

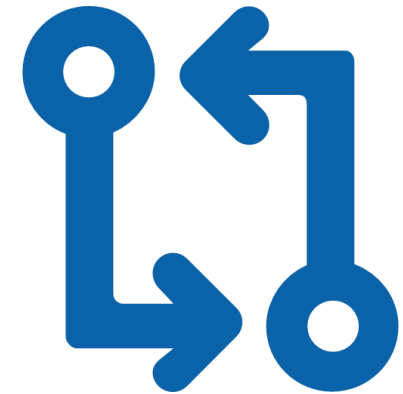


**АНАЛИЗ ПРОБЛЕМ
СОВРЕМЕННОЙ ВЕБ-РАЗРАБОТКИ
И ПУТИ ИХ РЕШЕНИЯ
ПРИМЕНЯЕМЫЕ В \$MOL.**



ПРОБЛЕМА: ДЛИННЫЙ ЦИКЛ ОТЛАДКИ

- | Редактирование кода
- | Запуск приложения
- | Проверка и обнаружение проблем
- | Исследование причины проблем



И так для любой опечатки.

Чем быстрее разработчик поймёт где и почему ошибся, тем быстрее он всё реализует.



ПРОБЛЕМА: ГДЕ ЧТО ЛЕЖИТ?

- | Как быстро перейти к определению сущности?
- | Как быстро найти все места использования сущности?

Детективные расследования снижают полезную продуктивность.





ПРОБЛЕМА: КАК ЭТО ИСПОЛЬЗОВАТЬ?

- | Что это за объект?
- | Какие поля и методы есть у объекта?
- | Какие параметры принимает функция?
- | Какие вообще есть классы и функции?
- | Правильно ли я использую сущность?



Ответы требуют времени, иногда продолжительного.



РЕШЕНИЕ: СТАТИЧЕСКАЯ ТИПИЗАЦИЯ

- | Проверка кода по мере редактирования
- | Быстрая навигация по коду
- | Контекстные подсказки и документация

Type Script

TypeScript = современный JavaScript + типизация



ПРОБЛЕМА: МНОГО КОДА

- | Много ошибок
- | Медленная разработка
- | Медленная работа приложения
- | Требуется больше тестов





ПРОБЛЕМА: РАСТУЩАЯ СЛОЖНОСТЬ

- | По мере роста приложения всё сложнее вносить в него изменения не разломав
- | Требуется учитывать всё большее число состояний одновременно, разбросанных по разным частям приложения
- | Множество внутренних соглашений неизбежно начинают противоречить друг другу (например, что в каком порядке следует вызывать)





ПРОБЛЕМА: ОБНОВЛЕНИЕ ПРИЛОЖЕНИЯ

- | Требуется перезагрузка всей страницы
- | Пользователь выпадает из контекста работы (сбрасываются позиции скроллингов, открытые панели, редактируемый текст)
- | Разные пользователи работают с разными версиями приложения одновременно



Обычно об этих проблемах не думают, просто некоторым пользователям иногда не везёт.



РЕШЕНИЕ: РЕАКТИВНАЯ АРХИТЕКТУРА

- | Описание состояний вместо действий
- | Надёжная каскадная синхронизация состояний
- | Эффективное обновление состояний
- | Минимум обслуживающего кода
- | Обновление приложения налету незаметно для пользователя

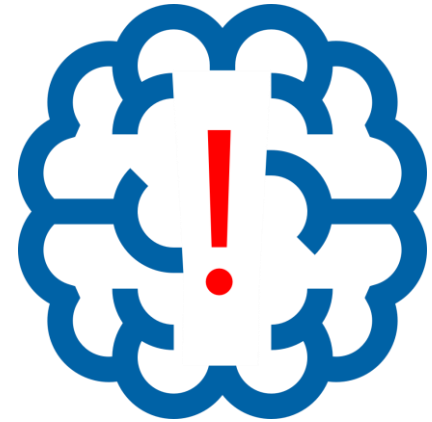


Статья о принципах работы [по ссылке](#)



ПРОБЛЕМА: СЛОЖНОСТЬ ОСВОЕНИЯ

- | Много разных сложных абстракций
- | Не очевидные соглашения
- | Много запутанного кода
- | Средства отладки зачастую не помогают
- | Требуются специалисты высокого класса
- | Низкая отдача от специалистов низкого класса





ПРОБЛЕМА: НЕУНИВЕРСАЛЬНОСТЬ

- | Несколько решений одной задачи
- | Сложно добавить функционал в готовое решение
- | Компонент умеет слишком много вообще и слишком мало из того, что требуется
- | Адаптивная реализация требует дополнительного существенного времени





