Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Iľkovičova 2, 842 16, Bratislava 4

Tímový projekt

MOB-UX

Metodika konvencií zdrojového kódu a databáz

Vedúci projektu: Ing. Eduard Kuric, PhD.
Názov tímu: MOB-UX
Členovia tímu: Bc. Tomáš Anda
Bc. Dávid Beňo
Bc. Matúš Buzássy
Bc. Martin Nagy
Bc. Patrik Pindeš
Bc. Igor Vereš
Vypracoval: Bc. Igor Vereš - Databázy
Bc. Patrik Pindeš - Knockout
Bc. Dávid Beňo - CakePHP
Bc. Tomáš Anda - Webové stránky

Kontakt: team11fiittp@gmail.com
Akademický rok: 2017/2018, zimný semester

Úvod

1. Účel dokumentu

Dokument slúži ako metodika, na opis pravidiel menovania databáz tímu č. 11 pod menom MobUX v predmete Tímový projekt.

Dokument slúži aj ako príručka pre formátovanie a upravovanie webových stránok a ich prvkov v rámci klientskej zóny. Pri vytváraní webových stránok je kvôli prehľadnosti a konzistentnosti potrebné, aby sa dodržiavali nasledujúce kritériá.

Okrem toho tento dokument opisuje style guide a best practices písania kódu v jazykoch PHP, JavaScript a vo frameworkoch CakePHP, Knockout. Dokument bude doplnený o nové poznatky v prípade zistenia nekonzistentnosti kódu.

2. Pravidlá menovania v databáze Postgres

- Každý stĺpec ktorý obsahuje kľúč musí byť nazvaný id_<meno tabuľky ktorej je to primárny klúč>.
- Pre primárne klúče v postgreSQL používajte dátový typ “bigserial” ktorý sa automaticky inkrementuje, v každom prípade kde to je vhodné.
- Pre primárne klúče v postgreSQL vždy používajte direktívu NOT NULL.
- V prípadoch kedy hodnota stĺpca nemôže byť prázdna použite direktívu NOT NULL.
- V prípadoch kedy stĺpec má defultnú hodnotu, použite direktívu DEFAULT.
- Všetky názvy by mali byť čo najkratšie, ale zároveň výstižné.
- Ak názov obsahuje viacero slov oddelte ich podtržníkom (napr. priklad_mena).

3. PHP

Zatiaľ bolo definované použitie PHP len vrámci frameworku CakePHP 3.

4. CakePHP 3

4.1 Názvy tried a súborov

Vo všeobecnosti sa názvy súborov zhodujú s názvami tried a postupujú podľa normy PSR-4 pre autoloading. Nižšie sú uvedené niektoré príklady názvov tried a ich názvov súborov:

- Trieda Controller *LastArticlesController* bude v súbore s názvom **LatestArticlesController.php**

- Trieda Component *MyHandyComponent* bude v súbore s názvom **MyHandyComponent.php**
- Trieda View *SuperSimpleView* sa nájde v súbore s názvom **SuperSimpleView.php**

Znamená to, že súbor je možné pomenovať ľubovoľne ale musí mať príponu, ktorá jasne špecifikuje o aký typ súboru sa jedná.

4.2 Štruktúra priečinkov

/bin - obsahuje spustiteľné súbory konzoly Cake.

/config - obsahuje niekoľko konfiguračných súborov, ktoré CakePHP používa. Podrobnosti o pripojení k databáze, bootstrapping, základné konfiguračné súbory a podobne by mali byť uložené tu.

/plugins - obsahuje pluginy využívané v aplikácii.

/logs - obsahuje log súbory

/src - obsahuje zdrojové kódy aplikácie. Podrobnejšia štruktúra tohto priečinku je popísaná ďalej

/tests - obsahuje testovacie skripty aplikácie

/tmp - obsahuje dočasné súbory. Skutočné údaje, ktoré ukladá, závisia od toho, ako ste nakonfigurovali CakePHP, ale táto zložka sa zvyčajne používa na ukladanie prekladateľských správ, popisov modelov a niekedy aj informácií o reláciách.

/vendor - obsahuje závislosti nainštalované priamo od CakePHP alebo pomocou Composera. Neodporúča sa meniť tieto súbory - pri update bývajú prepísané.

/webroot - verejný priečinok. Obsahuje súbory, ktoré môžu byť verejne dostupné.

4.3 Štruktúra priečinku /src

Controller - obsahuje controller-y aplikácie a ich komponenty

Locale - obsahuje súbory s textami pre rôzne jazyky

Model - Obsahuje tabuľky, entity a správanie aplikácie.

Shell - Obsahuje príkazy konzoly a konzolové úlohy pre aplikáciu

View - Tu sú umiestnené prezentačné triedy: zobrazenia, bunky, pomocníci.

