

Міністерство освіти і науки України
Прикарпатський національний університет імені Василя Стефаника
Представництво "Польська академія наук" в Києві
Науково-технологічний університет "Гірничо-металургійна академія"
імені Станіслава Сташіца в Кракові
Вінницький національний технічний університет
Харківський національний університет радіоелектроніки
Національний авіаційний університет
Тернопільський національний економічний університет
Фінансово-економічний інститут Таджикистану
Економічна академія "Д.А.Ценов", Болгарія
Лудзький університет, Польща
Штутгартський університет, Німеччина
Інститут інженерів з електротехніки та електроніки (ІЕЕЕ), Українська секція
Громадська організація "Івано-Франківський ІТ кластер"

КОМП'ЮТЕРНІ НАУКИ, ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ УПРАВЛІННЯ

**Матеріали
міжнародної науково-технічної конференції
молодих вчених, аспірантів та студентів**

**28–30 листопада 2018 року
Івано-Франківськ, Україна**

**COMPUTER SCIENCE, INFORMATION TECHNOLOGIES
AND MANAGEMENT SYSTEMS**

**Proceedings
of the International Scientific Young Scientists Conference**

**2018, November , 28th to 30th
Ivano-Frankivsk, Ukraine**

Івано-Франківськ
Прикарпатський національний університет імені Василя Стефаника
2018

Модифікація Засобів Open-Source для Підвищення Ефективності Розробки Програмного Забезпечення

Залізецький Василь Володимирович
кафедра обчислювальної техніки
Вінницький національний технічний університет
м. Вінниця, Україна
zwww@i.ua

The Open-Source Tools Modification to Improve Software Development Efficiency

Vasyl Zalizetskyi
department of computer technology
Vinnytsia National Technical University
Vinnytsia, Ukraine
zwww@i.ua

Анотація—У публікації розглянуто питання модифікації таких засобів розробки програмного забезпечення, як Bootstrap, Swagger та Handlebars. Проаналізовано особливості роботи та функціональні можливості цих засобів, а також вказано на їх недоліки. Запропоновано шляхи ліквідації недоліків окремо для кожного з цих засобів, що дозволило підвищити ефективність їх використання як окремо так і в комплексі.

Abstract—The publication addresses the modifying software development tools such as Bootstrap, Swagger, and Handlebars. Analyzed the features of work and functionality and the shortcomings of the mentioned software are identified. The ways of eliminating these disadvantages separately for each of these funds have been proposed, which made it possible to increase the efficiency of their use both separately and in a complex.

Ключові слова—підвищення ефективності, розробка програмного забезпечення, open-source, фреймворки, Bootstrap, Swagger, Handlebars

Keywords— increase efficiency, software development, open source, frameworks

I. ВСТУП

Ефективність розробки програмного забезпечення визначається прийняттям рішень на різних етапах, починаючи з вибору інструментів і засобів розробки. Від цього залежить архітектура програмного забезпечення, структура і представлення даних, функціональні можливості та швидкість розробки.

За останнє десятиліття з'явилося багато різноманітних засобів та інструментів розробки програмного забезпечення. Значне місце серед таких засобів займають фреймворки, що диктують правила побудови архітектури додатку, задають поведінку різних компонентів за промовчуванням та формують каркас. Фреймворки можуть включати допоміжні програми, бібліотеки, мови сценаріїв та інші компоненти, що полегшують розробку програмного забезпечення.

Проте при використанні фреймворків часто виникає потреба розширювати і змінювати їх функціональність відповідно до вимог певного проекту. В публікації розглядаються такі фреймворки, як Bootstrap, Swagger і Handlebars та пропонуються загальні модифікації для підвищення ефективності, що можуть використовуватись при розробці широкого кола проектів.

II. АНАЛІЗ ЗАСОБІВ OPEN-SOURCE

A. Реалізація адаптивності з використанням Bootstrap

Bootstrap — це фреймворк з відкритим кодом, призначений для створення веб-сайтів та веб-додатків, містить шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення JavaScript [1].

Для того, щоб зрозуміти, як працює адаптивність, потрібно розібратись з основними будівельними блоками та властивостями Bootstrap. Для створення, наприклад трьох однакових по-ширині колонок, які будуть ставати в один ряд на широких та середніх моніторах, потрібно використати класи вказані на рисунку 1. На мобільних пристроях і планшетах ці колонки будуть автоматично складатись.

.col-md-4	.col-md-4	.col-md-4
-----------	-----------	-----------

Рис. 1. Три однакові колонки Bootstrap сітки

Створення двох колонок з вкладеними колонками, які будуть ставати в один ряд на широких та середніх моніторах зображено на рисунку 2. На мобільних пристроях ці колонки (в тому числі і вкладені) будуть складатись.

