# VUE

Ena izmed dobrih alternativ Angularju je **Vue.js**, ki je odprtokoden framework za grajenje uporabniških vmesnikov in Single page aplikacij.

Podatki:

* **Avtor**: Evan You
* **Izdan**: Februar 2014
* **Jezik**: JavaScript

V primerjavi z Angularju je predvsem manjši in bolj fleksibilen ter se ga je lažje naučiti.

Vue vsebuje izjemno prilagodljivo arhitekturo ki se osredotoča na dinamično procesiranje in kompozicij komponent, podobno kot pri Angularju. Za podporo poti, upravljanja s stanjem in ostalih naprednih zadev, ima Vue več podpornih knjižnic in paketov, eden izmed najbolj popularnih je trenutno Nuxt.js. Vue.js omogoča razširitev HTML z direktivi. Ti so lahko vgrajeni, ali pa jih dodamo kar sami.

## Komponente

Vue razširja osnovne HTML elemente z uporabo komponent. Ti so poljubni elemetni, katerim Vue prevajalnik doda obnašanje (behaviour). V Vue je komponenta v bistvu instanca z že v naprej določenimi možnostmi.

## Predloge

Tako kot Angular, Vue uporablja povezavo med DOM in podatki Vue. Vse predloge so validni HTML, katere lahko prebavi vsak HTML parser. Vue predloge spremeni v virtualne DOM funkcije. Nato virtualni DOM objekt omogoči upodabljanje komponent v spominu še preden se te prikažejo na brskalniku. Kombinirano z odzivnostjo, Vue lahko izračuna minimalno številno komponent katere je potrebno ponovno upodobiti ter tudi minimalno število DOM manipulacij.

## Odzivnost

Vue odzivni sistem uporablja čisti JavaScript. Vsaka komponenta vsebuje seznam svojih odzivnih odvistnosti med upodabljanjem, zaradi tega sistem ve natančno kdaj ponovno upodobiti in s tem tudi katere komponente.

## Poti

Tako kot Angular, Vue ponuja vmesnik, ki se spreminja glede na podan URL, ne glede na to kako je bil ta spremenjen (email link, osvežitev…). Uporaba čelnega dela »routerja« omogoča tranzicijo ob dogodkih na brskalniku (kliki na gumbe, povezave). Vue ne vsebuje hashiranega čelnega routinga.

Ko smo primerjali Vue in Angular, nam razlika ni bila tako gromozanska, saj sta bolj kot ne skoraj ista JS ogrodja, Vue je le malce manjši in okretnejši. Za Angular smo se odločili, ker nam je bližje in smo z njim že imeli opravka. Prav tako je bolj kompleksen in vsebuje mnogo več knjižnic, katere je moč uporabiti pri samem projektu. Seznanjeni smo tudi z Angular material, knjižnico, ki vsebuje UI elemente za izdelavo izgleda spletne aplikacije.

# REACT

React ali ReactJS je JavaScript knjižnica za grajenje uporabniških vmesnikov. Trenutno ga vdržuje Facebook in skupina neodvisnih razvijalcev in podjetij.

React lahko uporabimo kot osnovo za SPA in mobilne aplikacije. Kakorkoli, React je namenjen le upodabljanju podatkov DOM-u, zato ustvarjanje React aplikacij potrebuje uporabo dodatnih knjižnic za vdrževanje stanja in poti. Primera teh sta **Redux** in **React Router.**

Podatki:

* Avtor: Jordan Walke
* Razvijalci: Facebook in skupnost
* Izdan: Maj 2013
* Jezik: JavaScript

## Osnovna uporaba



Videti je, da je poudarek na upodabljanju elementov DOM.

## Komponente

Kot pri Vue in Angular, React prav tako uporablja komponente ali entitete. Te lahko dodamo poljubnemu HTML objektu. Ko upodabljamo komponento, ta poleg pošlje še podatke ki so znani kot »props«.

## Funkcionalne komponente

Vračajo JSX (JavaScriptXML).

## Razredne komponente

Znane tudi kot izjavne komponente. Njihovo stanje lahko obdrži vrednosti vseskozi komponente in jih lahko pošiljamo otroškim komponentam preko »propsov«.

