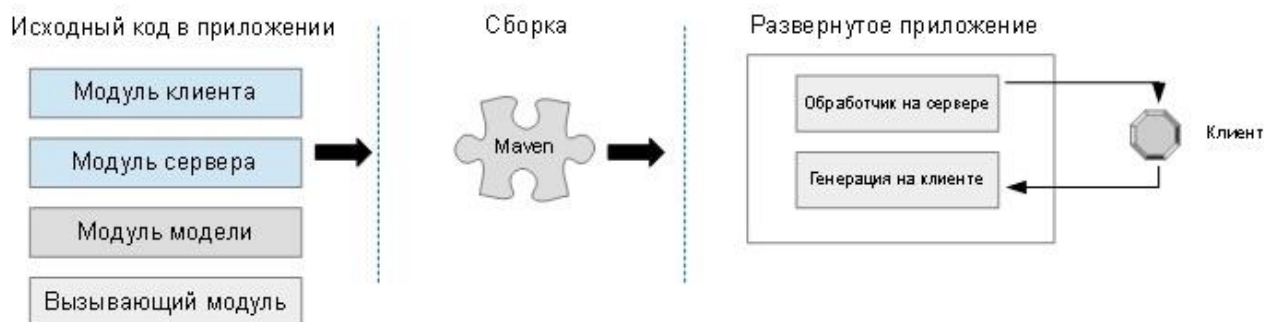
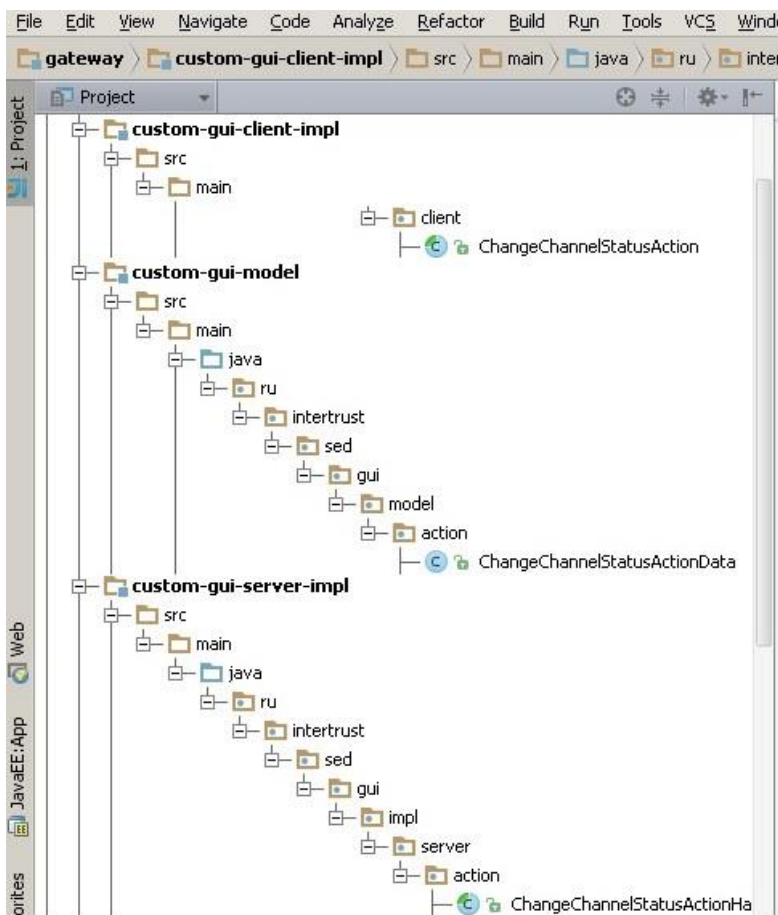


АРХИТЕКТУРА ПОЛЬЗОВАТЕЛЬСКОГО СОБЫТИЯ



Реализация пользовательского события распределена по трем основным модулям проекта. В основе этого подхода лежит использования фреймворка GWT. В модуле клиента реализован класс отвечающий за генерацию события. В модуле серверной части реализован класс обработчика данного события. В модуле модели реализован класс объекта передачи данных (DTO) и контекста (если не используется стандартный контекст). Этот модуль должен быть общедоступным для предыдущих двух.



При сборке проекта фреймворк GWT генерирует скрипты для клиентской части приложения опираясь на модуль клиента и объявление использования события в конфигурационном файле вызывающего модуля. При вызове данного скрипта на клиентской машине, событие обрабатывается на стороне сервера обработчиком, используя данные переданные в контексте, и возвращает результат клиенту в виде объекта передачи данных.

На данном примере у нас есть три основных модуля в проекте. Пользовательское событие, которое меняет статус канала реализовано в модуле **custom-gui-client-impl** в классе **ChangeChannelStatusAction**. Объект передачи данных описан в модуле модели **custom-gui-model**. Обработчик события реализован в модуле **custom-gui-server-impl**.

Рисунок 1. – Экран проекта.

Очень важно, чтобы клиентская и серверная имплементация находились в правильных модулях, клиентском и серверном соответственно. Если мы поместим одну из этих частей (или обе) в произвольные модули проекта, не являющиеся «клиентом» и «сервером» с точки зрения фреймворка GWT, то пользовательское событие не будет работать, даже при условии, что компиляция проекта пройдет нормально и весь остальной функционал будет работать правильно.