

اولین فنجان: آشنایی با Java EE

Your First Cup: An Introduction to the Java EE Platform

By

Sun Microsystems

October ۴, ۲۰۰۶

از

<http://javaborder.radmanitd.com>

فهرست

۲	مقدمه
۲	بخش ۱: مفاهیم
۲	Enterprise Applications
۲	Tiered Applications
۴	Java EE Servers
۴	Web Applications
۶	بخش ۲: ساخت firstcup Enterprise Application
۶	شرح کارکرد Application
۶	معماری
۶	Tier ها در firstcup Enterprise Application
۷	ساخت وب سرویس DukesAge
۸	ساخت firstcup Enterprise Application
۹	ساخت Enterprise Bean
۱۰	ساخت firstcup Web Client
۱۲	ساخت DukesBDay Managed Bean
۱۴	ساخت صفحه greeting.jsp
۱۶	ساخت صفحه response.jsp
۱۷	Building, Packaging, Deploying & Running
۱۸	منابع

مقدمه

درباره این نوشته

این نوشته ترجمه آزاد مقاله "Your First Cup: An Introduction to the Java EE Platform" است که توسط شرکت سان میکروسیستمز برای معرفی پلتفرم Java EE منتشر شده است و از آدرس اینترنتی <http://java.sun.com/javaee/o/docs/firstcup/doc/firstcup.pdf> قابل دریافت است. در این نوشته يك Web Application ساده بعنوان نمونه تعریف شده و مراحل پیاده سازی آن گام به گام توضیح داده شده است.

منظور از ترجمه آزاد این است که این نوشته ترجمه جزء به جزء مطلب اصلی نیست. در این ترجمه تمرکز روی ساخت Web Application است و برخی توضیحات اضافی و نامرتبط حذف شده است. همچنین در برخی قسمت ها برای روشن تر شدن مفاهیم، توضیحات و تصاویری اضافه گردیده است. در ضمن تعاریف Web Application در ابتدای نوشته نیز از راهنمای آموزشی اصلی Java EE شرکت sun به آدرس <http://java.sun.com/javaee/o/docs/tutorial/doc/JavaEETutorial.pdf> برداشت شده است.

این نوشته برای چه کسانی است

این نوشته برای کسانی است که برنامه نویسی می دانند و با زبان جاوا نیز آشنایی دارند. اما در تولید Java EE Application ها تازه کار اند. قبل از خواندن این نوشته باید:

- با زبان برنامه نویسی جاوا آشنا باشید.
- نرم افزارهای لازم را روی سیستم خود نصب کرده باشید.
- یک Browser سطح بالا روی سیستم خود داشته باشید.

نرم افزارهای مورد نیاز

برای پیاده سازی و اجرای این Application به نرم افزارهای زیر نیاز خواهید داشت.

- Java EE SDK
- NetBeans IDE ۵,۵

نرم افزارهای مذکور را می توانید به ترتیب از این آدرس ها دریافت نمایید:

<http://java.sun.com/javaee/downloads/index.jsp>

<http://www.netbeans.org/downloads/index.html>

نکته: استفاده از هر IDE دیگری نیز که با Java EE سازگار باشد امکان پذیر است. اما شرح مراحل کاری که در اینجا ارائه شده با توجه به امکانات NetBeans ۵,۵ است.

بخش ۱ : مفاهیم

Enterprise Applications

فلسفه ایجاد پلتفرم Java EE کمک به برنامه سازان جهت ساخت Application های بزرگ (-large scale)، چند لایه (multi-tiered)، قابل توسعه (Scalable)، قابل اطمینان (Reliable) و ایمن (Secure) است. اصطلاحاً چنین Application هایی Enterprise Application خوانده می شوند. تولید Enterprise Application با مشخصات ذکر شده اغلب کاری پیچیده است. وظیفه پلتفرم Java EE کاهش این پیچیدگی ها است. Java EE این کار را با ارائه مدل ها، API ها و محیط های اجرایی انجام می دهد و به برنامه سازان امکان می دهد فارغ از درگیر شدن با پیچیدگی های مذکور، تنها بر عملکرد نرم افزار متمرکز شوند. از مفاهیم مهم Enterprise Application ها مفهوم Tiering است که در ادامه به آن پرداخته خواهد شد.

Tiered Applications

Web Application معمولاً شامل Client Tier و Middle Tier و Data Tier است. توجه JavaEE بر لایه میانی است.

از نظر منطقی نیز می توان لایه های Application را بدین صورت در نظر گرفت:

Client Tier: عبارت است از کاربری که به Web Application دسترسی می یابد. تعریف مشخصات این بخش از دسترس برنامه ساز خارج است.

Web Tier: عبارت است از کامپوننت هایی که ارتباط بین client با Business tier را برقرار می کنند.

وظایف اصلی آن عبارت است از:

- ایجاد محتوای دینامیک به فرمت های گوناگون (برای client های گوناگون)
- دریافت اطلاعات وارد شده توسط کاربر و برگرداندن نتایج تولید شده توسط Business Tier
- کنترل نحوه حرکت client در صفحات
- مدیریت وضعیت داده های session کاربر
- انجام برخی کارهای پایه ای و نگهداری موقت برخی داده ها در JavaBean ها

تکنولوژی هایی که در لایه Web tier استفاده می شوند عبارتند از

- Servlet
- JSP
- JSF یا JavaServer Faces
- JavaServer Pages Standard Tag Library
- JavaBeans Component

Business Tier: این لایه شامل کامپوننت هایی است که Business logic را پیاده می کنند. منظور از Business logic کدهایی است که امکان استفاده از توابع و عملیات خاص حوزه کاری مربوطه را فراهم می کنند. در یک Application که از طراحی خوبی برخوردار است، باید توابع و کارکردهای اصلی تنها در این لایه پیاده شوند.

تکنولوژی هایی که در این لایه استفاده می شوند عبارتند از

- Enterprise JavaBeans components
- JAX-WS web service endpoints
- Java Persistence API entities

Enterprise Information Systems Tier یا EIS: این لایه شامل database server ها، سیستم های resource planning و کلیه منابع داده ای است. معمولاً این منابع روی کامپیوترهایی غیر از سرور Java EE نصب هستند. دستیابی به آنها از طریق کامپوننت های Business tier انجام می شود.

