

Hey VK — I can't help you with API calls or techniques to download videos from that site. They're clearly intended for in-browser streaming only, and giving steps to bypass that would likely violate the site's Terms of Service and could infringe copyright.

Here are safe, legit options you can try instead:

- Check whether they offer an official **download/offline mode** (mobile app, LMS, or "Download" button) for paying users.
- **Ask support for time-limited links** for offline study (common for learners with slow/patchy internet).
- See if the same videos are published on an official **YouTube/Vimeo channel** with downloads or Creative Commons licensing enabled.
- If you're the **content owner or have written permission to download**, I can help you design a compliant export workflow.

If you *do* have permission (e.g., the author granted you offline access), here's a **generic, compliant pattern** you can give the provider—or use if they expose an API. This is not specific to any site and assumes the owner allows downloads:

Authorized download flow (generic blueprint)

1. **Get a token (OAuth2 client credentials / auth code):**

POST `https://api.<provider>.com/oauth/token`

Body (x-www-form-urlencoded): `grant_type=client_credentials&client_id=...&client_secret=...`

2. **List assets you're entitled to:**

GET `https://api.<provider>.com/v1/videos?owned=true` (Bearer token in Authorization header)

3. **Request a time-limited signed URL for MP4:**

POST `https://api.<provider>.com/v1/videos/{video_id}:createDownloadUrl`

JSON: `{"format":"mp4","ttl_seconds":3600}`

4. **Download using the returned pre-signed link:**

GET `https://cdn.<provider>.com/downloads/<opaque-id>?sig=...&exp=...`

This preserves licensing (short-lived links, scoped entitlements) and stays within policy.

If you'd like, I can also draft a short message you can send them to request official offline access.