CU E-Receipt

API Gateway Integration Guide

สำหรับ External Developer / System Integrator

เวอร์ชัน	2.2
วันที่	พฤษภาคม 2569
ติดต่อ	thannaphat.j@zenithcomp.co.th

1. ภาพรวมระบบ (System Overview) 4
Environment & Base URL 4
Endpoints สรุป 4
2. ภาพรวม Flow 5
3. การขอ Access Token 6
4. ดึงรายการโครงการที่พร้อมชำระเงิน 8
4.1 Required Headers 8
4.2 Request Body (ทั้งหมด optional) 9
4.3 ตัวอย่าง Request 9
4.4 Response 200 OK 10
5. การสร้าง Payment Session 12
5.1 Required Headers 12
5.2 Request Body 12
5.3 data Array (รายการย่อยในใบเสร็จ) 14
5.4 ตัวอย่าง Request 15
5.5 Response 201 Created 17
6. การรับ Webhook Callback 18
6.1 Webhook Payload 18
6.2 Retry Policy 19
7. HMAC Signature — Inbound (คำนวณและส่งมาให้เรา) 20
7.1 สูตรคำนวณ 20
7.2 Python 20
7.3 PHP 21
7.4 Node.js 21
8. HMAC Signature — Outbound (Verify Webhook) 23
8.1 สูตร Verify 23
8.2 Python (Flask) 23

8.3 PHP 24
8.4 Node.js (Express) 25
9. ตัวอย่างโค้ด Full Flow 26
9.1 Python 26
9.2 PHP 29
9.3 Node.js 31
10. Error Codes 35
10.1 HTTP Status Codes 35
10.2 ข้อผิดพลาดที่พบบ่อย 35
10.3 Error Response Format 36
11. FAQ 37
12. API Documentation (Swagger UI) 40
12.1 Swagger UI URLs 40

1. ภาพรวมระบบ (System Overview)

ระบบ CU E-Receipt API Gateway เปิดให้ External System รวมระบบชำระเงินได้อย่างปลอดภัยและเป็นมาตรฐาน รองรับการทำงานหลัก 5 ด้าน ดังนี้

- **ดึงรายการโครงการ** — ทุกประเภทในเส้นทางเดียว (content, health_science, room)
- **สร้าง Payment Session** — ได้รับ URL สำหรับ redirect ผู้ใช้ไปชำระเงิน
- **ส่งรายการย่อย** — สำหรับแสดงในใบเสร็จ PDF (data object)
- **รับ Webhook** — แจ้งผลการชำระเงินแบบ real-time ทันทีที่ชำระสำเร็จ
- **รับเลขใบเสร็จ** — พร้อม receipt URL สำหรับดาวน์โหลด PDF

> การลงทะเบียน: ทำผ่าน Web UI ของ CU E-Receipt เท่านั้น — ผู้ดูแลระบบตรวจสอบและอนุมัติ จากนั้น client_id, client_secret และ webhook_secret จะถูกส่งทาง email โดยอัตโนมัติ

Environment & Base URL

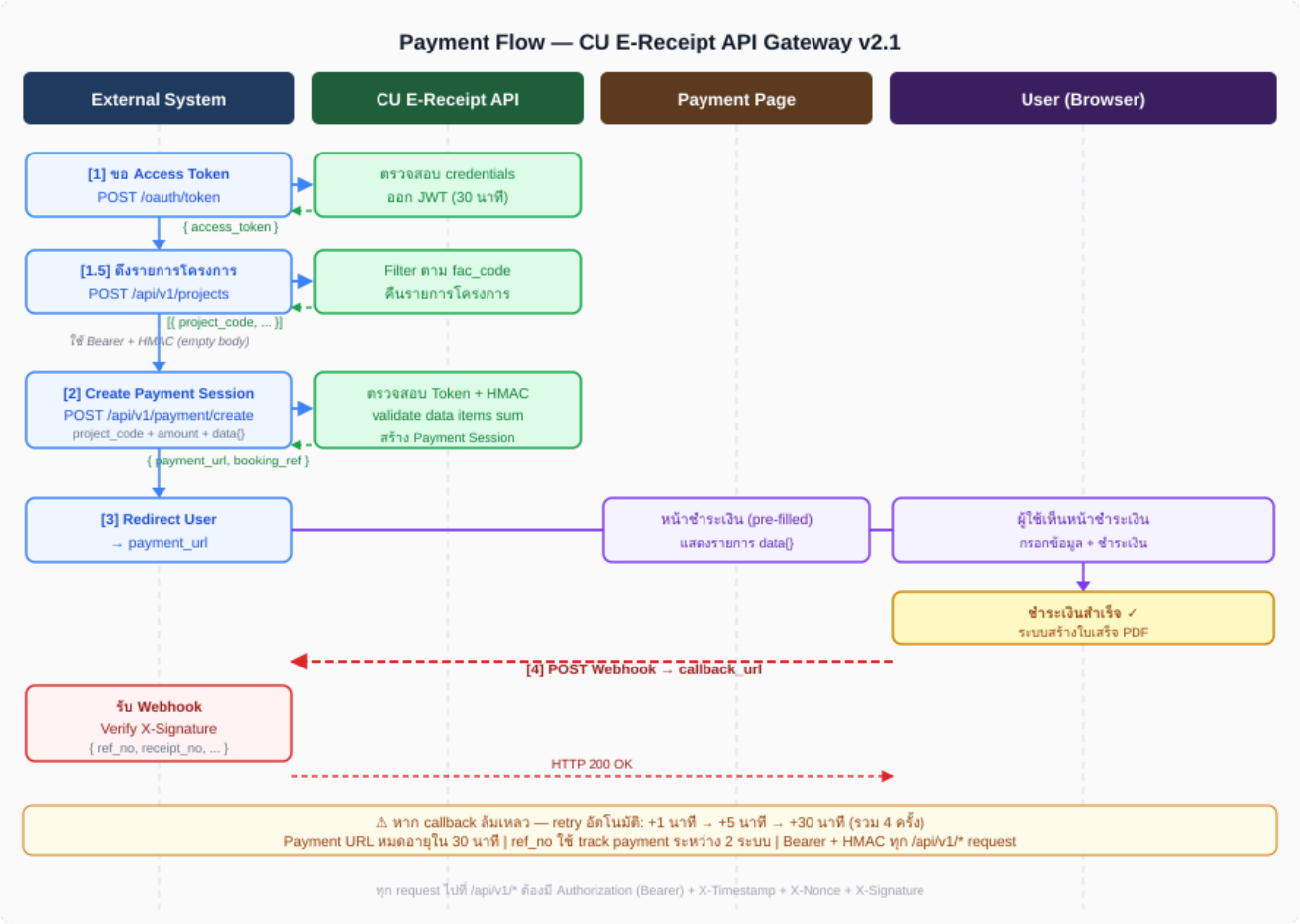
Environment	Base URL
UAT / Staging	https://ereceipt.pay.ofas.chula.ac.th
Production	https://ereceipt.ofas.chula.ac.th

