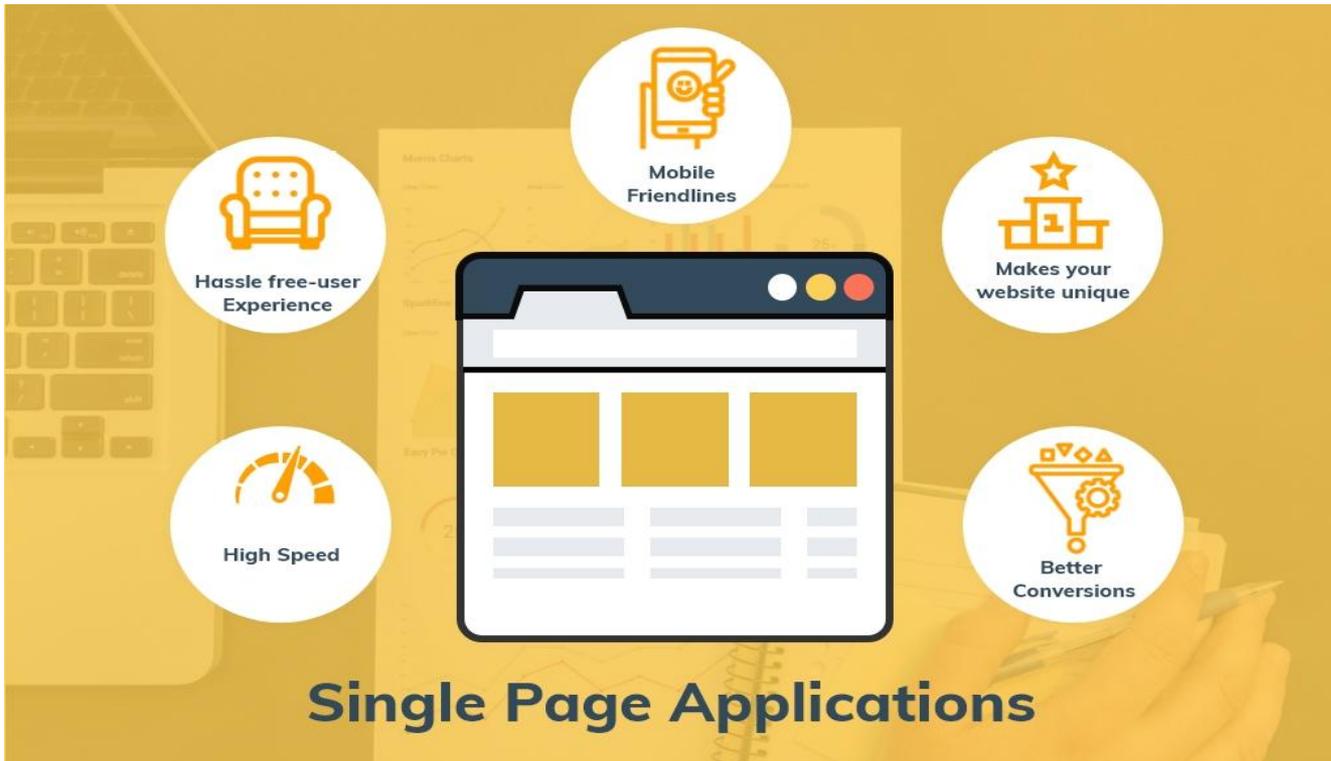


# Single Page Application

## На примере VueJS



# Обо мне

[github.com/volokolamskspb](https://github.com/volokolamskspb)

