

10.7. JavaScript фреймворки

JavaScript - це мова веб-програмування, яка допомагає веб-розробникам зробити статичні веб-сторінки більш динамічними для покращення досвіду користувача. В останні роки на піку популярності знаходяться JavaScript-фреймворки, які надають "дорожню карту" для створення клієнтських програм. Ці фреймворки стали популярні у веб-розробці, оскільки вони забезпечують кращий досвід користувача завдяки системам поділу коду.

При розробці фронтенду веб-додатків важливо знати, як кодувати ці JavaScript-фреймворки. Під час роботи знадобляться знання про такі бібліотеки, як React, Angular або Svelte.

JavaScript

JavaScript - це багатшаблонна мова програмування для подієво-орієнтованих, об'єктно-орієнтованих та заснованих на прототипах веб-додатків. Спочатку код JavaScript був орієнтований тільки на клієнтську сторону, але останніми роками він використовується і на серверній стороні Інтернету.

Сьогодні JavaScript є популярною мовою програмування в Інтернеті. Це єдина мова в Інтернеті, для якої є кілька фреймворків. JavaScript-фреймворки дозволяють користувачам створювати масштабовані веб-застосунки за допомогою стандартних інструментів. Вони мають кілька бібліотек коду JavaScript із шаблонами коду JS, які допомагають виконувати стандартні завдання програмування. Більшість людей плутають веб-фреймворки з бібліотеками через їхні функції, але вони відрізняються за своїми завданнями та масштабом.

Бібліотека JavaScript містить упакований код, класи, методи та функції, які можна повторно використовувати для розробки веб-програми. Ці бібліотеки пропонують JavaScript, який розробники можуть використовувати повторно. На відміну від JavaScript-коду бібліотек JavaScript, JavaScript-фреймворки визначають дизайн веб-додатків, викликають JS-бібліотеку та використовують код у ній для завершення програми. Ці фреймворки пропонують повну дорожню карту для створення веб-застосунків замість одного рішення. Розробники додали ці фреймворки у свій набір інструментів для розробки веб-рішень для своїх клієнтів. Перевагою використання цих фреймворків є у можливості створювати багато веб-сторінок, багаторазово використовуючи їх компоненти.

До появи сучасних фреймворків популярною бібліотекою, яку використовували розробники для написання коду JavaScript, сумісного зі всіма веб-браузерами, був jQuery. Всі функції були абстраговані у просту та легку в освоєнні бібліотеку, яку розробники використовували на веб-сторінках.

jQuery започаткував сучасні фреймворки, такі як React, Backbone, Angular та Ember, які забезпечують архітектуру для створення нових веб-додатків з динамічними сторінками.

Роль фреймворків JavaScript у веб-додатках

Фреймворк JavaScript надає схему, що допомагає розробникам створювати програми швидше та дешевше. Використовуючи ці фреймворки можна писати код без помилок.

За допомогою їх макетів можна розгортати високо інтерактивні веб-додатки та створювати різні частини інтерфейсів користувача, щоб повторно використовувати ці компоненти незалежно один від одного.

Фреймворки також допомагають розробникам:

- Реалізувати компоненти інтерфейсів користувача на основі даних.
- Вирішити проблеми фронтенду.
- Написання коду, що багаторазово використовується і легко читається.
- Повторне використання інкрементів для створення програм.
- Абстрагування ідей у зрозумілу мову.
- Стандартизація елементів інтерфейсу користувача, таких як колірна схема, кнопки і стиль шрифту, щоб зробити користувацький досвід кращим.

Фреймворки допомагають розробникам використовувати мови, специфічні для конкретної області, щоб зберегти код у більш зрозумілому форматі. Всі ці функції спрощують розробку для розробників завдяки інтеграції інструментів для тестування та налагодження коду. Крім того, ці JavaScript-фреймворки допомагають розробникам автоматизувати встановлення, оновлення та налаштування бібліотек без помилок.

