# Навык Алисы – ассистента Яндекса на языке PHP с использованием ООП и MVC

Для платформы Яндекс.Диалоги я разработал навык «Монетизация компьютера» на языке программирования PHP. Программное обеспечение построено по правилам архитектуры проектирования паттерна MVC (Model-View-Controller). Файлы с данными и скриптами организованы и структурированы по принципам парадигмы проектирования – ООП (объектно-ориентированного проектирования). Эта программа для Алисы, является продуктом рефакторинга процедурного кода **приложения Алисы «Заработок в интернете - финансово-выгодный».**

Содержание статьи.

[Навык Алисы – ассистента Яндекса на языке PHP с использованием ООП и MVC](#_Toc2922873)

[Навык Алисы 1](#_Toc2922874)

[Кто такая Алиса 1](#_Toc2922875)

[Навык 1](#_Toc2922876)

[Архитектура приложения – шаблон MVC 2](#_Toc2922877)

[Модель 3](#_Toc2922878)

[Контроллер 3](#_Toc2922879)

[Представление 3](#_Toc2922880)

[Точка входа 3](#_Toc2922881)

[Особенности архитектуры 4](#_Toc2922882)

[Использование ООП 4](#_Toc2922883)

[Каталог «c» 4](#_Toc2922884)

[Шаблоны каталога «v» 5](#_Toc2922885)

[Файлы каталога «m» 5](#_Toc2922886)

[ООП в цикле обработки запроса Яндекса к навыку 5](#_Toc2922887)

[Язык UML 6](#_Toc2922888)

[Условные обозначения UML 6](#_Toc2922890)

[Как использовать картинки в навыках Алисы 6](#_Toc2922891)

[Загрузка картинки на Яндекс.Диалоги cURL-ом 7](#_Toc2922892)

[Как для загрузки картинки использовать cURL в Windows 7](#_Toc2922893)

[Размеры картинок для навыка Алисы 8](#_Toc2922894)

Исходники навыка Алисы с использованием ООП и MVC

[Запуск PhpStorm из меню OpenServer 8](#_Toc2922895)

[Поднять сервер Open Server из терминала PhpStorm 9](#_Toc2922896)

[Алгоритм действий по поднятию сервера из терминала 9](#_Toc2922897)

## Навык Алисы

### Кто такая Алиса

«Привет, Алиса», или «Слушай, Алиса» - такими речевыми командами вызывается голосовой помощник Яндекса. Обязательным условием выполнения команды является наличие установленного бота-ассистента на устройстве. Робот поддерживается телефонами с системами Андроид или iOS, компьютерами с Windows, другими гаджетами (например, Yandex.Станция, Irbis A, DEXP Smartbox).

Для функционирования помощника на компьютере надо установить Яндекс.Браузер, или же скачать и развернуть отдельно приложение Алиса со страницы https://yandex.ru/support/browser/alice/install.html. Проверено: в Windows 10 голосовой ассистент поисковика работает при открытом браузере edge. Поэтому не обязательно ставить яндексовский обозреватель интернета Яндекс.Браузер.

Без активированного браузера бот не работает.

На телефон приложение скачивается из магазина приложений.

### Навык

Навык – диалог между одним пользователем интернета и другим, зарегистрированным пользователем российского поисковика Яндекс и разработавшим функционал общения. Диалог осуществляется на платформе Алисы - Яндекс.Диалоги. Общение происходит либо только голосом, либо только с помощью клавиатуры и экрана, либо и тем и другим одновременно.

Разработчик, написав новое приложение, отправляет его на модерацию, после успешного прохождения её оно появляется в каталоге навыков и становится доступным всем пользователям. Приватные навыки отсутствуют в каталоге, но они все равно доступны всем желающим, которые знают, или сумеют подобрать их активационные имена.

#### Модерация навыка

Требования к приложению опубликованы в документации на странице https://tech.yandex.ru/dialogs/alice/doc/requirements-docpage/. При первичном опубликовании проекта, модерацию пройти довольно легко. Думаю, такова политика Яндекс.Диалогов – принять как можно больше навыков от разработчиков, тем самым обучить нейросеть искусственного интеллекта Алисы.