.col-md-8		.col-md-4
.col-md-6	.col-md-6	

Рис. 2. Дві колонки, які вміщують ще дві колонки

Система сітки Bootstrap має чотири рівні класів: xs (телефони), sm (планшети), md (настільні комп'ютери), та lg (великі екрани). Можна поєднувати майже будь-яку комбінацію цих класів, для створення більш динамічних та гнучких макетів. При цьому сума значень колонок не повинна перевищувати число дванадцять.

ТАБЛИЦЯ 1. СИСТЕМА СІТКИ BOOTSTRAP ДЛЯ РІЗНИХ ПРИСТРОЇВ

Пристрої	Телефони (<768px)	Планшети (≥768px)	Середні монітори (≥992px)	Широкі монітори (≥1200px)
Префікс класу	.col-xs-	.col-sm-	.col-md-	.col-lg-
Поведінка сітки	Горизонтальна завжди	Складена спочатку, але горизонтальна, коли ширша за контрольні точки		
Кількість колонок	12			
Ширина колонки	Автоматична	~62px	~81px	~97px
Ширина проміжку	30px (15px з кожної сторони колонки)			
Можуть вкладатись	Так			
Можуть зміщуватись	Так			
Порядок колонок	Так			

Сітка Bootstrap передбачає, лише один клас для мобільних телефонів - col-xs, але більшість сучасних телефонів обладнанні акселерометром, що дозволяє змінювати орієнтацію зображення в залежності від положення телефону у просторі. А це створює певні труднощі, оскільки те зображення, яке адекватно виглядає при альбомному перегляді на телефоні, виглядає дуже стиснутим при перегляді в портретному режимі та текст накладається з одного блоку на інший.

Окрім того при використанні мобільної версії браузера Chrome, розмір вікна браузеру часто становить лише 640x360 у альбомному режимі та 360x640 у портретному режимі відповідно. При цьому розширення самого екрану може бути 1920x1680.

У зв'язку з цим, автором публікації пропонується рішення для кращої підтримки мобільних пристроїв, що дозволить скоротити часові затрати на реалізацію адаптивності.

В. Реалізація інтерактивної документації з використанням Swagger

Swagger являє собою набір скриптів, які генерують інтерактивну документацію для web-додатків з REST API. Інтерактивність проявляється в тім, що зі сторінки з документацією можна робити HTTP-запити, а сама документація оновлюється з коментарів, що написані у кодї програми [2].

Під час використання цього інструменту знайдено значний недолік, а саме, те, що обов'язково потрібно описувати варіанти відповідей від серверу, включаючи коди помилок та самі моделі об'єктів, що може повертати сервер. Приклад опису варіантів відповідей від серверу наведено нижче:

```
# responses:
# 200:
# description: Token response
```

```
# schema:
# type: object
# properties:
# token:
# type: string
# description: Authorization token that has to be placed into `X-Auth-Token` header
# 400:
# description: Wrong request format
# schema:
# $ref: '#/definitions/errorResponse'
# 403:
# description: Authorization data is invalid or user not confirmed
# schema:
# $ref: '#/definitions/errorResponse'
# 404:
# description: Specified user is not found
# schema:
# $ref: '#/definitions/errorResponse'
```

Навіть якщо з самими запитом проблем немає, але варіанти відповідей від серверу не описані то при запиті одержуємо помилку “Could not render Operation” Автором пропонується зробити опціональним опис варіантів відповідей від серверу. Це дозволить значно скоротити час на реалізацію інтерактивної документації.

С. Побудова шаблонів сторінок з використанням Handlebars

Handlebars є одним з найбільш популярних, швидких і багатofункціональних шаблонізаторів для JavaScript [3]. Handlebars разом з Ajax є чудовою альтернативою застарілим технологіям, таким, як: JSP, JSTL і org.springframework.web.servlet.ModelAndView.

Після одержання відповіді від серверу у форматі JSON, Handlebars дозволяє подібно до JSTL працювати з різними об'єктами, наприклад опрацьовувати колекції даних з використанням тегу `{{#each ...}}`. В середині тегу `{{#each ...}}` можна одержати індекс елемента з використанням `{{@index}}`. Фрагмент коду:

```
{{#each this}}
<tr>
<td>{{@index}}</td>
<td>{{objectName}}</td>
<td><a id="editObject" data-id="{{id}}"
class="material-icons icon-styled">edit</i></a></td>
<td><a id="deleteObject" data-id="{{id}}"
class="modal-trigger">delete_forever</a></td>
</tr>
{{/each}}
```

Але при відображенні списку елементів люди звикли бачити не id елементів і не індекс масиву, що починається з 0, а звичайну нумерацію починаючи з 1. На момент написання цієї публікації Handlebars не надавав таку можливість, тому автором пропонується така функціональність, що дозволить в подальшому скоротити час на пошук шляхів реалізації подібної функціональної можливості.

III. ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ЗАСОБІВ OPEN-SOURCE

А. Модифікація Bootstrap

Для подолання описаної проблеми, реалізовано додатковий клас сітки для Bootstrap.