Без сумніву, JavaScript-фреймворки популярні, але є завдання, в яких їх використання не обов'язково. Припустимо, розробка веб-програми проста і не вимагає складних функцій. У таких випадках використання фреймворку не рекомендується. Фреймворки пропонують багато функцій, але вони мають свою цінову політику. Ці фреймворки, безсумнівно, забезпечують декларативніший код, але в результаті можна отримати додаток з вищими обчислювальними витратами, ніж звичайний JavaScript.

Для створення невеликого проекту, який на потребує інтерактивності від користувача, фреймворк краще не використовувати. У цьому випадку можна просто використовувати JavaScript для створення програм.

Переваги використання JavaScript Frameworks

- **Забезпечують сильну підтримку спільноти.** Завдяки популярності фреймворків, утворюються спільноти розробників з площадками для обговорення проблем, де можна знайти вирішення конкретної проблеми.
- **Можливість повторного використання.** Оскільки фреймворки мають вбудовані методи, можна повторно використовувати шаблон для розробки програм, не написавши жодного рядка коду.
- **Швидка розробка.** Використання JavaScript-фреймворку дозволяє користувачам швидше створювати та запускати програми, заощаджуючи час.
- **Скорочення витрат.** Більшість фреймворків JavaScript безкоштовні для використання та скорочують витрати на розробку. Використання фреймворків є економічно вигідним, особливо для власників малого бізнесу.

Вибір JavaScript фреймворка для проекту

При розробці веб-застосунків необхідно використовувати сучасні фреймворки. Існує багато фреймворків, доступних розробникам для створення додатків, серед яких популярними є:

1. React
2. Svelte
3. Vue
4. Angular

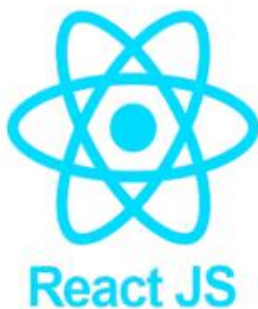
5. Ember
6. Backbone
7. NextJS

Перш ніж вибрати фреймворк, переконайтеся в наступному:

- Для фреймворка є підручники та посібники для початківців.
- Вибраний фреймворк, прийнятий розробниками протягом багатьох років.
- Вибраний фреймворк має сильну спільноту та документацію.

React JS (<https://uk.reactjs.org/>)

React JS React - це фреймворк з відкритим кодом, що належить Facebook (Meta). React - це вільний і декларативний фреймворк, що широко використовується для розробки інтуїтивних інтерфейсів для веб-додатків. Використовується для швидкого створення додатків, що масштабуються. React має невеликі пакети, тому його легко засвоїти початківцям. Фреймворк React є доцільним для розробки великомасштабних веб-додатків, що використовують різні дані. Спочатку React використовувався у Facebook і використовуватиметься для стрічок Instagram. Також, можна використовувати для великих веб-проектів із невеликою командою.



```
JS index.js X
src > JS index.js > ...
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import './index.css';
4  import App from './App';
5  import * as serviceWorker from './serviceWorker';
6
7  var element = React.createElement('h1', { className: 'greeting', 'Hello, world!'});
8  ReactDOM.render(element, document.getElementById('root'));
9
10 // If you want your app to work offline and load faster, you can change
11 // unregister() to register() below. Note this comes with some pitfalls.
12 // Learn more about service workers: https://bit.ly/CRA-PWA
13 | serviceWorker.unregister();
14
PROBLEMS 1 OUTPUT ... Filter: E.g.: text, **/*.ts, !**/node_modules/**
v JS index.js src 1
  x ' expected. ts(1005) [13, 26]
```

Рис.1. Інтерфейс фреймворку React JS

Згідно з опитуванням State of JavaScript, React є найбільш використовуваним фреймворком. React.js забезпечує швидкий інтерфейс користувача для миттєвого створення веб-додатків за допомогою компонентів, а не оновлення окремих частин, як це робить бібліотека. Цей фреймворк дозволяє інкрементально використовувати компоненти для швидшого оновлення веб-сторінок. React також надає інтерфейс користувача під назвою JSX для більш зручного використання.

