы.

Мы высоко ценим любую помощь по защите наших авторов, и помогающую нам предоставлять вам качественные материалы.

Вопросы

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

ГЛАВА 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 и запуск сервера приложений GlassFish;
- описание понятия доменов GlassFish;
- развертывание приложений Java EE;
- установка соединения с базой данных.

Общий обзор Java EE и GlassFish

Спецификация Java Enterprise Edition (Java EE, ранее называвшаяся J2EE, или Java 2 Enterprise Edition) включает стандартный набор технологий для разработки серверных приложений на Java. В число таких технологий входят: фреймворк **JavaServer Faces (JSF)**, компоненты корпоративных приложений **(Enterprise Java Beans**, **EJB)**, служба обмена сообщениями Java **(Java Messaging Service**, **JMS)**, прикладной интерфейс хранения данных **(Java Persistence API**, **JPA)**, Java API для веб-сокетов **(Java API for WebSocket)**, механизм контекстов и внедрения зависимостей **(Contexts and Dependency Injection**, **CDI)**, Java API для веб-служб XML **(Java API for XML Web Services**, **JAX-WS)**, Java API для веб-служб RESTful **(Java API for RESTful Web Services**, JAX-RS) и Java API для обработки данных в формате JSON **(Java API for JSON Processing**, **JSON-P)**.

Существует несколько коммерческих версий серверов приложений и несколько версий с открытым исходным кодом. Серверы приложений Java EE позволяют создавать и развертывать Java EE-совместимые приложения; одним из таких серверов приложений яв-

ляется сервер GlassFish. В числе других серверов приложений с открытым исходным кодом для Java EE можно назвать Red Hat WildFly (ранее известный как JBoss), Apache Software Foundation Geronimo и ObjectWeb JOnAS. Из коммерческих версий серверов можно назвать Oracle Weblogic, IBM Websphere и Oracle Application Server.

GlassFish – эталонная реализация сервера приложений для Java EE 7. В нем в самом первом были реализованы самые последние Java EE API. GlassFish является открытым и бесплатным продуктом, и распространяется на условиях общей лицензии разработки и распространения (Common Development and Distribution License, CDDL).

Полный текст лицензии CDDL можно найти по адресу: http://opensource.org/licenses/CDDL-1.0.

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

Новые возможности Java EE 7

На сегодняшний день Java EE 7 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) 2.2

Java EE 7 включает новую версию фреймворка **JavaServer Faces (JSF)**. В версии фреймворка JSF 2.2 появились следующие важные возможности:

- JSF 2.2 поддерживает HTML5-подобную разметку, то есть, веб-страницы теперь можно писать с использованием стандартной разметки HTML 5 и атрибутов JSF;
- JSF 2.2 включает также Faces Flows, средство для объединения взаимосвязанных страниц в последовательность с определенными точками входа и выхода;
- Третьей важной особенностью, появившейся в версии JSF 2.2, являются контракты библиотек ресурсов (resource library contracts). Контракты библиотек ресурсов упрощают разра-

ботку веб-приложений, которые могут выглядеть и действовать по-разному для разных пользователей.

Java Persistence API (JPA) 2.1

Прикладной интерфейс JPA был введен в спецификации Java EE 5. Главной его целью было заменить объектные компоненты (Entity Beans), использовавшиеся на тот момент в качестве стандартной основы для создания объектно-реляционных отображений в Java EE. JPA перенял идеи сторонних объектно-реляционных фреймворков, таких как Hibernate и JDO, и сделал их частью стандарта.

В JPA 2.1 появились следующие новые возможности:

- введено понятие **конвертеров** (Converters), обеспечивающих возможность писать свой код для выполнения преобразований между значениями в базе данных и в объектах Java. Типичная проблема при работе с базами данных состоит в том, что значения в объектах Java должны отличаться от хранящихся в базе данных. Например, значения `1` и `0` часто используются в базах данных для обозначения `true` и `false` соответственно. В языке Java имеется превосходный логический тип, поэтому значения `true` и `false` желательно было бы использовать непосредственно;
- JPA Criteria API теперь может выполнять массовые изменения и удаления;
- в JPA 2.1 появилась поддержка хранимых процедур;
- добавлена аннотация `@ConstructorResult`, позволяющая возвращать Java-классы (но не сущности JPA) из запросов SQL.

