DRM-X 5.0

Python + Django 自定义集成开发文档

基于 Django Session 与 zeep 的完整许可证流程示例

文档信息	内容
文档版本	v1.0.0
示例项目	DRM-X5_Python_Django_Integration_Example
适用对象	Python 开发人员、Django 开发人员、系统集成人员
覆盖范围	登录 Session、课程验证、ZJGet POST、zeep SOAP 方法、许可证返回

> **说明** 本教程以已经测试通过的演示代码为准。固定账号、固定课程和明文配置仅用于说明集成流程；正式部署时必须接入客户自己的登录与课程购买数据。

下载样例代码

目录

1、文档目标与示例边界 3

2、系统结构与许可证流程 3

3、项目结构与配置 4

4、登录与 Django Session 续接 6

5、ZJGet POST 入口与参数 7

6、课程 ID 验证规则 8

7、zeep 与 DRM-X 5.0 WSDL 9

8、DRM-X 5.0 接口方法详解 10

9、Django 路由与页面职责 15

10、本地运行与演示验证 16

11、正式项目改造建议 18

12、常见问题与故障排查 19

附录：文件与职责速查 20

1、文档目标与示例边界

本示例演示如何把 DRM-X 5.0 许可证服务接入一个 Django 网站。Django 接收 ZJGet 浏览器自动提交的许可证参数，检查登录 Session 和课程权限，然后使用 zeep 调用 DRM-X 5.0 SOAP 服务并返回许可证 HTML。

演示代码的重点是许可证流程，而不是构建完整 LMS。示例采用固定账号和固定已购买课程列表，便于开发人员快速运行、调试和理解每一步。

- 固定账号登录：student / 123456，用于验证登录与 Session 恢复。
- 固定已购买课程：在 config.py 中模拟订单、选课或订阅记录。
- 课程 ID：由 License Profile 的产品 ID（yourproductid）通过 ZJGet POST 提交，不在许可证入口中固定。
- 权限 ID：更新权限和获取许可证都使用同一次 POST 中的 rightsid。
- DRM-X 配置：直接保存在 drmx5_demo/config.py，演示代码不依赖环境变量。

2、系统结构与许可证流程

组件	职责
DRM-X 5.0 许可证模板	保存产品 ID（课程 ID）和 RightsID；播放时由 DRM-X 读取并交给 ZJGet。
ZJGet 浏览器	打开加密视频，并把 profileid、clientinfo、rightsid、yourproductid 等参数 POST 到集成地址。
Django 集成入口	处理 /licstore5、同源中继、登录 Session、课程验证和许可证响应。
zeep SOAP 客户端	从 WSDL 创建客户端，并按参数名调用 DRM-X 5.0 方法。
DRM-X 5.0 服务	检查或创建用户、检查吊销/锁定、更新权限并生成许可证 HTML。

1. ZJGet 读取当前加密文件对应的许可证模板信息，并 POST /licstore5。
2. Django 先执行同源 POST 中继，让浏览器在后续请求中携带网站 Session Cookie。

3. 后端验证参数，并把许可证请求保存到 Django Session。
4. 未登录时显示登录页；登录成功后，从 Session 恢复请求并继续。
5. 按连字符拆分 yourproductid，验证当前用户是否购买其中至少一个课程。
6. 调用 CheckUserExists；不存在时调用 AddNewUser。
7. 已有用户继续调用 CheckUserIsRevoked 和 GetUserDetails，拒绝吊销或锁定用户。
8. 使用 POST rightsid 调用 UpdateRightWithDisableVirtualMachine。
9. 使用同一个 POST rightsid 调用 getLicenseRemoteToTableWithVersionWithMac。
10. 验证返回 HTML 包含 license_div_drm-x5，随后返回给 ZJGet。

3、项目结构与配置

3.1 主要目录

```
DRM-X5_Python_Django_Integration_Example/
├─ drmx5_demo/
│  ├─ config.py          # DRM-X、测试账号和课程配置
│  ├─ settings.py        # Django 与 Session 配置
│  └─ urls.py
├─ integration/
│  ├─ views.py           # 登录、课程验证和全部 DRM-X 调用流程
│  ├─ urls.py
│  ├─ templates/         # 登录、中继、许可证和错误页面
├─ manage.py
└─ requirements.txt
```