Опубликованный проект можно редактировать. Если в результате редактирования изменяются настройки, то программу придется отправить на модерацию повторно. А вот повторную модерацию пройти довольно трудно. Требования к настройкам ужесточаются. В процессе постоянной эволюции проекта они добавляются новыми правилами. Если при первичной модерации название навыка, или активационные имена легко прошли модерацию, то при повторной проверке зачастую они обрастают замечаниями от модераторов.

Сейчас проверку производят специально обученные люди – модераторы. В дальнейшем это будут делать роботы.

#### Трудности функционирования на Windows

Сразу после запуска проекта у меня всё прекрасно работало на планшете с Windows 10. При обращении к приложению голосом или набором на клавиатуре получал ответ на экране планшета и голосом от динамиков. Чтобы Алиса ответила голосом, требуется настроить генерацию речи в формате ответа "tts" (https://tech.yandex.ru/dialogs/alice/doc/protocol-docpage/).

После очередного расширения функционала в октябре 2018 года, а именно возможности выбора мужского голоса в ответах «Алисы», навыки перестали отвечать голосом на устройствах с Windows.

Этот косяк разработчикам поисковика удалось исправить только к середине января 2019 года.

#### Проверка работоспособности навыка

Платформа диалогов ежеминутно проверяет готовность к ответу приложения, обращаясь к навыку командой *"ping"*. Таким образом, проверяется, живо ли приложение.

JSON обращения:

"request": {

"command": "ping",

"original\_utterance": "ping",

…

}

Кроме этого сервис приложения на сервере должен отвечать Диалогам в течение *1,5 секунды*.

При отрицательных результатах таких проверок навык деактивируется, автор разработки получает уведомление на почту. Если автор не принимает мер, то разработка удаляется из каталога.

## Архитектура приложения – шаблон MVC

Шаблон проектирования – это пакет повторно используемых решений проблем. MVC – один из таких паттернов. Он позволяет разделить данные, представление и обработку действий пользователя на 3 отдельных модуля. Приложение навыка разделено на модули:

MODEL (модель) – работа с базой данных,

VIEW (представление) - шаблоны вывода экрана,

CONTROLLER (контроллер) - код взаимодействия с пользователем.

Цели разбиения программы на отдельные модули:

повторное использование кода,

легкое ориентирование,

возможность быстрого расширения и изменения.

Рисунок. Схема MVC для навыка Алисы.

### Модель

Модель читает необработанные данные после HTTP-заголовков из тела запроса, отправленного Диалогами, принимает закодированную в JSON строку и преобразует ее в переменную PHP. Здесь же происходит соединение с базой данных, получение данных из базы, запись в базу информации об ошибках посетителя, возврат результатов регулярных выражений.

«MODEL» является библиотекой методов для эффективной работы с базой данных.

### Контроллер

В моем приложении Контроллер является обработчиком запросов. Модуль «CONTROLLER» получает от «MODEL» раскодированные элементы запроса Диалогов: "request" - "command", "original\_utterance", "markup", "payload"; "session" - "message\_id", session\_id, "user\_id"… И другие. В зависимости от вида запроса посетителя подготавливает ему ответ.

Если для ответа требуются какие-то материалы из базы данных, они берутся из Модели.

Нужное изображение (вид) выбирается из Представления (VIEW).

Готовый ответ кодируется в JSON.

### Представление

«VIEW» содержит шаблоны пользовательского интерфейса.

Пользовательский интерфейс (UI — англ. user interface) — интерфейс, обеспечивающий передачу информации между пользователем-человеком и программно-аппаратными компонентами компьютерной системы. <https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F>

Представление состоит из нескольких пассивных (push) шаблонов, состоящих из сведений о структуре вида выводимого экрана устройства, голосовых ответов.

### Точка входа

Единая точка входа приложения – индексный файл *index.php*. После команды посетителя 'запусти навык монетизация компьютера' платформа Яндекс.Диалоги обращается POST-запросом на Webhook URL приложения к индексному файлу JSON строкой.

