

# React va Redux haqida

## Reakt nima?

Kichik ta'rifda React - bu foydalanuvchi interfeyslarini yaratish uchun deklarativ, samarali va moslashuvchan JavaScript kutubxonasi. Uni Facebook va individual ishlab chiquvchilar va kompaniyalar jamoasi qo'llab-quvvatlaydi.

Hozirda bu poydevor va uni qo'llab-quvvatlovchi katta jamoatchilikka ega bo'lgan eng mashhur JavaScript-ni kutubxonalaridan biri.

Reakt bir sahifali dasturlarni yoki mobil ilovalarni ishlab chiqishda asos sifatida ishlatilishi mumkin. Complex React dasturlari odatda davlatni boshqarish, marshrutlash va API bilan ishlash uchun qo'shimcha kutubxonalardan foydalanishni talab qiladi.

*ReactJS - bu faqat **MVC**- ning **View** komponenti (Model - **View** - Controller) bilan shug'ullanadigan frontend kutubxonasi .*

## Nima uchun ReactJs-dan foydalanish kerak?

Barchamiz bilamizki, burchakli, magistral, Ember va boshqa ko'plab Javascript ramkalari bizning dasturimiz ko'rinishini boshqarish uchun sanoat sohasida allaqachon mavjud edi, keyin nima uchun reaksiya qilish kerak?

Javob:

- **Oddiy:** ReactJS go'zalligi uning soddaligi. Bu aniq belgilangan hayot aylanishiga ega komponentlarga asoslangan til. React dasturida ishlab chiquvchilar HTML-ni JavaScript bilan aralashtirishga imkon beruvchi JSX-dan foydalanadilar. Bu shart emas va ishlab chiquvchi oddiy JavaScript-da yozishi mumkin, ammo JSX-dan foydalanish ancha oson.
- **O'rganish oson:** agar kimdir dasturlash bo'yicha oldindan ma'lumotga ega bo'lsa, u holda Reactni oson tushunishi mumkin, ammo burchak va Ember "domenga xos til" deb nomlanadi, bu ularni o'rganishni qiyinlashtiradi. React uchun sizga CSS va HTML bo'yicha oddiy bilimlar kerak.
- **Mahalliy yondashuv:** Reakt yordamida mobil ilovalar yaratish mumkin (React Native).
- **Ma'lumotlarni bog'lash:** React bir tomonlama ma'lumotlarni bog'lashni qo'llaydi va Flux deb nomlangan dastur arxitekturasini ma'lumotlar oqimini tarkibiy qismlarga bitta boshqaruv nuqtasi - dispatcher orqali boshqaradi. Katta ReactJS dasturlarining o'z-o'zidan tuzilgan tarkibiy qismlarini disk raskadrovka qilish osonroq.
- **Ishlash:** Reakt qaramlik uchun o'rnatilgan konteyner haqida hech qanday ma'lumot bermaydi. Siz bog'liqliklarni avtomatik ravishda kiritish uchun Babel, ReactJS-di orqali foydalanishingiz mumkin bo'lgan Browserify, Require JS, EcmaScript 6 modullaridan foydalanishingiz mumkin.

- Sinov qobiliyati : ReactJS dasturlarini sinab ko'rish juda oson. Reakt ko'rinishlar holatning funktsiyalari sifatida ko'rib chiqilishi mumkin, shuning uchun biz ReactJS ko'rinishiga o'tadigan holat bilan manipulyatsiya qilishimiz va chiqish va tetiklenen harakatlar, hodisalar, funktsiyalar va boshqalarni ko'rib chiqishimiz mumkin.

## Reaksiyaga bo'lgan Component - asoslangan

Bu erda ReactJs haqida ta'kidlashni istagan asosiy nuqta - bu komponentlarga asoslangan til va bu erda hamma narsa komponent. Agar siz biron bir foydalanuvchi interfeysini yaratsangiz, u boshqalar uchun komponent sifatida ishlashi mumkin.

Masalan, bitta `<MyHeaderMenu>`komponent yaratishi mumkin edi

- Logotip, sarlavha, pastki sarlavha, qidiruv satrida kirish, navbar menyusini namoyish etish
- Qidiruv satrida kiritilgan matnga tekshiruvlarni bajaring
- Foydalanuvchi kirish tugmachasini bosganda shaklni yuboradi

Faqat bitta kod satrini qo'shish orqali `<MyHeaderMenu></MyHeaderMenu>:`

*Burchak dunyosidan kelganlar uchun siz React Componentlarni Direktivalarga yaqin o'xshashlik deb hisoblashingiz mumkin*

Biroq, React sizning qarashlaringiz uchun to'liq o'rnini bosa oladi, chunki siz tarkibiy qismlarni joylashtirishingiz mumkin. Katta dastur yaratish uchun siz kichik komponentlardan foydalanishingiz mumkin, bu komponentlar standart HTML komponentlari bo'lishi mumkin yoki siz o'zingizning komponentlaringizni yaratishingiz mumkin.

Komponentlardan foydalanishning eng katta afzalligi shundaki, istalgan komponentni istalgan vaqtda istalgan vaqtda boshqa dasturlarga ta'sir qilmasdan o'zgartirishingiz mumkin. Ushbu funktsiya ma'lumotlar tez-tez o'zgarib turadigan kattaroq va real vaqtda dasturlarda qo'llanilganda eng samarali hisoblanadi.

Biz biron bir ma'lumotni yangilaganimizda yoki o'zgartirganimizda, ReactJS holati *aslida* o'zgargan o'ziga xos komponentni avtomatik ravishda yangilaydi . Bu o'zgarishlarni aks ettirish uchun brauzerni butun dasturni qayta yuklash vazifasidan xalos qiladi.

ReactJs yordamida to'liq taqdim etilgan veb-sahifaning misoli bo'lishi mumkin

## ReactJS ning kuchli tomonlari

- Reakt foydalanuvchi interfeyslarini alohida komponentlarga ajratishga imkon beradi.
- React ma'lumotlarning bir yo'nalishli oqimi foydalanuvchi interfeysini yangilashni oldindan va ishonchli qiladi.
- Hayotiy tsikl davomida o'zgarib turadigan barcha ma'lumotlarni foydalanuvchi interfeyslarini yaratish uchun bitta **davlat** kodida saqlash orqali intuitiv va tushunarli bo'lishi mumkin.

Tayyorladi: Aminjonov Tokhirbek