Зборник радова Факултета техничких наука, НовиСад

УДК: 4.9
ДОИ: https://doi.org/10.24867/34BE05Djuric

УПРАВЉАЊЕ ДОГАЂАЈИМА У ВЈУ.ЏЕЈ-ЕС И АНГУЛАР ФРОНТЕНД ПРОГРАМСКИМ ОКВИРИМА

EVENT HANDLING IN VUE.JS AND ANGULAR FRONTEND FRAMEWORKS

Кристина Ђурић, *Факултет техничких наука, Нови Сад*

Студијски програм – РАЧУНАРСТВО И АУТОМАТИКА

Кратак садржај – *У овом раду је дат опис, као и поређење управљања догађајима помоћу Ангулар и Вју.џеј-ес фронтенд програмских оквира. Догађаји у овом контексту представљају акције или промене стања у апликацији (нпр. клик мишем, унос текста, учитавање података) на које систем може да реагује. На основу истраживања како сваки од оквира управља догађајима и компарација приступа имплементацији истих функционалности у сваком од њих, изведени су закључци о предностима и манама сваке од платформи, као и о томе колико је сваки од њих погодан за коришћење.*
Кључне речи: *догађај, Вју.џеј-ес, Ангулар*

Abstract – *This paper describes and compares event management using Angular and Vue.js frontend frameworks. Based on the research of how each of the frameworks manages events and the comparison of approaches to the implementation of the same functionalities in each of them. Conclusions were drawn about the advantages and disadvantages of each of the platforms, as well as how suitable each of them is for use.*

Keywords:: *event, Vue.js, Angular*

НАПОМЕНА: Овај рад проистекао је из мастер рада чији ментор је био др Дину Драган, ванр. проф.

1. УВОД

Основна карактеристика модерних веб апликација је њихова способност да обезбеде богат и реактиван приступ кориснику. Срж интерактивности је комплексно управљање догађајима, које подразумева како ће нека апликација реаговати на акције корисника. Ефикасност, скалабилност и једноставност развоја апликације у великој мери зависе од њеног приступа управљању догађајима [1].

С обзиром на развој и популарност савремених ЈавaСкрипт (енгл. JavaScript) фронтенд програмских оквира који су тема рада, избор одговарајуће технологије је кључан корак у планирању архитектуре апликације. Због тога је упоредна анализа њихових основних механизама, као што је управљање догађајима, од суштинског значаја за доношење праве одлуке. Иако су Вју.џеј-ес (енгл. Vue.js) и Ангулар (енгл. Angular) водећи оквири за развој динамичких корисничких интерфејса, они примењују различите механизме у управљању догађајима, што захтева детаљно истраживање.

Вју.џеј-ес је лаган и прогресиван ЈавaСкрипт фронтенд програмски оквир. Специфичности које су наглашене су једноставност, флексибилност и лакоћа усвајања. Ангулар, који је развио и одржава Гугл (енгл. Google), заснован је на ТајпСкрипту (енгл. TypeScript). Специфичности су јача типизација, модуларна архитектура, омогућава бољу организацију кода, одржавање и скалабилност апликације.

Развој веба је значајно еволуирао од класичних ЈавaСкрипт слушалаца догађаја. Данас савремени фронтенд програмски оквири нуде нове механизме који поједностављују управљање догађајима, укључујући директиве, модификаторе и обрасце комуникације између компоненти. Иако ови системи повећавају продуктивност, они уносе нове слојеве комплексности. Како у перформансама, тако и у архитектонском дизајну и искуству програмера. Постојећи извори обично пореде ове оквире уопштено, као што су перформансе, скалабилност или крива учења. Међутим, техничка анализа конкретно усмерена на механизме управљања догађајима не постоји. Овај рад има за циљ да попуни ту празнину анализом принципа и техника који постоје за управљања догађајима у фронтенд програмским оквирима Вју.џеј-ес и Ангулар.

