Using super-zaje to extract code from images

`zaje` is a syntax highlighter that aims to cover all your shell colouring needs. It can act as an ad-hoc replacement for cat and, with a spot of one-line shell functions tail and other friends.

I'm pleased to announce the release of `super-zaje` which can do everything `zaje` can but introduces one very useful enhancement: thanks to the magic of `tesseract` and **gosseract**, `super-zaje` is able to extract text right off an image (both HTTP(s) URLs and local files are supported).

The next slides demonstrate its usefulness with a few sample LinkedIn posts that shared code as screenshots (a common case, alas, due to LI's inexplicable refusal to support `markdown`).

As an aside note, this presentation was written entirely in VIM and converted into a PDF using **mdtopdf**: a project by Cecil New, to which I've contributed syntax highlighting, powered by my **gohighlight project**, which `zaje` and `super-zaje` also leverage. If you find yourself struggling to create similar presentations without using GUI tools, give it a go:)

First example: posts by Dr. Shlomi Boutnaru

Shlomi writes a lot of interesting presentations and articles about Linux, with emphasis on kernel internals. If you're interested in that topic, I strongly recommend following him.

This screenshot is taken from this **post**:

Using the below command, I converted it to this highlighted text, all in the comfort of my terminal:

```
$ super-zaje -l sh
https://media.licdn.com/dms/image/D4D22AQHOFz1ACjMgKA/feedshare-shrink_800/0/168926
6338913?e=1694044800&v=beta&t=Tg4y2TI8MpGe9JiAHxnSaYMVfNtHaGYx_iQmaJb6d18
```

Troller Session
Troller $ ls -l /proc/$$/fd/3
is: cannot access '/proc/23879/fd/3': No such file or directory
Troller $ exec 3<> /dev/nuLL *open td #3 for /dev/null
Troller $ ls -l /proc/$$/fd/3
rwx------ 1 user user 64 Jul 14 00:42 /proc/23879/fd/3 -> /dev/null
roller $ ls -li /proc/$$/fd/3
510793 Lrwx 1 user user 64 Jul 14 00:42 /proc/23879/fd/3 -> /dev/null
Troller $ ls -li readlink /proc/$$/fd/3
IS crw-rw-rw- 1 root root 1, 3 Jul 10 11:33 /dev/null *we <2 see that the inode number of
Troller $ echo $$ sa
23879
~ Troller Session -ox
File Actions Edit View Help
Troller Session
Troller $ sudo bpftrace -e 'kfunc:schedule { printf ("%d:%d:%s\n",cur
task->files->fdt->fd[3]->f_inode->i_ino,curtask->pid,curtask->comm) ;
}' | grep 23879
I5:23879:bash_ sthe inode number of fd #3 of bash is 5

Another example from a post by Shlomi:

```
[troller@localhost test]$ ls -lah ./troller
-rwxrwxrwx 1 root root 8 Jul 11  2023 ./troller
[troller@localhost test]$ cat ./troller
TrolLer
[troller@localhost test]$ rm ./troller
rm: cannot remove './troller': Permission denied
[troller@localhost test]$ exit
exit
root@localhost:/tmp/test# chmod o+w /tmp/test
root@localhost:/tmp/test# chmod 000 ./troller
root@localhost:/tmp/test# su troller
[troller@localhost test]$ ls -lah ./troller
---------- 1 root root 8 Jul 11  2023 ./troller
[troller@localhost test]$ cat ./troller
cat: ./troller: Permission denied
[troller@localhost test]$ rm ./troller
rm: remove write-protected regular file './troller'? y
[troller@localhost test]$ ls -lah ./troller
ls: cannot access './troller': No such file or directory
```

$ super-zaje -l sh
https://media.licdn.com/dms/image/D4D22AQH7m3jilXSHoQ/feedshare-shrink_800/0/1689137839765?e=1694044800&v=beta&t=fLX7NPx1kMr4CmlpQ3leUqWB3xca12Pw4EyA6XxcELE