核心业务逻辑集中在 integration/views.py，便于开发人员按执行顺序阅读和移植。

3.2 DRM-X 5.0 服务配置

```
DRMX5_ADMIN_EMAIL = "YOUR_ADMIN_EMAIL"
DRMX5_WEB_SERVICE_AUTH_STR = "YOUR_WEB_SERVICE_AUTH_STR"
```

```
DRMX5_GROUP_ID = "YOUR_GROUP_ID"
# 中国版
DRMX5_SERVICE_URL = "https://5.drm-x.cn/haihaisoftlicenseservice.asmx"
# 国际版
# DRMX5_SERVICE_URL = "https://5.drm-x.com/haihaisoftlicenseservice.asmx"
DRMX5_TIMEOUT_SECONDS = 30
```

配置项	必需	说明
DRMX5_ADMIN_EMAIL	是	DRM-X 5.0 管理员邮箱，对应 SOAP 的 AdminEmail。
DRMX5_WEB_SERVICE_AUTH_STR	是	网站集成验证码；属于敏感信息，只能保存在服务器端。
DRMX5_GROUP_ID	是	创建用户和获取许可证时使用的 DRM-X 用户组 ID。
DRMX5_SERVICE_URL	是	填写 .asmx 服务地址，不包含 ?wsdl；代码会自动添加。
DRMX5_TIMEOUT_SECONDS	否	WSDL 与 SOAP 请求超时，示例为 30 秒。

> **服务区域：**
>
> 国际版使用 5.drm-x.com，中国版使用 5.drm-x.cn。
>
> 配置值必须以 haihaisoftlicenseservice.asmx 结尾。

3.3 演示账号、课程与权限配置

```
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"
```

配置项	用途
DEMO_USERNAME / DEMO_PASSWORD	固定登录账号，仅用于测试 Session 流程。
DEMO_EMAIL / DEMO_FULL_NAME	创建 DRM-X 用户、水印文本及获取许可证时使用。
DEMO_PURCHASED_COURSE_IDS	模拟当前用户已经购买或加入的课程 ID 集合。
DEMO_COURSE_BEGIN_DATE	更新权限的开始日期，格式必须为 YYYY-MM-DD。
DEMO_COURSE_EXPIRATION_DATE	更新权限的截止日期，格式必须为 YYYY-MM-DD。
DEMO_BIND_NUMBER	AddNewUser 创建用户时设置的设备绑定数量。

4、登录与 Django Session 续接

Django Session 保存两个关键对象。文件型 Session 将数据保存在服务器的 .sessions 目录，可避免 clientinfo 较大时超过 Cookie 容量。正式系统可改用数据库或缓存 Session。

Session 键	保存内容	用途
drmx5_user	username、email、full_name	判断用户是否登录，并为 DRM-X 用户与许可证提供身份信息。
drmx5_license_request	验证后的九个 ZJGet 参数	未登录时暂存请求，登录成功后恢复许可证流程。

4.1 为什么需要同源 POST 中继

ZJGet 可能从不同来源 POST 到网站。浏览器采用 SameSite=Lax Cookie 时，第一次跨站 POST 可能不携带已有 Session Cookie。licstore5 首次收到请求后返回 relay.html，由该页面自动向同一站点再次 POST，并附加内部标记 _drmx5_relay=1。第二次请求再读取 Session，可识别用户已有登录状态。

> **内部参数：** _drmx5_relay 只用于标记同源中继，不是 DRM-X 或 ZJGet 业务参数，也不会传给 SOAP 服务。

4.2 未登录后的恢复流程

1. 许可证请求保存到 drmx5_license_request。
2. 后端跳转到 /login?resume=1。
3. 用户使用测试账号登录，Django cycle_key() 更换 Session ID。
4. 登录成功后跳转到 /licstore5?resume=1。
5. 后端从 Session 读取原请求并继续，不要求 ZJGet 再次 POST。

> **Session 设置：**
>
> 示例使用 SESSION_COOKIE_HTTPONLY=True、SESSION_COOKIE_SAMESITE='Lax'、30 分钟有效期。生产环境启用 HTTPS 后应设置 SESSION_COOKIE_SECURE=True。

