Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт математики и фундаментальной информатики СФУ

Кафедра математического анализа и дифференциальных уравнений

УТВЕРЖДАЮ Заведующий кафедрой

\_\_\_\_\_ Фроленков И.В.

подпись инициалы, фамилия

« \_\_\_\_\_» \_\_\_\_\_\_\_ 20 \_\_\_ г.

**ОТЧЕТ О ПРАКТИКЕ**

**ПО ПОЛУЧЕНИЮ ПРОФЕССИОНАЛЬНЫХ УМЕНИЙ И ОПЫТА ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ**

Место прохождения практики:

Институт математики и фундаментальной информатики

Тема практики: **Реализация логики работы Telegram бота, который уведомляет подписчика о резких изменениях стоимости акций на фондовом рынке.**

Руководитель\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата должность, ученая степень инициалы, фамилия

Студент \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

номер группы номер зачетной книжки подпись, дата инициалы, фамилия

Красноярск 2021

**Содержание**

|  |  |
| --- | --- |
| 1. Постановка задачи | 3 |
| 1.1 Формулировка | 3 |
| 1.2 Окружение | 4 |
| 2. Подзадача которую я решал | 4 |
| 2.1. Формулировка | 4 |
| 2.2. Создание бота | 4 |
| 2.3. Алгоритм решения | 4 |
| 2.3.1 Схема работы TelegramHelper | 4 |
| 2.3.2 Реализация метода getUpdates() | 5 |
| 2.3.3 Реализация метода sendMessage | 6 |
| 2.4 Взаимодействие с основной логикой программы | 7 |
| Список использованных источников | 8 |
| Приложение | 9 |

**1 Постановка задачи**

1.1 **Формулировка**

Целью данной работы является создание программы на языке программирования, предназначенный для разработки Web−приложений – Php7, с использованием объектно-ориентированного Фреймворка, который реализует логику Telegram бота для уведомления пользователя о резком изменении стоимости акциина фондовом рынке. Требуется написать программу, реализующую логику Telegram бота, который будет уведомлять пользователя о том, что:

* Стоимость акции выросла/упала за день на x процентов.
* Стоимость акции выросла/упала за неделю на x процентов.
* Стоимость акции пересекла скользящую средней цены.

Проект должен использовать публичное API Tinkoff инвестиции, Sandbox. Проект должен настраиваться через конфигурационный файл или через команды боту.

1.2 **Окружение**

Программа реализована в среде разработки PhpStorm на языке программирования PHP 7.4, с использованием Фреймворка, реализующего паттерн MVC – yii2.

**2 Подзадача которую я решал**

**2.1 Формулировка**

Написать модуль, работающий с Telegram API, позволяющий получать и отправлять сообщения в Telegram.

Создать телеграмм бота в непосредственно телеграмме.

**2.2** **Создание бота**

С помощью телеграмм бота @botfather был создан наш телеграмм бот, которого мы минимально настроили, выбрав ссылку и имя бота

**2.3 Алгоритм решения**

При решения данной задачи у нас было 2 варианта -webHook и getUpdates   
Мы решили выбрать второе, ибо в первом варианте нужно было иметь сервер с SSL сертификатом, что мы посчитали не очень удобно

**2.3.1** С**хема работы TelegramHelper**

Был создан TelegramHelper который реализовывает метод getComand, которая позволяет при каждом вызове ее узнать, есть ли новые сообщения от пользователя с помощью как раз метода getUpdates()

Рисунок 1 – Реализация метода getCommand

**2.3.2** **Реализация метода getUpdates()**

Метод getUpdates() реализует логику запроса, который работает с api телеграмма, который находится по ссылке <https://api.telegram.org/bot>   
Мы работаем с 2 методами: getUpdates и sendMessage   
Но сейчас подробнее о первом   
getUpdates - метод апи который получает все последние сообщения, но благодаря параметру “offset = -1” мы получаем последнее сообщение которое было когда либо отправлено.

