

БЫСТРАЯ РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ ДЛЯ СУЩЕСТВУЮЩИХ ВЕБ-САЙТОВ

М. А. Артемов, А. А. Чиченин

Воронежский государственный университет

Поступила в редакцию 11.11.2011 г.

Аннотация. Рассмотрены проблемы разработки мобильных приложений в дополнение к существующим сайтам, не имеющим развёрнутых программных интерфейсов для взаимодействия с внешними системами. Предлагаются решения для платформ «Android» и «iOS».

Ключевые слова: разработка мобильных приложений, веб-браузер в качестве транспортного уровня.

Annotation. Reviewed problems of mobile application development in addition to existing web-sites that do not have the detailed API for communication with external systems. Solutions for “Android” and “iOS” platforms proposed.

Keywords: mobile application development, webbrowser as a transport layer.

ВВЕДЕНИЕ

Достаточно часто встречается ситуация, при которой в дополнение к существующему веб-сайту требуется разработать мобильное приложение под несколько различных платформ (Android, iOS).

При этом нельзя или нет времени вносить изменения в код существующего веб-приложения. Кроме того, существующее веб-приложение не имеет API для взаимодействия с ним при помощи SOAP, REST и любых других протоколов, и какой-либо документации.

В статье определяется возможность и целесообразность разработки мобильного приложения для iPhone и Android в описанных условиях.

Единственным способом взаимодействия с веб-сайтом в описанном случае является использование протокола HTTP.

Мобильное приложение, поддерживающее данный протокол, должно получать с веб-сайта контент, обрабатывать и отображать его так, как этого требует дизайн мобильного приложения. Кроме того, в некоторых случаях, необходимо предусмотреть возможность отправки запросов с данными обратно на веб-сайт.

В первую очередь необходимо определить где в содержимом HTML страниц сайта нахо-

дится необходимая информация. Для этого можно использовать встроенный отладчик в веб-браузере «Google Chrome», или воспользоваться подобным инструментом для другого браузера.

Затем необходимо определить каким образом на сайте обеспечивается сессионность, то есть идентификация нескольких различных HTTP запросов с конкретным пользователем. Чаще всего эта информация находится в «Cookie» браузера (локальное хранилище произвольных данных, ассоциированных с конкретным веб-сайтом), однако конкретный механизм может быть специфическим для каждого сайта.

Наконец, необходимо реализовать механизмы выделения нужной информации из тела HTML страниц и механизмы сессионности. Ситуация может усложниться, если на сайте активно используется JavaScript. Разумеется, можно встроить в веб-приложение и интерпретатор этого скриптового языка, но такой подход может привести к задаче написания полноценного браузера. Очевидно, что данная задача комплексна и потребует несоизмеримого с результатом объёма работ.

Альтернативным вариантом ручной обработке HTML страниц и работе с протоколом HTTP может являться встраивание в приложение существующего браузера, входящего в

комплект базовых библиотек мобильной платформы. Разумеется, помимо этого потребуется наличие определённых механизмов получения данных из данного браузера в приложение-контейнер. Рассмотрим платформы iPhone и Android по-очереди.

iOS (платформа для устройств iPhone, iPod touch, iPad). В наборе базовых компонентов для разработки под данную платформу существует компонент `UIWebView` [2]. В число достаточно обширных методов класса, реализующего данный компонент, входит метод: *stringByEvaluatingJavaScriptFromString*.

В качестве параметра, данный метод принимает строку, в которой может содержаться произвольный код на языке программирования JavaScript. Этот код будет выполнен в контексте веб-страницы, что позволит получить доступ к её DOM (Объектной модели документа) и необходимую произвольную информацию, которая в виде строки будет возвоащена в качестве результата данного метода. Например, для получения значения элемента из HTML формы на странице можно использовать:

```
[wvstringByEvaluatingJavaScriptFromFromString:@"document.forms[0].elements[0]"];
```

Android. Для данной платформы существует подобный компонент, с названием: `WebView`[1, 2]. Как и свой аналог для платформы iOS, данный класс позволяет выполнить произвольный JavaScript код в контексте веб-страницы. Это осуществляется при помощи метода: `loadUrl`, в качестве параметра необходимо передать строку, начинающуюся с url-схемы: «`javascript:`». Но в отличие от своего аналога для iOS, данный метод не возвратит результат в виде строки.

