

ЗАХИЩЕНІСТЬ СУЧАСНИХ JAVASCRIPT ФРЕЙМВОРКІВ ДЛЯ РОЗРОБКИ ОДНОСТОРІНКОВИХ ВЕБ-ЗАСТОСУНКІВ

М. А. Южаков^{1, а}

¹Національний технічний університет України «Київський політехнічний інститут»

Анотація

В роботі представлені основні принципи побудови сучасних односторінкових веб-застосунків, типові фреймворки для їх побудови, розглянуті вразливості цих фреймворків, та надані рекомендації щодо застосування цих фреймворків, а також рекомендації щодо вибору кращого фреймворку з точки зору безпеки для побудови односторінкових веб-застосунків.

Ключові слова: JavaScript, односторінковий веб-застосунок, фреймворк, вразливість

Вступ

Веб-застосунки, або як їх раніше називали «веб-сайти» за весь час свого існування еволюціонували достатньо сильно. З набору окремих сторінок, яких зв'язують посилання вони перетворилися в односторінкові веб-застосунки (англ. *Single-page Application*), далі SPA. Фреймворки стали архітектурною основою для сучасного етапу розвитку веб-застосунків. Очевидною є важливість використання найактуальніших методів захисту. Метою даної роботи є аналіз типових фреймворків, що використовуються при побудові SPA, аналіз їх вразливостей, а також надання рекомендацій щодо вибору кращого фреймворку з точки зору безпеки.

1. Об'єкти дослідження

Об'єктами дослідження є JavaScript фреймворки, що використовуються для побудови SPA.

SPA – це збірна назва набору технологій, що дозволяють розробити веб-застосунок, який виконується веб-браузером як одна веб-сторінка, як, наприклад, реалізований сервіс Gmail від Google. З точки зору користувача, дана технологія забезпечує в першу чергу швидкість відгуку на дії в інтерфейсі, та відсутність повного перезавантаження веб-сторінки з сервера. Всі елементи конструюються прямо в браузері за допомогою JavaScript. Таким чином, веб-додатки стають дуже схожі на звичайні програми для робочих станцій, що завантажують інформацію з мережі Інтернет, тільки середовищем виконання для них є не операційна система, а браузер, який в результаті змушений нести на собі навантаження, а саме управління пам'яттю, забезпечення безпечного оточення, надання функціоналу для роботи з системними функціями тощо [1].

Архітектурним фундаментом будь-якого SPA є фреймворк (програмний каркас) – готовий до використання комплекс програмних рішень, включаючи дизайн, логіку та базову функціональність [2].

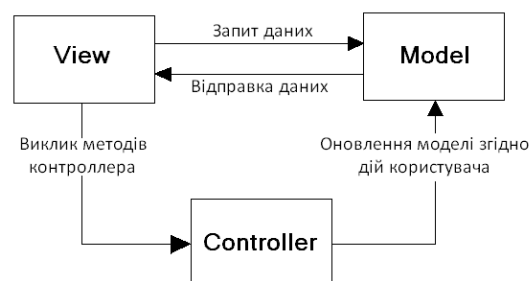


Рис. 1. Діаграма шаблону проектування MVC

Фреймворк є готовим для застосування шаблоном проектування SPA. В сучасних веб-застосунках використовуються три шаблони проектування:

- Model-View-Controller
- Model-View-View-Model
- Model-View-Presenter

1.1. Шаблон проектування MVC

Model-View-Controller або MVC – це фундаментальний шаблон, який знайшов застосування в багатьох технологіях, і кожен день полегшує життя розробникам веб-застосунків [3]. Концепція MVC дозволяє розділити дані, представлення та обробку дій користувача на три компоненти:

- Модель (англ. *Model*) – надає дані та методи роботи з цими даними, реагує на запити, змінюючи свій стан.
- Представлення (англ. *View*) – відповідає за відображення інформації (візуалізацію).
- Контролер (англ. *Controller*) – контролює введення даних користувачем і використовує модель та представлення для реалізації необхідної реакції.

