

# Dokumentation – Github und Parsor

Christopher Sauer

15. Juli 2014

## Inhaltsverzeichnis

1	Einführung	2
2	Fehlermeldungen und Co.	3
3	Aktuelle Parsorversion herunterladen	4
4	Automatischer Aufruf aus Matlab	5
5	Links	6

## Abbildungsverzeichnis

1	Github Issues . . . . .	3
2	Parsor Download . . . . .	4

# 1 Einführung

Um den Quellcode für Parsor besser zu verwalten, wurde dieser komplett auf die Opensource-Plattform Github [5] hochgeladen. Dies bringt folgende Vorteile:

- Einfache Einsicht in den Code
- Versionsvergleiche
- Zusammenarbeit am eigentlichen Code
- Verwaltung von Fehlermeldungen und generellen Problemen
- Anregungen und Wünsche für neue Funktionen

Der Quellcode ist unter <https://github.com/Balagrio/Parsor> erreichbar. Dieser wird mit dem Versionsverwaltungssystem `git` [1] verwaltet. Über die Kommandozeile oder verschiedene grafische Oberflächen wie die offizielle Windows Github-App [4], die App für Macintosh [3] oder ein Github Eclipse Plugin [2]. Bei allen Anwendungen wird die `git` Kommandozeilen-Anwendung mitgeliefert und Github setzt darauf jeweils mit einer grafischen Oberfläche auf. Wer nur das reine `git` verwenden möchte, dem empfehle ich die Kommandozeile oder die grafische Oberfläche `mysysgit` [6], die sich auch über das Windows-Rechtsklickmenu bedienen lässt.

## 2 Fehlermeldungen und Co.

Ein weiterer Vorteil ist, dass nach einer kurzen Registrierung Fehlerberichte und Co unter dem Reiter "Issues" in Github geschrieben werden können, dann kann ich mich dran machen diese Fehler auszumerzen oder neue Funktionen zu implementieren. Fragen, Wünsche, Anregungen gerne auf Github stellen. Auf diese Funktion könnt ihr nach erfolgreicher Registrierung auf der Hauptseite [5] zugreifen, siehe Abbildung 1.

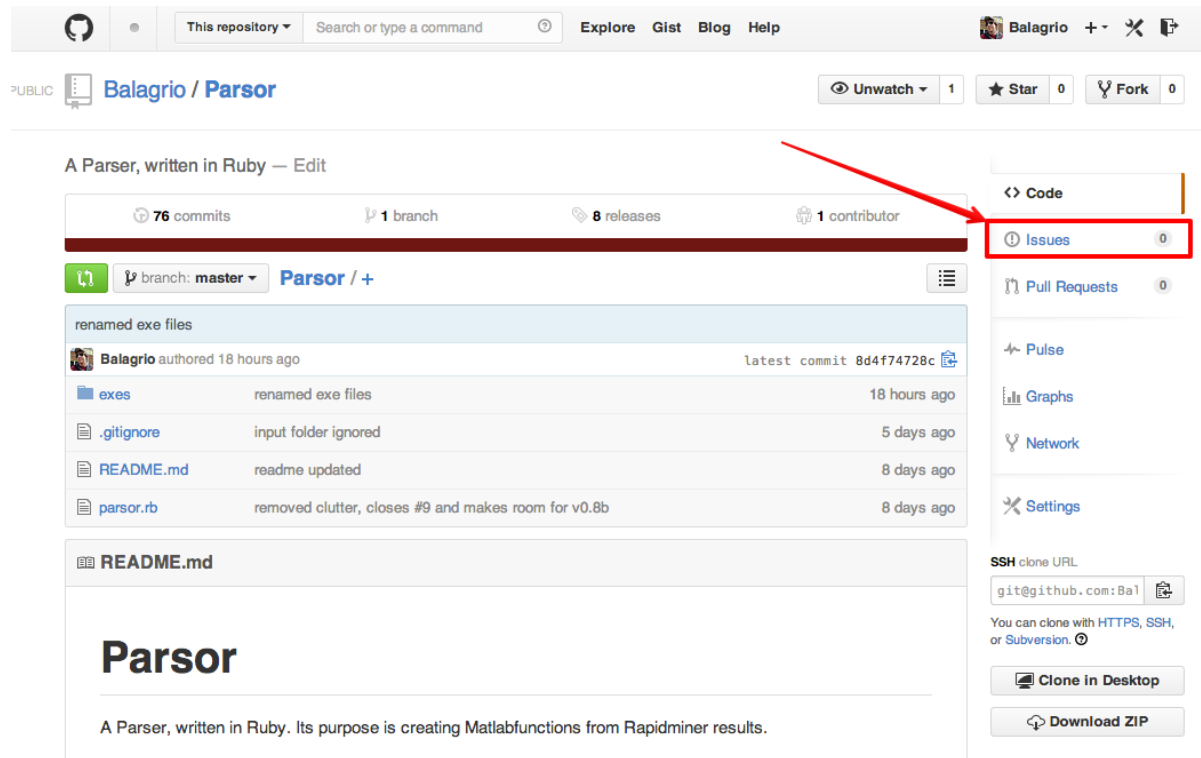


Abbildung 1: Github Issues

Ich hoffe, dass sich das ganze über Github besser verwalten lässt und Fehler und Co. schneller behoben werden können.

