JANZEER BLOCKCHAIN

دليل الربط لمنصات التداول وجهات الحفظ

العقدة · الإيداعات · السحوبات · المعروض

إصدار الوثيقة 1.0 · شبكة جنزير · 2026

جنزير — دليل الربط لمنصات التداول وجهات الحفظ

هذا الدليل موجَّه إلى كل خدمة تحتفظ بعملة JNZ لمستخدميها: منصة تداول، جهة حفظ، معالج مدفوعات، خدمة تبديل. يذكر ما تحتاجه هذه الخدمة فقط. الواجهة البرمجية الكاملة في دليل المطوّر، وقواعد التوقيع على مستوى البايت في الورقة الصفراء.

جنزير شبكة مستقلة من الطبقة الأولى. وهي **غير متوافقة مع EVM**: لن تعمل معها مكتبة إيثيريوم ولا محفظة إيثيريوم ولا طريقة اشتقاق عناوين إيثيريوم، مع أن شكل العناوين متشابه.

1. الحقائق باختصار

البند	القيمة
معرّف الشبكة	`janzeer` (الشبكة الرئيسية). يدخل في كل معاملة موقَّعة، فلا يمكن إعادة تنفيذ المعاملة على شبكة أخرى.
العملة	JNZ بثماني خانات عشرية. المبالغ في الواجهة نصوص عشرية (`"2.5"`).
المعروض	100,000,000 JNZ ثابت منذ التكوين. لا يُسكّ شيء ولا يُحرق؛ مكافآت الكتل تُدفع من حساب الخزينة.
نموذج الحسابات	حسابات لها أرصدة وعدّاد (nonce) لكل مُرسِل، وليس UTXO.
زمن الكتلة	15 ثانية.
النهائية	**كتلة واحدة.** المعاملة الموجودة في كتلة معتمدة نهائية؛ لا إعادة تنظيم للسلسلة ولا عدد تأكيدات.
العنوان	`0x` متبوعة بـ 40 خانة ست عشرية.
التوقيع	secp256k1 حتمي (RFC 6979) بقيمة S منخفضة، بترميز DER ثم Base64، على ملخّص SHA-256 مزدوج.
العقود الذكية	لا توجد. يوجد معيار واحد للعملات الأصلية (JZT-1): إنشاء، سكّ، حرق، سقف، تحويل.
الرسم	يُدفع بعملة JNZ. التحويل: الحد الأدنى `0.01` . لا سوق رسوم؛ الحد الأدنى كافٍ دائمًا.
المذكّرة	حقل اختياري `data` في التحويل، حتى 256 بايت.
أقل تحويل	`0.1` JNZ (يُقبل أي مبلغ عند إرفاق مذكّرة).
الواجهات	REST تحت `api/v1/`، و JSON-RPC 2.0 على `POST /rpc`، و WebSocket على `rpc/ws/` .
الحِزم البرمجية	TypeScript و Dart و Kotlin/JVM و Python و Go، وعميل سطر الأوامر `jz` . كلها مفتوحة المصدر.
المستكشف	https://explorer.janzeer.org

2. شغّل عقدتك الخاصة

على الخدمة أن تقرأ السلسلة من عقدة تشغّلها بنفسها. العقدة ملف تنفيذي أصلي واحد؛ لا تحتاج إلى JVM ولا إلى خادم قاعدة بيانات.

1. نزّل الحزمة المناسبة لمنصتك (لينكس x64، لينكس arm64، ويندوز x64، حزمة آرتش لينكس، أو صورة دوكر `ghcr.io/janzeerorg/janzeer-node`) من https://janzeer.org/download أو من إصدارات GitHub.
2. تحقّق منها. كل إصدار يحمل ملف `SHA256SUMS` موقَّعًا بمفتاح minisign الخاص بالمشروع:

```
minisign -Vm SHA256SUMS -P RWQBrkWzg4V5n5wUBa9GNPpa95Z7WFT2EY5QrabMv/DP+KG+gJsPIgdr
sha256sum -c SHA256SUMS --ignore-missing
```

3. فكّ الضغط وشغّلها بملف التشغيل الموجود في الحزمة. تنشئ العقدة هويتها، وتتصل بالشبكة، وتتزامن. وهي عقدة **متابِعة فقط**: تتابع السلسلة وتتحقق منها، ولا تصوّت، ولا تحتاج إلى منفذ وارد.
4. انتظر حتى يُظهر `GET /api/v1/info/sync` القيمة `"syncStatus": "SYNCHRONIZED"`.

