

بسم الله الرحمن الرحيم

معماری عمومی وب و روش جاوا

برای تشریح بهتر برنامه‌های کاربردی تحت وب مدرن و امروزی، معمولاً رسم بر آن است که آن را به چهار لایه مستقل تفکیک می‌کنند. (شکل یک) لایه مشتری یا clien ، لایه‌های میانی که عبارتند از presentation و Business و در نهایت آخرین لایه یعنی لایه داد. در مدل‌های کوچکتر معمولاً لایه‌های میانی در یک لایه مجتمع شده و در نتیجه یک مدل ساده شده سه لایه‌ای به دست می‌آید برای تصور بهتر این لایه‌ها ، شاید بهتر بتاشد که فرض کنید هر لایه بر روی یک کامپیوتر پیاده سازی می‌شود . با این سناریو ، فقط لایه مشتری یا clien در منزل و محل کار من یا شما قرار گرفته است و بقیه لایه‌ها بر روی کامپیوترهایی واقع هستند که ممکن است صدها یا هزارها کیلومتر از ما فاصله داشته باشد . حتی ممکن است که لایه‌های دیگر نیز در کنار هم واقع شده باشند . (هر چند که معمولاً چنین نیست .) لایه مشتری در این مدل سازشی همان

مرورگر وب است که بر روی کامپیوتر مشتری ، صفحات وب را نشان می‌دهد . در اینن لایه علاوه بر آنکه اطلاعات نمایش داده می‌شوند ، امکان ورود اطلاعات از طرف کاربر نیز وجود دارد و یکی از وظایف مرورگرها ، انتقال این اطلاعات جمع آوری شده و ارسال آن به لایه‌های فوقانی است . معمولا برای پیاده

سازی لایه‌های مشتری از دو روش استفاده می‌شود .

در روش اول ، از هوشمند بودن کامپیوتر کاربر هیچ استفاده‌ای نمی‌شود . هنگامی که کاربر اطلاعات یک فرم را تکمیل می‌کند و یا در خواست صفحه خاصی را می‌کند (submit) . کلیه عملیات پردازشی در لایه‌های میانی انجام شده و صفحه جدیدی برای او ارسال می‌گردد . این صفحه جدید فقط ممکن است شامل یک پیام خطا باشد . به این روش اصطلاحاً Dump HTML client اطلاق می‌گردد . در روش دوم از هوشمندی کامپیوتر کاربر به خوبی استفاده می‌گردد . مثلا هنگامی که فرمی را در یک صفحه وب تکمیل می‌کنیم و اگر دچار اشتباه شویم ، بلافاصله بدون آنکه ارتباطی بین لایه‌های فوقانی برقرار گردد ، کامپیوتر به شما هشدار می‌دهد .

مزیت روش دوم (Semi intelligent client) آشکار است . (کاربر

برای هر عملیات ساده و ابتدایی نیاز به انتظار کشیدن در پای کامپیوتر خود ندارد .) اما در این روش ، مرز بین لایه‌های میانی (Business و Presentation) مخدوش می‌گردد برای استفاده از هوشمندی کامپیوتر مشتری ، از اسکریپت نویسی به زبان‌های جاوا اسکریپت و یا (DHTML) vbscript در صفحات وب استفاده می‌شود . توجه داشته باشید که به همین علت ، توانایی مرورگرها در اجرای دستورات ، اهمیت می‌یابد و نسخه‌های 4 به بعد مرورگرهای Netscape/IE ، امروزه حداقل نیازمندی سایت‌های جدید قرار گرفته‌اند .

لایه Presentation

وظیفه لایه Presentation ، ارائه محتوای متغیر (دینامیک) در صفحات وب است . معمولاً محتوای دینامیک از یک بانک اطلاعاتی تامین می‌شود (مانند اسامی کاربران Hotmail به همراه مشخصات آنان) . وظیفه مهم‌تر این لایه شناسایی صفحات ارسال شده از سمت مشتری (client) است . (یعنی شناسایی کاربر و انتقال اطلاعاتی که او به لایه Business وارد کرده است) . روش‌های پیاده سازی این عملیات بر روی کامپیوتر web server ، متنوع بوده و در گذشته از برنامه‌های موسوم به CGI استفاده

می‌شده‌است . الب“ه استفاده از جاوا اسکریپت در سمت سرور نیز مدتی

مورد استفاده قرار گرفت ، اما روش‌های متداول امروزی عبارتند از :

- استفاده از روش‌های میکروسافت بر اساس صفحات فعال سمت

سرور یا ASP با استفاده از Visual Interdev .

- استفاده از راه حل‌های شرکت سان بر اساس ترکیب صفحات فعال

جاوا (JSP) و موجودیت‌هایی به نام servlet ها .

