

DRM-X 5.0

Java + Vue Custom Integration Development Guide

A complete license workflow sample with a Spring Boot backend and Vue frontend

Document information	Value
Document version	v1.0.0
Sample project	DRM-X5_Java_Vue_Integration_Example
Audience	Java developers, web frontend developers, and system integrators
Coverage	Login Session, course validation, ZJGet POST, SOAP methods, and license response

[Download Sample Code](#)

This guide follows the actual behavior of the sample code. For production deployment, replace the fixed account and fixed courses with the customer's own user, order, or course-authorization logic.

1. Guide Objectives and Sample Scope	3
2. System Architecture and License Workflow	3
3. Project Structure and Configuration	5
4. Login and Session Resume	7
5. ZJGet POST Endpoint and All Parameters	8
6. Course ID Validation Rules	10
7. DRM-X 5.0 SOAP Communication Rules	11
8. DRM-X Method Reference	12
9. Frontend Responsibilities and Page Navigation	20
10. Local Operation and End-to-End Testing	20
11. Production Integration Recommendations	23
12. Troubleshooting	24

1. Guide Objectives and Sample Scope

This sample shows how to integrate the DRM-X 5.0 license service into a web application with a separate frontend and backend. Spring Boot receives the license parameters submitted by ZJGet, validates login and course access, calls the DRM-X 5.0 SOAP service, and returns the license HTML. Vue provides the login interface, checks the Session, and resumes the workflow.

The purpose of the sample is not to build a complete LMS, but to show the DRM-X 5.0 method sequence, parameter sources, and failure handling clearly.

- Fixed-account login for direct testing of the Session workflow.
- A fixed purchased-course list that simulates course purchase or enrollment records in a database.
- Course ID in the License Profile: ZJGet obtains it from the Product ID setting (yourproductid) in the License Profile; it is not entered manually in the frontend.
- Rights ID: ZJGet obtains it from the Rights ID setting (rightsid) in the License Profile. The same value is used both to update rights and to obtain the license.
- DRM-X 5.0 configuration is stored directly in a backend configuration file; the sample does not depend on environment variables.

Important The account, password, course ID, validity period, and binding count in this sample are for demonstration only. A production project must connect to a real identity system and an order or enrollment database, and must protect the DRM-X 5.0 WebServiceAuthStr integration credential.

2. System Architecture and License Workflow

Component	Responsibility
DRM-X 5.0 License Profile	Stores profile information such as Course ID; DRM-X reads it during playback and passes it to ZJGet.
ZJGet browser	Opens the encrypted video and POSTs profileid, rightsid, yourproductid, clientinfo, and other parameters to the integration URL.

Component	Responsibility
Spring Boot	Handles /licstore5, HttpSession, course authorization, and DRM-X 5.0 SOAP calls.
Vue	Provides fixed-account login, queries the backend Session, and completes the license request after login.
DRM-X 5.0 SOAP service	Manages users, status, and rights, and generates license HTML.

1. ZJGet reads the current encrypted file and obtains its DRM-X 5.0 License Profile information.
2. ZJGet sends a POST request to backend /licstore5. Both yourproductid and rightsid come from this request.
3. The backend first performs a same-origin POST relay so the browser can include an existing Session Cookie.
4. The backend validates the parameters and stores the license request temporarily in HttpSession.
5. If the user is not logged in, the backend redirects to Vue /login?resume=1. If the user is already logged in, processing continues.
6. The backend splits yourproductid on hyphens and checks whether the user has purchased at least one listed course. Normally, a License Profile contains only one course ID.
7. The backend calls CheckUserExists and calls AddNewUser when necessary.
8. For an existing DRM-X user, the backend calls CheckUserIsRevoked and GetUserDetails to reject a revoked or locked account.
9. The backend calls UpdateRightWithDisableVirtualMachine to update the rights identified by the posted rightsid.
10. The backend calls getLicenseRemoteToTableWithVersionWithMac to obtain the license HTML and return it to ZJGet.