Новий клас отримав назву .col-lx. Оскільки col-xs-означає extra small devices, то клас .col-lx означає less

than extra small, тобто менше ніж стандартний мобільний альбомний режим.

Реалізовано це наступним чином:

```
.col-lx-1, .col-lx-2, .col-lx-3, .col-lx-4, .col-lx-5, .col-lx-6, .col-lx-7, .col-lx-8, .col-lx-9, .col-lx-10, .col-lx-11, .col-lx-12 .col-xs-1 {
  position: relative;
  padding-right: 15px;
  padding-left: 15px;
}
.col-lx-1, .col-lx-2, .col-lx-3, .col-lx-4, .col-lx-5, .col-lx-6, .col-lx-7, .col-lx-8, .col-lx-9, .col-lx-10, .col-lx-11, .col-lx-12 {
  float: left;
}
.col-lx-12 {
  width: 100%;
}
.col-lx-11 {
  width: 91.66666667%;
}
.col-lx-10 {
  width: 83.33333333%;
}
.....
.col-lx-3 {
  width: 25%;
}
.col-lx-2 {
  width: 16.66666667%;
}
.col-lx-1 {
  width: 8.33333333%;
}
```

Адаптивність працює не тільки завдяки відсотковому вказанню ширини для різних блоків, основний рушій адаптивності це використання медіа-селекторів @media.

Чому в поданому вище CSS-кодї, немає медіа-селектору, можна пояснити тим, що CSS-файл умовно ділиться на медіа-блоки, наступним чином:

```
@media (min-width: 768px) { .col-sm-... }
@media (min-width: 992px) { .col-md-... }
@media (min-width: 1200px) { .col-lg-... }
```

Як, видно із наведено прикладу, тут немає медіа-селектору для найменшого стандартного блоку .col-xs, тобто він використовується за замовчуванням, а при більших екранах спрацьовує відповідний медіа-селектор.

За таким же принципом реалізовано новий найменший блок .col-lx, із заданням медіа-селектору для старого найменшого блоку .col-xs:

```
@media (min-width: 640px) { .col-xs-... }
```

В. Модифікація Swagger

Зміни стосуються файлу src/core/components/operation.jsx в репозиторії swagger-ui.

```
import { OrderedMap } from "immutable"
```

```
До функції render() потрібно додати
if(responses === undefined) responses = new OrderedMap
if(response && response.size > 0) {
  let notDocumented = (responses !== undefined)?
!responses.get(String(response.get("status"))): true
```

```
response = response.set("notDocumented", notDocumented)
}
```

Запропоновані правки роблять опис варіантів відповідей від серверу опціональним, що дозволяє значно прискорити процес налаштування інтерактивної документації для своїх проектів. Після внесення змін і перевірки ресурсів засобами прт помилка більше не виникатиме.

С. Модифікація Handlebars

Для вирішення описаної проблеми, можна зареєструвати свій власний helper:

```
Handlebars.registerHelper({
  'math': function(lvalue, operator, rvalue, opts) {
    lvalue = parseFloat(lvalue);
    rvalue = parseFloat(rvalue);
    return {
      "+": lvalue + rvalue
    }[operator];
  });
});
```

І далі робити так `{{math @index "+" 1}}`, але цей варіант спрацьовує не завжди і при оновленні сторінок, деколи ми отримуємо помилку Missing helper: math. Тому автором тез пропонується модифікація handlebars.js, а саме функції execIteration:

```
function execIteration(field, index, last) {
  if (data) {
    data.key = field;
    data.index = index;
    data.number = index + 1;
    data.first = index === 0;
    data.last = !last;
    if (contextPath) {
      data.contextPath = contextPath + field;
    }
  }
  ret = ret + fn(context[field], {
    data: data,
    blockParams: _utils.blockParams([context[field], field], [contextPath + field, null])
  });
}
```

Після такої модифікації можна використовувати `{{@number}}`, де потрібно вивести номер елемента в списку.

REFERENCES ЛІТЕРАТУРА

- [1] Bootstrap - The most popular HTML, CSS, and JS library in the world. [Електронний ресурс]. —Режим доступу: <https://getbootstrap.com>, вільний (дата звернення: 16.11.2018). — Назва з екрана.
- [2] The Best APIs are Built with Swagger Tools [Електронний ресурс]. —Режим доступу: <https://swagger.io>, вільний (дата звернення: 16.11.2018). —Назва з екрана.
- [3] Handlebars.js: Minimal Templating on Steroids [Електронний ресурс]. —Режим доступу: <https://handlebarsjs.com>, вільний (дата звернення: 16.11.2018). —Назва з екрана.
- [4] О. Азаров, О. Черняк, і Л. Савицька, Аспекти критичного підходу до викладання поняття поліморфізму в об'єктно-орієнтованому програмуванні, ІТКІ, vol 39, № 2, с. 31-34, Жов 2017.