Storing and Sharing Code Online

Shomiron Ghose Thomas Jefferson High School For Sci. and Tech.

Storing and Sharing Code Online

1 Introduction

In the last few years, the advent of online services that help you to code, check syntax and host code has created a large opportunity for coders. Among this is the opportunity to share code with a wide audience such as open source projects or with a smaller audience such as a team project for a competition such as Botball. These online services typically provide many services that would be difficult or at least time-consuming to implement by one's self and provide a great resource for any botball team.

2 Online Storing and Saving

2.1 Saving Code Online

Pastie.org

```
1 class HelloThere
2 {
3 public static void main(String[] args)
4 {
5 System.out.println("Hello World");
6 }
7 }
```

Figure 1

Pastebin.org



Figure 2

As a result of this, it is possible to save code and widely distribute it using a link. For example

the pastebin link above is <u>http://pastebin.com/Dhaixcp2</u>. This allows quick copying and is a useful tool for asking questions as well as sharing code. Furthermore many, such as pastebin allow you to limit the link from public to private (if it is visible with or without the url) and how long you would like it to be saved on the website.

2.2 Quick Code Collaboration

In recent years, many services have popped up that allow you to collaborate with other members of a team in real-time and without even logging in. They typically provide syntax highlighting (but not always, some as plaintext) and save your code for a certain amount of time. Examples of such websites are Figure 3 and Figure 4.



Figure 3

Collabedit.com

```
1 public class HelloWorld
2 {
3 public static void main(String [] args)
4 {
5 System.out.println("Hello World!");
6 }
7 }
```

Figure 4

2.3 Subversion and Quick Code Collaboration

When I see teams storing code and then sharing it via USB drives and not-so-commonly cd's, though I have seen that in general, I wonder why. I think there is a much better alternative which many teams are not aware of. This is web-based code hosting. A prime example of this is Github (Figure 5 shows the front page), a location where many open source projects such as have their code hosted.





Github offers many services that you may not have implemented or have the infrastructure for. An example of this is collaborating between users across the world and the ability to easily give access. Furthermore the chance of Github and it's various servers and mirrors crashing is far less especially when you consider it is hosted on Rackspace, one of the giants of the cloud hosting world comparable to Amazon's EC3 Service.

To users using public repositories - meaning anyone can see your code online, it is free while for private or corporate use it is a monthly cost. Git offers privilege levels to users as well as shared administration, wiki documentation and bug tracking. Even better, it offers comments for pull and push requests and allows users to see what is being modified and where the changes will happen. This is all in addition to versioning control which means if you accidentally add or delete code, the previous version will be saved and you can revert back. Even better, Github allows forking and merging of different branches that allows for different versions of the code to be worked on simultaneously while allowing comparing of different version of code. [1]

Accessing Github is not a complicated coding matter and if you're not comfortable with pulling, pushing and forking via git commands after ssh-ing a convenient step-by-step tutorial is offered. If you just do not like the barebones text interface there are GUIs that can access Github such as Tortoise Git, Eclipse, and Tower (specifically for Mac, example screenshot shown in Figure 6).

