

Recherchebericht

Manche Menschen kommen in ein dunkles Zimmer und beginnen emsig zu arbeiten. Sie ergründen die Ursachen der Dunkelheit, finden Schuldige und erstellen ein mittelfristiges Konzept zur schrittweisen Reduzierung der Finsternis. Und dann kommt einer und macht einfach das Licht an.
– Peter Hohl

Inhaltsverzeichnis

1	Begriffe	2
1.1	Widget	2
1.2	Portal	2
1.3	Framework	2
1.4	Object-Relational-Mapping (ORM, z.B. Hibernate)	2
1.5	AJAX (Asynchronous JavaScript and XML)	2
1.6	Servlet	3
1.7	Servlet-Container (z.B. Apache Tomcat)	3
1.8	Web Archive bzw. Web Application Archive	3
1.9	Smart Client	3
1.10	GWT (Google Web Toolkit) / SmartGWT	3
2	Konzepte	4
2.1	Client-Server-Modell	4
2.2	Servlets-Architektur	4
2.3	Google Web Toolkit	5
3	Beschreibung der zu studierenden Applikation	6
3.1	GWT (inkl. SmartGWT) im Anwendungsfall	6
3.2	Innerhalb der GWT-Umgebung:	7
3.3	Außerhalb der GWT-Umgebung:	8

1 Begriffe

1.1 Widget

... ist ein kleines Computerprogramm, das nicht als eigenständige Anwendung betrieben, sondern in eine grafische Benutzeroberfläche oder Webseite (Web-Widget) eingebunden wird. Meist handelt es sich um Hilfs- bzw. Dienstprogramme oder Tools.

1.2 Portal

... ist eine Applikation, die einen zentralen Zugriff auf personalisierte Inhalte sowie bedarfsgerecht auf Prozesse bereitstellt. Charakterisierend für Portale ist die Verknüpfung und der Datenaustausch zwischen heterogenen Anwendungen über eine Portalplattform. Eine manuelle Anmeldung an den in das Portal integrierten Anwendungen ist durch Single-Sign-On nicht mehr notwendig, denn es gibt einen zentralen Zugriff über eine homogene Benutzungsoberfläche. Portale bieten die Möglichkeit, Prozesse und Zusammenarbeit innerhalb heterogener Gruppen zu unterstützen.

1.3 Framework

... beschreibt ein Programmiergerüst, das in der Softwaretechnik, insbesondere im Rahmen der objektorientierten Softwareentwicklung sowie bei komponentenbasierten Entwicklungsansätzen, verwendet wird. Ein Framework ist selbst kein fertiges Programm, sondern stellt den Rahmen, in welchem der Programmierer eine Anwendung erstellt, zur Verfügung, wobei u.a. durch die in dem Framework verwendeten Entwurfsmuster auch noch die Struktur der individuellen Anwendung beeinflusst wird.

1.4 Object-Relational-Mapping (ORM, z.B. Hibernate)

... ist eine Technik der Softwareentwicklung, mit der ein in einer objektorientierten Programmiersprache geschriebenes Anwendungsprogramm seine Objekte in einer relationalen Datenbank ablegen kann. Dem Programm erscheint diese Datenbank dann als objektorientierte Datenbank, wodurch der Impedance Mismatch beseitigt und so die Programmierung erleichtert wird.

Hibernate ist ein Open-Source-Persistenz-Framework für Java, welches diese Technik implementiert.

1.5 AJAX (Asynchronous JavaScript and XML)

... beschreibt ein Konzept der asynchronen Datenübertragung zwischen einem Server und dem Browser, das es ermöglicht, innerhalb einer HTML-Seite eine HTTP-Anfrage durchzuführen, ohne die Seite komplett neu laden zu müssen.

Das eigentliche Novum besteht in der Tatsache, dass nur gewisse Teile einer HTML-Seite oder auch reine Nutzdaten sukzessiv bei Bedarf nachgeladen werden, womit AJAX eine Schlüsseltechnik zur Realisierung des Web 2.0 darstellt.

1.6 Servlet

... bezeichnet eine Java-Klasse, deren Instanz innerhalb eines Servlet-Containers Anfragen von Clients entgegen nehmen und dynamisch Antworten erzeugen kann.

Servlets sind die javabasierte Lösung zur Erstellung von dynamischen Webinhalten und somit eine Antwort auf Technologien wie CGI, PHP und ASP.NET.

