

Анализ и Разработка Системы Push-уведомлений с Использованием Технологий Google Inc

Павлов В.В., Намиот Д.Е.

Аннотация — В настоящее время уровень операционных систем мобильных устройств позволяет технологически использовать различные формы получения информации и контента через приложения. Это позволяет проектировать сервисы, которые не зависят от провайдеров телекоммуникационных услуг (операторов связи). Одним из перспективных направлений в этой области можно выделить создание и развитие системы push-уведомлений для непосредственной отправки сообщений, графического материала и прочей информации посредством сети Интернет на приложения мобильного устройства.

Ключевые слова — push-уведомления, мобильное приложение, мобильное устройство, GCM, сервер приложений, Java.

I. ВВЕДЕНИЕ

Push-технология представляет собой способ уведомления клиентов, который предполагает наличие мобильного устройства с установленным приложением для осуществления подписки к службам на сервере приложений. Идея подхода состоит в получении информации непосредственно мобильным приложением по совершению определенного события. В данном случае событие является триггером для сервера приложений, который срабатывает при определенных условиях, например, в соответствии с установленным расписанием или завершением определенной работы. Сервер приложений и осуществляет рассылку требуемой информации клиенту. Реализация данной технологии подразумевает использование клиент-серверной архитектуры.

В настоящее время эта технология является одним из вариантов предоставления контента пользователям мобильных устройств, например, стоимости акций на биржах, прогноза погоды, либо любого другого информационного сообщения. Большим достоинством является то, что в данном случае от получателя не требуется каких-либо действий в отношении отправителя. Данная технология начинает активно

Павлов Владимир Владимирович – студент третьего курса второго высшего образования факультета вычислительной математики и кибернетики Московского Государственного Университета им. Ломоносова.

Намиот Дмитрий Евгеньевич – кандидат физико-математических наук, старший научный сотрудник лаборатории открытых информационных технологий факультета вычислительной математики и кибернетики Московского Государственного Университета им. Ломоносова.

использоваться именно в силу минимальных затрат, необходимых для ее запуска.

Идея развития push-технологии в мобильных устройствах предполагает, что основное приложение находится в неактивном режиме, и призвана прежде всего преодолеть ряд ограничений, с которыми сталкиваются пользователи и разработчики:

- 1) ограниченный емкостной ресурс системы питания;
- 2) относительно высокий уровень оплаты услуг за контент телекоммуникационным провайдерам (SMS, MMS);
- 3) высокий уровень конкуренции на рынке мобильных операционных систем и как следствие, развитие дополнительных бесплатных сервисов;
- 4) сложность реализации взаимодействия с телекоммуникационной инфраструктурой.

Данные факторы создают основу для популярности мобильных приложений, которые способствуют развитию инфраструктуры для передачи push - сообщений.

II. PUSH-ТЕХНОЛОГИЯ GOOGLE INC

Система облачных сообщений Google (Google Cloud Messaging, далее GCM) - служба, построенная с использованием ресурсов Интернета, которая позволяет организовать рассылку данных со стороннего сервера пользователям устройств с операционной системой Android. Сообщения представляют собой легковесные данные объемом 4 Кб, отправляемые сервером при наступлении определенного типа событий. Данная служба управляет возникающими очередями сообщений и доставкой их до целевого приложения.

Общая схема архитектуры push-уведомлений Google для мобильных устройств представлена ниже.

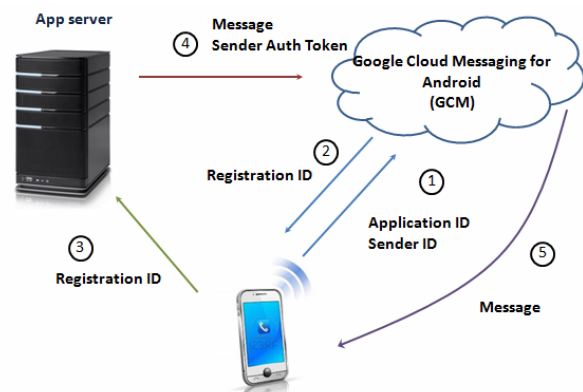


Рис. 1 Google Cloud Messaging

Можно выделить следующие основные особенности GCM-службы [1]:

- 1) возможность использовать сторонний сервер приложений для отправки данных;
- 2) приложение на мобильном устройстве в момент получения сообщения может быть неактивным. В случае соответствующей настройки приложения система запустит данное приложение для получения сообщения в фоновом режиме;
- 3) установленная операционная система Android версии 2.2 и выше;
- 4) наличие Google-аккаунта для устройств с операционной системой версии ниже Android 4.0.4.

