



بررسی معماری Applet-to-Servlet در جاوا

در این مقاله پس از معرفی Applet ها و Servlet ها، چندین معماری Applet-to-Servlet بررسی می‌شود. در طراحی برنامه‌های مبتنی بر وب و چند لایه می‌توان از روش Applet-to-Servlet استفاده کرد. اپلت‌ها یک مکانیزم مناسب جهت تولید رابط‌های گرافیکی قدرتمند و پویا هستند و از طرف دیگر Servlet ها نیز یک راه حل کارآمد برای کنترل درخواست‌ها در یک سرویس دهنده وب (Web Server) یا سرویس دهنده برنامه‌های کاربردی (Application Server) هستند. در مدل‌های برنامه‌نویسی شرکت سان که بهترین راه حل‌های تولید برنامه‌های بزرگ جاوا را بر پایه پلاتفرم Java 2 ارائه می‌کند، استفاده از اپلت‌ها، HTML و Java Server Page در سمت سرویس گیرنده و Servlet ها همراه EJB (Enterprise Java Beans) یا سایر Component های کمکی در سمت سرویس دهنده توصیه می‌شود.

اپلت‌ها:

اپلت‌ها برنامه‌های جاوا هستند که در داخل سرویس دهنده وب اجرا می‌شوند و از کلاس `java.applet.Applet` یا `javax.swing.JApplet` ارث می‌گیرند. در مقایسه با صفحات HTML، اپلت‌ها می‌توانند رابط کاربر قدرتمندتر و پویاتری تولید نمایند. از اپلت‌ها می‌توان در برنامه‌های بزرگ برای مشاهده یا تغییر اطلاعات ارسالی از سمت سرویس دهنده استفاده کرد. برای مثال از یک اپلت می‌توان برای مشاهده یا تغییر رکوردهای بانک اطلاعاتی یا کنترل وضعیت سرویس دهنده یا برنامه‌های در حال اجرای روی آن استفاده کرد.

یک اپلت می‌تواند دارای یک کلاس منفرد یا چندین کلاس باشد. برای تسریع در انتقال کلاس‌ها به سرویس گیرنده از حالت فشرده کلاس‌ها در قالب JAR (Java Archive File) می‌توان استفاده کرد. در نتیجه کد اپلت هر زمان که در سمت سرویس دهنده تغییر کند، با اولین مراجعه مجدد کاربر به سایت بروزرسانی می‌شود.

از آنجاییکه اپلت‌ها در سمت سرویس گیرنده اجرا می‌شوند، یکسری تدابیر امنیتی برای اپلت‌ها در نظر گرفته شده است که نواقص و ایرادهای Microsoft ActiveX را برطرف می‌کند. از جمله محدودیت‌های اپلت‌ها این است که به منابع سرویس گیرنده دسترسی ندارند و در روی سرویس گیرنده نمی‌توانند فایل‌ها را بخوانند یا بنویسند و یا برنامه‌ای را اجرا نمایند.

Servlet ها:



Ahoo Engineering Group

Java Servletها اجزاء نرم‌افزاری سمت سرویس دهنده هستند که مشابه برنامه‌های CGI کنترل درخواستهای وب، برگرداندن اطلاعات پویا برای سرویس گیرنده، دسترسی به بانک اطلاعاتی و ارتباط با سایر اجزاء نرم‌افزاری موجود روی سرویس دهنده را انجام می‌دهند. بر خلاف CGI که به ازای هر درخواست سرویس گیرنده یک کپی از برنامه در سمت سرویس دهنده بارگذاری و اجرا می‌شود، Servletها تنها یکبار در سرویس دهنده بارگذاری و به ازای هر درخواست سرویس گیرنده فراخوانی می‌شوند

معماری‌های موجود

چندین معماری برای روش Applet-to-Servlet وجود دارد که در ادامه سه مورد از آن بررسی می‌شود.

معماری دو لایه (Two-Tier)

در این معماری تنها از اپلت در سمت سرویس گیرنده استفاده می‌شود و اپلتها از طریق پروتکل‌هایی مانند JDBC و (Remote Method Invocation) RMI به منابع سیستم -بانکهای اطلاعاتی، دایرکتوریهای Light LDAP (Weight Directory Access Protocol) و EJB Component - مستقیماً دسترسی پیدا می‌کنند.

اگر چه این معماری ساده به نظر می‌رسد، اما دارای مشکلات و معایبی است که عبارتند از:

در این روش تمام اطلاعات دسترسی به سرویس دهنده و منابع آن شامل شناسه کاربر، رمز عبور، مشخصات بانک اطلاعاتی، مشخصات سرویس دهنده و هر چیز دیگری بایستی در داخل کد اپلت تعریف شود. در نتیجه احتمال دسترسی کاربر به این اطلاعات وجود دارد. (برای مثال استفاده از یک De-Compiler) بانک اطلاعاتی یا هر سیستمی که اپلت به آن نیاز دارد روی همان ماشین سرویس دهنده وب قرار می‌گیرد و در نتیجه سرویس دهنده همزمان بعنوان سرویس دهنده بانک اطلاعاتی و سرویس دهنده وب عمل می‌کند. در اینصورت اگر دسترسی به یکسری از اطلاعات توسط Firewall محدود شده باشد، اپلتها نمی‌توانند به این منابع دسترسی پیدا کنند.

مکانیزم Clustering و Pooling روی سرویس دهنده بسیار مشکل و حتی ناممکن می‌باشد.

معماری سه لایه (Three-Tier)

یک راه حل بهتر نسبت به معماری دو لایه، استفاده از Servlet برای کپسوله کردن ارتباط با منابع سرویس دهنده می‌باشد. در این معماری Servlet مشکلات امنیتی حاصل از محدودیت‌های اپلت را برطرف می‌کند و همچنین دسترسی اپلتها را به Business Logic و Enterprise Information Systems کنترل می‌کند. در این روش اپلت درخواست را به Servlet می‌فرستد و Servlet پس از برقراری ارتباط با بانک اطلاعاتی یا هر منبع مورد نیاز دیگر و انجام محاسبات، نتیجه را برای اپلت ارسال می‌نماید.



مزایا و معایب این معماری عبارتند از:

در این معماری لایه‌های سیستم می‌توانند روی چندین سرورس دهنده توزیع شوند و تمام لایه‌ها با یک بانک اطلاعاتی واحد ارتباط برقرار می‌کنند. استفاده از این معماری باعث قابلیت گسترش (Scalability) و ماژوله بودن سیستم می‌شود. قرار گرفتن لایه منطق برنامه در Servlet که پیچیدگی برنامه را افزایش داده و قابلیت حمل (Portability) را کاهش می‌دهد.

معماری چند لایه (N-Tier)

اگر از تکنولوژی Enterprise Java Beans استفاده شود، Servlet بعنوان یک میانجی عمل می‌کند. استفاده از EJB باعث حذف Business Logic و انتقال آن به EJB می‌شود. در این روش اپلت با Servlet تماس می‌گیرد و Servlet اطلاعات را به EJB می‌فرستد و بالعکس. استفاده از HTML، Servlet، EJB و اپلت حداکثر انعطاف پذیری و کارایی را در کنار پیچیدگی و هزینه بیشتر به همراه دارد.