Endpoints สรุป

Method	Path	Auth	Description
POST	/oauth/token	Client Credentials	ขอ Access Token
POST	/api/v1/projects	Bearer + HMAC	ดึงรายการโครงการ (filter ใน body)
POST	/api/v1/payment/create	Bearer + HMAC	สร้าง Payment Session

2. ภาพรวม Flow

หลังจากได้รับ credentials ทาง email แล้ว ขั้นตอนการทำงานทั้งหมดประกอบด้วย 4 ขั้นตอนหลัก ดังแผนภาพด้านล่าง



แผนภาพ Payment Flow — ตั้งแต่ขอ Token, ดึงรายการโครงการ (POST), สร้าง Session จนถึง Webhook

3. การขอ Access Token

ใช้ OAuth 2.0 Client Credentials flow — ขอ token ใหม่ทุกครั้งก่อนเรียก API ไม่ต้อง cache หรือ track expiry

Endpoint

```
POST  /oauth/token
```

Request Body (JSON)

```
{
  "grant_type":    "client_credentials",
  "client_id":     "550e8400-e29b-41d4-a716-446655440000",
  "client_secret": "AbCd1234...64chars...",
  "scope":         "payment:create"
}
```

Response 200 OK

JSON

```
{
  "token_type":   "Bearer",
  "expires_in":   1800,
  "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9..."
}
```

Field	ค่า
expires_in	1,800 วินาที = 30 นาที
token_type	Bearer — JWT (RS256)
refresh_token	*ไม่มี — ขอใหม่ด้วย client credentials ได้ตลอด*

ℹ ขอ Access Token ใหม่ทุกครั้งก่อนเรียก API — ไม่ต้อง cache หรือ track expiry แค่ขอแล้วใช้ทันที แล้วทิ้ง

4. ดึงรายการโครงการที่พร้อมชำระเงิน

ระบบ auto-filter ตาม fac_code ของ Gateway Client — ไม่ต้องระบุคณะเอง ใช้ project_code และ type ที่ได้รับจาก response นี้ไปส่งใน POST /api/v1/payment/create (ต้องส่งคู่กันเสมอ)

Endpoint

```
POST  /api/v1/projects
```

ทำไมใช้ POST? — filter ถูกส่งใน JSON body ซึ่งถูก sign ด้วย HMAC เพื่อป้องกันการแก้ไข filter ระหว่างทาง

4.1 Required Headers

Header	คำอธิบาย
Authorization	Bearer {access_token}
Content-Type	application/json
X-Timestamp	Unix timestamp (วินาที) — ต้องไม่ต่างจากเวลาเซิร์ฟเวอร์เกิน ±5 นาที
X-Nonce	Random string ยาว 16–64 chars — ห้ามซ้ำต่อ request
X-Signature	HMAC-SHA256 signature คำนวณจาก JSON body (ดูวิธีคำนวณ Section 6)

ถ้าไม่ต้องการ filter ให้ส่ง body เป็น {} (empty JSON object) — หมายเหตุ: {} และ "" ให้ sha256 ต่างกัน ให้ใช้ค่าจริงที่ส่งไป

4.2 Request Body (ทั้งหมด optional)

Field	Type	Default	คำอธิบาย
search	string	—	ค้นหาจากชื่อโครงการ
type	string	ทุกประเภท	content | health_science | room
page	integer	1	หน้าที่ต้องการ
per_page	integer	20	รายการต่อหน้า (สูงสุด 50)

4.3 ตัวอย่าง Request

HTTP Request

```
POST /api/v1/projects
Authorization: Bearer eyJ0eXAiOiJKV1Qi...
Content-Type: application/json
X-Timestamp: 1741852800
X-Nonce: k9xM2pLnQr7vYwZ3
X-Signature: base64encodedHmacSignature==

{
  "type":     "content",
  "search":   "Python",
  "page":     1,
  "per_page": 20
}
```

4.4 Response 200 OK

JSON


```
{
  "status":   "success",
  "fac_code": "3700",
  "data": [
    {
      "type":         "content",
      "project_code": "37000001",
      "project_name": "อบรม Python Basics รุ่นที่ 3",
      "amount":       1700.00,
      "faculty":      "3700",
      "start_date":   "2026-01-01 00:00:00",
      "end_date":     "2026-12-31 00:00:00"
    },
    {
      "type":         "health_science",
      "project_code": "HS690001",
      "project_name": "บริการตรวจสุขภาพพนักงาน",
      "amount":       "1810.00",
      "faculty":      "09001",
      "start_date":   null,
      "end_date":     null
    },
    {
      "type":         "room",
      "project_code": "RM690001",
      "project_name": "ห้องพักนิสิต A101",
      "amount":       "3500.00",
```

```
      "faculty":      "0104",
      "start_date":   null,
      "end_date":     null
    }
  ],
  "pagination": { "total": 12, "per_page": 20, "current_page": 1, "last_page": 1 }
}
```


Field	คำอธิบาย
data[].type	ประเภทโครงการ: content, health_science, room — ต้องส่งเป็น type ใน POST /api/v1/payment/create
data[].project_code	รหัสโครงการ — ต้องส่งคู่กับ type ใน POST /api/v1/payment/create
data[].project_name	ชื่อโครงการ
data[].amount	จำนวนเงิน (บาท)
fac_code	รหัสคณะของ Gateway Client (null = ระดับ System Admin)

