

# TG-БОТ - UML Sequence - блоки SEQ-01-10 (PlantUML)

```
@startuml
title SEQ-01 P1 Онбординг и проверка подписки
hide footbox
autoactivate on
skinparam shadowing false
skinparam ParticipantPadding 8
skinparam BoxPadding 6
skinparam maxMessageSize 55
skinparam SequenceMessageAlign center

actor "Пользователь\n(Telegram)" as U
participant "Telegram Platform\n(входящие события)" as TP
participant "Бэкенд TG-бота\n(P1 Онбординг)" as B
participant "Telegram Bot API" as TG
database "БД D1\nПользователи" as D1

note over TP,B
Webhook update. Если callback_query - ответить answerCallbackQuery (не показываем в каждом шаге).
end note

== /start ==
U -> TP : /start
TP ->> B : update (message)

B -> TG : sendMessage(приветствие + кнопка "Старт")
TG --> B : ok
TG -> TP : deliver
TP --> U : приветствие

== "Старт" и проверка подписки ==
U -> TP : "Старт"
TP ->> B : update (callback_query)

B -> TG : getChatMember(канал)
TG --> B : статус_подписки

alt не подписан
B -> TG : sendMessage(экран подписки + ссылка на канал)
TG --> B : ok
TG -> TP : deliver
TP --> U : экран подписки

note over U,B
Подписка не шлет update. Проверяем при следующем действии пользователя.
end note

U -> TP : любое действие
TP ->> B : update (message/callback_query)
B -> TG : getChatMember(повтор)
```

TG --> B : подписан  
end

B -> D1 : upsert Пользователь(статусподписки, датапроверки)  
D1 --> B : ok

B -> TG : sendMessage(главное меню)

TG --> B : ok

TG -> TP : deliver

TP --> U : главное меню

@enduml

---

@startuml

title SEQ-02 P2 Главное меню - переход в "Поиск мест" и "Категории"

hide footbox

autoactivate on

skinparam shadowing false

skinparam ParticipantPadding 8

skinparam BoxPadding 6

skinparam maxMessageSize 55

skinparam SequenceMessageAlign center

actor "Пользователь\n(Telegram)" as U

participant "Telegram Platform\n(входящие события)" as TP

participant "Бэкенд TG-бота\n(P2 Навигация)" as B

participant "Telegram Bot API" as TG

database "БД D1\nПользователи" as D1

note over TP,B

Webhook: backend быстро отвечает ACK 200 (пустое тело).

end note

note over U,B

В главном меню также: Избранное, Помощь, Обратная связь, Промокоды  
(эти ветки описаны в отдельных SEQ).

end note

== Переход в Поиск мест ==

U -> TP : нажать "Поиск мест"

TP -> B : update (callback\_query)

B -> TG : answerCallbackQuery

TG --> B : ok

opt проверка статуса/контекста (опционально)

B -> D1 : получить Пользователь (контекст/статус)

D1 --> B : ok

end

B -> TG : sendMessage(экран "Категории")

TG --> B : ok

TG -> TP : доставить сообщение в чат

TP --> U : экран "Категории"\n(Заведения, Наши подборки, Афиша)

@enduml

---

@startuml

title SEQ-03 Р3 Заведения - список и переход в карточку (Р6)

hide footbox

autoactivate on

skinparam shadowing false

skinparam ParticipantPadding 8

skinparam BoxPadding 6

skinparam maxMessageSize 55

skinparam SequenceMessageAlign center

actor "Пользователь\n(Telegram)" as U

participant "Telegram Platform\n(входящие события)" as TP

participant "Бэкенд TG-бота\n(Р3 Заведения)" as B

participant "Сервис карточек\n(Р6)" as P6

participant "Telegram Bot API" as TG

database "БД D2\nЗаведения и справочники" as D2

database "БД D4\nИзбранное" as D4

participant "Яндекс Карты\n(приложение)" as YM

note over TP,B

Списки: sendMessage. Карточки: sendPhoto + fallback sendMessage.

Если callback\_query - выполнить answerCallbackQuery.

