

# 1 Einleitung

## 1.1 Was ist Vaadin?

Vaadin ist ein auf dem Google Web Toolkit (siehe [GWT]) basierendes Open-Source-Framework, das die einfache Erstellung auch komplexer browserbasierter Applikationen ermöglicht.

Tatsächlich bringt sich Vaadin auch sehr stark in das GWT-Projekt ein: Es stellt ein Mitglied des Steuerungsausschusses und organisiert auch die Konferenz gwt.create – sieht sich also nicht als Konkurrenz, sondern in positiver Koexistenz mit GWT.

Vaadin abstrahiert hierbei für die normale Programmierung von HTML, JavaScript, AJAX und browserspezifischen Details und bietet ein javabasiertes Programmiermodell mit Widgets und Events an, das sehr stark an klassische UI-Programmierung in Java angelehnt ist.

Tatsächlich müssen wir uns in der normalen Programmierung mit Vaadin keine Gedanken über verschiedene Datenmodelle auf Client und Server machen und haben keine Schwierigkeiten mit der Aktualisierung der jeweiligen Daten. Wir können unsere verschiedenen Widgets und ihr Layout mit Java programmieren und kommen mit HTML und JavaScript nicht in Berührung.

*Programmierung  
vollständig in Java*

Zusätzlich können die Details des Layouts getrennt von der Programmierung beeinflusst und über CSS applikationsweit konsistent gesteuert werden, so dass bei Bedarf auch Look & Feel der gesamten Applikation über *Themes* ausgetauscht werden kann.

*Steuerung des Layouts  
über CSS*

Über einen Erweiterungsmechanismus besteht die Möglichkeit, Widgets von Drittanbietern einfach zu integrieren und zu programmieren. Das *Vaadin Directory* (siehe [Vaadin]) bietet aktuell mehr als 400 Komponenten und Widgets an, die die Funktionalität von Vaadin erweitern.

Hinter den Kulissen implementiert Vaadin neben dem Framework auf der Serverseite, das wir in der normalen Programmierung benutzen, eine *Client-Side Engine*, die im Browser läuft und sowohl die Darstellung der Benutzerschnittstelle als auch die Übermittlung der Benutzerinteraktionen an den Server übernimmt und sich dabei auch um browserspezifische Anpassungen kümmert. Diese Client-Side Engine ist in Java geschrieben und wird mit dem GWT-Compiler in JavaScript übersetzt.

Während wir bei der normalen Programmierung nicht viel mit der Client-Side Engine zu tun haben, wird diese interessant, wenn wir eigene Widgets für die Erweiterung von Vaadin programmieren. Dies ist allerdings ein Thema für Fortgeschrittene und wird in diesem Buch nicht weiter betrachtet.

## 1.2 Historie

Die Entwicklung an Vaadin wurde bereits im Jahr 2000 gestartet, damals als Adapter für das Framework Millstone der Firma *IT Mill Ltd.*, und die erste Version wurde 2002 veröffentlicht. Die Funktion umfasste damals Kommunikation mittels Ajax und eine eigene, proprietäre Render-Engine für die Darstellung von Widgets.

Ab 2006 wurde das Framework eigenständig kommerziell weiterentwickelt und 2007 umbenannt zu *IT Mill Toolkit*. Da sich im Lichte des Open-Source-Frameworks GWT die Weiterentwicklung einer eigenen Render-Engine nicht lohnt, wurde der Wechsel zu GWT als Basis für das Rendering gestartet. Ende 2007 wurde die Lizenz in eine Open-Source-Lizenz geändert.

2008 investierte einer der Gründer von MySQL (Monty Widenius) einen Teil seines Vermögens in die Firma und schuf damit die Grundlage für die Entwicklung der ersten Version des IT Mill Toolkit 5, die nach langer Betatestperiode 2009 erschien. Kurz nach Erscheinen wurde das Framework umbenannt in Vaadin (der finnische Name für ein weibliches Rentier) und eine Vorabversion der Version 6 veröffentlicht. Zusätzlich wurde auch IT Mill Ltd. umbenannt zu Vaadin Ltd., um klarzumachen, wie sehr die Firma hinter ihrem Framework steht.