Java API для веб-служб RESTful (JAX-RS) 2.0

JAX-RS – это Java API для разработки веб-служб RESTful. Веб-службы RESTful используют архитектуру передачи репрезентативного состояния (**Representational State Transfer**, **REST**). JAX-RS была принята в состав официальной спецификации Java EE в версии 6.

Служба сообщений Java Message Service (JMS) 2.0

Прикладной интерфейс Java Message Service (JMS) был полностью переработан в Java EE 7. Предыдущие версии JMS требовали писать массу типового кода; при использовании новой, переработанной версии JMS 2.0 API, этого больше не требуется.

Java API для обработки JSON (JSON-P) 1.0

JSON-P – это совершенно новый прикладной интерфейс, введенный в Java EE 7. Он позволяет генерировать и анализировать строки в формате **JSON (JavaScript Object Notation** – форма записи объектов JavaScript).

Java API для веб-сокетов 1.0

Традиционные веб-приложения используют модель запрос/ответ, то есть, клиент (обычно веб-браузер) запрашивает ресурсы, а сервер возвращает ответ. В этой модели взаимодействий инициатором всегда является клиент.

В спецификацию HTML5 была включена новая технология WebSockets (веб-сокеты). Она обеспечивает полноценные двусторонние взаимодействия между клиентом и сервером.

Преимущества GlassFish

Имеется много версий серверов приложений для Java EE, но почему чаще всего выбирают именно GlassFish? Помимо очевидных преимуществ бесплатного сервера, GlassFish предлагает множество других:

- **эталонная реализация Java EE**: сервер GlassFish является эталонной реализацией поддержки Java EE. Это означает, что разработчики других серверов приложений могут использовать GlassFish для сравнения, чтобы убедиться, что их продукты соответствуют спецификации. Теоретически GlassFish можно было бы использовать для отладки других серверов приложений. Если приложение, развернутое на другом сервере приложений, выполняется с ошибками, но после развертывания в GlassFish действует правильно, это скорее всего свидетельствует о неправильной работе другого сервера приложений;
- **поддерживает самые последние версии спецификации Java EE**. Поскольку GlassFish является эталонной реализацией спецификации Java EE, он начинает поддерживать самые последние нововведения в спецификации раньше, чем любые другие серверы приложений на рынке. Действительно, на момент написания этой книги GlassFish являлся единственным Java EE-совместимым сервером приложений на рынке, который поддерживал спецификацию Java EE 7 в полном объеме.

Получение GlassFish

Сервер GlassFish можно загрузить по адресу: http://glassfish.java.net/.

Сервер GlassFish 4.0 также входит в состав дистрибутива NetBeans IDE версии 7.4 и выше.

После перехода по указанному адресу, в окне веб-браузера откроется страница, как показано на рис. 1.1.

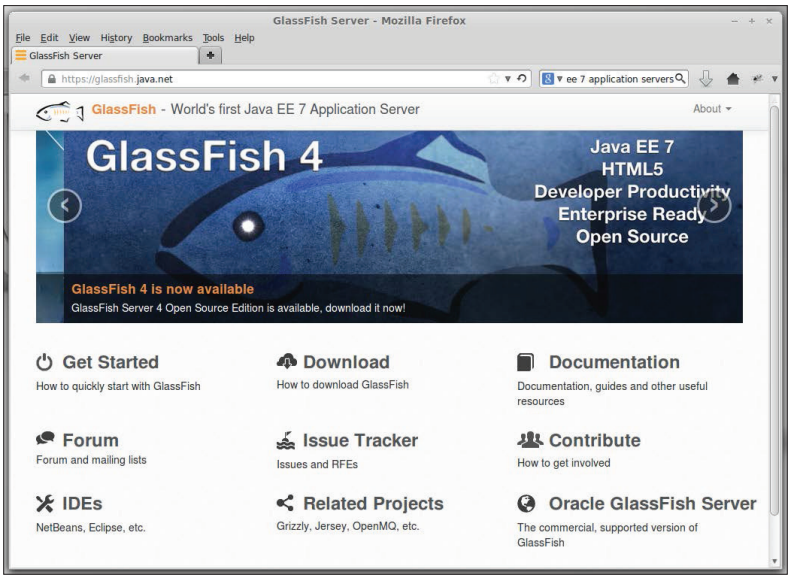

Рис. 1.1. Домашняя страница проекта GlassFish

Если щелкнуть на ссылке **Download** (Загрузить), откроется страница мастера, где будет предложено несколько вариантов для загрузки, как показано на рис. 1.2.

