

РАЗРАБОТКА ЭРГОНОМИЧНОГО ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА С ИСПОЛЬЗОВАНИЕМ ФРЕЙМВОРКА GOOGLE WEB TOOLKIT

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Чех Д. О.

Жданович С.В. – к.т.н., доцент

Проектирование и разработка эргономичного пользовательского интерфейса с использованием фреймворка GWT.

Google Web Toolkit (GWT) – это набор инструментов для разработки интернет-приложений RICH (RIA). Вот некоторые из его примечательных особенностей:

- GWT предоставляет разработчикам возможность написать клиентское приложение на JAVA.
- GWT компилирует код, написанный на JAVA, в код JavaScript.
- Приложение, написанное на GWT, является кросс-браузерным. GWT автоматически генерирует код JavaScript, подходящий для каждого браузера.
- GWT является открытым исходным кодом, полностью бесплатным и используется тысячами разработчиков по всему миру. Он лицензируется под лицензией Apache версии 2.0.

В целом, GWT – это инфраструктура для создания крупномасштабных и высокопроизводительных веб-приложений с простотой обслуживания. Основываясь на Java, вы можете использовать JAVA IDE, такие как Eclipse или IntelliJ Idea, для разработки GWT-приложения.

Разработчики могут использовать автозаполнение кода / рефакторинг / навигацию / управление проектами и все функции IDEs.GWT, которые обеспечивают полную возможность отладки. Разработчики могут отлаживать клиентское приложение так же, как Java-приложение.

GWT обеспечивает простую интеграцию с Junit и Maven, что позволяет упростить и ввести единообразие для модульного тестирования и сборки всего проекта, включающего серверную часть, написанную на Java.

GWT генерирует оптимизированный код JavaScript, самостоятельно создает специальный код JavaScript для браузера.

GWT предоставляет библиотеку виджетов, которая выполняет большинство задач, необходимых в приложении. Также пользовательский виджет может быть создан для удовлетворения потребностей приложений.

Кроме того, приложения GWT могут работать на всех основных браузерах и смартфонах, включая телефоны / планшеты на базе Android и iOS.

SDK предлагает некоторые элементы пользовательского интерфейса и макеты для проектирования графических интерфейсов. Все компоненты пользовательского интерфейса происходят из класса Widget. Визуально есть виджеты элементов, которые пользователь может видеть, нажимать или перемещать на экране: виджеты элементов - TextBox, TextArea, Button, RadioButton, CheckBox и т.д. И есть виджеты макета или панели, которые составляют и организуют экран: виджеты панели - HorizontalPanel, VerticalPanel, PopupPanel, TabPanel и т.д.

Каждый раз, когда разработчик добавляет виджет или любой другой компонент в код, GWT усердно работает, чтобы связать элемент представления с DOM браузера. Конструктор всегда инициализирует корневой элемент DOM. Когда мы присоединяем дочерний виджет к родительскому компоненту, это также вызывает привязку на уровне DOM. Класс точки входа содержит функцию загрузки, которая будет вызвана первой. В этом месте идет объявление используемых виджетов.

Отложенное связывание (Deferred Binding) является одним из важных элементов фреймворка. Оно позволяет GWT формировать хорошо оптимизированный код JavaScript. Следует отметить, что GWT компилятор не поддерживает java reflection или динамическую загрузку классов, также называемую динамическим связыванием. Связано это с тем, что JavaScript среда, в которой GWT приложение будет функционировать, не поддерживает это. Ввиду невозможности динамического связывания в GWT, вместо него используется так называемое отложенное связывание Deferred Binding. Отложенное связывание похоже на динамическую загрузку классов. Если динамическая загрузка класса происходит во время выполнения приложения, то отложенное связывание происходит во время компиляции.

Во время компиляции клиентского кода GWT приложения компилятор определяет "особенности браузера", которые код должен поддерживать, и генерирует оптимизированный вариант приложения для каждой конкретной конфигурации. Поэтому при правильной настройке файла конфигурации можно получить различные версии приложения для каждого из поддерживаемых браузеров. Но отложенное связывание это нечто большее, чем просто мультибраузерная поддержка.

Хороший пример того, где отложенное связывание может быть полезным является мультязыковая поддержка. GWT компилятор использует Deferred Binding для создания совершенно разных версий приложения для каждого языка. Таким образом, если GWT поддерживает 4 типа разных браузеров и в файле конфигурации приложения будет определено 4 языка, то GWT компилятор будет генерировать 16 различных реализаций. Во время запуска и выполнения, GWT выбирает подходящую версию приложения для браузера.

Отложенное связывание имеет несколько преимуществ:

- снижает размер сгенерированного JavaScript кода, таким образом клиенту будет необходимо скачать только код необходимый для запуска под определенным браузером (использовано в Internationalization модуле).

