



STUD.IP

3. Stud.IP-Entwickler-Workshop • 2. Juni 2006

Workshop Ia: Gib mir 5: Lohnt sich der Umstieg auf PHP5 und MySQL5?

André Noack

Gliederung

- ▶ **(nicht mehr so) Neues in PHP 5**
- ▶ **Neues in MySQL 4.1 / 5**
- ▶ **Probleme mit Stud.IP**
- ▶ **Vor- und Nachteile aus Entwickler / Betreibersicht**
- ▶ **Diskussion**

(nicht mehr so) Neues in PHP 5

- ▶ **Nicht mehr so neu?**
 - ▶ PHP 5.0.0 wurde am 13.07.2004 released!
 - ▶ Derzeit laufen noch 93 Prozent aller PHP-Server weltweit mit Version 4 (Stand Februar 2006)
 - ▶ Aber: „Anwendungen in PHP 5 kommen mit bis zu 30 Prozent weniger Code aus und neue Funktionen lassen sich dank besserer Modellierungsmöglichkeiten und transparenterem Code-Aufbau in einem Drittel weniger Zeit entwickeln“
 - ▶ Wieso benutzt es dann niemand?

(nicht mehr so) Neues in PHP 5

▶ Zend II Engine

▶ neues Objektmodell

- ▶ Objektvariablen werden nicht mehr by-value zugewiesen
- ▶ Automatische Dereferenzierung von Objekten
- ▶ Public/private/protected Zugriffsmodifikatoren für Methoden und Eigenschaften
- ▶ Einheitliche Konstruktoren / Destruktoren
- ▶ Interfaces (aber weiterhin schwache Typisierung)
- ▶ Final / Abstract Klassen / Methoden
- ▶ Static Methoden / Eigenschaften
- ▶ Klassenkonstanten

(nicht mehr so) Neues in PHP 5

▶ Allgemeine Änderungen

▶ Exception Handling

- ▶ Try / Throw / Catch Konstrukte

▶ XML

- ▶ SAX, DOM, XSLT über libxml2
- ▶ Einheitlicher, effizienter, performanter (aber nicht abwärtskompatibel)
- ▶ SimpleXML, XML als natives PHP Objekt
- ▶ SOAP Extension – einfach und schnell
- ▶ WSDL (5.1)

(nicht mehr so) Neues in PHP 5

▶ Interessantes in PHP 5.1

▶ PDO (PHP Data Objects)

- ▶ einheitliche Datenbankabstraktion
- ▶ Schnell, leichtgewichtig

▶ SPL (Standard PHP Library)

- ▶ Sammlung von Standard Klassen
- ▶ Enthält Klassen für Iteratoren, Datei- und Verzeichniszugriff, XML-Zugriff, Array-Overloading, auf- und abzählbare Datenstrukturen, Umgang mit Exceptions, Standard-Design-Patterns

Neues in MySQL 4.1 / 5

▶ MySQL 4.1

- ▶ Unterabfragen (subqueries)
- ▶ Komplette Unicode Unterstützung (UTF-8)
- ▶ Multi Zeichensatz Unterstützung auf Datenbank, Tabellen und Felder Ebene
- ▶ Prepared Statements
- ▶ Persönliches Lieblingsfeature: GROUP_CONCAT()

Neues in MySQL 4.1 / 5

▶ MySQL 5

- ▶ Views (updatefähig)
- ▶ Stored Procedures / Functions (SQL:2003 Syntax)
- ▶ Triggers („rudimentary“)
- ▶ VARCHAR kann 64K Zeichen speichern
- ▶ MySQL Cluster

Probleme mit Stud.IP

▶ MySQL

- ▶ 4.1: keine bekannten Probleme mehr in der 1.3
- ▶ Aber: Vorsicht beim Update, Zeichensatz sollte latin1 sein, Kollation latin1_german1_ci
- ▶ Bei parallel laufenden PHP Applikationen, die UTF-8 benutzen (Ilias!) können Probleme entstehen
- ▶ 5.0: Probleme mit der JOIN Reihenfolge, in der unstable Version behoben

Probleme mit Stud.IP

▶ PHP 5

- ▶ Im Moment nicht lauffähig
- ▶ Probleme:
 - ▶ Benutzung von PHP 5 Schlüsselwörtern, überambitioniertes OOP bei Votes / Evals
 - ▶ „\$this is evil“, polymorph-perverse Konstruktoren im Kalender
 - ▶ Jedes Modul, dass Klassen benutzt, kann schwer zu findende Probleme wegen „by-reference“ bekommen (Ressourcenverwaltung)
 - ▶ XML Handling (export, Ilias Schnittstelle, RSS)

Vor- und Nachteile

▶ PHP 5

▶ Pro:

- ▶ Neue OOP Funktionalität könnte die Entwicklung einer Stud.IP API vorantreiben
- ▶ Neue XML Funktionen sehr mächtig und trotzdem einfach einzusetzen, außerdem performanter
- ▶ In der Entwicklung befindliche Frameworks / Bibliotheken werden vermutlich nur PHP 5 unterstützen

Vor- und Nachteile

▶ PHP 5

▶ Contra:

- ▶ Umstellungsaufwand, viel Arbeit um bestehende Funktionalität wieder lauffähig zu machen
- ▶ Schrittweises Umstellen ist sinnlos, ohne die neue Funktionalität benutzen zu können macht sich niemand die Mühe
- ▶ Solange der PHP 4 Zweig weiter gepflegt wird besteht von Seiten der Betreiber kein Bedarf („ntars“), d.h. es wird niemand Ressourcen investieren wollen
- ▶ Et küt wie et küt und et hät noch immer jot jejange!!

Vor- und Nachteile

▶ MySQL

▶ Pro:

- ▶ 4.1 bereits in allen Distributionen enthalten, 5.0 auch schon vielfach im Einsatz
- ▶ Um eine Unterstützung kommen wir sowieso nicht herum
- ▶ Subqueries machen das Leben leichter, wenn man keine Joins mag
- ▶ Views und stored Procedures helfen die Struktur der DB zu entschlacken

Vor- und Nachteile

▶ MySQL

▶ Contra:

- ▶ 4.0 ist auch noch weit verbreitet
- ▶ Subqueries, views, etc haben wir bis jetzt auch nicht gebraucht
- ▶ Bloß keine weiteren MySQL spezifischen Abhängigkeiten, ich will sowieso Oracle

Diskussion

- ▶ Bis hierhin: Danke für die Aufmerksamkeit!
- ▶ Los gehts...