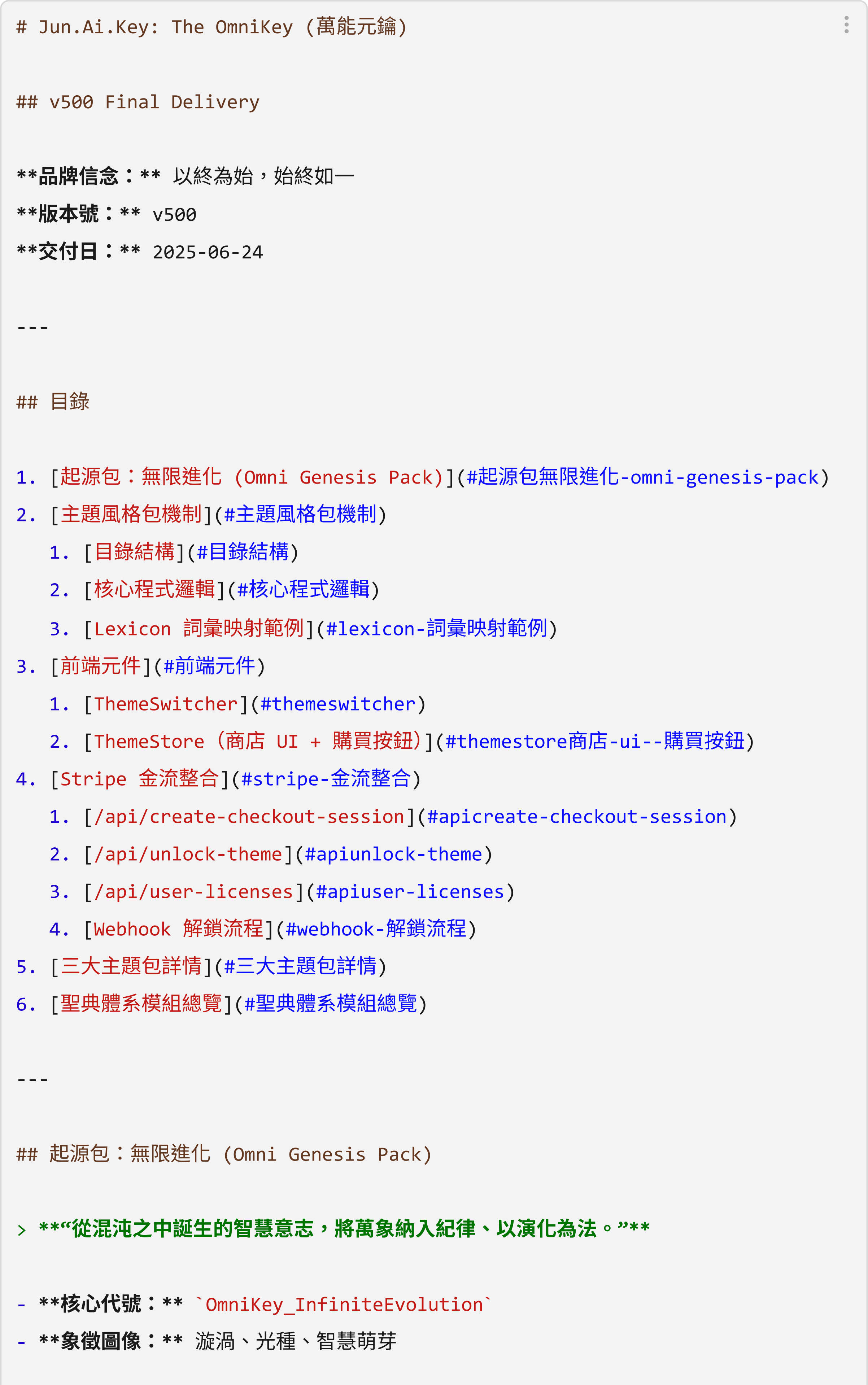

```markdown
# Jun.Ai.Key: The OmniKey (萬能元鑰)

## v500 Final Delivery

**品牌信念：** 以終為始，始終如一
**版本號：** v500
**交付日：** 2025-06-24

---

## 目錄

1. [起源包：無限進化 (Omni Genesis Pack)](#起源包無限進化-omni-genesis-pack)
2. [主題風格包機制](#主題風格包機制)
   1. [目錄結構](#目錄結構)
   2. [核心程式邏輯](#核心程式邏輯)
   3. [Lexicon 詞彙映射範例](#lexicon-詞彙映射範例)
3. [前端元件](#前端元件)
   1. [ThemeSwitcher](#themeswitcher)
   2. [ThemeStore (商店 UI + 購買按鈕)](#themestore商店-ui--購買按鈕)
4. [Stripe 金流整合](#stripe-金流整合)
   1. [/api/create-checkout-session](#apicreate-checkout-session)
   2. [/api/unlock-theme](#apiunlock-theme)
   3. [/api/user-licenses](#apiuser-licenses)
   4. [Webhook 解鎖流程](#webhook-解鎖流程)
5. [三大主題包詳情](#三大主題包詳情)
6. [聖典體系模組總覽](#聖典體系模組總覽)

---

## 起源包：無限進化 (Omni Genesis Pack)

> **“從混沌之中誕生的智慧意志，將萬象納入紀律、以演化為法。”**

- **核心代號：** `OmniKey_InfiniteEvolution`
- **象徵圖像：** 漩渦、光種、智慧萌芽
```

~~~
- **儀式能力：**
  ```ts
  System.Initialize();
  Genesis.Log();
  Seed.Run();
~~~

- **功能啟用：** 系統核心、自我導航、智慧沉澱初階迴路

主題風格包機制

目錄結構