Кнопка "Найти на Яндекс-картах" - url, клика в backend нет.

end note

== Список ==

U -> TP : запрос списка/фильтра

TP ->> B : update

B -> D2 : select места (page, фильтры)

D2 --> B : список

B -> TG : sendMessage(список мест)

TG --> B : ok

TG -> TP : deliver

TP ->> U : список

opt пагинация/фильтры

loop

U -> TP : Next/Back/фильтр

TP ->> B : update

B -> D2 : select места (page=N)

D2 --> B : список

B -> TG : sendMessage(обновленный список)

TG -> TP : deliver

```

TP --> U : обновленный список
end
end

== Карточка ==
U -> TP : выбрать место
TP ->> B : update (callback_query): place_id

B -> P6 : buildCard(place_id, тип=заведение)
P6 -> D2 : read место + photo_url
D2 --> P6 : данные
P6 -> D4 : read избранное (опционально)
D4 --> P6 : да/нет
P6 --> B : card_text + photo_url + reply_markup(url maps)

alt photo_url доступен
B -> TG : sendPhoto(photo_url, caption=card_text, reply_markup)
TG --> B : ok | media error
alt media error
B -> TG : sendMessage(card_text, reply_markup)
TG --> B : ok
end

else photo_url нет/недоступен
B -> TG : sendMessage(card_text, reply_markup)
TG --> B : ok
end

TG -> TP : deliver
TP --> U : карточка

```

---

```

@startuml
title SEQ-04 Р5 Подборки - список подборок, список мест, карточка (Р6)
hide footbox
autoactivate on
skinparam shadowing false
skinparam ParticipantPadding 8
skinparam BoxPadding 6
skinparam maxMessageSize 55
skinparam SequenceMessageAlign center

actor "Пользователь\n(Telegram)" as U
participant "Telegram Platform\n(входящие события)" as TP
participant "Бэкенд TG-бота\n(Р5 Подборки)" as B
participant "Сервис карточек\n(Р6)" as P6
participant "Telegram Bot API" as TG

```

database "БД D5\nПодборки" as D5  
database "БД D2\nЗаведения и справочники" as D2  
database "БД D4\nИзбранное" as D4

note over TP,B

Списки: sendMessage. Карточки: sendPhoto + fallback sendMessage.

Если callback\_query - выполнить answerCallbackQuery.

end note

== Список подборок ==

U -> TP : открыть "Подборки"

TP ->> B : update

B -> D5 : select подборки (page)

D5 --> B : список

B -> TG : sendMessage(список подборок)

TG -> TP : deliver

TP ->> U : список подборок

== Выбор подборки (список мест) ==

U -> TP : выбрать подборку

TP ->> B : update (callback\_query): collection\_id

B -> D5 : read подборка + place\_id[]

D5 --> B : place\_id[]

B -> D2 : select места (по place\_id[])

D2 --> B : список мест

B -> TG : sendMessage(список мест из подборки)

TG -> TP : deliver

TP ->> U : список мест

== Карточка места ==

U -> TP : выбрать место

TP ->> B : update (callback\_query): place\_id

B -> P6 : buildCard(place\_id, тип=заведение)

P6 -> D2 : read место + photo\_url

D2 --> P6 : данные

P6 -> D4 : read избранное (опционально)

D4 --> P6 : да/нет

P6 ->> B : card\_text + photo\_url + reply\_markup

alt photo\_url доступен

B -> TG : sendPhoto(...)

TG --> B : ok | media error

alt media error

B -> TG : sendMessage(...)

TG --> B : ok

end

else photo\_url нет/недоступен

B -> TG : sendMessage(...)

TG --> B : ok

end

TG -> TP : deliver  
TP --> U : карточка

@enduml

---

@startuml  
title SEQ-05 P4 Афиша - список мероприятий и выбор карточки (см. SEQ-06)

hide footbox  
autoactivate on  
skinparam shadowing false  
skinparam ParticipantPadding 8  
skinparam BoxPadding 6  
skinparam maxMessageSize 55  
skinparam SequenceMessageAlign center

actor "Пользователь\n(Telegram)" as U  
participant "Telegram Platform\n(входящие события)" as TP  
participant "Бэкенд TG-бота\n(P4 Афиша)" as B  
participant "Telegram Bot API" as TG  
database "БД D3\nМероприятия" as D3

note over TP,B  
Афиша - списки sendMessage. Карточка события в SEQ-06 (sendPhoto + fallback).  
Если callback\_query - выполнить answerCallbackQuery.  
end note

U -> TP : открыть "Афиша"  
TP ->> B : update  
  
B -> D3 : select мероприятия (режим/дата/page)  
D3 --> B : список  
B -> TG : sendMessage(список мероприятий)  
TG -> TP : deliver  
TP --> U : список

opt выбрать мероприятие  
U -> TP : выбрать событие  
TP ->> B : update (callback\_query): event\_id  
ref over U,B : SEQ-06 P6 Карточка мероприятия  
end