Особливості фреймворку React:

- **Дозволяє інкрементне використання компонентів.** React дозволяє користувачам розділяти веб-сторінку на багато компонентів. Кожна частина відноситься до

інтерфейсу користувача, має власний код і структуру і допомагає розробникам створювати веб-додатки з багаторазово використовуваним кодом.

- **Робота з віртуальною об'єктною моделлю документа (DOM).** React надає віртуальну об'єктну модель документа (DOM) для представлення даних у деревоподібній структурі. React також поділяє дані на компоненти, що містять код для кожного незалежного елемента інтерфейсу користувача в документі. Наявність DOM у фреймворку React дозволяє оновлювати лише частину веб-сторінки, а не весь сайт.
- **Підтримка JavaScript XML (JSX).** JavaScript XML (JSX) схожий на вбудовування HTML у JavaScript. Ці виклики допомагають веб-розробникам вбудовувати модулі JavaScript у елементи HTML.
- **Забезпечення динамічного інтерфейсу користувача (UI).** JavaScript дозволяє контролювати дизайн та потік програми, використовуючи синтаксис JSX у стилі HTML. Вибір веб-архітектури допомагає вирішити, як має виглядати ваш веб-додаток.

Svelte (<https://svelte.dev/>)

Svelte - це JavaScript-фреймворк з відкритим вихідним кодом, який допомагає перетворити необроблений код статичних веб-додатків на інтуїтивно зрозумілі та інтерактивні інтерфейси користувача. Svelte – це більше, ніж фреймворк; він працює як компілятор, що перетворює код Svelte на ванільний JavaScript для отримання більш швидких веб-рішень, ніж VueJS і React.

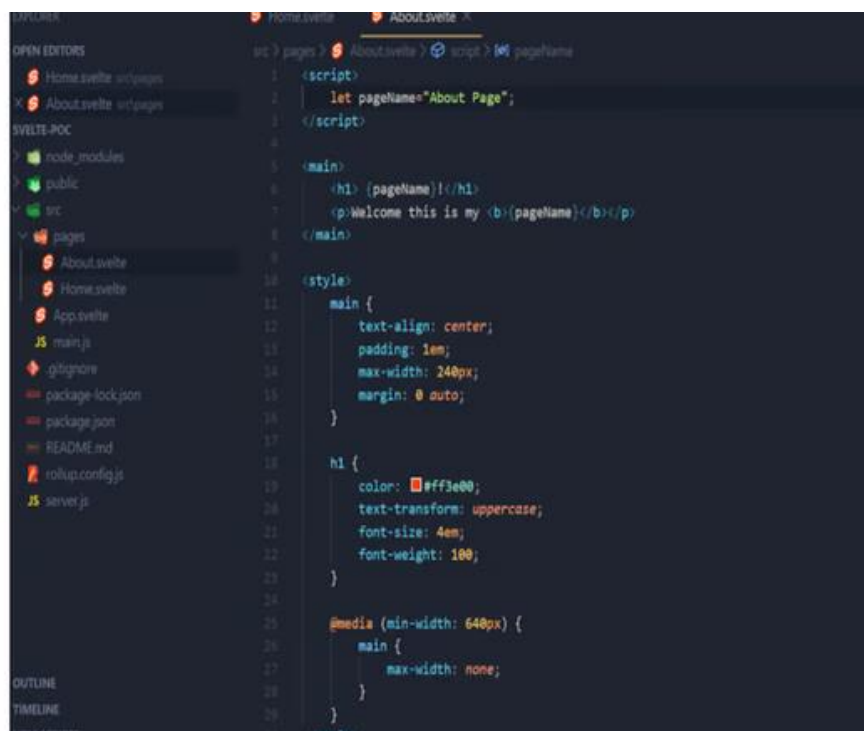


Рис.2. Інтерфейс фреймворку Svelte

Svelte компілює код як незалежні компоненти JavaScript і завантажує програми швидше без особливих завдань браузера.

Особливості Svelte:

- **Робота без віртуального DOM.** На відміну від React, фреймворк Svelte працює з кодом без Document Object Module (DOM) і переносить більшу частину коду на компіляцію, досягаючи більш швидких результатів, ніж інші фреймворки.