<https://habr.com/ru/users/sashavoloh>

<https://t.me/julsmao>



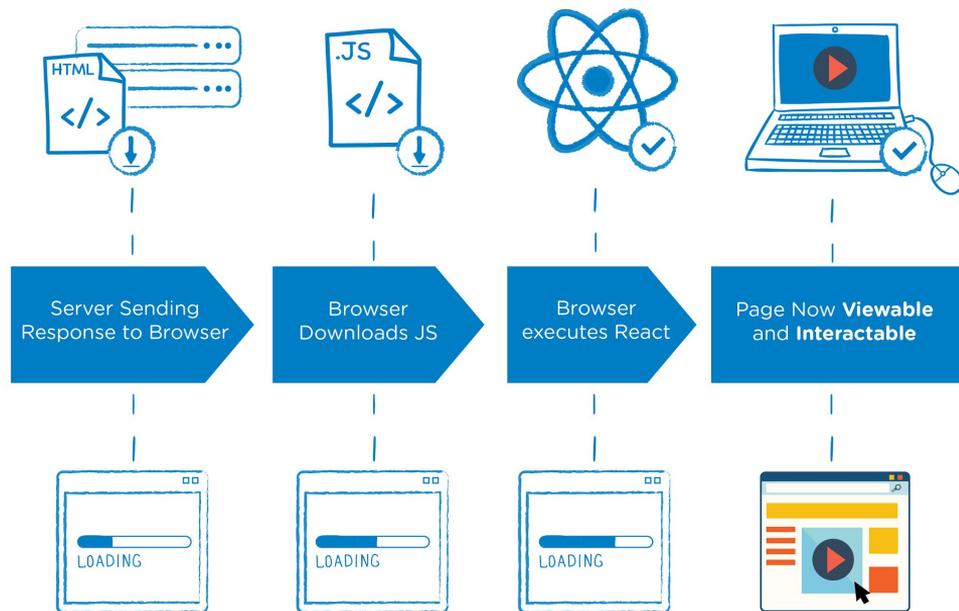
Юлия Кузнецова

Frontend разработчик

 Moscow Exchange

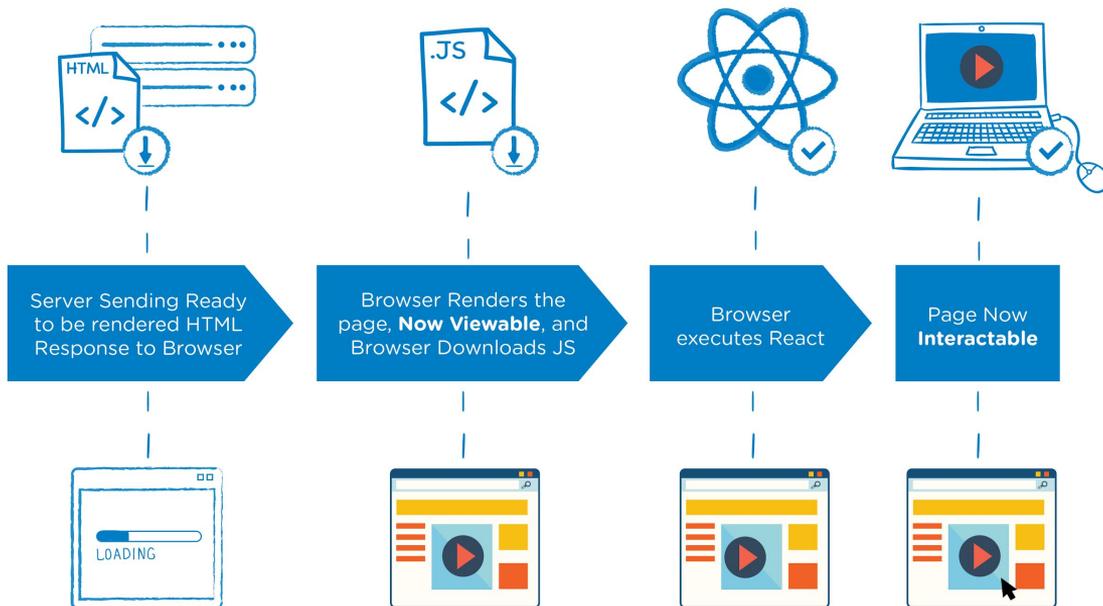
# Рендеринг на стороне клиента

## CSR



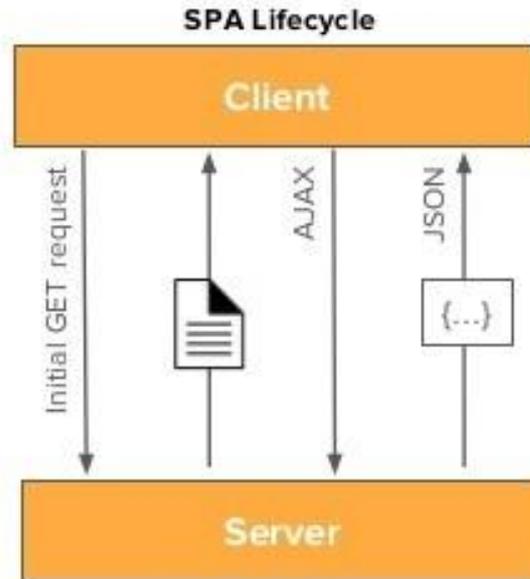
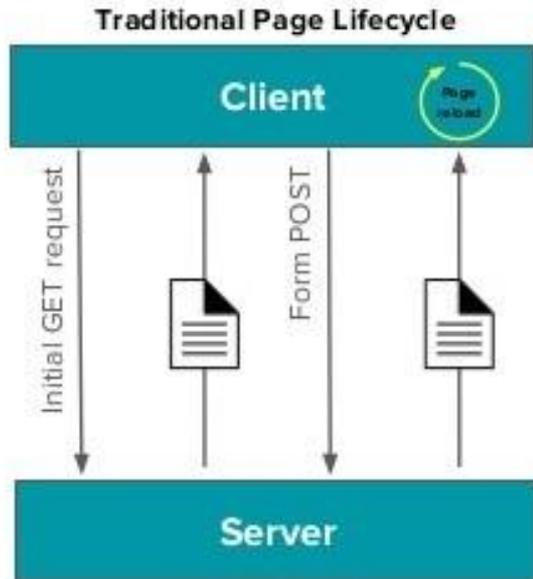
# Рендеринг на стороне сервера

## SSR



# Что такое SPA

Single Page Applications work differently



# Преимущества

- Скорость за счет разбивки на чанки
- User Experience
- Меньше нагрузка на сервер
- Универсальная платформа
- Отзывчивость интерфейса

# Почему VueJS



- Минималистичность
- Скорость
- Простота

# Routing



- Режим #хэша  
<http://oursite.com#user>
- History mode  
<http://oursite.com/user/id>
- Редирект на index.html

# Пример

- Передача параметров через props
- Redirect
- Watch

```
1
2 const User = {
3   props: ['id'],
4   template: '<div>Пользователь {{ id }}</div>'
5 }
6 const router = new VueRouter({
7   routes: [