1.7 Servlet-Container (z.B. Apache Tomcat)

... ist ein spezialisierter Web-Server, der die Ausführung von Servlets unterstützt. Er vereint die grundlegenden Funktionen eines Web-Servers mit Java/Servlet spezifischen Optimierungen und Erweiterungen, wie beispielsweise einem integrierten Java Runtime Environment und der Fähigkeit, bestimmte URLs in Servlet-Anfragen zu übersetzen. Apache Tomcat ist eine in Java entwickelte Open-Source-Implementierung eines Servlet-Containers.

1.8 Web Archive bzw. Web Application Archive

... ist ein Dateiformat, welches beschreibt, wie eine vollständige Webanwendung nach der Java-Servlet-Spezifikation in eine Datei im JAR-bzw. ZIP-Format verpackt wird. Solche Dateien haben immer die Endung „*.war“ und werden daher auch „WAR-File“ genannt.

1.9 Smart Client

... ist in Abgrenzung zum Thin Client, Fat Client und Rich Client ein Programm, welches lokal auf dem PC läuft, aber vollautomatisch über das Internet oder Intranet installiert und aktualisiert wird. Der Smart-Client-Ansatz verspricht durch besseres Laufzeitverhalten und höheren Bedienkomfort bei gleichzeitiger Vermeidung lokaler Installationsprobleme eine Kombination der Vorteile von klassischen Desktop- und Webanwendungen.

Außerdem ist SmartClient ein in JavaScript entwickeltes Open-Source-Framework zur Entwicklung von Smart-Clients mittels des AJAX-Konzepts, das sich vor allem durch seine reichhaltige Komponenten-Bibliothek auszeichnet.

1.10 GWT (Google Web Toolkit) / SmartGWT

... ist ein Framework zur Entwicklung von Webanwendungen. Seine Besonderheit ist ein Java-nach-Javascript-Compiler, so dass nahezu die gesamte Entwicklung von Client und Server auf Basis von Java realisiert werden kann. Weiterhin ist GWT mit einem XML-Parser, Internationalisierungs-Unterstützung, einer Schnittstelle für Remote Procedure Calls, Integration von JUnit, und einem kleinen Widget-Paket zur Gestaltung der graphischen Oberfläche ausgestattet.

SmartGWT ist ein Framework zum Entwickeln von GWT-Applikationen, welches zusätzlich zum GWT die Funktionalitäten der SmartClient-Komponenten-Bibliothek nutzt.

2 Konzepte

Das Ziel unseres Projektes ist eine Webanwendung, weshalb wir uns zunächst mit der allgemeinen Webarchitektur auseinander setzen müssen, sprich mit dem Client-Server-Modell:

2.1 Client-Server-Modell

Dieses beschreibt eine Möglichkeit, Aufgaben und Dienstleistungen innerhalb eines Netzwerkes zu verteilen. In dieser Struktur stellt das Server-Programm einen Dienst zur Verfügung (z.B. eine Webseite), welcher von verschiedenen Clients beansprucht werden kann. Das Server-Programm läuft in der Regel im Hintergrund, wartet auf Anfragen (Requests) der Clients, die es verarbeitet und beantwortet (Response - z.B. Auslieferung der Webseite). Anschließend steht es für weitere Anfragen bereit.

Da es sich bei unserer Anwendung um eine dynamische Startseite handelt und die geforderte Programmiersprache Java ist, haben wir uns informiert, wie man in Java flexible Webdienste zu Verfügung stellt. Hierbei wollen wir näher auf Servlets eingehen, da diese die Grundlage der dynamischen Java-Webprogrammierung darstellen:

2.2 Servlets-Architektur

Servlets sind zunächst einmal nichts anderes als ganz normale Java-Klassen, die ein bestimmtes Interface (javax.servlet.Servlet) implementieren. Damit diese Klassen wirklich auf HTTP-Anfragen eines Browsers reagieren und die gewünschte Antwort (zumeist eine HTML-Seite) liefern können, müssen die Servlets in einer bestimmten Umgebung laufen. Diese Umgebung wird vom so genannten Servlet-Container bereit gestellt.

