

Metodika Code Conventions

Vypracoval: Daniel Hroš

Účel metodiky

Účelom tejto metodiky je zdokumentovať súbor usmernení pre špecifický programovací jazyk použitý v našom projekte pre unifikovanie písania kódu jednotlivými vývojarmi tímu. Konvencie sú pre členov tímu vysoko odporúčané pre zlepšenie čitateľnosti a údržby ich vlastného zdrojového kódu.

Všeobecné konvencie

- Zložité časti kódu sú okomentované s vysvetlením
- V každom repozitári je použitý linter

CHANGELOG.md

Všetky repozitáre majú vo svojom koreňovom adresári súbor CHANGELOG.md, ktorý zaznamenáva zmeny. Najnovšia zmena je na vrchu dokumentu. Changelog sa aktualizuje po každom merge do develop alebo master a má predpísanú štruktúru:

```
1 # CHANGELOG
2
3 - Order: add new entries to the top
4 - Update with all notable changes, features and bugfixes
5 - Keep formatting consistent - date followed by type of changes, e.g. `Added`, `Changed`,
`Fixed`, `Removed`
6
7 ## 2021 October 9th
8
9 Changes based on [this](https://team-4-2021.atlassian.net/wiki/spaces/ECEH/pages/edit-
v2/2916487) task
10
11 ### Added
12
13 - Added xy
14
15 ### Changed
16
17 - Changed xy
18
19 ### Fixed
20
21 - Fixed xy
22
23 ### Removed
```



TypeScript & React

V projektoch sa používajú nástroje Prettier a ESLint s pluginom pre React a konfiguráciou airbnb v predvolenom nastavení.

.eslintrc

```
1{
2  "root": true,
3  "parser": "@typescript-eslint/parser",
4  "parserOptions": {
5    "ecmaFeatures": {
6      "jsx": true
7    },
8    "project": "tsconfig.json"
9  },
10 "settings": {
11   "react": {
12     "version": "detect"
13   }
14 },
15 "plugins": [
16   "@typescript-eslint",
17   "prettier"
18 ],
19 "extends": [
20   "eslint:recommended",
21   "react-app/jest",
22   "airbnb-typescript",
23   "plugin:@typescript-eslint/eslint-recommended",
24   "plugin:@typescript-eslint/recommended",
25   "plugin:react/recommended",
26   "plugin:prettier/recommended",
27   "prettier/@typescript-eslint"
28 ],
29 "rules": {
30   "prettier/prettier": "error",
31   "react/jsx-one-expression-per-line": "off",
32   "react/jsx-props-no-spreading": [
33     "error",
34     {
35       "custom": "ignore"
36     }
37   ],
38   "import/extensions": "off",
39   "import/prefer-default-export": "warn",
40   "no-use-before-define": "off",
41   "import/no-named-as-default": "off",
42   "@typescript-eslint/no-use-before-define": [
43     "error"
```

```
44   ],
45   "react/prop-types": "off",
46   "react/require-default-props": "off"
47 }
48 }
```

tsconfig.json

```
1{
2  "compilerOptions": {
3    "target": "es5",
4    "lib": ["dom", "dom.iterable", "esnext"],
5    "allowJs": false,
6    "skipLibCheck": true,
7    "esModuleInterop": true,
8    "allowSyntheticDefaultImports": true,
9    "strict": true,
10   "forceConsistentCasingInFileNames": true,
11   "noFallthroughCasesInSwitch": true,
12   "module": "esnext",
13   "moduleResolution": "node",
14   "resolveJsonModule": true,
15   "isolatedModules": true,
16   "noEmit": true,
17   "jsx": "react-jsx"
18 },
19 "include": ["src"]
20 }
```

.prettierrc

```
1{
2  "printWidth": 100,
3  "tabWidth": 2,
4  "singleQuote": true,
5  "arrowParens": "always"
6}
```



Užitečné linky:

- [Airbnb React/JSX Style Guide](#)



PHP & Laravel

Pri písaní kódu v Laravel aplikácií je kľúčové dodržiavať konvencie názvov. V tejto časti popíšem aké konvencie budeme dodržiavať v našom projekte.