- در این لایه از تکنولوژی های زیر استفاده می شود:
- JDBC یا Java Database Connectivity API
 - Java Persistence API
 - J2EE Connector Architecture
 - JTA یا Java Transaction API

Java EE Servers

Java EE Server یک Application است که روی سرور اجرا می شود و امکان استفاده از API ها و سرویس های استاندارد Java EE را فراهم می کند. Java EE Server گاه Application Server نیز نامیده می شود. Java EE server در قالب چندین container، متناظر با لایه ها در multi-tiered Application سرویس هایی را برای انواع کامپوننت ها فراهم می سازد. Java EE Container ها عبارتند از:

Web Container

این container واسط میان web component ها و web server است. Web component می تواند Servlet، یک صفحه JSP یا یک صفحه JavaServer Faces باشد. Container چرخه حیات (lifecycle) و ارسال درخواست ها به کامپوننت ها را مدیریت می کند و برخی داده ها از جمله اطلاعات مربوط به درخواست جاری را در دسترس قرار می دهد.

Application Client Container

این container واسط بین Java EE Application Client (یک Application جاوا که از کامپوننت های Java EE Server استفاده می کند) و Java EE Server است. این container روی سیستم client اجرا می شود.

EJB Container

این container واسط میان enterprise bean ها (که business logic در آنها پیاده می شود) و Java EE Server است. این container روی Java EE Server اجرا می شود و اجرای enterprise bean ها را مدیریت می نماید.

Web Applications

بطور کلی Web Application نرم افزاری است که تحت وب مورد دستیابی قرار می گیرد. یکی از دلایل عمومیت یافتن Web Application ها این است که از همه جا قابل دسترس است و دسترسی به آن نیاز به نصب نرم افزار خاصی روی client ندارد.

sun، یک Web Application را بعنوان یک Web Server یا Application Server پیشرفته تعریف می کند و آن را به دو گونه، تقسیم می کند:

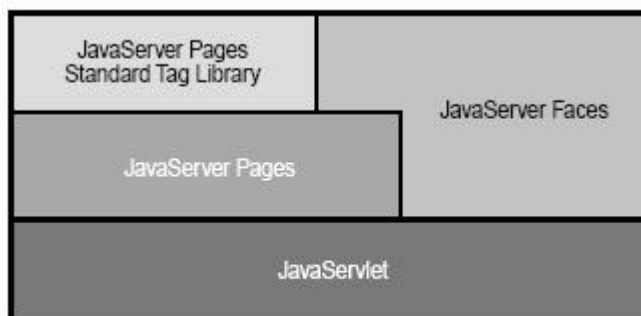
- Presentation-oriented: این نوع Web Application صفحات ویی ایجاد می کند که امکان تعامل با کاربر را فراهم می سازند. این صفحات می تواند انواع markup language ها همچون HTML و XML را دربر می گیرد. همچنین محتوای آنها بصورت دینامیک، با توجه به درخواست کاربر تولید می شوند.
- Service-oriented: این نوع Web Application امکان استفاده از Web Service را فراهم می کند. اغلب، Presentation-oriented Application ها نقش client را برای Service-oriented Application ها ایفا می کنند.

:Web Components

در Java 2 platform تولید محتوای دینامیک با صفحات jsp یا Servlet ها انجام می شود که به Web Component معروفند.

Servlet ها کلاس های ایجاد شده به زبان جاوا هستند که بصورت دینامیک درخواست ها را پردازش و پاسخ ها را تولید می کنند. صفحات jsp فایل های متنی هستند که مانند servlet عمل می کنند اما رفتار آنها در تولید محتوای دینامیک قابل درک تر است. اگرچه jsp و servlet به جای یکدیگر قابل استفاده هستند، اما هر کدام نقطه قوت و ضعف خاص خود را دارند. Servlet ها بیشتر برای ایجاد Service-oriented Application ها، همچنین توابع کنترلی Presentation-oriented Application

مناسب هستند. اما jsp ها بیشتر برای تولید متن های markup مثل HTML، WML، XML و SVG (Scalable Vector Graphics) مناسب هستند. پس از معرفی Servlet و jsp، تکنولوژی ها و framework های جدیدی برای تولید Web Application ها ایجاد شد. از جمله JavaServer Faces. با این حال Servlet زیربنای همه تکنولوژی های مذکور است.



Java Web Application Technologies

هریک از تکنولوژی های بعد از Servlet لایه ای بر abstraction افزودند و باعث افزایش سرعت و ارتقاء کیفیت تولید Web Application ها شدند. Web Component ها روی پلتفرمی به نام Web Container اجرا می شوند. انواع Web Container سرویس ها و امکانات لازم برای استفاده Web Component ها، همچنین امکان دسترسی به سایر API ها را فراهم می سازد. رفتار Web Application هنگامی که روی Web Container نصب (deploy) می شود قابل تنظیم (configuration) است. این تنظیمات در فایل هایی به فرمت XML به نام Deployment descriptor یا DD درج می شوند.

مراحل تولید یک Web Application

۱. ایجاد کدهای برنامه
۲. ایجاد Deployment descriptor برنامه
۳. کامپایل برنامه با همه کلاس های کمکی استفاده شده توسط Web Component ها
۴. فرایند اختیاری Package کردن Application به یک Deployment unit
۵. Deploy کردن Application روی Web Container
۶. انتساب یک URL به Web Application

بخش ۲ : ساخت firstcup Web Application

در این بخش مراحل ساخت یک Enterprise Application نمونه ای به نام firstcup بصورت گام به گام ارائه خواهد شد.

شرح کارکرد Application

نحوه کارکرد Application بدین صورت است که وقتی شما آن را توسط Browser اجرا می کنید، در صفحه ای تاریخ میلادی تولد شما دریافت می شود. سپس Application تاریخ تولد شما را با تاریخ تولد Duke که يك تاریخ مشخص (۲۳ ماه می ۱۹۹۵ ، تاریخی که اولین نسخه دمو زبان جاوا عرضه عمومی شد) است مقایسه می نماید. درنهایت Application وضعیت سن شما را نسبت به Duke به عنوان پاسخ برمی گرداند. اینکه آیا شما بزرگتر، کوچکتر یا هم سن Duke هستید.

معماری

این Enterprise Application شامل سه کامپوننت اصلی است:

- DukesAgeService : یک Web Service است و با JAX-WS پیاده می شود و سن Duke را محاسبه می نماید.
- DukesBirthdayBean : یک Enterprise bean است که تفاوت سن وارد شده توسط کاربر را از سن Duke محاسبه می کند.
- firstcup : یک Web Application که از JSF در صفحات استفاده می کند. firstcup از DukesAgeService برای نمایش سن Duke استفاده می کند. همچنین تاریخ وارد شده توسط کاربر را می خواند. با استفاده از DukesBirthdayBean تعیین می کند که سن کدام بیشتر است. سپس نتیجه مقایسه دو سن را برای کاربر نمایش می دهد.

