

The awesome-json-resume* class

Chris Waltrip[†]

Released ?

0.1 Predefined Colors

There are a number of predefined colors available

- myfirstcolor,
- mysecondcolor

1 awesome-json-resume implementation

Start the DocStrip guards.

```
1 <*class>
```

Identify the internal prefix (L^AT_EX3 DocStrip convention)

```
2 <@@=ajr>
```

1.1 Initial setup

```
3 <*init>
```

Here is where we declare all of the basic information about the class, and do initial checks to make sure the class will run (e.g., the version of L^AT_EX 2_ε is new enough).

I plan on writing more here once I make sure that this is building properly.

Set up a couple of commands in recent-ish L^AT_EX 2_ε releases.

```
4 \providecommand\DeclareRelease[3]{}
5 \providecommand\DeclareCurrentRelease[2]{}
6 % ^^A \DeclareRelease{x}{1970-01-01}{awesome-json-resume-oldversion.sty}
7 \DeclareCurrentRelease{}{2023-08-08}
```

Provide current release information, and if necessary, older (breaking) releases.

```
8 % ^^A \DeclareRelease{x}{1970-01-01}{awesome-json-resume-oldversion.sty}
9 \DeclareCurrentRelease{}{2023-08-08}
```

Make sure expl3 “up-front” is loaded, because we start using L^AT_EX3 syntax early on.

```
8 \RequirePackage{expl3}
9 \RequirePackage[all]{tcolorbox}
```


Provide package information.

```
10 \ProvidesExplClass{awesome-json-resume}{2023-08-00}{0.1.0}
11 {My take on a resume class}
```

This package is based on an empty article.

```
12 \LoadClass{article}
```

* /latex-awesome-json-resume

† /cawaltrip

```

13 \pagestyle{empty}
14 \</init>

```

1.2 Global Variables

These are variables with initial values set. They can be changed by the user.

```

15 \<*options>

```

1.2.1 General

`\g__ajr_resumepath_tl` Stores the path to the résumé for parsing later.

```

16 \tl_new:N\g__ajr_resumepath_tl

```

(End of definition for \g__ajr_resumepath_tl.)

`\g__ajr_debug_bool` Used for printing debug information in Lua files.

```

17 \bool_new:N\g__ajr_debug_bool

```

(End of definition for \g__ajr_debug_bool.)

1.2.2 Color Names

```

18 \color_set:nnn{white}{RGB}{255, 255, 255}
19 \color_set:nnn{lightgray}{RGB}{158, 158, 158}
20 \color_set:nnn{gray}{RGB}{117, 117, 117}
21 \color_set:nnn{darkgray}{RGB}{21, 21, 21}
22 \color_set:nnn{black}{RGB}{0, 0, 0}
23 \color_set:nnn{red}{255, 59, 48}
24 \color_set:nnn{orange}{255, 149, 0}
25 \color_set:nnn{yellow}{255, 204, 0}
26 \color_set:nnn{green}{40, 205, 65}
27 \color_set:nnn{blue}{0, 122, 255}
28 \color_set:nnn{indigo}{88, 86, 214}
29 \color_set:nnn{purple}{175, 82, 222}

```

1.2.3 Colors Used

```

30 \color_set_eq:nn{header}{black}
31 \color_set_eq:nn{headerbg}{white}
32 \color_set_eq:nn{workheading}{red}
33 \color_seq_eq:nn{volunteerheading}{orange}
34 \color_seq_eq:nn{educationheading}{yellow}
35 \color_seq_eq:nn{awardsheading}{green}
36 \color_seq_eq:nn{certificatesheading}{blue}
37 \color_seq_eq:nn{publicationsheading}{indigo}
38 \color_seq_eq:nn{skillsheading}{purple}
39 \color_seq_eq:nn{languagesheading}{red}
40 \color_seq_eq:nn{interestsheading}{orange}
41 \color_seq_eq:nn{referencesheading}{yellow}
42 \color_seq_eq:nn{projectsheading}{green}

```

1.3 Fonts

`\g__ajr_headerfont_tl` `\g__ajr_headingfont_tl` `\g__ajr_mainfont_tl` These variables contain the names of the fonts the user wants to use for:

- header - full name at top of resume,

- heading - each individual section name (e.g., Work, Education), and
- main - the main font.

```

43 \tl_new:N\g__ajr_headerfont_tl
44 \tl_new:N\g__ajr_headingfont_tl
45 \tl_new:N\g__ajr_mainfont_tl

```

(End of definition for `\g__ajr_headerfont_tl`, `\g__ajr_headingfont_tl`, and `\g__ajr_mainfont_tl`.)