1 chunk, 9 insertions, 17 deletions Q × -	Co	mpare	Revision: ca4133cc3f 💌 to Revision:	af1dc64bee		(Diff Tool)
63 362 } 63 363 } 64 364 65 removeAttr: function(elem, value var propName, attrNames, name i = 0; timmywil Sep 19, 2011 66 var propName, attrNames, name i = 0; timmywil Sep 19, 2011 68 removeAttr: function(elem, name 366 timmywil Sep 19, 2011 70 attrNames = (value ""). 1 = attrNames [i]; 71 Check for property support in the boolHook before falling back to attribute node. Fixes #102 73 for (; i < l; i++) { name = jQuery.attrFix[iname = jQuery.attrFix[jQuery.attr(elem, name elem.removeAttribute(name jQuery.attr(elem, name, " Sep 19, 2011 74 // See #9699 for explanti jQuery.attr(elem, name, " Rick Waldron Fixes typo Sep 19, 2011 73 // See #9699 for explanti jQuery.attr(elem, name, " Rick Waldron Sep 19, 2011 nem removal) 73 // See #9699 for explanti jQuery.attr(elem, name, " Sep 19, 2011 nem removal) 74 // See #9699 for explanti jQuery.attr(elem, name, " Rick Waldron Ad notes about #9699 fix Sep 19, 2011 73 // Set corresponding pr I75 Commits Load All	1 ch	unk, 9	insertions, 17 deletions	Q	×	-
63 363 }, 64 364 , 65 i removeAttr: function(elem, valutivation valutivati valutivati valutivation valutivation valutivati valutiv	-		00 -362,26 +362,18 00 jQuery.extend			
63 363 }, 64 364 , 65 immywil Sep 19, 2011 70 immywil Sep 19, 2011 70 fill emoveAttr: function(elem, name 70 if elem.nodeType === 1) { 71 if elem.nodeType === 1) { 72 if if 73 if if 74 name = jQuery.attrFix[75 name = jQuery.attrFix[76 if 77 if 78 if 79 if 71 if 72 if 73 if 74 if 75 if 76 if 77 if 78 if 79 if 70 if 71 if 72 if 73 if 74 if 75 if 76 if 77	62	362	}	timmywil	Sep 20, 2011	-
64 364 55 removeAttr: function(elem, value i = 0; 365 removeAttr: function(elem, name i = 0; 366 var propName; 9 attrNames = (value "'). 1 = attrNames.length; 71 1 = attrNames.length; 72 - 73 for (; i < 1; i++) { name = jQuery.attrFix[nam removeAttribute (name i gQuery.attr(elem, name i gQuery.attr(elem, name, '' 76 369 77 /// See #9699 for explanti jQuery.attr(elem, name, '' 78 elem.removeAttribute(name 79 // See #9699 for explanti jQuery.attr(elem, name, '' 71 jQuery.attr(elem, name, '' 72 // See #9699 for explanti jQuery.attr(elem, name, '' 73 // Set corresponding pr 74 // Set corresponding pr	63	363		Landing null reg	jest 497 1 7 Remove multiple	
66 - var propName, attrNames, name i = 0; immywil Sep 19, 2011 Landing pull request 510. Add notes about #9699 fix. 67 - removeAttr: function(elem, name war propName; - 68 - if (elem.nodeType === 1) { attrNames = (value ""), 1 = attrNames.length; - 70 - attrNames.length; - 72 - - - 73 - for (; i < 1; i++) { name = jQuery.attrFix[add name = jQuery.attrFix[name = jQuery.attrFix[nam rg Sep 19, 2011 74 - name = jQuery.attrFix[add notes about #9699 for explanti jQuery.attr(elem, name elem.removeAttribute(n Rick Waldron Sep 19, 2011 - 74 - // See #9699 for explanti jQuery.attr(elem, name, elem.removeAttribute(name Rick Waldron Sep 19, 2011 - 73 - // See #9699 for explanti jQuery.attr(elem, name, elem.removeAttribute(name Rick Waldron Sep 19, 2011 - 73 - // Set corresponding pr - - 74 - // Set corresponding pr - 75 - - - 76 - - - 77 - // See #9699 for explanti jQuery.attr(elem, name,	64	364				
67 i = 0; Sep 19, 2011 68 i = 0; Immywil Sep 19, 2011 68 var propName; if (elem.nodeType === 1) { 66 attrNames = (value ""); if (elem.nodeType === 1) { 70 attrNames = (value ""); if (i = 1, i = attrNames.length; 71 1 = attrNames.length; Check for property support in the boolHook before falling back to attribute node. Fixes #102 73 for (; i < 1; i++) {	65	-	removeAttr: function(elem, value			
67 i = 0; Landing pull request 510. Add notes about #9699 86 removeAttr: function(elem, name fix. 866 var propName; fit (elem.nodeType === 1) (attrNames = (value ""). 70 if (elem.nodeType === 1) (attrNames = (value ""). 71 l = attrNames.length; 72 for (; i < 1; i++) (aname = attrNames[i];aname = jQuery.attrFix[aname = jQuery.attrFix[aname = jQuery.attrFix[aname = jQuery.attrFix[aname = jQuery.attr(elem, name = elem.removeAttribute(aname = elem.removeAtt	66	-	var propName, attrNames, name	timmywil	Sep 19, 2011	
