📺 YouTube Stream Sync Library

> Бібліотека для створення 24/7 YouTube трансляцій з автоматичною синхронізацією

Show Image

Show Image

🚀 Швидкий старт

Встановлення

```html
<!-- Підключіть бібліотеку -->
<script src="youtube-stream-sync.min.js"></script>
```

Базове використання

```html
```

```html
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Моя трансляція</title>
</head>
<body>
  <!-- Контейнер для плеєра -->
  <div id="my-stream"></div>

  <!-- Підключення бібліотеки -->
  <script src="youtube-stream-sync.js"></script>

  <script>
    // Ініціалізація
    const stream = new YouTubeStreamSync({
      container: '#my-stream',
      playlistUrl: 'https://example.com/playlist.json',
      theme: 'light',
      autoplay: false
    });
  </script>
</body>
</html>
```

📖 API Документація

Конструктор

```javascript
new YouTubeStreamSync(options)
```

Параметри (options)

Параметр	Тип	За замовчуванням	Опис
`container`	string	`'#youtube-stream-sync'`	CSS селектор контейнера
`playlistUrl`	string	**обов'язково**	URL JSON файлу з плейлистом
`theme`	string	`'light'`	Тема: `'light'`, `'dark'`, `'custom'`
`autoplay`	boolean	`false`	Автоматичний старт

Параметр	Тип	За замовчуванням	Опис
`showSchedule`	boolean	`true`	Показати розклад
`showControls`	boolean	`true`	Показати контроли
`locale`	string	`'uk-UA'`	Мова для часу
`timezone`	number	`0`	Зміщення часового поясу (секунди)
`quality`	string	`'default'`	Якість відео
`customStyles`	string	`null`	Кастомні CSS стилі
`onReady`	function	`null`	Callback коли готово
`onPlay`	function	`null`	Callback при відтворенні
`onVideoChange`	function	`null`	Callback при зміні відео
`onError`	function	`null`	Callback при помилці

📋 Формат плейлиста (JSON)

```json
{
  "broadcastStartTime": "2026-01-17T00:00:00Z",
  "programSchedule": [
    {
      "videoId": "dQw4w9WgXcQ",
      "title": "Назва відео",
      "duration": 210
    },
    {
      "videoId": "9bZkp7q19f0",
      "title": "Інше відео",
      "duration": 180
    }
  ]
}
```

Поля:

- `broadcastStartTime` - час початку трансляції (ISO 8601)
- `videoId` - YouTube ID відео

- `title` - назва відео
- `duration` - тривалість в секундах

🎨 Методи

play()

Почати відтворення

```javascript
stream.play();
```

pause()

Призупинити відтворення

```javascript
stream.pause();
```

stop()

Зупинити відтворення

```javascript
stream.stop();
```

setQuality(quality)

Змінити якість відео

```javascript
stream.setQuality('hd1080'); // 'small', 'medium', 'large', 'hd1080', 'default'
```

getCurrentVideo()

Отримати поточне відео

```javascript
const video = stream.getCurrentVideo();
console.log(video.title);
```

destroy()

Знищити інстанс

```javascript
stream.destroy();
```

🎯 Приклади використання

1. Базовий приклад

```javascript
const stream = new YouTubeStreamSync({
  container: '#stream',
  playlistUrl: '/api/playlist.json'
});
```

2. З темною темою

```javascript
const stream = new YouTubeStreamSync({
  container: '#stream',
  playlistUrl: '/api/playlist.json',
  theme: 'dark'
});
```

3. З автостартом

```javascript
const stream = new YouTubeStreamSync({
  container: '#stream',
  playlistUrl: '/api/playlist.json',
  autoplay: true
});
```

4. З callbacks

```javascript
```

```javascript
const stream = new YouTubeStreamSync({
  container: '#stream',
  playlistUrl: '/api/playlist.json',
  onReady: (instance) => {
    console.log('Плеєр готовий!');
  },
  onPlay: (videoIndex) => {
    console.log('Відтворюється відео:', videoIndex);
  },
  onVideoChange: (index, video) => {
    console.log('Нове відео:', video.title);
  },
  onError: (error) => {
    console.error('Помилка:', error);
  }
});
```

