DRM-X 5.0

Node.js + Express Custom Integration Development Guide

A practical guide to DRM-X 5.0 website integration, license validation, and course access control

Item	Details
Document version	v1.0.0
Technology stack	Node.js 20+ / Express 5 / Handlebars / node-soap
Integration service	DRM-X 5.0 SOAP License Service
Language	English
Release date	2026-09-15

Scope This example demonstrates the DRM-X 5.0 API workflow. The fixed account, purchased-course list, and in-memory Session are for testing only and must be replaced with the customer's production systems.

Download the sample code

Contents

1. Purpose and Example Scope

This guide explains the configuration, login Session continuation, course validation, ZJGet POST parameters, and the ordered use of six DRM-X 5.0 SOAP methods in DRM-X5_NodeJS+Express_Integration_Example.

- Receive license requests submitted automatically by DRM-X 5.0 and ZJGet.
- Verify website login state and course access before issuing a license.
- Create a DRM-X user when necessary, check user status, update rights, and request the license.
- Use the project as an integration reference rather than a complete LMS, store, or production identity system.

> **Security boundary** The browser submits parameters generated from protected content. DRM-X credentials, user-status checks, rights updates, and license acquisition must remain on the server.

2. Architecture and License Workflow

The workflow connects ZJGet, the Node.js + Express website, the website's user/course data, and the DRM-X 5.0 SOAP service.

```
ZJGet / DRM-X 5.0
    POST /licstore5
          |
Express same-origin POST relay and Session
          |
Login check -> course validation -> DRM-X user checks
          |
Update POST RightsID -> request license HTML
          |
      Return result to ZJGet
```

1. ZJGet reads parameters from the encrypted file and DRM-X License Profile, then posts them to /licstore5.

2. Express saves the parameters and checks the login Session. If the user is not signed in, the request resumes after login.

3. The server compares the posted yourproductid values with courses owned by the current user.
4. The server checks the DRM-X user, creates a missing user, or checks revocation and lockout for an existing user.
5. The server updates the posted rightsid and then requests the license with the same rightsid.
6. DRM-X returns license HTML, which Express validates and passes to the result page and ZJGet.

3. Project Structure and Configuration

3.1 Main Files

File	Responsibility
config.js	DRM-X service settings, demo account, course dates, purchased courses, bind count, Session secret, and port.
DRMXService.js	Express routes, validation, Session continuation, course matching, and the complete SOAP workflow.
views/login.hbs	Fixed demo account login page.
views/relay.hbs	Automatically reposts the first cross-site POST to the same-origin /licstore5 endpoint.
views/licstore.hbs	Outputs the DRM-X license HTML and return information.
views/licerror.hbs	Displays safe 400, 403, 404, and 502 errors.
package.json	Dependencies, start command, and JavaScript syntax-check command.
README.md	English quick-start instructions.

3.2 DRM-X 5.0 Service Settings

Enter the three website-integration values in config.js and select the international or China service endpoint. Do not append ?wsdl; the code adds it while loading the service definition.

```
DRMX5_ADMIN_EMAIL: 'YOUR_ADMIN_EMAIL',
DRMX5_WEB_SERVICE_AUTH_STR: 'YOUR_WEB_SERVICE_AUTH_STR',
```

```
DRMX5_GROUP_ID: 'YOUR_GROUP_ID',
// International
DRMX5_SERVICE_URL: 'https://5.drm-x.com/haihaisoftlicenseservice.asmx',
// China
// DRMX5_SERVICE_URL: 'https://5.drm-x.cn/haihaisoftlicenseservice.asmx',
```

Setting	Purpose
DRMX5_ADMIN_EMAIL	DRM-X 5.0 administrator email used for SOAP authentication.
DRMX5_WEB_SERVICE_AUTH_STR	Website integration authentication string. Never expose it to the browser or a public repository.
DRMX5_GROUP_ID	DRM-X user group assigned when a new user is created.
DRMX5_SERVICE_URL	International or China DRM-X 5.0 ASMX service endpoint.
DRMX5_TIMEOUT_MS	SOAP client connection/call timeout; the example uses 30,000 milliseconds.