Рад је организован на следећи начин. У другом поглављу дат је кратак осврт на појам догађаја и њихов значај у савременим фронтенд оквирима, уз објашњење основних термина. У трећем поглављу је описано на које све начине је могуће управљање догађајима у Вју.џеј-есу. У четвртом поглављу је описано на које све начине је могуће управљати догађајима у Ангулару. Пето поглавље доноси компаративну анализу ова два фронтенд програмска оквира са практичним примерима, уз сажетак предности, мана и сложености. Шесто поглавље је закључак рада.

2. УПРАВЉАЊЕ ДОГАЂАЈИМА

Догађај у веб апликацијама представља било коју интеракцију са HTML елементом – клик на дугме или линк, унос текста у поље, притисак тастера на тастатури, као и имплицитне радње попут завршетка

учитавања странице или промене величине прозора прегледача. Ови догађаји чине основу динамике и интерактивности, јер покрећу извршавање одређених функција путем обрађивача догађаја (енгл. event handler). Начин на који се управља догађајима варира у зависности од технологије која се користи. Најнижи ниво представља ЈавaСкрипт, који дефинише основне механизме рада са DOM-ом и омогућава регистровање и обраду догађаја. Зато ЈавaСкрипт чини основу на којој су изграђени фронтенд оквири Ангулар и Вју.џеј-ес. DOM је стандардизован од стране W3C/WHATWG и описује структуру HTML документа као стабло објеката. Сам DOM није део самог језика, него API који веб прегледач приказује. Фронтенд оквири додају апстракцију и лакшу синтаксу. Сама логика обраде догађаја се заснива на ЈавaСкрипту и DOM API-ју, што значи да се ниједан фронтенд оквир не удаљава потпуно од основног модела, већ практично га прилагођава својој парадигми.

2.1. Кључни појмови за лакше разумевање управљања догађајима у Вју.џеј-есу и Ангулару

ЈавaСкрипт је скрипт програмски језик који служи за изградњу динамичких и функционалних веб решења заједно са HTML-ом и CSS-ом. Пошто је реч о клијент-скрипт језику, то значи да се овај програмски језик извршава, тачније интерпретира, са стране самог веб прегледача на корисничком уређају.
ТајпСкрипт је објектно орјентисани програмски језик високог нивоа који је развио Мајкрософт (енгл. Microsoft) и он проширује ЈавaСкрипт додавањем типова. ТајпСкрипт представља надскуп ЈавaСкрипта.
Фронтенд програмски оквири је унапред написана колекција стандардизованог HTML-а, CSS-а (ЦСС – каскадни стилови) и ЈавaСкрипт кода који програмери могу да користе за ефикасније прављење веб апликација [2]. Они обухватају скуп алата, библиотека и компоненти које програмерима олакшавају имплементирање корисничког интерфејса, а то укључује дизајн, изглед, корисничко искуство, управљање догађајима, управљање стањима апликације и интеракцију са АПИ-јима. Уз помоћ фронтенд програмских оквира може се избећи писање свега од нуле, убрзавају процес развоја.

3. УПРАВЉАЊЕ ДОГАЂАЈИМА УПОТРЕБОМ ВЈУ.ЏЕЈ-ЕСА

Вју.џеј-ес представља ЈавaСкрипт фронтенд програмски оквир који омогућава лакше креирање интерактивних и реактивних веб страница. Његов творац је Еван Ју (енгл. Evan You). Према спроведеном истраживању, Вју.џеј-ес се убраја међу једноставније ЈавaСкрипт оквире. Неке од његових карактеристика су брзина, једноставан је за употребу, има веома детаљну документацију и велику заједницу.

3.1. Вју.џеј-ес директиве

Вју.џеј-ес за руковање догађајима користи директиве које се каче на DOM елементе. Главна је v-on (скраћено @) и њом се повезује догађај са обрађивачем. Обрађивач може бити инлајн, метода у компоненти или метода са прослеђеним аргументима.

3.2. Вју.џеј-ес модификатори

Модификатори догађаја проширују могућности руковања догађајима у Вју.џеј-есу, додају додатни ниво контроле. Функционишу тако што се додају постфикси као што су .stop, .prevent, .capture, .self, .once, .passive на директиве. Омогућавају прецизно и читљиво управљање понашањем без додатног кода у методама.