8     { path: '/user/:id', component: User, props: true },
9   ]
10 })
```

```
1
2 <router-view class="view one"></router-view>
3 <router-view class="view two" name="user"></router-view>
4 <router-view class="view three" name="product"></router-view>
5 <router-link to="/user">Go to User</router-link>
```

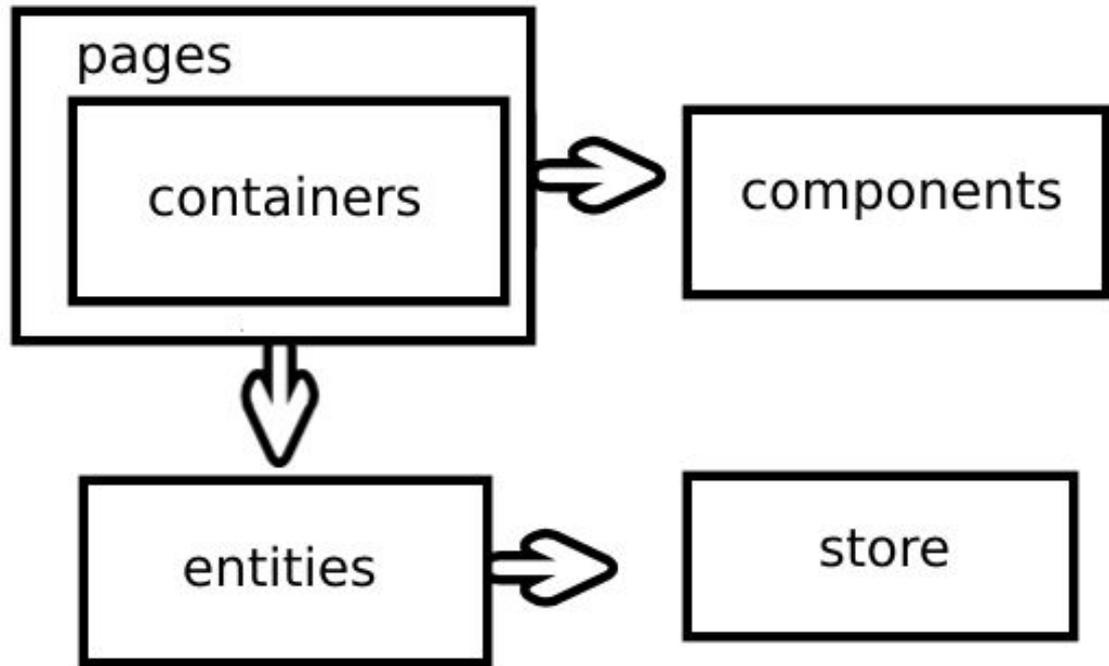
```
import Vue from "vue";
import Router from "vue-router";
import Home from "./pages/Home.vue";
import About from "./pages/About.vue";
import Product from "./pages/Product.vue";
import Cart from "./pages/Cart.vue"
```

```
Vue.use(Router);
```

```
export default new Router({
  routes: [
    {
      path: "/",
      name: "home",
      component: Home
    },
    {
      path: "/about",
      name: "about",
      component: About
    },
    {
      path: '/products/:id',
      name: "product",
      component: Product
    },
    {
      path: "/cart",
      name: "cart",
      component: Cart
    }
  ]
});
```