- **Забезпечує реактивність.** Фреймворк Svelte перетворює модулі на функції DOM, які реагують на зміни даних та відображають модифікації у вигляді коду JavaScript.
- **Вимагає мінімального коду від веб-розробника.** Фреймворк Svelte вимагає мінімальної кількості коду, ніж React та Vue, що допомагає веб-розробникам витратити більше часу на інші завдання, крім кодування.
- **Забезпечує модульну стилізацію веб-сторінок.** Svelte забезпечує послідовний дизайн усіх веб-сторінок додатків за рахунок додавання унікальних стилів та створення унікальних імен для класів.

Vue (<https://vuejs.org/>)

Vue - фреймворк з відкритим вихідним кодом, який допомагає створювати інтуїтивно зрозумілі інтерфейси користувача, поєднує популярні функції Angular і React. Він використовує шаблони Angular та прив'язку даних, а також реквізити React.

Фреймворк Vue має відмінну особливість від інших фреймворків, оскільки його можна поступово переймати для створення програм. До інкрементів входять рішення для маршрутизації, інструментарій, командне введення (CLI) та управління станом.



```

App.vue
1  <template>
2    <div id="app">
3      
4      <HelloWorld msg="Welcome to Your Vue.js App"/>
5    </div>
6  </template>
7
8  <script>
9    import HelloWorld from './components/HelloWorld.vue'
10
11    export default {
12      name: 'app',
13      components: {
14        HelloWorld
15      },
16      met
17    }
18  </scr
19
20  <style>

```

Рис.3. Інтерфейс фреймворку Vue

Цей фреймворк пропонує модулі, що інкрементально впроваджуються, тому можна кодом додавати нові інкременти у свій веб-проект. Архітектура цього фреймворку проста у використанні, і не потрібно розумітися на всіх функціях. Vue вимагає великих знань JavaScript, HTML і CSS для створення масштабних веб-додатків та об'єднання їх у нові або вже існуючі програми за допомогою JSX.

Особливості Vue

- **Забезпечення інтерактивності.** Vue дозволяє додавати ефекти переходу, щоб поживавити веб-сторінки. Можете додати сторонні бібліотеки анімації для більшої інтерактивності веб-додатків.

- **Зв'язування DOM з об'єктними даними за допомогою HTML-шаблонів.** Фреймворк Vue надає HTML-шаблони для зв'язування DOM із даними об'єктів. Запускає компіляцію шаблонів у вигляді HTML, сумісного з усіма браузерами.
- **Дозволяє користувачам швидше переміщатися сайтом.** Маршрутизатор Vue перемикається між сторінками без частого оновлення веб-сторінок і прискорює навігацію для користувачів.
- **Надають директиви для інтеграції коду.** Директиви – це набір інструкцій для інтеграції коду екземплярів Vue. Ці візуальні засоби забезпечують інтерактивність вашої програми для покращення користувацького досвіду.

Angular (<https://angular.io/>)

Angular є фреймворком з відкритим вихідним кодом, написаним мовою Typescript. Це фреймворк, що базується на інкрементах, який пропонує багато інтегрованих бібліотек та інструментів для створення, тестування та підтримки коду веб-додатків. Є надійним варіантом для веб-розробників, що дозволяє створювати інтерактивні програми за допомогою коду JavaScript.



```
<script type="text/javascript">
var myApp = angular.module('myApp', []);

myApp.directive('fileModel', ['$parse', function ($parse) {
    return {
        restrict: 'A',
        link: function(scope, element, attrs) {
            var model = $parse(attrs.fileModel);
            var modelSetter = model.assign;

            element.bind('change', function(){
                scope.$apply(function(){
                    modelSetter(scope, element[0].files[0]);
                });
            });
        }
    };
}]);

myApp.service('fileUpload', ['$http', function ($http) {
    this.uploadFileToUrl = function(file, uploadUrl){
        var fd = new FormData();
        fd.append('file', file);
        $http.post(uploadUrl, fd, {
            transformRequest: angular.identity,
            headers: {'Content-Type': undefined }
        })
        .success(function(){
        })
        .error(function(){
        });
    }
}]);

myApp.controller('myCtrl', ['$scope', 'fileUpload', function($scope, fileUpload){
    $scope.uploadFile = function(){
        var file = $scope.myFile;
        console.log('file is ' );
        console.dir(file);
        var uploadUrl = 'http://localhost:kites/upload/';
        fileUpload.uploadFileToUrl(file, uploadUrl);
    };
}]);
```