3.3. Прилагођени догађаји (енгл. Custom Events)

Прилагођени догађаји пружају посебан начин за руковање комуникацијом међу компонентама. Вју.џеј-ес омогућава употребу пропса за пренос података од родитеља ка детету, али за комуникацију у супротном смеру се користи емитовање догађаја, односно прилагођени догађаји. Компонента дете као одговор родитељској компоненти емитује догађај употребом $emit методе. Родитељ ослушкује те догађаје у темплејту и позива своје методе.

3.4. Управљање стањем у Вју.џеј-есу употребом Вуекса

Вјуекс је библиотека за управљање стањем у Вју.џеј-есу са централним стором (енгл. store) који је реактиван. Стање (енгл. state) стора се мења искључиво преко мутација (енгл. mutations). Мутације не садрже пословну логику, већ служе искључиво за измену стања. Акције (енгл. actions) су те које служе за које имплементирају логички део и затим позивају мутације. Мутације је могуће и директно комитовати. Гетери (енгл. getters) изводе податке из стања, а модули (енгл. modules) организују стор на мање целине.

4. УПРАВЉАЊЕ ДОГАЂАЈИМА УПОТРЕБОМ АНГУЛАРА

Ангулар је фронтенд оквир заснован на HTML-у и ТајпСкрипту за креирање интерактивних, реактивних веб апликација, са нагласком на јасну архитектуру и добру документацију. Настао је као AngularJS (2010), а потпуно је ревидиран у Angular 2 (2016) и од тада се развија као засебан, типизован оквир. Истраживање управљања догађајима у Ангулару је спроведено на темељима његове архитектуре коју чине компоненте, модули, шаблони, директиве, сервиси и Dependency Injection (DI - инјектовање зависности). Компоненте су основни градивни блокови (класе са логиком и руковаоцима догађаја повезане са HTML шаблоном), модули логички групишу функционалности, директиве мењају структуру или понашање DOM-а, док сервиси инкапсулирају пословну логику и убризгавају је путем DI ради боље организације, тестабилности и поновне употребе.

4.1. Везивање догађа (енгл. Event Binding)

Везивање догађаја омогућава ослушкивање и реаговање на догађаје стављањем имена догађаја у заграде у HTML шаблону и повезивањем са методом компоненте (нпр. *(click)="onClick($event)"*). Метод дефинише радњу која се извршава када се догађај догоди.

4.2. Прилагођени догађаји

Прилагођени догађаји у Ангулару омогућавају комуникацију између компоненти. Емитују се из компоненте дете ка родитељској компоненти помоћу EventEmitter класе и @Output() декоратора, а слање догађаја се врши преко методе EventEmitter.emit().

4.3. Управљање стањем у Вју.џеј-есу употребом ЕнЏиАрИкса (енгл. NgRx)

У Ангулару се стање може чувати локално у компонентама, што је једноставно, али се губи при навигацији и није погодно за сложеније апликације. За предвидљиво и централизовано управљање препоручује се употреба ЕнЏиАрИкса. То је библиотека која уводи стор (енгл. store) као централни репозиторијум, док се промене стања врше искључиво преко акција које обрађују редуктори (енгл. reducer), а приступ подацима преко селектора (енгл. selector). Овим се добија структуриран, контролисан и трајнији модел управљања стањем током рада апликације.

5. КОМПАРАТИВНА АНАЛИЗА

У овој секцији је кроз примере урађена компаративна анализа управљања догађајима у фронтенд програмским оквирима Вју.џеј-ес и Ангулар. Приказане су методе и механизми управљања догађајима, њихове карактеристике у погледу лакоће имплементације, читљивости и погодности за различите типове апликација.

5.1. Везивање догађаја

Вју.џеј-ес користи v-on директиву, скраћено @, што је једноставније и прегледније. Ангулар користи (event) синтаксу у темплејту која позива методу у класи, формално и дужег облика. Вју.џеј-ес пружа већу једноставност, Ангулар више формалности и конзистентности.