Таким образом, для использования мобильного приложения на стороне клиента не предъявляются высокие требования, что, безусловно, позволяет популяризировать технологию.

Для организации полноценной работы система должна включать в себя следующий минимальный набор компонент:

- 1) мобильное устройство;
- 2) сторонний сервер приложений для отправки данных через службу GCM;
- 3) службу GCM – служба доставки сообщений до приложения клиента.

При работе в системе используются следующие параметры доступа [1]:

- 1) идентификатор отправителя (Sender ID) – номер проекта, регистрируемый разработчиком в сервисах Google. Используется в момент регистрации Android-приложения для идентификации устройства при отправке сообщения;
- 2) идентификатор приложения (Application ID) – наименование приложения, которому адресовано сообщение;
- 3) регистрационный идентификатор (Registration ID) – номер, формируемый GCM сервером для Android-приложения, которое позволяет принимать сообщения. Данный идентификатор используется сторонним сервером приложений для рассылки данных;
- 4) аккаунт пользователя в системе Google (Google User Account);
- 5) маркер авторизованного отправителя (Sender Auth Token) – ключ, используемый сервером приложений для доступа к службам Google.

Основные этапы процесса работы данной системы push-уведомления можно выделить в две основные группы: регистрация, отправка и получение сообщения.

Рассмотрим эти группы этапов более подробно [1].

1) Регистрация в GCM.

1.1) В момент регистрации активируется регистрационный Интент (*com.google.android.c2dm.intent.REGISTER*), который содержит идентификаторы отправителя и приложения (1).

1.2) В случае успешной регистрации GCM сервер сообщает приложению регистрационный идентификатор (2).

1.3) Для завершения регистрации Android-приложение отправляет регистрационный идентификатор серверу приложений (3).

2) Отправка и получение сообщения.

2.1) Сервер приложения отправляет сообщение GCM-серверу (4).

2.2) Google ставит в очередь и сохраняет сообщение, если устройство выключено или недоступно.

2.3) Когда устройство доступно, GCM служба перенаправляет сообщение мобильному приложению (5).

2.4) Специальная служба операционной системы Android (notification service) ретранслирует сообщение целевому приложению.

Момент доставки уведомления на мобильное устройство может не совпадать с моментом отправки сообщения GCM службе и получения сервером приложения присвоенного идентификационного номера уведомления, что связано с использованием системы Throttling.

Данная система предназначена для предотвращения лавинообразного потока сообщений пользователю и общей оптимизации работы сети и батареи мобильного устройства. Основана она на использовании ограниченного числа токенов для приложения, которые расходуются по мере поступления уведомлений. На каждое сообщение система вычитает один токен. Токены в свою очередь пополняются с определенным временным лагом. В случае исчерпания лимита сообщения помещаются в буферную очередь GCM службы, что может создавать временные задержки по доставке уведомлений.

В целом можно отметить, что использование данной архитектуры позволит создать достаточно простую и легко масштабируемую систему уведомлений.

III. РАЗРАБОТКА СИСТЕМЫ PUSH-УВЕДОМЛЕНИЙ

Архитектура предлагаемого решения приведена ниже и включает в себя:

- сервер приложения с расширенной функциональностью;
- база данных для хранения и обработки информации;
- клиентское мобильное приложение для Android.

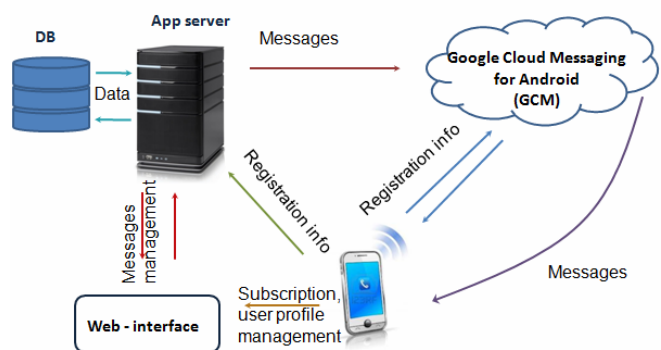


Рис. 2 Сервер уведомлений

На первоначальном этапе разработки системы уведомлений были определены следующие основные требования:

- 1) формирование и отправка сообщения осуществляется с использованием web-интерфейса;
- 2) на мобильном устройстве по каждому сообщению можно идентифицировать отправителя;
- 3) отправленный текст должен представлять собой html-страницу, рассылка которой осуществляется в соответствии с определенным графиком;
- 4) вся информация по совершенным транзакциям должна храниться в реляционной базе данных;
- 5) легкая масштабируемость системы для новых проектов и заказчиков.