در ابزارهای ارائه شده فوق ، روش‌هایی برای قرار دادن محتوای

دینامیک در درون صفحات HTML و همچنین تکنیک‌هایی برای دریافت

اطلاعات ورودی کاربر ، در نظر

گرفته شده‌اند .

لایه Presentation معمولاً درون برنامه‌های وب سرور (نظیر Apache webserver

Microsoft IISS IBM websphere , و غیره) پیاده سازی می‌شوند . وب سرورها

توانایی دریافت و اجراتی چندین درخواست از سوی برنامه‌های کاربردی

لایه‌های دیگر و همچنان کار بران را به طور همزمان دارند . وب سرورها

بر اساس آرایش اولیه (configuration) می‌دانند که درخواست‌های (requests)

کاربر مشتری را باید به کدام برنامه کاربردی (application) هدایت نمایند .

لیه Business

حجم زیاد از عملیات یک سایت مدرن در این لایه ، پیاده سازی می‌ود .

این عملیات عبارتند از :

- اجرای تمام محاسبات و تایید (validation) داده‌ها
- مدیریت جریان کار (workflow) . فرض کنید که شما به عنوان یک کاربر Hotmail با وارد کردن نام کاربری و اسم رمز وارد سایت شده‌اید ، تا زمانی که از سایت خارج نشده‌اید . سایت شما رامی‌شناسد و امکان استفاده از امکانات آن برای شما فراهم است . حضور شما تا لحظه خروج یک session یا جلسه نام دارد و مدیریت این جلسات نیز در لایه Business انجام می‌گردد .
- مدیریت دستیابی به اطلاعات (بانک اطلاعاتی) و ارائه آن به لایه

.Presentation

روش‌های پیاده سازی این لایه در کاربردهای جدید به یکی از صورتهای

زیر است :

- استفاده از اشیای COM (ارائه شده توسط مایکروسافت) با استفاده

از ویژوال بیسیک یا ویژوال c++ و غیره ،

- استفاده از اشیای متناظر شرکت سان یا (EJB (Enterprise java Beans.

اصولا هرگاه نیاز با ارتباط یا مبادله اطلاعات بین چند برنامه مستقل وجود

داشته باشد ، نام این تکنولوژی‌ها را خوتهد شنید . (در شمارات

گذشته مقالاتی در این مورد منتشر شده است) روشی که قبل از ارائه

راه حل های مایکروسافت و سان در این مورد به کار می رفته است ،

پیروی از مشخصه های اشیای CORBA بوده است . این مشخصه CORBA (

(در سال 1992 ارائه شده است و قابل حل در راه حل ارائه شده توسط

شرکت سان و زبان جاوا می باشد .

لایه Business معمولا درون برنامه هایی موسوم به (Application sever) پیاده

سازی می شوند (نظیر Microsoft MTS , Oracle Application server و غیره) .

اگر فعالیت هایی که در لایه های یک سایت کاربرد انجام می گیرند را به

امور یم منزل تشبیه کنیم ، کلیه فعالیت های خانه داری بر عهده

لایه Business و برنامه های Application

server خواهد بود .

برنامه‌های شرکت‌های تولید کننده Application server بر اساس قابلیت‌های امنیتی ، مقیاس پذیری ، قابلیت اطمینان و غیره از یکدیگر تفکیک می‌شوند .

لایه داده

این لایه مسئول نگهداری اطلاعات است و می‌تواند شامل بانک‌های اطلاعاتی مدرن یا مجموعه‌ای از فایل‌های متنی ساده باشد .
در کاربردهای ساده‌تر می‌توان مدل چهار لایه‌ای شرح داده شده را به مدل کوچکتری تبدیل کرد . در این مدل کوچک شده می‌توان لایه‌های Business و Presentation را در هم ترکیب کرده و بروی یک وب سرور اجرا نمود .

مروری بر معماری J2EE

(Java 2 Enterprise Edition)

در سال‌های اولیه پیدایش جاوا (1990) ، از این زبان بیشتر به عنوان یک زبان قوی برای برنامه نویسی در سمت مشتری client یاد می‌شد .

بهتدریج این زبان از اقبال عمومی در کاربردهای لایه‌های میانی برخوردار گردید و نهایتاً با عرضه راه‌های جدید (EJB و Servlets) ، شرکت سان (حدود سال 200) مشخصه‌های J2EE را معرفی نمود . مشخصه‌های J2EE نیز چهار لایه بوده و منطبق بر مدل معرفی شده « معماری چهار لایه وب » است . (شکل دو) در لایه مشتری ، صفحات وب که می‌توانند شامل جاوا اسکریپت باشند نمایش داده می‌شوند . در لایه میانی Presentation ، به کمک Servlet ها و JSP محتوای پینامیک تولید می‌شود . Servlet ها و JSP درون وب سرور اجرا می‌شوند . لایه Business با استفاده از تکنولوژی EJB پیاده سازی می‌شود . EJB ها درون Application server اجرا شده و همان تور که گفته شد وظایف خانه داری از قبیل انجام محاسبات ، تایید داده‌ها و دسترسی به بانک اطلاعاتی به یکی از روش‌های زیر انجام پذیر است :

استفاده از واسطه JDBC (مشابه ODBC مایکروسافت) و یا استفاده از SQLJ (مشابه SQL مایکروسافت) .