На странице загрузки предлагается на выбор несколько вариантов; можно загрузить полный комплект платформы Java EE (**Java EE 7 Full Platform**) или только комплект, ориентированный на разработку веб-приложений (**Java EE 7 Web Profile**). Можно также загрузить GlassFish в виде ZIP-архива или инсталлятора для выбранной операционной системы.

Чтобы иметь возможность опробовать все примеры из этой книги, нужно загрузить полный комплект платформы Java EE. Далее будет

описана установка из ZIP-архива, поскольку порядок установки в этом случае почти не отличается в разных операционных системах, но вы можете загрузить инсталлятор для своей платформы и выполнить установку с его помощью.

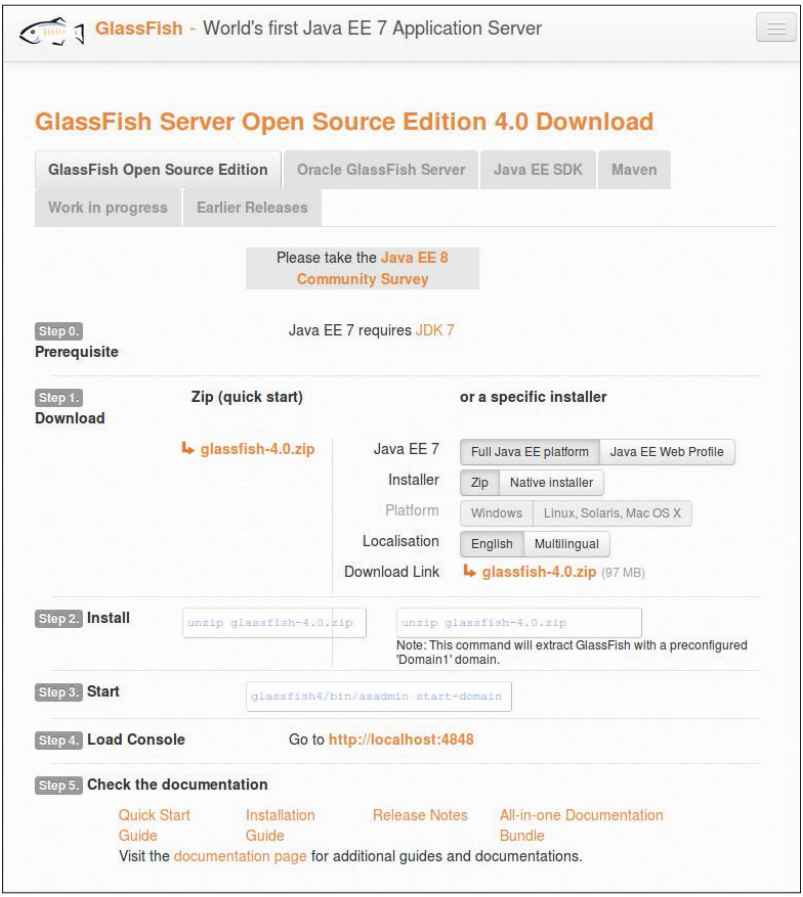

Рис. 1.2. Страница мастера с несколькими вариантами для загрузки

Установка GlassFish

Процесс установки будет проиллюстрирован на примере установки из ZIP-архива. Этот процесс почти не отличается во всех основных операционных системах.

Процесс установки GlassFish достаточно прост; тем не менее GlassFish предполагает, что в системе должны присутствовать определенные элементы, от которых он зависит.

