

УДК 004.522

**РАЗРАБОТКА НАВЫКА НА ПЛАТФОРМЕ
ЯНДЕКС.ДИАЛОГИ ДЛЯ УПРАВЛЕНИЯ
УСТРОЙСТВАМИ УМНОГО ДОМА**

Беломутов Георгий Валентинович

магистрант

Гомельский государственный технический университет

им. П.О. Сухого

(Беларусь, г. Гомель)

Мурашко Игорь Александрович

доктор технических наук, профессор

Гомельский государственный технический университет

им. П.О. Сухого

(Беларусь, г. Гомель)

В статье рассматривается архитектура программных компонентов для управления устройствами умного дома, принципы взаимодействия пользователя с данным программным решением и дальнейшие направления разработки по созданию бесшовного процесса развертывания на облачных серверах.

Ключевые слова: умный дом, облачные сервисы, программная архитектура, интеграция

**SKILL CREATION BASED ON THE YANDEX.DIALOGUES
PLATFORM FOR CONTROLLING DEVICES OF A SMART
HOUSE**

Belamutau Heorhi Valentioovich

Undergraduate

Gomel State Technical University. P.O. Sukhoy

(Belarus, Gomel)

Murashko Igor Aleksandrovich

Dr. Tech. Science, Professor

Gomel State Technical University. P.O. Sukhoy

(Belarus, Gomel)

The article considers the architecture of the program components required for smart house devices managing, user interaction principles with a created solution and further development steps for seamless deployment process to the cloud services.

Keywords: smart house, cloud services, program architecture, integration

В то время как устройства умного дома становятся все доступнее, появляется серьезная проблема фрагментированности этих устройств. Так как данный сегмент устройств еще только формируется, нет общего видения по каким протоколам должны взаимодействовать устройства между собой. Это приводит к тому что при обилии центров управления умными домами и различных «смарт-устройств» возникает проблема что один центр работает с одними устройствами но не способен работать с другими. И наоборот, устройство умного дома может работать не со всеми центрами управления.

Однако зачастую разработчики делают прикладной интерфейс, API, открытым для третьих лиц, что позволяет расширять его функционал всем желающим. С одной стороны это приводит к еще большей фрагментации рынка, и перекладывания ответственности за интеграцию устройств на третьи лица. С другой – открывает возможность намного большему количеству разработчиков разработать интеграцию по своему видению. Таким образом, обилие различных вариантов решения задачи способствует нахождению единого подхода к решению задачи.

Поставлена задача в интеграции устройств умного дома на базе платформы SmartThings для голосового управления с помощью центра управления умного дома Яндекс.Станция.

Яндекс.Станция это умная колонка со встроенным голосовым помощником «Алиса». Помимо мультимедийных возможностей, она оснащена Wi-Fi модулем и обрабатывает большинство голосовых команд на удаленном сервере.

Для расширения функционала голосового помощника «Алиса», Яндекс предоставляет сторонним разработчикам осуществлять разработку т.н. навыков на базе платформы Яндекс.Диалоги.

В рамках реализации поставленной задачи был разработан навык умного дома для данной системы на платформе .NET. Использование готовых библиотек (пакетов NuGet) позволяет избавиться от повторного решения тривиальных задач, для которых и использовать существующее решение. Таким образом возможно сосредоточиться на разработке решения для конкретной поставленной задачи [2]. Схема взаимодействия пользователя с устройством умного дома с помощью навыка на платформе Яндекс.Диалоги изображена на рисунке 1. Задача заключается в реализации Adapter API который будет отвечать за передачу необходимой команды в Provider IOT API с которым тот в свою очередь взаимодействует с устройством умного дома.

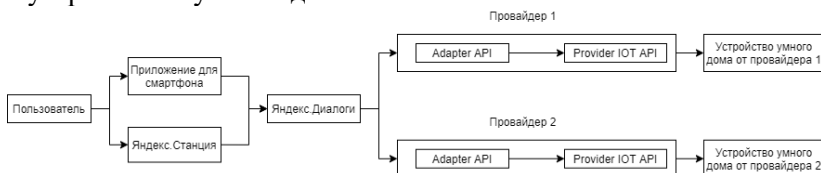


Рисунок 1 – Общая схема работы навыка умного дома

Для использования в жизненном цикле навыка умного дома, Adapter API должен соответствовать определенному формату платформу, указанному в документации разработчика Яндекс [3]. Согласно формату, Adapter API должен работать по протоколу https и по поддерживать соединяющие методы работы:

1. *HEAD* <server-address>/v1.0/ – Проверка доступности Endpoint URL провайдера.

2. *POST* <server-address>/v1.0/user/unlink – Оповещает о разъединении аккаунта пользователя на стороне провайдера и аккаунта пользователя на стороне Яндекса (удаление связки аккаунтов, которое инициировал пользователь).

3. *GET* `<server-address>/v1.0/user/devices` –
Запрашивает информацию о поддерживаемых устройствах пользователя.

4. *GET* `<server-address>/v1.0/user/devices/query` –
Запрашивает информацию о текущем состоянии устройств пользователя.

5. *POST* `<server-address>/v1.0/user/devices/action` –
Изменяет текущее состояние устройств пользователя.

Форматы запросов и ответов так же описаны в документации. Эти данные передаются Adapter API в формате JSON. Для сериализации ответов и десериализации запросов используется NuGet пакет `Newtonsoft.JSON`.

Для возможности использования данного навыка на физических устройствах (Яндекс.Станция, приложение на смартфоне), необходима его публикация из консоли разработчика. После этого навык становится доступен в каталоге навыка для всех желающих, или только для разработчика навыка (если навык был отмечен как приватный). Так или иначе, это требует размещения Adapter API на удаленном сервере, глобально доступным в сети Internet.

Ввиду того что Adapter API по своей природе не требует постоянной активности, и используется только при отправке команд от платформы Яндекс.Диалоги, нет нужды заводить полноценную машину, доступную круглосуточно. Для таких целей подходят так называемые Lambda функции. Которые не требуют наличия выделенного сервера, и ответственность за запуск берут на себя облачные сервисы. В качестве таких сервисов использовались Amazon Web Services – AWS. AWS предоставляет множество различных сервисов для различных нужд. Для реализации поставленной задачи, а именно Adapter API, так же необходимо связать разработанные Lambda функции для представления в виде REST API. Для этого AWS предоставляет технологию Amazon API Gateway. После успешного развертывания Lambda функций, обернутых через API Gateway, на выходе имеется выделенный адрес сервера, при

обращения к методам которого, запускаются на выполнение Lambda функции. Такой подход не требует оплаты аренды сервера, и счет выставляется только за непосредственное использование методов [1].

Разработанный Adapter API, интегрированный в платформу Яндекс.Диалоги позволяет производить управление телевизионным устройством на базе платформы SmartThings с помощью голосовых команд или через приложение для смартфонов. В дальнейшем планируется наладить процесс развертывания Adapter API на сервера AWS с помощью инструмента Terraform.

Список литературы

1. AWS Lambda Documentation документация [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://docs.aws.amazon.com/lambda/>
2. Троелсен, Э. Язык программирования C# 5.0 и платформа .NET 4.5, 6-е изд.: Пер. с англ. / Э. Троелсен. – М.: ООО «И.Д. Вильямс», 2015. – 1312 с. – Парал. тит. англ.
3. Яндекс.Диалоги документация [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://yandex.ru/dev/dialogs/alice/doc/smart-home/reference/resources-docpage/>

© Беломутов Г.В., Мурашко И.А., 2020