IOP Conference Series: Materials Science and Engineering

PAPER • OPEN ACCESS

A Comprehensive review on the open source hackable text editor-ATOM

To cite this article: K Sumangali *et al* 2017 *IOP Conf. Ser.: Mater. Sci. Eng.* **263** 042061

View the article online for updates and enhancements.

A Comprehensive review on the open source hackable text editor-ATOM

K Sumangali, Lokesh Borra and Amol Suraj Mishra

School of information technology engineering, VIT University, Vellore - 632014, Tamilnadu, India

E-mail : ksumangali@vit.ac.in

Abstract. This document represents a comprehensive study of "Atom", one of the best open-source code editors available with many features built-in to support multitude of programming environments and to provide a more productive toolset for developers.

1. Introduction

Atom is a free and open-source text and source code editor for OS X, Linux, and Windows with support for plug-ins written in Node.js, and embedded Git Control, developed by GitHub [1]. Atom is a desktop application built using web technologies. Most of the extending packages have free software licenses and are community-built and maintained. It is written in Coffee Script and Less. It can also be used as an Integrated Development Environment (IDE).

Atom is based on Electron, a framework that enables cross-platform desktop applications using Chromium and Node.js. This in a way allows Atom to be very flexible and "hackable" as the developers call it. Atom is a testament to the power of the Electron API and is a fast, customizable alternative to many other popular code editors. Users can easily extend it by developing rich packages or by editing configuration files that are open for modification.

2. Features

2.1. Hackable and Customizable Features

In Atom, all built-in features are implemented as packages. So, anyone can add their own packages on top of the existing stack or remove all the built-in packages to make their own custom version of Atom. This allows a great amount of flexibility and already, many companies are modifying Atom to fit their development process. In many ways, Atom can be converted into a lightweight IDE. Developers can make packages that extends Atom and publish it the Atom packages repository so that every user can install it through APM.

2.2. Fuzzy Finder

Atom indexes your current project and makes it easy to search from the Fuzzy Finder for any file you want [2]. Mind you unless you tell the settings for filetree to ignore folders like node modules or vendor you can end up with slow searches or sluggish performance due to the number of files typically present in these folders.

2.3. Tree View
View and open files in your current project. Provides many common actions like cutting and pasting folders, copying the project path to a file, or opening a file up in your native file explorer.

2.4. Built-in Autocompletion support for various programming languages
Atom has built-in autocomplete packages that are implemented via another package called autocomplete-plus which gives useful features like code completion, code linting, syntax highlighting and code file management. Hence, it is very easy to get started with Atom no matter which language you work with — all the features all built-in from the get go.

2.5. In-built Git support
Atom was made by GitHub and they made sure that Atom contained features that will make working with a Git repository as simple as possible. In Atom, visual hints will be displayed to indicate stats like the number of lines added or deleted from a file, the current working branch, colored bars next to lines added or removed and color of the filenames changing to yellow or green indicating whether the files were modified or added to the index. The git-plus package can be added to Atom to add even more Git features such as performing git add, git commit, git push, git log from a GUI interface.

2.6. Theming
The Atom Package Repository contains many user-made themes as well as official Atom themes to make Atom look and feel different. These can be added to suit particular development processes or to fit the user's needs. The themes can be customized thereafter using configuration files that are easy to understand and modify. In case of any errors, the user can always revert back to a known good state. Since Atom is hackable, every single component of the editor can be customized using the theme file to change the colors, fonts or to add new features. The Atom website provides a complete API [3] package so that users can hack it and write their own packages, themes or publish a modified version of the editor itself.

2.7. Support for multiple panes
In Atom, you can create multiple panes using keyboard shortcuts or from the menu bar. This allows you to view and work on multiple files at the same time. Say, in a web development project, one pane can be used to edit the HTML file, the next for the main CSS file and the last for the main JavaScript file. Atom also has keyboard shortcuts so that you can quickly switch between open panes.

2.8. Package Manager
The Package Manager is perhaps the most powerful tool in Atom. It allows users to download and install packages that other developers have published in order to extend the functionality of the base editor. You can also make your own packages utilizing the Atom API and publish packages and themes via the package manager to the Atom Package Repository. The package manager is accessible both through the Settings pane in Atom or through the command-line utility, apm.

2.9. Atom is cross-platform
Since Atom is built on Electron, it works on all the major OS platforms including Windows, Mac and Linux. Hence, it allows for cross-platform development and your development configurations and settings can be carried over to other Atom installations manually or through the atom-sync [4] package which makes use of GitHub to keep settings and user preferences in sync across Atom installations.

3. Comparative Study

In this section, we compare Atom with three of the most popular code-editors available as of today — Sublime Text, Brackets and Visual Studio Code [5]. Sublime Text is the currently the king of code-editors as it has amassed a huge amount of popularity over the last years. It is a proprietary code-editor which has several built-in features such as syntax highlighting and language support for multifarious languages. Atom's key advantage over Sublime Text is it's open-source nature which leads to bugs getting fixed and new features getting rolled out in quick successions which is very important for a code-editor. Comparatively, Sublime Text releases are very slow taking years or months to release even though the developers were able to make it very stable for the better part of the last half-decade. Sublime Text lacks an official package manager even though a Package Manager is available for install. However, Atom comes built-in with one which like the unofficial Package Manager, hosts thousands of packages and is quickly increasing in population. This is a big number taking account the newness of Atom. Sublime Text is difficult to configure and get started with for newcomers while Atom is quick and easy to setup with a rich interface for installing and managing packages.