Es dauerte bis März 2013, bis die Version 7 herauskam. Ende Juni 2013 kam die Version 7.1 heraus, die abgesehen von Fehlerkorrekturen als wichtigste Funktionalität *Server Push* enthielt. Hiermit wird der Server in die Lage versetzt, eigenständig Informationen auf dem Client zu aktualisieren, was das Programmiermodell noch flexibler macht.

## 1.3 Wofür ist Vaadin gut?

Vaadin ist ein typischer Vertreter eines *Rich Internet Application Frameworks* (RIA Framework), mit dessen Hilfe Webapplikationen implementiert werden, die so weit wie möglich einer klassischen Desktop-Applikation entsprechen. Hierbei zeigt sich als Trend, dass mehr und mehr weggegangen wird von proprietären Lösungen wie zum Beispiel Adobe Flash hin zu HTML5 und JavaScript. GWT und damit auch Vaadin setzen auf diesem Trend auf.

Mit immer größerer Leistungsfähigkeit der unterliegenden Browser und eines Frameworks wie Vaadin wird die Distanz zwischen klassischen Applikationen und Applikationen, die im Browser laufen, immer geringer, und Analysten sagen bereits seit 2007 voraus, dass die Grenzen zwischen diesen Applikationsarten immer mehr verschwimmen werden (siehe [Forrester 2007]).

Damit stellt sich weniger die Frage, was wir mit Vaadin machen können, als was sich nicht machen lässt. Die folgenden beiden Punkte sind problematisch:

- Verarbeitung großer Datenmengen auf dem Client
- vollständige Funktionalität ohne Netzzugang (Offline-Funktion für mobile Geräte)

Im ersten Fall stellt sich die Frage, warum die Datenverarbeitung auf dem Client durchgeführt werden muss und ob es nicht sinnvoller wäre, die Daten zum Server oder in eine Cloud-Infrastruktur zu bewegen.

Im zweiten Fall haben wir eine Situation, die das Programmiermodell von Vaadin ad absurdum führt. Wenn es hingegen um zeitlich beschränkte Offline-Funktionalität geht, dann gibt es zwei Aspekte: zum einen die Berechnung von Informationen auf dem Server. Dies kann man in den Griff bekommen, indem die entsprechenden Berechnungen lokal durchgeführt werden oder eine Meldung angezeigt wird, dass der Service aktuell nicht zur Verfügung steht. Der andere Aspekt ist die Eingabe von Daten. Hier besteht die Möglichkeit, die Daten lokal zu cachen und zum Server zu übertragen, sobald dieser wieder verfügbar ist.

Die Vaadin-Erweiterung TouchKit für mobile Geräte bietet Unterstützung für diese Vorgehensweisen. In jedem Fall ist es aber sinnvoll, den Nutzer auf die fehlende Verbindung hinzuweisen; das Touchkit ermöglicht, hierfür ein eigenes Theme zu verwenden, um dies sehr deutlich zu signalisieren.

## 1.4 Unsere Erfahrungen mit Vaadin

Wir verwenden Vaadin in vielen unserer Projekte und haben die verschiedensten Arten von Anwendungen bereits erfolgreich umgesetzt.

Es gibt prinzipiell keinen Typ Anwendung, bei dem wir sagen würden, dass Vaadin nicht verwendet werden kann – im Gegenteil sind die durch Vaadin zur Verfügung gestellten Abstraktionen und Funktionen so hilfreich, dass wir bei neuen Webapplikationen immer zur Verwendung von Vaadin tendieren. Dies gilt für Anwendungen im Intranet, im Internet, mit wenigen und vielen Benutzern, mit einfachen und komplexen Benutzerschnittstellen. Allerdings muss man bei komplexeren Benutzerschnittstellen darauf achten, dass die Komponenten nicht zu sehr verschachtelt werden (wozu man als unerfahrener Vaadin-Entwickler neigen kann), um die Applikation nicht unnötig zu bremsen.

