

ОБЗОР ГОТОВЫХ КОМПОНЕНТОВ ДЛЯ РАЗРАБОТКИ ИНТЕРФЕЙСОВ СОВРЕМЕННЫХ WEB-ПРИЛОЖЕНИЙ

**Горбешко Б. Н., ст. преподаватель каф. СПО ИКС Онищенко Т. В.
Научный руководитель: зав. каф. СПО, профессор Крисилон В. А.
Одесский национальный политехнический университет, УКРАИНА**

Аннотация. В докладе проанализирована целесообразность использования технологий HTML5 (HyperText Markup Language) и JavaScript, а также готовых решений для построения интерфейса с их использованием. Рассмотрены связанные с ними возможности и особенности. Описана разработка системы учёта коммерческих курсов по оказанию образовательных услуг.

Введение. Интернет-приложения — это актуальная отрасль разработки ПО (программного обеспечения), обеспечивающая централизованную работу с данными и удобный, быстрый, платформонезависимый интерфейс. Наиболее актуальным для построения интернет-приложений является набор технологий HTML5+CSS3+JavaScript (CSS — Cascade Style Sheets).

Цель работы. Уменьшение времени разработки за счёт снижения трудоёмкости, а также упрощение поддержки, путём обоснованного выбора интерфейсных библиотек при разработке RIA (Rich Internet Application — Насыщенное Интернет-приложение).

Основная часть работы. При построении клиент-серверных интернет-приложений требования к серверной части второстепенны и подлежат выбору; при создании же клиентского ПО системные требования определить сложно — возникает требование кроссплатформенности. Поскольку различных программно-аппаратных платформ много, наиболее целесообразной выглядит разработка клиентского ПО на кроссплатформенных технологиях — для распространённой, лёгкой в установке, а также ориентированной на сетевые технологии среды. Этим требованиям соответствуют web-браузеры. До относительно недавнего времени их возможности не позволяли строить приложения сложнее «тонких клиентов, что вынуждало использовать различные сторонние технологии [1]; сейчас же реализация стандартных технологий в большинстве современных браузеров практически одинакова и включает необходимые для RIA возможности [2].

Тем не менее, браузерные API для JavaScript слишком низкоуровневые для быстрой разработки сложных систем. Рассмотрим наиболее распространённые библиотеки для построения графических интерфейсов «толстых клиентов» по ряду важных для разработчиков критериев (таблица 1), помимо которых, есть и другие особенности, влияющие на выбор библиотеки для конкретного проекта.

Таблица 1 – Сравнение JS-движков для построения интерфейса

| Название библиотеки | Предназначение | jQuery | Лицензия | Стоимость |
|---------------------|----------------|------------|----------|------------|
| DOJO Toolkit | Универсальный | Не зависит | BSD/AFL | бесплатная |

| | | | | |
|---------------|--|------------|-------------------|---------------------|
| jQuery-UI | Сложные элементы интерфейса | Зависит | MIT | бесплатная |
| Kendo UI | Сложные элементы интерфейса и взаимодействие с данными | Зависит | GPL3/Коммерческая | бесплатная/от 699\$ |
| Sencha ExtJS | Универсальный | Не зависит | GPL3/Коммерческая | бесплатная/от 329\$ |
| YUI Framework | Универсальный | Не зависит | BSD | бесплатная |

Для примера рассмотрим выбор интерфейсной библиотеки при разработке автоматизированной системы учебного центра кафедры СПО «К&П 2014». Изначально система разрабатывалась без использования каких-либо готовых JS-библиотек, каждая форма генерировалась на сервере, все изменения внешнего вида также обрабатывались сервером и реализовывались полной перезагрузкой элементов с помощью технологии AJAX. Все формы были независимы и отделены от вывода, изменение архитектуры было слишком трудоёмким. В итоге для ускорения и упрощения разработки проект был переведён на готовую интерфейсную библиотеку.

При выборе библиотеки первоочерёдное внимание уделено наличию многофункционального виджета для работы с табличными данными. Из рассмотренных таким виджетом обладают все библиотеки, однако наиболее удобный API и широкие возможности по расширению оказалось у Kendo UI. На выбор также повлияли такие факторы: табличный виджет поддерживает CRUD-модель (Create, Read, Update, Destroy) и самостоятельную работу с сервером по заранее определённым схемам данных [3]; широкие возможности по работе с окнами; зависимость от jQuery; ориентированность на паттерн проектирования MVVM (Model-View-ViewModel) [3]; поддержка локализации и тем оформления.

Выводы. Использование готовой интерфейсной библиотеки позволяет повысить скорость разработки web-приложений в 2 и более раза за счёт отсутствия временных затрат на самостоятельное создание необходимых компонентов. Также это позволяет быстро снизить трафик между клиентом и сервером, компактно передавая данные (доля служебной разметки при передаче в HTML может достигать до 90% кода, что ощутимо даже при компрессии трафика). К примеру, в разрабатываемой системе JSON-ответы имеют примерно в 5 раз меньший объём, чем HTML; время реакции на запросы увеличилось примерно в 3 раза и находится в пределах 1-2 секунд, при этом количество запросов уменьшилось за счёт пакетного ввода и редактирования данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Национальный Открытый Университет «ИНТУИТ» [Электронный ресурс]. — Режим доступа: URL: <http://www.intuit.ru/studies/courses/611/467/lecture/20763?page=1>. — Лекция 12: Rich Internet application.
2. The Insider's Guide to Next-Generation BPM [Электронный ресурс]. — Режим доступа: URL: http://www.ebizq.net/blogs/cloudtalk/2010/8/the_future_of_rias_is_html5.php. The future of RIAs is HTML5 - Cloud Talk.
3. DENIVIP Media [Электронный ресурс]. — Режим доступа: URL: <http://blog.denivip.ru/index.php/2012/07/разработка-сложных-js-приложений/>. — Разработка крупных JavaScript приложений.