Template - Tu sú umiestnené prezentačné súbory: elementy, chybové stránky, rozloženia (layouts) a súbory šablón (templates) pre zobrazenie.

5. Knockout

1. Udržuj prepoužiteľné bindings vo zvlášť javascript súbore CustomBindings.js

```
// Reusable bindings - ideally kept in a separate file

ko.bindingHandlers.fadeVisible = {
    init: function(element, valueAccessor) {
        // Start visible/invisible according to initial value
        var shouldDisplay = valueAccessor();
        $(element).toggle(shouldDisplay);
    },
    update: function(element, valueAccessor) {
        // On update, fade in/out
        var shouldDisplay = valueAccessor();
        shouldDisplay ? $(element).fadeIn() : $(element).fadeOut();
    }
};

ko.bindingHandlers.jqButton = {
    init: function(element) {
       $(element).button(); // Turns the element into a jQuery UI button
    },
    update: function(element, valueAccessor) {
        var currentValue = valueAccessor();
        // Here we just update the "disabled" state, but you could update other properties too
        $(element).button("option", "disabled", currentValue.enable === false);
    }
};
```

Obrázok č. 1: Bindings

2. ViewModel s pomocnými triedami udržuj zvlášť v javascript súbore
3. ViewModel definuj ako funkciu

4. Pomocné triedy definuj ako funkciu, ukážka na obrázku č.2

```
function Answer(text) {
    var self = this
    self.answerText = text;

    self.points = ko.observable(1);
}

function SurveyViewModel(question, pointsBudget, answers) {
    var self = this;
    self.question = question;
    self.pointsBudget = pointsBudget;
    self.answers = $.map(answers, function(text) { return new Answer(text) });
    self.save = function() { alert('To do') };

    self.pointsUsed = ko.computed(function() {
        var total = 0;
        for (var i = 0; i < self.answers.length; i++)
            total += self.answers[i].points();
        return total;
    });
}

ko.applyBindings(new SurveyViewModel("Which factors affect your technology choices?", 10, [
   "Functionality, compatibility, pricing - all that boring stuff",
   "How often it is mentioned on Hacker News",
   "Number of gradients/dropshadows on project homepage",
   "Totally believable testimonials on project homepage"
]));
```

Obrázok č. 2: Ukážka vievModel a pomocnej triedy

5. Zadeklaruj si "var self = this" na začiatku viewModel a odkazuj sa na definície zvyšného obsahu viewModel pomocou "self".
6. Všetky properties, observables a functions k viewModel definuj vo vnútri viewModel
7. Ukážka štruktúry spolu s naming konvenciou vidno na obrázku č.3
    a. properties, methods, observables - všetko cammelCase
    b. viewModel,
    c. pomocne triedy - PascalCase, MojNazovViewModel
    d. viewModel sa nachádza v js súbore moj_nazov_view_model.js

```
//content of file WebMail.js
//other support classes goes here
...

function WebmailViewModel() {
    // Data
    var self = this;
    self.folders = ['Inbox', 'Archive', 'Sent', 'Spam'];

    //Observables
    self.chosenFolderId = ko.observable();
    self.chosenFolderData = ko.observable();
    self.chosenMailData = ko.observable();
    self.firstName = ko.observable('Bob');
    self.lastName = ko.observable('Smith');

    //Computed
    self.fullName = ko.computed(function() {
        return self.firstName() + " " + self.lastName();
    });

    // Behaviours
    self.goToFolder = function(folder) { location.hash = folder };
    self.goToMail = function(mail) { location.hash = mail.folder + '/' + mail.id };

    // Client-side routes
    Sammy(function() {
        this.get('#:folder', function() {
            self.chosenFolderId(this.params.folder);
            self.chosenMailData(null);
            $.get("/mail", { folder: this.params.folder }, self.chosenFolderData);
        });

        this.get('#:folder/:mailId', function() {
            self.chosenFolderId(this.params.folder);
            self.chosenFolderData(null);
            $.get("/mail", { mailId: this.params.mailId }, self.chosenMailData);
        });

        this.get('', function() { this.app.runRoute('get', '#Inbox') });
    }).run();
};

ko.applyBindings(new WebmailViewModel());
```

Obrázok č. 3: Štruktúra Knockout

6. JavaScript

Táto sekcia je prázdna. Zatiaľ bola identifikované využitie Javascriptu len v rámci frameworku KnockOut.