5. การสร้าง Payment Session

Endpoint

```
POST /api/v1/payment/create
```

5.1 Required Headers

Header	คำอธิบาย
Authorization	Bearer {access_token}
Content-Type	application/json
X-Timestamp	Unix timestamp (วินาที) — ต้องไม่ต่างจากเวลาเซิร์ฟเวอร์เกิน ±5 นาที
X-Nonce	Random string ยาว 16–64 chars — ห้ามซ้ำต่อ request
X-Signature	HMAC-SHA256 signature (ดูวิธีคำนวณ Section 6)

5.2 Request Body

Field	Type	จำเป็น	คำอธิบาย
mode	string	—	โหมดการทำงาน: test (default) หรือ production — ดูหมายเหตุด้านล่าง
project_code	string	✓	รหัสโครงการ (field project_code จาก POST /api/v1/projects)
type	string	✓	ประเภทโครงการ (field type จาก POST /api/v1/projects): content | health_science | room
amount	number	⚠	จำนวนเงินรวม (บาท) — required เมื่อ type=content หรือ health_science; optional เมื่อ type=room (คำนวณจาก room.unit_price + data อัตโนมัติ)

payer_name	string	⚠	ชื่อผู้ชำระ — required เมื่อ type=content หรือ health_science; optional เมื่อ type=room
payer_email	string	⚠	อีเมลผู้ชำระ — required เมื่อ type=content หรือ health_science; optional เมื่อ type=room
payer_phone	string	⚠	เบอร์โทรผู้ชำระ — required เมื่อ type=content หรือ health_science; optional เมื่อ type=room
payer_organization	string	—	ชื่อองค์กรผู้ชำระ
ref_no	string	✓	เลขอ้างอิงจาก system ของคุณ — สูงสุด 55 อักษร
callback_url	string	⚠	URL รับ webhook — required เมื่อ mode=production
title	string	⚠	ชื่อหัวข้อรายการในใบเสร็จ — required เมื่อส่ง data (type!=room); optional เมื่อ type=room
data	array	—	รายการค่าใช้จ่ายเพิ่มเติม เช่น ค่าน้ำ ค่าไฟ — optional ทุก type (ดูรายละเอียดด้านล่าง)
room	object	⚠	ข้อมูลห้องพัก — required เมื่อ type=room (ดูรายละเอียดด้านล่าง)
receipt_info	object	—	ข้อมูลผู้รับใบเสร็จ — optional สำหรับ type=room (ดูรายละเอียดด้านล่าง)

mode	callback_url	การทำงาน
test (default)	optional	ทดสอบ flow ได้โดยไม่ต้องมี callback server
production	**required**	ใช้งานจริง — ระบบ POST webhook ไปที่ callback_url เมื่อชำระเงินสำเร็จ

⚠ บน Production server (ereceipt.ofas.chula.ac.th) — callback_url บังคับเสมอ ไม่ว่าจะส่ง mode อะไรมา

5.3 room Object (required เมื่อ type=room)

เมื่อ type=room ต้องส่ง object room เพื่อระบุห้องที่ต้องการจอง

Field	Type	จำเป็น	คำอธิบาย
room.building_id	integer	✓	รหัสอาคาร (≥ 1)
room.floor	integer	✓	ชั้น (≥ 1)
room.room_id	integer	✓	รหัสห้อง (≥ 1)
room.booking_type	string	✓	ประเภทการจอง: daily (รายวัน) | monthly (รายเดือน) | term (รายเทอม)
room.start_date	string	✓	วันที่เริ่มต้น รูปแบบ YYYY-MM-DD เช่น 2026-06-01
room.end_date	string	✓	วันที่สิ้นสุด รูปแบบ YYYY-MM-DD — ต้องไม่น้อยกว่า start_date
room.unit_price	number	✓	ราคาค่าเช่าหลัก (บาท) — ใช้เป็น main item ในใบเสร็จ

5.3b receipt_info Object (optional — สำหรับ type=room)

ข้อมูลผู้รับใบเสร็จ สำหรับพิมพ์ในใบเสร็จห้องพัก — optional ทุก field

Field	Type	จำเป็น	คำอธิบาย
receipt_info.is_personal	boolean	—	true = บุคคลธรรมดา (default), false = นิติบุคคล
receipt_info.id_card	string	—	เลขบัตรประชาชน (กรณี is_personal=true)
receipt_info.tax_number	string	—	เลขประจำตัวผู้เสียภาษี (กรณี is_personal=false)
receipt_info.name	string	—	ชื่อที่ออกใบเสร็จ
receipt_info.phone	string	—	เบอร์โทรในใบเสร็จ

5.4 data Array (required เมื่อ type=room, optional สำหรับ type อื่น)

เมื่อ type=room — ต้องส่ง data และ title เสมอ สำหรับ type=content / health_science — ถ้าไม่ส่ง data ระบบจะใช้ชื่อโครงการเป็นรายการเดียวในใบเสร็จ PDF

Field	Type	จำเป็น	คำอธิบาย
data[].name	string	✓	ชื่อรายการย่อย
data[].quantity	integer	✓	จำนวน (≥ 1)
data[].unit_price	number	✓	ราคาต่อหน่วย (บาท)
data[].description	string	—	รายละเอียดเพิ่มเติม

⚠ ข้อกำหนด: sum(quantity × unit_price) ของทุก item ต้องเท่ากับ amount (tolerance ±0.01 บาท) ระบบจะตอบ 422 ถ้าไม่ตรง

5.4 ตัวอย่าง Request — type=content

HTTP Request + JSON Body

```
POST /api/v1/payment/create
Authorization: Bearer eyJ0eXAiOiJKV1Qi...
Content-Type: application/json
X-Timestamp: 1741852800
X-Nonce: k9xM2pLnQr7vYwZ3
X-Signature: base64encodedHmacSignature==

{
  "project_code": "37000001",
  "type":         "content",
  "amount":       1700.00,
  "payer_name":   "สมชาย ใจดี",
  "payer_email":  "somchai@example.com",
  "payer_phone":  "0891234567",
```

```
  "ref_no":       "YOUR-SYS-REF-001",
  "title":        "อบรม Python Basics รุ่นที่ 3",
  "data": [
    { "name": "ค่าลงทะเบียน",    "quantity": 1, "unit_price": 1500.00, "description":
"หลักสูตรสำหรับผู้เริ่มต้น" },
    { "name": "ค่าเอกสารประกอบ", "quantity": 2, "unit_price": 100.00 }
  ]
}
```