@enduml

---

@startuml  
title SEQ-06 P6 Карточка мероприятия - детали, избранное, карты  
hide footbox  
autoactivate on  
skinparam shadowing false  
skinparam ParticipantPadding 8

```
skinparam BoxPadding 6
skinparam maxMessageSize 55
skinparam SequenceMessageAlign center

actor "Пользователь\n(Telegram)" as U
participant "Telegram Platform\n(входящие события)" as TP
participant "Бэкенд TG-бота\n(P4 Афиша)" as B
participant "Сервис карточек\n(P6)" as P6
participant "Сервис избранного\n(P7)" as P7
participant "Telegram Bot API" as TG
database "БД D3\nМероприятия" as D3
database "БД D4\nИзбранное" as D4
participant "Яндекс Карты\n(приложение)" as YM
```

note over TP,B

Карточки: sendPhoto + fallback sendMessage.

Если callback\_query - выполнить answerCallbackQuery.

Кнопка "Найти на Яндекс-картах" - url (без update).

end note

== Карточка ==

U -> TP : выбрать мероприятие

TP ->> B : update (callback\_query): event\_id

B -> P6 : buildCard(event\_id, тип=мероприятие)

P6 -> D3 : read мероприятие + photo\_url

D3 --> P6 : данные

P6 -> D4 : read избранное (опционально)

D4 --> P6 : да/нет

P6 --> B : card\_text + photo\_url + reply\_markup(url maps)

alt photo\_url доступен

B -> TG : sendPhoto(...)

TG --> B : ok | media error

alt media error

B -> TG : sendMessage(...)

TG --> B : ok

end

else photo\_url нет/недоступен

B -> TG : sendMessage(...)

TG --> B : ok

end

TG -> TP : deliver

TP --> U : карточка

opt открыть карту по url-кнопке

U -> YM : открыть url "Найти на Яндекс-картах"

end

deactivate

== Избранное ==

opt добавить/убрать

U -> TP : "В избранное"/"Убрать"  
TP ->> B : update (callback\_query)

B -> P7 : toggleFavorite(event\_id)  
P7 -> D4 : upsert/delete  
D4 --> P7 : ok  
P7 --> B : ok

B -> TG : sendMessage(подтверждение)  
TG -> TP : deliver  
TP --> U : подтверждение  
end

@enduml

---

@startuml

title SEQ-07 P8 Промокоды - список, карточка акции с фото, переход в место  
hide footbox  
autoactivate on  
skinparam shadowing false  
skinparam ParticipantPadding 8  
skinparam BoxPadding 6  
skinparam maxMessageSize 55  
skinparam SequenceMessageAlign center

actor "Пользователь\n(Telegram)" as U  
participant "Telegram Platform\n(входящие события)" as TP  
participant "Бэкенд TG-бота\n(P8 Промокоды)" as B  
participant "Сервис карточек\n(P6)" as P6  
participant "Telegram Bot API" as TG  
database "БД D6\nАкции" as D6  
database "БД D2\nЗаведения и справочники" as D2

note over TP,B  
Список промокодов: sendMessage. Карточка промокода: sendPhoto (фото заведения) + fallback  
sendMessage.  
Если callback\_query - выполнить answerCallbackQuery.  
end note

== Список ==  
U -> TP : открыть "Промокоды"  
TP ->> B : update

B -> D6 : select акции (page)  
D6 --> B : список  
B -> TG : sendMessage(список акций)  
TG -> TP : deliver  
TP --> U : список

== Карточка акции ==  
U -> TP : выбрать акцию

TP -> B : update (callback\_query): promo\_id

B -> D6 : read акция(promo\_id -> place\_id)

D6 --> B : данные акции

B -> D2 : read место(place\_id) + photo\_url

D2 --> B : данные места

alt photo\_url доступен

B -> TG : sendPhoto(photo\_url, caption=детали акции, reply\_markup)

TG --> B : ok | media error

alt media error

B -> TG : sendMessage(детали акции, reply\_markup)

TG --> B : ok

end

else photo\_url нет/недоступен

B -> TG : sendMessage(детали акции, reply\_markup)

TG --> B : ok

end

TG -> TP : deliver

TP --> U : карточка акции

opt "Подробнее о месте"

U -> TP : "Подробнее о месте"