Зависимости GlassFish

Для установки GlassFish 4 на рабочей станции должна быть установлена свежая версия комплекта разработчика Java **(Java Development Kit, JDK)** – требуется версия JDK 1.7 или выше – а выполняемый файл Java должен находиться в одном из каталогов, перечисленных в системной переменной окружения PATH. Последнюю версию JDK можно загрузить с сайта: http://www.oracle.com/technetwork/java/javase/downloads/index.html. За инструкциями по установке JDK обращайтесь к документации для своей платформы по адресу: http://docs.oracle.com/javase/7/docs/webnotes/install/index.html.

Установка

После установки JDK можно установить GlassFish, просто распаковав загруженный ZIP-архив, как показано на рис. 1.3.

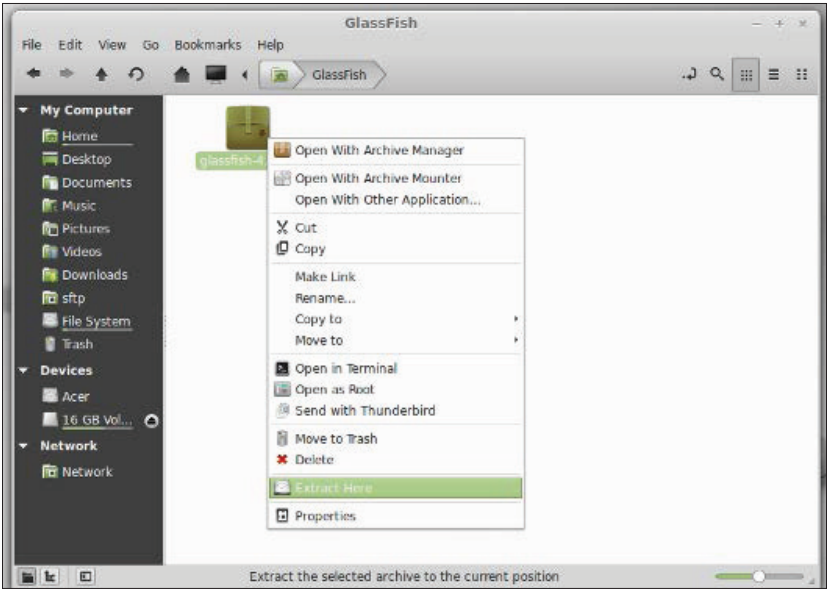

Рис. 1.3. Распаковка ZIP-архива

Все современные операционные системы, включая Linux, Windows и Mac OS X, имеют встроенную поддержку для распаковки ZIP-файлов. За подробностями обращайтесь к документации для своей операционной системы.

После распаковки появится новый каталог с именем `glassfish4`. Этот новый каталог содержит готовую к использованию версию GlassFish.

Запуск GlassFish

Чтобы запустить GlassFish, нужно перейти в `[каталог установки GlassFish]/GlassFish4/bin` и выполнить следующую команду:

```
./asadmin start-domain domain1
```

Эта команда, как и большинство команд, описываемых в этой главе, предполагает использование Unix или Unix-подобной операционной системы, такой как Linux или Mac OS. В Windows начальные символы `./` в команде не требуются.

Спустя несколько коротких мгновений после запуска команды, в окне терминала должно появиться следующее сообщение:

```
$ ~/GlassFish/glassfish4/bin $ ./asadmin start-domain

Waiting for domain1 to start ........

Successfully started the domain : domain1

domain Location: /home/heffel/GlassFish/glassfish4/glassfish/
domains/domain1

Log File: /home/heffel/GlassFish/glassfish4/glassfish/domains/
domain1/
logs/server.log

Admin Port: 4848

Command start-domain executed successfully.
```

Загрузите примеры кода. *Загрузить файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте www.dmkpress.com или www.дмк.рф в разделе «Читателям – Файлы к книгам».*

Затем можно открыть окно браузера и ввести в адресной строке строку URL:

```
http://localhost:8080.
```

Если все в порядке, должна появиться страница, как показано на рис. 1.4.