در مشخصه‌های عرضه شده در J2EE ، برای کامپیوترهای مشتری (client) غیر وابسته به وب نیز پیشبینی‌هایی انجام شده است . برای دسترسی

کامپیوترهای مشتری به لایه Business (مثلا تحت شبکه LAN که client IIOP نام گذاری شده اند) توابع دسترسی از راه دور نام RMI API عرضه شده است.

Servlet چیست ؟

CGI یکی از اولین روش هایی بود که برای تولید صفحات وب با محتوای دینامیک مورد استفاده قرار می گرفت . به مرور زمان روش های جدیدتری نیز برای این منظور ، عرضه شد . همانطور که ذکر شد ، مایکروسافت تکنیک ASP را عرضه نمود و پس از زمان کوتاهی JSP از طرف سان ارائه گشت . اما قبل از آن شرکت سان در سال 1997 ، Servlet ها را معرفی کرده بود .

Servlet ها به عنوان جایگزینی برای اسکریپت های CGI ، (درون ماشین مجازی جاوا یعنی JVM) Servlet ها برخلاف اپلت های جاوا نیازی به پشتیبانی مرورگر از جاوا ندارند و تقریبا بر روی تمام وب سروها (به صورت مستقیم و یا با استفاده از plug-in ها) قابل اجرا هستند . servlet ها قابلیت دریافت اطلاعات دریافت شده از کاربر و کوکی ها و غیره را

دارند . servlet ها قابلیت دسترسی مستقیم به بانکهای اطلاعاتی (در مدل های ساده معماری وب) و یا دسترسی به داده از طریق EJB (در مدل های کامل تر) را دارند .

به عنوان مثالی از یک servlet به قطعه کد - در کادر پایین - توجه نمایید .
این قطعه یک صفحه وب با عبارت "Welcome bake username" تولید می کند که در این عبارت username نام کاربر فعلی است .

JSP

همانطور که دیده می شود ، استفاده مکرر و اجباری از عبارت out.println() در servlet ها ، نمی توانست در برابر راه حل سر راست تر مایکروسافت (ASP) ایستادگی نماید . این موضوع شرکت سان را بر آن داشت تا در سال 1999 ، JSP را معرفی نماید . قطعه کد نشان داده شده در بخش دوم همان کار servlet نمونه ما را انجام می دهد و حجم آن بسیار کوچکتر است . واقعیت آن است که عبارت «%%» در JSP بر روی وب سرور به servlet ترجمه می شوند .

EJB

تا قبل از سال 1996 ، تکنولوژی جاوا محدود به اجرا در سمت مشتری
clint بوده است ، اما شرکت سان با معرفی قطعات کد قابل استفاده مکرر
به نام Java Bean در این سال همچنان با رقابت با مایکروسافت ادامه داد .
(در همان سال ها نیز مایکروسافت تکنولوژی COM را پایه گذار نمود)
یک Java Bean که رابط گرافیکی نداشته باشد می تواند کلیه وظایف لایه
Business را بر عهده گیرد .

این Java Bean می تواند از درون Servlet یا یک صفحه JSP فراخوانی شود .
اما با پیچیده تر شدن کاربردهای اینترنت و پیدایش مدل چهار لایه ، لازم
شد تا Java Bean ها بر روی کامپیوترهای جداگانه (Application server) اجرا
شوند . به این منظور سان EJB ها را عرضه کرد EJB ها برخلاف Java Bean
های معمولی همواره بدون واسطه گرافیکی هستند و می توانند از راه
دور به خدمت گرفته شوند . این برنامه ها ، برنامه هاب کاربردی توزیع
شده نام دارند .