ملاحظات تشغيلية:

- تستمع الواجهة على المنفذ `7019`. أبقِها على واجهة شبكة خاصة؛ العقدة لا تحمل مفاتيح، لكن الواجهة خاصة بك.
- تعيد عقدتك تنفيذ كل كتلة معتمدة وتتحقق منها قبل تخزينها، فهي لا تثق بالعقد التي تتزامن منها.
- `GET /actuator/health` هو فحص الحياة. و `GET /api/v1/info/update` يخبرك بوجود إصدار أحدث؛ وملف `update.sh` في الحزمة يثبّته بعد التحقق من التوقيع. الإصدارات وأي تغيير في البروتوكول يُعلن عنها في قناة المشروع (https://t.me/janzeerorg).
- طلبات القراءة بلا حدّ. طلبات الإرسال `POST` و `rpc/` تسمح بعشرين طلبًا في الثانية لكل عنوان IP مع اندفاعة حتى 40 (الرد `429` مع `Retry-After` بعد ذلك).

يمكنك استخدام العقدة العامة `https://explorer.janzeer.org` للاختبار. لا تبنِ خدمة إنتاجية عليها.

3. العناوين

- العنوان هو أول 20 بايتًا من `keccak256(المفتاح العام المضغوط)`، ويُكتب `0x` متبوعة بـ 40 خانة ست عشرية. هذه **ليست** قاعدة إيثيريوم (إيثيريوم يجزّئ المفتاح غير المضغوط)، فالمفتاح الخاص نفسه يعطي عنوانًا مختلفًا عن عنوانه في إيثيريوم.
- الصيغة المعتمدة بأحرف صغيرة. توجد صيغة بأحرف مختلطة للتحقق وهي مقبولة؛ قارن العناوين دون تمييز حالة الأحرف وخزّنها بأحرف صغيرة.
- تحقّق من العنوان قبل قبوله من المستخدم. دون اتصال: الدالة `isValidAddress` في حزمة TypeScript ونظيراتها في الحِزم الأخرى. عبر الشبكة: `POST /api/v1/wallets/validateAddress` يعيد `400` للعنوان غير السليم.
- كل مفتاح خاص سليم من نوع secp256k1 هو حساب. لا يحتاج الحساب إلى «تفعيل»: يصبح موجودًا عند أول استلام.

عناوين الإيداع

النموذجان الشائعان كلاهما يعمل:

- **عنوان لكل مستخدم.** ولّد مفتاحًا لكل مستخدم، أو اشتقّها من بذرة واحدة بدالة الاشتقاق في الحزمة البرمجية (`derivePath`). راقبها كلها (القسم 4).
- **عنوان مشترك مع مذكّرة.** أعطِ كل مستخدم قيمة مذكّرة واطلب منه وضعها في حقل المذكّرة في التحويل (`data`، حتى 256 بايت). في المحافظ الرسمية حقل للمذكّرة في شاشة الإرسال.

4. اكتشاف الإيداعات

لأن النهائية كتلة واحدة، يمكن قيد الإيداع فور ظهور معاملته بالحالة `FINAL`.

الدفع الفوري (الموصى به). افتح اتصال WebSocket على `rpc/ws/` واشترك بعناوينك:

```
{"jsonrpc":"2.0","id":1,"method":"janzeer_subscribe",
 "params":{"kind":"addressActivity","params":{"addresses":["0x…","0x…"]}}}
```

تصل كل معاملة نهائية تمسّ عنوانًا مراقَبًا في إشعار يحمل ارتفاع الكتلة وتجزئتها والمعاملة. يقبل الاشتراك الواحد حتى 1,000 عنوان، والاتصال الواحد حتى 16 اشتراكًا.