3.3 Demo Account Course and Rights Settings

```
DEMO_USERNAME: 'student',
DEMO_PASSWORD: '123456',
DEMO_EMAIL: 'student@example.com',
DEMO_FULL_NAME: 'Demo Student',
DEMO_PURCHASED_COURSE_IDS: ['1001'],
DEMO_COURSE_BEGIN_DATE: '2026-01-01',
DEMO_COURSE_EXPIRATION_DATE: '2036-01-01',
DEMO_BIND_NUMBER: '1',
```

The course dates are converted to yyyy/MM/dd before UpdateRightWithDisableVirtualMachine is called. ExpirationAfterFirstUse is fixed at -1, so the absolute expiration date controls validity.

4. Login and Express Session Continuation

4.1 Why a Same Origin POST Relay Is Used

When ZJGet first posts from an external page, the browser may omit the existing Session cookie because of SameSite rules. The example renders relay.hbs and automatically posts the validated fields again from the website's own origin, allowing the following request to establish or restore the Session cookie.

> **Important** The relay must forward only validated fields and must never output DRM-X secrets. The example rejects return_url values that use javascript:, data:, or vbscript: schemes.

4.2 Resuming a Request After Login

1. License parameters arrive in a POST request to /licstore5.
2. The server stores them in req.session.drmx5LicenseRequest.
3. An unauthenticated user is sent to /login?resume=1.
4. After the fixed account is validated, user data is stored in req.session.drmx5User.
5. The browser requests /licstore5?resume=1, and the server restores the saved parameters and continues.

Session key	Stored data
drmx5User	Current username, email address, and full name.
drmx5LicenseRequest	License POST parameters and processing state saved before login.

The demo cookie is named drmx5.sid and uses httpOnly=true, sameSite=lax, and a 30-minute lifetime. secure=false is suitable only for local development. Production HTTPS deployments should use secure=true and replace MemoryStore with Redis or a database-backed store.

5. ZJGet POST Endpoint and Parameters

5.1 Integration URL

Local test: http://localhost:3000/licstore5

Production: https://customer-domain.example/licstore5

Configure this URL in the DRM-X 5.0 website integration settings. Do not use /login, /index, or the site root as the license acquisition URL, and do not test the endpoint by opening it directly in a normal browser address bar.

5.2 The Nine POST Parameters

POST field	Source and meaning	Server use
profileid	Current DRM-X 5.0 License Profile ID.	Submitted as ProfileID when requesting the license.
clientinfo	Client license data generated by ZJGet.	Submitted unchanged as ClientInfo.
rightsid	Rights ID selected in the License Profile.	Used for both the rights update and license request.
yourproductid	Product ID stored in the License Profile.	Split into course IDs and checked against the user's purchases.
platform	ZJGet platform information.	Submitted as Platform.
contenttype	Protected content type.	Submitted as ContentType.
version	ZJGet client version.	Submitted as Version.
return_url	Return address or identifier used after licensing.	Validated and handled by the result page.
mac	Device or client binding information.	Submitted as Mac.

The code requires profileid, clientinfo, rightsid, and yourproductid to be non-empty and enforces length limits for all fields. These values come from the ZJGet POST request; users do not enter them manually in a web form.

6. Course ID Validation Rules

yourproductid comes from the DRM-X 5.0 License Profile. When one profile applies to multiple courses, separate the IDs with hyphens, for example 1001-1002-1003. The server checks the posted values in order and selects the first course that the current user owns or has joined.

Courses owned	POST yourproductid	Result
1001, 1003	1001	Allowed; course 1001 is selected.
1001, 1003	1001-1002	Allowed; the first match is 1001.
1001, 1003	1002-1003-1001	Allowed; the first match is 1003.
1001, 1003	1002-1004	Rejected with HTTP 403.