سرزمین جاوا

یکی از دلایل شهرت شرکت سان مایکروسیستمز ارائه و توسعه پلافرم جاوا (Java Platform) است که در قلب آن زبان برنامه نویسی جاوا قرار دارد . مهمترین نکته ای که در ارتباط با جاوا باید دانست ، این است که جاوا تنها یک زبان برنامه نویسی مانند C++ نیست ، بلکه یک فن آوری کامل و جامع است که تقریباً برای هر مساله ای که در دنیای شبکه و اینترنت وجود دارد راه حل دارد . اغلب افرادی که نسبت به جاوا مبتدی هستند مرتکب همین اشتباه می شوند و چون زبان برنامه نویسی جاوا همچون زبان مادری خود C++ ، ”سبت به زبان هایی مثل بیسیک کمی پیچیده به نظر می رسد ، خیلی زود ممکن است از آن خسته و دلزده شوند ولی اگر به جاوا به چشم یک فن آور پیشرفته نگاه کنیم آنگاه زبان برنامه نویسی جاوا شیرین و جذاب به نظر خواهد رسید خصوصاً که زبان جاوا نسبت به فن آوری جاوا یک موضوع جنبی دیده می شود در حقیقت زبان جاوا ابزار عینیت بخشیدن به تفکر حاکم بر فن آوری جاوا و راه حل های آن است .

پروژه سبز

اوایل سال 1991 میلادی گروهی از متخصصان سان و پروژه‌ای به نام

Green Project را

آغاز کردند . هدف این پروژه که تا نهایی شده نتیجه آن مخفی ماند ،

یافتن پاسخ این سوال بود که «موج بعدی» جهان کامپیوتر (در آن سالها)

چه خواهد بود و چگونه می‌توان بر آن مسلط شد ؟ تیم نخستین پروژه

سبز متشکل از جمیز گاسلینگ ، مایم شریدان و پاتریک ناگتون خیلی زود

به این نتیجه رسیدند که دست کم یکی از تحولات مهم آینده آمیزش و

ترکیب شدن کامپیوترها و وسایل دیجیتالی مصرفی و خانگی (Digital

Consumer Devices) خواهد بود . به عبارت ساده تر تکنولوژی به سمتی خواهد

رفت که این دوگروه از ابزارها با هم تلفیق شوند و در نتیجه محصولات

جدیدی پدید آید که هم کامپیوتر باشند و هم مصارف عمومی و یا خانگی

داشته باشند . اکنون سوال این بود که اعضای گروه سبز باید چه

می‌کردند ؟ پاسخ را از روی همیشگی سان اقتباس کردند : « وسیله

جدیدی بسازیم و تخیلات خود را جامه عمل بپوشانیم » سر انجام پس از

حدود یک سال و اندی کار تلاش پیوسته یک وسیله دیجیتالی بر اساس

پردازنده SPARC ساخته شد ککه STAR7 (یا به اختصار 7*) نامیده شد Star 7

یک PDA دستی و بی سیم بود .

PDAها (Personal Digital Assistant یا دستیار دیجیتالی شخصی) ابزارهایی شبه

رایانه‌ای هستند که می‌توان آنان را چیزی بین ماشین حساب‌های

پیشرفته و رایانه‌های رومیزی دانست و معمولاً صفحه مانیتور کوچکی

دارند که ابعاد آن حدوداً به اندازه یک کف دست است . Star 7 در نوع خود

کم نظیر بود و پیشرفته ترین امکانات آن روز را داشت. تیم سازنده 7

اعضای گروه کوشیدند برای این محصول در بازار صنایع تلویزیونی و

ویدئویی جایی باز کنند اما در بازاریابی موفق نشدند . این ناکامی سبب

شد تا به دنبال بازارهای دیگری بگردند . « ایتترنت » چیزی بود که

ناگهان به اهمیت آن پی بردند . فن آوری استفاده شده در 7* و زبان Oak

به گونه‌ای بود مه می‌توانست برای انتقال محتوای صوتی و تصویری از

طریق ایتترنت مورد استفاده قرار گیرد . «بیعتا اولین چیزی که جلب توجه

می‌کرد صفحات وب و فایل‌های HTML بود به به این ترتیب زبان جاوا

بوجود آمد . هدف نخستین جاوا این بود که از طریق اپلت‌های جاوا ،

تصاویر گرافیکی و انیمیشن را در صفحات ایستا و بی روح HTML به

حرکت و پویایی وادار کند . مروگر HotJava که در سال 1994 معرفی گردید برای تحقق همین ایده ساخته شد و خیلی زود زبان و فن آوری جاوا محبوبیت پیدا کرد .

اوایل سال 1995 نخستین مجموعه کد سورس جاوا (نسخه 1.0a2) بر روی اینترنت قرار گرفت . بلافاصله پس از انتشار رایگان سورس جاوا در دنیا چیزی بالغ بر 10 هزار نفر برنامه نویس مشتاق آن را دریافت کردند و به این ترتیب انقلاب جاوا آغاز شد .

جاوانویس ، جاوایی فکر کن ، جاوابنوش !