Индексный файл подключает файлы модуля «CONTROLLER», создает требуемый контроллер, и управление передается ему.

CONTROLLER, прежде всего, обращается к Модели за данными. Готовит ответ. Затем получает нужный шаблон из Представления для визуального и звукового оформления ответа. Отправляет ответ.

### Особенности архитектуры

VIEW и MODEL ничего не знают друг о друге и не связаны между собой. CONTROLLER берет данные из MODEL и подставляет в VIEW.

Шаблон MVC с «толстым» контроллером и «тонкими» моделью и представлением. Пользователь управляет приложением через контроллер. Он не может напрямую обращаться ни к «MODEL», ни к «VIEW» и видит экран, слышит голосовое сообщение, сформированные «CONTROLLER»-ом.

Применена трехуровневая архитектура «клиент–сервер–база данных», раскритикованная во многих источниках разработчиков. Но в данной разработке работает и легко изменяется.

## Использование ООП

Для объединения ООП с MVC в моем приложении навыка Алисы «Монетизация компьютера» выполнено следующее:

Файлы Controller, Model, View разнесены в три отдельных каталога, соответственно – «c», «m», «v».

Каждому выводу экрана (отдельному представлению) будет соответствовать свой controller.

Каждый controller представлен отдельным файлом, в котором объявлен class, инкапсулирующий логику работы модуля Controller и предоставляющий методы для обработки запросов от Яндекс.Диалогов (пользователей). Классы построены в виде иерархии с наследованием.

Модели системы реализованы в виде двух классов.

Представления реализованы в виде классов–наследников от одного родительского class-а.

Приложение навыка находится на сервере. Яндекс.Диалоги обмениваются с ним обращениями-ответами по протоколу HTTPS по методу webhook через специальную точку входа – файл *index.php*.

В индексном файле в первых строках происходит подключение class-ов модуля Controller. Создается объект одного из class-ов. Дальше проверяется соответствие request пользователя свойствам этого объекта. В зависимости от совпадения request и определенного свойства создается нужный для данного request экземпляр контроллера. И с помощью полиморфизма дальнейшее управление приложением передается этому контроллеру.

### Каталог «c»

Как сообщалось **выше** каталог «c» содержит файлы модуля Controller.

Классы контроллеров служат для обеспечения обработки request пользователей, поступающих через платформу – получения параметров входных запросов и подготовки результирующей JSON строки. Для выполнения этой работы они задействуют классы модели и представления.

Рисунок. Диаграмма иерархии классов модуля CONTROLLER.

Данная диаграмма выполнена на языке UML. Об условных обозначениях описательного **языка UML** здесь.

Родительский class *Controller* инкапсулирует логику для любого контроллера и содержит общие элементы модуля. В частности - метод *Request()*, ответственный за обработку request от ассистента Яндекса и формирование обратно ответа. В его теле абстрактные методы *Input()* – обработка запроса и *Output()* – генерация ответа. Эти функции переопределяются в дочерних классах.

Есть еще вспомогательный метод *IsButtons($buttons)* проверяющий, что переменная *buttons* является двумерным массивом.

Базовый controller *C-Base* отвечает за кодирование ответа в строку JSON, в нем определены базовые для навыка Алисы переменные PHP.

*C\_View*, *C\_Help*, *C\_Card*, *C\_Gallery* – контроллеры конкретных ответов пользователю голосового ассистента во время диалога с навыком. В них переопределяются методы *Input()* и *Output()*, но управление передается *C-Base*. Функция *Input()* сначала вызывает родительский метод, затем в зависимости от полученного request пользователя инициализирует нужные переменные PHP. Функция *Output()* начинается с подстановки этих подготовленных переменных в конкретный, закрепленный для данного контроллера, шаблон модуля «VIEW». После чего вызывается родительский метод *Output()*.

### Шаблоны каталога «v»

Рисунок. Диаграмма иерархии классов модуля VIEW.