> **Authorization meaning** Matching one course authorizes only the current license request. It does not grant access to any unpurchased course in the posted list. A production system should query order, enrollment, subscription, or LMS data in real time.

7. node-soap and the DRM-X 5.0 WSDL

DRMXService.js creates the SOAP client with soap.createClient and wraps callbacks in Promises. The WSDL address is formed by appending ?wsdl to DRMX5_SERVICE_URL. clientPromise caches the client so the service definition is not reloaded for every request.

```
const wsdlUrl = `${config.DRMX5_SERVICE_URL}?wsdl`;
clientPromise = soap.createClient(wsdlUrl, {
    wsdl_options: { timeout: config.DRMX5_TIMEOUT_MS },
});
```

All six business methods are awaited in sequence. A configuration error, SOAP Fault, unrecognized response, or business failure stops the workflow and returns an appropriate 400, 403, or 502 response.

8. DRM-X 5.0 Method Reference

Order	Method	Condition and result
1	CheckUserExists	Checks whether the website username exists in DRM-X.
2	AddNewUser	Creates a missing user; the result must be 1.
3	CheckUserIsRevoked	Existing users only; a revoked user receives HTTP 403.
4	GetUserDetails	Reads IsLockedOut; a locked user receives HTTP 403.
5	UpdateRightWithDisableVirtualMachine	Updates the posted rightsid; the result must be 1.
6	getLicenseRemoteToTableWithVersionWithMac	Requests license HTML with the same rightsid.

8.1 CheckUserExists

Called after course validation to determine whether the signed-in website username already exists in DRM-X 5.0.

Parameter	Source
AdminEmail	config.js
WebServiceAuthStr	config.js
UserLoginName	Current username in the Session

False, false, or 0 means the user is missing and triggers AddNewUser. True, true, or 1 means the user already exists. Any other value is treated as an error.

8.2 AddNewUser

Called when CheckUserExists reports that the user is missing. It creates a DRM-X account corresponding to the website user.

Parameter group	Example source or value	Purpose
AdminEmail / WebServiceAuthStr / GroupID	config.js	Authenticates the call and assigns the DRM-X user group.
UserLoginName	Session username	DRM-X login name.
UserPassword	N/A	The sample does not use a DRM-X password for login.
UserEmail / UserFullName	Session user data	Email address and full name.
IP	Remote request address	Client IP for the license request.
Money	0	Initial account balance.
BindNumber	DEMO_BIND_NUMBER	Number of permitted device bindings.
IsApproved / IsLockedOut	yes / no	Creates an approved, unlocked user.

AddNewUserResult must equal 1. A newly created user is not checked again for revocation and lockout during the same request.

8.3 CheckUserIsRevoked

Called only when the DRM-X user already existed. It checks whether an administrator has revoked the user. User-license revocation requires a DRM-X Enterprise edition.

Parameter	Source
AdminEmail	config.js
WebServiceAuthStr	config.js

Parameter	Source
UserLoginName	Current username in the Session

A value of true, 1, or yes is treated as revoked. The server returns HTTP 403 and does not update rights or request a license.

8.4 GetUserDetails

Called after an existing user passes the revocation check. The example reads IsLockedOut, including cases where the field is nested under GetUserDetailsResult.

Parameter	Source
AdminEmail	config.js
WebServiceAuthStr	config.js
UserLoginName	Current username in the Session

If IsLockedOut is true or 1, the server returns HTTP 403. A missing field is treated as an abnormal response rather than being allowed by default.

8.5 UpdateRightWithDisableVirtualMachine

Updates the Rights ID supplied in the ZJGet POST field rightsid. The sample updates it for every license request. A production system should decide whether this matches the customer's rights model.

Parameter group	Example value or source	Purpose
AdminEmail / WebServiceAuthStr	config.js	DRM-X service authentication.
RightsID	POST rightsid	Rights ID to update.
Description	Node.js Express Sample	Rights description.
PlayCount	-1	Unlimited play count.