TP ->> B : update (callback\_query): place\_id

ref over U,B : SEQ-03 (карточка места через P6)

end

@enduml

---

@startuml

title SEQ-08 Р7 Избранное - список, удаление, карточка элемента (с фото)

hide footbox

autoactivate on

skinparam shadowing false

skinparam ParticipantPadding 8

skinparam BoxPadding 6

skinparam maxMessageSize 55

skinparam SequenceMessageAlign center

actor "Пользователь\n(Telegram)" as U

participant "Telegram Platform\n(входящие события)" as TP

participant "Бэкенд TG-бота\n(P7 Избранное)" as B

participant "Сервис карточек\n(P6)" as P6

participant "Telegram Bot API" as TG

database "БД D4\nИзбранное" as D4

database "БД D2\nЗаведения и справочники" as D2

database "БД D3\nМероприятия и справочники" as D3

note over TP,B

Список избранного: sendMessage. Карточка элемента: sendPhoto + fallback sendMessage.

Если callback\_query - выполнить answerCallbackQuery.  
end note

== Список ==

U -> TP : открыть "Избранное"

TP ->> B : update

B -> D4 : select избранное (page)

D4 --> B : ссылки (id по типам)

B -> D2 : read места (по списку)

D2 --> B : данные мест

B -> D3 : read мероприятия (по списку)

D3 --> B : данные мероприятий

B -> TG : sendMessage(список избранного)

TG -> TP : deliver

TP --> U : список

== Удаление ==

opt удалить элемент из списка

U -> TP : "Удалить"

TP ->> B : update (callback\_query)

B -> D4 : delete избранное (по элементу)

D4 --> B : ok

ref over U,B : обновить список (как в блоке "Список")

end

== Карточка элемента ==

U -> TP : открыть элемент

TP ->> B : update (callback\_query): элемент

B -> P6 : buildCard(объект)

P6 --> B : card\_text + photo\_url + reply\_markup

alt photo\_url доступен

B -> TG : sendPhoto(...)

TG --> B : ok | media error

alt media error

B -> TG : sendMessage(...)

TG --> B : ok

end

else photo\_url нет/недоступен

B -> TG : sendMessage(...)

TG --> B : ok

end

TG -> TP : deliver

TP --> U : карточка

@enduml

---

```

@startuml
title SEQ-09 P9 Помощь - FAQ и обратная связь (без хранения)
hide footbox
autoactivate on
skinparam shadowing false
skinparam ParticipantPadding 8
skinparam BoxPadding 6
skinparam maxMessageSize 55
skinparam SequenceMessageAlign center

actor "Пользователь\n(Telegram)" as U
participant "Telegram Platform\n(входящие события)" as TP
participant "Бэкенд TG-бота\n(P9 Помощь)" as B
participant "Telegram Bot API" as TG

note over TP,B
FAQ/контакты: только sendMessage. Если callback_query - выполнить answerCallbackQuery.
end note

U -> TP : открыть "Помощь"
TP ->> B : update
B -> TG : sendMessage(FAQ/инструкция)
TG -> TP : deliver
TP --> U : FAQ/инструкция

opt "Обратная связь"
U -> TP : открыть "Обратная связь"
TP ->> B : update
B -> TG : sendMessage(контакты/канал связи)
TG -> TP : deliver
TP --> U : контакты
end

@enduml

```

---

```

@startuml
title SEQ-10 P10 Импорт контента из CMS (сервисный контур)
hide footbox
autoactivate on
skinparam shadowing false
skinparam ParticipantPadding 8
skinparam BoxPadding 6
skinparam maxMessageSize 55
skinparam SequenceMessageAlign center

participant "CMS\n(источник контента)" as CMS
participant "Бэкенд\n(P10 Импорт)" as B
database "БД D2\nЗаведения и справочники" as D2
database "БД D3\nМероприятия и справочники" as D3

```

database "БД D5\nПодборки" as D5

database "БД D6\nАкции" as D6

CMS -> B : payload контента (пакет/дельта)

B -> D2 : upsert заведения/справочники (+ photo\_url)

D2 --> B : ok

B -> D3 : upsert мероприятия/справочники (+ photo\_url)

D3 --> B : ok

B -> D5 : upsert подборки + связи

D5 --> B : ok

B -> D6 : upsert акции + привязки

D6 --> B : ok

note over CMS,B

[служебное] Логирование sync\_id/traceld и счетчиков - в документации интеграции.

end note

@enduml