3. Project Structure and Configuration

3.1 Main Directories

Path	Purpose
backend/config/drmx5-config.yml	DRM-X account, authentication string, GroupID, service URL, and timeouts.
backend/src/main/resources/application.yml	Port, Session, sample account, course list, validity period, and frontend URL.
backend/src/main/java/.../web	License endpoint, authentication API, and license HTML pages.
backend/src/main/java/.../service	Course validation, workflow, and DRM-X SOAP client.
backend/src/main/java/.../model	Models for ZJGet POST requests, users, and license results.
frontend/src/App.vue	Login, Session checking, and license workflow resume.
frontend/src/config.js	Base URL used by the frontend to access the backend.

3.2 DRM-X Service Configuration

```
drmx5:
  admin-email: "YOUR_ADMIN_EMAIL"
  web-service-auth-str: "YOUR_WEB_SERVICE_AUTH_STR"
  group-id: "YOUR_GROUP_ID"
  service-url: "https://5.drm-x.cn/haihaiisoftlicenseservice.asmx"
  connect-timeout-seconds: 20
  request-timeout-seconds: 30
```

Setting	Required	Description
admin-email	Yes	DRM-X 5.0 administrator email corresponding to the SOAP parameter AdminEmail.
web-service-auth-str	Yes	DRM-X Web Service authentication string. This is sensitive and must remain on the backend.
group-id	Yes	DRM-X user Group ID used when creating a user and obtaining a license.
service-url	Yes	ASMX service URL without ?wsdl. Use 5.drm-x.com for the International service and 5.drm-x.cn for the China service.
connect-timeout-seconds	No	Connection timeout in seconds. The default is 20 seconds when the value is invalid or less than or equal to 0.
request-timeout-seconds	No	SOAP request timeout in seconds. The default is 30 seconds when the value is invalid or less than or equal to 0.

International WSDL: <https://5.drm-x.com/haihaisoftlicenseservice.asmx?wsdl>

China WSDL: <https://5.drm-x.cn/haihaisoftlicenseservice.asmx?wsdl>

3.3 Sample Application Configuration

```
server:
  port: 8080
  servlet:
    session:
      timeout: 30m
    cookie:
      http-only: true
      secure: false
      same-site: lax

demo:
  account:
    username: student
    password: "123456"
  course:
    purchased-course-ids: 1001
    begin-date: 2026-01-01
    expiration-date: 2036-01-01
  frontend-base-url: http://localhost:5173
```

Production environment After enabling HTTPS, set the Session Cookie secure option to true and change frontend-base-url and frontend/src/config.js to the actual domain. If the frontend and backend use the same domain, set frontend BACKEND_BASE_URL to an empty string.

4. Login and Session Resume

The sample backend stores two key objects in HttpSession: the logged-in user and the pending license request. When the user opens an encrypted video for the first time, a cross-site POST may not include the Session Cookie. The backend therefore returns a relay page that automatically submits to same-origin /licstore5 before reading the Session.

Session content	Constant	Purpose
Logged-in user	DRMX5_DEMO_USER	Stores DemoUser after successful login; later license requests use it to determine whether the user is logged in.
Pending license request	DRMX5_PENDING_LICENSE	Temporarily stores LicenseRequest before login and restores it from the Session after login.

Endpoint	Method	Description
/api/auth/session	GET	Returns authenticated, the username, and whether a pending license request exists.
/api/auth/login	POST	Validates the fixed account, changes the Session ID after login, and stores the user.
/api/auth/logout	POST	Invalidates the current Session.
/licstore5?resume=1	GET	Reads the original license request from the Session and continues without requiring another POST from ZJGet.

Security design changeSessionId reduces Session Fixation risk. License responses use Cache-Control: no-store so the browser does not cache sensitive results.

5. ZJGet POST Endpoint and All Parameters

5.1 Integration URL

The integration URL configured in DRM-X 5.0 must point directly to the Spring Boot backend, not to the Vue development server.

Local test: <http://localhost:8080/licstore5>

Production: <https://your-domain.example/licstore5>

This endpoint accepts application/x-www-form-urlencoded data. The Vue page does not provide input fields for these DRM-X parameters because DRM-X 5.0 and ZJGet generate and submit them automatically from the current encrypted content.