1. Controller

- názvy by mali byť v jednotnom čísle
- mal by sa používať CamelCase

Správne	Nesprávne
CustomerController.php	CustomersController.php

2. Pomenovanie URI

- názvy uri by mali byť v množnom čísle
- mal by sa používať kebab-case v prípade viacslovných názvov

Správne	Nesprávne
/customers/25	/customer/25
/customers/password-reset	/customers/password_reset
/customers/password-reset	/customers/passwordReset

3. Pomenovanie routes

- v prípade viacslovných názvov by sa mal používať snake_case
- najlepšie používať rovnaký názov ako URI

Správne	Nesprávne
->('customers.view')	->('customers-view')
->('customers.view')	->('customers_view')
->('customers.password_reset')	->('customers.password.reset')
->('customers.password_reset')	->('customers.password-reset')
->('customers.password_reset')	->('customer-password-reset')

4. Pomenovanie migrácií

- rovnaký názov ako má model ale píšeme pomocou snake_case.

Správne	Nesprávne
2021_03_19_033513_create_customers_table.php	2021_03_19_033513_customers.php
2021_03_19_033513_add_image_id_to_customers_table.php	2021_03_19_033513_add_image_id_customers.php
2021_03_19_033513_drop_image_id_from_customers_table.php	2021_03_19_033513_remove_image_id_customers.php

5. Pomenovanie databázových tabuliek

- množné číslo
- používame snake_case

Správne	Nesprávne
customers	customer
cart_items	cartItems , CartItems , Cart_item

6. Pomenovanie pivotných tabuliek

- jednotné číslo
- snake_case
- poradie tabuliek by malo byť podľa abecedy

Správne	Nesprávne
course_student	student_courses , students_courses ,course_students

7. Pomenovanie stĺpcov v tabuľkách

- snake_case
- Názov tabuľky by nemal byť zakomponovaný aj v názve stĺpca

Správne	Nesprávne
first_name	user_first_name , FirstName

8. Pomenovanie cudzích kľúčov (FK)

- snake_case
- používame jednotné číslo tabuľky s prefixom id

Správne	Nesprávne
course_id	courseId , id ,courses_id ,id_course

9. Pomenovanie primárnych kľúčov (PK)

- používame len ID

Správne	Nesprávne
---------	-----------

id	custom_id
----	-----------

10. Pomenovanie modelov

- názov modelu by mal byť v jednotnom čísle
- musí sa používať CamelCase
- model musí byť jednotné číslo z názvu tabuľky

Správne	Nesprávne
Customer	Customers, customer

11. Pomenovanie funkcií

- mal by sa používať snake_case

12. Pomenovanie metód v Controllery

- mal by sa používať CamelCase
- musí používať jedno slovo príbuzné k akcií

Správne	Nesprávne
store	saveCustomer
show	viewCustomer
destroy	deleteCustomer
index	allCustomersPage

13. Pomenovanie premenných

- mal by sa používať CamelCase
- používame jasné slová, ktoré jasne opisujú čo pomenúvajú

Správne	Nesprávne
\$customerMessages	\$CustomerMessages, \$customer_messages, \$c_messages, \$c_m

14. Pomenovanie collections

- množné číslo
- používame jasné slová, ktoré jasne opisujú čo pomenúvajú

15. Pomenovanie objektov

- jednotné číslo
- používame jasné slová, ktoré jasne opisujú čo pomenúvajú

16. Pomenovanie interfaces

- malo by to byť prídavné meno alebo podstatné meno

Správne	Nesprávne
Authenticable	AuthenticationInterface ,Authenticate



Užitočné linky:

- [Laravel Naming Conventions - Web dev etc - my software development blog](#)
- [DEV Laravel Best Practice \[Coding Standards Part 01\]](#) 🤖🤖🤖



HTML & CSS

1. Deklarácia typu dokumentu v HTML

Je dôležité určiť typ dokumentu, ktorý by mal prehliadač zobrazovať. Vďaka tomu môže prehliadač vedieť, aký typ dokumentu by mal vykresliť, a tak využiť svoje zdroje.

```
1 <!DOCTYPE Html>
```