مدیر عامل سان ، اسکات مک نیلی میگوید : « پلتفرم جاوا تمام API ها و زبان های برنامه نویسی دیگر را پشت سر گذاشت . امروزه جاوا زبان برنامه نویسی ، پلتفرم و معماری دنیای شبکه های رایانه ای است » در حقیقت جاوا جهانی است که با آن باید بر اساس تفکر و معماری جاوایی و با استفاده از ابزارهای برنامه نویسی جاوا برای محیط و پلتفرم جاوا فعالیت کرد .

جدول 1 به طور خلاصه عناصر کلیدی دنیای جاوا را نشان می دهد .

شکل 1 مربوط به گزارش شرکت IDC در مورد وضعیت بازار محصولات سان در زمینه اینترنت و شبکه است . به ادعای IDC ، 51 درصد بازار برنامه نویسی در اختیار جاوا و فن آوری مرتبط با آن است و چیزی حدود 2/5 میلیون برنامه نویس در سراسر جهان به کار با این فن آوری مشغولند .

این موضوع نشان دهنده آن است که با وجود حضور گسترده سایر فن آوری های مرتبط با اینترنت (از جمله پلتفرم میکروسافت و زبان ++C و VB) هنوز صنعت اطلاعات اعتماد بیشتری به پلتفرم جاوا دارد . جدیدترین نسخه پلتفرم جاوا (Java 2 Enterprise J2EE Edition) است که سورس کد آن (SDK) قابل اجرا بر روی پلتفرم ویندوز 98 میکروسافت نیست اولین بار در سال 1999 معرفی شد و کماکن محوری ترین بخش فن آوری جاوا است . از میان مهمترین شاخه های فن آوری جاوا میتوان به خود زبان جاوا اشاره کرد در این زمینه جاوا اشاره کرد در این زمینه جاوا از زبان رقب خود یعنی ++VC موفق تر است و سهم بیشتری را نصیب خود کرده است .

J++ محصول ناکامی است که شرکت مایکروسافت به تقلید از نسخه اصلی جاوا آن را پیاده سازی کرد و پس از ماجرای دادگاه مایکروسافت آن را ادامه نداد اما در عین حال زبان برنامه نویسی جدیدی را ابداع کرد به نام C# (سی شارپ) که بسیار شبیه زبان جاوا است و تقریباً همان قابلیت‌ها را دارد . C# هنوز خردسال است و بسیار زود است که در مورد آن قضاوت کنیم ولی بعید است بتواند در جذب برنامه نویسان و طرفداران جاوا موفقیتی بعه دست آورد ولی در مورد آن دسته از کاربران و برنامه نویسان که هنوز فن آوری مورد نظر خود را انتخاب نکرده اند یا بر روی فن آوری مایکروسافت کار می کنند ممکن است بازده مناسبی به دست آورند . Java Script ها (برنامه های کوتاه کامپایل نشده ای که اغلب در صفحات وب به کار می روند) هنوز در میا سازندگان صفحات وب طرفداران زیادی دارد و لب با گسترش طیف علاقه مندان به برنامه نویسی با زبان VB قسمت اعظم بازار HTML Spriting را توجه به این که 80 درصد بازار مرورگرها در اختیار IE است ، تصاحب کند . Java Applet ها در نبردر پایپای با رقیب اصلی خود Activex (پلاتفرم مایکروسافت) هنوز در چالش هستند . علت این موضوع کاملاً روشن

است : اپلت‌ها بر روی مرورگرها و یا سایر Client‌های استفاده شده توسط کاربران عادی عمل می‌کنند و چون مایکروسافت و سیستم عامل ویندوز و مرورگر IE در بازار و مربوط به کاربران نهایی (End Users)

(Users) هنوز تسلط دارند این چالش بسیار جدی است . اما دست کم پشت صحنه یعنی بازار سرویس دهنده‌های اینترنتی (Servers) وضعیت کاملاً بر عکس است با وجود موفقیت محصولات مایکروسافت در بازار ویندوز عملاً درصد کمتری از از گسترده اینترنت در اختیار این پلتفرم است . به عبارت روشن‌تر ، اگر عناصر برنامه نویسی همچون Java Beans را نوع مخصوص server از اشیائی مثل Java Applets در نظر بگیریم و همین رابطه بین ActiveX ها و اشیاء COM فرض کنیم آنگاه وضعیت بازار نشان می‌دهد که Java Beans ها از اشیاء COM جلوتر هستند و تقریباً 60 درصد بازار را در اختیار خود دارند جدول زیر مقایسه روشنی میان وضعیت این دو فن آوری موجود در بازار را نشان می‌دهد : (تعداد بیشتر علامت‌های ۷ نشان دهنده برتری بیشتر در بازار است)

همانطور که در دول مشاهده می‌شود EJB ها و JAXP نسبت به محصولات و فن آوری های مشابه رقیب برتری کامل دارند . دقیقاً به همین دلیل