Brackets is made by Adobe and is an open-source code editor built for web developers. It in-fact, has a lot of useful built-in features that supports rapid web application development such as live-preview and integration with other Adobe applications so that assets can be imported or work can be done in parallel. However, the fact that Brackets has web-development specific features makes it better just in the realm of web development. Outside that, it's features are limited to none. Atom has packages that provide the same functionality and with a near equal amount of robustness which makes it as competitive. Brackets' performance is not as great when you compare it with Atom and the extra number of features and customization options are limited.

Figure 1. Atom editor with an open project

Table 1. Comparison of the text editors

	Atom	Brackets	Visual Studio Code	Sublime text
Released on	June, 2015	November, 2014	April,2015	January,2008
Developer	GitHub Inc	Adobe	Microsoft	Jon Skinner
Supported by	OS X 10.8 or later, Windows 7 and later, RedHat Linux and Ubuntu Linux [3]	macOS, Windows and Linux	Windows 7 SP1 and later Windows Server 2008 R2 SP1and later[5]	Linux 32/64-bit, Mac OS X 10.6 or later (version 2), 10.7 or later (version 3), Microsoft Windows32/64-Bit
Written in	Electron (CoffeeScript / JavaScript / Less / HTML)	HTML, CSS and JavaScript	Electron (HTML / CSS / JavaScript)	C++, Python
License	MIT License (free software)	MIT License	MIT License	Proprietary Software
Nature	Open Source	Open Source	Open Source	Proprietary
Inbuilt Git Features	Yes	No	Yes	No
Primary focus	A code-editor that is modular and can be hacked into a full-fledged IDE	A code-editor aimed solely for front-end web development	A code-editor that is flexible but mainly targeted at .NET Developers	A code-editor for all sorts of programming
Cross platform development	Yes	Yes	Yes	Yes
Extension support	Yes	Yes	Yes	Availed with Package Manager

Visual Studio Code is the newcomer among the four and like Brackets is geared towards a particular domain — .NET application development. For that, Microsoft has provided a number of built-in utilities like it's hallmark Intellisense feature for code-completion, syntax-highlighting and inline documentation. It also has built-in Git features just like Atom which also is a big advantage for developers starting out with Code. Emmet is built-in to Code which provides quick code insertion for HTML and CSS files. Like Atom, Code is built using Electron which makes it cross-platform. However, Code hasn't gained much steam unlike Atom and has a stale repository of packages compared with that of Atom's. Code's emmet integration is incomplete while Atom has an Emmet package available to install through apm which has the complete set of features. Atom is much more hackable than Code which makes it a more compelling choice. While Code succeeds in attracting .NET developers, the vast majority will find Atom more suited for their work.

4. Conclusion

Atom has got many added features which are nott available in other text editors like Atom has the Command Palette, Developer Tools, Integrated Package Manager, Easy to customize the UI, a LOT of support from GitHub crew (You can see these guys pushing to Atom repo's all day long), Fuzzy finder for quickly opening files and above all its feels really easy to use. That said it has got a few shortcomings too like Slow when compared to Sublime Text 3 which is insanely fast, we need to use coffee script to write the packages (Not sure if it is a con, because most developers out there are Python guys), It cannot handle files bigger than a couple MB's, There is no Corona Package yet etc. Hence we conclude that atom has the potential to become the best editor of this world if it manages to overcome it shortcomings.

References

[1] Atom (text editor) 2017 — https://en.wikipedia.org/wiki/Atom_(text_editor).

[2] Michael C J 2016 Best of Atom: Features, Plugins, Acting Like Sublime Text — https://scotch.io/bar-talk/best-of-atom-features-plugins-acting-like-sublime-text

[3] Bottos B A and Kintala C M R 2016 Generation of syntax-directed editors with atom-text editor- https://discuss.atom.io/t/syntax-highlighting-punctuation/16254

[4] Srivastava A and Thiagarajan J 2002 Effectively prioritizing tests in development environment. *In ACM SIGSOFT Software Engineering Notes* **27** 97-106 ACM.

[5] Cemerlang P , Lim J H, You Y, Zhang J and Chevallet J P 2006 Towards automatic mobile blogging. *International Conference on Multimedia and Expo* 2033- 2036 IEEE.

[6] Clark T H, Malisewski S C, Cooper P R, Crosswy W C, Crochet L J 2001 System for automatic synchronization of common file between portable computer and host computer via communication channel established with user approval of charge to be incurred. United States patent US 6,317,797.

[7] Brian Rinaldi 2017 Battle of the Free Code Editors – http://developer.telerik.com/featured/battle-of-the-free-code-editors/