1.2. Обрані фреймворки

Як було згадано вище, саме фреймворк є готовим для застосування шаблоном проектування. В на час

^аyuzhakov.nikita@gmail.com

Табл. 1. Результати аналізу обраних фреймворків

	Angular	Backbone	Ember
Залежності	немає	Underscore та jQuery	jQuery та Handlebars
Відомі вразливості	0	0	0
Виправлені вразливості	12	2	10
Кількість зірок на GitHub	49,097	24,741	16,125
Наявність політики безпеки	ні	ні	так
Наявність контактів для питань з безпеки	так	ні	так
Наявність документації щодо безпеки	так	ні	ні
Вбудовані механізми безпеки	так	ні	ні

саме MVC є більш розповсюдженим шаблоном, а отже для аналізу були вибрані фреймворки, що використовують цей шаблон. Конкретні фреймворки обиралися за такими властивостями:

- актуальність;
- поширеність.

Згідно цих критеріїв з декількох актуальних [4] для подальшого аналізу були вибрані такі фреймворки:

- Angular
- Ember
- Backbone

2. Результати дослідження

Кожен фреймворк підходить до задачі створення односторінкових веб-застосунків трохи по-різному і пропонує різні компроміси. Щоб проаналізувати безпечність кожного з них, були обрані та проаналізовані критерії, які приведені у табл. 1.

2.1. Angular

Всі проаналізовані фреймворки в даний час оперативно розвиваються, тому показник актуальних вразливостей завжди буде на рівні близькому до нуля. Angular має найбільшу кількість виявлених в минулом вразливостей, що залежить від того, наскільки розробники враховували безпеку під час написання продукту. Також цей показник залежить від популярності фреймворку (кількість зірок на GitHub), яка в Angular найбільша. Це значно впливає на кількість знайдених помилок – чим більше людей використовує фреймворк, тим більшою кількістю він буде протестований.

Angular пропонує одну сторінку документації, але політика безпеки відсутня. Має контакти для питань з безпеки, які перераховані веб-сайті.

Він використовує свої власні шаблони і не має залежностей. Також має свою власну версію виразів JavaScript, яка є більш суворою і виконується в пісочниці.

2.2. Ember

У Ember в милому було знайдено 10 вразливостей, але його популярність найменша серед обраних фреймворків. Можна зробити висновок, що його тестувала менша кількість людей, але була знайдена більша кількість помилок.

Наявність залежностей означає, що їх захищеність в цілому напряму впливає на фреймворк. Це ж саме стосується і Backbone.

Він має політику безпеки на своєму веб-сайті та контакти що для вирішення питань з порушення безпеки.

В ньому відсутня пісочниця, а отже розробникам необхідно самостійно реалізовувати належне використання виразів JavaScript при вставці в шаблони на стороні клієнта.

2.3. Backbone

Наступним є Backbone, який має найменшу кількість знайдених помилок і середню популярність.

Він не має політики безпеки, а також не пропонує документацію про те, як створювати захищені веб-застосунки з його допомогою. Має залежності від jQuery та Underscore.

Також в ньому відсутнє поняття пісочниці, а отже розробникам, що використовують Backbone, самим необхідно піклуватись про JavaScript вирази, що можуть бути розміщені всередині шаблонів.

Висновки

Всі проаналізовані фреймворки вирішують подібні проблеми по-різному. Проаналізувавши результати всіх критеріїв оцінювання, можна зробити висновок, що з точки зору безпеки Angular виявився більш захищеним, хоча Ember та Backbone лише трохи йому поступаються. На це слід звертати увагу, вибираючи фреймворк для побудови односторінкового веб-застосунку та знаючи про можливі обмеження проектних рішень.

Перелік використаних джерел

1. Fernando Monteiro. Learning Single-page Web Application Development. — Packt Publishing Ltd. — 2014. — с.27.
2. Dirk Riehle. Framework Design: A Role Modeling Approach. — Swiss Federal Institute of Technology. — 2000. — с.38.
3. Martin Fowler. The evolution of MVC and other UI architectures. — 2006. — с.11.
4. Martin Angelov. The Languages And Frameworks You Should Learn In 2016. — 2015. — с. 2.