5.2 Nine Parameters Submitted by ZJGet

Parameter	Required by code	Source and meaning	Handling in the sample
profileid	Yes	License Profile ID	Passed unchanged as ProfileID to the license method.
clientinfo	Yes	Client information obtained by ZJGet	Client data required by DRM-X 5.0 to generate a license.
rightsid	Yes	Rights ID in the License Profile	Used to update rights and obtain the license.
yourproductid	Yes	Course ID in the License Profile	Supports one or more course IDs separated by hyphens; used for local course-access validation.
platform	Yes	ZJGet platform identifier	Passed unchanged as Platform to the license method.
contenttype	Yes	Protected content type	Passed unchanged as ContentType.
version	Yes	ZJGet browser version	Passed unchanged as Version.
return_url	Yes	Return URL or flag after license completion	Used for navigation after the license page completes. It must be placed in the Title element of the licstore5 HTML so ZJGet can recognize it automatically.
mac	Yes	Device or client binding information	Passed unchanged as Mac to the license method.

Required-field rule

LicenseRequest.validate requires profileid, clientinfo, yourproductid, and rightsid to be non-empty.

5.3 Sample POST Request

POST /licstore5 HTTP/1.1

Content-Type: application/x-www-form-urlencoded

```
profileid=PROFILE_001&clientinfo=ZJGET_GENERATED_DATA&rights=RIGHT_100
&yourproductid=1001-1002&platform=Windows&contenttype=video
&version=5.0&return_url=https%3A%2F%2Fexample.com%2Fcourse&mac=DEVICE_DATA
```

The values above only illustrate the field structure. The actual request must be submitted automatically by ZJGet and must not be forged.

5.4 Internal Sample Parameters

Parameter	Generated by	Purpose
_drmX5SameSiteRelay	Backend relay page	A value of 1 indicates that the request has completed the same-origin relay. It is not a DRM-X or ZJGet business parameter and is not sent to the SOAP service.
resume=1	Vue and backend	After login, GET /licstore5?resume=1 resumes the license request stored in the Session. It is not a license parameter.

6. Course ID Validation Rules

yourproductid represents the course ID configured in the DRM-X License Profile.

```
# Courses purchased by the sample user in application.yml
demo:
  course:
    purchased-course-ids: 1001,1003
```

POST yourproductid	Purchased list	Result
1001	1001,1003	Pass; use course 1001.
1001-1002	1001,1003	Pass; select the first matching course, 1001, in POST order.
1002-1003	1001,1003	Pass; 1002 does not match, so 1003 is selected.
1002-1004	1001,1003	Deny; no course matches.

The implementation splits yourproductid on hyphens, trims surrounding whitespace, skips empty values, and returns the first ID found in the purchased-course set. Multiple course IDs mean that the user needs to own any one of them to pass this license request. They do not grant access to the other courses in the list.

Production system Replace DemoCourseService with a database or LMS API query. Validation must bind both the current logged-in user and the course ID; checking only whether the course exists is insufficient.

7. DRM-X 5.0 SOAP Communication Rules

The backend uses Java HttpClient to send SOAP 1.1 XML directly. Method names, parameter names, and capitalization must match the ASMX service.

Item	Value or rule
Service URL	drmx5.service-url ending with haihaisoftlicenseservice.asmx and excluding ?wsdl.
SOAP namespace	http://tempuri.org/ASPNET.StarterKit.Commerce/HaihaisoftLicenseService
SOAPAction	Namespace + / + method name, sent as a quoted HTTP header.
Content-Type	text/xml; charset=UTF-8
XML security	Disable DOCTYPE and external entities to reduce XXE risk.

Item	Value or rule
Timeouts	Connection and request timeouts are read from drmx5-config.yml.

Common error If SOAPAction uses `http://tempuri.org/CheckUserExists`, the server returns “Server did not recognize the value of HTTP Header SOAPAction.” Use the complete namespace shown in the table above.

8. DRM-X Method Reference

LicenseWorkflowService calls the following methods in a fixed sequence. Apart from creating a new user, all rights and license calls use the DRM-X username mapped from the current website or business-system user.