Алгоритм работы данной системы push-уведомлений с появлением нового функционала существенно расширился. Помимо ранее рассмотренных этапов он включает в себя следующие новые позиции:

- после регистрации мобильного устройства в системе пользователю предоставляется возможность определения наиболее интересных тем для подписки, а также планируемой к получению информации. В данном случае пользователь через предоставляемый интерфейс приложения осуществляет все необходимые процедуры по подписке;

- клиент (заказчик системы рассылки) через предлагаемый веб-интерфейс осуществляет управление рассылкой. Непосредственно через веб-браузер происходит выбор тем, устанавливается график рассылки для каждого сообщения, вводится текст сообщения с использованием html-тегов;

- вся информация о совершенных действиях клиента и пользователя поступает на хранение в реляционную базу данных.

Дальнейшее развитие системы будет включать в себя открытый API для серверной компоненты. Это позволит создавать различные клиентские приложения на общей базе [2].

А. Описание клиентской части

Клиентская часть приложения для пользователей мобильных устройств написана под операционную систему Android версии 2.3.3. (и выше) с использованием среды разработки Eclipse.

При работе с приложением необходимо пройти процедуру регистрации для предоставления идентификационной добровольной информации. Данный тип информации используется исключительно для возможности обращения при взаимодействии с пользователем и получении общей статистики по различным группам. К предлагаемым для заполнения полям относятся: имя, фамилия, пол, возраст, электронная почта. Пример регистрационной формы приведен ниже на рис. 3. Отметим также, что открытый API как раз и предназначен, в том числе, для произвольного описания собираемой информации.

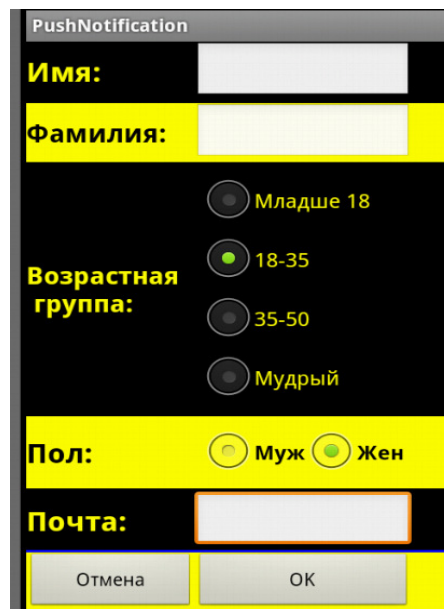


Рис. 3 Регистрационная форма

Данная идентификационная информация сохраняется в базе данных «PushNotification» SQLite на мобильном устройстве, а также дублируется в централизованной базе данных PostgreSQL, используемой сервером приложения для осуществления процесса отправки уведомлений.

В случае успешной регистрации и подписки, при отправлении уведомления на мобильное приложение можно просматривать сообщения.

В описываемом варианте продемонстрированы две условные компании (*Kremlin* и *WhiteHouse*), от имени которых осуществлялась рассылка по различным темам. С мобильного приложения подписка была осуществлена на условные темы *Thema 1* и *Thema 5* (рис. 4)

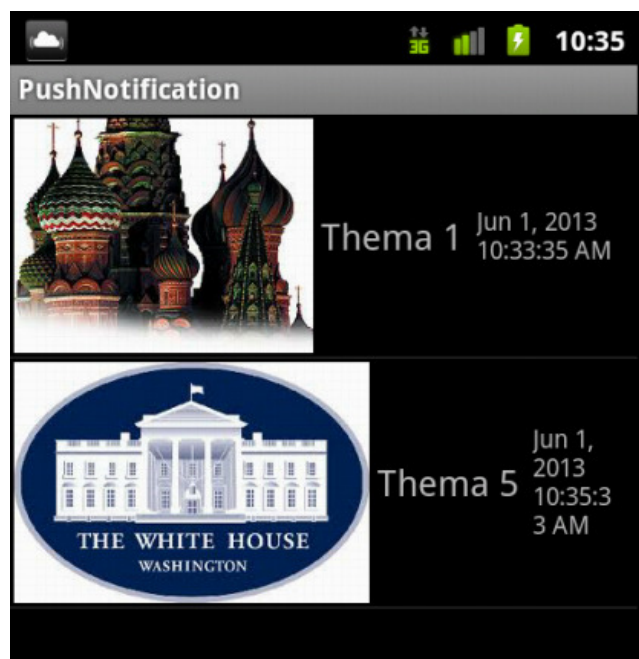


Рис. 4 Подписки

При работе с приложением пользователю предоставляется возможность совершать подписку на

определенные темы, просматривать сообщения, осуществлять отказ от регистрации в системе.