### 3 Aktuelle Parsorversion herunterladen

Die jeweils aktuelle Parsorversion kann unter [7] heruntergeladen werden, siehe Abbildung 2.

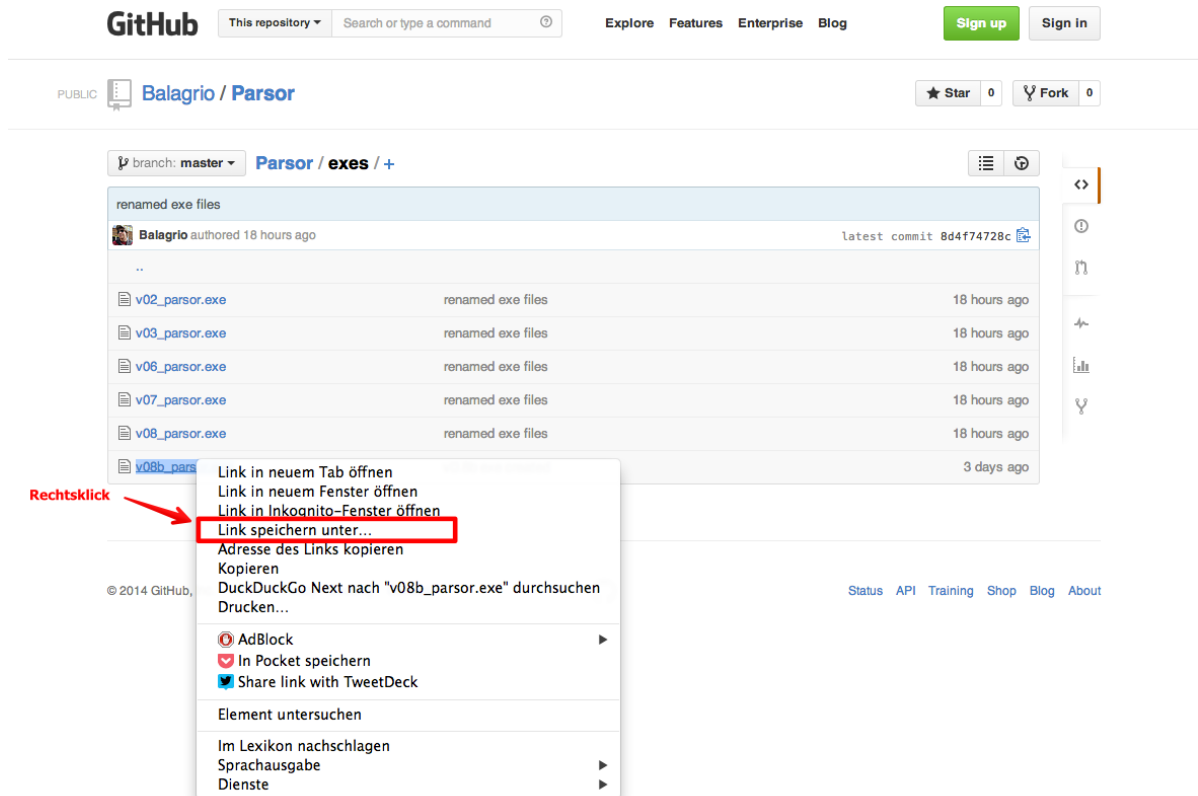


Abbildung 2: Parsor Download

Ich würde mich freuen wenn wir über Github einen Weg finden, Parsor weiter zu verbessern.

## 4 Automatischer Aufruf aus Matlab

Über die `runparser.m` Datei lässt sich der Parser auch automatisch aus Matlab aufrufen, dabei können folgende Optionen gesetzt werden

1. Name der Parser-exe
2. Verwendetes Verfahren: LR, M5P, M5R, PR
3. Dateiname der Eingangsdaten
4. Vorhersage Label, bei M5P, M5R
5. Erster Buchstabe der Variable
6. Funktionsname

Ein Beispielaufruf kann sein: `runparser('parser.exe','LR','lr.txt','','X','testfunc')`.

## 5 Links

- [1] *Git Versionsverwaltung*. <https://git-scm.com/>
- [2] *Github Eclipse Plugin*. <https://eclipse.github.com/>
- [3] *Github for Mac*. <https://mac.github.com/>
- [4] *Github for Windows*. <https://windows.github.com/>
- [5] *Github Hauptseite*. <https://github.com/>
- [6] *Msysgit*. <http://msysgit.github.io/>
- [7] *Parsor Downloadseite*. <https://github.com/Balagrio/Parsor/tree/master/exes>