است که وقتی وضعیت بازار نرم افزارها و راه حل های تجارت الکترونیکی را بررسی می کنیم مشاهده می کنیم که سهم عمده این بازار (خصوصاً در مورد XML) در اختیار پلاتفرم جاوا است. این موضوع سبب شده است که سایر شرکت های بزرگ همچون اوراکل و IBM بر روی فن آوری جاوا سرمایه گذاری کنند و صد البته باز بودن (Open Platform) فن آوری جاوا و انحصاری نبودن آن در این مورد بی تاثیر نیست. در انتهای همین گزارش فهرست نسبتاً جامعی از مهم ترین مراجع اینترنتی در مورد جاوا آمده است که اطلاعاتی در مورد شرکت های اوراکل و آی بی ام و محصولات مربوطه آنها نیز در بردارد. اوراکل، یکی از مهمترین رقبای مایکروسافت در بازار نرم افزارهای سرویس دهنده اینترنت و پایگاه های اطلاعاتی (Database) سرمایه گذاری عظیمی بر روی جاوا انجام داده است که همگی تحت عنوان Oracle Jdeveloper قرار گرفته اند.

JDBC (معادل ODBC در ویندوز) و SQLJ عناصر اصلی برنامه نویسی برای پایگاه های داده در پلاتفرم جاوا هستند البته در بازار اینترنت یک استثنا بزرگ نیز وجود دارد: ASP ها در پلاتفرم مایکروسافت هنوز

محدودیت غیر قابل انکاری نسبت به JSP ها دارند و آینده Java Server (JSP Pages) بدون یاری گرفتن از وضعیت مطلوب EJB چندان روشن و امیدوار کننده به نظر نمی‌رسد البته JSP نوپاست و برخی در ازار نسبت به آینده آن امیدوارند . هر سال یک همایش عظیم توسط شرکت سان مایکروسیستمز برگزار می‌شود به نام JavaOne Conference که رویدادی مهم در صنعت اینترنت به شمار می‌رود و معمولاً مهمترین و معتبرترین اخبار ، تحولات و گزارشات مربوط به فن آوری جاوا در زمان برگزاری این همایش در طول سالهلی اخیر ، سومین سال تولید جاوا یکی از خاطره انگیزترین همایش‌ها بوده است .

15 هزار نفر در این کنفرانتس تاریخی شرکت کردند که در کل تاریخ کامپیوتر بی سابقه بود . امروزه برخی تحولات جدید در وضعیت بازار نوید موفقیت‌های بیشتری را برای فن آوری جاوا می‌دهد . به عنوان مثال اخیراً شرکت مشهور Macromedia شرکت Allaire و محصولات آن Jrun و Kawa را خریداری کرده است . این اقدام شرکت ماکرو مدیا که تمام محصولاتش از محبوبیت خاصی برخوردار است نشان دهنده ایناست که امیدواری بازار نسبت به پیروزی این نوع محصولات افزایش یافته است

علاقه‌مندان به ساختن صفحات JSP می‌توانند جدیدترین نسخه نرم افزارهای Jrun و Kawa را از سایت ماکرو مدیا دریافت کنند .

صفحات JAVAServer

صفحات JAVAServer روش ساده تولید صفحات وب هستند که محتوای پویا خدمت می‌دهند و می‌توانند با ابزارهای تالیف HTML و XML ایجاد شوند .

روش‌های متعددی برای ساخت یک برنامه کاربردی وب برای سرویس دادن محتوای پویا وجود دارد . اما بیشتر برنامه نویسان صفحات JAVAServe را ترجیح می‌دهند و دلیل آن سادگی و ساختار دستوری کد نویسی صریح می‌باشد که بوسیله توان زیربای جاوا توسعه یافته‌است .

فن آوری صفحات JAVAServe استاندارد است که توسط سان میکروسیستمز به ینوان روشی برای تولید محتوای پویا در صفحات خروجی یک برنامه کاربردی وب مشخص شده‌است . JSP ها با فن آوری Serve جاوا آمیخته شده‌اند . آنها به ما اجازه می‌دهند تا قطعات کد جاوا را

داشته باشیم و اجزاء خارجی را در کد HTML (یا کد markup دیگری همچون XML) فراخوانی کنیم هر چیزی که در یک صفحه HTML طبیعی وجود دارد، می‌تواند در یک صفحه JSP نیز باشد. می‌توان از کد JSP در اسکریپت نویسی سمت سرور وب به صورت JavaScript یا Perl یا صفحات سرویس دهنده فعال (ASPها) استفاده کرد. به دلیل مبتنی بر جاوا بودن، صفحات JAVAServe دستیابی کامل به API های شیء گرا و چارچوب کاری قابل حمل جاوا را فراهم می‌کنند.