После чего мы разбираем ответ, смотрим, что мы это сообщение еще не обработали (последнее мы запоминаем)

В зависимости от ответа апи мы формируем соответсвующий ответ

Рисунок 2 – Реализация метода getUpdates

**2.3.3** **Реализация метода sendMessage**

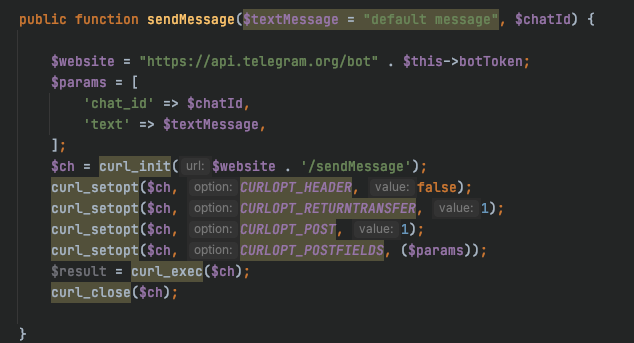
Данный метод еще более прост, ибо в нем нет дополнительной логики, а только отправление сообщения на переданный id телеграмма, с переданным текстом.

Рисунок 3 – Реализация метода sendMessage

**2.4** **Взаимодействие с основной логикой программы**

Для того чтоб пользоваться функционалом хелпера которого я реализовал, необходимо создать экземпляр класса, и у него вызывать нужный метод. Что и происходит в BotController в методе actionStart()

**Список использованных источников**

1.<https://core.telegram.org/api>

**Приложение**

<?php

namespace app\helpers;

class TelegramHelper

{

   private $lastMessageId = 0;

   private $botToken = *TELEGRAM\_TOKEN*;

   public function getCommand() {

       $message = $this->getUpdates();

       if ($message != null) {

           $matches = explode(' ', $message['text']);

           $command\_name = substr($matches[0], 1);

           $parameters = array\_slice($matches, 1);

           return ['id\_telegram' => $message['id\_telegram'], 'command\_name' => $command\_name, 'parameters' => $parameters];

       }

   }

   public function sendMessage($textMessage = "default message", $chatId) {

       $website = "https://api.telegram.org/bot" . $this->botToken;

       $params = [

           'chat\_id' => $chatId,

           'text' => $textMessage,

       ];

       $ch = curl\_init($website . '/sendMessage');

       curl\_setopt($ch, *CURLOPT\_HEADER*, false);

       curl\_setopt($ch, *CURLOPT\_RETURNTRANSFER*, 1);

       curl\_setopt($ch, *CURLOPT\_POST*, 1);

       curl\_setopt($ch, *CURLOPT\_POSTFIELDS*, ($params));

       $result = curl\_exec($ch);

       curl\_close($ch);

   }

   public function getUpdates() {

       $website = "https://api.telegram.org/bot" . $this->botToken;

       $ch = curl\_init($website . '/getUpdates?offset=-1');

       curl\_setopt($ch, *CURLOPT\_HEADER*, false);

       curl\_setopt($ch, *CURLOPT\_RETURNTRANSFER*, 1);

       curl\_setopt($ch, *CURLOPT\_POST*, 1);

       $result = curl\_exec($ch);

       curl\_close($ch);

       $result = json\_decode($result);

       if (!empty($result->result)) {

           $result = $result->result[0];

           if (!$this->isAlreadyViewed($result)) {

               $this->lastMessageId = $result->update\_id;

               if (isset($result->message)) {

                   return [

                       'id\_telegram' => $result->message->from->id,

                       'text' => $result->message->text

                   ];

               }

               elseif (isset($result->my\_chat\_member)) {

                   return [

                       'id\_telegram' => $result->my\_chat\_member->from->id,

                       'text' => '/stop',

                   ];

               }

           }

       }

   }

   public function isAlreadyViewed($result) {

       return $result->update\_id == $this->lastMessageId;

   }

}