

## Aufbau des Buchs

Dieses Buch ist in 9 Kapitel unterteilt. Die ersten beiden Kapitel geben eine Einführung in die Thematik der Performance-Optimierung und einen kurzen Einblick in grundlegende Optimierungsmöglichkeiten. Optimierungen können jedoch erst durchgeführt werden, sobald entsprechende Schwachstellen in der Webanwendung erkannt wurden. Mit der Identifizierung dieser Schwachstellen befasst sich daher Kapitel 3 und zeigt, wie die Gesamtperformance einer Webanwendung bestimmt und die Grenzen ermittelt werden können.

Die darauffolgenden Kapitel beschäftigen sich im Detail mit der Erläuterung und der Optimierung von Flaschenhälsern, beginnend mit Datenbanken in Kapitel 4. Des Weiteren zeigen Kapitel 5 und 6 einen Einstieg in die Konzepte des Cachings und Queueings, mit deren Hilfe Webanwendungen beschleunigt werden können.

Neben neuen Konzepten in Datenbanken, Caching und Queueing spielt bei Webanwendungen das Netzwerk, genauer das Internet, eine wichtige Rolle. Daten werden von Servern abgefragt, um diese in Browser- oder Clientanwendungen darzustellen. Da die Kommunikation zum Teil über sehr weite Strecken und schlechte Verbindungen erfolgt, bestehen gerade hier Anforderungen an die Antwortzeit und die Datengröße. Kapitel 7 befasst sich daher ausgiebig mit der grundlegenden Thematik der Datenübertragung und der Optimierung der übermittelten Daten.

Sind die Daten an das Frontend übertragen, ergeben sich auch bei der Darstellung der Webanwendung besondere Anforderungen. Kapitel 8 widmet sich der Darstellung dieser übertragenen Inhalte auf dem Client und zeigt Verarbeitungsoptimierungen der Inhalte wie HTML, Grafiken, Stylesheets sowie JavaScript.

Um aufwendige Optimierungsprozesse nicht immer manuell ausführen zu müssen, beschreibt Kapitel 9, welche Optimierungen bei der Entwicklung bzw. Auslieferung der Webanwendung automatisiert werden können. In Kapitel 9 werden diverse Verfahren und Tools vorgestellt, die die Automatisierung von Optimierungen erleichtern.

Dieses Buch ist dabei so aufgebaut, dass nach den allgemeinen Optimierungsmöglichkeiten mit serverseitigen Optimierungen begonnen wird. Im weiteren Verlauf werden dann die Datenübertragung bis hin zur Darstellung auf dem Client optimiert. Es empfiehlt sich daher, die Kapitel nacheinander zu lesen, da die Themen von der Serverseite bis hin zur Clientseite nacheinander bearbeitet werden und aufeinander aufbauen.

**Kapitel 1: Einführung in die Performance-Optimierung** Das erste Kapitel definiert den Begriff »Performance-Optimierung« in diesem Buch und stellt einige Mythen vor, die sich in diesem Bereich etabliert haben. Daran anschließend werden einige Gründe dargelegt, die Entwickler und Firmen zur Performance-Optimierung bringen, und erläutert, warum Performance-Optimierung für alle Bereiche einer Anwendung wichtig ist. Diese Bereiche werden im Buch als Performance-Bereiche bezeichnet und ihre Auswirkungen auf die Performance des Gesamtsystems erläutert.

**Kapitel 2: Grundlegende Optimierungsmöglichkeiten** In diesem Kapitel werden Ansätze zur Optimierung vorgestellt, die einen Überblick geben, welche Arten von Optimierungen durchgeführt werden können. Diese Grundsätze werden dann im weiteren Verlauf des Buches detailliert behandelt.

**Kapitel 3: Last- und Performance-Test** Das Kapitel 3 zeigt, mit welchen Mitteln die Performance von bestehenden und neuen Webanwendungen gemessen werden kann. Es wird erläutert, wie die Anwendung gemessen werden kann und wie Tests auf andere Komponenten ausgeweitet werden können. Dabei wird der gesamte Verlauf eines Tests, von der Erstellung eines Testplans über die Durchführung bis hin zu Auswertung und Optimierung, dargestellt.

**Kapitel 4: Datenbankoptimierung** Datenbanken sind heutzutage der Grundbaustein vieler Webanwendungen. Daher wird gezeigt, welche Unterschiede zwischen verschiedenen Datenbanksystemen bestehen, auf welche Kriterien es bei der Auswahl von Datenbanken ankommt und welche grundlegenden Optimierungen und Einstellungen an Datenbanken vorgenommen werden können.

**Kapitel 5: Caching** Caching ist eine der wichtigsten Disziplinen bei der Performance-Optimierung, da mit Caching viele Systeme entlastet werden können. Dabei kommt es allerdings auf den richtigen Einsatz an. Das Kapitel zeigt, welche Caching-Mechanismen existieren und wie ein Caching-Plan für eine Webanwendung, basierend auf deren Daten und Nutzung, erstellt werden kann.

**Kapitel 6: Asynchrone Verarbeitung und Queuing** Der Einsatz von Queuing ist eine einfache Möglichkeit, um Systeme verfügbar und skalierbar zu machen, da damit Systeme voneinander entkoppelt werden können. Im Frontend-Bereich gibt es seit Jahren den asynchronen Ansatz mittels

Ajax. Im Backend-Bereich wird die asynchrone Bearbeitung jedoch immer noch weitestgehend ignoriert. Dieses Kapitel behandelt daher Queuing-Services und zeigt, wie durch den Einsatz von Queuing Webanwendungen performanter und ausfallsicherer werden können.

**Kapitel 7: Datenübertragung und Auslieferung** Bei Webanwendungen werden Daten zwischen Client und Server ausgetauscht. In der Regel sind dies HTML-, CSS-, JavaScript- und Grafikdateien. Das Kapitel zeigt, wie diese Daten optimal über das Netzwerk übertragen und die Datenmengen durch die Wahl der richtigen Formate und Kompression verringert werden können.

**Kapitel 8: Darstellung auf dem Client** Neben der optimierten Übertragung der Daten kann auch die Darstellung der Inhalte im Browser beschleunigt werden. Dazu wird die Funktionsweise eines Browsers erklärt und erläutert, wie sowohl das Rendering der Seite als auch die Ausführung von CSS und JavaScript verbessert werden können.

**Kapitel 9: Automatisierung** Um eine ständig optimierte Webanwendung zu gewährleisten, ist es sinnvoll, möglichst viele der angewandten Optimierungsschritte zu automatisieren und in den Build- bzw. Deployment-Prozess des Projektes zu integrieren. Dieses Kapitel zeigt einige Ansätze, wie diese Integration erreicht werden kann.