Order	Method	Condition	Failure handling
1	CheckUserExists	Every license request	A response other than true/false or 1/0 is treated as an error.
2	AddNewUser	DRM-X user does not exist	Stop when the result is not 1.
3	CheckUserIsRevoked	DRM-X user exists	Deny the license when the user is revoked.
4	GetUserDetails	DRM-X user exists	Deny the license when IsLockedOut is true.
5	UpdateRightWithDisableVirtualMachine	User is active	Stop when the result is not 1.
6	getLicenseRemoteToTableWithVersionWithMac	Rights update succeeded	Treat the result as a failure when license_div_drm-x5 is missing.

8.1 CheckUserExists

When called: after course validation succeeds

Purpose: checks whether the current website user already exists in DRM-X 5.0.

Parameter	Source	Description
UserName	Current logged-in username	DRM-X login name and the common user identifier used by later methods.
AdminEmail	drmx5.admin-email	DRM-X administrator email.
WebServiceAuthStr	drmx5.web-service-auth-str	Web Service authentication string.

Read CheckUserExistsResult. true or 1 means the user exists; false or 0 means the user does not exist.

8.2 AddNewUser

When called: when CheckUserExists reports that the user does not exist

Purpose: creates a DRM-X 5.0 account corresponding to the current website user.

Parameter	Sample source or value	Description
AdminEmail	Configuration	Administrator email.
WebServiceAuthStr	Configuration	Web Service authentication string.
GroupID	Configuration	DRM-X user group assigned to the new user.
UserLoginName	User username	DRM-X login name.
UserPassword	N/A	The sample does not use a DRM-X password for login.
UserEmail	User email	User email address.

Parameter	Sample source or value	Description
UserFullName	User fullName	User's full name.
Title	N/A	The sample does not collect the user's job title.
Company	N/A	The sample does not collect the company name.
Address	N/A	The sample does not collect the address.
City	N/A	The sample does not collect the city.
Province	N/A	The sample does not collect the state or province.
ZipCode	N/A	The sample does not collect the postal code.
Phone	N/A	The sample does not collect the phone number.
CompanyURL	N/A	The sample does not collect the company website.
SecurityQuestion	N/A	The sample does not use a DRM-X password-recovery question.
SecurityAnswer	N/A	The sample does not use a DRM-X password-recovery answer.
IP	HTTP remote address	Remote IP that initiated the license request.
Money	0	Initial balance for the new user.
BindNumber	1	Binding count allowed by the sample; adjust it to the production business rules.
IsApproved	yes	Approve immediately after creation.
IsLockedOut	no	Do not lock the account when it is created.

Read `AddNewUserResult`. The sample requires a result of 1; otherwise, the license workflow stops. A newly created user is not checked again for revocation or lockout in the same request.

8.3 CheckUserIsRevoked Enterprise Edition Only

When called: only when the DRM-X user already existed

Purpose: determines whether an administrator has revoked the user.

Parameter	Source	Description
AdminEmail	Configuration	DRM-X administrator email.
WebServiceAuthStr	Configuration	Web Service authentication string.
UserLoginName	Current username	DRM-X username to check.

Read CheckUserIsRevokedResult. true, 1, or yes is treated as revoked; the backend returns 403 and denies the license request.

8.4 GetUserDetails

When called: after CheckUserIsRevoked passes

Purpose: reads DRM-X user details and checks the lockout state.

Parameter	Source	Description
AdminEmail	Configuration	DRM-X administrator email.
WebServiceAuthStr	Configuration	Web Service authentication string.
UserLoginName	Current username	DRM-X username to query.

GetUserDetails returns detailed user information. The sample reads only the IsLockedOut field; true or 1 means the user is locked.

8.5 UpdateRightWithDisableVirtualMachine

When called: when the user exists and is not revoked or locked, or immediately after successful creation

Purpose: updates the rights template identified by the posted rightsid.