Модуль «VIEW» содержит родительский class и его классы-наследники. Родительский class - это шаблон, в котором объявленная переменная PHP находится в конструкторе и является ассоциативным массивом. Ключи – это имена полей объекта JSON в запросе от Яндекс.Диалогов. Значения – это переменные, значения которых находятся в базовом контроллере *C-Base*.

Дочерние классы – это такие же шаблоны. Каждый дочерний class предназначен для конкретного controller-а: *Template\_resp* – для *C\_View* и *C\_Help*, *Template\_card* – для *C\_Card*, *Template\_gallery* – для *C\_Gallery*.

### Файлы каталога «m»

Модель состоит из класса соединения с базой данных и класса с функциями обработки запроса, получения сведений из базы данных, ввода ошибок пользователей, возврата результатов регулярных выражений.

### ООП в цикле обработки запроса Яндекса к навыку

Обработка запроса начинается в файле *index.php*. В этом документе приложения происходит выбор нужного контроллера. Выбранный controller вызывает метод *Request()*. С помощью этого метода запускаются 2 этапа обработки запроса:

адресование к модели для получения входных данных (Input());

генерация JSON (Output()).

Первый этап цикла происходит в такой последовательности:

обращение к базовому controller-у (*C\_Base*);

управление конкретным контроллером (*C\_View*, или *C\_Help*, или *C\_Card*, или *C\_Gallery*);

Второй этап – в обратной последовательности:

конкретный controller (*C\_View*, или *C\_Help*, или *C\_Card*, или *C\_Gallery*);

возврат к базовому controller-у (*C\_Base*).

На UML-диаграмме последовательностей обработка запроса выглядит так, как показано на рисунке Цикл обработки request Яндекс.Диалогов к навыку.

Рисунок. Цикл обработки request Яндекс.Диалогов к навыку.

## Язык UML

Аббревиатура UML декодируется как Unified Modeling Language и переводится – унифицированный язык моделирования. Язык UML применяется в программировании для работы с объектно-ориентированными технологиями. Этот визуальный язык использует графические и буквенные элементы для создания моделей программ.

Диаграмма классов (Class Diagrams) используется на этапе проектирования для представления взаимодействия отдельных классов системы.

Диаграмма последовательностей (на английском языке - Sequence Diagram) – разновидность диаграммы взаимодействий. В отличие от диаграммы классов, отражающей статическое отношение классов, рассказывает о взаимодействии в течение времени жизни сущностей и действующих лиц. Такую диаграмму необходимо изучать сверху вниз.

### Условные обозначения UML

Прямоугольник соответствует классу.

Стрелка от прямоугольника показывает отношения наследования и обозначает, что другие классы являются его потомками. Направление стрелки указывает на родителя.

В прямоугольнике класса описываются имя, методы, члены, свойства class-а.

**Имя** class-а выделено полужирным шрифтом в верхней части прямоугольника. Дополнительное выделение имени class-а курсивом указывает, что этот class является ***абстрактным***, то есть не имеет экземпляров, применяется только для определения интерфейса своих классов-потомков.

Прямым шрифтом – методы. Курсивом – *свойства* class-а. После двоеточия «:» - тип.

Условные обозначения UML для описания уровня доступа:

**+** (плюс) - public (открытый),

**#** (номер) – protected (защищенный),

**-** (минус) – private (закрытый).

Вертикальная прерывистая линия – течение времени жизни.

## Как использовать картинки в навыках Алисы

Платформа Диалогов предоставила возможность использовать иллюстрации в навыках. Иллюстрации можно использовать в двух вариантах:

одиночное изображение,

изображения в списке ответов.

Изображение требуется загрузить на сервис голосового ассистента и применять выданные идентификатор и URL изображения в разработке.