5.6 ตัวอย่าง Request — type=room

HTTP Request + JSON Body

```
POST /api/v1/payment/create
Authorization: Bearer eyJ0eXAiOiJKV1Qi...
Content-Type: application/json
X-Timestamp: 1741852800
X-Nonce: k9xM2pLnQr7vYwZ3
X-Signature: base64encodedHmacSignature==

{
  "project_code": "0104001",
  "type":         "room",
  "amount":       3000.00,
  "payer_name":   "สมชาย ใจดี",
  "payer_email":  "somchai@example.com",
  "payer_phone":  "0891234567",
  "ref_no":       "YOUR-SYS-REF-002",
  "callback_url": "https://your-system.example.com/callback",
  "room": {
    "building_id": 1,
    "floor": 3,
    "room_id": 42,
    "booking_type": "monthly",
    "start_date": "2026-06-01",
    "end_date":   "2026-06-30",
    "unit_price": 3000.00
  },
  "receipt_info": {
    "is_personal": true,
```

```
    "id_card": "1100100100001",
    "name": "นายสมชาย ใจดี",
    "phone": "0891234567"
  },
  "title": "ค่าเช่าห้องพัก เดือนมิถุนายน 2569",
  "data": [
    { "name": "ค่าน้ำ", "quantity": 1, "unit_price": 200.00 },
    { "name": "ค่าไฟ", "quantity": 1, "unit_price": 350.00 }
  ]
}
```

5.7 Response 201 Created

JSON

```
{
  "status":      "success",
  "payment_url": "https://ereceipt.ofas.chula.ac.th/project/payment/aB3xK9mP...",
  "booking_ref": "BK-690001",
  "expires_at":  "2026-03-13T16:30:00+07:00"
}
```

Field	คำอธิบาย
mode	โหมดที่ใช้งาน (test หรือ production)
payment_url	URL สำหรับ redirect user ไปชำระเงิน — ใช้ได้ 30 นาที
booking_ref	เลขอ้างอิง booking ของระบบ CU (format: BK-690001)
expires_at	เวลาหมดอายุของ payment_url (ISO 8601, Asia/Bangkok)

ℹ Redirect user ไปที่ payment_url ทันที — user จะเห็นหน้าชำระเงินพร้อมข้อมูล pre-filled และรายการ data{}

6. การรับ Webhook Callback

หลังผู้ใช้ชำระเงินสำเร็จ ระบบจะ POST ไปที่ callback_url ของคุณโดยอัตโนมัติ

6.1 Webhook Payload

Incoming Webhook

```
POST https://your-system.example.com/payment/callback
Content-Type: application/json
X-Timestamp: 1741852900
X-Signature: a3f8c2d1e4b7...  (hex string)

{
  "event":       "payment.success",
  "booking_ref": "BK-690001",
  "ref_no":      "YOUR-SYS-REF-001",
  "receipt_no":  "CU37690001",
  "amount":      1700.00,
  "paid_at":     "2026-03-13T15:45:00+07:00",
  "receipt_url": "https://ereceipt.ofas.chula.ac.th/project/payment/success/aB3xK9..."
}
```

Field	คำอธิบาย
event	"payment.success" เสมอ
booking_ref	เลข booking ของระบบ CU
ref_no	เลขอ้างอิงที่คุณส่งมาตอน create session — ใช้ track payment ใน system ของคุณ
receipt_no	เลขใบเสร็จ เช่น CU37690001
amount	จำนวนเงินที่ชำระ (บาท)

paid_at	เวลาชำระ (ISO 8601, Asia/Bangkok)
receipt_url	URL สำหรับดูใบเสร็จ (HTML + print + PDF download)

การ track ระหว่าง 2 ระบบ: ใช้ ref_no ที่คุณส่งมาตอน create session เพื่อ map กับ order ใน system ของคุณ

6.2 Retry Policy

Server ต้องตอบกลับ HTTP 200 — หากตอบ non-2xx ระบบจะ retry อัตโนมัติ:

ครั้งที่	หน่วงเวลา	หมายเหตุ
1 (initial)	ทันที	การส่งครั้งแรก
2	+1 นาที	retry ครั้งที่ 1
3	+5 นาที	retry ครั้งที่ 2
4	+30 นาที	retry ครั้งสุดท้าย — หากยังล้มเหลว error จะถูก log ไว้ฝั่งระบบ CU

7. HMAC Signature — Inbound (คำนวณและส่งมาให้เรา)

ทุก request ไปที่ /api/v1/* ต้องมี X-Signature เพื่อยืนยันความถูกต้องของข้อมูล

7.1 สูตรคำนวณ

Formula

```
message   = client_id + "\n" + timestamp + "\n" + nonce + "\n" + sha256(body)
signature = base64( HMAC-SHA256(message, client_secret) )
```

⚠ ส่ง body {} (empty JSON object) หรือ "" (empty string) ⟶ sha256 ค่าต่างกัน — ใช้ sha256 ของ body จริงที่ส่งไปเสมอ

ตัวแปร	ค่า
client_id	UUID ที่ได้รับทาง email — ต้องเป็น lowercase
client_secret	Plain text secret จาก email (ไม่ใช่ bcrypt)
timestamp	ค่าเดียวกับ X-Timestamp header
nonce	ค่าเดียวกับ X-Nonce header (16–64 chars, ไม่ซ้ำต่อ request)
sha256(body)	SHA-256 ของ raw request body (hex lowercase) — ใช้ค่าจริงที่ส่ง ไม่ใช่ค่า default
\n separator	Newline character (LF, 0x0A)

7.2 Python

Python

```
import hashlib, hmac, base64, json, time, uuid


def create_signature(body_str: str, timestamp: int, nonce: str) -> str:
    """body_str = json.dumps(payload) หรือ json.dumps({}) สำหรับ filter ว่าง"""
```

```
    body_hash = hashlib.sha256(body_str.encode('utf-8')).hexdigest()
    message   = "\n".join([client_id, str(timestamp), nonce, body_hash])
    sig = hmac.new(client_secret.encode('utf-8'), message.encode('utf-8'), hashlib.sha256).digest()
    return base64.b64encode(sig).decode('utf-8')
```