Identity and Basic Rights Parameters

Parameter	Sample value or source	Description
AdminEmail	Configuration	DRM-X administrator email.
WebServiceAuthStr	Configuration	Web Service authentication string.
RightsID	POST rightsid	Rights ID to update.
Description	Java Vue DRM-X 5.0 sample	Rights description.
PlayCount	-1	The sample sets unlimited playback count.
BeginDate	demo.course.begin-date	Formatted as yyyy/MM/dd before submission.
ExpirationDate	demo.course.expiration-date	Formatted as yyyy/MM/dd before submission.
ExpirationAfterFirstUse	-1	Do not use a relative expiration period after first playback.
RightsPrice	0	Rights price is 0.

Content Control and Preview Parameters

Parameter	Sample value	Description
AllowPrint	False	Disable printing.
AllowClipBoard	False	Disable clipboard access.
AllowDoc	False	Disable document-related permissions.

Parameter	Sample value	Description
EnableWatermark	True	Enable the first watermark.
WatermarkText	User email or username	Display text for the first watermark.
WatermarkArea	1,2,3,4,5,	Display area for the first watermark.
RandomChangeArea	True	Randomly change the first watermark area.
RandomFrquency	12	Random-change frequency for the first watermark.
EnableBlacklist	True	Enable blacklist control.
EnableWhitelist	True	Enable playback-only screen recording control (Enterprise Edition).
ExpireTimeUnit	Day	Expiration time unit.
PreviewTime	3	Allow preview for 3 time units. Not currently effective.
PreviewTimeUnit	Day	Preview time unit is Day. Not currently effective.
PreviewPage	3	Preview page count for document content. Not currently effective.
DisableVirtualMachine	True	Disable use in a virtual machine.
require_admin_permission	False	Whether ZJGet must run with administrator privileges.
bind_1st_screen	False	Whether to enable first-screen binding.

Third-Layer Dynamic Watermark Parameters

Parameter	Sample value	Description
enable_wm_3rd	1	Enable the third-layer watermark.
wm_3rd_random_change_angle	1	Randomly change the watermark angle.
wm_3rd_textalpha	90	Watermark text opacity parameter.
wm_3rd_rows	3	Number of watermark rows.
wm_3rd_columns	3	Number of watermark columns.
wm_3rd_fixed_angle	0	Fixed angle value.
wm_3rd_textsize	14	Text size.
wm_3rd_textcolor	16744576	Text color represented as a decimal value.
wm_3rd_angle_change_interval	10	Angle-change interval.
wm_3rd_text	User email or username	Third-layer watermark text.

Second-Layer Scrolling Watermark Parameters

Parameter	Sample value	Description
enable_wm_2nd	0	Disable the second-layer watermark.
wm_2nd_text	User email or username	Second-layer watermark text; it is submitted as part of the method structure even when disabled.
wm_2nd_speed	1	Scrolling speed.
wm_2nd_textsize	22	Text size.
wm_2nd_textcolor	16744576	Text color represented as a decimal value.
wm_2nd_textalpha	90	Opacity parameter.

Read `UpdateRightWithDisableVirtualMachineResult`. License retrieval continues only when the result is 1. These values are a demonstration policy and are not appropriate for every customer.

8.6 getLicenseRemoteToTableWithVersionWithMac

When called: after the rights update succeeds

Purpose: generates the final license from the profile, client information, rights, and user information.

Parameter	Source	Description
AdminEmail	Configuration	DRM-X administrator email.
WebServiceAuthStr	Configuration	Web Service authentication string.
ProfileID	POST profileid	Current DRM-X License Profile ID.
ClientInfo	POST clientinfo	Client license data generated by ZJGet.
RightsID	POST rightsid	Uses the same Rights ID as <code>UpdateRightWithDisableVirtualMachine</code> .
UserLoginName	Current username	DRM-X user login name.
UserFullName	Current user fullName	User's full name.
GroupID	Configuration	DRM-X user Group ID.
Message	N/A	ASMX input/output parameter; the service response message is displayed on the result page.
IP	HTTP remote address	Remote IP for the license request.
Platform	POST platform	ZJGet platform information.
ContentType	POST contenttype	Content type.
Version	POST version	ZJGet client version.
Mac	POST mac	Device binding information.