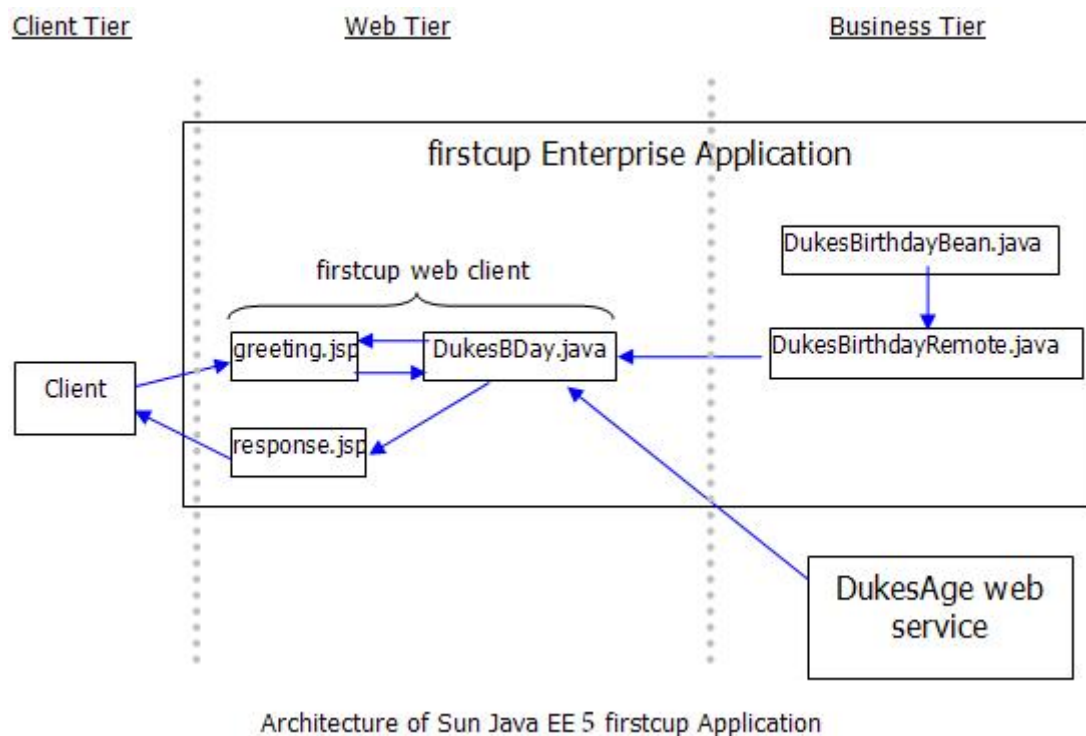
این Web Application شامل فایل های زیر است:

- فایل greeting.jsp یک صفحه jsp است که کاربر می توان در آن تاریخ تولد خود را وارد کند.
- فایل response.jsp یک صفحه jsp است که پاسخ را به کاربر نشان می دهد و اعلام می کند که آیا سن او از Duke بیشتر است یا کمتر یا هم سن هستند.
- فایل DukesBDay.java که یک managed Bean است و مشخصه ها را تعریف می کند، سن جاری Duke را از Web Service می گیرد. همچنین تفاضل دو سن را از enterprise Bean دریافت میکند.
- فایل faces-config.xml برای configure نمودن بخش های مرتبط با JSF به کار می رود.
- فایل web.xml که Deployment Descriptor برنامه است و تنظیمات Application برای نصب در آن ثبت می گردد.

Tier ها در این Application

این مثال دارای یک کامپوننت Web Tier (firstcup) ، دو کامپوننت Business Tier (DukesAgeService و DukesBirthdayBean) است. همچنین Browser کاربر client Tier است. ضمناً این مثال EIS Tier ندارد.

Tier های این Application را به همراه کلیه فایل ها و کلاس های اصلی Application در شکل زیر ملاحظه می کنید.



شروع پیاده سازی

ایجاد یک فولدر برای پروژه

ابتدا برای قرار دادن کلیه فایل های مربوط به پروژه یک فولدر خاص ایجاد می نمایم. نام این فولدر را myexample بگذارید.

ایجاد وب سرویس

خروجی این وب سرویس، برگرداندن سال سن Duke است. این وب سرویس هیچ پارامتر ورودی ندارد. در این مرحله یک پروژه برای این وب سرویس ایجاد می کنیم. سپس یک کلاس جاوا ساده می سازیم و تغییرات لازم را برای تبدیل آن به یک وب سرویس اعمال می کنیم و کدهای لازم را می نویسیم. مراحل گام به گام ساخت آن بدین شرح است:

تعریف پروژه

۱. در NetBeans از منوی File منوی New Project را انتخاب نمایید.
۲. در category های موجود web را انتخاب کنید.
۳. از انواع پروژه ها Web Application را انتخاب نمایید.
۴. Next را کلیک کنید.
۵. نام پروژه را firstcup-dukes-age قرار دهید.
۶. Location پروژه را همان فولدر myexample که ساخته اید قرار دهید.
۷. Application Server را Java System Application Server انتخاب کنید.
۸. Java EE version را Java EE ۵ قرار دهید.
۹. Finish را کلیک کنید. اکنون باید پروژه شما ساخته شده باشد و در پنل Projects در سمت چپ NetBeans قابل رویت باشد.
۱۰. در قسمت web pages از منوی درختی پروژه فایل index.jsp را بیابید. روی آن کلیک راست کنید و سپس با استفاده از گزینه delete آن را حذف کنید.

ساخت کلاس DukesAge

۱. پروژه firstcup-dukes-age را در NetBeans انتخاب کنید.
۲. از منوی File گزینه New File را انتخاب کنید.
۳. از category ها Java Classes را انتخاب کنید.
۴. از انواع فایل نیز Java Class را انتخاب کنید.
۵. Next را کلیک کنید.
۶. نام کلاس را DukesAge قرار دهید.
۷. در قسمت Package عبارت com.sun.firstcup.webservice را تایپ کنید.
۸. Finish را کلیک کنید. با این کار فایل DukesAge.java در همان Package مشخص شده ایجاد می شود و برای ویرایش، در صفحه باز می شود.