7.3 PHP

PHP

```
/**
 * $bodyJson = json_encode($payload) หรือ json_encode([]) สำหรับ filter ว่าง
 */
function createSignature(string $bodyJson, int $timestamp, string $nonce,
                         string $clientId, string $clientSecret): string
{
    $bodyHash = hash('sha256', $bodyJson);
    $message  = implode("\n", [$clientId, $timestamp, $nonce, $bodyHash]);
    return base64_encode(hash_hmac('sha256', $message, $clientSecret, true));
}
```

7.4 Node.js

Node.js

```
const crypto = require('crypto');

// bodyStr = JSON.stringify(payload) หรือ JSON.stringify({}) สำหรับ filter ว่าง
function createSignature(bodyStr, timestamp, nonce) {
  const bodyHash = crypto.createHash('sha256').update(bodyStr, 'utf8').digest('hex');
  const message  = [clientId, timestamp, nonce, bodyHash].join('\n');
```

```
  const rawHmac  = crypto.createHmac('sha256', clientSecret).update(message, 'utf8').digest();
  return Buffer.from(rawHmac).toString('base64');
}
```

8. HMAC Signature — Outbound (Verify Webhook)

เมื่อรับ Webhook จากระบบ CU ต้อง verify X-Signature ก่อนประมวลผลทุกครั้ง เพื่อป้องกัน replay attack

8.1 สูตร Verify

Formula

```
signature = hex( HMAC-SHA256(timestamp + "." + body, webhook_secret) )

// ผลลัพธ์เป็น hex string (ไม่ใช่ base64)
```

8.2 Python (Flask)

Python (Flask)

```
import hmac, hashlib
from flask import Flask, request, jsonify

WEBHOOK_SECRET = "xK9mP2..."  # จาก email

@app.route('/payment/callback', methods=['POST'])
def payment_callback():
    timestamp    = request.headers.get('X-Timestamp', '')
    received_sig = request.headers.get('X-Signature', '')
    body         = request.get_data()  # raw bytes

    message      = f"{timestamp}.{body.decode('utf-8')}"
    expected_sig = hmac.new(WEBHOOK_SECRET.encode('utf-8'),
                            message.encode('utf-8'), hashlib.sha256).hexdigest()

    if not hmac.compare_digest(expected_sig, received_sig):
        return jsonify({'error': 'Invalid signature'}), 401
```

```
    data = request.get_json()
    if data.get('event') == 'payment.success':
        ref_no     = data['ref_no']   # เลขอ้างอิงของคุณ
        receipt_no = data['receipt_no']
        # อัปเดต order ใน system ของคุณโดยใช้ ref_no...

    return jsonify({'status': 'ok'}), 200
```

8.3 PHP

PHP

```
$webhookSecret = 'xK9mP2...';
$timestamp    = $_SERVER['HTTP_X_TIMESTAMP'] ?? '';
$receivedSig  = $_SERVER['HTTP_X_SIGNATURE'] ?? '';
$body         = file_get_contents('php://input');

$expectedSig  = hash_hmac('sha256', $timestamp . '.' . $body, $webhookSecret);

if (!hash_equals($expectedSig, $receivedSig)) {
    http_response_code(401);
    echo json_encode(['error' => 'Invalid signature']);
    exit;
}

$data = json_decode($body, true);
if ($data['event'] === 'payment.success') {
    $refNo = $data['ref_no'];  // อัปเดต order ใน system ของคุณ
}
```

```
http_response_code(200);
echo json_encode(['status' => 'ok']);
```

8.4 Node.js (Express)

Node.js (Express)

```
app.post('/payment/callback', express.raw({ type: 'application/json' }), (req, res) => {
  const timestamp   = req.headers['x-timestamp'] || '';
  const receivedSig = req.headers['x-signature'] || '';
  const body        = req.body.toString('utf-8');

  const message     = `${timestamp}.${body}`;
  const expectedSig = crypto.createHmac('sha256', WEBHOOK_SECRET)
    .update(message, 'utf-8').digest('hex');

  if (!crypto.timingSafeEqual(
    Buffer.from(expectedSig, 'utf-8'),
    Buffer.from(receivedSig, 'utf-8')
  )) return res.status(401).json({ error: 'Invalid signature' });

  const data = JSON.parse(body);
  if (data.event === 'payment.success') {
    const { ref_no, receipt_no, amount } = data;
    // อัปเดต order ใน system ของคุณโดยใช้ ref_no...
  }
  res.status(200).json({ status: 'ok' });
});
```

9. ตัวอย่างโค้ด Full Flow

Flow: ขอ Token ⟶ ดึงรายการโครงการ ⟶ สร้าง Payment Session ⟶ Redirect user — ขอ Token ใหม่ทุกครั้ง ไม่ต้อง cache