## Virtualni DOM (Document Object Model)

React v pomnilniku ustvari cache podatkovne strukture, izračuna razlike in nato upodobi prikaz na brskalniku. Ta proces je spravo ali reconciliation. To omogoči programerju, da programira v smislu kot da je stran osvežena ob vsaki spremembi, medtem ko React osveži le tiste komponente, katere se dejansko spremenijo. To omogoči gromozanski skok v zmogljivosti. Zaradi tega ni potrebno ponovno preračunavati CSS stila, postavitve strani in upodabljati celotne strani.

## Življenski cikel

Pri angularju poznamo več metod, ki predstavljajo cikel komponente (NgOnInit, NgOnDestroy..), React pa ima malce drugačne. Med temi so shouldComponentUpdate, componentDidMount, componentWillUnmount… Ena izmed najpomembnejših pa je render.

# AURELIA

Aurelia je eden novejših ogrodij za izdelavo spletnih aplikacij (izdan je bil leta 2015 in je trenutno v prvi verziji). Lahko bi tudi rekli, da je naslednja generacija le teh. Aurelia ponuja svež in vznemerljiv pristop razvoju čelnega dela aplikacij, ki je močnejši alternativam. Sama sintaksa je zelo podobna tisti, ki jo ima Angular.

## Priročnost > konfiguracija

Cilj Aurelie je, da prioritizira priročnost tako, da ima manjši vpliv na razvoj. Ta način zmanjša število odločitev, ki jih morajo razvijalci narediti, brez da izgubijo fleksibilnost.

## Moduli

Tako kot Angular je tudi Aurelia skupek JS modulov. Grajenje aplikacij je prav tako podobno Angularju; kompozicija preprostih komponent s standardnim JS jezikom in TypeScript razredi skupno z HTML predlogami. Moduli omogočajo slednje: Binding, Dependency Injection, Metadata, Routing ter Templating 🡪 skoraj identično Angularju. Te module je moč uporabiti individualno v vsakem JS projektu, vključno z Node.js.

## Standardi odprtega spleta

Aurelia uporablja DOM standard, ki je neodvisen od platforme in jezika. Prav tako uporablja lastni HTML parser in JS razširitve ter ob tem ohranja DOM API za optimizacijo delovanja.

Aurelia temelji na W3C, kar bo aplikacijam, ki bodo temeljile na Aurelii, omogočalo neoviran razvoj.

## Binding ali povezovanje

Aurelia z lahkoto podpira enosmerno in dvosmerno povezovanje, medtem ko je to malce težje v primeru Angularja in lahko privede do zmede med razvijalci. Pri enosmernem povezovanju podatki tečejo v eno smer, iz objektov v UI, pri dvosmernem pa v obe smeri, se pravi izmenjava med UI in objekti, kar omogoča sinhronizacijo.

Aurelia v osnovi uporablja enosmerno povezavo.

## Popularnost

V primerjavi je Angular v večih primerih bolj popularen kot Aurelia, verjetno zato, ker še ne obstaja veliko uspešnih aplikacij, ki bi temeljije na Aurelii.

## Jezikovna podpora

Angular zahteva, da razvijalci pišejo strogo JS v TypeScriptu. Aurelia razvijalci lahko prav tako uporabljajo TypeScript vendar lahko uporabijo tudi ES ali EcmaScript, kar razvijalcem omogoči večjo fleksibilnost.

## Routing

Oba ogrodja se soočata z routing konfiguracijo na podoben način, vendar je definicija poti malce različna. V Aurelii, razvijalci definirajo starševske poti na enem mestu, otroke pa v posameznih komponentah. Ta pristop popolnoma enkapsulira komponente ter skrije njihovo notranjo komplekstnost. Za primerjavo, v Angularju je potrebno definirati poti modula v centralizirani konfiguracijski datoteki, vključno s potmi, komponentami in otroki poti. To naredi definicijo poti kompleksnejšo v Angularju

# Zakaj Angular in ne njegove alternative?

Aurelia je eden novejših ogrodij in ima podobno osnovo kot Angular, vendar odpravi nekaj kompleksnosti le tega. Vue je prav tako podoben Angularju, vendar v osnovi ne vsebuje toliko možnosti kot Angular. React JS je dandanes eden najhitrejših ogrodij kar se tiče spletnih aplikacij, vendar ne ustreza naši ciljni aplikaciji, predvsem zaradi omejenosti na UI. Zdelo se nam je, da bi naši aplikaciji bolj ustrezal Angular, predvsem zaradi predznanja o njemu in zaradi njegove zmogljivosti (preprosta implementacija Google maps, storitve, izgled).