Для удобства пользователя также разработано меню для возможности осуществления удаления всех ранее полученных сообщений, отказа от подписки в системе рассылки сообщений.

Приложение также позволяет осуществлять сбор анонимной статистики по возрасту, полу, времени открытия сообщения.

V. Описание серверной части

В качестве базовых технологий для разработки архитектуры серверной части системы push-уведомлений выбрана платформа JavaEE 6 (java enterprise edition). В текущей реализации используется стабильный и достаточно эффективный сервер приложений с открытым исходным кодом GlassFish 3.1.2, позволяющий реализовать все возможности данной платформы. Для общения с GCM-службой используется стандартный HTTPS протокол.

Ядро серверной части системы реализовано в качестве Java приложения с back-end базой данных PostgreSQL, в которой хранятся списки пользователей, клиентов, темы подписки и отправленные сообщения.

Общая схема архитектуры серверной части представлена ниже. Она включает в себя две реализованные подсистемы: модуль обработки данных, которые поступают на сервер, а также используются для формирования статистики, и модуль управления расписанием и рассылкой уведомлений. Модули, указанные в правой части (выделены красным) планируются к реализации.

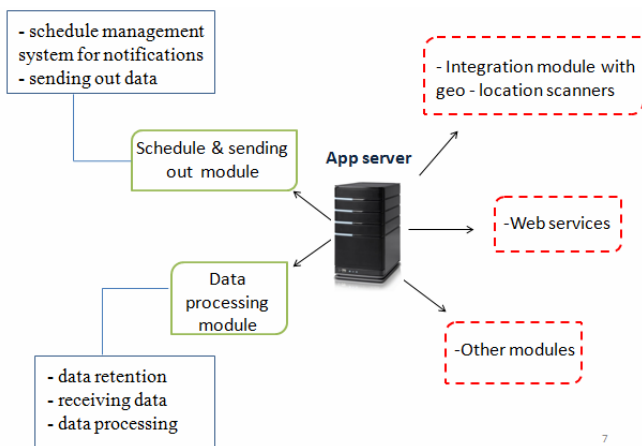


Рис. 5 Архитектура back-end

Для отправки сообщения приложение формирует POST запрос по адресу <https://android.googleapis.com/gcm/send>. В HTTP заголовке в обязательном порядке указывается поле авторизации, содержащее маркер отправителя, и тип контента (Content-Type: *application/json*).

IV. МОДЕЛИ ПРИМЕНЕНИЯ СИСТЕМЫ PUSH-УВЕДОМЛЕНИЙ

В настоящее время рост конкуренции на рынке заставляет большинство продавцов развивать различные программы лояльности среди своих клиентов. В большинстве случаев такой маркетинговый ход позволяет сохранить наиболее ценных клиентов.

Среди таких подходов активно развивается система глубокого анализа предпочтений клиента, основанная на отслеживании их поведения (например, в торговых сетях) для формирования уникального рыночного предложения. Все передвижения клиента фиксируются, на основе этого составляется карта маршрута со временем его прохождения. По полученным картам определяется, например, в каких отделах клиент задерживался; путь, по которому он проходит до необходимого товара и т.д.

Среди компаний, которые предоставляют возможность внедрить такую систему, являются Navizion, WalkBase, Cisco [4] [5] [6].

Основной принцип работы системы заключается в том, что сетевые узлы (сканеры) в фоновом режиме могут определять присутствие мобильных устройств с включенным Wi-Fi модулем. При этом нет необходимости устанавливать специальные приложения или присоединяться к какой-либо Wi-Fi сети. Это так называемый пассивный Wi-Fi мониторинг. Все сканеры размещаются, как правило, на определенном расстоянии друг от друга с целью покрытия максимальной территории. Естественно, определить какую-либо информацию о мобильном абоненте таким образом невозможно, но можно использовать доступный при Wi-Fi мониторинге MAC-адрес. Отметим, что адрес используется только для повторной идентификации клиентов и может быть заменен на некоторый хэш-код для сохранения приватности.