صفحات JSP پیش از اجرا به یک Serve جاوا ترجمه می‌شوند. و درخواست‌های HTTP را پردازش می‌کنند و پاسخ‌هایی را همانند هر Serve دیگری تولید می‌کنند. فن آوری JSP روش راحت تری را برای کد نویسی Serve ارائه می‌کند. بنابراین، صفحات JSP می‌توانند دارای خروجی از یک Serve باشند و یا خروجی را به یک Serve بفرستند. و Serve ها می‌توانند دارای خروجی از صفحات JSP باشند یا خروجی را به صفحات JSP بفرستند. این دو فن آوری مکمل یکدیگر هستند و انتخابی جالب نسبت به دیگر انواع برنامه نویسی پویای وب هستند.

مزایای دیگر JSP ها و زیربنای جاوا عبارت است از :

❖ جداسازی منطق تجاری پویا از طرح کلی صفحه ایستا

❖ ساختار دستوری راحت

❖ استقلال زیربنا

❖ دستیابی به API های استاندارد جاوا

❖ سهولت استفاده

ساختار دستوری جاوا

در یک سناریو مناسب برای یک برنامه کاربردی بانک اطلاعاتی صفحات JSP جزئی از قبیله JavaBean یا Enterprise JavaBean را فراخوانی می کنند و Bean عموماً از طریق API های استاندارد از قبیل JDBC یا SQLJ به پایگاه داده دستیابی خواهند داشت . کد زیر مثالی از یک JSP است که به پایگاه داده دستیابی ندارد . اما ساختار دستوری اسامی را به شما نشان می دهد پیام های JSP در ساختار دستوری <%..%> قرار می گیرند .

در این مثال . قطعات جاوا نام کاربر را از یک شیء در خواست HTTP میگیرند . نام کاربر را چاپ می کنند و تاریخ جاری را دریافت می کنند خروجی زیر در صورتی توسط صفحه JSP تولید می شود که کاربر "Amy" را وارد کند .

عناوین و دستورهای ساختار دستوری JSP سطح بالا عبارتند از :

❖ رهنمودها ، که بیانگر اطلاعات با توجه به صفحه JSP به عنوان یک کل هستند .

❖ عناصر اسکریپت نویسی . که عناصر کد نویسی جاوا از قبیل معرفی ها . عبارات ، Scriptlet ها ، و توضیحات هستند (هر یک با ساختار دستوری متفاوت ، از قبیل `<%=...%>` برای عبارات یا `<%....%>` برای معرفی ها)

اشیاء یا حوزه ها داده شده . از قبیل صفحات JSP یا جلسه HTTP قابل دسترسی هستند .

❖ اعمال ، که اشیاء را ایجاد می کنند تا بر جریان خروجی در پاسخ JSP تاثیر گذارند .

کدنویسی JSP در مقابل کد نویسی Sevelt

ترکیب کد جاوا و فرا خوانی‌های جاوا در یک صفحه HTML استفاده از کد جاوا در یک Servlet است . ساختار دستوری JPS ,یانبری برای کد نویسی صفحات پویای وب به شما می‌دهد . که نیاز به کد کمتری نسبت به ساختار دستوری Servlet جاوا دارد . در زیر مثالی از کد Servlet معادل با مثال قبلی صفحات JSP داریم .

ساختار دستوری JSP ساده‌تر است (سر بار جاوا را همانند صدور بسته‌ها و بلوک‌های try...catch ذخیره می‌کند) . مترجم JSP نیز به طور خودکار مقدار کافی از سر بار کد نویسی Servlet را در فایل جاوایی که خروجی است ، مدیریت می‌کند . از قبیل پیاده سازی مستقیم یا غیرمستقیم رابط javax.servlet.jsp.HTTPjspPage استاندارد و افزودن کد برای دستیابی به یک جلسه HTTP .

همچنین بایستس ذکر شود که به این دلیل که متن HTML و XML صفحات JSP در عبارات چاپی جاوا تعبیه نشده‌اند . می‌توان از

ابزارهای تالیف HTML و XML برای ایجاد صفحات JSP استفاده کرد .

جداسازی منطق تجاری از ارائه صفحه

یکی از بزرگترین منافع فناوری JSP این است که امکان جداسازی تلاش‌های توسعه بین کد HTML را فراهم می‌کند که این کد ارائه صفحه ایستا و کد جاوا را که منطق تجاری را پردازش می‌کند و محتوای پویا را تولید می‌کند . تعیین می‌نمایند . تقسیم مسئولیت‌های نگهداری بین ارائه و متخصصین طراح که ممکن است متخصص جاوا باشند نه HTML ، و متخصصین کدی که ممکن است در جاوا متخصص باشند نه در HTML ، کار بسیار خوبی است .