Parameter group	Example value or source	Purpose
BeginDate / ExpirationDate	Configured course dates	Converted from YYYY-MM-DD to yyyy/MM/dd.
ExpirationAfterFirstUse	-1	Disables relative expiration after first use.
RightsPrice	0	Sets the rights price to zero.
AllowPrint / AllowClipBoard / AllowDoc	False	Disables printing, clipboard, and document permissions.
DisableVirtualMachine	True	Disallows use in a virtual machine.
require_admin_permission / bind_1st_screen	False / False	Does not require approval or bind the first screen.

Watermark Blacklist and Screen Recording Protection

Parameter group	Example value	Purpose
EnableWatermark / WatermarkText	True / username	Enables the first watermark and identifies the current user.
WatermarkArea	1,2,3,4,5,	Display areas for the first watermark.
RandomChangeArea / RandomFrequency	True / 12	Randomly changes the watermark area and frequency.
EnableBlacklist / EnableWhitelist	True / True	Enables blacklist and intelligent anti-recording settings.
enable_wm_3rd	1	Enables the third dynamic watermark layer.
enable_wm_2nd	0	Disables the second scrolling layer; related fields are still submitted.

UpdateRightWithDisableVirtualMachineResult must equal 1. These values demonstrate the request structure and are not a universal policy for every customer.

8.6 getLicenseRemoteToTableWithVersionWithMac

After the rights update succeeds, this method generates the final license HTML from the License Profile, client data, user identity, and device information.

Parameter	Source	Purpose
AdminEmail / WebServiceAuthStr	config.js	DRM-X service authentication.
ProfileID	POST profileid	Current License Profile ID.
ClientInfo	POST clientinfo	ZJGet client license data.
RightsID	POST rightsid	The same ID used by the rights update.
UserLoginName / UserFullName	Session user	DRM-X username and full name.
GroupID	config.js	DRM-X user group ID.
IP	Remote request address	Client IP for the license request.
Platform / ContentType / Version / Mac	Matching POST fields	ZJGet platform, content, version, and device data.

The code reads getLicenseRemoteToTableWithVersionWithMacResult as license HTML and also reads Message. Successful HTML must contain license_div_drm-x5. Otherwise, the HTML is stripped from the error text and the server returns HTTP 502.

9. Express Routes and View Responsibilities

Route	Method	Responsibility
/	GET	Redirects to the login page.
/login	GET / POST	Displays the demo login, creates the Session, and resumes a pending request.
/logout	POST	Destroys the current Session.
/api/session	GET	Returns authentication state, user data, and whether a request is pending.
/licstore5	POST	Main ZJGet endpoint: validates data, relays same-origin POST, and issues the license.
/licstore5?resume=1	GET	Restores the license request saved before login.

Route	Method	Responsibility
/index	POST	Compatibility alias for the older sample; do not use it for new settings.

View	Responsibility
login.hbs	Shows login state, the fixed-account form, and safe errors.
relay.hbs	Automatically reposts the first cross-site POST to same-origin /licstore5.
licstore.hbs	Outputs DRM-X license HTML and handles return_url.
licerror.hbs	Shows a safe integration error without exposing credentials.
layout.hbs	Provides shared HTML structure and basic styles.

10. Local Setup and End to End Verification

10.1 Prerequisites

- Node.js 20 or later is recommended; confirm that npm is available.
- Prepare a DRM-X 5.0/ZJGet test environment, a License Profile, a Rights ID, and an encrypted file.
- Confirm that the test computer can reach the selected China or international DRM-X SOAP service.

10.2 Install Check and Start

```
cd "DRM-X5_NodeJS+Express_Integration_Example"
npm install
npm run check
npm start
```

After startup, open http://localhost:3000/login. The default demo username is student and the password is 123456.