تعریف کلاس DukesAge بعنوان یک وب سرویس

برای این کار عبارت @WebService را درست قبل از جایی که کلاس، تعریف شده قرار دهید. بدین صورت:

```
@WebService
public class DukesAge {
...
}
```

حذف Constructor پیش ساخته

از آنجا که وب سرویس نیاز به constructor ندارد. آن را از کلاس DukesAge حذف کنید.

افزودن متد getDukesAge

متدی به نام getDukesAge با خروجی int در داخل کلاس، ایجاد کنید و قبل از آن نیز عبارت @WebMethod را قرار دهید. بدین صورت:

```
@WebMethod
public int getDukesAge() {
}
```

سپس دستورات زیر را در بدنه متد کپی کنید.

```
Calendar dukesBirthday = new GregorianCalendar(۱۹۹۵,Calendar.MAY,۲۳);
Calendar now = Calendar.getInstance();
int dukesAge = now.get(Calendar.YEAR) - dukesBirthday.get(Calendar.YEAR);
dukesBirthday.add(Calendar.YEAR,dukesAge);
```

```
if (now.before(dukesBirthday)){
    dukesAge--;
}
return dukesAge;
```

رفع مشکل Import

۱. روی ادیتور کلیک راست نمایید .
۲. گزینه Fix Imports را انتخاب نمایید.
۳. پکیج javax.jws.WebService را برای کلاس WebService انتخاب نمایید.

Build و Deploy نمودن وب سرویس

از پنل پروژه ها firstcup-dukes-age را انتخاب نموده و کلیک راست کنید. سپس گزینه Build Project را انتخاب نمایید. با این کار پروژه در firstcup-dukes-age.war پکیج می شود. پس از پایان build مجدداً کلیک راست نمایید و این بار گزینه Deploy Project را انتخاب کنید.

ساخت firstcup Enterprise Application**ساخت پروژه**

۱. از منوی فایل New Project را انتخاب کنید.
۲. از Category ها Enterprise را انتخاب کنید.
۳. از پروژه ها Enterprise Application را انتخاب نمایید.
۴. Next را کلیک کنید.

۵. نام پروژه را firstcup قرار دهید.
۶. Location پروژه را مسیر فولدر myexample که قبلاً ساخته اید قرار دهید.
۷. بقیه تنظیمات موجود را تغییر ندهید و روی Finish کلیک کنید.

تنظیم Context Root

- این مرحله برای فراهم نمودن امکان شناسایی Web Application در J2EE server است.
۱. از قسمت پروژه ها firstcup را باز کنید
 ۲. روی گزینه application.xml از Configuration Files دبل کلیک نمایید.
 ۳. در این فایل عنصر context-root را بیابید.
 ۴. مقدار موجود را از firstcup_war به firstcup تغییر دهید و فایل را ذخیره کنید.
 ۵. در قسمت قسمت پروژه ها، روی firstcup-war کلیک راست کنید و properties را انتخاب نمایید.
 ۶. از category ها، Run را انتخاب کنید.
 ۷. مسیر Context را به firstcup/ تغییر دهید و OK را کلیک کنید.

ساخت Enterprise Bean

این Enterprise Bean یک Stateless session bean است. منظور این است که نیازی به تعامل با client و نگهداری state جاری ندارد. به عبارت دیگر درخواست های ارسالی به این Bean هیچ وابستگی به حالت قبلی و بعدی client ندارد.

این Enterprise Bean شامل دو کلاس است: یکی DukesBirthdayBean و دیگری DukesBirthdayRemote.

ساخت کلاس DukesBirthdayBean

۱. از قسمت پروژه ها firstcup-ejb را انتخاب کنید.
۲. از منوی فایل گزینه New File را انتخاب نمایید.
۳. از category ها Enterprise را انتخاب نمایید.
۴. از قسمت File Types گزینه Session Bean را انتخاب نمایید.
۵. Next را کلیک کنید.
۶. نام EJB را DukesBirthdayBean قرار دهید.
۷. نام Package را com.sun.firstcup.ejb وارد نمایید.
۸. Session Type را مساوی stateless قرار دهید.
۹. تیک فیلد Local را بردارید و Remote را تیک بزنید.
۱۰. Finish را کلیک کنید.

اصلاح کدهای DukesBirthdayBean.java

این کلاس حاوی کدهایی است که تفاوت سال سن Duke و کاربر را محاسبه می نماید.

۱. ابتدا Constructor پیش فرض را که NetBeans ایجاد کرده، حذف کنید.
۲. در درون بدنه کلاس، کدهای زیر را کپی نمایید.

```
private static Logger logger = Logger.getLogger("com.sun.firstcup.ejb.DukesBirthdayBean");
```

```
public int getAgeDifference(Date date){
    int ageDifference;
```

```
    Calendar theirBirthday = new GregorianCalendar();
    Calendar dukesBirthdate = new GregorianCalendar(۱۹۹۵,Calendar.MAY,۲۲);
```

```
    // Set the Calendar object to the passed in Date
    theirBirthday.setTime(date);
```

```
    // Subtract the user's age from Duke's age
    ageDifference = dukesBirthdate.get(Calendar.YEAR) - theirBirthday.get(Calendar.YEAR);
    logger.info("Raw ageDifference is: "+ageDifference);
    // Check to see if Duke's birthday occurs before the user's. If so,
    // subtract one from the age difference
    if (dukesBirthdate.before(theirBirthday) && (ageDifference > ۰)){
        ageDifference--;
```

```

}
logger.info("Final ageDifference is: "+ageDifference);

return ageDifference;
}

```

۳. روی ادیتور کلیک راست کنید و Fix Imports را انتخاب نمایید.
۴. java.util.date را برای کلاس date و java.util.logging.logger را برای کلاس logger انتخاب نمایید و OK را کلیک کنید.
۵. روی بدنه متد getAgeDifference کلیک راست کنید و از منوی EJB Methods گزینه Add to Remote Interface را انتخاب کنید.
۶. فایل را ذخیره کنید.

اصلاح کد DukesBirthdayRemote.java

این کلاس بعنوان رابط کلاس DukesBirthdayBean عمل می کند و اجزای Web Tier برای دسترسی به DukesBirthdayRemote از DukesBirthdayBean استفاده خواهند کرد. این کلاس توسط NetBeans ایجاد می شود. شما فقط باید تغییرات لازم را در آن اعمال کنید. بدین ترتیب:

۱. مازول firstcup-ejb را از قسمت پروژه ها باز کنید.
۲. روی DukesBirthdayRemote.java در پکیج com.sun.firstcup.ejb دبل کلیک کنید.
۳. پس از باز شدن فایل ، روی ادیتور کلیک راست کرده و گزینه Fix Imports را انتخاب کنید.
۴. برای کلاس Date گزینه java.util.Date را انتخاب کنید.
۵. در صورتی که دستور throws در تعریف متد وجود داشت، آن را حذف نمایید.
۶. فایل را ذخیره کنید.

