

# Примеры использования - SSV Monetization Tool



## Содержание

1. Простые примеры
2. Продвинутые примеры
3. Интеграция с другими инструментами
4. Кастомные сценарии

## Простые примеры

### Пример 1: Базовая обработка видео

```
#!/usr/bin/env python3
"""Базовый пример обработки видео с монетизацией."""

from utils.config_loader import load_and_validate_config
from modules.strategy_planner import determine_actions_for_strategy
from modules.content_injector import inject_monetization_elements

# Загрузка конфигурации
config = load_and_validate_config("monetization_config.yaml")

# Исходный контент видео
video_content = {
    'id': 'video_gastric_surgery_001',
    'title': 'Техника гастрэктомии при раке желудка',
    'description': ''
}
Подробный разбор техники выполнения гастрэктомии при раке желудка.
В видео используются современные хирургические инструменты.
Рекомендую ознакомиться с медицинскими книгами по онкохирургии.
    '''.strip()
}

# Определение действий на основе стратегии
strategy = config['monetization']['strategy']
actions = determine_actions_for_strategy(strategy, config)

# Внедрение элементов монетизации
result = inject_monetization_elements(video_content, actions, config)

# Вывод результата
print("=" * 60)
print(f"ID: {result['id']}")
print(f"Title: {result['title']}")
print(f"\nDescription:\n{result['description']}")
print("=" * 60)
```

Ожидаемый результат (стратегия MASKED):

```
=====
ID: video_gastric_surgery_001
Title: Техника гастрэктомии при раке желудка

Description:
Подробный разбор техники выполнения гастрэктомии при раке желудка.
В видео используются современные хирургические инструменты (https://medicalstore.com/tools?ref=ssvproff&utm\_source=youtube&utm\_campaign=video\_gastric\_surgery\_001).
Рекомендую ознакомиться с медицинскими книгами (https://amazon.com/books?tag=ssvproff-20&utm\_source=youtube&utm\_campaign=video\_gastric\_surgery\_001) по онкохирургии.

 Получите полный доступ к курсу на https://ssvnauka.com/premium?utm\_source=youtube&utm\_campaign=video\_gastric\_surgery\_001
=====
```

## Пример 2: Проверка соответствия YouTube

```
#!/usr/bin/env python3
"""Проверка описания видео на соответствие политикам YouTube."""

from modules.compliance_checker import check_youtube_description_compliance

# Описание видео
description = """
КУПИТЬ СЕЙЧАС!!! СКИДКА 90%!!!
Переходите по ссылке ПРЯМО СЕЙЧАС!!!
https://link1.com https://link2.com https://link3.com
... (еще 20 ссылок)
"""

# Проверка соответствия
issues = check_youtube_description_compliance(description)

if issues:
    print("⚠️ Обнаружены проблемы с соответствием YouTube:")
    for issue in issues:
        print(f" ✗ {issue}")
else:
    print("✅ Описание соответствует политикам YouTube")
```

### Ожидаемый результат:

- ⚠️ Обнаружены проблемы с соответствием YouTube:
  - ✗ Excessive caps detected
  - ✗ Excessive exclamation marks
  - ✗ Too many links (23), YouTube limit is 15
  - ✗ Spam patterns detected

### Пример 3: Генерация уникальных партнёрских ссылок

```
#!/usr/bin/env python3
"""Генерация уникальных партнёрских ссылок с UTM-метками."""

from modules.analytics_tracker import generate_unique_affiliate_link

# Базовая ссылка на продукт
base_url = "https://amazon.com/surgical-instruments"

# Генерация уникальной ссылки для видео
video_link = generate_unique_affiliate_link(
    base_url=base_url,
    content_id="video_001",
    source="youtube",
    medium="description"
)

# Генерация уникальной ссылки для книги
book_link = generate_unique_affiliate_link(
    base_url=base_url,
    content_id="book_001",
    source="amazon_kdp",
    medium="chapter"
)

print(f"Ссылка для видео: {video_link}")
print(f"Ссылка для книги: {book_link}")
```

#### Ожидаемый результат:

```
Ссылка для видео: https://amazon.com/surgical-instruments?
utm_source=youtube&utm_medium=description&utm_campaign=video_001
Ссылка для книги: https://amazon.com/surgical-instruments?
utm_source=amazon_kdp&utm_medium=chapter&utm_campaign=book_001
```

