# 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