9.1 Python

Python

```
import hashlib, hmac, base64, json, time, uuid, requests

BASE_URL      = "https://ereceipt.ofas.chula.ac.th"
CLIENT_ID     = "550e8400-e29b-41d4-a716-446655440000"
CLIENT_SECRET = "AbCd1234..."

def _get_access_token() -> str:
    resp = requests.post(f"{BASE_URL}/oauth/token", json={
        "grant_type": "client_credentials", "client_id": CLIENT_ID,
        "client_secret": CLIENT_SECRET, "scope": "payment:create",
    }, timeout=10)
    resp.raise_for_status()
    return resp.json()["access_token"]

def _sign(body_str: str, ts: int, nonce: str) -> str:
    body_hash = hashlib.sha256(body_str.encode("utf-8")).hexdigest()
    message   = "\n".join([CLIENT_ID, str(ts), nonce, body_hash])
    return base64.b64encode(hmac.new(CLIENT_SECRET.encode("utf-8"),
        message.encode("utf-8"), hashlib.sha256).digest()).decode("utf-8")

def list_projects(token: str, type_filter: str = "") -> list:
    payload  = {}
```

```
    if type_filter: payload["type"] = type_filter
    ts, nonce = int(time.time()), uuid.uuid4().hex[:16]
    body_str  = json.dumps(payload, separators=(",", ":"), ensure_ascii=False)
    resp = requests.post(f"{BASE_URL}/api/v1/projects",
        data=body_str.encode("utf-8"),
        headers={"Authorization": f"Bearer {token}",
                 "Content-Type": "application/json",
                 "X-Timestamp": str(ts), "X-Nonce": nonce,
                 "X-Signature": _sign(body_str, ts, nonce)}, timeout=10)
    resp.raise_for_status()
    return resp.json()["data"]


def create_payment_session(token, project_code, project_type, amount, ref_no,
                           payer_name="", payer_email="", payer_phone="",
                           data=None) -> dict:
    payload = {"project_code": project_code, "type": project_type, "amount": amount, "ref_no": ref_no}
    if payer_name:  payload["payer_name"]  = payer_name
    if payer_email: payload["payer_email"] = payer_email
    if payer_phone: payload["payer_phone"] = payer_phone
    if title:       payload["title"]       = title
    if data:        payload["data"]        = data
    ts, nonce = int(time.time()), uuid.uuid4().hex[:16]
    body_str  = json.dumps(payload, separators=(",", ":"), ensure_ascii=False)
    resp = requests.post(f"{BASE_URL}/api/v1/payment/create",
        data=body_str.encode("utf-8"),
        headers={"Authorization": f"Bearer {token}", "Content-Type": "application/json",
                 "X-Timestamp": str(ts), "X-Nonce": nonce,
                 "X-Signature": _sign(body_str, ts, nonce)}, timeout=15)
    resp.raise_for_status()
```

```
    return resp.json()


# ──── ตัวอย่างการใช้งาน
# ──────────────────────────────────────────────────────────────────────────

token    = _get_access_token()
projects = list_projects(token, type_filter="content")
project  = next(p for p in projects if "Python" in p["project_name"])
project_code = project["project_code"]  # เช่น "37000001"
project_type = project["type"]          # เช่น "content"

result = create_payment_session(
    token, project_code, project_type, 1700.00, "ORDER-001",
    payer_name="สมชาย ใจดี", payer_email="somchai@example.com",
    payer_phone="0891234567",
    title="อบรม Python Basics รุ่นที่ 3",
    data=[
        {"name": "ค่าลงทะเบียน",    "quantity": 1, "unit_price": 1500.00},
        {"name": "ค่าเอกสารประกอบ", "quantity": 2, "unit_price": 100.00},
    ],
)
print("Payment URL:", result["payment_url"])
# ⟶ Redirect user ไปที่ result["payment_url"]
```

9.2 PHP

PHP

```php
<?php
const BASE_URL = 'https://ereceipt.ofas.chula.ac.th';
const CLIENT_ID = '550e8400-e29b-41d4-a716-446655440000';
const CLIENT_SECRET = 'AbCd1234...';

function getAccessToken(): string {
    $ch = curl_init(BASE_URL . '/oauth/token');
    curl_setopt_array($ch, [CURLOPT_RETURNTRANSFER => true, CURLOPT_POST => true,
        CURLOPT_POSTFIELDS => json_encode(['grant_type' => 'client_credentials',
            'client_id' => CLIENT_ID, 'client_secret' => CLIENT_SECRET, 'scope' => 'payment:create']),
        CURLOPT_HTTPHEADER => ['Content-Type: application/json'], CURLOPT_TIMEOUT => 10]);
    $data = json_decode(curl_exec($ch), true); curl_close($ch);
    return $data['access_token'] ?? throw new \RuntimeException('Token failed');
}

function sign(string $body, int $ts, string $nonce): string {
    $msg = implode("\n", [CLIENT_ID, $ts, $nonce, hash('sha256', $body)]);
    return base64_encode(hash_hmac('sha256', $msg, CLIENT_SECRET, true));
}

function listProjects(string $token, string $type = ''): array {
    $payload  = $type ? ['type' => $type] : [];
    $body     = json_encode($payload, JSON_UNESCAPED_UNICODE | JSON_UNESCAPED_SLASHES);
    $ts = time(); $nonce = bin2hex(random_bytes(8));
    $ch = curl_init(BASE_URL . '/api/v1/projects');
    curl_setopt_array($ch, [CURLOPT_RETURNTRANSFER => true, CURLOPT_POST => true,
        CURLOPT_POSTFIELDS => $body, CURLOPT_TIMEOUT => 10,
```

```
        CURLOPT_HTTPHEADER => ['Authorization: Bearer ' . $token,
            'Content-Type: application/json',
            'X-Timestamp: ' . $ts, 'X-Nonce: ' . $nonce, 'X-Signature: ' . sign($body, $ts, $nonce)]]);
    $resp = json_decode(curl_exec($ch), true); curl_close($ch);
    return $resp['data'] ?? [];
}

function createPaymentSession(string $token, string $code, string $projectType,
    float $amount, string $refNo, string $payerName = '', string $payerEmail = '',
    string $payerPhone = '', ?array $data = null): array {
    $payload = ['project_code' => $code, 'type' => $projectType, 'amount' => $amount, 'ref_no' => $refNo];
    if ($payerName)  $payload['payer_name']  = $payerName;
    if ($payerEmail) $payload['payer_email'] = $payerEmail;
    if ($payerPhone) $payload['payer_phone'] = $payerPhone;
    if ($title)      $payload['title']       = $title;
    if ($data)       $payload['data']        = $data;
    $ts = time(); $nonce = bin2hex(random_bytes(8));
    $body = json_encode($payload, JSON_UNESCAPED_UNICODE | JSON_UNESCAPED_SLASHES);
    $ch   = curl_init(BASE_URL . '/api/v1/payment/create');
    curl_setopt_array($ch, [CURLOPT_RETURNTRANSFER => true, CURLOPT_POST => true,
        CURLOPT_POSTFIELDS => $body, CURLOPT_TIMEOUT => 15,
        CURLOPT_HTTPHEADER => ['Authorization: Bearer ' . $token,
            'Content-Type: application/json',
            'X-Timestamp: ' . $ts, 'X-Nonce: ' . $nonce, 'X-Signature: ' . sign($body, $ts, $nonce)]]);
    $result = json_decode(curl_exec($ch), true); curl_close($ch);
    return $result;
}
```

```
// ──── ตัวอย่างการใช้งาน
// ═══════════════════════════════════════════════════════════════

$token    = getAccessToken();
$projects = listProjects($token, 'content');
$code     = $projects[0]['project_code'] ?? '37000001';
$type     = $projects[0]['type']         ?? 'content';
$result   = createPaymentSession(
    $token, $code, $type, 1700.00, 'ORDER-001',
    'สมชาย ใจดี', 'somchai@example.com', '0891234567',
    'อบรม Python Basics รุ่นที่ 3',
    [
        ['name' => 'ค่าลงทะเบียน',    'quantity' => 1, 'unit_price' => 1500.00],
        ['name' => 'ค่าเอกสารประกอบ', 'quantity' => 2, 'unit_price' => 100.00],
    ]
);
echo $result['payment_url'];
```