## Продвинутые примеры

### Пример 4: Обработка нескольких видео

```

#!/usr/bin/env python3
"""Пакетная обработка нескольких видео."""

from utils.config_loader import load_and_validate_config
from modules.strategy_planner import determine_actions_for_strategy
from modules.content_injector import inject_monetization_elements
from modules.analytics_tracker import prepare_monetization_report, calculate_monetization_metrics

# Загрузка конфигурации
config = load_and_validate_config("monetization_config.yaml")
strategy = config['monetization']['strategy']
actions = determine_actions_for_strategy(strategy, config)

# Список видео
videos = [
    {
        'id': 'video_001',
        'title': 'Лапароскопическая холецистэктомия',
        'description': 'Разбор техники удаления желчного пузыря...'
    },
    {
        'id': 'video_002',
        'title': 'Резекция желудка при раке',
        'description': 'Онкохирургическая операция на желудке...'
    },
    {
        'id': 'video_003',
        'title': 'Панкреатодуоденальная резекция',
        'description': 'Операция Уиппла: техника и нюансы...'
    }
]

# Обработка всех видео
processed_videos = []
for video in videos:
    result = inject_monetization_elements(video, actions, config)
    metrics = calculate_monetization_metrics(result)
    result['metrics'] = metrics
    processed_videos.append(result)

# Генерация общего отчёта
print("📊 Общий отчёт о монетизации:")
print(f"  Обработано видео: {len(processed_videos)}")
print(f"  Стратегия: {strategy.upper()}")
print(f"\n  Результаты:")

for video in processed_videos:
    print(f"\n    📹 {video['title']}")
    print(f"    ID: {video['id']}")
    print(f"    Партнёрских ссылок: {video['metrics'].get('affiliate_links_count', 0)}")
    print(f"    Длина описания: {video['metrics'].get('description_length', 0)} символов")

```

## Пример 5: Кастомная стратегия монетизации

```

#!/usr/bin/env python3
"""Создание кастомной стратегии монетизации."""

from typing import Dict, Any, List

def custom_monetization_strategy(content: Dict[str, Any], config: Dict[str, Any]) ->
List[str]:
    """
    Кастомная логика определения действий монетизации.

    Правила:
    - Если заголовок содержит "рак" → FULL стратегия (информационный контент)
    - Если заголовок содержит "техника" → PARTIAL (образовательный контент)
    - Иначе → MASKED (обычный контент)
    """

    actions = []
    title = content.get('title', '').lower()

    if 'рак' in title or 'онкология' in title:
        # Полная монетизация для информативного контента
        actions = ['inject_affiliate_links', 'add_affiliate_disclaimer', 'add_premium_cta']
    elif 'техника' in title or 'операция' in title:
        # Частичная монетизация для образовательного контента
        actions = ['inject_affiliate_links', 'add_affiliate_disclaimer']
    else:
        # Замаскированная монетизация для обычного контента
        actions = ['inject_affiliate_links', 'add_premium_cta']

    return actions

# Использование
from modules.content_injector import inject_monetization_elements
from utils.config_loader import load_and_validate_config

config = load_and_validate_config("monetization_config.yaml")

content = {
    'id': 'video_004',
    'title': 'Ранняя диагностика рака желудка',
    'description': 'Методы ранней диагностики...'
}

# Применение кастомной стратегии
custom_actions = custom_monetization_strategy(content, config)
print(f"Действия для '{content['title']}': {custom_actions}")

# Внедрение элементов
result = inject_monetization_elements(content, custom_actions, config)

```

## Интеграция с другими инструментами

### Пример 6: Интеграция с ssv-video

```

#!/usr/bin/env python3
"""Интеграция с генератором видеопакетов ssv-video."""

import sys
sys.path.append('../ssv-video') # Путь к ssv-video

from ssv_video_generator import VideoPackageGenerator
from utils.config_loader import load_and_validate_config
from modules.strategy_planner import determine_actions_for_strategy
from modules.content_injector import inject_monetization_elements

# Инициализация генератора видеопакетов
video_gen = VideoPackageGenerator()

# Создание видеопакета
video_package = video_gen.create_package(
    title="Лапароскопическая холецистэктомия",
    script="scripts/cholecystectomy.md",
    keywords=["хирургия", "холецистэктомия", "лапароскопия"]
)

# Загрузка конфигурации монетизации
monetization_config = load_and_validate_config("../ssv-monetization-tool/monetization_config.yaml")

# Применение монетизации к видеопакету
content = {
    'id': video_package['id'],
    'title': video_package['title'],
    'description': video_package['description']
}

strategy = monetization_config['monetization']['strategy']
actions = determine_actions_for_strategy(strategy, monetization_config)
monetized_content = inject_monetization_elements(content, actions,
monetization_config)

# Обновление видеопакета
video_package['description'] = monetized_content['description']

# Экспорт готового видеопакета
video_gen.export(video_package, format='youtube')

print("✅ Видеопакет с монетизацией готов к загрузке на YouTube!")

