合约代码

必需依赖
确保已安装：

```
npm install thirdweb @thirdweb-dev/sdk ethers
```

1. 创建 Thirdweb Client

```
// client.ts
import { createThirdwebClient } from "thirdweb";

export const client = createThirdwebClient({
  clientId: "你的 ClientId", // 到 thirdweb dashboard 获取
});
```

2. 前端集成

```
// MintPay.tsx
import { PayEmbed } from "thirdweb/react";
import { prepareContractCall } from "thirdweb";
import { client } from "./client";

// 后端 API 拉取到的签名数据（你需要实际通过接口获取）
const signatureData = {
  payload: {
    /* see backend example 下 */
  },
  signature: "0x...", // 从你后端获取
};

const nftMeta = {
  name: "BASED KPOP NFT",
  image: "https://图片 URL",
  description: "Redefine KPOP with pioneer art.",
};

export default function MintNftPay() {
  return (
    <PayEmbed
      client={client}
      payOptions={{
        mode: "transaction",
        transaction: prepareContractCall({
          contract:
            "0x9c38ed5bC072972dffC59bfaab6ADE77D3FC285B", // BASED KPOP 合约地址
          functionName: "mintWithSignature",
          args: [
            signatureData.payload,
            signatureData.signature,
          ],
        }),
        metadata: nftMeta,
      }}
    />
  );
}
```

3. 后端签名（Node.js/TypeScript 示例）

你必须在后端用带 Minter 权限的钱包生成签名授权。每次用户支付前，拉接口获取最新 signature！

```
import { ThirdwebSDK } from "@thirdweb-dev/sdk";
import { generateMintSignature } from "thirdweb/extensions/erc1155";
import { ethers } from "ethers";

const PRIVATE_KEY = process.env.PRIVATE_KEY!; // .env
const sdk = ThirdwebSDK.fromPrivateKey(PRIVATE_KEY, "base");
const contract = await sdk.getContract(
  "0x9c38ed5bC072972dffC59bfaab6ADE77D3FC285B",
);

export async function getMintSignature(
  to: string,
  quantity: number,
  metadata: object,
) {
  const { payload, signature } =
    await generateMintSignature({
      account: await sdk.getSigner(),
      contract,
      mintRequest: {
        to,
        quantity: BigInt(quantity),
        metadata,
        pricePerToken: BigInt(7 * 1e6), // 7 USDC (6 位小数，Base USDC)
        currency:
          "0xd9AA3A156c9E2e7Fc4b90CDd1CFFE45A731B1a14", // Base USDC 地址
        validityStartTimestamp: BigInt(
          Math.floor(Date.now() / 1000),
            ),
            validityEndTimestamp: BigInt(
               Math.floor(Date.now() / 1000) + 3600,
            ),
            primarySaleRecipient: "你的收款钱包地址",
            uid: ethers.utils.hexlify(
               ethers.utils.randomBytes(32),
            ),
         },
      });
   return { payload, signature };
}
```

将此接口暴露给前端(API)，前端在每次购买请求时拉取。

4. 钱包连接嵌入与多端适配

```
// App.tsx
import { ConnectEmbed } from "thirdweb/react";
import { client } from "./client";

export default function App() {
   return <ConnectEmbed client={client} chain="base" />; // 默认 Base 链
}
```

5. NFT 列表、用户资产展示
可用 hooks 查询当前用户钱包拥有的 NFT，用于展示

```
import {
   useActiveWallet,
   useERC1155Tokens,
} from "thirdweb/react";

const { wallet } = useActiveWallet();
const { data: tokens } = useERC1155Tokens({
   contract: "0x9c38ed5bC072972dffC59bfaab6ADE77D3FC285B",
   owner: wallet?.address,
});
```

该整套方案在继续优化中