5.2. Модификатори догађаја

Вју.џеј-ес има уграђене модификаторе (.prevent, .stop, .once, .self, .right) који омогућавају чисту логику, без додатног кода у методама. Ангулар нема уграђене модификаторе, па је потребно ручно управљати DOM догађајима у методама и то захтева више знања о DOM API-ју, уместо да методе садрже само чисту логику. Модификатори у Вју се дефинишу заједно уз догађај, директно у шаблону, што омогућава јаснију, краћи код и лакше разумевање. Овај приступ је интуитивнији за почетнике и поједностављује комбиновање више услова. У Ангулару се иста логика мора имплементирати у методама због чега недостаје директна повезаност са самим догађајем у темплејту.

5.3. Прилагођени догађаји и комуникација родитељ–дете компоненти

У Вју компоненти дете на слици 1, унутар *methods* дефинисане су три методе које позивом *$emit* емитују догађај са подацима. Догађај *this.$emit("archive-fiesta", this.fiesta.id)* родитељ хвата преко *@archive-fiesta="archiveFiesta"*.

```
<script>
export default {
  name: "FiestaCard",
  props: {
    fiesta: Object,
  },
  methods: {
    editFiesta() {
      this.$emit("edit-fiesta", this.fiesta);
    },
    archiveFiesta() {
      this.$emit("archive-fiesta", this.fiesta.id);
    },
    deleteFiesta() {
      this.$emit("delete-fiesta", this.fiesta.id);
    },
  },
};
</script>
```

Слика 1. Компонента дете у Вју.џеј-есу

У Ангулару компонента дете емитује догађај преко декоратора *@Output()* и класе *EventEmitter<T>*. На слици 2 је приказана имплементација догађаја *@Output() archiveFiesta = new EventEmitter<any>()*. Родитељ догађај хвата преко *(archiveFiesta)= "onArchive($event)"*.

```
import { Component, EventEmitter, Input, Output } from '@angular/core';

@Component({
  selector: 'fiesta-card',
  standalone: true,
  imports: [],
  templateUrl: './fiesta-card.component.html',
  styleUrl: './fiesta-card.component.css'
})
export class FiestaCardComponent {
  @Input() fiesta!: { title:string; date: Date; venue: string; description: string };

  @Output() archiveFiesta = new EventEmitter<any>();
  @Output() deleteFiesta = new EventEmitter<any>();
  @Output() editFiesta = new EventEmitter<{ id: number, content: string }>();

  onArchive(id: number) {
    this.archiveFiesta.emit(id);
  }

  onDelete(id: number) {
    this.deleteFiesta.emit(id);
  }

  onEdit(id: number, content: string) {
    this.editFiesta.emit({ id, content });
  }
}
```

Слика 2. Компонента дете у Ангулару

Иако Вју има једноставнији и бржи начин емитовања догађаја, Ангуларов приступ са EventEmitter<T> је професионалнији и поузданији у већим системима. Разлог је јака типизација и боље контроле података. У Вју то није подразумевано понашање, већ је потребно додатно подешавање.

5.4. Управљање стањем

У већим системима прилагођени догађаји брзо постају непрегледни, па је бољи приступ да се подаци централизовано чувају и ажурирају у оквиру целе апликације. У Вју.џеј-есу то обезбеђује Вјуекс, а у Ангулару ЕнЏиАрИкс. Тригеровање догађаја у шаблону остаје исто, за Вју @click, а за Ангулар (click). Компоненте само диспечују акције (енгл. actions), док се стварне измене стања изводе у мутацијама (енгл. mutations) код Вјуекса и редукторима (енгл. reducers) код ЕнЏиАрИкса. Податке компоненте читају преко гетера (енгл. getters) у Вјуексу, односно селектора (selectors) у ЕнЏиАрИксу. Оваква организација пребацује логику измена из компоненти у централизовани стор (енгл. store), поједностављује компоненте и јасно раздваја акције од измена (мутације/редуктори). Резултат је предвидљив,

тестабилан и скалабилан ток промена, са мање логике у компонентама и јаснијим током података кроз апликацију.
Вјуекс је лакши за интеграцију у мањим и средњим апликацијама, али је проблем валидација типова, што може бити ограничење у већим системима. ЕнЏиАрИкс је постављен сложеније, али систем је стабилан, скалабилан и једноставан за тестирање, што га чини погоднијим за велике пројекте.

