

Inhaltsverzeichnis

1	Webframeworks	2
1.1	Definition	2
1.2	Warum Python	2
1.3	Fullstack vs. Glue	3
1.4	Weitere Unterscheidungsmerkmale	4
1.4.1	Funktionalität	4
1.4.2	Dokumentation	4
1.5	Bekannte Webframeworks im Pythonumfeld	5
1.5.1	Django	5
1.5.2	Pylons	5
1.5.3	Zope	5
1.5.4	web2py	5
1.6	Webframeworks anderer Programmiersprachen	6
1.6.1	Überblick	6
1.6.2	Zend	6
1.6.3	Ruby on Rails	6
1.6.4	Catalyst	7
1.6.5	Wicket	7
1.7	Arbeiten ohne Framework	7

1 Webframeworks

1.1 Definition

Webframeworks sind im Grunde eine Sammlung von Funktionalität, die es dem Webentwickler erlaubt Anwendungen zu schreiben. Die Funktionalität besteht aus verschiedenen Libraries, die den Zugriff auf Datenbanken, die Erstellung von Templates für die Ausgabe in HTML oder die Erstellung von Klassen und Methoden für die Geschäftslogik.

Webanwendungen, die sich auf Datenbanken stützen können mit Webframeworks relativ schnell erstellt werden, dabei helfen OR-Mapper¹, die aus Klassenkonstrukte SQL² für eine ausgewählte Datenbank erstellen. Die Mapper bieten darüberhinaus eine eigene Sprache, um auf die Klassen bzw. Informationen in der Datenbank zuzugreifen. Der Vorteil liegt darin, dass der Code unabhängig von einer Datenbank erstellt werden kann. Die Geschäftslogik kann abhängig vom Framework in einer beliebigen Programmiersprache geschrieben werden, meistens Java, PHP oder Python. Um die Ausgabe zu generieren bringen die Frameworks eine Templatesprache mit. Diese ermöglicht die einfache Erstellung von HTML aufgrund der Mischung von HTML und Templatecode. Dieser Code dient überwiegend zur Darstellung von Information, die aus der Geschäftslogik stammt.

1.2 Warum Python

Die Programmiersprache Python ist eine sehr flexible Programmiersprache. Sie kann in verschiedenen Gebieten eingesetzt werden. Unter anderem bei der Verarbeitung von XML, zur Verbindung von Datenbanken, in der Netzwerk Programmierung, im Webbereich auch die Erstellung von Programmen mit grafischer Oberfläche (wxPython, PyQt) ist möglich. In dieser Arbeit wird vor allem der Webbereich genauer analysiert und verschiedene Webframeworks am Beispiel einer Anwendung verglichen.

Python ist nicht an ein bestimmtes Programmierparadigma gebunden und je nach

¹Objekt Relationale Mapper

²Structured Query Language

Aufgabe kann ein Paradigma, wie z.B. Objektorientierung, Funktional Programmierung oder ein aspektorientierter Ansatz gewählt werden.

Im Jahre 1991 wurde die Sprache veröffentlicht. Sie wurde im *Centrum voor Wiskunde en Informatica* in Amsterdam entwickelt. Dabei gilt *Guido van Rossum* als Erfinder der Sprache. Python wurde unter anderem von C, LISP und Smalltalk beeinflusst. Wobei Python selbst wiederum Ruby, Boo oder Groovy beeinflusst hat.

Ja, Python ist gewachsen und hat sich zu einer ausgereiften Sprache samt eigenem Ökosystem entwickelt.[1]

Im Vergleich zu z.B. PHP[3] gibt es zwar auch viele Python-Webframeworks[4], diese sind aber weniger bekannt, mit der Ausnahme von z.B. Django oder TurboGears. Es werden diese zwei beim Vergleich eingesetzt, zusätzlich auch neuere oder weniger bekannte Frameworks im Pythonökoystem.

1.3 Fullstack vs. Glue

Frameworks werden hauptsächlich unterschieden zwischen Fullstack- und Glueframeworks.

Ein Fullstackframework bringt alle möglichen Komponenten mit, um eine Webanwendung schreiben zu können. Unter anderem sind das ein OR-Mapper für die Datenbank, eine eigene Templatesprache oder ein URL-Dispatcher³ für lesbare Links. Der Vorteil eines solchen Frameworks ist, dass eine klare Linie vorgegeben wird, um eine Anwendung schreiben zu können. Dies führt in der Regel zu schnellen Ergebnissen. Vertreter dieser Art sind z.B. Django.

Auf der anderen Seite gibt es Frameworks, die aus einzelnen unabhängigen Komponenten bestehen. Dabei sind diese oft lose gekoppelt und können leicht ausgetauscht werden. So können für diese Frameworks immer die neuesten Komponenten benutzt werden, die von Gluecode⁴ zusammengehalten werden. Es ist aber ein größerer Aufwand nötig, um diese losen Teile zusammenzubringen. Das bekannteste Framework dieser Art in Python ist TurboGears.