با ساخت EJB پیاده سازی Business Tier از معماری پایان یافت. در ادامه به پیاده سازی مازول های Web Tier خواهیم پرداخت.

ساخت firstcup Web Client

- برای ساخت Web Client کارهای زیر را باید انجام دهید:
- تنظیم firstcup-war به این منظور که JSF را پشتیبانی نماید.
 - ساخت یک Web service client
 - ساخت یک resource bundle برای نگهداری پیغام هایی که توسط صفحات JSP استفاده می شوند.
 - تنظیم resource bundle در فایل configuration
 - ساخت کلاس مدیریت DukesBDay
 - تنظیم DukesBDay در فایل configuration
 - ساخت صفحه greeting.jsp
 - تنظیم قواعد navigation
 - ساخت صفحه response.jsp
- در ادامه، مراحل بالا یک به یک توضیح داده خواهند شد.

تنظیم firstcup-war جهت پشتیبالی JSF

همه Application های JSF باید شامل یک map به نمونه FacesServlet در فایل deployment descriptor باشند. نمونه FacesServlet درخواست ها را دریافت می کند و برای پردازش به life cycle می فرستد. همچنین مقداردهی اولیه منابع را انجام می دهد.

این mapping بدین صورت انجام می شود:

۱. روی مازول firstcup-war کلیک راست کنید و properties را انتخاب نمایید.
۲. از category ها framework را انتخاب نمایید.
۳. روی Add کلیک کنید.
۴. از لیست انتخاب ها، JavaServer Faces را انتخاب نمایید و OK را کلیک کنید.
۵. Servlet URL mapping را به /*/firstcupWeb/ تغییر دهید. این مسیر نمونه FacesServlet است.
۶. تیک Validate XML را بردارید.

۷. OK را کلیک کنید.

فوروارد صفحه `index.jsp` به `greeting.jsp`

می خواهیم بلافاصله پس از اجرای این Web Application به صفحه `greeting.jsp` برویم. لذا باید `index.jsp` را به `greeting.jsp` فوروارد نماییم. بدین صورت:

۱. ماژول `firstcup-war` را باز کنید و روی Web Pages دبل کلیک نمایید.
۲. روی `welcomeJSF.jsp` کلیک راست کرده بوسیله Delete آن را حذف نمایید.
۳. روی `index.jsp` دبل کلیک کنید.
۴. هر چیزی که داخل این صفحه نوشته شده است را پاک کنید.
۵. این عبارت را در صفحه `index.jsp` وارد کنید:
`<jsp:forward page="/firstcupWeb/greeting.jsp"/>`
۶. فایل را ذخیره کنید.

توضیح اینکه صفحه `greeting.jsp` را بعداً خواهیم ساخت.

ساخت Web Service Client برای وب ماژول `firstcup-war`

این ماژول از وب سرویس `firstcup-dukes-age` استفاده می نماید. به این منظور لازم است یک Web Service Client در آن ایجاد نمایید. بدین صورت:

۱. ماژول `firstcup-war` را انتخاب کنید.
۲. از منوی File گزینه New File را انتخاب کنید.
۳. از category ها گزینه Web Services را انتخاب کنید.
۴. از انواع فایل، Web Service Client را ایجاد نمایید.
۵. Next را کلیک کنید.
۶. WSDL URL را انتخاب کنید.
۷. در قسمت WSDL URL مقدار زیر را وارد کنید:
`http://localhost:۸۰۸۰/DukesAgeService/DukesAgeService?WSDL`
۸. در فیلد package نیز این مقدار را وارد نمایید:
`com.sun.firstcup.webservice`
۹. Finish را کلیک کنید.

ساخت Resource Bundle

منظور از Resource Bundle فایلی است از نوع properties که در آن کلیه متون ثابت، شامل پیغام های خطا و عباراتی که در صفحه `jsp` به کار می روند، تعریف می شود. نحوه ساخت آن بدین ترتیب است:

۱. ماژول `firstcup-war` را انتخاب نمایید.
۲. از منوی فایل گزینه New File را انتخاب کنید.
۳. category را other و نوع فایل را Properties File انتخاب کنید و Next را کلیک کنید.
۴. نام فایل را WebMessages وارد نمایید.
۵. فولدر را نیز این مسیر وارد کنید:
`src/java/com/sun/firstcup/web`
۶. Finish را کلیک نمایید.

۷. پس از اینکه NetBeans فایل را ایجاد نمود متن زیر را در آن کپی نمایید:

```
Welcome=Hi. My name is Duke. Let us find out who is older -- You or me
DukeIs=Duke is
YearsOldToday=years old today.
Instructions=Enter Your birthday and click submit.
YourBD=Your birthday
Pattern=MM/dd/yyyy
DateError=Please enter the date in the form MM/dd/yyyy.
YouAre=You are
Year=year
Years=years
Older=older than Duke!
Younger=younger than Duke!
SameAge= the same age as Duke!
Submit=Submit
Back=Back
```

۸. فایل را ذخیره کنید.

۹. در صورت نیاز می توانید برای هر زبانی مانند زبان فارسی یک Properties File مشابه بالا تهیه نمایید.
۱۰. توسط گزینه Refresh All Files از منوی File همه فایل ها را Refresh نمایید.

تنظیم و بیکر بندی Resource Bundle

۱. ماژول firstcup-war را باز کنید. از فولدر Web Pages فولدر WEB-INF را باز کنید.
۲. روی faces-config.xml دبل کلیک نمایید.
۳. قبل از نگ پایانی عنصر faces-config عناصر زیر را اضافه نمایید:

```
<application>
  <resource-bundle>
    <base-name>com.sun.firstcup.web.WebMessages</base-name>
    <var>bundle</var>
  </resource-bundle>
  <locale-config>
    <default-locale>en</default-locale>
    <supported-locale>es</supported-locale>
  </locale-config>
</application>
```

۴. فایل را ذخیره نمایید.