مسح الكتل (شبكة الأمان). اشترك بـ `newBlocks` مع `fromHeight` يساوي آخر ارتفاع عالجته، أو استعلم دوريًا بـ `janzeer_getBlocks` (`fromHeight` و `toHeight` و `includeTransactions: true` ، حتى 100 كتلة في الطلب). تصل الكتل بالترتيب، فبعد إعادة التشغيل تكمل من الارتفاع المخزَّن ولا يفوتك شيء. شغّل هذا المسح حتى لو استخدمت الدفع الفوري.

لكل عنوان. `janzeer_getTransactionsByAddress` (`address` و `page` و `size`) يسرد التحويلات النهائية المرسَلة والمستلَمة، مع المعاملات المعلّقة للعنوان.

قواعد القيد:

1. لا تقيّد إلا معاملة حالتها `FINAL` وإيصالها يقول `successful: true` .
2. استخدم تجزئة المعاملة مفتاحًا لمنع التكرار. التجزئة لا تظهر في كتلتين.
3. قيّد قيمة الحقل `amount` . الرسم يدفعه المُرسِل فوق المبلغ؛ المستلم يستلم المبلغ كاملًا.
4. مكافآت الكتل تصل أيضًا إلى محافظ المُصادِقين كمعاملات من النوع `reward` . تجاهلها ما لم يكن العنوان لمُصادِق.
5. لا تقيّد معاملة `PENDING` أبدًا. المعاملة المعلّقة قد يستبدلها مُرسِلها أو تنتهي صلاحيتها.

5. إرسال السحوبات

العقدة لا تحمل مفاتيح أبدًا. تبني المعاملة وتوقّعها في نظامك ثم ترسل النتيجة الموقَّعة.

1. اقرأ العدّاد التالي: `janzeer_getNonce` (`address`) . يحسب معاملاتك المعلّقة أيضًا.
2. ابنِ تحويلًا ووقّعه بإحدى الحِزم البرمجية (`TxBuilder` في كل حزمة)، برسم `"0.01"` ومعرّف الشبكة `janzeer` . التوقيع لا يحتاج إلى اتصال بالشبكة، فيمكن تنفيذه على جهاز معزول.
3. أرسل: `janzeer_sendTransfer` مع الكائن الموقَّع، أو `POST /api/v1/transactions/transfers` . الرد `{"status: "PENDING` . إعادة إرسال المعاملة الموقَّعة نفسها لا تضر.
4. انتظر: `janzeer_waitForFinality` (`hash` و `timeoutMs` حتى 60,000) أو استعلم بـ `janzeer_getTransactionByHash` . اعتبر السحب منجزًا عند `FINAL` .

قواعد العدّاد وقائمة الانتظار التي تهم المحفظة الساخنة:

- العدّاد متسلسل لكل مُرسِل ويبدأ من 0. تُرفض المعاملة التي تترك فجوة.
- يُسمح بـ 64 معاملة معلّقة على الأكثر لكل مُرسِل. لمعدّل أعلى استخدم عدة محافظ ساخنة.
- لا تُستبدل المعاملة المعلّقة إلا بمعاملة لها **العدّاد نفسه ورسم أعلى حتمًا**.
- تُحذف المعاملة المعلّقة التي لا تدخل كتلة خلال 6 ساعات. عندها وقّعها من جديد بالعدّاد الذي تعيده العقدة.
- الأخطاء تسمّي سببها: `INVALID_NONCE` و `INSUFFICIENT_BALANCE` و `INCORRECT_SIGNATURE` و `INCORRECT_ADDRESS` (في REST: الحقل `type` في جسم الخطأ؛ في JSON-RPC: الرمز `-32001` مع `data.type`).

إرسال كامل بلغة TypeScript (`npm install janzeer-sdk`):

```
import { Account, JanzeerClient, JanzeerRpc, TxBuilder, waitForFinality } from
'janzeer-sdk';

const client = new JanzeerClient('http://127.0.0.1:7019');
const rpc = new JanzeerRpc('http://127.0.0.1:7019');
const hot = Account.fromMnemonic(process.env.HOT_WALLET_MNEMONIC!);

const tx = hot.signTx(TxBuilder.transfer({
  from: hot.address, to: '0x598b1301acef3baba6ce25e38dd17b723f7b98b1',
  amount: '1.25', data: 'withdrawal 8841', nonce: await client.nonce(hot.address),
}));
await client.submit(tx);                       // {hash, status: "PENDING"}
const done = await waitForFinality(rpc, tx.hash);
console.log(done.blockHeight, done.receipt?.successful);
```

المسار نفسه موجود في كل حزمة برمجية، وكل حزمة تحمله ملفًا قابلًا للتشغيل في مجلد `/examples` .