5. Без розкладу та контролів (мінімалістичний)

javascript

```javascript
const stream = new YouTubeStreamSync({
  container: '#stream',
  playlistUrl: '/api/playlist.json',
  showSchedule: false,
  showControls: false,
  autoplay: true
});
```

6. З кастомними стилями

javascript

```
const stream = new YouTubeStreamSync({
  container: '#stream',
  playlistUrl: '/api/playlist.json',
  theme: 'custom',
  customStyles: `
    .ytss-wrapper {
      --ytss-primary: #FF6B6B;
      --ytss-bg: #FFF5F5;
      --ytss-text: #333;
    }
    .ytss-player-container {
      border-radius: 20px;
      box-shadow: 0 10px 40px rgba(0,0,0,0.2);
    }
  `
});
```

7. З часовим поясом

javascript

```
const stream = new YouTubeStreamSync({
  container: '#stream',
  playlistUrl: '/api/playlist.json',
  timezone: 3600, // +1 година
  locale: 'en-US'
});
```

🎨 Кастомізація

CSS змінні

Ви можете перевизначити ці змінні для кастомізації:

css

```
.ytss-wrapper {
  --ytss-bg: #ffffff;
  --ytss-text: #2B2D42;
  --ytss-border: #e5e7eb;
  --ytss-primary: #D4AF37;
}
```

Теми

Світла тема (за замовчуванням):

```javascript
theme: 'light'
```

Темна тема:

```javascript
theme: 'dark'
```

Кастомна тема:

```javascript
theme: 'custom',
customStyles: `/* ваші стилі */`
```

🔧 Інтеграція з фреймворками

React

```jsx
import { useEffect, useRef } from 'react';

function StreamPlayer() {
  const containerRef = useRef(null);
  const streamRef = useRef(null);

  useEffect(() => {
    streamRef.current = new YouTubeStreamSync({
      container: containerRef.current,
      playlistUrl: '/api/playlist.json'
    });

    return () => {
      streamRef.current?.destroy();
    };
  }, []);

  return <div ref={containerRef}></div>;
}
```

Vue

vue

```
<template>
  <div ref="streamContainer"></div>
</template>

<script>
export default {
  mounted() {
    this.stream = new YouTubeStreamSync({
      container: this.$refs.streamContainer,
      playlistUrl: '/api/playlist.json'
    });
  },
  beforeUnmount() {
    this.stream?.destroy();
  }
}
</script>
```

Angular

typescript

```
import { Component, ElementRef, ViewChild, OnInit, OnDestroy } from '@angular/core';

@Component({
  selector: 'app-stream',
  template: '<div #streamContainer></div>'
})
export class StreamComponent implements OnInit, OnDestroy {
  @ViewChild('streamContainer') container!: ElementRef;
  private stream: any;

  ngOnInit() {
    this.stream = new (window as any).YouTubeStreamSync({
      container: this.container.nativeElement,
      playlistUrl: '/api/playlist.json'
    });
  }

  ngOnDestroy() {
    this.stream?.destroy();
  }
}
```

🌍 Підтримка браузерів

- ✅ Chrome 90+
- ✅ Firefox 88+
- ✅ Safari 14+
- ✅ Edge 90+
- ✅ Opera 76+

📄 Ліцензія

MIT License - вільне використання у комерційних та некомерційних проектах

🤝 Підтримка

Якщо знайшли баг або маєте пропозицію - створіть issue на GitHub!

📊 Версії

v1.0.0 (2026-01-17)

- ✨ Початковий реліз
- ✅ Автоматична синхронізація
- ✅ Підтримка тем
- ✅ API для розробників
- ✅ Адаптивний дизайн

Створено з ❤️‍🩹 для розробників