

Shaka Player: чек-лист для продакшена

Версии, движки, ABR-ручки, LL-* конфиг, DRM-матрица, категории ошибок, частые ошибки.

Движки, которые реально трогают

NetworkingEngine: каждый HTTP-запрос; плагин на URI-схему; request- и response-фильтры.

ManifestParser: на каждый формат (DASH, HLS, Smooth); даёт внутренние Variant.

DrmEngine: мост EME к CDM; блок `drm.servers`; advanced на каждую key system.

StreamingEngine: state machine сегментов; кладёт байты в MSE SourceBuffer.

AbrManager: throughput EWMA + safety factor; замена через `configure({abrFactory})`.

Storage: `shaka.offline.Storage`; скачивания в IndexedDB; persistent EME-сессии.

UI Overlay: опциональный bundle `shaka-player.ui.js`; контролы; кнопка Cast.

ABR-ручки, которые имеют значение

`abr.bandwidthDowngradeTarget` (0.95) safety factor для steady-state.

`abr.bandwidthUpgradeTarget` (0.85) более строгий для upshift; асимметрия осознанная.

`abr.defaultBandwidthEstimate` стартовая полоса до первого замера сегмента.

`abr.restrictions` лимиты width/height/bandwidth; защита тира железа.

v5.1 dropped-frame ABR включён по умолчанию; downshift при 4K HEVC на слабом TV.

v5.1 low-latency hint AbrManager знает про low-latency; ближе к live-edge.

Категории ошибок → runbook'и

NETWORK → health CDN, failover, давление ретраев; пользователю: «Проблема с сетью».

STREAMING → тот же CDN runbook + поведение origin shielding.

MEDIA → Packaging / MSE; `unload()` + `load()` для сброса SourceBuffer.

DRM → Runbook лицензий; ротация токена; «Сессия истекла».

MANIFEST → Каталог ассетов; гео-права; «Сейчас недоступно».

Recoverable → маленький toast; CRITICAL → красный баннер. Recoverable никогда не красный.

Плеер на семь строк (канонический паттерн)

```
import shaka from 'shaka-player';
shaka.polyfill.installAll();
if (!shaka.Player.isBrowserSupported()) throw
...;
const video = document.getElementById('v');
const player = new shaka.Player();
await player.attach(video);
await player.load('https://.../asset.mpd');
Тот же код играет .mpd, .m3u8 и .ism — Shaka
выбирает парсер.
```

Конфигурация low-latency (DASH + HLS)

`streaming.lowLatencyMode` true → включает LL-DASH chunks И LL-HLS parts.

`streaming.inaccurateManifestTolerance` 0 → более жёсткое слежение за live-edge.

`streaming.rebufferingGoal` 0.01 → старт, как только один сегмент готов.

`streaming.bufferingGoal` меньшая цель для live-latency; в паре с `rebufferingGoal`.

Пакейджер должен производить CMAF chunks (DASH) или EXT-X-PART (HLS).

CDN должен пропускать chunked transfer encoding без срезания на edge.

Главные частые ошибки

Без `polyfill.installAll()` → Safari тихо ломается на edge-кейсах.

Перепутать порядок `attach()/load()` на ретраях → используйте `unload()+load()`, а не `re-attach`.

Показывать recoverable пользователю → рабочий плеер выглядит сломанным.

`lowLatencyMode` без поддержки пакейджера → бесконечный startup-stall.

Один `retryParameters` на все классы → лицензии откатываются как CDN-обрывы.

Загрузить UI bundle без `controls.css` → кнопки рендерятся как голые прямоугольники.

Конфигурация multi-DRM (карта `drm.servers`)

Key system	Ключ <code>drm.servers</code> (Shaka)	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 + <code>serverCertificateUri</code>
PlayReady	<code>com.microsoft.playready</code>	Edge на Windows, Xbox, Tizen, webOS	cenc, cbcs