

NvChad Installation

```
1 const Generator = require("yeoman-generator");
2 const chalk = require("chalk");
3 const yosay = require("yosay");
4
5 module.exports = class extends Generator {
6   prompting() {
7     this.log(chalk.red("This is just an example"));
8
9     const prompts = [
10      {
11        message: "Enter the daily exercise title",
12      },
13    ];
14    return this.prompt(prompts).then((props) => {
15      this.props = props;
16    });
17  };
18 };
19
20 export const loadProduct = (product) => {
21   let xhr = new XMLHttpRequest();
22   xhr.open("GET", "testFiles/products.json", true);
23   xhr.onload = function () {
24     if (this.status === 200) {
25       const product = JSON.parse(this.responseText);
26       document.querySelector("#product").innerHTML = output;
27     }
28   };
29 };
30
31 function
```

```
1 // Deletes whole todos in a project
2
3 import { DeleteIcon } from "solid-icons/ri";
4 import { delTodos } from "./todoFuncs";
5
6 export const RemoveAllTodos = () => {
7   return (
8     <button onClick={() => delTodos()}>
9       <DeleteIcon size={21} /> Todos
10    </button>
11  );
12 };
13
14 };
15
16 import { createData, getUrl } from "./utils";
17
18 const getData = async (area) => {
19   const url = getUrl(area);
20   let response = await fetch(url);
21
22   if (response.status !== 200) {
23     throw new Error("place not found");
24   }
25
26   let result = await response.json();
27   createData(result);
28 };
29
30 @example/docs@0.0.1 start
31 > astro dev
```

```
astro v1.0.0-beta.53 started in 27ms
Local http://localhost:3000/
Network use --host to expose
This is a beta prerelease build
Feedback? https://astro.build/issues
```

```
1 import ansiStyles from "ansi-styles";
2 import supportsColor from "supports-color";
3 import {
4   stringRepeat,
5   stringRepeatWithFirstIndex,
6 } from "./utilities.js";
7
8 const { stdout: stdoutColor, stderr: stderrColor } = supportsColor;
9
10 const GENERATOR = Symbol("GENERATOR");
11 const STYLER = Symbol("STYLER");
12 const IS_DUPY = Symbol("IS_DUPY");
13
14 // repeatFunction(level, ansiStyles.colorName) mapping
15 const styles = Object.create(null);
16
17 const applyOptions = (object, options = {}) => {
18   if (options.level && !Number.isInteger(options.level) && options.level >= 0 && options.level <= 3) {
19     throw new Error("The 'level' option should be an integer from 0 to 3");
20   }
21
22   // Detect level of support available
23   const colorLevel = stdoutColor ? stdoutColor.level : 0;
24   object.level = options.level === undefined ? colorLevel : options.level;
25 };
26
27 const chalkFactory = options => {
28   const chalk = (...strings) => strings.join(' ');
29   applyOptions(chalk, options);
30   Object.setPrototypeOf(chalk, createChalk.prototype);
31   return chalk;
32 };
33
34 function createChalk(options) {
35   return chalkFactory(options);
36 }
37
38 Object.setPrototypeOf(createChalk, chalkFactory);
39
40 for (const [styleName, style] of Object.entries(styles)) {
41   const builder = createBuilder(this, this[STYLER], true);
42   Object.defineProperty(this, styleName, { value: builder });
43   return builder;
44 };
45
46 styles.visible = {
47   set() {
48     const builder = createBuilder(this, this[STYLER], true);
49     Object.defineProperty(this, 'visible', { value: builder });
50     return builder;
51   },
52 };
53
54 const getModeansi = (model, level, type, ...arguments) => {
55   if (model === 'rgb') {
56     if (level === 'ansi16') {
57       return ansiStyles[type].ansi16(...arguments);
58     }
59   }
60
61   // Route prototype all = function all() {
62   var handles = flatten(slice.call(arguments));
63
64   for (var i = 0; i < handles.length; i++) {
65     var handle = handles[i];
66
67     if (typeof handle === 'function') {
68       var type = toString.call(handle);
69       var msg = "Route.all() requires a callback function but got a " + type
70         throw new TypeError(msg);
71     }
72
73     var layer = Layer('/', {}, handle);
74     layer.method = undefined;
75     this.methods.all = true;
76     this.stack.push(layer);
77   }
78
79   methods.forEach(function(method) {
80     Route.prototype[method] = function() {
81       var handles = flatten(slice.call(arguments));
82
83       for (var i = 0; i < handles.length; i++) {
84         var handle = handles[i];
85
86         if (typeof handle === 'function') {
87           var type = toString.call(handle);
88           var msg = "Route.all() requires a callback function but got a " + type
89             throw new TypeError(msg);
90         }
91
92         debug("ks %s", method, this.path);
93
94         var layer = Layer('/', {}, handle);
95         layer.method = method;
96         this.methods[method] = true;
97       }
98     };
99   });
100 }
```

<https://nvchad.com/docs/quickstart/install>

```
# essentials für zb C-Compiler
apt install build-essential

# Nerdfont installieren
wget -P ~/.local/share/fonts https://github.com/ryanoasis/nerd-
fonts/releases/download/v3.0.2/JetBrainsMono.zip \
&& cd ~/.local/share/fonts \
&& unzip JetBrainsMono.zip \
&& rm JetBrainsMono.zip \
&& fc-cache -fv

# Nvim installieren
snap install nvim --classic

# NvChad installieren
git clone https://github.com/NvChad/starter ~/.config/nvim && nvim

# in nvim:
:MasonInstallAll
```

Falls es Probleme auf einem Hetzner Server mit dem Nutzerverzeichnis unter root gibt:

(in nvim mit `:echo stdpath("config")` prüfen was nvim als home-Verzeichnis nimmt)

```
mkdir -p /home/root/.config
mv /root/.config/nvim /home/root/.config/
```

Revision #7

Created 2025-07-29 16:29:00 UTC

Updated 2025-07-29 16:46:00 UTC