Для различных вариантов загрузки выделены разные API (на английском языке - Application Programming Interface). В частности, чтобы залить картинку из интернета, позволено воспользоваться cURL следующего API:

curl \

-H 'Authorization: OAuth <OAuth-токен>' \

-H 'Content-Type: application/json' \

-X POST \

-d '{ "url": "<адрес изображения>" }' \

'https://dialogs.yandex.net/api/v1/skills/<идентификатор навыка>/images'

### Загрузка картинки на Яндекс.Диалоги cURL-ом

cURL – программа для взаимодействия с серверами, переноса данных, обмена информацией между сайтами. Если ее нет на вашем компьютере, то придется скачать архив, подходящей версии для используемой операционной системы с официального сайта https://curl.haxx.se/download.html. Рекомендую воспользоваться опцией «curl Download Wizard». В Интернете легко найти мастеров (например, ресурс https://o7planning.org/ru/11617/installing-curl-on-windows) определения нужной версии для загрузки этой утилиты. Самому сложно в этом разобраться, т.к. утилита кроссплатформенная с богатым выбором версий.

Программа не требует установки, надо просто распаковать папку в любое место. Вес программы небольшой: например, *curl-7.62.0-win32-mingw* занимает 6,36 МБ.

Для того чтобы загрузить иллюстрацию по API, скачивать *libcurl* - многопротокольную библиотеку передачи файлов не требуется.

### Как для загрузки картинки использовать cURL в Windows

Я использовал cURL в командной строке с правами администратора из-под Windows 10.

Предоставленный Яндексом API для загрузки картинок вызвал у меня много трудностей. Прежде всего, пришлось в аргументах коротких опций исключить одинарные кавычки, т.к. Windows их не поддерживает. При их использовании ответы курла выкидывали множество ошибок. Заменил их двойными кавычками.

Там где короткая опция «-d» записан аргумент JSON. Этот аргумент сначала я писал вообще без кавычек. Так не работало. Затем в двойных. Так рекомендовано в документации утилиты: «Чтобы сама строка содержала двойные кавычки, что обычно встречается, когда вы, например, хотите отправить строку JSON на сервер, вам может понадобиться использовать одинарные кавычки (кроме Windows, где одиночные кавычки работают не так), отправьте строку JSON { "name": "Darth" }.» Согласно такой рекомендации заключил аргумент в двойные знаки препинания. В результате курл снова сообщил об ошибках.

Рисунок. Курл сообщил об ошибках в использовании двойных знаков препинания для аргумента. D:\OpenServer\domains\shchej\images/curl.jpg

Перепробовал множество вариантов, пока не получил подсказку Alexey Kaliberda из сообщества разработчиков Яндекс.Диалогов в Telegram. Надо этот аргумент заключить в двойные кавычки, а элементы – в экранированные обратным слешем \ двойные кавычки. Адрес Яндекс.Диалогов с идентификатором навыка – вообще без кавычек.

Рисунок. Правильно оформленный курл для загрузки иллюстрации на Яндекс.Диалоги D:\OpenServer\domains\shchej\images/cur.jpg

В итоге правильный запрос для загрузки иллюстрации в терминале из-под Windows в следующей редакции:

curl -H "Authorization: OAuth <OAuth-токен>" -H "Content-Type: application/json" -X POST -d "{ \"url\": \"<адрес изображения>\" }" https://dialogs.yandex.net/api/v1/skills/<идентификатор навыка>/images

### Размеры картинок для навыка Алисы

В документации Яндекса (https://tech.yandex.ru/dialogs/alice/doc/resource-upload-docpage/) о размерах иллюстраций ничего не сообщается. Но есть рекомендации об их использовании в видео на 3:13 и 4:16 минутах Школы Алисы про выбор компонента для дизайна навыка (https://www.youtube.com/watch?v=YC7R0RCZ7gU&list=PLJ0wXoH5OsouSN8zS48jo7ATkQONYhWVm&index=2).

Необходимо следовать этим рекомендациям, чтобы изображения отображались на экране, так как задумано, а не были обрезаны по усмотрению Диалогов.

В моем приложении навыка Алисы на языке PHP за вывод картинок ответственны шаблоны представления *Template\_card* – для показа одиночной картинки (тип карточки *BigImage*) и *Template\_gallery* – для показа в виде галереи (тип карточки *ItemsList*). Это я вернулся к теме моего применения ООП для голосового ассистента.