Для получения данных со страницы в Android предусмотрен более сложный, но вместе с тем более гибкий механизм: при помощи метода `addJavascriptInterface`, необходимо зарегистрировать промежуточный объект, методы которого будут добавлены напрямую в контекст JavaScript. После чего JavaScript будет иметь возможность вызова методов этого промежуточного объекта и передачи в них строковых и числовых данных. Данный подход называется делегированием.

Пример регистрации объекта на языке программирования Java:

```
internalWebView.addJavascriptInterface(
    new JavascriptInjectorInterface()
) {
    @Override
    public void onDataPassed(String data) {
        data == "data";
    }
}, "appInterface");
```

Пример обращения из JavaScript:
`window.appInterface.onDataPassed("data");`

Концептуальная архитектура приложения с применением данного подхода. Как для iOS, так и для Android можно использовать подход, при котором браузер выносится на низкий уровень в качестве программного сервиса (библиотеки). Этот сервис, при необходимости, может создавать несколько экземпляров браузера, работающих параллельно. Библиотека работает независимо от основного кода приложения и не блокирует его выполнение. Основной код приложения делегирует обработку необходимых событий в сервис.

ЗАКЛЮЧЕНИЕ

Используя описанные подходы можно добавить в необходимую веб-страницу произвольный JavaScript код и получить результаты его работы. При этом веб-страница может быть скрыта от конечного пользователя и данные будут отображаться так, как этого требует дизайн приложения.

Использовать данный подход следует с осторожностью, так как в любой момент структура внешнего документа может измениться. Разумеется, необходимо предусмотреть все необходимые уровни проверки входных данных.

Кроме того, описанные компоненты успешно справляются с автоматическим управлением сессионностью, так-как по-сути содержат полноценный веб-браузер.

Следует отметить и тот факт, что подобный подход может быть использован и для простого расширения функциональности сайта. Путём дополнения к его стандартному интерфейсу компонентов, напрямую взаимодействующих с теми аппаратными частями мобильного устройства, с которыми веб-браузер в полном объёме

ёме взаимодействовать не может. Примерами являются: фотоаппарат, динамик, световые индикаторы, датчики положения и освещённости, микрофон, вибровознок и т. п. Это позволяет без существенных затрат получить полноценное клиентское приложение для наиболее распространённых мобильных платформ, даже для сайтов, не представляющих каких-либо API, при этом можно даже расширять их функциональность.

Артемов Михаил Анатольевич – заведующий кафедрой математического обеспечения и администрирования информационных систем Воронежского государственного университета, доктор физико-математических наук, профессор. E-mail: atremov_m_a@mail.ru

Чиченин Александр Александрович – аспирант кафедры программного обеспечения и администрирования информационных систем факультета прикладной математики информатики и механики Воронежского государственного университета. E-mail: achichenin@dataart.com

СПИСОК ЛИТЕРАТУРЫ

1. AndroidDevelopmentGuide : WebView / GoogleInc — (<http://developer.android.com/reference/android/webkit/WebView.html>).
2. Javascript interface with iPhone webview / Stack Exchange Inc — (<http://stackoverflow.com/questions/6160592/javascript-interface-with-iphone-webview>).
3. UIWebView Class Reference / Apple Inc — (http://developer.apple.com/library/ios/#documentation/uikit/reference/UIWebView_Class/).

Artemov Mikhail A. – Head of Department Software & Information System Administering, Voronezh State University, doctor of Physics-math. Sciences, Professor. E-mail: atremov_m_a@mail.ru

Chichenin Alexander A. – Post-Graduate Student, Department Software & Information System Administering, Voronezh State University. E-mail: achichenin@dataart.com