РЕШЕНИЕ: \$mol

- | Минимум абстракций и соглашений
- | Статическая типизация
- | Реактивная архитектура
- | Высокая эффективность
- | Компактный размер
- | Простота компоновки
- | Динамическая адаптация к возможностям устройства
- | Высокая настраиваемость



mol



ПРОБЛЕМЫ EXT JS

- | Длинный цикл отладки
- | Где что лежит?
- | Как это использовать?
- | Много кода
- | Растущая сложность
- | Обновление приложения
- | Сложность освоения
- | Неуниверсальность
- | Производительность

Ext JS



ПРОБЛЕМЫ SAP UI5

- | Длинный цикл отладки
- | Где что лежит?
- | Как это использовать?
- | Много кода
- | Растущая сложность
- | Обновление приложения
- | Сложность освоения
- | Неуниверсальность
- | Производительность





ПРОБЛЕМЫ ANGULAR JS

- | Длинный цикл отладки
- | Где что лежит?
- | Как это использовать?
- | Растущая сложность
- | Сложность освоения
- | Неуниверсальность
- | Производительность





ПРЕИМУЩЕСТВА \$mol

Быстрая разработка

За счет простой архитектуры и обилия
ГОТОВЫХ КОМПОНЕНТОВ

| | | | | | |
|---|---|---|--|----------------------|--|
| <input type="radio"/> Radio button | <input type="checkbox"/> Checkbox | <input type="checkbox"/> Switcher off | <input checked="" type="checkbox"/> Switcher off hover | Link | Link with a chevron > |
| <input checked="" type="radio"/> Selected | <input checked="" type="checkbox"/> Checked | <input checked="" type="checkbox"/> Switcher on | <input checked="" type="checkbox"/> Switcher on hover | | |
| <input type="radio"/> Hover | <input type="checkbox"/> Hover | <input type="checkbox"/> Disabled | | | |
| <input checked="" type="radio"/> Selected hover | <input checked="" type="checkbox"/> Checked hover | | | | |
| <input type="radio"/> Disabled | <input checked="" type="checkbox"/> Disabled | | | | |

Correct text

✓

Entered text

!

🔍

Search

Unable Button

Dropdown Button ▾

Dropdown Button ▾

eigenmethod.com

↻

Dropdown Item Hover

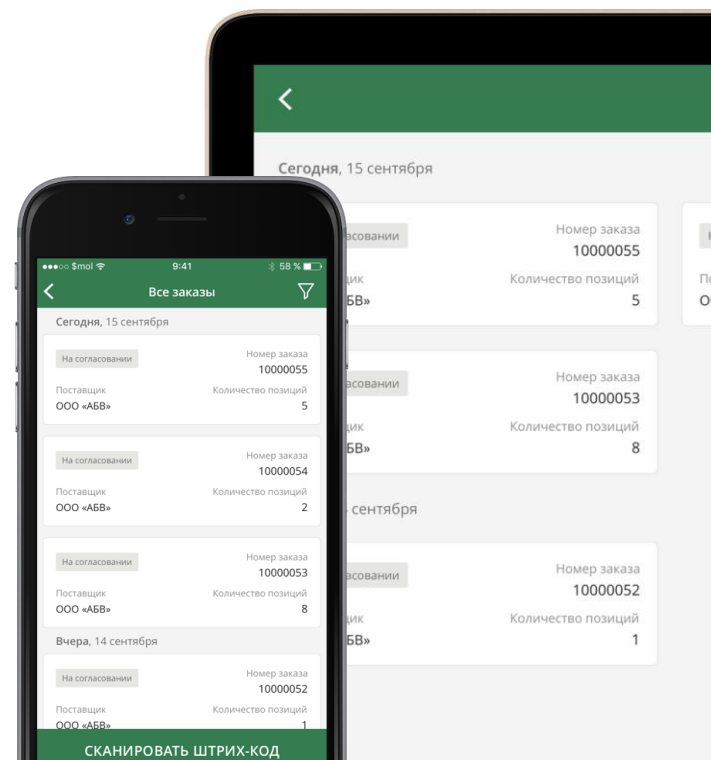
Dropdown Item

ПРЕИМУЩЕСТВА \$mol



Умный дизайн

Адаптивные компоненты подстраиваются под любые разрешения экранов и типы устройств, а уникальный дизайн позволяет производить тонкие настройки под Ваш бренд.





ПРЕИМУЩЕСТВА \$mol

Высокая скорость работы

За счёт интеллектуальных алгоритмов отрисовки экранов приложения и загрузки данных по требованию.



Надёжность и минимизация ошибок

За счёт малого объёма кода, использования реактивной архитектуры и логирования всех изменений.





Спасибо за внимание!



Москва

115114, 1-й Дербеневский пер.
дом 5, офис 305,
тел: +7 (495) 663-77-59



Санкт-Петербург

197342, ул. Белоостровская,
17/2А, офис 309,
тел: +7 (812) 438-27-84



Киев

02140, пр-т Бажана,
14а, 3 этаж
тел: +38 (044) 585-22-43

www.saprun.com