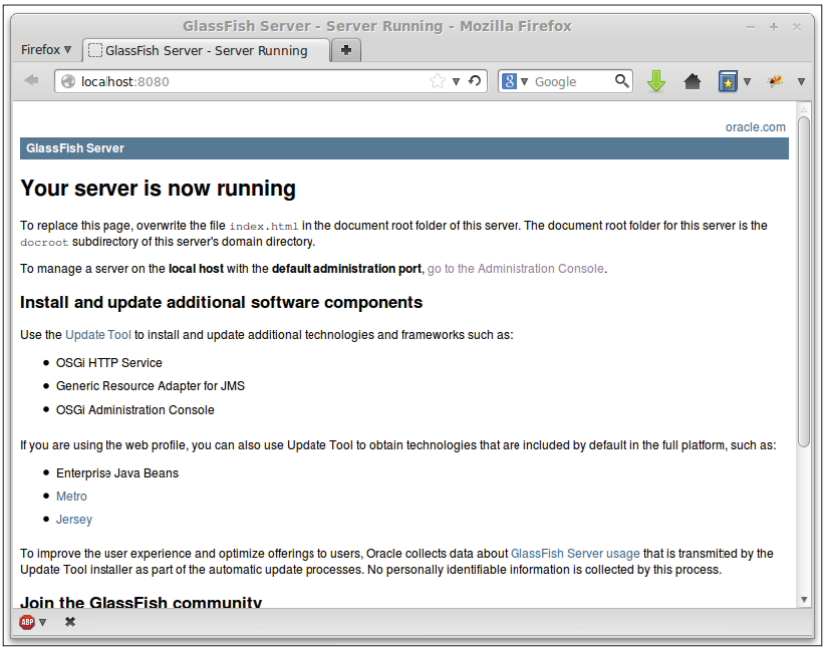

Рис. 1.4. Начальная страница сервера GlassFish

Получение справки. *Если какой-либо из предыдущих шагов завершился неудачей либо вам требуется общая справочная информация о сервере GlassFish, обращайтесь к замечательному информационному ресурсу – форуму GlassFish, по адресу: http://www.java.net/forums/glassfish/glassfish.*

Развертывание первого приложения Java EE

Чтобы убедиться в работоспособности сервера GlassFish, развернем **WAR**-файл (**Web ARchive** – веб-архив) и убедимся, что он развер-

тывается и выполняется без ошибок. Прежде чем двинуться дальше, загрузите файл `simpleapp.war` с веб-сайта www.dmkpress.com.

Развертывание приложения через веб-консоль

Чтобы развернуть `simpleapp.war`, запустите браузер и откройте в нем страницу с URL: `http://localhost:4848`. В браузере должна появиться страница администрирования, как показано на рис. 1.5.

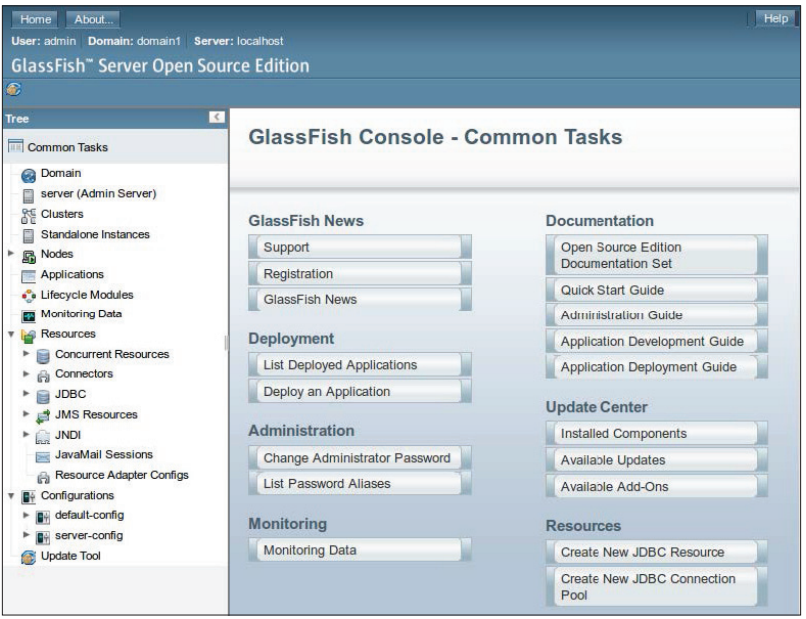

Рис. 1.5. Страница администрирования сервера GlassFish

По умолчанию сервер GlassFish настроен на работу в режиме разработки. В этом режиме нет необходимости вводить имя пользователя и пароль для доступа к веб-консоли GlassFish. В эксплуатационном окружении обязательно следует защитить доступ к веб-консоли паролем.