10.3 Configure DRM-X 5.0

1. Enter AdminEmail, WebServiceAuthStr, and GroupID in config.js, then select the service region.
2. Add the test course ID to DEMO_PURCHASED_COURSE_IDS.
3. Set the same Product ID in the License Profile for the encrypted test content, such as 1001. Use 1001-1002 for multiple courses.
4. Select custom login page integration in the DRM-X 5.0 website integration settings.
5. Set the license acquisition URL to http://localhost:3000/licstore5. Use a public HTTPS URL in production.

10.4 Verify the Complete Flow

1. Open the encrypted content with ZJGet instead of navigating directly to /licstore5.
2. If signed out, complete login and verify that the page resumes automatically. If already signed in, processing should continue directly.
3. Confirm that the course ID matches and that DRM-X user creation or status checks succeed.
4. Confirm that the rights update succeeds and that ZJGet receives a license and plays the content.

11. Production Integration Recommendations

Demo implementation	Production recommendation
Fixed username and password	Connect the customer's database, SSO, OAuth, or existing identity system and keep the username mapping stable and unique.
Fixed purchased-course list	Query orders, enrollments, subscriptions, or LMS data for the currently authenticated user.
Plain-text config.js	Move real credentials into protected configuration or the customer's secret-management approach; never commit them publicly.

Demo implementation	Production recommendation
MemoryStore Session	Use Redis, a database, or another shared Session store for multi-server deployments.
Development HTTP	Use HTTPS, secure cookies, trusted proxy settings, and an appropriate SameSite policy.
Fixed course dates	Derive BeginDate and ExpirationDate from the real course, order, or subscription record.
Update POST RightsID for every request	Evaluate play count, validity, watermarking, binding, and shared Rights IDs before choosing an update policy.
Direct error page	Keep traceable server logs while returning only necessary information to the client.

Production boundary Course validation must prove that the signed-in user owns at least one course in the POST data; checking only that a course exists is insufficient. Trust forwarded IP headers only after configuring a trusted proxy.

12. Troubleshooting

Symptom	Likely cause	Resolution
Opening /licstore5 shows No saved license request	There is no ZJGet POST and no pending request in the Session.	Open an encrypted file with ZJGet and test the complete flow.
The request does not continue after login	The cookie, Session store, or relay failed.	Check cookies, Session storage, and same-origin URLs. Do not mix localhost and 127.0.0.1.
Configure these values in config.js	DRM-X credentials still contain YOUR_... placeholders.	Set AdminEmail, WebServiceAuthStr, and GroupID.

Symptom	Likely cause	Resolution
WSDL or SOAP connection failure	Wrong region, URL, network, TLS, credentials, or timeout.	Confirm the ASMX URL has no duplicate ?wsdl and verify connectivity and credentials.
HTTP 400	A required POST field is missing, too long, or return_url uses a dangerous scheme.	Check profileid, clientinfo, rightsid, yourproductid, and return_url.
HTTP 403	No course matches, or the DRM-X user is revoked or locked.	Check purchases, the License Profile Product ID, and DRM-X user status.
HTTP 502	Configuration, SOAP, rights update, or license content failed.	Check credentials, GroupID, RightsID, method results, and server logs.
Port 3000 is already in use	Another process is listening on the port.	Change PORT in config.js and update the integration URL.
License HTML is invalid	The service returned error text or an incomplete structure.	Check Message; successful HTML must include license_div_drm-x5.

Appendix File Responsibility Quick Reference

File	Responsibility
config.js	Central DRM-X settings, demo account, dates, purchases, bind count, Session secret, and port.
DRMXService.js	Input validation, same-origin relay, login and Session handling, course matching, SOAP client, and six DRM-X methods.

File	Responsibility
views/layout.hbs	Shared HTML structure and base styles.
views/login.hbs	Demo account login and Session state.
views/relay.hbs	Same-origin repost of the first cross-site POST.
views/licstore.hbs	License HTML output and return handling.
views/licerror.hbs	Safe integration errors.
package.json / package-lock.json	Node.js dependencies, version lock, and start/check commands.
README.md	English quick-start guide.
README-CN.md	Chinese quick-start guide.