6. الأرصدة والمطابقة

- (`address`) `janzeer_getAccount` يعيد `committedBalance` (الرصيد النهائي) و `balance` (القابل للصرف: النهائي ناقص خصومك المعلّقة) و `nextNonce` و `pendingCount` .
- لا يتغيّر الرصيد إلا بمعاملات داخل كتل، فمجموع الإيداعات المقيَّدة ناقص السحوبات والرسوم يجب أن يساوي `committedBalance` عند أي ارتفاع.
- `janzeer_getTip` يعطي الارتفاع الحالي وتجزئته. خزّن الارتفاع الذي طابقت عنده.

7. أرقام المعروض

متاحة منذ العقدة 0.1.14:

المسار	النتيجة
`GET /api/v1/supply/total`	المعروض الكلي رقمًا واحدًا مجزّدًا (`100000000`).
`GET /api/v1/supply/circulating`	المعروض المتداول رقمًا واحدًا مجزّدًا.
`GET /api/v1/supply`	الرقمان مع الحسابات غير المتداولة وأرصدتها الحالية.

المعروض الكلي ثابت. **المعروض المتداول** يُحسب كما تحسبه مواقع بيانات السوق: الكلي ناقص كل عملة يحتفظ بها البروتوكول أو المشروع، بالأرصدة الحالية. هذه الحسابات هي الخزينة التي لا مفتاح لها (`0x0000000000000000000000000000000000000000`)، وهي تدفع مكافآت الكتل وتستلم 20% من كل رسم)، ومخصّصا المؤسِّس والمنظومة في التكوين، ومحافظ المُصادِقين المؤسِّسين، وأي محفظة أخرى يحتفظ بها المشروع. يسرد `GET /api/v1/supply` كل حساب منها بعنوانه ورصيده، فيستطيع أي أحد إعادة حساب الرقم. ويعرض `GET /api/v1/explorer/info` رقم المتداول نفسه (قبل 0.1.14 كان يطرح الخزينة فقط).

8. عملات JZT-1

العملات الصادرة وفق معيار العملات الأصلية تعيش في الحسابات نفسها. تصل إيداعاتها عبر تغذية `addressActivity` نفسها كمعاملات من النوع `token` ، وتُقرأ الأرصدة بـ `janzeer_getTokenBalances` . مبالغ العملات أعداد صحيحة بوحدات العملة

الأساسية تُرسل نصوصًا، ولكل عملة عدد خاناتها العشرية. رسم تحويل العملة يُدفع بـ JNZ.

9. قائمة التحقق قبل التشغيل الفعلي

- [ ] عقدتك الخاصة متزامنة ومراقَبة (`actuator/health/` و `api/v1/info/sync/`).
- [ ] العناوين يُتحقق منها عند الإدخال وتُقارن دون تمييز حالة الأحرف.
- [ ] الإيداعات تُقيَّد عند `FINAL` فقط مع إيصال ناجح، مرة واحدة لكل تجزئة معاملة.
- [ ] مسح الكتل يكمل من ارتفاع مخزَّن بعد إعادة التشغيل.
- [ ] السحوبات تعالج `INVALID_NONCE` بإعادة قراءة العدّاد، وتبقى دون 64 معاملة معلّقة لكل محفظة ساخنة.
- [ ] مفاتيح التوقيع لا تلامس العقدة؛ وواجهة العقدة غير مكشوفة للإنترنت.
- [ ] جرى إيداع تجريبي وسحب تجريبي كاملان على الشبكة الرئيسية بمبلغ صغير.

10. التواصل

للأسئلة التقنية: افتح بلاغًا على https://github.com/janzeerOrg/janzeer-node/issues أو راسل info@janzeer.org. يجيب المشروع عن أسئلة الربط مباشرة، ويمكنه تقديم مبلغ تجريبي من JNZ لتجربة الربط.

حالة الوثيقة: الإصدار 1.1 (13-11-2026). تعكس العقدة 0.1.14.