Теперь щелкните на пункте **Deploy an Application** (Развернуть приложение) в разделе **Deployment** (Развертывание), в центральной панели.

Чтобы развернуть приложение выберите переключатель **Local packaged file or directory that is accessible from the Application Server** (Локальный упакованный файл или каталог, доступный из сервера приложений) и введите путь к WAR-файлу или выберите его,

щелкнув на кнопке **Browse Files...** (Обзор файлов...). В последнем случае появится окно, как показано на рис. 1.6.

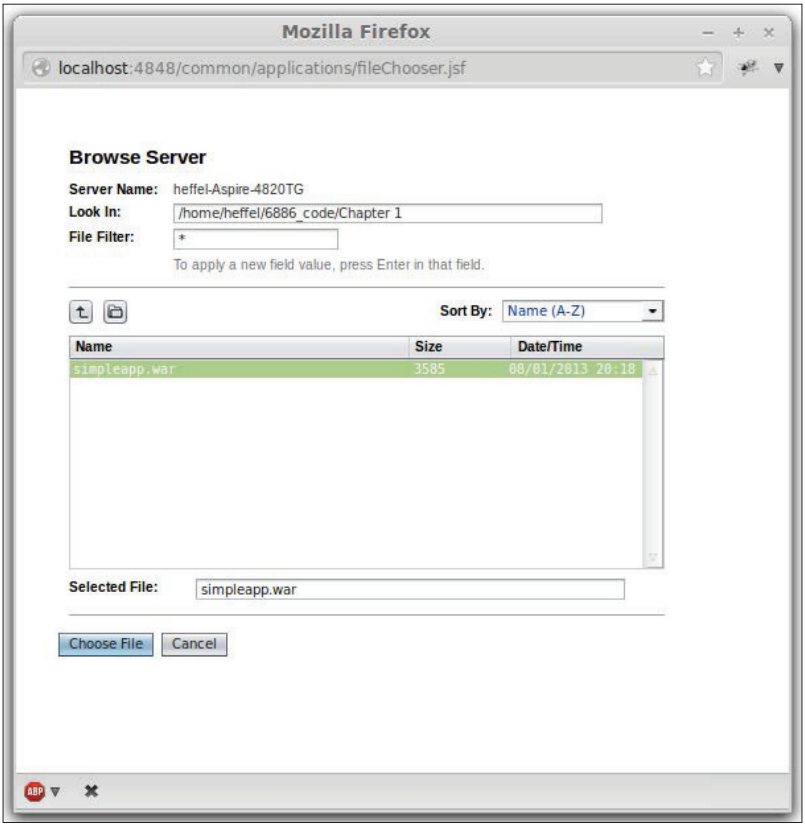

Рис. 1.6. Окно выбора файла для развертывания

После выбора WAR-файла будет предоставлена возможность уточнить некоторые параметры (см. рис. 1.7). Для нашей цели вполне подходят значения по умолчанию, поэтому можно просто щелкнуть на кнопке **OK** справа вверху, чтобы развернуть файл.

После развертывания в веб-консоли сервера GlassFish отобразится страница **Applications** (Приложения), где в списке будет присутствовать только что развернутое приложение (см. рис. 1.8).

Чтобы запустить приложение `simpleapp`, введите в адресной строке браузера строку URL:

```
http://localhost:8080/simpleapp/simpleservlet.
```

