

## Модель данных

Ключевым компонентом фреймворка является реактивная реализация хранилища данных. Под реактивным хранилищем подразумевается такая структура данных, которая абстрагирует добавление\удаление\изменение своих узлов и при этом способна генерировать соответствующие события.

### Реактивность

Самой наглядной демонстрацией концепции реактивности является Excel. Одни ячейки зависят от других, изменении одного значения способно породить моментальное изменение значений всех связанных ячеек.

Пример:

```
A = 10
B = A + 5
```

В императивной реализации последующее изменение A не затронет B, в реактивной же -- B является не конкретным значением, а формулой от A.

### Типы данных

**Key-value:** объект, состоящий из пар ключ-значение. Генерирует события:

- > \* **Создание новой записи**
- > \* **Изменение существующей**
- > \* **Удаление**

**Массив:** некоторая последовательность (обычно однотипных) данных. События:

- > \* **Добавление элемента**
- > \* **Вставка элемента\элементов**
- > \* **Модификация элемента**
- > \* **Удаление элемента**

> Ссылка: любое значение может быть как конкретным значением, так и формулой с зависимостью от других элементов

### Базовая модель

Обычная модель, которая может как использоваться в чистом виде, так и быть основой для всех перечисленных ниже типов зависимых моделей. Элементы этой модели могут иметь любой из приведённых выше типов.

### Взаимоотношения моделей

Перечисленные ниже отношения могут быть применены друг к другу в любом порядке и иметь любую степень вложенности.

### Компоновка

Хранимые объекты могут иметь любую степень вложенности. В массивах могут лежать объекты, в которых лежат массивы. При удалении\модификации такого сложного узла модель сообщает всем завязанным на дочерние элементы значениям об этом событии.

### Срез

Модель даёт возможность вынести любой узел в отдельную модель. После этой операции первоначальная модель и получившаяся остаются связанными, т.е. модификация среза приводит к модификации вложенного дерева базовой

модели.

## Копия

Аналогично срезу, но теряется связь с базовой моделью.

## Наследование

Модель Б может наследовать модель А. В этом случае добавление\изменение\удаление узлов в модели Б не приводят к изменению модели А. Но если у модели Б запрашивают некоторое значение, которого в ней нет, то значение ищется у родительской модели. Наследование может быть любого уровня вложенности. По своей природе данное наследование ближе всего к "прототипному".

## Группировка

Reduce модель. Используется для применения операции группировки и порождения новой модели, зависимой от родительской (обычно применяется на моделях-массивах). Например, группировка способов оплаты по поддерживаемым валютам. Изменение элемента модели может привести к его переходу в другую группу.

## Сортировка

Зависимая модель, являющаяся отсортированным вариантом родительской (обычно применяется на моделях-массивах). Изменение данных используемых для сортировки в любом элементе ведёт к проверке правильности положения данного элемента в списке. При перемещении элемента модель ведёт себя так же как массив, а именно - сообщает что элемент удалился со старого места и добавился в новое.

## Фильтр

Зависимая модель, все данные которой удовлетворяют заданному фильтру (обычно применяется на моделях-массивах).

## Мар

Зависимая модель, каждый элемент которой зависит формулой от родительской модели. Например, массив списка цен в рублях может быть приведён с помощью формулы от курса доллара к массиву цен в долларах.

## Комбинации

Взаимоотношения моделей могут выстраиваться в цепочку в любом порядке.

## Возможности

Данная реализация модели даёт возможность чётко видеть потоки данных: наблюдать как исходные данные трансформируются в используемые в бизнес логике и пользовательском интерфейсе.

Возможность минимальной модификации пользовательского интерфейса при изменении данных. При удалении элемента из списка - в html вёрстке будет удалён только отвечающий за этот элемент блок, без перерисовки всего окружения. А при модификации некоторого текстового значения - будет модифицирован всего один TextNode. Обычно для такой хирургической точности приходилось писать настолько большой объём кода, что как правило программисты избегали применения данной практики. Но это очень важная часть, увеличивающая производительность изменений интерфейса, что особенно важно на мобильных и embed платформах.

## Результат

- Рабочая модель со всеми типами зависимостей и событий.
- Так как модель является ключевым элементом фреймворка - покрытие тестами будет составлять 100%.
- Документация по модели, её методам, генерируемым событиям и всем зависимостям. Документация по зависимостям будет сопровождаться поясняющими схемами.
- Каждый раздел документации включает в себя примеры кода (часть примеров будет взята из тестов), которые можно будет проверить в работе непосредственно на странице документации (интерактивное исполнение и вывод результатов).
- \*Дополнительно. После реализации IDE примеры кода станет возможно модифицировать (необходим тот же модуль редактора с подсветкой синтаксиса и автодополнением, который будет разработан в рамках IDE)