7. Základné rozloženie stránky

Hlavná stránka klientskej zóny bude rozdelená na 3 hlavné časti:

1. Header – hlavička, ktorá bude obsahovať informácie o klientskom účte
2. Sidebar – menu s dôležitými funkciami aplikácie

3. Content – obsah konkrétnej stránky

8. Formátovanie prvkov na stránke

Všetky štýly sa budú ukladať do css súborov, ktoré sa potom zavolajú v hlavičke html stránky, napr. *<link rel="stylesheet" href="css/vendor.css">*.

8.1. Typografia

8.1.1. Fonty a základné nastavenie textu

Na stránkach sa budú implicitne používať generické fontové rodiny typu *sans-serif*. Štruktúra časti *body* v css súbore bude vyzerať nasledovne:

```
body {
  font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, "Helvetica Neue", Arial, sans-serif;
 font-size: 1rem;
 font-weight: normal;
 line-height: 1.5;
 color: #212529;
 background-color: #fff;
}
```

Vzhľadom na to, aby sa zlepšil výkon používanej aplikácie sa budú používať systémové fonty. Preto je font-family nastavená týmto spôsobom. Takisto sa tento font bude používať aj v texte vyskakovacích okien a tooltipov.

8.1.2. Hlavičky

Nadpisy určitých sekcií a takisto nadpisy v bežnom texte budú formátované a označované ako hlavičky. Veľkosti daných hlavičiek sú znázornené nižšie v tabuľke:

Hlavička	Veľkosť hlavičky
H1	font-size: 2.5rem;
H2	font-size: 2rem;
H3	font-size: 1.75rem;
H4	font-size: 1.5rem;

H5	font-size: 1.25rem;
H6	font-size: 1rem;

Tabuľka č. 1: Prehľad formátovania veľkosti textu hlavičiek

Ďalšie informácie sú spoločné pre všetky typy hlavičiek:

```
h1, h2, h3, h4, h5, h6 {
  margin-bottom: 0.5rem;
  font-family: inherit;
  font-weight: 500;
  line-height: 1.1;
  color: inherit;
}
```

Kde farbu a font budú dediť z rodičovských elementov.

8.1.2. Nastavenie textu

Paragrafy

Súvislý text bude písaný vo forme 4 typov paragrafov, ktoré sa použijú podľa potreby. Text v paragrafe vložte do tagov podľa tabuľky:

Typ paragrafu - tag	Css vlastnosti
small	*small {* *font-size: 80%;* *}*
strong, b	*b, strong {* *font-weight: bolder;* *}*
em	

Tabuľka č. 2: Prehľad formátovania paragrafov

Small tag predstavuje zmenšený text, strong alebo b (bold) tučný text a em pre emphasized alebo italic.

Odkazy

Bežne sa budú používať odkazy v texte, ktoré budú mať tieto vlastnosti:

```
a {
  color: #007bff;
  text-decoration: none;
  background-color: transparent;
  -webkit-text-decoration-skip: objects;
```

```
}

a:hover {
  color: #0056b3;
  text-decoration: underline;
}
```

Zvýraznenie textu podľa typu správy

Ide taktiež o súvislý text napísaný paragrafom, ale pri týchto špeciálnych výpisoch sa ďalej použijú triedy podľa typu správy, a tým sa zmení farba daného textu. Každá trieda musí byť zadefinovaná, konkrétne takto (príklad):

Použitie:

```
<p class="text-muted">This is an example of muted text.</p>
```

Css súbor:

```
.text-muted {
  color: #868e96 !important;
}
```

Triedy môžete vidieť v tabuľke:

Trieda	Veľkosť hlavičky	Farba textu
text-muted	*.text-muted { color: #868e96 !important; }*	sivá
text-primary	*.text-primary { color: #007bff !important; }*	modrá
text-success	*.text-success { color: #28a745 !important; }*	zelená
text-info	*.text-info { color: #17a2b8 !important; }*	tyrkysová
text-warning	*.text-warning { color: #ffc107 !important; }*	oranžová
text-danger	*.text-danger { color: #dc3545 !important; }*	červená

Tabuľka č. 3: Prehľad správ podľa typu

Zoznamy

Zoznamy budeme používať očíslované a neočíslované, kde hlavička bude typu h4. Neočíslované zoznamy sú označené párovým tagom , zatiaľ čo očíslované zoznamy sú označené párovým tagom . V oboch prípadoch platí, že sa položky pridávajú párovým tagom . Medzi jednotlivými položkami môžu vznikať kombinácie, ako vidíte na príklade:

```
<ul>
        <li>List Item</li>
        <li>List Item</li>
        <li>
                <ol>
                        <li>List Item</li>
                        <li>List Item</li>
                </ol>
        </li>
</ul>
```

8.1.2. Tlačidlá

Tlačidlá sa budú na html stránke zapisovať týmto štýlom:

<button type="button" class="(nazov triedy – budú použité viaceré)">(text napísaný na tlačidle)</button>