

# Автоматическая функция запаса времени (Padding) для нарезки видео

---

## Описание

Эта функция автоматически добавляет запас времени (padding) в начале и конце каждого видео сегмента при нарезке. Это предотвращает обрезание речи или аудио на краях сегментов.

## Ключевые возможности

### 1. Автоматический запас времени

- По умолчанию: **+2 секунды до начала, +2 секунды после конца**
- Применяется автоматически ко всем нарезкам
- Сохраняет естественное звучание речи на краях сегментов

### 2. Настраиваемые параметры

- Отдельные настройки для начала и конца сегмента
- Варианты: **0, 1, 2, 3, 5 секунд**
- Сохраняются в базе данных для каждого пользователя

### 3. Интерфейс настроек

- Команда `/settings` для доступа к настройкам
- Интуитивный интерфейс с кнопками
- Мгновенное обновление настроек

### 4. Информация в подтверждении

- Показывает текущий запас времени
- Пример: `4:40:00-4:51:36 → 4:39:58-4:51:38`
- Кнопка “Без запаса” для точной нарезки
- Кнопка “Изменить запас” для быстрой настройки

### 5. Умная обработка граничных случаев

- Не уходит в отрицательные значения (начало видео = 00:00:00)
- Не превышает длительность видео
- Предупреждение о пересечении сегментов

## Как использовать

### Настройка запаса времени

1. Отправьте команду `/settings`
2. Выберите нужное значение для начала сегмента (0-5 сек)
3. Выберите нужное значение для конца сегмента (0-5 сек)
4. Настройки сохраняются автоматически

## Нарезка видео с запасом

1. Отправьте видео или YouTube ссылку
2. Отправьте таймкоды для нарезки (например, 4:40:00-4:51:36 )
3. Бот покажет подтверждение с информацией о запасе:

...

 Найдено 38 отрезков

С запасом: +2 сек в начале, +2 сек в конце

Пример: 4:40:00-4:51:36 → 4:39:58-4:51:38

...

4. Выберите:

- **Да, начать** - нарека с запасом
-  **Изменить запас** - настроить запас
-  **Без запаса** - точная нарека без padding
-  **Отмена** - отменить



## Технические детали

### Обновленные файлы

#### 1. database.py

- Добавлены поля `start_padding` и `end_padding` в таблицу `users`
- Методы `get_padding_settings()` и `set_padding_settings()`
- Автоматическая миграция для существующих баз данных

#### 2. utils.py

- Функция `apply_padding_to_timecodes()` - применяет padding к таймкодам
- Функция `check_overlapping_segments()` - проверяет пересечения
- Обработка граничных случаев (границы видео)

#### 3. bot.py

- Команда `/settings` для настройки padding
- Обновлено подтверждение нареки с показом padding
- Callback handlers для кнопок настроек
- Поддержка “Без запаса” для точной нареки

#### 4. test\_padding.py

- Комплексные тесты для всех функций padding
- Проверка граничных случаев
- Тест реального примера из требований

## Примеры использования кода

```
# Применение padding к таймкодам
from utils import apply_padding_to_timecodes

timecodes = [(100, 200), (300, 400)] # Оригинальные таймкоды
video_duration = 500 # Длительность видео

padded = apply_padding_to_timecodes(
    timecodes,
    start_padding=2, # +2 сек в начале
    end_padding=2, # +2 сек в конце
    video_duration=video_duration
)
# Результат: [(98, 202), (298, 402)]
```

```
# Проверка пересечений
from utils import check_overlapping_segments

timecodes = [(0, 15), (10, 25), (30, 40)]
overlaps = check_overlapping_segments(timecodes)
# Результат: [(0, 1)] # Сегменты 0 и 1 пересекаются
```

```
# Получение настроек пользователя
from database import db

settings = await db.get_padding_settings(user_id)
# Результат: {'start_padding': 2, 'end_padding': 2}
```

## Тестирование

Все функции протестированы:

```
cd /home/ubuntu/wowsilizing_bot
python3 test_padding.py
```

Результаты тестов:

- Базовое применение padding
- Проверка границ (не уходит в отрицательные значения)
- Проверка длительности видео
- Обнаружение пересечений
- Реальный пример: 4:40:00-4:51:36 → 4:39:58-4:51:38



## База данных

### Схема таблицы users

```
CREATE TABLE users (
    user_id INTEGER PRIMARY KEY,
    username TEXT,
    is_premium INTEGER DEFAULT 0,
    start_padding INTEGER DEFAULT 2, -- Новое поле
    end_padding INTEGER DEFAULT 2, -- Новое поле
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    last_active TIMESTAMP DEFAULT CURRENT_TIMESTAMP
)
```

### Миграция

Для существующих баз данных автоматически добавляются новые колонки:

```
ALTER TABLE users ADD COLUMN start_padding INTEGER DEFAULT 2;
ALTER TABLE users ADD COLUMN end_padding INTEGER DEFAULT 2;
```



### Рекомендации

- **Для речи:** используйте 2-3 секунды запаса
- **Для музыки:** можно использовать 1-2 секунды
- **Для точной нарезки:** используйте 0 секунд или кнопку “Без запаса”
- **При большом количестве сегментов:** проверяйте предупреждения о пересечениях



### Примеры использования

#### Пример 1: Нарезка стрима с речью

```
Настройки: начало = 2 сек, конец = 2 сек
Таймкод: 1:30:00-1:35:00
Результат: 1:29:58-1:35:02
```

#### Пример 2: Точная нарезка без запаса

```
Кнопка "Без запаса" или настройки: начало = 0, конец = 0
Таймкод: 1:30:00-1:35:00
Результат: 1:30:00-1:35:00
```

#### Пример 3: Разный запас для начала и конца

```
Настройки: начало = 3 сек, конец = 1 сек
Таймкод: 1:30:00-1:35:00
Результат: 1:29:57-1:35:01
```



## Преимущества

1. **Сохранение речи** - речь не обрезается на краях сегментов
2. **Автоматическое применение** - работает автоматически по умолчанию
3. **Гибкость** - настраивается под разные задачи
4. **Безопасность** - проверяет границы и пересечения
5. **Удобство** - интуитивный интерфейс на русском языке



## История изменений

### Версия 1.0 (2025-11-25)

- Добавлена функция автоматического запаса времени
- Команда /settings для настройки
- Показ padding в подтверждении нарезки
- Проверка граничных случаев и пересечений
- Автоматическая миграция базы данных
- Полное тестирование функций

Разработано для **WOWsilizing Bot**

Сохраняем каждое слово вашего контента