Собственно, на использовании MAC-адреса и построена идея интеграции сервера уведомлений. Подписка на новости (рассылки) осуществляется из мобильного приложения. Следовательно, доступен MAC-адрес клиента. Этот адрес (заменяющий его хэш-код) может быть сохранен вместе с идентификатором подписчика. Далее, при пассивном мониторинге, можно просто сравнивать найденный адреса с базой подписчиков (сравнивать хэш-коды адресов). Пассивный мониторинг в этом случае работает как автоматический check-in (отметка о присутствии) [7].

В совокупности с предложенной системой push-уведомлений заказчик существенно расширяет свои возможности по работе с клиентами за счет способности идентифицировать пользователя по МАК адресу Wi-Fi-модуля в мобильном устройстве. Один из возможных подходов для построения модели применения – это расширение системы SpotEx [8]. Бизнес-правила по определению доступной мобильному абоненту информации могут быть расширены информацией о его физическом присутствии. Более подробно такие системы рассмотрены в [9].

Таким образом, заказчик может осуществлять таргетированную рассылку пользователям, которые

непосредственно осуществляют покупку товаров в определенные моменты времени.

В этом случае необходимо развивать графическую составляющую при работе с системой на клиентской и серверной частях приложений. В рамках такого подхода интересным решением является персонализация приложений с логотипом заказчика. В связи с этим важно рассмотреть общую архитектуру системы, которая позволила бы быстро (например, без дополнительной компиляции исходного кода) и самостоятельно силами клиента изменять настройки приложения и его графический интерфейс.

Некоммерческим вариантом может быть рассмотрена система, позволяющая отслеживать посещаемость и успеваемость ученика в школе или на отдельных специальных дополнительных мероприятиях. В данном случае возникает необходимость в двух клиентских приложениях для родителя и ученика, на которые будет приходить информация об изменениях в расписании, о домашних заданиях и т.д.

Следующая модель применения предполагает использование текущего варианта системы с технологией веб-сервисов, когда программа идентифицируется веб-адресом со стандартным интерфейсом.

Такой подход позволяет уже рассматривать разработанную систему как самостоятельное «облачное решение», которое берет полное управление по хранению, управлению всей необходимой информацией и рассылке уведомлений клиентам.

V. ЗАКЛЮЧЕНИЕ

В настоящей работе была предложена система push-уведомлений с использованием открытых технологий компании Google Inc. Архитектура реализована на языке программирования Java.

Данная система push-уведомлений позволяет вести учет по различным темам подписки в разрезе компаний, клиентов, а также осуществлять рассылку уведомлений с учетом расписания. Клиентское приложение под Android позволяет пользователям осуществлять регистрацию в системе, производить подписку к интересующим темам, непосредственно читать сообщения, поступающие в виде html-текста, а также формировать статистику для модуля аналитики.

Кроме того, в работе рассмотрены различные модели применения разработанной системы push-уведомлений в реальности.

БИБЛИОГРАФИЯ

- [1] Google Cloud Messaging for Android <http://developer.android.com/google/gcm/gs.html>
- [2] Левин Ю. "Разработка веб-интерфейса для сервера уведомлений". International Journal of Open Information Technologies. – 2013. – Т. 1. – №. 2. – С. 12-16.
- [3] Push Service https://developer.blackberry.com/develop/platform_services/push_overview.html
- [4] D.Namiot and M. Sneps-Sneppé. "Geofence and Network Proximity." arXiv preprint arXiv:1303.5943 (2013).
- [5] D. Namiot and M. Sneps-Sneppé. "Mobile Services and Network Proximity." arXiv preprint arXiv:1305.4348 (2013).
- [6] M. Sneps-Sneppé and D. Namiot "Spotique: A New Approach to Local Messaging", 11th International Conference, WWIC 2013, St. Petersburg, Russia, June 5-7, 2013, pp. 192-203 DOI: 10.1007/978-3-642-38401-1_15
- [7] D.Namiot and M.Sneps-Sneppé. "Advertising in Social Networks: Business-oriented Check-ins". In AFIN 2011, The Third International Conference on Advances in Future Internet, 2011, pp. 20-24.
- [8] Namiot, D., & Sneps-Sneppé, M. (2012, October). Context-aware data discovery. In Intelligence in Next Generation Networks (ICIN), 2012 16th International Conference on (pp. 134-141). IEEE. DOI: 10.1109/ICIN.2012.6376016
- [9] D. Namiot. "Local Area Messaging for Smartphones". International Journal of Open Information Technologies, vol. 1, N.2, pp. 8-11, 2013