

Developer Notes

Practical notes, advises on application development (java, seam, android, google api)

[Home](#)[About](#)

[Home](#) › [Software Development](#) › TinyMCE spellchecker java implementation

TinyMCE spellchecker java implementation

Posted on [August 11, 2009](#) by [Andrey Chorniy](#) — [38 Comments](#)

15 Votes

TinyMCE rich-editor have a plugin to provide Ajax-spell-checking functionality. See the [link to TinyMCE plugin documentation](#)

So, if you have a PHP and PSpell/ASpell on your server – then you are fine and you can use them, but this post is about the situation then you don't have them or want to use your own implementation. In that case we can configure “spellcheck” plugin to use our own service.

What we have to do is to create a webservice compatible with the “TinyMCE spellchecker plugin” and configure spellcheck with the URL to that service. Because of browser security reasons this service should be hosted on the same domain as the application which use it (ajax calls can be blocked if requested URL is on another domain then your page). And since we are using java then we can use open-source [Jazzy spellchecker library](#) to implement spell-checking functions required by the TinyMCE spellchecker plugin

Top Posts

- [How to launch android camera using intents](#)
- [How to use autofocus in Android](#)
- [JPA and stored procedures](#)
- [Howto avoid concurrent call to conversation problem in Seam](#)
- [TinyMCE JMySpell and Jazzy spellchecker implementation](#)

a4j [ajax](#) [ajax-push](#) [android](#) [android](#)

```

1 tinyMCE.init({
2     theme : "advanced",
3     mode : "textareas",
4     plugins : "spellchecker",
5     theme_advanced_buttons3_add : "spellchecker",
6     spellchecker_languages : "+English=en-us,Swedish=sv",
7     spellchecker_rpc_url : "/spellchecker/jazzy-spell-check
8 });

```

In the configuration example above we set the link to our splechecker-service-url with the “spellchecker_rpc_url : “/spellchecker/jazzy-spell-check;”
 “/spellchecker/jazzy-spell-check” will be served by the servlet which use Jazzy project.
 TinyMCE spellchecker plugin send JSON request to spellchecker_rpc_url with two possible methods:

1. **checkWords** – send the array of words as params attribute and e
 JSONArray of misspelled words in the result JSON-response at
2. **getSuggestions** – send the checked word as the first and only el
 params attribute and expects the JSONArray of suggested words
 JSON-response attribute.

So, when user click on “check spellcheck” button, plugin get the list of them as the JSON request, with two parameters “method”=“checkWor
 “params”=“JSONArray(words)”. Servlet return the list of misspelled wc
 response “{‘id’:null,‘result’: JSONArray(misspelled-
 words),‘error’:null}” , misspelled words are highlighted and onc
 one of them – request to get the list of suggested words is send to the
 “method”=“getSuggestions” and “params”=“JSONArray(checked-word
 the “{‘id’:null,‘result’: JSONArray(word-suggestions),‘error’:null}”
 JSON-response and list of suggestions is displayed to the user.

Update: You can see the backend java source code at the [jspellchecker](#) sourceforge project. SVN is accessible via “svn co

layout optimization android sources api
 asynchronous asynchronous data processing
 camera classification collaboration
 database did-you-mean eclipse email
 example Excel export feed free fulltext-
 search gdata gdata api gmail google
 google I/O hibernate-search
howto icefaces imap

inputNumberSpinner issue tracker **java**
 javascript jazzy jboss JPA jquery jsf
 json json feed json generation jxls

Follow

Follow “Developer Notes”

Get every new post delivered to your Inbox.

Join 32 other followers

Enter your email address

Sign me up

Build a website with WordPress.com

natural language
 e meeting
 ce predicion project
 lmine reporting
 sdk **seam** security
 :oftware source code
 :ker SQL ssl ssl certificate
 › suggestion engine
 ification technology
 :at tools usability **web**
lication

<https://jspellchecker.svn.sourceforge.net/svnroot/jspellchecker> jspellchecker” or you can [browse source-code](#) . Please take a notice that source-code doesn’t include dictionaries. The dictionaries are located in the /WEB-INF/dictionary/\${lang}-\${country-code}.

Special thanks to the Rich Irwin who initially provide me the code.

The quality of spell-checking with jazzy depends on the quality of your dictionary. I think it possible to get the list of words from [Mozilla Dictionaries](#) by saving XPI and opening it as ZIP file, saving dictionary file and remove “/xyz” from the end of the words, but please check the licensing notice before. You can also extract dictionaries from aspell (have no details how to do it)

Integration steps (thanks to [Adam-Musial Bright](#))

1. Check out / download
svn co <https://jspellchecker.svn.sourceforge.net/svnroot/jspellchecker/trunk>
2. cd trunk/tinymce-spellchecker
3. svn export web ~/temp/jspellchecker
4. svn export build/classes/org ~/temp/jspellchecker/WEB-INF/classes/org
5. Now deploy it on your webapp server
6. Now put something like this into your tinyMCE.init:
spellchecker_languages : “+English=en,German=de”,
spellchecker_rpc_url : “/jspellchecker/google-spellchecker”
possible values are

- “/jspellchecker/jmyspell-spellchecker”
- “/jspellchecker/jazzy-spellchecker”
- “/jspellchecker/google-spellchecker”
- “/jspellchecker/lucene-spellchecker”

Update: To integrate TinyMCE-4 spellchecker with your server-side see
<https://achorniy.wordpress.com/2013/05/27/tinymce-4-spellchecker-integration/>

[About these ads](#)

Share this:



Be the first to like this.

Related

[TinyMCE JMySpell and Jazzy spellchecker implementation](#)
In "Software Development"

[TinyMCE 4 Spellchecker integration](#)
In "Software Development"

[Lucene spellchecker for TinyMCE](#)
In "Software Development"



About [Andrey Chorniy](#)

<https://www.facebook.com/achorniy>

[JBoss SSL configuration](#)

[Howto delete large emails from GMail](#)

Tagged with: howto, java, jazzy, json, service, spellchecker, tinymce, web

Posted in Software Development