365 removeAttr: function(elem, name 366 var propName; 69 367 368 var propName; 69 367 368 if (elem.nodeType === 1) { attrNames = (value ""). 1 = attrNames.length; 72 for (; i < 1; i++) { name = jQuery.attrFix[name = jQuery.attrFix[a68 368 name = jQuery.attrFix[name = jQuery.attrFix[a68 76 369 77 // See #9699 for explan jQuery.attr(elem, name elem.removeAttribute(n 70 // See #9699 for explanti jQuery.attr(elem, name, " elem.removeAttribute(name 80 373 81 // Set corresponding pr	67	-	i = 0;			
365 removeAttr: function(elem, name 366 var propName; 9 367 9 367 9 367 9 1 f (elem.nodeType === 1) { (attrNames = (value "). 1 1 = attrNames.length; 71 1 = attrNames.length; 72 - 73 - 74 - 75 - 76 10 78 - 79 - 76 369 77 - 78 - 79 - 70 - 71 - 72 - 73 - 74 - 75 - 76 369 77 - 78 - 79 - 71 - 72 - 73 - 74 - 75 - 77 -	68	-			Jest 510. Add holes about #9699	
69 367 if (elem.nodeType === 1) { attrNames = (value ""). 1 = attrNames.length; timmywil Sep 19, 2011 70 attrNames = (value ""). 1 = attrNames.length; Check for property support in the boolHook before falling back to attribute node. Fixes #102 72 - 73 for (; i < 1; i++) { name = jQuery.attrFix[a68 Check for property support in the boolHook before falling back to attribute node. Fixes #102 74 name = attrNames[i]; name = jQuery.attrFix[a68 Override Sizzle attribute retrieval with jQuery.attr(elem, name relem.removeAttribute(n 76 // See #9699 for explanti jQuery.attr(elem, name, relem.removeAttribute(name attribute (attribute about #9699 fix Nen removal) 77 // See corresponding pr Rick Waldron Add notes about #9699 fix removal) 80 373 // Set corresponding pr 175 Commits Load All		365 -	removeAttr: function(elem, name			
36 367 If ("effent nodes" = [value ""). 70 attrNames = (value ""). Check for property support in the bolHook before falling back to attribute node. Fixes #102 71 1 = attrNames.length; Check for property support in the bolHook before falling back to attribute node. Fixes #102 72 - for (; i < 1; i++) {		366 -			c 10 2011	
71 1 = attrNames.length; 72 - 73 - 74 - 75 - 76 369 77 - 78 - 370 - 371 - 370 - 371 - 372 - 80 370 77 - 78 - 9		367		,		
72 - for (; i < l; i++) {		-				
73 - for (; i < 1; i++) {			<pre>l = attrNames.length;</pre>	before failing ba	tk to attribute node. Fixes #102	
74 name = attrNames[i]; 75 name = jQuery.attrFix[368 name = jQuery.attrFix[368 name = jQuery.attrFix[369	72					-
75 name = jQuery.attrFix[iJQuery.attr.Fix[iIQuery.attr.Fix[timmywil	Sep 19, 2011	
368 name = jQuery.attrFix[nam 76 369 77 - // See #9699 for explan 78 - jQuery.attr(elem, name 79 - elem.removeAttribute(n 370 // See #9699 for explanti 371 jQuery.attr(elem, name, " 372 elem.removeAttribute(name 80 373 91 // Set corresponding pr		-		the state of the second		
76 369	75	-		jQuery.attr. Fixes	#5637, #7128, #9261, #9570,	
77 // See #9699 for explan jQuery.attr(elem, name Kick Waldron Sep 19, 2011 78 jQuery.attr(elem, name 79 elem.removeAttribute(n 370 // See #9699 for explanation 371 jQuery.attr(elem, name, " 372 elem.removeAttribute(name 80 373 81 // Set corresponding pr			name = jQuery.attrFix[nam			-
78 jQuery.attr(elem, name Fixes typo 79 elem.removeAttribute(n removal 370 // See #9699 for explanati Rick Waldron Sep 19, 2011 371 jQuery.attr(elem, name, " Add notes about #9699 fix 372 elem.removeAttribute(name ITS Commits 80 373 // Set corresponding pr		369	11 - "	Rick Waldron	Sep 19, 2011	
79 elem.removeAttribute(n. removal) 370 // see #9699 for explanati Rick Waldron sep 19, 2011 371 jQuery.attr(elem, name, " Add notes about #9699 fix 372 elem.removeAttribute(name 80 373 81 // Set corresponding provide to the second s				Fixes typo		nen removar)
370 - // See #9699 for explanationationationationationationationati						
371 jQuery.attr(elem, name, " Rick Waldron Sep 19, 2011 372 elem.removeAttribute(name Add notes about #9699 fix 80 373 81 // Set corresponding pre	79	370				
372 elem.removeAttribute(name Add notes about #9699 fix 80 373 175 Commits 81 // Set corresponding pre				Rick Waldron	Sep 19, 2011	removar)
80 373 81 // Set corresponding pr. 175 Commits Load All				Add notes about	#9699 fix	
81 - // Set corresponding pr	20		erem.removeAttribute(hame			
		373	// Set corresponding pre		(Load All)	1
				the second se	me = iQuery.propFix/ name	1 name) in elem