9.3 Node.js

Node.js

```
const crypto = require('crypto');

const BASE_URL      = 'https://ereceipt.ofas.chula.ac.th';
const CLIENT_ID     = '550e8400-e29b-41d4-a716-446655440000';
const CLIENT_SECRET = 'AbCd1234...';

async function getAccessToken() {
  const res = await fetch(`${BASE_URL}/oauth/token`, { method: 'POST',
    headers: { 'Content-Type': 'application/json' },
```

```
    body: JSON.stringify({ grant_type: 'client_credentials', client_id: CLIENT_ID,
      client_secret: CLIENT_SECRET, scope: 'payment:create' }) });
  return (await res.json()).access_token;
}

function sign(bodyStr, ts, nonce) {
  const h = crypto.createHash('sha256').update(bodyStr, 'utf8').digest('hex');
  const m = [CLIENT_ID, ts, nonce, h].join('\n');
  return Buffer.from(crypto.createHmac('sha256', CLIENT_SECRET).update(m).digest()).toString('base64');
}

async function listProjects(token, type = '') {
  const payload   = type ? { type } : {};
  const bodyStr   = JSON.stringify(payload);
  const ts = Math.floor(Date.now()/1000), nonce = crypto.randomBytes(8).toString('hex');
  const { data } = await postJson(`${BASE_URL}/api/v1/projects`, payload, {
    'Authorization': `Bearer ${token}`, 'Content-Type': 'application/json',
    'X-Timestamp': String(ts), 'X-Nonce': nonce,
    'X-Signature': sign(bodyStr, ts, nonce) });
  return data.data || [];
}

async function createPaymentSession({ token, projectCode, projectType, amount, refNo, payerName,
payerEmail, payerPhone, data }) {
  const payload = { project_code: projectCode, type: projectType, amount, ref_no: refNo };
  if (payerName)  payload.payer_name  = payerName;
  if (payerEmail) payload.payer_email = payerEmail;
  if (payerPhone) payload.payer_phone = payerPhone;
  if (title)      payload.title       = title;
```

```
  if (data)       payload.data        = data;
  const ts = Math.floor(Date.now()/1000), nonce = crypto.randomBytes(8).toString('hex');
  const bodyStr = JSON.stringify(payload);
  const res = await fetch(`${BASE_URL}/api/v1/payment/create`, { method: 'POST',
    headers: { 'Authorization': `Bearer ${token}`, 'Content-Type': 'application/json',
      'X-Timestamp': String(ts), 'X-Nonce': nonce, 'X-Signature': sign(bodyStr, ts, nonce) },
    body: bodyStr });
  return res.json();
}

// ──── ตัวอย่างการใช้งาน
// ──────────────────────────────────────────────────────────────────────────────
(async () => {
  const token    = await getAccessToken();
  const projects    = await listProjects(token, 'content');
  const projectCode = projects[0]?.project_code ?? '37000001';
  const projectType = projects[0]?.type         ?? 'content';
  const result   = await createPaymentSession({
    token, projectCode, projectType, amount: 1700, refNo: 'ORDER-001',
    payerName: 'สมชาย ใจดี', payerEmail: 'somchai@example.com', payerPhone: '0891234567',
    title: 'อบรม Python Basics รุ่นที่ 3',
    data: [
      { name: 'ค่าลงทะเบียน',    quantity: 1, unit_price: 1500 },
      { name: 'ค่าเอกสารประกอบ', quantity: 2, unit_price: 100 },
    ],
  });
  console.log('Payment URL:', result.payment_url);
  // res.redirect(result.payment_url);
})();
```

10. Error Codes

10.1 HTTP Status Codes

Status	ความหมาย
200	สำเร็จ
201	สร้างสำเร็จ
401	Unauthorized — token ไม่ถูกต้อง / signature ไม่ตรง / timestamp เกิน ±5 นาที / nonce ซ้ำ
403	Forbidden — Gateway ถูกระงับหรือยังไม่ผ่านการอนุมัติ
404	ไม่พบ resource ที่ขอ
422	ข้อมูลไม่ถูกต้อง — ดูรายละเอียดใน errors field (เช่น data.items sum ไม่ตรง amount)
429	Too Many Requests — Rate limit เกิน (30 req/นาที)
500	เกิดข้อผิดพลาดระบบ

10.2 ข้อผิดพลาดที่พบบ่อย

Error Message	สาเหตุ	วิธีแก้
Missing required security headers	ขาด X-Timestamp, X-Nonce, หรือ X-Signature	ตรวจสอบ headers ครบหรือไม่
Request timestamp is too old	เวลาต่างกันเกิน 5 นาที	sync NTP ให้เวลาตรง
Nonce has already been used	ส่ง nonce ซ้ำ	ใช้ nonce ใหม่ทุก request
Invalid signature	คำนวณ HMAC ผิด หรือ client_id ไม่ใช่ lowercase	ตรวจลำดับ message, lowercase UUID, client_secret
X-Nonce must be 16-64 characters	nonce สั้น/ยาวเกิน	ใช้ 16–64 chars
กรุณาระบุรหัสโครงการ	ไม่ส่ง project_code	ส่ง project_code จาก POST /api/v1/projects

กรุณาระบุประเภทโครงการ	ไม่ส่ง type	ส่ง type จาก POST /api/v1/projects (content, health_science, หรือ room)
กรุณาระบุ callback_url เมื่อใช้ mode=production	ส่ง mode=production แต่ไม่มี callback_url	ใส่ callback_url ที่เป็น HTTPS endpoint ของคุณ
กรุณาระบุ title เมื่อส่ง data	ส่ง data แต่ไม่มี title	ใส่ title เป็นชื่อหัวข้อรายการ
ผลรวมรายการ data ไม่ตรงกับ amount	sum(data[]) ≠ amount	ตรวจสอบ quantity × unit_price ทุก item

10.3 Error Response Format

JSON

```
{
  "status": "error",
  "message": "ข้อผิดพลาดหลัก",
  "errors": {
    "field_name": ["รายละเอียดข้อผิดพลาด"]
  }
}
```

11. FAQ

Q: mode=test กับ mode=production ต่างกันอย่างไร?

