

Чек-лист hls.js для продакшена

Версии, контроллеры, ABR-ручки, LL-HLS, DRM, восстановление ошибок, типовые ошибки.

Контроллеры, которых вы реально касаетесь

Stream controller: state machine (IDLE → FRAG_LOADING → PARSING → BUFFER_APPENDING). Любое событие срабатывает на известном переходе.

Level controller: владеет master + media playlists; live-обновление по расписанию.

ABR controller: throughput EWMA × коэффициент запаса; заменяемый: `hls.abrController = ...`

Buffer controller: обёртка MSE SourceBuffer; сериализация append / remove / endOfStream.

EME controller: вход через `drmSystems`; открывает `MediaKeySession` и тянет лицензию.

Loader: fetch / xhr; переопределение через `xhrSetup` / `fetchSetup` для токенов и заголовков.

ABR-ручки, которые имеют значение

`abrBandWidthFactor` (0,95) коэффициент запаса для стационарного выбора.

`abrBandWidthUpFactor` (0,7) более строгий коэффициент для повышения; асимметрия осознанная.

`abrSwitchInterval` (v1.7-alpha) ограничивает число переключений в секунду — лекарство от ABR-флапа.

`startLevel` форсировать ступень при старте; -1 = авто.

`capLevelToPlayerSize` true → не грузить ступень выше, чем размер `<video>`.

`maxBufferLength` (30с) цель forward-буфера; парно с `maxMaxBufferLength`.

Правило восстановления (по одному предложению)

NETWORK_ERROR → `hls.startLoad()`; ставьте лимит на ретраи.

MEDIA_ERROR (1-я) → `hls.recoverMediaError()`; сбросить флаг через 5с.

MEDIA_ERROR (2-я в 5с) → `swapAudioCodec()` + `recoverMediaError()`.

KEY_SYSTEM_ERROR → автоматического восстановления нет; покажите сообщение.

MUX / OTHER_ERROR → `destroy()` и `reload`.

Вызов `destroy()` на каждую ошибку — однострочный баг; всегда сначала читайте `data.type`.

Плеер в семь строк (канонический паттерн)

```
import Hls from 'hls.js';
const v = document.querySelector('video');
if (Hls.isSupported()) {
  const hls = new Hls();
  hls.loadSource('/master.m3u8');
  hls.attachMedia(v);
} else { v.src = '/master.m3u8'; }
```

Откат для Safari использует нативную AVFoundation — предпочтителен для HLS на iOS.

Настройка LL-HLS

`lowLatencyMode` true (по умолчанию с v1.4); включает `parts` + `preload hints`.

`liveSyncDuration` 3 → ~3с от живого края.

`liveMaxLatencyDuration` 10 → макс. отставание до seek к live edge.

`maxLiveSyncPlaybackRate` 1.1 → догонять live ускорением воспроизведения.

CDN должен поддерживать `blocking playlist reload` и `chunked-transfer parts`.

Apple убрала HTTP/2 push из LL-HLS в 2023; текущая спецификация — `blocking reload` + `preload hints`.

Топ ошибок в продакшене

Не ветвиться на Safari → `hls.js` падает или тратит батарею; нужен `canPlayType`-откат.

Вызывать `destroy()` на любую ошибку → пользователь видит `reload` на временный 5xx.

Слушать MANIFEST_LOADED вместо MANIFEST_PARSED → нестабильные падения в Firefox.

Считать, что live `currentTime` начинается с 0 → используйте `seekable.start(0)` / `seekable.end(0)`.

Дефолтные таймауты на спутнике или 3G → расслабьте `manifestLoadingTimeout` / `fragLoadingTimeout`.

Multi-DRM конфигурация (карта `drmSystems`)

Key system	ключ <code>drmSystems</code> (hls.js)	Браузеры / платформы	Схемы
Widevine	<code>com.widevine.alpha</code>	Chrome, Edge, Firefox, Android, ChromeOS	cenc, cbcs
FairPlay	<code>com.apple.fps</code>	Safari (macOS, iOS, iPadOS), tvOS	только cbcs
PlayReady	<code>com.microsoft.playready</code>	Edge на Windows, Xbox, Tizen, webOS	cenc, cbcs

К статье: «hls.js: подробный разбор»

fora-soft.ru/learn/video-streaming/articles-streaming/hls-js-podrobnyy-razbor

Fora Soft · fora-soft.ru