Will output...

```
[troller@localhost test]$ ls -lah ./troller
-ruxruxeux 1 root root 8 Jul 11 2023 ./troller
[troller@localhost test]$ cat ./troller

TrolLer

[troller@localhost test]$ rm ./troller

rm: camot remove './troller': Permission denied
[troller@localhost test]$ exit

exit

root@localhost:/tmp/test# chmod o+u /tmp/test
root@localhost:/tmp/test# chmod 000 ./troller
root@localhost:/tmp/test# su troller
[troller@localhost test1$ 1s -lah ./troller

---------- 1 root root 8 Jul 11 2023 ./troller
[troller@localhost test]$ cat ./troller

cat: ./troller: Permission denied

[troller@localhost test]$ rm ./troller

rm: remove urite-protected regular file './troller'? y
[troller@localhost test1$ 1s -lah ./troller

ls: cannot access ./troller: No such file or directory
```

As you can see, it's not always perfect; for example, notice the **rm: camot remove './troller: Permission denied** bit, where it should clearly be "cannot remove". I also had to add a closing `'` around `./troller` to the original text `super-zaje` extracted to get proper highlighting; still, it's not bad:)

Important notes:

`zaje` can detect the lexer to use based on:

- The file name (when acting in `cat` mode)
- The first line of text (so it will usually work nicely when piping as well)

When operating on an image, the file name will of course be of no use and the first line will not always be enough. This is why I explicitly specify the lexer with `super-zaje -l sh` in the above examples.

Note also that you should always encapsulate the input URL with quotes.

Second example: Shaul Triebitz's C quiz

Amongst other posts, Shaul has a semi-regular C quiz series where he shares code examples (that may seem trivial but aren't always that) and asks you to vote on the expected output.

I thoroughly enjoy these posts but being the cautious bloke that I am, I never vote before I test my hypothesis. If you're like me, you'll want to do the same:)

Here's an example from
https://www.linkedin.com/feed/update/urn:li:activity:7073590519842377728/?updateEntityUrn=urn%3Ali%3Afs_feedUpdate%3A%28V2%2Curn%3Ali%3Aactivity%3A7073590519842377728%29:

If you think both f() and g() compile? hit the support icon.
If you think both f() and g() don't compile? hit the insightful icon
If you think only f() compiles? hit the celebrate icon
If you think only g() compiles? hit the love icon

Feel free to elaborate on your vote in the comments.

**#c #cpp #cplusplus #programming #ShaulsCQuiz**

```c
void f(int n) {
    int arr[n];
    memset(arr, 0, sizeof(arr));
}

void g(int n) {
    int arr[n] = {0};
}
```

Below is what `super-zaje` makes of it when using `-l c`; again, you'll notice that the result isn't perfect; there's a `@` where there should be a `0` but otherwise, it's accurate. Generally speaking, the quirkier the font, the more challenging `tesseract` will find it:)

```c
void f(int n) {
    int arr[n];
    memset(arr, @, sizeof(arr))3;
}
void g(int n) {
    int arr[n] = {0};
}
```

Last examples: a post by me:)