5.5. Општа оцена

У Вју је код краћи, једноставнији и визуелно јаснији јер је већина логике ближа темплејту, а то значи да су логика и изглед компоненте повезани, налазе се у истом фајлу. Ово омогућава брже разумевање и лакше учење, али може бити и проблем на дуже стазе када апликације нарасте. Зато се за Вју често каже да је погодан за мање апликације.
Ангулар код је обимнији и садржи више структура (класе, декораторе, типове), али зато пружа бољу организацију и лакше тестирање на нивоу система. Велика предност Ангулара је типизација података и то му је главна предност спрам Вју.
Избор између ова два фронтенд програмска оквира зависи од потреба конкретне апликације, Вју је идеалан када је приоритет једноставност и брзина, док је Ангулар погоднији за велике системе са строгим захтевима.

6. ЗАКЉУЧАК

Догађаји представљају суштински део сваке интерактивне веб апликације, омогућавајући корисницима интеракцију, комуникацију и динамичко искуство. У овом раду су анализирани програмски оквири Вју.џеј-ес и Ангулар који пружају различите начине за руковање догађајима. Акценат рада је на компарацији имплементације истих функција везаних за управљање догађајима, као и на анализи предности и недостатака оба програмска оквира у различитим контекстима употребе.
Вју.џеј-есова синтакса је једноставнија и интуитивнија, приликом читања кода лако је формирати реченице које објашњавају како се управља неким догађајем и шта он ради. Цео ток је јасно исказан у једном реду, што значајно олакшава читање и одржавање кода, као што су на пример уграђени модификатори догађаја. Сам еханизам управљања догађајима у Вју је флексибилан и брз за имплементацију у мањим и средње сложеним апликацијама. Међутим, за сваки догађај потребно је имплементирати неку комплекснију логику и ту све постаје компликованије, бар кад су у питању већи тимови и пројекти који немају стриктно дефинисана правила организације кода.
Са друге стране, Ангулар је погоднији за апликације које имају велики број интеракција, сложених догађаја и захтевну динамику корисничког интерфејса, због своје структуре и контроле над логиком апликације, као и због боље организације кода. Иако је тежи почетницима за учења и разумевање, искуснији програмери ће га лако савладати. Још неке од предности су ТајпСкрипт, што значи да се јасно дефинише тип података, подстиче архитектуру по шаблону и има ефикасан алат за рад преко командне линије.
При избору између ових оквира, важно је размотрити специфичне захтеве пројекта, величину и искуство тима, као и дугорочне циљеве одржавања и развоја [3]. На основу личног искуства стеченог радом на академским пројектима са Вју.џеј-есом и на комерцијалном пројекту са Ангуларом, може се потврдити да је Вју.џеј-ес погоднији за мање апликације и брзу имплементацију, а Ангулар за веће и сложеније апликације.
Као будући правац истраживања може се предложити проширење компаративне анализе и на друге популарне фронтенд програмске оквире као што су Риект (енгл. React), који користи модел синтетичких догађаја (енгл. Synthetic Events), као и Ванила ЈавaСкрипт (енгл. Vanilla JavaScript). На тај начин могла би се спровести анализа како сваки од њих управља догађајима и на који начин приступају DOM-у, као и перформансе свакога од њих. Када је реч о Вју.џеј-есу, била би занимљива компарација Вјуекса и Пиније (енгл. Pinia), истражити да ли Пинија заиста доноси побољшања у односу на Вјуекс.

7. ЛИТЕРАТУРА

[1] https://www.geeksforgeeks.org/blogs/introduction-to-scripting-languages/ (приступљено у септембру 2025).

[2] https://www.geeksforgeeks.org/blogs/top-front-end-frameworks/ (приступљено у септембру 2024).

[3] https://www.capitalnumbers.com/blog/state-management-front-end-development (приступљено у октобру 2024).

Кратка биографија:

Кристина Ђурић рођена је у Лозници 1998. год. Мастер рад на Факултету техничких наука из области Електротехнике и рачунарства – Рачунарство и аутоматика одбранила је 2025.год.
Контакт:
kristinadjuriic@gmail.com