5、ZJGet POST 入口与参数

5.1 集成地址

DRM-X 5.0 后台的许可证获取 URL 应直接指向 Django 后端：

```
本地测试：http://localhost:8080/licstore5
正式部署：https://your-domain.example/licstore5
```

该入口接收 application/x-www-form-urlencoded POST。正常情况下参数由 DRM-X 5.0 和 ZJGet 根据当前加密内容自动生成，网页不需要也不应该提供手工输入表单。

5.2 九个 POST 参数

参数	代码必填	来源和含义	示例处理
profileid	是	当前许可证模板 ID。	原样作为 ProfileID 获取许可证。
clientinfo	是	ZJGet 生成的客户端许可证数据。	原样作为 ClientInfo；最大读取 8192 字符。
rightsid	是	License Profile 中选择的权限 ID。	同时用于更新权限与获取许可证。
yourproductid	是	License Profile 中配置的产品 ID；本示例解释为课程 ID。	按 - 拆分并与已购买课程匹配。
platform	是	ZJGet 平台信息。	原样作为 Platform。
contenttype	是	受保护内容类型。	原样作为 ContentType。
version	是	ZJGet 客户端版本。	原样作为 Version。
return_url	是	获取许可证后的返回地址或标识。	显示在结果页并供完成后返回；拒绝危险协议。
mac	是	设备或客户端绑定信息。	原样作为 Mac。

6、课程 ID 验证规则

yourproductid 来自当前加密内容的 DRM-X 5.0 许可证模板。演示代码按连字符拆分该值，然后按 POST 顺序寻找第一个出现在 DEMO_PURCHASED_COURSE_IDS 中的课程。

```
DEMO_PURCHASED_COURSE_IDS = ("1001", "1003")
```

POST yourproductid	已购买列表	结果
1001	1001, 1003	通过，选中课程 1001。
1001-1002	1001, 1003	通过，按 POST 顺序选中 1001。
1002-1003-1001	1001, 1003	通过，先遇到 1003，因此选中 1003。

POST yourproductid	已购买列表	结果
1002-1004	1001, 1003	拒绝，没有任何已购买课程匹配。

> **权限边界：**列表中只要有一个课程匹配，就允许本次许可证请求；这不表示用户获得 yourproductid 列表中其他课程的访问权限。正式系统必须同时绑定当前登录用户和真实订单、选课或订阅记录。

7、zeep 与 DRM-X 5.0 WSDL

示例使用 zeep 根据 WSDL 自动生成 SOAP 请求。get_drmx_client() 第一次真正获取许可证时加载 WSDL，并通过 lru_cache(maxsize=1) 复用客户端。与手工拼接 XML 相比，zeep 会根据 WSDL 处理参数结构和 SOAPAction。

项目	规则
配置地址	DRMX5_SERVICE_URL 保存 .asmx 地址，不包含 ?wsdl。
WSDL 地址	代码自动使用 DRMX5_SERVICE_URL + '?wsdl'。
参数传递	所有方法按 WSDL 参数名使用关键字参数调用，大小写必须一致。
超时	Transport 的 timeout 与 operation_timeout 均读取 DRMX5_TIMEOUT_SECONDS。
XML 设置	strict=True，xml_huge_tree=False。
异常	zeep 或连接异常被转换为 DrmxError，并由入口返回 HTTP 502。

国际版 WSDL：https://5.drm-x.com/haihaisoftlicenseservice.asmx?wsdl
中国版 WSDL：https://5.drm-x.cn/haihaisoftlicenseservice.asmx?wsdl

8、DRM-X 5.0 接口方法详解

issue_license() 按固定顺序执行六个 DRM-X 方法。

除新建用户外，后续调用都围绕当前网站用户映射出的 DRM-X 5.0 用户名；权限更新与许可证获取使用同一 POST rightsid。

顺序	方法	条件	失败处理
1	CheckUserExists	课程验证通过后	不是 true/false 或 1/0 时终止。
2	AddNewUser	用户不存在	返回值不是 1 时终止。
3	CheckUserIsRevoked	用户已存在	吊销时返回 403。
4	GetUserDetails	用户已存在且未吊销	IsLockedOut 为真时返回 403。
5	UpdateRightWithDisableVirtualMachine	用户可用	返回值不是 1 时终止。
6	getLicenseRemoteToTableWithVersionWithMac	权限更新成功	返回内容缺少许可证标记时终止。