³<http://www.w3.org/Provider/Style/URI>

⁴http://en.wikipedia.org/wiki/Glue_code

1.4 Weitere Unterscheidungsmerkmale

1.4.1 Funktionalität

Wenn ein neues Produkt das Interesse geweckt hat, liegt es meistens an der Funktionalität, die es anbietet. Für eine Webapplikation ist es wichtig, dass auf eine Art und Weise der Client auf die Dateien mittels Browser zugreifen kann. Als Schnittstelle bietet sich im Pythonumfeld WSGI⁵ an, damit der Webserver mit der Webapplikation kommunizieren kann, die wiederum HTML an den Client liefert. Für einfache Anwendungen reicht es aus HTML-Code mit Hilfe von Python zu generieren und diesem dem User zur Verfügung zu stellen. Soll die Anwendung aber komplexere Ausmaße haben und zum Beispiel eine Userverwaltung mit LogIn Möglichkeit bieten ist es ratsam ein Framework auszuwählen, welches diese Funktionalität mit einfachen Handgriffen bietet. Auch die Kommunikation zu Datenbanken ist ein wichtiger Bestandteil, da viele Webapplikationen Klassen bereitstellen, mittels derer Objekte generiert werden, die miteinander interagieren. Daraus entstehen meistens Informationen die persistent gehalten werden müssen. Beispiele sind Webshops, Blogs/Wikis oder soziale Netzwerke. Bei der Entwicklung sollte auch berücksichtigt werden, dass Webdesigner leicht im Entwicklungsprozess integriert werden können. Dabei sollte wichtig sein, dass CSS und JavaScript (oder JavaScript-Frameworks), die als Standard bei der Gestaltung gelten eingesetzt werden können. Dabei spielen Templatesprachen für die Erstellung des HTML-Codes eine große Rolle, denn viele Webframeworks bieten Templatesprachen, die sich an Pythoncode anlehnen aber nur geringe logische Funktionalität bieten und im Grunde nur für die Darstellung der einzelnen Seiten konzipiert sind.

1.4.2 Dokumentation

Viele Projekte und im Speziellen hier Webframeworks werden hauptsächlich wegen ihrer Funktionalität ausgewählt. Ein weiterer Grund kann auch ihre Popularität sein auf Grund eines Hypes. Ein sehr wichtiger Aspekt, um überhaupt mit einem Framework oder Programmiersprache anzufangen, ist die Lernhürde die manchmal hoch sein kann. Deswegen ist es wichtig und sollte eine Priorität der Entwickler sein die Frameworks sehr gut zu dokumentieren und Anfängertutorials zu anbieten.

⁵<http://wsgi.org/wsgi/>

1.5 Bekannte Webframeworks im Pythonumfeld

1.5.1 Django

Django ist ein High-Level Python Webframework, das eine schnelle Entwicklung fördert und ein sauberes und pragmatisches Design verfolgt.

Django wurde designed, um zwei Herausforderungen zu meistern: Intensiver Deadlines eines Newsrooms und den strengen Anforderungen der erfahrenen Webentwickler, die das Framework geschrieben haben, gerecht zu werden. Django ermöglicht es performante und elegante Anwendungen, schnell zu entwickeln. [6]

1.5.2 Pylons

Pylons verbindet Ideen aus der Welt von Ruby, Python und Perl. Dabei ist es ein sehr flexibles Python-Webframework und verhalf dem WSGI-Standard zum Durchbruch. Das Ziel von Pylons ist es, die Webentwicklung schnell, flexibel und einfach zu gestalten.[5]

1.5.3 Zope

Die Zope Softwarebibliothek ermöglicht die komponentenbasierte Entwicklung von Web-Anwendungen in der objektorientierten Programmiersprache Python. In der komponentenbasierten Programmierung werden komplexe Anwendungen mit Hilfe wiederverwendbarer Komponenten erstellt. Eine Komponente stellt dabei die Implementierung einer bestimmten genau spezifizierten Funktionalität dar.[7]

1.5.4 web2py

web2py wurde inspiriert von Ruby on Rails und, wie Rails, liegt der Fokus in schneller Webentwicklung und folgt dem Model View Controller Muster. web2py unterscheidet sich von Rails, weil es auf Python basiert. Weil es eine umfassende Web-basierte Administrationsoberfläche bereitstellt, darüber hinaus beinhaltet es Bibliotheken, um mit mehr Protokollen zurechtzukommen und ist auf der Google App Engine lauffähig.[8]

1.6 Webframeworks anderer Programmiersprachen

1.6.1 Überblick

Zur Erstellung von Webapplikationen gibt es viele Möglichkeiten dies zu bewerkstelligen. Der einfachste Ansatz ist, eine Programmiersprache zu lernen, mit deren Hilfe HTML-Code erzeugt und dem Client zur Verfügung gestellt wird.

Der erste Kontakt wird sicherlich mit PHP gemacht. Leicht zu erlernen und mit einer Vielzahl von Funktionen. Darüber hinaus existieren eine Menge Frameworks, die den Entwickler bei der Arbeit unterstützen. Unter anderen Zend, CakePHP oder Symphony.