1.4 awesome-json-resume class options

All of the options specified here (except for `{(resume-path)}`), can be redefined later by calling `\AJRSetup`, one of the other defined setter commands, or in some instances, by directly updating the variable.

```

46 \DeclareKeys[ajr] {
47   resumepath .store = \g__ajr_resumepath_tl,
48   headerfont .store = \g__ajr_headerfont_tl,
49   headingfont .store = \g__ajr_headingfont_tl,
50   mainfont .store = \g__ajr_mainfont_tl,
51   headercolor .code = \color_set:nn{header}{#1},
52   headerbgcolor .code = \color_set:nn{headerbg}{#1},
53   workheadingcolor .code = \color_set:nn{workheading}{#1},
54   volunteerheadingcolor .code = \color_set:nn{volunteerheading}{#1},
55   educationheadingcolor .code = \color_set:nn{educationheading}{#1},
56   awardsheadingcolor .code = \color_set:nn{awardsheading}{#1},
57   certificatesheadingcolor .code = \color_set:nn{certificatesheading}{#1},
58   publicationsheadingcolor .code = \color_set:nn{publicationsheading}{#1},
59   skillsheadingcolor .code = \color_set:nn{skillsheading}{#1},
60   languagesheadingcolor .code = \color_set:nn{languagesheading}{#1},
61   interestsheadingcolor .code = \color_set:nn{interestsheading}{#1},
62   referencesheadingcolor .code = \color_set:nn{referencesheading}{#1},
63   projectsheadingcolor .code = \color_set:nn{projectsheading}{#1},
64   debug .bool_set:N\g__ajr_debug_bool,
65
66   resumepath .usage = preamble,
67   headerfont .usage = general,
68   headingfont .usage = general,
69   mainfont .usage = general,
70   headercolor .usage = general,
71   headerbgcolor .usage = general,
72   workheadingcolor .usage = general,
73   volunteerheadingcolor .usage = general,
74   educationheadingcolor .usage = general,
75   awardsheadingcolor .usage = general,
76   certificatesheadingcolor .usage = general,
77   publicationsheadingcolor .usage = general,
78   skillsheadingcolor .usage = general,
79   languagesheadingcolor .usage = general,
80   interestsheadingcolor .usage = general,
81   referencesheadingcolor .usage = general,
82   projectsheadingcolor .usage = general,
83   debug .usage = general
84 }

```

```

85
86 \SetKeys[ajr]{
87   resumepath = {resume.json},
88   debug = false
89 }
90
91 \ProcessKeyOptions[ajr]

```

I should probably also pass keys to other packages (or do I do that automatically?).

```

92 </options>

```

1.5 Required Packages

```

93 <*interfaces>

```

This list is gonna get large in a hurry I think.

```

94 \RequirePackage{fontspec}

```

1.6 Document-level Commands

The document-level commands can be separated into setup commands and generation commands.

1.6.1 Setup Command

These commands are basic setters for the different options that are able to be set from `\documentclass`.

\AJRSetup This is the most general setup command. It will handle whatever could be set from the `\documentclass` command.

```

95 \NewDocumentCommand{\AJRSetup}{\keys_set:nn{ajr}{#1}}

```

(End of definition for \AJRSetup. This function is documented on page ??.)

1.6.2 Résumé-generation Commands

These functions all make calls to their underlying Lua functions. `\GenerateResume` relies on the `order` array in the `metadata` key of the résumé JSON file. The rest of the functions call their underlying functions directly. In this way, if an order wasn't setup, a user could still call each of the section they want to print.

\CreateResume Creates the entire résumé.

```

96 \NewDocumentCommand{\CreateResume}{*}{\lua_now:e{ajr.create_resume("\luaescapestring{g__ajr_

```

(End of definition for \CreateResume. This function is documented on page ??.)

\CreateSection Creates an individual section of the résumé. This is the function that all the following functions call.

#1 : section, the section name to print, based on the key names defined in the “JSON Resume” schema.

```

97 \NewDocumentCommand{\CreateSection}{m}{
98   \lua_now:e{ajr.create_section(
99     "\luaescapestring{#1}",
100     "\luaescapestring{g__ajr_resumepath_tl}"
101   )}
102 }

```

(End of definition for `\CreateSection`. This function is documented on page ??.)

`\InitializeResume` Places just the header of the résumé on the page. Don't know if this command will actually work.

103 `\NewDocumentCommand{\InitializeResume}{\CreateSection{basics}{\g__ajr_resumepath_tl}}`

(End of definition for `\InitializeResume`. This function is documented on page ??.)

1.7 Internal commands

These functions call out to Lua when necessary to parse the résumé JSON file and provide

104 `</interfaces>`

105 `</class>`

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

A	
ajr internal commands:	
<code>\g__ajr_debug_bool</code>	<u>17</u> , 64
<code>\g__ajr_headerfont_tl</code>	<u>43</u> , 48
<code>\g__ajr_headingfont_tl</code>	<u>43</u> , 49
<code>\g__ajr_mainfont_tl</code>	<u>43</u> , 50
<code>\g__ajr_resumepath_tl</code>	<u>16</u> , 47, 96, <u>100</u> , 103
<code>\AJRSetup</code>	3, <u>95</u>
B	
bool commands:	
<code>\bool_new:N</code>	17
C	
color commands:	
<code>\color_seq_eq:nn</code>	33, 34, 35, 36, 37, 38, 39, 40, 41, 42
<code>\color_set:nn</code>	51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63
<code>\color_set:nnn</code>	18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29
<code>\color_set_eq:nn</code>	30, 31, 32
<code>\CreateResume</code>	<u>96</u>
<code>\CreateSection</code>	<u>97</u> , 103
D	
<code>\DeclareCurrentRelease</code>	5, 7
<code>\DeclareKeys</code>	46
<code>\DeclareRelease</code>	4, 6
<code>\documentclass</code>	4
G	
<code>\GenerateResume</code>	4
I	
<code>\InitializeResume</code>	<u>103</u>
K	
keys commands:	
<code>\keys_set:nn</code>	95
L	
<code>\LoadClass</code>	12
lua commands:	
<code>\lua_now:n</code>	96, 98
<code>\luaescapestring</code>	96, 99, 100
N	
<code>\NewDocumentCommand</code>	95, 96, 97, 103
P	
<code>\pagestyle</code>	13
<code>\ProcessKeyOptions</code>	91
<code>\providecommand</code>	4, 5
<code>\ProvidesExplClass</code>	10
R	
<code>\RequirePackage</code>	8, 9, 94
S	
<code>\SetKeys</code>	86
T	
tl commands:	
<code>\tl_new:N</code>	16, 43, 44, 45