The sample reads `getLicenseRemoteToTableWithVersionWithMacResult` as the license HTML and also reads `Message`. To avoid treating error text as a license, the code requires the HTML to contain `license_div_drm-x5`.

9. Frontend Responsibilities and Page Navigation

The Vue frontend does not call the DRM-X SOAP service and does not collect `profileid`, `clientinfo`, `rightsid`, or `yourproductid`. It only manages the user experience through the backend authentication API.

1. Call `GET /api/auth/session` when the page loads.
2. If `authenticated=true` and `hasPendingLicenseRequest=true`, navigate to backend `/licstore5?resume=1`.
3. If the user is not logged in, show the fixed-account login form.
4. Submit `POST /api/auth/login?resume=1` and use credentials: include to send the Cookie.
5. After the backend returns `resumeUrl`, the browser navigates to it and continues the license request.

Key point When the frontend and backend use separate ports in development, `BACKEND_BASE_URL` is `http://localhost:8080`. The DRM-X integration URL also points to `/licstore5` on port 8080, not 5173. The license page or route must be named `licstore5`.

10. Local Operation and End-to-End Testing

10.1 Requirements

- JDK 17 or a compatible version.
- Apache Maven 3.9 or Maven provided by the IDE.
- Node.js 20 or a compatible version, and npm.
- Network access to the selected DRM-X 5.0 service domain.

10.2 Start the Backend

```
cd DRM-X5_Java_Vue_Integration_Example/backend
mvn spring-boot:run
```

After successful startup, Spring Boot listens at `http://localhost:8080`. If port 8080 is already in use, stop the process using it or change `server.port` in `application.yml` and update the frontend configuration and integration URL accordingly.

10.3 Start the Frontend

```
cd DRM-X5_Java_Vue_Integration_Example/frontend
npm install
npm run dev
```

The development server normally listens at `http://localhost:5173`. The sample account is read from `application.yml`; the default username is `student` and the default password is `123456`.

10.4 Configure and Test

1. Enter the test account's `AdminEmail`, `WebServiceAuthStr`, and `GroupID` in `drmx5-config.yml`, and select the correct International or China service URL.

The screenshot shows the 'Profile - Website Integration Preferences' dialog in the DRM-X 5.0 web interface. The left sidebar contains navigation links: Users, User Groups, License Profiles, Rights, Reports, Cloud Storage, Account Settings (highlighted), and Protect Files. The main content area shows the 'Integration Method' section with two options: 'Synchronize user integration' (unselected) and 'Custom login page integration' (selected). Below this, there is a text description for each method. A red box highlights the 'Web Service Authentication String' input field, which is currently empty. Below that, the 'License URL' is set to 'http://localhost:8080/licstore5'. At the bottom, there are 'Submit' and 'Cancel' buttons.

2. Add the course ID that the DRM-X License Profile will submit to `purchased-course-ids` in `application.yml`.
3. Set the Product ID in the DRM-X 5.0 License Profile for the test encrypted video to the corresponding course ID.

Security Version*	2
ProductID in your system(Optional):	1001
Created Date:	9/3/2026 9:48:57 PM
Updated Date:	9/9/2026 6:35:09 AM

4. In DRM-X 5.0, open Account Settings > Website Integration Settings > Custom Login Page Integration, and set the URL to `http://localhost:8080/licstore5`.

Profile - Website Integration Preferences

Integration Method: [Integration Help](#)

☐ Synchronize user integration
 ☒ Custom login page integration

Synchronize user integration: Sync your website's user data to the DRM-X platform. The license acquisition method and login page will continue to use the default DRM-X 5.0 login interface.

Custom login page integration: When users obtain a license, they will be redirected to your custom page for verification. You can call DRM-X 5.0 service APIs to implement your business logic and deliver licenses to users. This method requires access to your website source code and database support.

Please set a Web Service Authentication String for DRM-X web service (DRM-X need you provide this string to access DRM-X web service.)

