

# Extrakcia dát z webu

[WebExtraction]

*Metodika pre vývoj*

<b>Tím:</b>	č. 16, WebX
<b>Vedúci tímu:</b>	Ing. Ivan Srba, PhD.
<b>Členovia tímu:</b>	Ján Brechtel, Tomáš Juhaniak, Martin Kalužník, Rastislav Krchňavý, Michal Kren, Martin Lacek, Andrej Vaculčíak
<b>Akademický rok:</b>	2016/2017
<b>Autor:</b>	Martin Kalužník
<b>Verzia číslo:</b>	1.0
<b>Dátum poslednej zmeny:</b>	25.10.2016

## 1 Úvod

Cieľom tohto dokumentu je zdefinovanie základných postupov pre jednotnú tvorbu a komentovanie zdrojových kódov. Metodika zahŕňa konvencie písania zdrojového kódu, ktoré sú jeho tvorcovia povinní dodržiavať. Uvedená metodika je zameraná na programovacie jazyky Ruby a JavaScript, ktoré budú zdieľať spoločné základné konvencie pokiaľ to ich syntax dovoľí.

## **2 Skratky a značky**

## 3 Postupy

### 3.1 Spoločné pravidlá

#### 3.1.1 Základné pravidlá

- Komentáre v kóde budú písané výhradne v angličtine.
- Každý riadok musí končiť znakom nového riadku.

#### 3.1.2 Vynucovanie konvencií

Súčasťou prehliadky zdrojového kódu je aj poukázanie na akékoľvek nezhody konvencií zdrojového kódu. Kód nemôže prejsť prehliadkou, pokiaľ neboli všetky objavené nezhody v konvenciách opravené, aj v prípade, že je inak plne funkčný.

#### 3.1.3 Opravovanie konvencií v existujúcom kóde

V prípade, že robíte zmenu na riadku, v ktorom je použitá nesprávna konvencia, je povolené túto konvenciu opraviť.

V prípade, že si všimnete nedodržanú konvenciu na riadku, ktorý neupravujete, je zakázané úpravu tohto riadku spojiť do commitu s inými nesúvisiacimi zmenami. Na úpravu konvencií vytvorte samostatný commit, podľa možnosti použite “git commit --fixup <commit ktorý pridal nekonzistentnú konvenciu>”.

## 3.2 Ruby

### 3.2.1 Základné pravidlá

Ak pravidlo nie je špecifikované v tejto metodike, riadime sa pravidlami vytvorenými Ruby komunitou:

<https://github.com/bbatsov/ruby-style-guide/blob/master/README.md#source-code-layout>

- Úvodzovky pri stringoch - používať jednoduché úvodzovky v prípadoch, keď nie je potrebná interpolácia alebo špeciálne znaky ako `\t` `\n` alebo `‘`.

```
name = 'Bozhidar'
```

Nie:

```
name = "Bozhidar"
```

- Pridávanie závislostí (gemov) - preferujeme špecifikáciu verzie pomocou optimistického operátora, prípadne bez špecifikácie verzie:

```
gem 'devise', '>= 3.4.1'
```

## Metodika pre vývoj

V prípade známej nekompatibility nových verzií je možné použiť optimistický operátor:

```
gem 'devise', '~> 3.4.1'
```

Alebo špecifikovanie konkrétnej verzie:

```
gem 'devise', '3.4.1'
```

### 3.2.2 Výber najčastejšie porušovaných konvencií

Disclaimer: Tento výber slúži na pripomenutie si najčastejšie porušovaných konvencií počas prehliadky kódu, konvencie v ňom môžu často pribúdať a ubúdať podľa toho, ako rýchlo sa ich naučíme. Ideálne bude na konci prázdny.

### 3.2.3 Úvodzovky pri stringoch

Používať jednoduché úvodzovky v prípadoch, keď nie je potrebná interpolácia alebo špeciálne znaky ako `\t` `\n` alebo ```:

```
name = 'Bozhidar'
```

Nie:

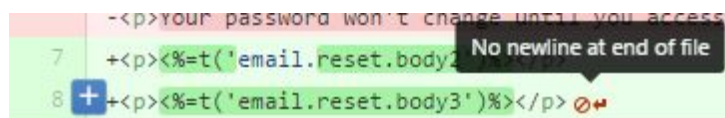
```
name = "Bozhidar"
```

RubyMine dokáže na toto upozorňovať.