ساخت DukesBDay Managed Bean

افزودن Dependency روی Enterprise Bean Module

- ماژول firstcup-war به برخی از کلاس های firstcup-ejb وابسته است. شما باید وجود این وابستگی را به NetBeans اعلام نمایید. بدین ترتیب:
۱. روی ماژول firstcup_war کلیک راست نمایید و properties را انتخاب نمایید.
 ۲. از category ها Libraries را انتخاب نمایید.
 ۳. روی Add Project کلیک کنید.
 ۴. مسیر myexample/firstcup را انتخاب نمایید.
 ۵. پروژه firstcup-ejb را انتخاب کنید.
 ۶. دکمه Add Project JAR Files را کلیک کنید.
 ۷. توجه کنید که Build Required Projects تیک خورده باشد.
 ۸. OK را کلیک نمایید.

ساخت کلاس DukesBDay و تعریف متغیرهای لازم

DukesBDay یک Backing Bean است. منظور از Backing Bean، یک Bean کنترل شده (managed Bean) توسط JSF است که برای ذخیره سازی داده های مربوط به مقادیر کامپوننت های موجود در صفحه JSF به کار می رود.

DukesBDay در واقع واسط بین صفحات JSF و Business Tier است. صفحات JSF اطلاعات مورد نیاز خود را از این managed Bean دریافت می کنند و به آن نیز تحویل می دهند.

برای ساخت کلاس DukesBDay این مراحل را انجام دهید:

ساخت یک کلاس خالی:

۱. روی ماژول firstcup-war کلیک راست کنید.
۲. از New گزینه Java Class را انتخاب نمایید.
۳. نام کلاس را DukesBDay قرار دهید.
۴. Package را com.sun.firstcup.web وارد کنید.
۵. Finish را کلیک کنید.

افزودن Enterprise Bean Reference:

۶. درست بعد از سطر تعریف کلاس (اولین سطر از بدنه کلاس) متغیر private زیر را به همراه عبارت @EJB درج نمایید:

```
@EJB
Private DukesBirthdayRemote dukesBirthday;
```

افزودن property ها به Bean:

۷. پس از تعریف متغیر dukesBirthday متغیرهای زیر را بعد از آن درج نمایید:

```
private int age;
private Date YourBD;
private int ageDiff;
private int absAgeDiff;
```

۸. با افزودن کدهای زیر به constructor پیش ساخته، متغیرها را initialize نمایید:

```
age = -۱;
YourBD = null;
ageDiff = -۱;
absAgeDiff = -۱;
```

ساخت متدهای دستیابی (Accessor Methods) برای property ها

در این مرحله با استفاده از امکانات refactoring موجود در NetBeans متدهای get و set را ایجاد می کنیم. بدین ترتیب:

۱. روی ادیتور کلیک راست کنید.
۲. از منوی باز شده گزینه Refactor و سپس Encapsulate Fields را انتخاب نمایید.
۳. در فرمی که باز می شود، تیک مربوط به getDukesBirthday و setDukesBirthday را بردارید.
۴. Next را کلیک نمایید.
۵. در پنل refactoring روی Do Refactoring کلیک کنید. با این کار کدهای مربوط به get و set متغیرها به کلاس اضافه خواهند شد.

افزودن کدهای لازم جهت دریافت سن Duke

۱. مازول firstcup-war را باز کنید.
۲. قسمت Web Service References را باز کنید.
۳. وارد قسمت زیر شوید:

DukesAgeService->DukesAgeService->DukesAgePort->getDukesAge
 ۴. از ادیتور در فایل DukesBDay.java متد getAge را بیابید. سپس getDukesAge را که در مرحله قبل انتخاب نموده اید به متد getAge قبل از عبارت return age با ماوس drag نمایید. محتویات این متد پس از انجام این عمل چنین خواهد شد:

```
public int getAge() {
    try { // Call Web Service Operation
        com.sun.firstcup.webservice.DukesAgeService service =
            new com.sun.firstcup.webservice.DukesAgeService();
        com.sun.firstcup.webservice.DukesAge port = service.getDukesAgePort();
        // TODO process result here
        int result = port.getDukesAge();
        System.out.println("Result = "+result);
    } catch (Exception ex) {
        // TODO handle custom exceptions here
    }
    return age;
}
```

۵. در کدهای متد getAge خط

```
int result = port.getDukesAge();
```

را حذف کنید و به جایش عبارت زیر را قرار دهید.

```
age = port.getDukesAge();
```

۶. خط زیر را حذف نمایید:

```
System.out.println("Result = "+result);
```

افزودن کدهای دریافت تفاوت سن از DukesBirthdayBean

داخل متد getAgeDiff درست قبل از دستور return کدهای زیر را وارد نمایید:

```
ageDiff = dukesBirthday.getAgeDifference(YourBD);
if (ageDiff < ۰) {
    setAbsAgeDiff(Math.abs(ageDiff));
} else {
    setAbsAgeDiff(ageDiff);
}
```

افزودن Imort ها

۱. روی ادیتور کلیک راست کنید.
۲. گزینه Fix Imports را انتخاب کنید.

۳. عبارت `java.util.Date` را برای کلاس `Date` انتخاب نمایید.
 ۴. OK را کلیک کنید.

در خاتمه فایل را ذخیره نمایید.

تنظیم پیکربندی **DukesBDay**

تکنولوژی JSF امکان استفاده از فایل های Configuration را برای مقداردهی اولیه، پیکربندی و ذخیره سازی managed Bean ها در ۳ scope زیر می دهد:

- **Request** که وقتی کاربر صفحه ای را submit می کند آغاز می شود و پس از ارسال پاسخ پایان می پذیرد.
- **Session** که با اولین دسترسی کاربر به صفحه آغاز می شود و پس از خاتمه یافتن session کاربر، پایان می یابد.
- **Application** که تا زمانی که سرور، Application را متوقف نکرده اعتبار دارد.

Scope در managed Bean مثال ما session است. به محض اینکه Bean، پیکربندی شد یک صفحه JSF می تواند آن را ایجاد کند و به آن دسترسی داشته باشد. در اینجا ما پیکربندی managed Bean `DukesBDay` را انجام می دهیم:

۱. مازول `firstcup-war` را باز کنید.
۲. فولدر `Web Pages` و سپس `WEB-INF` را باز کنید.
۳. روی `faces-config.xml` دبل کلیک نمایید.
۴. روی ادیتور کلیک راست کرده و از `JavaServer Faces` گزینه `Add Managed Bean` را انتخاب کنید.
۵. نام Bean را `dukesBDay` وارد کنید.
۶. کلاس را `com.sun.firstcup.web.DukesBDay` قرار دهید.
۷. از منوی `scope` گزینه `session` را انتخاب کنید.
۸. در قسمت `Description` نیز عبارت `dukesBDay backing bean` را وارد کنید.
۹. `Add` را کلیک کنید. با اینکار عناصر XML مربوطه به فایل اضافه می شوند.
۱۰. فایل را ذخیره نمایید.

