

Отчет по чат-боту “Раписание”

Пункт 0

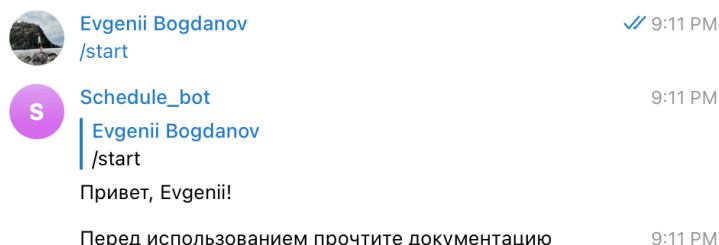
Сразу хочу сказать, что амбициозная идея сделать “идеальный парсер добавления календарных событий” оказалась достаточно затруднительной в силу ограниченного времени. Чат-бот работает не идеально, но есть потенциал для масштабирования. Так же вы могли заметить, что все классы и функции описаны в одном скрипте + пользуемся только одним состоянием - это нормально, интеллектуальные ресурсы были затрачены на реализацию :)

Пункт 1. Идея

Идея была в том, чтобы свободно писать о манипуляциях с добавлением/удалением/модификацией календарных уведомлений.

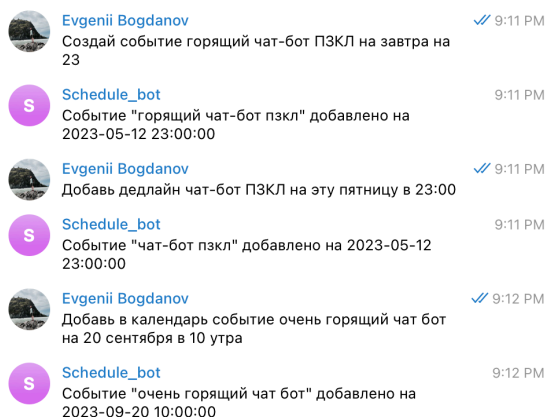
Пункт 2. Функции

Начинаем работу с ботом через команду `/start`. Я избавился от всех кнопок



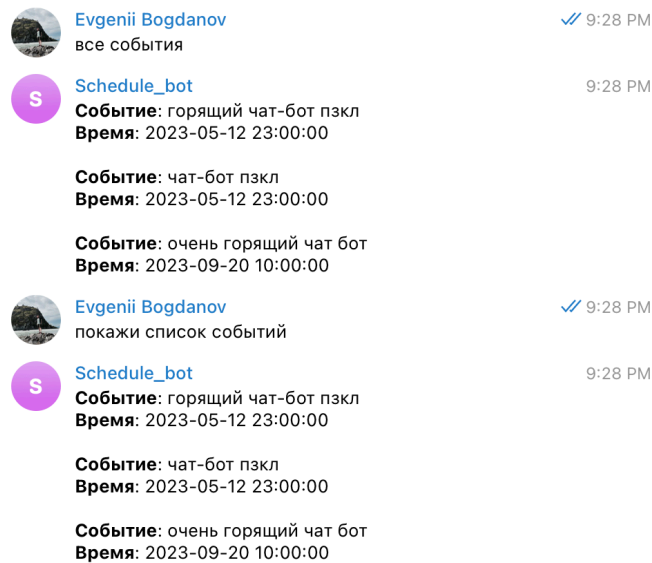
Итак, наш бот включает следующие функции: добавление событий, удаление событий, отображение текущих событий и корректировка ошибок. Разберем чуть подробнее каждый пункт

- **Добавление.** В идеале вводимое сообщение должно содержать “команду добавления”, “название события” и “время события”. Например “Добавь дедлайн чат-бот ПЗКЛ на эту пятницу в 23:00”.

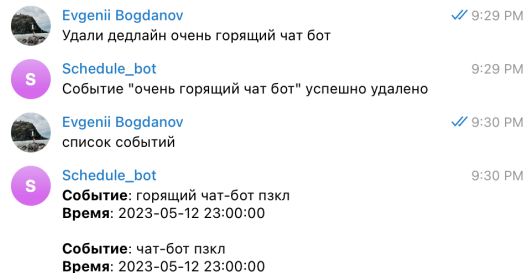


Как вы могли заметить, мы используем различные шаблоны для времени (их количество ограничено)











- Отображение событий. Тут та же история, только поддерживается одна команда. Например: “Все события” или “Покажи список событий”



- Удаление событий. Все то же самое, только мы вводим только “команду добавления” и “название события”. Например “Удали дедлайн очень горящий чат бот”



- Исправление ошибок. Поскольку мы исправляем ошибку, надо написать команду исправления и точно задать что именно мы хотим удалить и как правильно. Можно заметить, что мы не только исправили наше событие, но и “дообучились” распознавать новый паттерн команды добавления события.

	Evgenii Bogdanov	✓ 10:10 PM
	надо добавить событие календарный чат бот на завтра в 3 дня	
	Schedule_bot	10:10 PM
	Событие "надо календарный чат бот" добавлено на 2023-05-12 15:00:00	
	Evgenii Bogdanov	✓ 10:10 PM
	все события	
	Schedule_bot	10:10 PM
	Событие: чат-бот пзкл Время: 2023-05-12 23:00:00	
	Событие: горящий чат-бот пзкл Время: 2023-05-12 23:00:00	
	Событие: надо календарный чат бот Время: 2023-05-12 15:00:00	
	Evgenii Bogdanov	✓ 10:11 PM
	исправь ошибку для события "надо календарный чат бот" на "календарный чат бот"	
	Schedule_bot	10:11 PM
	Событие "надо календарный чат бот" успешно заменено на "календарный чат бот"	
	Evgenii Bogdanov	✓ 10:11 PM
	надо добавить событие исправленный календарный чат бот на завтра на 9 утра	
	Schedule_bot	10:11 PM
	Событие "исправленный календарный чат бот" добавлено на 2023-05-12 09:00:00	
	Evgenii Bogdanov	✓ 10:11 PM
	все события	
	Schedule_bot	10:11 PM
	Событие: чат-бот пзкл Время: 2023-05-12 23:00:00	
	Событие: горящий чат-бот пзкл Время: 2023-05-12 23:00:00	
	Событие: календарный чат бот Время: 2023-05-12 15:00:00	
	Событие: исправленный календарный чат бот Время: 2023-05-12 09:00:00	

Пункт 3. Реализация

• Команды

Тут все просто. У нас есть множество синонимов для каждой команды. Изначально мы лемматизируем все команды и ищем похожие среди всех множеств. По умолчанию используем tf-idf и косинусное расстояние, но так же можем использовать коэффициент Танимото и bm25 (пробуйте локально немного поменять код в 478й строчке на соответствующую функцию, название которой можно найти в строчках 196-198).

• Дата и время

Тут мы используем регулярные выражения для определения даты и времени. Обычные дата и время в разных форматах не поддерживаются, однако можно добавить функцию `parser` из модуля `dateutil`. Поддерживаемые даты: сегодня, завтра, на следующую пятницу, на этот понедельник, на 19 мая. Поддерживаемое время: в 8 утра/дня/вечер, в 21 вечера, в 22:00