Построение синтаксического дерева.

Любой язык программирования (кроме функциональных типа lisp, haskel, erlang) перед исполнением программы строит её AST (abstract syntax tree) представление. Так как в фреймворке используется нестандартный синтаксис - нам этого шага тоже не избежать.
Абстрактное синтаксическое дерево - это то, что переводи 2 + 4 в 2, 4, + (именно в такой последовательности это исполняется процессором. Это шаг перевода исходного текста программы в то что может на самом деле исполняться.

Польза

Во-первых, это - один из ключевых моментов фреймворка и без этого он просто не взлетит.
2) AST даёт возможность производить оптимизацию, заменяя некоторые паттерны другими более оптимальными.
3) Каждый компонент будет иметь возможность любой кастомизации синтаксиса. Это метапрограммирование такого уровня, что можно не просто перегрузить оператор сложения или написать макросы, а сделать **другой** синтаксис, принципиально другой. Эта возможность не очень нужна стандартным компонентам, но может быть полезна уже в компоненте соединения с базой данных, что бы 'DB postges user1@host:port db1' выдрал все составные части из такой строки и разложил в соответствующие переменные.

Реализация

Значительная часть парсера уже написана
Написать абстрактную часть (пункт 3)
Придумать синтаксис для многострочных строк и просто переносов длинных выражений.

Результат

После этого шага код, записанный в формате фреймворка (та самая помесь yaml, реактивности и es2015) превращается в дерево компонентов со свойствами.