در صفحات JSP مناسب ، بیشتر کد جاوا و منطق تجاری در قطعات تعبیه شده در صفحه قرار نخواهد گرفت ، این منطق را در JavaBeans یا JavaBeansEnterprise خواهید یافت که از صحنه در

خواست می‌ود مشخصات JSP ساختار دستوری زیر را برای ایجاد

یک نمونه از کلاس JavaBeans ارائه می‌دهد :

این مثال نمونه‌ای از mybeans ایجاد می‌کند (pagBean) حوزه (

قابلیت دسترسی) PageBean می‌تواند هر صفحه ، درخواست .

جلسه یا برنامه‌ای کاربردی باشد . سپس در این صفحه می‌توانیم

از این نمونه Bean استفاده کنیم:

این نمونه Bean مثلاً "Hello Julie" را چاپ می‌کند . اگر نام "Julie"

در ویژگی NewName ، PageBean باشد .

برنامه نویس جاوا مالک کد است و آن را برای کلاس NameBean

نگهداری می‌کند . خبر HTML مسئول ارائه ایستا طرح کلی صفحه

وب است که کاربر برنامه می‌بیند ، و همچنین مالک کد در فایل

JSP است و آن را نگهداری می‌کند .

اجرای مختصر JSP

ماژول برنامه‌ای که صفحات JSP را ترجمه ، اجرا و پردازش می‌کند

و در خواست‌ها را به آنها تحویل می‌دهد ، تحت کانتینر JSP شناخته

می‌شود . ساختار دقیق یک کانتینر JSP در هر پیاده سازی فروش متفاوت است . اما همیشه حاوی یک Servlet یا مجموعه‌ای از Servlet ها می‌باشد . کانتینر JSP در یک کانتینر Servlet اجرا می‌شود . هر کانتینر JSP ممکن است در یک سرویس دهنده وب شرکت کند . اگر سرویس دهنده وب به جاوا نوشته شده باشد ، یا کانتینر ممکن است مرتبط به سرویس دهنده وب اشد و توسط آن مورد استفاده قرار گیرد .

صفحات JSP توسط کانتینر JSP به عنوان نتیجه یک توسعه فایل jsp URL در خواست می‌شوند . هنگامی که صفحه‌ای برای اولین بار اجرا می‌شود به کد Servlet ترجمه شده و سپس کامپایل می‌شود .

کتابخانه برچسبهای JSP

علاوه بر برچسب‌های JSP استاندارد ، مشخصات jsp به فروشندگان و مشتریان اجازه می‌دهد تا کتابخانه برچسب‌های خودشان را با پیاده سازی یک چارچوب کاری کتابخانه برچسب استاندارد تعریف کنند .

هر برچسب کتابخانه مجموعه‌ای از برچسب‌های سفارشی است و می‌تواند به عنوان زیرزبان‌یاز jsp در نظر گرفته شود . برنامه نویسان می‌توانند از کتابخانه برچسب‌ها در هنگام کد نویسی دستی صفحات jsp به طور مستقیم استفاده کنند ، و این کتابخانه‌ها نیز می‌توانند توسط ابزارهای برنامه نویسی جاوا مورد استفاده قرار گیرند. هر برچسب کتابخانه‌ای بایستی بین پیاده سازی‌های مختلف

کاتتینر JSP قابل حمل باشد .

پیاده سازی یک کتابخانه برچسب شامل ایجاد یک کلای مدیر برچسب برای هر چسب ، و یک فایل توصیف کتابخانه برچسب (TLD) برای کتابخانه به عنوان یک کل است . TLD حاوی اطلاعاتی از قبیل نام کلاس‌های مدیر برچسب می‌باشد .

رهنمودهای آتی

سان مایکروسیستمز مشخصات JSP1.2 و Servlet 2.3 را به انجام

رسانده است (مشخصات JSP جاری ، نگارش 1/1 . بر اساس

مشخصات Servlet22

است) بهنگام سازی‌های JSP 1.2 شامل نگاشت از صفحات JSP به

یک سند XML و اصلاحات برای حمایت زمان اجرا برای کتابخانه

برچسب‌های سفارشی است . مشخصات jsp تاکید بیشتری بر

کتابخانه برچسب‌ها دارد . علاوه بر این . javasoft سان یک برنامه

ریزی کتابخانه برچسب استاندارد است .

1. یک برنامه جاوا کوچک که در یک سرویس دهنده اجرا می‌شود .

این برنامه ها وظیفه انجام خدمات سبک وب را بر عهده دارند ،

مثلا تغییر مسیر یک کاربر از یک نشانی قدیمی به صفحه درسی

که به وسیله برنامه‌های کاربردی CGI مدیریت می‌شوند سرعت

این برنامه‌ها بسیار بالاست و باری سیستم را کاهش می‌دهد .