Here's the image of the original:
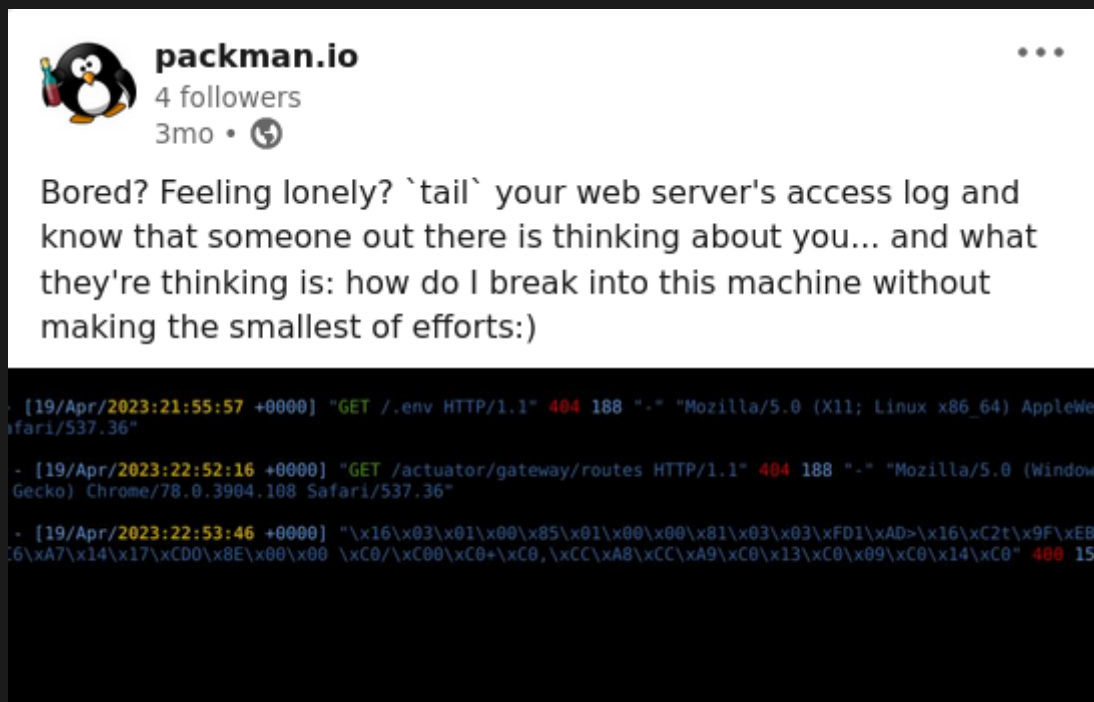
And the output for:

super-zaje -l server-log
**https://media.licdn.com/dms/image/D4E22AQEYTIhEebCD1Q/feedshare-shrink_800/0/168194
6618380?e=1694044800&v=beta&t=62Hq-rDZWOMNIt8sYAvQEUD0D8JJ4rZNYChcxGuRIOU**

- [19/Apr/2023:21:55:57 +0000] "GET /.env HTTP/1.1" 404 188 "-" "Mozilla/5.0 (X11; Linux x86_64)
AppleWet afari/537.36"

= [19/Apr/2023:22:52:16 +0000] "GET /actuator/gateway/routes HTTP/1.1" 404 188 "-" "Mozilla/5.@
(Windows Gecko) Chrome/78.0.3904.108 Safari/537.36"

+ [19/Apr/2023:22:53:46 +0800] "\x16\x03\x01\x00\x85\x01\x00\ x00\ x81\ x03\ x03\ xFD1\xAD>\
x16\ xC2t\x9F\xEBh6\XA7\x14\x17\ xCDO\ xBE\x00\X00_\xCO/\xCOO\XxCO+\xCO, \xCC\XAB\XCC
\XA9\xCO\x13\xCO\x99\xCO\x14\xCO" 400 157

The image was taken off an Nginx access log but the `server-log` lexer produces good results for
other log formats as well. I tested it with Apache and other syslogs and would welcome any inputs or
contributions of more customised lexers. See **Revising and adding new lexers** for details.

Next steps

Further down the line, I hope to:

- Train `tesseract-ocr` on inputs containing code to improve results
- Handle indentation following text extraction

I'm also working on a script that, using `FFMpeg` will detect scene changes in videos, capture images whenever these are detected and run `super-zaje` on these to extract the code. This will be especially useful when viewing all these video training courses that seem to be very popular these days.

My sincere thanks to...

- The hard working people from **tesseract-ocr** and **gosseract**
- Cecil New for creating **mdtopdf** and swiftly merging my various contributions

To install `super-zaje`, visit the **installation intructions section**

The source `markdown` for this presentation can be found in my **Crash course in...** repo.

I've used this command to convert it into a PDF:

```
md2pdf -i https://github.com/jessp01/crash-course-in/raw/main/courses/super-zaje/super-zaje.md \
    -o super-zaje.pdf \
    --theme dark \
    --new-page-on-hr \
    -s ~/.config/zaje/syntax_files
```

Contributions (as well as shares) are welcomed.

Final note

Psst.. Liked this content and have a role I could be a good fit for? I'm open to suggestions. See **https://packman.io/#contact** for ways to contact me.

Cheers, Jesse