## Исходники навыка Алисы с использованием ООП и MVC

Если кого-то заинтересовал мой опыт реализации навыка с применением ООП и MVC на языке PHP, то можно ознакомиться с исходниками кода.

В моей программе для голосового помощника использованы материалы и рекомендации о совместном использовании принципов MVC и ООП на языке программирования PHP из книги Дмитрия Ляпина, Александра Никитина «PHP – это просто. Начинаем с видеоуроков».

Дальнейшая информация не относится конкретно к данной теме статьи про использование MVC и ООП в навыке Алисы, а имеет общее применение в программировании. А расположил я эти разделы здесь, так как нижеописанными знаниями воспользовался при изготовлении приложения для Яндекса – навыка Алисы.

## Запуск PhpStorm из меню OpenServer

Вначале небольшое вступление об IDE PhpStorm. В интегрированной среде разработки (Integrated Development Environment) PhpStorm удобно писать код PHP. Сегодня это лучший продукт для редактирования кода.

Локальным сервером Open Server пользуюсь уже давно, после древнего Денвера. Считаю, что это удобная площадка для разработчика.

Иногда возникает потребность запуска IDE из меню OpenServer. Чтобы это делать, надо создать в настройках локального сервера закладку на PhpStorm и тогда можно запускать IDE через эту закладку.

Клик правой кнопкой мыши на значке «Настройки» OpenServer-а.

Вверху окна «Настройки» кликнуть «Закладки».

Назначить имя - например, PhpStorm, категория – можно IDE, в поле Выполнить записать путь до файла PhpStorm.exe – может быть <!-- noindex--> c:\program files\jetbrains\phpstorm 9.0.2\bin\PhpStorm.exe<!--/ noindex -->

Нажать кнопки Добавить – Сохранить.

Теперь IDE можно запустить через закладку в OpenServer.

## Поднять сервер Open Server из терминала PhpStorm

Не надо запускать терминал командной строки в операционной системе, он уже имеется в редакторе кода. В терминале (командная строка), который уже встроен в редактор кода, можно вводить команды и проверять их выполнение. Результаты выполненных команд отображаются в самом терминале и в браузере.

### Алгоритм действий по поднятию сервера из терминала

Запустить на своей машине Open Server (в трее будет зеленый флажок) и перейти в переменные среды следующим образом.

В Windows 10 в строке поиска забить «изменение системных переменных среды».

Откроется найденный вариант Панели управления «Изменение системных переменных среды». Кликнуть.

Выскочит окно «Свойства системы».

На вкладке «Дополнительно» – внизу окна «Переменные среды…» - переход на строку Path (выделить ее) – Изменить.

В окне «Изменить переменную среды» создать строку PHP 7 или другую версию, например <!-- noindex -->D:\OpenServer\modules\php\PHP-5.4<!--/ noindex -->, т.е. директорию до php Open Server-а.

Далее Ok Ok Ok.

Если Windows 7, или XP, то после этих манипуляций перезагрузиться, чтобы обновились переменные среды. Если восьмерка, или десятка, то просто закрыть-открыть консоль.

В PhpStorm включить консоль – внизу кнопка “Terminal”. После этих действий в консоли можно будет выполнить команду *php –v*. Команду надо ввести, а не копипастить. В результате выведется версия php. Значит все в порядке и можно выполнять команду на поднятие сервера.

Рисунок. Успешная установка php. (terminal.gif)

Перейти в нужную директорию (левой кнопкой мыши кликнуть директорию и зажатой кнопкой перетащить ее в терминал). В консоли написать строку: *php -S localhost:8080*. Выполнить, нажав Enter.

И вот сервер Open Server поднялся. В консоли скопировать строку сервера *http://localhost:8080* и открыть его в браузере. К получившейся адресной строке можно добавлять имена файлов. Например: *http://localhost:8080/controller.php*.

Всё. Результаты работы теперь можно контролировать в браузере и в терминале PhpStorm.