8.1 CheckUserExists

作用：检查当前网站用户是否已经存在于 DRM-X 5.0。

参数	来源	说明
UserName	当前登录用户 username	DRM-X 用户登录名，也是后续调用的统一用户标识。
AdminEmail	config.py	DRM-X 管理员邮箱。
WebServiceAuthStr	config.py	Web Service 验证字符串。

返回 true 或 1 表示用户存在；false 或 0 表示不存在。其他值被视为异常。

8.2 AddNewUser

调用时机：CheckUserExists 返回用户不存在时。在 DRM-X 5.0 中创建与当前网站用户对应的账号。

参数组	示例来源或值	说明
AdminEmail / WebServiceAuthStr / GroupID	config.py	认证并指定新用户所属 DRM-X 用户组。
UserLoginName	登录用户 username	DRM-X 登录名。
UserPassword	N/A	示例不使用 DRM-X 密码登录。
UserEmail / UserFullName	Session 用户信息	邮箱和完整姓名；邮箱也可作为水印内容。
Title / Company / Address / City / Province / ZipCode / Phone / CompanyURL	N/A	演示未采集的资料。
SecurityQuestion / SecurityAnswer	N/A	演示不使用 DRM-X 密码找回。
IP	REMOTE_ADDR	许可证请求的远端 IP。
Money	0	新用户初始金额。
BindNumber	DEMO_BIND_NUMBER	允许的绑定数量。
IsApproved / IsLockedOut	yes / no	创建后立即批准且不锁定。

AddNewUserResult 必须为 1。新创建的用户在同一次请求中不会再次执行吊销和锁定检查。

8.3 CheckUserIsRevoked（企业版才支持吊销用户许可证功能）

调用时机：仅当 DRM-X 用户原本已经存在。作用是检查管理员是否已经吊销该用户。

参数	来源
AdminEmail	config.py
WebServiceAuthStr	config.py

参数	来源
UserLoginName	当前用户 username

返回 true、1 或 yes 均视为已吊销，后端返回 HTTP 403 并拒绝获取许可证。该功能要求 DRM-X 版本支持用户吊销。

8.4 GetUserDetails

调用时机：已有用户通过吊销检查后。示例只读取 IsLockedOut 字段；若顶层不存在，则继续从 GetUserDetailsResult 中读取。

参数	来源
AdminEmail	config.py
WebServiceAuthStr	config.py
UserLoginName	当前用户 username

IsLockedOut 为 true 或 1 时返回 HTTP 403；若响应中完全缺少该字段，则视为服务响应异常。

8.5 UpdateRightWithDisableVirtualMachine

作用：更新 POST rightsid 对应的权限。示例每次发放许可证前都会更新该 RightsID；正式系统应根据业务策略评估是否需要每次更新。

参数	示例值或来源	说明
AdminEmail / WebServiceAuthStr	config.py	DRM-X 服务认证。
RightsID	POST rightsid	要更新的权限 ID。
Description	Python Sample	权限说明。
PlayCount	-1	不限制播放次数。
BeginDate / ExpirationDate	课程日期配置	从 YYYY-MM-DD 转成 yyyy/MM/dd。

参数	示例值或来源	说明
ExpirationAfterFirstUse	-1	不启用首次使用后的相对过期天数。
RightsPrice	0	权限价格为 0。
AllowPrint / AllowClipBoard / AllowDoc	False	关闭打印、剪贴板和文档相关权限。
DisableVirtualMachine	True	禁止在虚拟机环境使用。
require_admin_permission / bind_1st_screen	False / False	不要求管理员权限，不启用第一屏绑定。