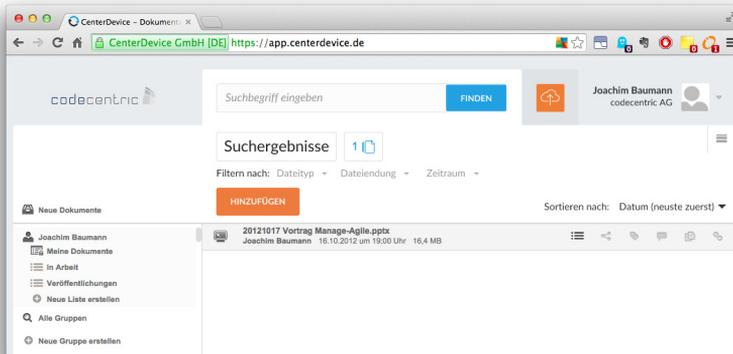
Ein weiterer wichtiger Punkt ist, wie bei jeder Webapplikation, die Größe der Session. Vaadin legt die Benutzerschnittstellenobjekte (alle Objekte, die an den UI-Objekten hängen) in der Session ab und vergrößert diese damit natürlich (logischerweise umso mehr, je mehr verschachtelte Komponenten die Anwendung enthält). Nach dem Schließen des Browserfensters bleibt die Session so lange erhalten, bis sie vom System abgeräumt wird, und verbraucht damit weiterhin Speicher (sofern sich der Benutzer nicht explizit abmeldet). Wenn zusätzlich die Benutzer die gleiche Applikation mehrfach öffnen, dann legt Vaadin diese Objektbäume natürlich mehrfach in der Session ab. Dies ist aber nichts Ungewöhnliches und wird auch von anderen serverzentrierten Frameworks in gleicher Weise gelöst. In der praktischen Verwendung haben wir noch keine Probleme erlebt, die die Verwendung von Vaadin verhindert hätten, und auch die notwendige Größe der verwendeten Server ist im normalen Bereich. Ob allerdings eine Webapplikation wie Facebook mit Vaadin performant wäre, wagen wir zu bezweifeln.

Die eingebauten Kommunikationsmechanismen erlauben eine gute und reaktionsfähige Interaktion auch mit vergleichsweise niedrigen Datenraten des unterliegenden Netzes, solange die Latenzzeiten nicht zu hoch werden.

*CenterDevice*

Ein Beispiel, das die Verwendung von Vaadin mit vielen Nutzern und hohen nichtfunktionalen Anforderungen sehr gut demonstriert, ist die Cloud-Anwendung CenterDevice (siehe [CenterDevice]), die von Mitarbeitern der codecentric entwickelt wurde. CenterDevice ist eine Anwendung für Cloud Storage, in der Sie Ihre Daten in einer deutschen Cloud ablegen, automatisch verschlagworten und mit anderen teilen können. Hier haben wir eine komplexe Benutzerschnittstelle kombiniert mit sehr vielen Benutzern und großen Datenmengen, die vom und zum Nutzer transferiert werden müssen und für die auch

(zumindest für die gängigsten Formate) eine direkte Darstellung in der Applikation möglich sein muss.



**Abb. 1-1**

Beispiel einer komplexen  
Vaadin-Applikation  
(CenterDevice)

Hier waren tatsächlich einige Iterationen notwendig, bis die Applikation die gewünschte Reaktivität hatte, auch und gerade beim Umgang mit großen Dateien. Aber das Ergebnis spricht für sich, und wir würden uns auch in diesem Fall mit den sehr hohen Anforderungen wieder für Vaadin entscheiden. Holen Sie sich bei Interesse einfach einen Test-Account und urteilen Sie selbst.

Einzig Offline-Applikationen, die ohne Verbindung mit dem Server uneingeschränkt funktionieren, sind mit Vaadin ohne zusätzliche Arbeit nicht möglich. Vaadin bietet zwar rudimentäre Unterstützung, aber für die Offline-Unterstützung ist trotzdem eine Menge an Handarbeit nötig, die die Vorteile von Vaadin einschränkt.

## 1.5 Weitergehende Informationen

Auf die Website von Vaadin gelangen Sie unter folgender URL:

*<http://www.vaadin.com>*

Hier gibt es nicht nur die jeweils aktuelle Version von Vaadin, Add-ons für die Entwicklung und die Integration in verschiedene Entwicklungs-umgebungen, sondern auch Tutorials und eine Online-Version des »Book of Vaadin«, der Referenz für die Verwendung von Vaadin.

Außerdem finden Sie dort das Vaadin-Forum, das die direkte Kommunikation mit den Vaadin-Entwicklern ermöglicht. Falls Sie nicht den Umweg über die Vaadin-Website nehmen wollen, folgen Sie einfach der URL:

*<http://www.vaadin.com/forum>*