Рис.4. Інтерфейс фреймворку Angular

В Angular інтегровано інтерактивні шаблони та інструменти наскрізної інтеграції для пом'якшення проблем у веб-проектах. Цей фреймворк можна використовувати для створення одноразових веб-застосунків. У зв'язку з великою популярністю цього фреймворку існує багато версій Angular. Цей фреймворк є найкращим варіантом для розробки програм корпоративного рівня шляхом інтеграції складних функцій. Часто люди плутають Angular з AngularJS (<https://angularjs.org/>), але між цими двома фреймворками є різниця. Angular сумісний з усіма мобільними браузерами, тоді як версія JS підтримує лише настільні браузери.

Особливості Angular

Підтримка прив'язки даних у двох вимірах. Фреймворк Angular підтримує двостороннє зв'язування даних для представлення шару моделі. Має попередній перегляд для відстеження змін у моделі. Таким чином, користувачі можуть автоматично переглядати зміни в моделі, скорочуючи час на розробку програми.

Забезпечення взаємозалежності. Дозволяє класам, методам, функціям та модулям працювати із взаємозалежними модулями та підтримувати узгодженість коду, скорочуючи зміни класів.

Забезпечення архітектури Model View Controller (MVC). Структура цього фреймворку відокремлює код програми від інтерфейсу користувача, що допомагає веб-розробникам економити час на коді.

Ember (<https://emberjs.com/>)

Фреймворк Ember широко використовується для створення програм, сумісних з настільними та мобільними браузерами, та односторінкових веб-додатків. Використовує шаблон HTMLBars, який автоматично оновлює зміни даних.



```
config/environment.js
1  'use strict';
2
3  module.exports = function(environment) {
4    let ENV = {
5      modulePrefix: 'super-rentals',
6      environment,
7      rootURL: '/',
8      locationType: 'auto',
9      EmberENV: {
10       FEATURES: {
11         // Here you can enable experimental features on an ember canary build
12         // e.g. EMBER_NATIVE_DECORATOR_SUPPORT: true
13       },
14       EXTEND_PROTOTYPES: {
15         // Prevent Ember Data from overriding Date.parse.
16         Date: false
17       }
18     },
19
20     APP: {
21       // Here you can pass flags/options to your application instance
22       // when it is created
23     }
24   };
25 }
```

Рис.5. Інтерфейс фреймворку Ember

Особливості Ember

- **Надання мови шаблонів.** Синтаксис Handlebar відноситься до мови шаблонів. Використовує шаблон та вхідні дані для створення HTML або інших форматів даних. Ці шаблони схожі на звичайний текст, у якому використовується синтаксис рукоятки із фігурними дужками.
- **Надає інструмент перевірки.** Надає інструмент інспекції для перевірки коду Ember, моніторингу продуктивності додатків, що використовують Ember, та швидше налагодження помилок у додатках.

- **Забезпечує автентифікацію.** Пропонує шаблони безпеки для автентифікації та авторизації програм. Надає абстрактні шаблони та інтегрується з іншими варіантами безпеки для суворіших протоколів.
- **Ініціалізатори додатків.** Пропонує функцію ініціалізаторів програм для запуску та налаштування ін'єкції залежностей програми.