水印、黑名单和智能防录屏

参数	示例值	说明
EnableWatermark / WatermarkText	True / 用户名	启用第一水印并显示当前用户信息。
WatermarkArea	1,2,3,4,5,	第一水印显示区域。
RandomChangeArea / RandomFrquency	True / 12	随机切换水印区域及频率；
EnableBlacklist / EnableWhitelist	True / True	启用黑名单和智能防录屏功能。
ExpireTimeUnit	Day	过期时间的单位。 Year/Month/Day/Hour/Minute
PreviewTime / PreviewTimeUnit / PreviewPage	3 / Day / 3	无实际功能效果，请忽略。

第三层动态水印和第二层滚动水印

参数组	示例值	说明
enable_wm_3rd	1	启用第三层动态水印。

参数组	示例值	说明
wm_3rd_random_change_angle / textalpha	1 / 90	随机角度和透明度。
wm_3rd_rows / columns	3 / 3	水印行列数。
wm_3rd_fixed_angle / textsize / textcolor	0 / 14 / 16744576	角度、字号和十进制颜色值。
wm_3rd_angle_change_interval / text	10 / 用户邮箱或用户名水印	角度切换间隔和水印文本。
enable_wm_2nd	0	关闭第二层滚动水印。
wm_2nd_text / speed / textsize / textcolor / textalpha	用户水印 / 1 / 22 / 16744576 / 90	即使关闭仍按接口结构提交。

UpdateRightWithDisableVirtualMachineResult 必须为 1。以上是演示策略，不代表所有客户都应使用相同权限。

8.6 getLicenseRemoteToTableWithVersionWithMac

作用：在权限更新成功后，根据当前模板、客户端数据、用户和设备信息生成最终许可证 HTML。

参数	来源	说明
AdminEmail / WebServiceAuthStr	config.py	DRM-X 服务认证。
ProfileID	POST profileid	当前 License Profile ID。
ClientInfo	POST clientinfo	ZJGet 客户端许可证数据。
RightsID	POST rightsid	与权限更新使用同一 RightsID。
UserLoginName / UserFullName	Session 用户	DRM-X 用户登录名和完整姓名。
GroupID	config.py	DRM-X 用户组 ID。
Message	N/A	ASMX 输入输出参数；服务信息会显示在结果页面。

参数	来源	说明
IP	REMOTE_ADDR	许可证请求的远端 IP。
Platform / ContentType / Version / Mac	对应 POST 字段	ZJGet 平台、内容、版本及设备信息。

代码读取 getLicenseRemoteToTableWithVersionWithMacResult 作为许可证 HTML，并读取 Message。成功内容必须包含 license_div_drm-x5；否则去除 HTML 标签后生成错误信息并返回 HTTP 502。

9、Django 路由与页面职责

路径	方法	职责
/	GET	跳转到登录页面。
/login	GET / POST	显示固定账号登录、创建 Session，并在需要时恢复许可证流程。
/logout	POST	清空当前 Session。
/api/session	GET	返回 authenticated、用户信息和是否存在待处理许可证请求。
/licstore5	POST	ZJGet 集成入口；接收参数、同源中继并发放许可证。
/licstore5?resume=1	GET	从 Session 恢复此前保存的许可证请求。

模板	职责
login.html	显示登录状态、固定账号登录表单和错误信息。
relay.html	把首次跨站 POST 的参数自动提交到同源 /licstore5。
license.html	输出 DRM-X 返回的许可证 HTML、Message 和返回地址逻辑。
error.html	统一显示 400、403 或 502 等错误。

10、本地运行与演示验证

10.1 准备环境

- Python 3.10 或兼容版本。
- DRM-X 5.0/ZJGet 测试环境及已经加密的视频。

10.2 创建测试环境并安装依赖

```
cd "DRM-X5_Python_Django_Integration_Example"
python -m venv .venv
.\.venv\Scripts\Activate.ps1
python -m pip install -r requirements.txt
```