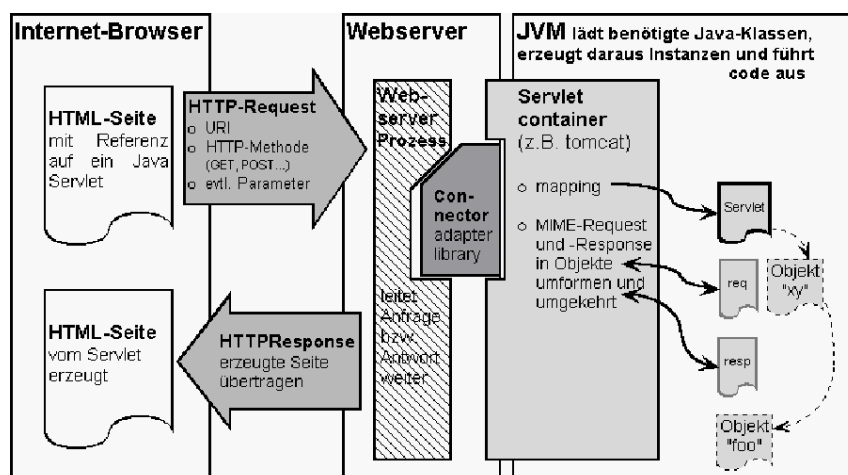


Abbildung 1: Zusammenspiel von Webserver und Servlet-Container.

Der Container sorgt zunächst einmal für den korrekten Lebenszyklus der Servlets, für das Pooling von Instanzen (bspw. von Datasources, aber auch der Servlets selbst) und dafür, dass Konfigurations-Parameter ausgelesen und bereitgestellt werden. Bei Anfragen entscheidet der Container anhand von Konfigurationsparametern darüber, an welche Servlets die Anfragen geroutet werden, und stellt den aufgerufenen Servlets mittels zweier Objekte vom Typ `javax.servlet.ServletException` bzw. `javax.servlet.ServletResponse` Informationen über Client-Anfragen bzw. Response zur Verfügung.

Zumeist arbeitet der Container dabei im Zusammenspiel mit dem Webserver. Der Webserver bedient z.B. die Anfragen nach statischem HTML-Content, nach Bildern, multimedialen Inhalten oder Download-Angeboten. Kommt hingegen eine Anfrage nach einem Servlet oder einer JSP herein, so leitet der Webserver diese an den Servlet Container weiter. Dieser ermittelt das zugehörige Servlet und ruft dieses mit den Umgebungsvariablen auf. Ist das Servlet mit seiner Abarbeitung des Requests fertig, wird das Ergebnis zurück an den Webserver geliefert, der dieses dem Browser wie gewöhnlich serviert. Der Nutzer bekommt von den Unterschieden nichts mit und für den Browser stellen sich alle Seiten einheitlich als Anfragen nach einer benannten Ressource dar.

Des Weiteren wurde uns für unserer Projekt das Google Web Toolkit (GWT) empfohlen. Dieses Framework stellt die Basis unserer Projektentwicklung dar und unterliegt folgender Architektur:

2.3 Google Web Toolkit

GWT stellt einen neuen Ansatz für die Entwicklung von Webanwendungen dar. Die Grundidee ist, den Client und den Server in derselben Programmiersprache, nämlich Java, zu schreiben und den Code für den Client automatisch in JavaScript zu konvertieren. Zur Umsetzung dieses Ansatzes bedient sich GWT des Java-to-JavaScript-Compilers, der im Toolkit enthalten ist. Dadurch ist es möglich, die gleichen Werkzeuge für die Entwicklung (z.B. Eclipse) zu verwenden. Für Java stehen viel mehr Tools und Möglichkeiten bei der Entwicklung zur Verfügung, wie z.B. Testen mit JUnit oder debuggen.

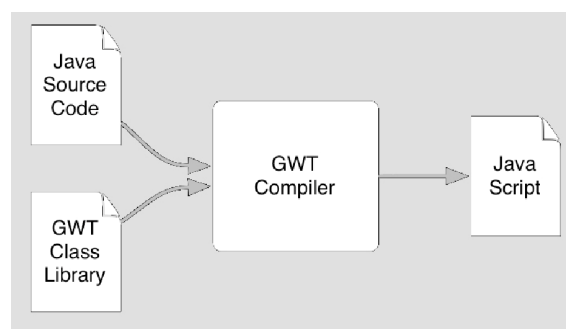


Abbildung 2: Funktionsweise des GWT-Compilers.

Die Komponenten von GWT lassen sich in folgende 2 Gruppen einteilen:

- Development Tools:

Bei den Development Tools handelt es sich um Hilfsmittel, die dem Entwickler zur Verfügung stehen und den Softwareentwicklungsprozess vereinfachen sollen.