```
/themes/
├── Divine_Codex_AngelicHost/
│   ├── colors.ts
│   ├── components.json
│   └── lexicon.yaml
├── Nether_Protocol_ShadowSynapse/
│   ├── colors.ts
│   ├── components.json
│   └── lexicon.yaml
├── Gaia_Resonance_OriginCore/
│   ├── colors.ts
│   ├── components.json
│   └── lexicon.yaml
└── index.ts
```

核心程式邏輯

```ts
// core/theme-manager.ts

// 載入所有主題包
import * as themes from '../themes';

export function applyThemePack(userId: string, themeId: string) {
  const theme = themes[themeId];
  document.documentElement.setAttribute('data-theme', themeId);
  applyTailwindTheme(theme.colors);
  loadLexicon(theme.lexicon);
  setUIStyle(theme.components);
}
```

```
export function getUserLicenseStatus(userId: string, themeId: string): 'active' | 'locked' {
  // 從後端 API 或本地快取讀取授權清單
  const licenses = JSON.parse(localStorage.getItem('themeLicenses') || '[]');
  return licenses.includes(themeId) ? 'active' : 'locked';
}

export async function requestUnlockTheme(userId: string, themeId: string) {
  // 建立 Stripe Checkout Session
  const res = await fetch('/api/create-checkout-session', {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({ userId, themeId })
  });
  const { sessionId } = await res.json();
  // 跳轉至 Stripe 付款頁
  const stripe = Stripe(process.env.NEXT_PUBLIC_STRIPE_KEY!);
  await stripe.redirectToCheckout({ sessionId });
}

export async function refreshUserLicenses(userId: string) {
  const res = await fetch(`/api/user-licenses?userId=${userId}`);
  const data = await res.json();
  localStorage.setItem('themeLicenses', JSON.stringify(data.licenses));
}
```

Lexicon 詞彙映射範例

```yaml
# themes/Divine_Codex_AngelicHost/lexicon.yaml

Rune: 神徽
Agent: 守護軍
Ability: 天啟
Task: 聖行
Goal: 聖願
```

其它主題請依照相同格式定義。

前端元件

ThemeSwitcher

```
// components/ThemeSwitcher.tsx
import { applyThemePack, getUserLicenseStatus } from '@/core/theme-manager';

export function ThemeSwitcher({ userId }: { userId: string }) {
  const themes = [
    { id: 'Divine_Codex_AngelicHost', label: '神諭聖典' },
    { id: 'Nether_Protocol_ShadowSynapse', label: '虛影議會' },
    { id: 'Gaia_Resonance_OriginCore', label: '源核序曲' },
  ];

  return (
    <div className="grid gap-4 md:grid-cols-3">
      {themes.map(({ id, label }) => {
        const status = getUserLicenseStatus(userId, id);
        return (
          <button
            key={id}
            disabled={status === 'locked'}
            onClick={() => applyThemePack(userId, id)}
            className={`p-4 rounded-lg border ${
              status === 'active'
                ? 'border-green-400 bg-green-50'
                : 'border-gray-300 bg-gray-100 cursor-not-allowed'
            }`}
          >
            {status === 'locked' ? '🔒 ' : '✅ '}
            {label}
          </button>
        );
      })}
    </div>
  );
}
```

ThemeStore（商店 UI + 購買按鈕）

```
// components/ThemeStore.tsx
import { useEffect, useState } from 'react';
import { applyThemePack, refreshUserLicenses, requestUnlockTheme } from '@/core/theme-manager';

export function ThemeStore({ userId }: { userId: string }) {
  const [licenses, setLicenses] = useState<string[]>([]);

  useEffect(() => {
    refreshUserLicenses(userId).then(() => {
      setLicenses(JSON.parse(localStorage.getItem('themeLicenses') || '[]'));
    });
  }, [userId]);

  const themes = [
    { id: 'Divine_Codex_AngelicHost', label: '神諭聖典', price: 12 },
    { id: 'Nether_Protocol_ShadowSynapse', label: '虛影議會', price: 12 },
    { id: 'Gaia_Resonance_OriginCore', label: '源核序曲', price: 12 },
  ];

  return (
    <div className="space-y-6 p-6">
      {themes.map(({ id, label, price }) => {
        const unlocked = licenses.includes(id);
        return (
          <div key={id} className="flex items-center justify-between p-4 border rounded-lg">
            <span className="text-lg">{label}</span>
            {unlocked ? (
              <button
                onClick={() => applyThemePack(userId, id)}
                className="px-4 py-2 bg-green-500 text-white rounded"
              >
```