ساخت صفحه `greeting.jsp`

این صفحه شامل پیغام خوش آمدگویی، نمایش سن فعلی `Duke` و دریافت تاریخ تولد کاربر است. مراحل ساخت صفحه چنین است:

- ساخت یک صفحه خالی:
۱. روی مازول `firstcup-war` کلیک راست نمایید.
 ۲. از قسمت `New` گزینه `JSP` را انتخاب کنید.
 ۳. نام فایل `JSP` را `greeting` وارد کنید.
 ۴. `Finish` را کلیک کنید.
 ۵. تگ `head` را به این صورت تغییر دهید:

```
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-۸">
  <title>Firstcup Greeting Page</title>
</head>
```

۶. تگ های ابتدا و انتهای `body` را حذف کنید.
۷. عبارت زیر را حذف کنید:

```
<h۱>JSP Page</h۱>
```

۸. فایل را ذخیره نمایید.

Declare نمودن **Tag Library** های JSF

همانگونه که می دانید تکنولوژی `JavaServer Faces` در واقع مجموعه ای از custom tag های `jsp` است که برای نمایش کامپوننت های واسط کاربری، اعتبار سنجی و سایر موارد مربوطه به کار می رود. Custom tag ها در `tag library` ها تعریف می شوند. تکنولوژی `JavaServer Faces` در دو `tag library` تعریف می شود:

- `HTML Render Kit library` که در آن همه کامپوننت های استاندارد واسط کاربری تعریف شده اند.

- Core tag library که تگ های مورد نیاز برای مدل ها (convertors) ، اعتبارسنج ها (Validators) و event listener ها و سایر توابع عملکردی اشیاء تعریف می شوند.

برای استفاده از این تگ ها در صفحه باید tag library را declare نمایید. در این مثال باید خطوط زیر را در فایل greeting.jsp درست پس از تگ بسته شدن head قرار دهید:

```
<%@ taglib prefix="h" uri="http://java.sun.com/jsf/html" %>
<%@ taglib prefix="f" uri="http://java.sun.com/jsf/core" %>
```

افزودن تگ های f:view و f:form

یک صفحه JSF بوسیله درختی از کامپوننت های واسط کاربری نمایش داده می شود. هر صفحه JSP که از تکنولوژی JSF استفاده می کند باید دارای تگ view باشد. این تگ در Core tag Library تعریف شده است و ریشه یا root درخت کامپوننت ها را مشخص می نماید. سایر تگ های JSF باید درون تگ view تعریف شوند.

همچنین از آنجا که معمولاً صفحات JSF شامل فرمی برای دریافت اطلاعات از کاربر است، معمولاً باید شامل تگ form نیز باشد. همه کامپوننت های موجود در یک فرم باید درون تگ form قرار گیرند. یا مثال خود بحث را پی می گیریم.

در ادامه declare نمودن tag library ها تگ های view و form را بدین صورت تعریف نمایید:

```
<f:view>
  <h:form>
    </h:form>
</f:view>
```

افزودن Output Label ها برای نمایش محتوای فقط خواندنی

یکی از مهمترین و پر کاربردترین تگ های JSF تگ outputText است. greeting.jsp برای نمایش پیغام خوش آمدگویی، دستورالعمل ها، Label ها و سن Duke از آن استفاده می کند. برای این کار از پیغام های تعریف شده در resource bundle استفاده می نماید. همچنین تگ outputText ای که سن Duke را نشان می دهد از متغیر age موجود در DukesBDay bean استفاده می کند. در داخل تگ h:form این کدها را بنویسید:

```
<h:outputText value="#{bundle.Welcome}"/></h:outputText>
<h:outputText value="#{bundle.DukeIs}"/>
<h:outputText value="#{dukesBDay.age}"/>
<p><h:outputText value="#{bundle.Instructions}"/>
<p><h:outputText value="#{bundle.YourBD}"/>
```

افزودن یک کامپوننت input برای دریافت تاریخ تولد کاربر

یکی دیگر از تگ های پرکاربرد، تگ inputText است که یک فیلد متنی برای دریافت ورودی از کاربر است. در این مثال در صفحه greeting.jsp تاریخ تولد کاربر دریافت می شود و مقدارش در متغیر YourBD از DukesBDay Bean ریخته می شود. بدین ترتیب:

```
<h:inputText id="userBirthday" value="#{dukesBDay.yourBD}">
</h:inputText>
```

انتساب یک convertor به کامپوننت ورودی

JSF شامل مجموعه از convertor ها و validator های استاندارد است که می توان به کامپوننت نسبت داد. هدف از آنها اعتبارسنجی داده های ورودی و کنترل انطباق آنها با الگویی خاص است. در اینجا ما می خواهیم نحوه ورود تاریخ توسط کاربر را مشخص نماییم و روی آن کنترل داشته باشیم. به این منظور چنین عمل کنید:

۱. تگ زیر را در داخل تگ h:inputText اضافه نمایید:

```
<f:convertDateTime pattern="MM/dd/yyyy"/>
```

۲. برای نمایش الگوی ورود تاریخ به کاربر تگ زیر را پس از تگ بسته شدن h:inputText قرار دهید:

```
<h:outputText value="#{bundle.Pattern}"/>
```

۳. برای اینکه در صورت بروز خطا هنگام conversion پیغامی به کاربر نشان داده شود لازم است مشخصه converterMessage را به تگ h:inputText اضافه نمایید. این تگ بدین صورت خواهد شد:

```
<h:inputText id="userBirthday"
  value="#{dukesBDay.yourBD}"
```

```

        converterMessage="{bundle.DateError}">
        <f:convertDateTime pattern="MM/dd/yyyy"/>
    </h:inputText>

```

۴. برای مشخص کردن جای نمایش پیام خطا به کاربر تگ زیر را درست قبل از تگ بسته شدن h:form قرار دهید. بدین صورت:

```
<p><h:message for="userBirthday" style="color:red"/>
```

افزودن دکمه برای submit

درست قبل از تگ message که در مرحله قبل ایجاد کردید تگ زیر را اضافه نمایید:

```
<p><h:commandButton value="{bundle.Submit}" action="success"/>
```

در نهایت فایل را ذخیره نمایید.