Der **GWT-JavaScript-Compiler** stellt das Herzstück des GWTs dar. Dieser Compiler wandelt Java-Quelltext in äquivalenten JavaScript-Quelltext um.

Der **GWT-Hosted-Webbrowser** ist speziell für die Entwicklungsphase gedacht. Das GWT stellt einen Webbrowser zur Verfügung, der im JVM (java virtual machine)-Prozess läuft. Dadurch können Interaktionen mit dem Browser abgefangen und in der JVM verarbeitet werden. So wird das Debuggen der Webanwendung ermöglicht.

- Class Libraries:

Die Class Libraries stellen notwendige Klassenbibliotheken dar, die die unterschiedlichen GWT-Betriebsarten ermöglichen.

Die **JRE-emulation-library** beinhaltet eine Implementierung der Java-Standardklassenbibliothek, die jedoch auf die Möglichkeiten von JavaScript in den aktuellen Browsern beschränkt ist.

Die **GWT Web UI class library** stellt die eigentliche Benutzerschnittstellenbibliothek zur Verfügung, welche eine Sammlung von Standard-Widgets beinhaltet.

Zusätzlich erweitert SmartGWT diese Bibliothek um eine Vielzahl von Widgets.

3 Beschreibung der zu studierenden Applikation

3.1 GWT (inkl. SmartGWT) im Anwendungsfall

Das Google Web Toolkit ist mit der Erweiterung SmartGWT der zentrale Baustein unserer Projektentwicklung. Im folgenden möchten wir deshalb die Entwicklung eines GWT-Projektes genauer beschreiben:

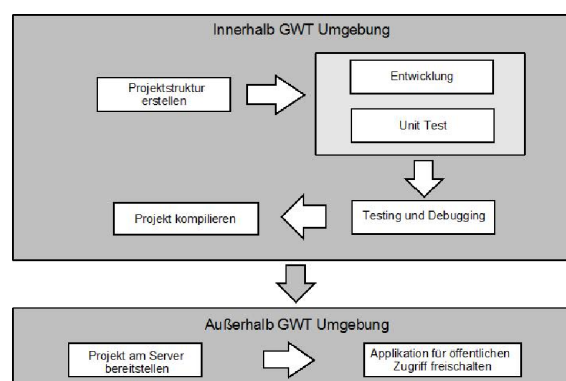


Abbildung 3: Typischer Lifecycle einer GWT-Applikationserstellung.

In der Abbildung (oben) zeigt sich der große Vorteil von GWT, die Applikation kann vollständig innerhalb der GWT Umgebung erstellt werden. Der später verwendete Servlet-Container, in unserem Fall Apache Tomcat, ist in der Entwicklungsphase nur insofern relevant, dass er die verwendeten Javaroutinen unterstützen muss.

3.2 Innerhalb der GWT-Umgebung:

Wir werden für unser Projekt die Entwicklungsumgebung Eclipse benutzen, welche durch ein GWT-Plugin erweitert wird. Dadurch können wir alle Möglichkeiten von GWT direkt in Eclipse benutzen, wie beispielsweise das Erstellen der Projektstruktur. Außerdem kann das Projekt, wie oben angesprochen, im GWT-Hosted-Mode ausgeführt werden, sprich in der JVM von GWT.