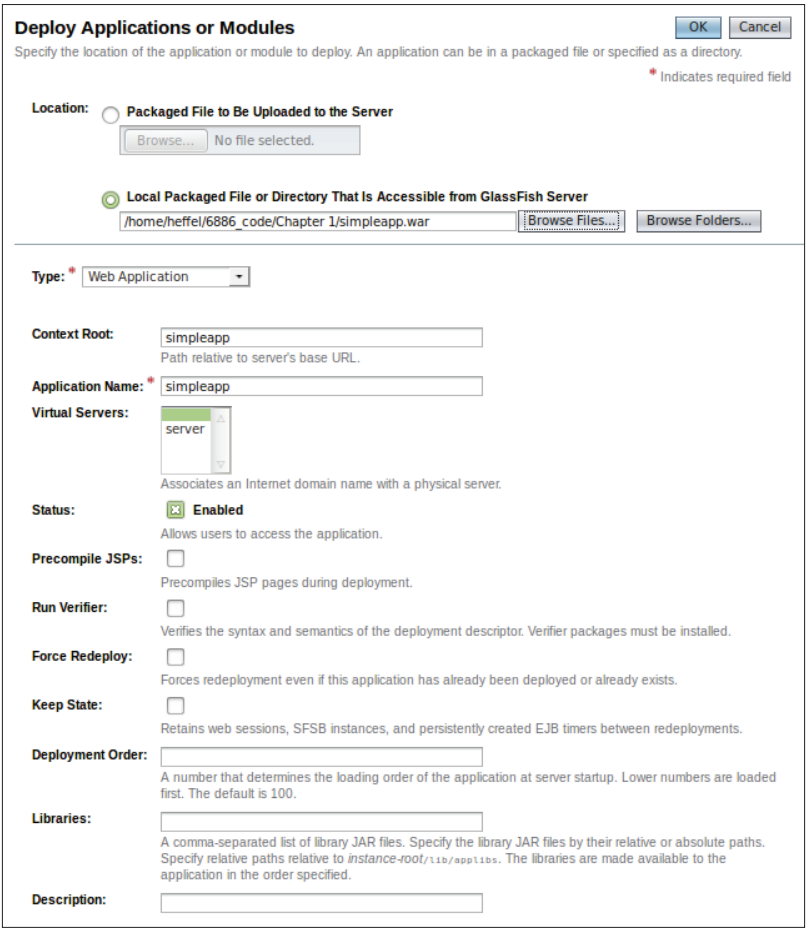

Рис. 1.7. Окончание развертывания WAR-файла

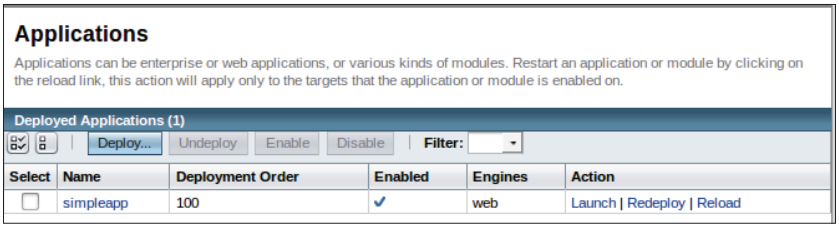

Рис. 1.8. Вновь развернутое приложение появится в списке приложений

В окне браузера должна открыться страница, как показано на рис. 1.9.

Рис. 1.9. Начальная страница приложения simpleapp

Вот и все! Мы успешно развернули первое приложение Java EE.

Удаление приложения через веб-консоль GlassFish

Чтобы удалить приложение, развернутое в предыдущем разделе, откройте консоль администрирования GlassFish, введя следующий URL в адресной строке браузера:

```
http://localhost:4848.
```

Затем щелкните на пункте **Applications** (Приложения) в панели навигации слева или на элементе **List Deployed Applications** (Список развернутых приложений) в основной панели консоли администрирования.

В любом случае должна открыться страница управления приложениями (см. рис. 1.10).

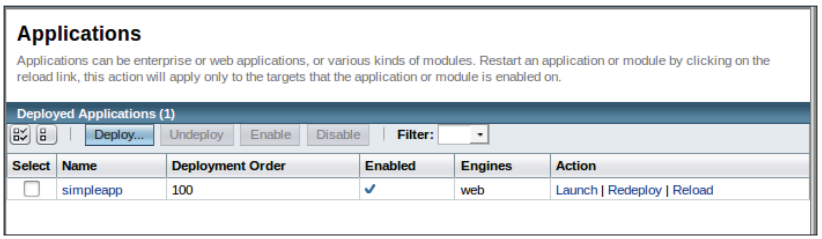

Рис. 1.10. Страница управления приложениями

Приложение можно удалить, просто выбрав его в списке развернутых приложений и щелкнув на кнопке **Undeploy** (Удалить). После этого страница управления приложениями будет выглядеть, как показано на рис. 1.11.