Andere Programmiersprachen, wie Ruby, Java oder Perl⁶ ermöglichen ebenfalls Programme fürs Web zu schreiben und einem breiten Publikum zur Verfügung zu stellen. Darunter bekannte wie Ruby on Rails, Spring oder Jifty. Zwar liegt der Fokus auf Python, dennoch ist es wichtig andere Möglichkeiten der Webentwicklung für einen besseren Überblick kurz vorzustellen.

1.6.2 Zend

Das Zend Framework erweitert die Kunst und den Geist von PHP, dabei basiert es auf Einfachheit, beste praktische Vorgehensweise, freundliche Lizenzierung für Firmen und einer gründlich getesteten Codebasis. Dabei liegt der Fokus des Zend Frameworks auf sicherere und moderne Web 2.0 Anwendungen und Webservices. [9]

1.6.3 Ruby on Rails

Ruby on Rails ist eine OpenSource Webframework für die Zufriedenstellung der Webentwickler und kontinuierliche Unterstützung der Produktivität. Damit lässt es sich wunderbaren Code schreiben wobei das Design Paradigma Convention over Configuration⁷ angewandt wird.[10]

„Ruby on Rails is a breakthrough in lowering the barriers of entry to programming. Powerful web applications that formerly might have taken weeks or months to develop can be produced in a matter of days.“⁸

⁶http://www.socialtext.net/perl5/index.cgi?web_frameworks

⁷Convention over Configuration: Dabei ist das Design Paradigma gemeint, bei dem der Entwickler Regeln folgen muss um z.B. dass Primärschlüssel in der Datenbank Integer sein müssen oder Klassennamen und Tabellennamen gleich lauten müssen

⁸Tim O'Reilly, Founder of O'Reilly Media: <http://rubyonrails.org/quotes>

1.6.4 Catalyst

Das Catalyst ist ein in Perl geschriebenes MVC⁹ Webframework. Dieses Framework kann auf verschiedenen Plattformen installiert werden und ist für schnelle Webentwicklung geeignet.[11]

Bei der Entwicklung wird bedacht auch KISS¹⁰ gelegt, was dazu führt, dass Catalyst skalierbar und robust bleibt. Was sich bei den daraus entwickelten Applikationen bemerkbar macht.[12]

1.6.5 Wicket

Mit zweckmäßiger Aufteilung von Darstellung und Logik, ein POJO¹¹ Datenmodell und erfrischendem Verzicht von XML, macht Apache Wicket das Entwickeln von Webapplikationen wieder einfach und angenehm. Es wird mit unnötigem Code, komplexem Debugging und brüchigem Code gebrochen, um wiederverwendbare Komponenten, die nur auf Java und HTML aufbauen zu erstellen.[13]

1.7 Arbeiten ohne Framework

Die Erstellung einer Anwendung ohne ein entsprechendes Framework ist natürlich möglich doch oft sehr mühsam. Der Entwickler hat alle Freiheiten eine Anwendung zu schreiben, ihm wird aber mit einem guten Framework vieles erleichtert. Webframeworks bringen Unterstützung in Aspekten, wie z.B. Sicherheit, Benutzerverwaltung und Applikationsadministration. Diese müssen, falls benötigt von Grund auf immer wieder neu geschrieben werden und lenken somit von der eigentlichen Entwicklung der Kernapplikation ab.

Die meisten Frameworks arbeiten nach dem Prinzip „Don't repeat your self“ oder „KISS“ keep it simple stupid. Somit werden immer wiederkehrende Aufgaben, die mit komplexen Projekten verbunden sind bereits vorgefertigt geliefert und der Entwickler kann sich sofort der eigentlichen Kernaufgabe widmen ohne sich um nötigen Boilerplate kümmern zu müssen.

⁹MVC: Model View Controller Design Muster <http://www.oracle.com/technetwork/java/mvc-140477.html>

¹⁰KISS: Keep it simple stupid <http://techcrunch.com/2009/04/28/keep-it-simple-stupid/>

¹¹POJO: Plain Old Java Object <http://www.martinfowler.com/bliki/POJO.html>

Literaturverzeichnis

- [1] Farid Hajji, Das Python Praxisbuch - Der große Profi-Leitfaden für Programmierer, 2008
- [2] Python Beschreibung, <http://www.python.org/about/>, 28.07.2010
- [3] phpwact.org, http://www.phpwact.org/php/mvc_frameworks, 28.07.2010
- [4] Python Webframeworks Übersicht , <http://wiki.python.org/moin/WebFrameworks>, 28.07.2010
- [5] Pylons, <http://pylonshq.com/>, 29.07.2010
- [6] Django, <http://www.djangoproject.com/>, 29.07.2010
- [7] zope, <http://www.zope.de/uber-zope>, 29.07.2010
- [8] web2py, <http://www.web2py.com/examples/default/what>, 29.07.2010
- [9] zend, <http://framework.zend.com/about/overview>, 30.07.2010
- [10] Ruby on Rails, <http://rubyonrails.org/>, 30.07.2010
- [11] perl.net.au, <http://perl.net.au/wiki/Catalyst>, 30.07.2010
- [12] Catalystframework, catalystframework.org/, 30.07.2010
- [13] Wicket Webframework, catalystframework.org/, 30.07.2010