Backbone (<https://backbonejs.org/>)

Backbone – мінімалістський JavaScript-фреймворк, що дозволяє проектувати клієнтські програми, сумісні зі всіма веб-браузерами. Пропонує модулі MVC для абстракції даних і об'єднує ці компоненти для створення елегантних інтерфейсів (UI) шляхом написання кількох рядків коду.



```

9 <div class="container">
10 <div>User Manager</div>
11 </div>
12 <div class="page"></div>
13 </div>
14
15 <script src="//cdnjs.cloudflare.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
16 <script src="//cdnjs.cloudflare.com/ajax/libs/underscore.js/1.4.2/underscore-min.js"></script>
17 <script src="//cdnjs.cloudflare.com/ajax/libs/backbone.js/0.9.2/backbone-min.js"></script>
18 </script>
19
20 $.ajaxPrefilter(function( options, originalOptions, jqXHR ) {
21   options.url = 'http://backbonejs-beginner.herokuapp.com' + options.url;
22 });
23
24 var Users = Backbone.Collection.extend({
25   url: '/users'
26 });
27
28 var UserList = Backbone.View.extend({
29   el: 'page',
30   render: function () {
31     var users = new Users();
32     users.fetch({
33       success: function () {
34         this.$el.html('CONTENT SHOULD SHOW HERE!');
35       }
36     });
37   }
38 });
39
40
41 var Router = Backbone.Router.extend({
42   routes: {
43     '': 'home'
44   }
45 });
46
47 var userList = new UserList();
48
49 var router = new Router();
50 router.on('route:home', function () {
51   userList.render();
52 });

```

Рис.6. Інтерфейс фреймворку Backbone

Використовує імперативне програмування для створення бажаних веб-додатків, тоді як інші фреймворки пропонують декларативний стиль розробки.

Особливості Backbone

- **Використання методів та функцій JavaScript.** Використовує функції та модулі JavaScript як будівельні блоки коду JavaScript і забезпечує прив'язку модулів та користувацьких входжень.
- **Безкоштовний фреймворк з відкритим кодом.** Пропонує понад сотню бібліотек для створення програм.
- **Підтримка крос-платформної розробки додатків.** Фреймворк Backbone допомагають веб-розробникам створювати програми, сумісні з усіма пристроями та браузерами.

NextJS (<https://nextjs.org/>)

NextJS - це безкоштовний фреймворк з відкритим вихідним кодом для створення статичних веб-додатків за допомогою React. Забезпечує кращий досвід користувача, покращену продуктивність і більш швидку розробку веб-додатків. Сьогодні цей фреймворк знаходиться на висоті завдяки сильній підтримці спільноти. Допомагає розробляти програми на основі існуючих шаблонів, а не писати код із нуля.

NextJS - найкращий фреймворк для бізнесу завдяки повному контролю над продуктами, такими як веб-сайти або додатки. Використовуючи цей фреймворк, вам не потрібно покладатися на плагіни, як у інших фреймворках. Більшість розробників вважають за краще повторно використовувати компоненти React, оскільки це скорочує час та вартість розробки.



```
axios({
  method : 'GET',
  url : `http://localhost:8000/account/${decoded_token.user_id}`,
  headers : {'Authorization' : 'Bearer'+ ' ' + jwt_token}
})
.then((response) => {
  response.data.role !== 'client' ? router.push('/login') : ''
  axios({
    method : 'GET',
    url : `http://localhost:8000/profile/${decoded_token.user_id}`,
    headers : {'Authorization' : 'Bearer'+ ' ' + jwt_token}
  })
  .then((response) => {
    console.log(response.data)
  })
  .catch((error) => {
    console.log(error.response)
  })
})
})
```

Рис.7. Інтерфейс фреймворку NextJS

Висновки

В останні роки розробка додатків отримала новий напрямок, такий як no-code. No-code - це перспективний напрямок, що дозволяє розробникам відмовитися від рутинних завдань та зайнятися складнішими та цікавішими речами, наприклад, проектуванням архітектури.

JavaScript-фреймворки - це розширення розробки без коду, яке пропонує більш швидкі та дешеві рішення. JavaScript - це популярна мова сценаріїв для веб-розробки. Ця мова дозволяє вести розробку з використанням бібліотек та фреймворків. Рекомендуємо використовувати будь-який фреймворк на ваш вибір, щоб скоротити час і вартість розробки веб-додатків.

Джерела інформації

1. JavaScript фреймворки

<https://appmaster.io/ru/blog/luchshie-javascript-freimvorki-dlia-ispol-zovaniia-v-2022-godu-polnoe-rukovodstvo>