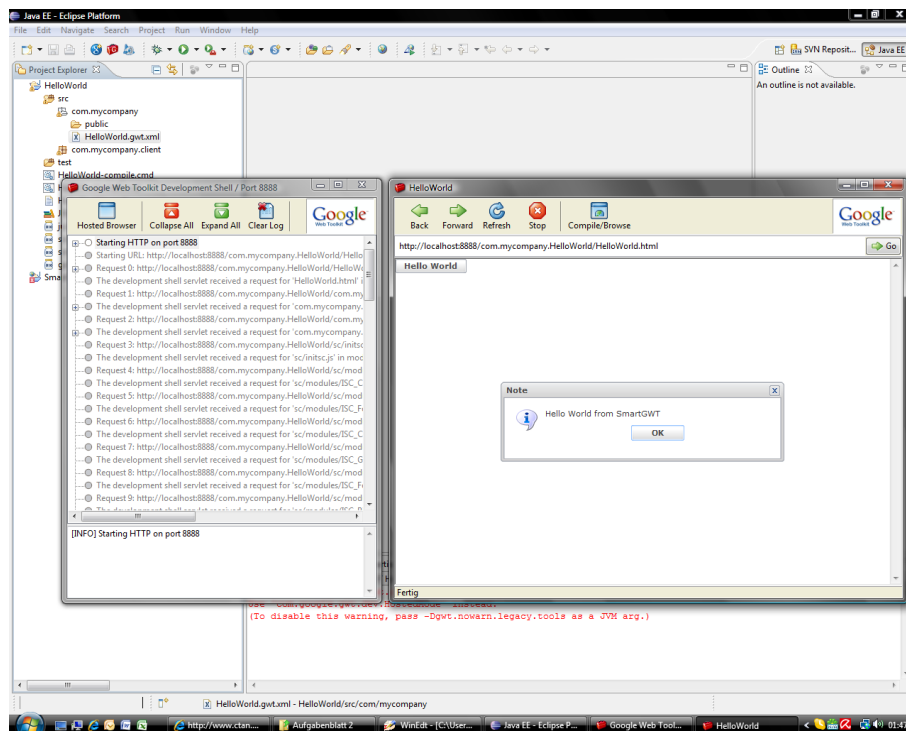


Abbildung 4: GWT-Hosted-Mode: Das GWT stellt neben einem Browser (rechts) noch eine Development-Shell (links) zur Verfügung.

Das GWT stellt jedoch nur eine kleine Sammlung an Widgets zu Verfügung, weshalb wir die Erweiterung SmartGWT benutzen werden. SmartGWT ist eine von mehreren Widget-Bibliotheken, welche das GWT um Tabellen, Kalender, Fenster usw. erweitert.

Um die Vielzahl an Widgets von SmartGWT kennen zu lernen, bietet dieses einen Showcase, indem die Verwendung und Implementierung der einzelnen Widgets dargestellt wird.

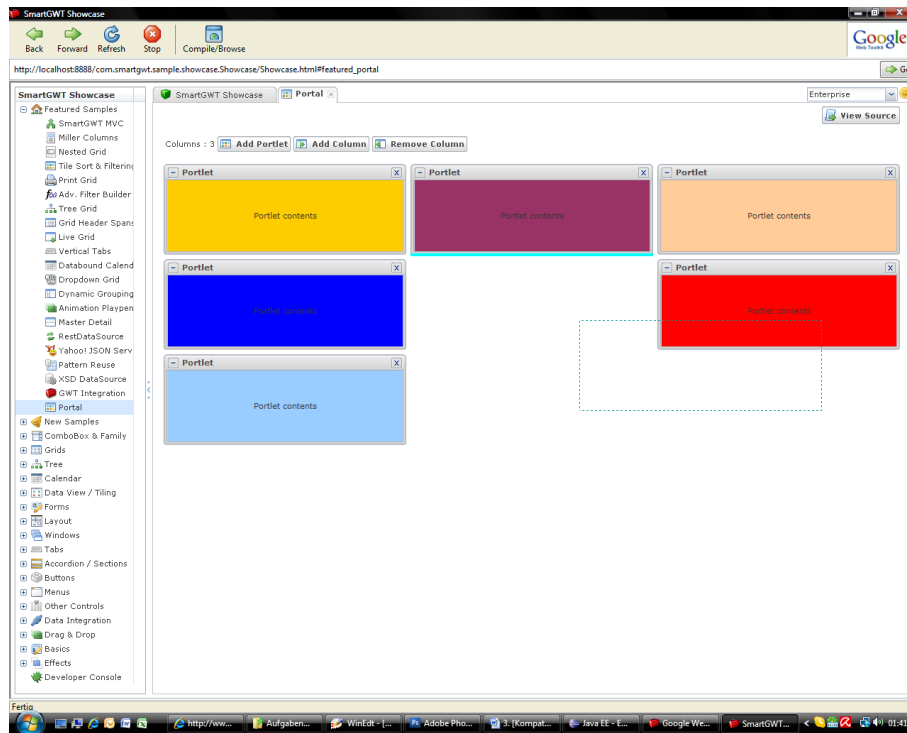


Abbildung 5: Beispielportal aus dem SmartGWT-Showcase.

3.3 Außerhalb der GWT-Umgebung:

Das fertige GWT-Projekt kann nach der Entwicklung direkt als WAR-File exportiert werden. Solch eine Datei stellt auch das Abgabeformat unseres Projektes dar. Diese Datei kann dann dem Java-Container (z.B. Tomcat) des Webservers übergeben werden, welcher die Applikation für die Ziel-Clients bereitstellt.