

Bevier F.F.

Das japanische Juwel

Ein schneller interaktiver Debugger

Die von Rails unterstützte „testgetriebene Entwicklung“ ([TDD test-driven development](#)) senkt den Bedarf an individuellem Debugging zwar gewaltig, aber gerade für Einsteiger kann es doch sehr nützlich sein, sich das Verhalten einzelner Code-Stücke oder Funktionen ausführlichst ansehen zu können.

Das war bis vor gar nicht langer Zeit freilich keineswegs erfreulich. Sicher funktionierte der mit Ruby mitgelieferte Standard-Debugger einwandfrei und zeigte auch mit der Standard-Vorgehensweise (ein „breakpoint“-Statement im Code einzufügen) ein akzeptables Laufzeitverhalten, für komplexe Variable war die Kommandozeilen-Optik des angebotenen Fensters jedoch nicht wirklich geeignet.

Deshalb bieten praktisch alle integrierten Entwicklungsumgebungen (IDE) interaktive Debugger an, mit denen Breakpoints leicht gesetzt werden können und Variable mit ihren aktuellen Inhalten übersichtlich aufgelistet werden. Auch für Ruby (und damit für Rails) existieren solche graphischen Oberflächen für Debugger längst, sie mussten bisher jedoch auf den Standard-Debugger zurückgreifen, der unter diesen Bedingungen nicht wirklich schnelle Ergebnisse lieferte.

Das hat sich mit „[ruby-debug](#)“ nun geändert, einem flotten Debugger für Unix und Windows, zu dem es mit „`ruby-debug-ide`“ auch ein [RubyGems](#)-Projekt zur Integration in IDEs gibt. Mit dem Befehl

```
gem install ruby-debug-ide
```

lässt sich dabei nicht nur die Anbindung an die IDEs herunterladen, sondern natürlich bei Bedarf auch der entsprechende Debugger.

Für Anwender von [Aptana/RadRails](#) ist die Sache damit bereits völlig erledigt, denn diese IDE hat den flotten Debugger bereits als Standard in ihren Präferenzen aufgenommen, falls „`ruby-debug`“ installiert ist.