2. Použitie malých názvov tagov v HTML

```
1 <body>
2 <p>Welcome to DataFlair</p>
3 </body>
```

3. Zatvorenie všetkých tagov v HTML

HTML nepožaduje zatvorenie všetkých prvkov, ale je lepšie to urobiť.

```
1 <body>
2 <p>Welcome to DataFlair.
3 </body>
```

4. Používajte malé písmo pre názvy atribútov

```
1 <a href="https://data-flair.training/">Welcome to DataFlair</a>
```

5. Používajte úvodzovky pre hodnoty atribútov

```
1 <table class="bordered">
```

6. Špecifikácia alt, width a height atribútov obrázku

```
1 
```

7. Nevynechávajte tagy <html> a <body> a <head>

Webová stránka sa úspešne vykreslí, aj keď vynecháme prvky <html> a <body>. Je však lepšie ich používať, pretože vynechanie tagu <body> spôsobuje chyby v starých prehliadačoch a vynechanie tagu <head> môže spôsobiť zlyhanie softvéru XML a DOM.

```
1 <head>
2 <title>Page Title</title>
3 </head>
4 <h1>Hi</h1>
5 <p>Welcome</p>
```

8. Prázdne riadky a odsadenie

Neprikladajte prázdne riadky, medzery ani odsadenia bezdôvodne. Kvôli čitateľnosti pridajte prázdne riadky na oddelenie veľkých alebo logických blokov kódu. Kvôli čitateľnosti pridajte dve medzery odsadenia a nie tab.

9. Názvy CSS tried

Namiesto kryptických mien vždy používajte názvy tried, ktoré odrážajú účel príslušného prvku alebo ktoré sú inak všeobecné.

```
1 /* Not recommended: meaningless */
2 #yee-1901 {}
3
4 /* Recommended: presentational */
5 .button-green {}
6 .clear {}
```

10. Skrátenejší zápis

Ak je to možné, používajte skrátenejší zápis. CSS ponúka množstvo skrátenejších vlastností (napríklad písmo), ktoré by sa mali používať vždy, keď je to možné.

Používanie skrátenejších vlastností je užitočné pre efektivitu kódu a zrozumiteľnosť.

```
1 /* Not recommended */
2 border-top-style: none;
3 font-family: palatino, georgia, serif;
4 font-size: 100%;
5 line-height: 1.6;
6 padding-bottom: 2em;
7 padding-left: 1em;
```



```
8 padding-right: 1em;
9 padding-top: 0;
10
11 /* Recommended */
12 border-top: 0;
13 font: 100%/1.6 palatino, georgia, serif;
14 padding: 0 1em 2em;
```

11. Oddelovače v názve triedy

Slová v názvoch ID a triedach oddel'te spojovníkom.

Nespajajte slová a skratky vo výberoch s inými znakmi (vrátane žiadnymi) s výnimkou spojovníkov, aby sa zlepšilo porozumenie a prehľadnosť.

```
1 /* Not recommended: */
2 .demoimage {}
3 .error_status {}
4
5 /* Recommended */
6 #video-id {}
7 .ads-sample {}
```

12. Poradie deklarácii atribútov

Usporiadajte deklarácie atribútov v abecednom poradí, aby ste dosiahli konzistentný kód spôsobom, ktorý je ľahko zapamätateľný a udržateľný.

13. Odsadzovanie

Odsad'te všetok obsah bloku, tj pravidlá v rámci pravidiel, ako aj deklarácie, aby odrážali hierarchiu a zlepšovali porozumenie.

```
1 @media screen, projection {
2
3   html {
4     background: #fff;
5     color: #444;
6   }
7
8 }
```

14. Úvodzovky v CSS

Na selektory atribútov a hodnoty vlastností používajte jednoduché (") a nie dvojité (") úvodzovky. V hodnotách URI (url ()) nepoužívajte úvodzovky.

```
1 /* Recommended */
2 @import url(https://www.google.com/css/maia.css);
3
4 html {
5   font-family: 'open sans', arial, sans-serif;
6 }
```



Užitočné linky:

- [HTML Style Guide and Coding Conventions](#)