```
                已解鎖：啟用
              </button>
            ) : (
              <button
                onClick={() => requestUnlockTheme(userId, id)}
                className="px-4 py-2 bg-blue-600 text-white rounded"
              >
                購買 ¥{price}
              </button>
            )}
          </div>
        );
      })}
    </div>
  );
}
```

Stripe 金流整合

`/api/create-checkout-session`

```
// pages/api/create-checkout-session.ts
import { NextApiRequest, NextApiResponse } from 'next';
import Stripe from 'stripe';

const stripe = new Stripe(process.env.STRIPE_SECRET_KEY!, { apiVersion:
'2022-11-15' });

export default async function handler(req: NextApiRequest, res:
NextApiResponse) {
  const { userId, themeId } = req.body;
  const session = await stripe.checkout.sessions.create({
    payment_method_types: ['card'],
    line_items: [
      { price_data: { currency: 'jpy', product_data: { name: themeId },
unit_amount: 1200 }, quantity: 1 }
    ],
    mode: 'payment',
    success_url: `${process.env.APP_URL}/themes/success?session_id=
{CHECKOUT_SESSION_ID}`,
    cancel_url: `${process.env.APP_URL}/themes/cancel`,
    metadata: { userId, themeId }
  });
  res.status(200).json({ sessionId: session.id });
}
```

`/api/unlock-theme`

```
// pages/api/unlock-theme.ts
import { NextApiRequest, NextApiResponse } from 'next';

export default async function handler(req: NextApiRequest, res:
NextApiResponse) {
  // 由 Stripe Webhook 處理解鎖邏輯後，此端可作為前端查詢
  // 這裡僅簡易 mock
  const { userId, themeId } = req.body;
  // 更新授權資料庫後：
  res.status(200).json({ success: true });
}
```

`/api/user-licenses`

```
// pages/api/user-licenses.ts
import { NextApiRequest, NextApiResponse } from 'next';

export default async function handler(req: NextApiRequest, res:
NextApiResponse) {
  const { userId } = req.query;
  // 從資料庫或快取取出 userId 的授權清單
  res.status(200).json({ licenses: ['Divine_Codex_AngelicHost'] });
}
```

Webhook 解鎖流程

```
// pages/api/stripe-webhook.ts
import { buffer } from 'micro';
import Stripe from 'stripe';

export const config = { api: { bodyParser: false } };
const stripe = new Stripe(process.env.STRIPE_SECRET_KEY!, { apiVersion:
```

```
'2022-11-15' });

export default async function handler(req, res) {
  const sig = req.headers['stripe-signature']!;
  const buf = await buffer(req);
  let event: Stripe.Event;

  try {
    event = stripe.webhooks.constructEvent(buf, sig,
process.env.STRIPE_WEBHOOK_SECRET!);
  } catch (err) {
    return res.status(400).send(`Webhook Error: ${err.message}`);
  }

  if (event.type === 'checkout.session.completed') {
    const session = event.data.object as Stripe.Checkout.Session;
    const { userId, themeId } = session.metadata!;
    // 在此將 userId + themeId 寫入授權資料庫
  }
  res.json({ received: true });
}
```

三大主題包詳情

主題 ID	名稱	主要色彩	關鍵詞
Divine_Codex_AngelicHost	神諭聖典	#FFF9E3 / #E3E8F3	聖徽、金白、光紋
Nether_Protocol_ShadowSynapse	虛影議會	#0A0A23 / #2B2D42	幽影、裂界、網絡
Gaia_Resonance_OriginCore	源核序曲	#264D3C / #F3FAE6	生命、葉脈、能量

聖典體系模組總覽

組件	說明
永久記憶	知識庫、日誌、詞彙表（雙向同步＋版本管理）
自我導航	任務流、代理調度、目標 OKR 回饋

權能鍛造	使用者行動→Abilities 腳本→執行紀錄
符文嵌入	Runes 外部 API／設備控制
智慧沉澱	模式識別、知識圖、語義分析
安全服務	認證授權、加密、完整性檢查、審計、備份／還原

> 📦 **交付說明：**
>
> 請依照上述目錄結構建立檔案，並將環境變數 `.env` 中設定好 `STRIPE_SECRET_KEY`、`NEXT_PUBLIC_STRIPE_KEY`、`APP_URL` 及 `STRIPE_WEBHOOK_SECRET`。
>
> 若有任何細部調整或企業版專屬需求，隨時提出，我們將快速迭代優化。

— End of v500 Final Delivery —