Figure 6 from [2]

If Github is not your thing, there are similar services provided by other websites such as Gitorious, Bitbucket and SourceForge and wikipedia actually gives a wonderful comparison of a wide selection of these http://en.wikipedia.org/wiki/Comparison_of_open_source_software_hosting_facilities

2.4 Online IDE (Integrated Development Environment)

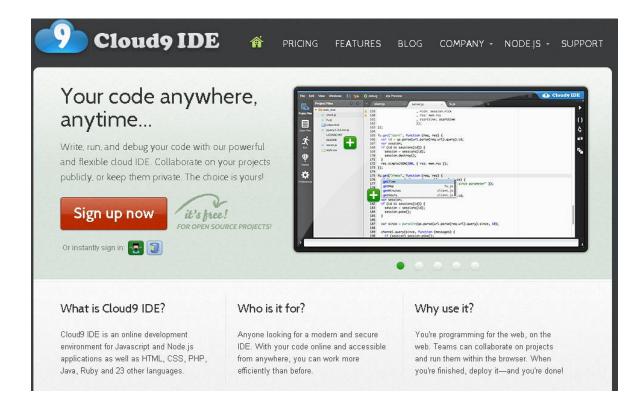


Figure 7 from [3]

Another resource teams can use if they're away from a computer is an online IDE such as Cloud9. It allows for code editing in multiple languages and even has a debugger! Some online IDEs also allow the compiling of small programs. Another service, codepad.org, though not an IDE, allows for the compiling and running of a small pieces of code and is useful for testing syntax.

3 Summary

In this paper I have discussed a few services that let you put your code online and share with others. This provides a degree of security, reliability and convenience that traditional local storage does not. The versioning features help observe progress and change over time and revert accidental changes. Even better, online services you can download your code from anywhere and share code snippets fast! As a result, I hope that all of you will consider using an online code hosting service in your next project!

References

1) GitHub Inc. "Features." *Github*. GitHub Inc., 2012. Web. 28 May 2012. <<u>https://github.com/</u><u>features/projects/>.</u>

2) Fournova GmbH. "Git Client for Mac." *Tower*. Fournova GmbH, 2011. Web. 28 May 2012. <<u>http://www.git-tower.com/>.</u>

3) Cloud9ide. "Cloud9 IDE." Cloud9 IDE. Cloud9ide, 2012. Web. 26 May 2012. ">http://c9.io/>.