تعریف Navigation صفحات JSF

این پروژه شامل دو صفحه jsp است که از تکنولوژی JSF در آنها استفاده شده. یکی greeting.jsp و دیگری response.jsp است. قواعد حرکت بین صفحات JSF باید در فایل configuration تعریف شود. این کار در مورد مثال جاری بدین صورت انجام می شود:

۱. مازول firstcup-war را باز کنید.
 ۲. فولدر Web Pages و سپس WEB-INF را باز کنید.
 ۳. روی faces-config.xml دبل کلیک نمایید.
 ۴. روی ادیتور کلیک راست کنید و از قسمت JavaServer Faces گزینه Add Navigation Rule را انتخاب نمایید.
 ۵. در فیلد Rule from view عبارت /greeting.jsp را وارد کنید.
 ۶. دکمه Add را کلیک کنید.
 ۷. مراحل ۴ تا ۶ را تکرار کنید. با این تفاوت که این بار عبارت /response.jsp را بعنوان Rule From view وارد کنید.
 ۸. روی ادیتور کلیک راست کنید و از JavaServer Faces گزینه Add Navigation Case را انتخاب کنید.
 ۹. فیلد From View را /greeting.jsp انتخاب نمایید.
 ۱۰. در فیلد From Outcome عبارت success را وارد کنید.
 ۱۱. فیلد To View را /response.jsp انتخاب نمایید.
 ۱۲. Add را کلیک کنید.
 ۱۳. مراحل ۸ تا ۱۲ را تکرار کنید. با این تفاوت که این بار /response.jsp را در From View و greeting.jsp را در To View وارد کنید.
- بدین ترتیب دو Navigation Rule خواهید داشت. یکی از greeting.jsp به response.jsp و دیگری برعکس. در پایان فایل را ذخیره نمایید.

ساخت صفحه response.jsp

۱. روی مازول firstcup-war کلیک راست نمایید.
۲. از قسمت New گزینه JSP را انتخاب کنید.
۳. نام فایل را response وارد کنید.
۴. Finish را کلیک کنید.
۵. در فایل response.jsp تگ head را تغییر دهید که اینگونه گردد:

```

<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-۸">
    <title>Response Page</title>
</head>

```
۶. تگ های ابتدا و انتهای body را پاک کنید.
۷. تگ <h1> را نیز به همراه تگ خاتمه و محتویاتش حذف نمایید.
۸. تگ های زیر را پس از تگ بسته شدن head قرار دهید.

```

<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>

```
۹. تگ های زیر را در ادامه تگ های مرحله قبل در صفحه کپی کنید:

```

<f:view>
<h:form>

```

```

<h2><h:outputText value="#{bundle.YouAre}" />
<h:outputText value="#{bundle.SameAge}" rendered="#{dukesBDay.ageDiff == 0}"/>

<h:outputText value="#{dukesBDay.absAgeDiff}" rendered="#{dukesBDay.ageDiff < 0}"/>
<h:outputText value="#{bundle.Year}" rendered="#{dukesBDay.ageDiff == -1}"/>
<h:outputText value="#{bundle.Years}" rendered="#{dukesBDay.ageDiff < -1}"/>
<h:outputText value="#{bundle.Younger}" rendered="#{dukesBDay.ageDiff < 0}"/>

<h:outputText value="#{dukesBDay.absAgeDiff}" rendered="#{dukesBDay.ageDiff > 0}"/>
<h:outputText value="#{bundle.Year}" rendered="#{dukesBDay.ageDiff == 1}"/>
<h:outputText value="#{bundle.Years}" rendered="#{dukesBDay.ageDiff > 1}"/>
<h:outputText value="#{bundle.Older}" rendered="#{dukesBDay.ageDiff > 0}"/>

</h:form>
</f:view>

```

۱۰. قبل از بسته شدن تگ h:form تگ زیر را قرار دهید:

```
<p><h:commandButton id="back" value="#{bundle.Back}" action="success"/>
```

۱۱. فایل را ذخیره نمایید.

Building, Packaging, Deploying & Running

در این مرحله شما DukesBirthdayBean و firstcup web client را Build خواهید نمود، آنها را در فایل های EAR بصورت package درخواست دهید آورد. فایل های EAR را در سرور deploy خواهید نمود و سرانجام Application را اجرا خواهید کرد.

آماده سازی Deployment Descriptor

در این مرحله برخی پارامترها که خود NetBeans در فایل Deployment Descriptor ایجاد کرده، اما ما به آنها نیاز نداریم را پاک خواهیم کرد. بدین ترتیب:

۱. ماژول firstcup-war را باز کنید.
۲. فولدر Configuration Files را باز کنید.
۳. روی web.xml دبل کلیک نمایید.
۴. در بالای پنجره ادیتور، روی تب General کلیک نمایید.
۵. علامت + کنار Context Parameters را انتخاب کنید. با این کار یک جدول باز می شود.
۶. از جدول مذکور تک تک پارامترها را انتخاب کنید و با دکمه remove همه آنها را یکی یکی پاک نمایید.
۷. در بالای ادیتور روی تب XML کلیک کنید.
۸. عنصر welcome-file-list و همه اجزای آن را حذف نمایید.
۹. فایل را ذخیره نمایید.

Packaging و Building

در این مرحله در مورد firstcup Enterprise Application عمل Building و packaging را انجام می دهیم. بدین ترتیب:

۱. firstcup را در قسمت پروژه ها انتخاب نمایید.
۲. کلیک راست کنید و گزینه Build Project را انتخاب کنید.

Deploying

در مرحله قبل شما DukesBirthdayBean enterprise bean و firstcup web client را بصورت فایل firstcup.ear پکیج نمودیم. حالا آن را روی سرور deploy می کنیم. بدین صورت:

۱. firstcup را در قسمت پروژه ها انتخاب کنید.
۲. کلیک راست کرده و گزینه Deploy Project را انتخاب کنید.

Running

۱. Browser را اجرا نمایید و در قسمت آدرس آن URL زیر را وارد کنید.
http://localhost:۸۰۸۰/firstcup

منابع:

<http://java.sun.com/javaee/5/docs/firstcup/doc/firstcup.pdf>
<http://java.sun.com/javaee/5/docs/tutorial/doc/JavaEETutorial.pdf>
http://en.wikipedia.org/wiki/Web_application

لطفا هرگونه نظر و پیشنهادی درباره این نوشته را به آدرس mahyar_e_vafa@yahoo.com ارسال نمایید.

مهیار ابراهیمی وفا
<http://javaborder.radmanitd.com>