# Архитектура SPA

- Модель
- Представление
- Контроллер



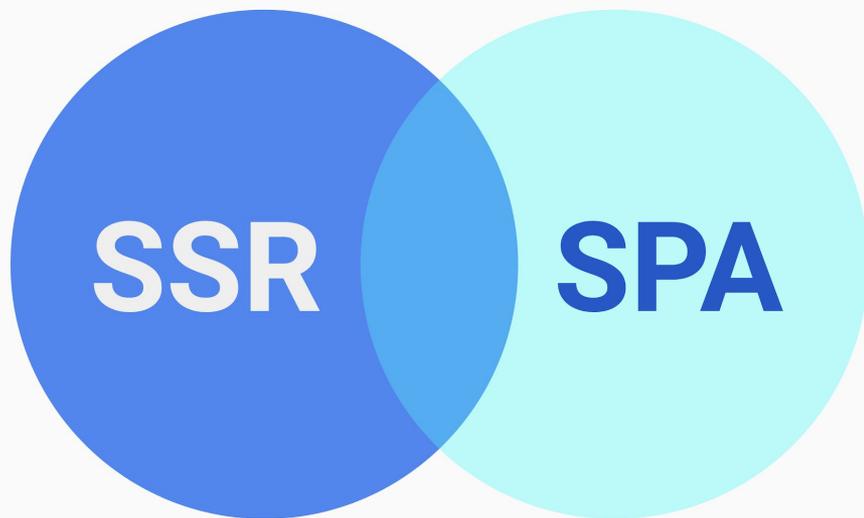
# Vue CLI

- Сборка проекта в 1 шаг
- Возможность конфигурировать webpack
- Конфигурация сборки для production
- Hot reload в режиме разработки

# Преимущества SSR

-  Оптимизация в поисковых системах
-  Оптимизация в соц. сетях
-  Производительность
-  User Experience

# SPA vs SSR



- Для React используется [Next.js](#)
- Для Vue.js популярен [Nuxt.js](#)

```
export default {  
  computed: {  
    fooCount () {  
      return this.$store.state.foo.count  
    }  
  },  
  
  // Server-side only  
  serverPrefetch () {  
    this.registerFoo()  
    return this.fooInc()  
  },  
  
  // Client-side only  
  mounted () {  
    const alreadyIncremented = !!this.$store.state.foo  
  
    this.registerFoo()  
  
    if (!alreadyIncremented) {  
      this.fooInc()  
    }  
  }  
}
```

# Vue 3

- Всплытие статического контента
- Tree-shaking
- TypeScript
- Proxies
- Разделение слота и родителя
- Composition API

```
1
2  import { reactive, computed } from 'vue'
3
4  export default {
5    setup() {
6      const state = reactive({
7        count: 0,
8        double: computed(() => state.count * 2)
9      })
10
11     function increment() {
12       state.count++
13     }
14
15     return {
16       state,
17       increment
18     }
19   }
20 }
21
```

- Рендеринг на стороне клиента
- Рендеринг на стороне сервера
- Что такое SPA
- Почему VueJS
- Routing
- Архитектура SPA
- Vue CLI
- Преимущества SSR
- SPA vs SSR
- Vue 3