

Features & FAQ

Реактивный подход - изменение данных влечёт за собой лавину обновлений всех связанных значений.

Yaml подобный синтаксис - устали писать открывающий, закрывающий тэг, вставлять портянки почти одинаковой вёрстки для списка текстовых полей? Всё, этого больше не надо делать.

Компоненты - это хорошо и yaml синтаксис здорово сократил количество кода, но мне надо написать калькулятор с четырьмя арифметическими операциями, неужели я буду вынужден 4 раза написать

```
#button: +  
.click()->  
  result = a+b
```

подобный код для каждой операции? И снова - нет!

```
#operationButton = {{component}}:{{operation}} -> // объявляем макрос!  
#button: #{{operation}}  
.click()->  
  result = a #{{operation}} b
```

```
#operationButton:+  
#operationButton:-  
#operationButton:*  
#operationButton:/
```

Но как это отлаживать?

Средства отладки позволяют сосредоточиться на текущей задаче, отбросив всё окружающее её изменчивое состояние. Достаточно один раз привести приложение к требуемому состоянию и можно пробовать различные варианты кода, решающего задачу. Ура, можно решать бизнес задачи методом глупого перебора пока результат не сойдётся! Но не забывайте сделать тесты для различных исходных значений, иначе может получиться решить только один конкретный случай и проигнорировать остальные. Чуть не забыл, сохранённое исходное состояние можно клонировать, а потом хачить на живую значения полей и получить новое состояние начальных данных для теста.

Не хочу yaml, хочу зелёную кнопку с блёстками!

И в этом случае у нас есть что предложить. Визуальный редактор даёт сделать кнопку сколь угодно зелёной!

И вот я отправляю данные на сервер.

В зависимости от результата переходим на другой экран или меняем данные на текущем, всё что пожелаете.

А если я хочу сделать веб приложение с разделами?

Реактивный роутер позволяет завязать на себя текущий отображаемый экран или открытый таб. Путь в браузере оказывается напрямую связан с интерфейсом приложения. А интерфейс приложения точно так же напрямую оказывает влияние на путь в браузере.