A: mode=test (default) — ทดสอบ flow ได้ callback_url เป็น optional ไม่ต้องมี callback server; mode=production — ใช้งานจริง callback_url required ระบบจะ POST webhook ไปที่ callback_url เมื่อชำระเงินสำเร็จ

Q: ได้รับ credentials อย่างไร?

A: ลงทะเบียนผ่าน Web UI ของ CU E-Receipt ⟶ ผู้ดูแลระบบตรวจสอบและอนุมัติ ⟶ ระบบส่ง client_id, client_secret, webhook_secret ทาง email โดยอัตโนมัติ

Q: Access Token หมดอายุต้องทำอย่างไร?

A: ขอใหม่ด้วย POST /oauth/token ด้วย credentials เดิมได้เลย ไม่มี refresh token

Q: POST /api/v1/projects คืนโครงการของคณะไหน?

A: ระบบ filter โครงการ content ตาม fac_code ของ Gateway Client อัตโนมัติ โครงการ health_science และ room คืนทั้งหมด (ไม่ filter ตามคณะ)

Q: project_code ของโครงการแต่ละประเภทมีรูปแบบอย่างไร?

A: content = รหัสที่คณะกำหนด (เช่น 37000001); health_science = integer ID ของโครงการ (เช่น 1, 2); room = รหัสห้องที่กำหนด (เช่น RM690001) — ใช้ค่าที่ได้จาก POST /api/v1/projects เสมอ อย่า hardcode

Q: ต้องส่ง type ใน POST /api/v1/payment/create ด้วยไหม?

A: ต้องส่งเสมอ — type และ project_code ต้องส่งคู่กัน ใช้ค่า type ที่ได้จาก POST /api/v1/projects ตรงๆ (content, health_science, หรือ room)

Q: data จำเป็นต้องส่งหรือไม่?

A: ไม่จำเป็น — data เป็น optional ทุก type สำหรับ type=room ให้ใช้ data สำหรับค่าใช้จ่ายเพิ่มเติมเท่านั้น (เช่น ค่าน้ำ ค่าไฟ) ไม่ใส่ค่าเช่าหลักใน data (ค่าเช่าหลักอยู่ใน room.unit_price)

Q: data sum ต้องตรงกับ amount เสมอไหม?

A: เฉพาะ type=content และ health_science — ถ้าส่ง data, sum(quantity × unit_price) ต้องตรงกับ amount (tolerance ±0.01 บาท) ระบบจะตอบ 422 ถ้าไม่ตรง

Q: ref_no มีข้อจำกัดอะไรบ้าง?

A: ref_no เป็น string สูงสุด 55 อักษร และจะถูกส่งกลับมาใน webhook payload — ใช้ track payment ระหว่าง 2 ระบบ

Q: ref_no กับ booking_ref ต่างกันอย่างไร?

A: ref_no คือเลขอ้างอิงที่คุณส่งมา (เช่น เลข order ใน system ของคุณ); booking_ref คือเลข booking ที่ระบบ CU ออกให้ (format: BK-690001)

Q: amount สำหรับ type=room ต้องส่งหรือไม่?

A: optional — ระบบคำนวณ amount = room.unit_price + sum(data) อัตโนมัติ ถ้าส่ง amount มาต้องตรงกับค่าที่คำนวณได้ (tolerance ±0.01 บาท) ระบบจะตอบ 422 ถ้าไม่ตรง

Q: type=room ต้องส่ง payer_name, payer_email, payer_phone หรือไม่?

A: ไม่บังคับ — สำหรับ type=room fields เหล่านี้เป็น optional
ผู้ชำระสามารถกรอกข้อมูลในหน้าชำระเงินได้โดยตรง

Q: ถ้า Webhook ล้มเหลวจะเกิดอะไรขึ้น?

A: ระบบ retry อัตโนมัติ 3 ครั้ง (+1 นาที ⟶ +5 นาที ⟶ +30 นาที) หากยังล้มเหลวทุกครั้ง error จะถูก log ไว้ฝั่งระบบ CU

Q: ผู้ใช้ไม่ชำระเงินภายใน 30 นาที ทำอย่างไร?

A: payment_url หมดอายุ user จะเห็นหน้า "Link หมดอายุ" — สร้าง Payment Session ใหม่ได้ทันที

Q: client_secret หายต้องทำอย่างไร?

A: ติดต่อ thannaphat.j@zenithcomp.co.th เพื่อ regenerate credentials — ไม่สามารถดึงค่าเดิมได้

Q: ทำไม signature ไม่ผ่าน?

A: ตรวจสอบ: (1) client_id เป็น lowercase UUID (2) ใช้ client_secret แบบ plain text (3) hash body จริงที่ส่ง — {} และ "" ได้ sha256 ต่างกัน (4) เวลา NTP sync

12. API Documentation (Swagger UI)

ระบบมี Swagger UI สำหรับทดสอบ API แบบ interactive แยกเป็น 2 ชุดตามกลุ่มผู้ใช้

ชุด Documentation	URL	สำหรับ
Internal	{base_url}/api/documentation	Dev ภายใน — ครบทุก endpoint
Gateway (External)	{base_url}/api/gateway-docs	External Developer — เฉพาะ Gateway API

ℹ ใช้ Gateway Documentation (/api/gateway-docs) สำหรับทดสอบ API ที่อธิบายในเอกสารนี้ — รองรับ OAuth2 token และส่ง request ได้โดยตรงจาก Swagger UI

12.1 Swagger UI URLs

Environment	Gateway Swagger URL
UAT / Staging	https://ereceipt.pay.ofas.chula.ac.th/api/gateway-docs
Production	https://ereceipt.ofas.chula.ac.th/api/gateway-docs

⚠ แนะนำให้ทดสอบใน UAT/Staging ก่อนเสมอ — Swagger UI ของ Production แสดง ⚠ STAGING warning ไว้เตือนก่อนใช้งานจริง