```

## Пример 7: Интеграция с ssv-book-generator

```

#!/usr/bin/env python3
"""Интеграция с генератором книг ssv-book-generator."""

import sys
sys.path.append('../ssv-book-generator')

from book_generator import BookGenerator
from utils.config_loader import load_and_validate_config
from modules.content_injector import inject_monetization_elements

# Создание книги
book_gen = BookGenerator()
book = book_gen.create_book(
    title="Руководство по хирургии желудка",
    chapters=[
        "Анатомия желудка",
        "Предоперационная подготовка",
        "Техника резекции",
        "Послеоперационное ведение"
    ]
)

# Применение монетизации
config = load_and_validate_config("../ssv-monetization-tool/monetization_config.yaml")

# Монетизация описания книги
book_content = {
    'id': book['id'],
    'title': book['title'],
    'description': book['description']
}

monetized_book = inject_monetization_elements(
    book_content,
    ['inject_affiliate_links', 'add_affiliate_disclaimer'],
    config
)

# Обновление книги
book['description'] = monetized_book['description']

# Экспорт для Amazon KDP
book_gen.export(book, format='kdp')

print("✅ Книга с монетизацией готова к публикации на Amazon KDP!")

```

## Кастомные сценарии

### Пример 8: А/В тестирование стратегий

```

#!/usr/bin/env python3
"""А/В тестирование различных стратегий монетизации."""

from utils.config_loader import load_and_validate_config
from modules.strategy_planner import determine_actions_for_strategy
from modules.content_injector import inject_monetization_elements
from modules.analytics_tracker import calculate_monetization_metrics

# Исходный контент
content = {
    'id': 'video_ab_test',
    'title': 'Техника лапароскопической холецистэктомии',
    'description': 'Подробный разбор техники операции...'
}

# Тестируемые стратегии
strategies = ['full', 'partial', 'masked', 'hidden']

# Результаты тестирования
results = {}

for strategy in strategies:
    # Загрузка конфигурации с соответствующей стратегией
    config = load_and_validate_config("monetization_config.yaml")
    config['monetization']['strategy'] = strategy

    # Применение монетизации
    actions = determine_actions_for_strategy(strategy, config)
    result = inject_monetization_elements(content.copy(), actions, config)

    # Расчёт метрик
    metrics = calculate_monetization_metrics(result)

    # Сохранение результатов
    results[strategy] = {
        'actions': actions,
        'description_length': metrics.get('description_length', 0),
        'affiliate_links_count': metrics.get('affiliate_links_count', 0),
        'description': result['description'][:100] + '...' # Первые 100 символов
    }

# Вывод результатов
print("📊 Результаты А/В тестирования:\n")
for strategy, data in results.items():
    print(f"⌚ Стратегия: {strategy.upper()}")
    print(f"  Действия: {' '.join(data['actions'])} {data['actions']} if data['actions'] else 'Нет'")
    print(f"  Длина описания: {data['description_length']} символов")
    print(f"  Партнёрских ссылок: {data['affiliate_links_count']}")
    print(f"  Описание: {data['description']}")
    print()

```

## Пример 9: Мониторинг метрик в реальном времени

```

#!/usr/bin/env python3
"""Мониторинг метрик монетизации в реальном времени."""

import time
from modules.analytics_tracker import track_monetization_event, calculate_monetization_metrics

def monitor_monetization_metrics(content_list, interval=60):
    """
    Мониторинг метрик каждые N секунд.

    Args:
        content_list: Список контента для мониторинга
        interval: Интервал обновления в секундах
    """

    while True:
        print("\n" + "=" * 60)
        print(f"📊 Обновление метрик: {time.strftime('%Y-%m-%d %H:%M:%S')}")
        print("=" * 60)

        for content in content_list:
            metrics = calculate_monetization_metrics(content)

            print(f"\n🎥 {content['title']}")
            print(f">ID: {content['id']}")
            print(f"Ссылок: {metrics.get('affiliate_links_count', 0)}")
            print(f"Длина: {metrics.get('description_length', 0)} символов")

        # Отслеживание события
        track_monetization_event(
            'metrics_updated',
            content['id'],
            {'metrics': metrics}
        )

        print("\n" + "=" * 60)
        time.sleep(interval)

# Использование
monitored_content = [
    {
        'id': 'video_001',
        'title': 'Видео 1',
        'description': 'Описание с партнёрскими ссылками...'
    },
    {
        'id': 'video_002',
        'title': 'Видео 2',
        'description': 'Другое описание...'
    }
]

# Запуск мониторинга (обновление каждые 60 секунд)
monitor_monetization_metrics(monitored_content, interval=60)

```

## Дополнительные ресурсы

---

- Руководство пользователя (USAGE.md)
  - API документация (API.md)
  - GitHub репозиторий (<https://github.com/Serg2206/ssv-monetization-tool>)
- 

**Автор:** Профессор С.В. Сушков

**Проект:** SSVproff-Ecosystem

**Лицензия:** MIT