- автоматически генерирует код, имплементирующий интерфейс или создающий прокси класс - использовано в GWT RPC модуле.

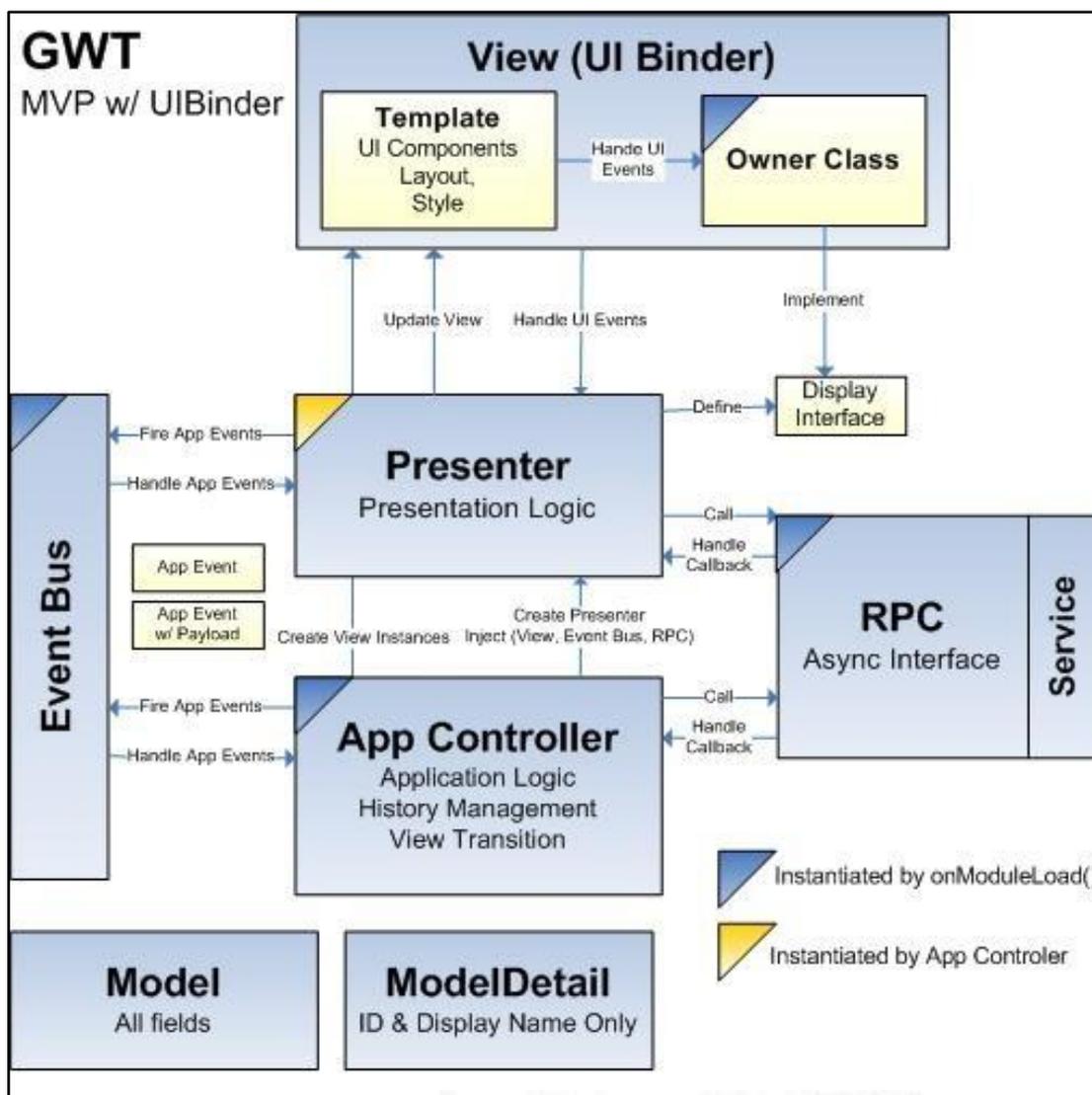


Рисунок 1 – Архитектура фреймворка GWT

Список использованных источников:

1. GWT - краткое руководство. – Режим доступа: <https://coderlessons.com/tutorials/veb-razrabotka/izuchite-google-web-toolkit/gwt-kratkoe-rukovodstvo>. – Дата доступа: 21.02.2020.
2. Фреймворк GWT. – Режим доступа: <http://java-online.ru/libs-gwt.xhtml>. – Дата доступа: 22.02.2020.
3. Model-View-Presenter Architecture With GWT. – Режим доступа: <http://e-softlearning.blogspot.com/2011/07/solamenteniel-gwt-model-view-presenter.html>. – Дата доступа: 22.02.2020.