Please enter License URL(Must begin with `http://` or `https://`) (License URL is the url used by pop window when acquiring license)

`http://localhost:8080/licstore5`

Submit Cancel

- Open the encrypted video in ZJGet. Do not visit `/licstore5` directly in a normal browser address bar because the normal workflow requires POST parameters.
- If the user is not logged in, the page navigates to the Vue login page. After successful login, the license request should resume automatically.
- Confirm that DRM-X user creation or status checks succeed, the rights settings are updated, and ZJGet obtains the license and plays the content.

11. Production Integration Recommendations

Sample implementation	Production recommendation
Fixed username and password	Integrate with the customer's existing SSO, Spring Security, JWT, or server Session. Do not hard-code passwords in the frontend.
Fixed purchased-course-ids	Query orders, enrollment, subscriptions, or course-group membership for the current logged-in user.
Use username directly as DRM-X username	Define a stable, unique mapping strategy to avoid duplicate DRM-X accounts when a user changes their username.
Plaintext authentication string in configuration	At minimum, restrict file permissions. Customers needing stronger protection can move the value to a secret-management service or environment variable.
HTTP in development	Production must use HTTPS and enable Secure Cookie.
Fixed rights-update policy	Adjust playback count, validity period, watermark, binding count, and virtual-machine restrictions to the customer's product policy.
Use remoteAddr as IP	When using a trusted reverse proxy, parse the forwarded real client IP according to a secure proxy configuration.
Display errors directly	Record detailed server logs in production and return only necessary, non-sensitive information to the client.

Concurrency note The sample updates the RightsID received by POST for every license request. If multiple License Profiles share one RightsID, the production design must evaluate the risk of concurrent updates and incorrect license settings.

12. Troubleshooting

Symptom	Cause	Resolution
Direct access to /licstore5 displays No saved license request	There is no ZJGet POST and no pending request in the Session.	Open the encrypted file with ZJGet and test the complete workflow.
/login?resume=1 remains on Checking Session	Vue cannot access the backend, the port is wrong, or the request does not include the Cookie.	Check BACKEND_BASE_URL, the service on port 8080, and credentials: include.
SOAPAction is not recognized	The wrong tempuri namespace was used.	Use the complete HaihaisoftLicenseService namespace and method name.
Returns 400	A required POST parameter is missing or return_url uses an unsafe scheme.	Check profileid, clientinfo, rightsid, and yourproductid.
Returns 403	The course does not match, or the DRM-X user is revoked or locked.	Verify the purchased-course list and DRM-X user status.
Returns 502	DRM-X configuration, network, SOAP, or response content is invalid.	Check the service region, authentication fields, timeouts, and backend logs.
Port 8080 already in use	Another process is listening on port 8080.	Stop that process or change server.port, then update the related URLs.
mvn is not recognized	Maven is not installed or is not in PATH.	Use Maven provided by the IDE, Maven Wrapper, or the full Maven executable path.
Course 1001-1002 validation fails	The profile and purchased-course list do not intersect, or the IDs use the wrong meaning.	Configure actual Course IDs in yourproductid and separate multiple values with hyphens.
License HTML is invalid	DRM-X returned error text or incomplete structure.	Check the service Message; successful content must contain license_div_drm-x5.

Appendix File and Responsibility Reference

Class or file	Responsibility
LicenseController	Receives /licstore5 POST, performs the same-origin relay, stores pending requests, redirects to login, and resumes the workflow.
LicenseRequest	Reads, sanitizes, truncates, and validates the nine ZJGet POST parameters.
DemoCourseService	Parses yourproductid and matches it against the purchased-course set.
LicenseWorkflowService	Orchestrates user handling, status checks, rights updates, and license retrieval.
DrmxSoapClient	Builds secure SOAP XML, sets SOAPAction, calls six DRM-X methods, and parses the results.
AuthController	Provides Session query, login, and logout APIs.
DemoAuthService	Validates the fixed account and defines Session attribute names.
HtmlPages	Generates the same-origin POST relay page, license result page, and error page.
App.vue	Displays the login page, checks the Session, and resumes the license request after login.
drmx5-config.yml	Stores confidential DRM-X server configuration.
application.yml	Stores the sample account, courses, dates, port, and Session settings.