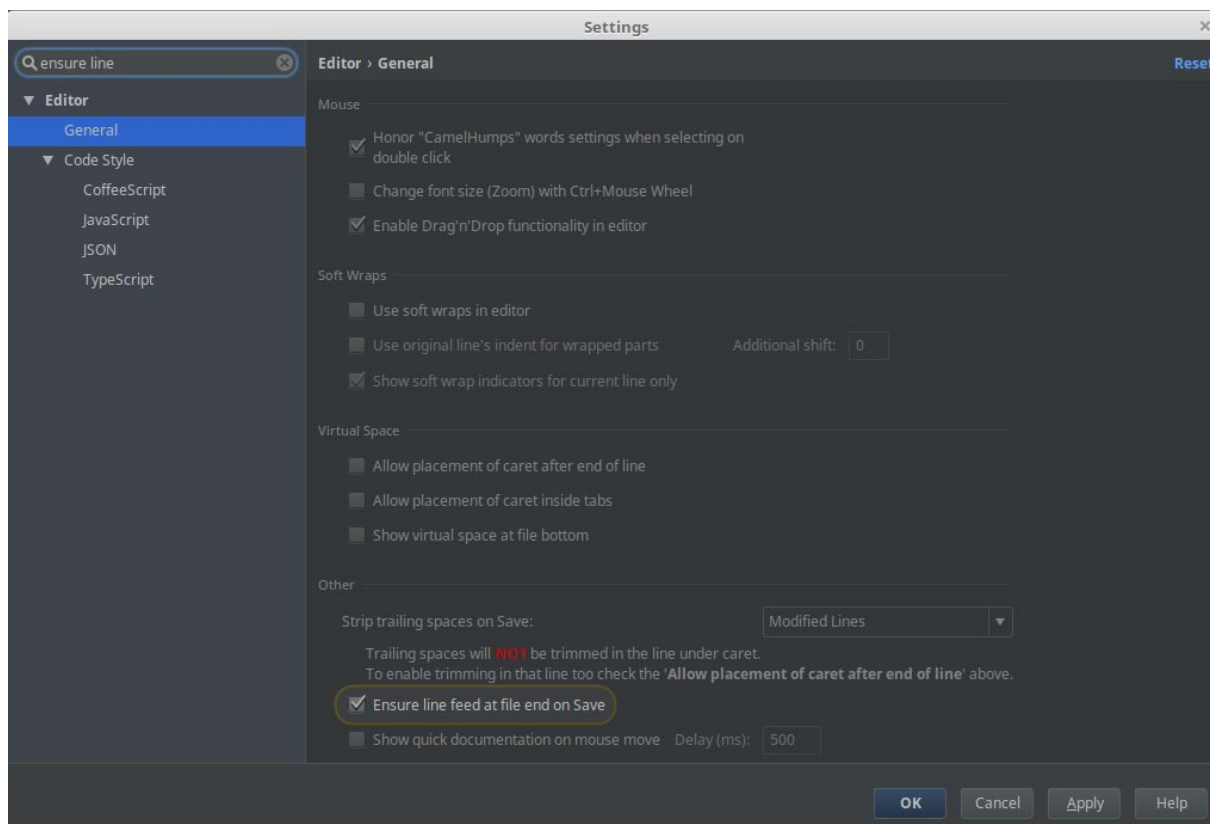
### 3.2.4 Nový riadok na konci súboru

Pri diff pohľade pri pull requeste na githube je jednoduché si všimnúť chýbajúce konce riadkov.



Pre používateľov RubyMine stačí nastaviť editor podľa obrázku a nikdy sa s tým nebudú musieť zaoberať.

# Metodika pre vývoj



## 3.2.5 Prázdný riadok

- Pred aj za blokom kódu (do - end, each...) je prázdny riadok.

## 3.2.6 Jednoduché podmienky

Jednoduché podmienky, t.j. vykonáva sa iba jeden riadok a neexistuje else vetva, podmienka je napísaná na jednom riadku:

```
do_something if true
```

Nie:

```
if true
  do_something
end
```

## 3.3 JavaScript

### 3.3.1 Základné pravidlá

Na napísanie keď začneme používať JS

### 3.3.2 Výber najčastejšie porušovaných konvencií