

# Swift On Linux/Conclusion/

Aus SwiftWiki

## Inhaltsverzeichnis

- 1 Conclusion
  - 1.1 Die Sprache
  - 1.2 Das Framework
- 2 Einzelnachweise

## Conclusion

Mit dem Ziel die beste Sprache der Entwicklung zu werden, hat sich Swift ein sehr hohes Ziel gesteckt, dass beinahe nicht zu erreichen sein wird. Bei der Entwicklung des Swift Servers sind einige Probleme aufgetaucht, die wir hier in unserer subjektiven Meinung beschreiben möchten.

## Die Sprache

Unter anderem ist uns aufgefallen, dass die Syntax anfangs ungewohnt ist und etwas Zeit in Anspruch nimmt, sich mit dieser vertraut zu machen. Die Dokumentation bzw. die Guides, die auf Swift.org oder auch auf anderen Seiten zur Verfügung gestellt werden, sind eine Unterstützung, jedoch kommt man leider nicht darum herum, sehr vieles mit "try and error" auszutesten. Sehr hilfreich war der auf Apples Developer Seite zu findende Language Guide, welcher die Basics und auch viele fortgeschrittene Themen behandelt. Für Kontrollfluss-Funktionen gibt es die Möglichkeit z.B. Initialisierung, Condition-Überprüfung und "else"-Block in eine Zeile zu schreiben. Leider wird der Code dadurch schwer lesbar und für Entwickler anderer Sprachen beinahe unmöglich zu lesen, sollte keine Zeit zum Investieren vorhanden sein.

## Das Framework

Zum verwendeten Framework ist zu sagen, dass es bereits sehr gute Ansätze für die Abhandlung von Request und Response bereithält, jedoch noch einen sehr weiten Weg zum Referenzprodukt Node.js hat. Das Handling im Allgemeinen ist sehr unausgereift und muss mit vielen kleinen Workarounds dazu gebracht werden, das Verhalten von Node.js zu imitieren. Leider verführt das vorgeschlagene Requesthandling dazu, dass viele Methoden angelegt werden, die alle das gleiche Verhalten aufweisen, jedoch auch nur schwer refactored werden können. Leider gab es für Perfect auch keinen Beispiel-Server mit mehreren Seiten und Funktionen, der veranschaulicht, wie die Entwickler des Frameworks sich die Umsetzung bzw. die Verwendung des Frameworks gedacht haben. Bei den vorhandenen Beispielen war oft die Dokumentation unzureichend, sodass die Anwendung verschiedener Packages in der begrenzten Zeit nicht möglich war. Als Beispiel lässt sich das Perfect-Crypto Package anführen, wo die Beschreibung nicht reichte, um ein Passwort zu verschlüsseln. Dazu wurde von uns das Packet SwiftyBeaver nach längerem Studieren der Klassen und Methoden von Perfect-Crypte als Ersatz verwendet. Einige Male mussten die Packages selbst analysiert werden, um zu verstehen, wie die Pakete angewandt werden, was natürlich wiederum zu einem besseren Verständnis führte, jedoch eine Menge Zeit verschlang. Grundsätzlich ist das Framework weiter zu empfehlen, jedoch muss bis zum sicheren Anwenden des Frameworks sehr viel Zeit und Ärger investiert werden.

Abschließend bleibt noch zu sagen, dass diese Sprache für sich genommen eine sehr schöne und lesbare Sprache ist, wenn nur wenige Kurzschreibweisen verwendet werden. Der Ansatz der Sicherheit und der daraus schließenden "absoluten Definition" ist nachvollziehbar und auch sehr vorteilhaft, da man bei jeder Methode und Funktion das Verhalten genau erkennt. Jedoch ist es bis zum Einholen des Node.js Frameworks noch viel Arbeit, da das Node.js Framework ausgereifter und mehr Funktionen in Form von Modulen bietet. Leider ist Perfect für die Version 3.0 entwickelt worden, der Server bereits auf 3.1 entwickelt und während der Entwicklung Version 4.0 von Swift veröffentlicht worden, wodurch sich Fehler in der Dokumentation eingeschlichen haben.

## Einzelnachweise

Abgerufen von „[http://192.168.42.2/mediawiki/index.php?title=Swift\\_On\\_Linux/Conclusion/&oldid=103](http://192.168.42.2/mediawiki/index.php?title=Swift_On_Linux/Conclusion/&oldid=103)“

---

Diese Seite wurde zuletzt am 1. August 2017 um 20:02 Uhr bearbeitet. Der Inhalt ist verfügbar unter der Lizenz GNU-Lizenz für freie Dokumentation 1.3 oder höher, sofern nicht anders angegeben.