10.3 检查和启动

```
python manage.py check
python manage.py runserver 8080
```

启动成功后，登录页为 http://localhost:8080/login。测试账号为 student，密码为 123456。

10.4 DRM-X 5.0 后台配置与端到端验证

1. 在 config.py 填写测试账号对应的 AdminEmail、WebServiceAuthStr、GroupID，并选择中国版或国际版服务地址。



2. 把测试课程 ID 添加到 DEMO_PURCHASED_COURSE_IDS。

3. 在测试视频对应的许可证模板中设置同一个课程 ID，例如 1001；多个值可写为 1001-1002。



4. 在 DRM-X 5.0 网站集成参数中选择自定义登录页面集成，并把许可证获取 URL 设为 http://localhost:8080/licstore5



5. 使用 ZJGet 打开加密视频，不要在普通浏览器地址栏直接访问 /licstore5。
6. 未登录时完成登录，确认页面自动恢复请求；已登录时应直接进入课程和 DRM-X 用户检查。
7. 确认用户创建或状态检查成功、权限已更新，并且 ZJGet 能获得许可证并播放内容。

11、正式项目改造建议

演示实现	正式系统建议
固定账号和密码	接入 Django Authentication、SSO 或客户现有身份系统，并保证用户名映射稳定且唯一。
固定已购买课程列表	按当前登录用户从订单、选课、订阅或 LMS 数据库查询真实课程权限。
明文 config.py	客户需要更高保护时可迁移到密钥服务或环境变量。
文件型 Session	多服务器部署时使用数据库、Redis 或其他共享 Session 后端。
开发环境 HTTP	正式环境必须使用 HTTPS，设置 DEBUG=False、SESSION_COOKIE_SECURE=True，并配置 ALLOWED_HOSTS。
REMOTE_ADDR 作为 IP	经过反向代理时，只在可信代理配置完成后读取转发的真实客户端 IP。
每次更新 POST RightsID	根据播放次数、有效期、水印、绑定和虚拟机策略决定是否更新；评估多个 Profile 共用同一 RightsID 的并发影响。

演示实现	正式系统建议
错误直接返回页面	服务器记录可追踪日志，但客户端只显示必要信息，避免泄露认证值。

> **生产边界** 课程验证必须证明“当前登录用户拥有当前 POST 中至少一个课程”，不能只检查课程是否存在。DRM-X 用户名映射也应在用户改名后保持稳定。

12、常见问题与故障排查

现象	常见原因	处理
直接访问 /licstore5 显示 No saved license request	没有 ZJGet POST，也没有 Session 中的待处理请求。	用 ZJGet 打开加密文件，从完整流程测试。
登录后没有继续获取许可证	Cookie、Session 或 relay 中继没有生效。	检查 .sessions 可写、浏览器 Cookie、同源 URL 和 /login?resume=1。
ModuleNotFoundError: zeep	依赖尚未安装或使用了错误 Python 环境。	在当前虚拟环境执行 python -m pip install -r requirements.txt。
WSDL 或连接失败	服务区域、地址、网络或 TLS 配置错误。	确认 .asmx 地址、5.drm-x.com/.cn、网络和超时；不要在配置中重复 ?wsdl。
返回 400	必填 POST 参数缺失，或 return_url 使用危险协议。	检查 profileid、clientinfo、rightsid、yourproductid 及 return_url。
返回 403	课程不匹配，或 DRM-X 用户被吊销/锁定。	核对课程购买列表、License Profile 产品 ID 和 DRM-X 用户状态。
返回 502	DRM-X 配置、SOAP 调用或服务返回内容异常。	核对认证值、GroupID、WSDL、后端日志及接口返回。
端口 8080 已占用	另一进程正在监听该端口。	停止占用进程，或更换端口并同步修改 DRM-X 集成 URL。
许可证 HTML 无效	服务返回错误文本或结构不完整。	检查 Message；成功 HTML 必须包含 license_div_drm-x5。

附录：文件与职责速查

文件	职责
drmx5_demo/config.py	DRM-X 服务端配置、演示账号、课程日期、购买列表和绑定数量。
drmx5_demo/settings.py	Django 中间件、模板、文件 Session、Cookie 和静态资源配置。
drmx5_demo/urls.py	把根路径交给 integration.urls。
integration/views.py	参数读取、课程验证、登录、Session、zeep 客户端及六个 DRM-X 方法。
integration/urls.py	定义 /login、/logout、/api/session 和 /licstore5。
integration/templates/integration/*.html	relay 完成同源 POST 中继；login 负责登录；license 呈现许可证；error 显示错误响应。
requirements.txt	声明 Django 与 zeep 依赖。
README.md	项目快速说明、配置与运行入口。