JANZEER BLOCKCHAIN

مرجع واجهة REST البرمجية

نقاط النهاية والأظرف والأمثلة

إصدار الوثيقة 1.0 · شبكة جنزير · 2026

مرجع الواجهة البرمجية REST لـ Janzeer

هذا هو المرجع الكامل الموجّه للبحث لواجهة HTTP البرمجية الخاصة بعقدة **Janzeer**: كل مورد ومنهج ومسار ومُعامِل وجسم طلب وحمولة مُعادة. إن Janzeer هي طبقة أولى (Layer-1) بمعروض ثابت، بلا إثبات حصة (**stake-free**)، ومفتوحة العضوية بنهائية BFT، مع مخطط تشفيري لمحفظة HD غير حافظة — تتحقّق العقدة من التواقيع وتطبّق انتقالات الحالة لكنها **لا تحتفظ بالمفاتيح أبداً**، لذا لا توجد هنا أي نقاط نهاية لتوليد المفاتيح أو عبارة الاسترداد أو التوقيع. للحصول على شرح بأسلوب تعليمي (تشغيل عقدة، خط أنابيب التوقيع من جهة العميل، متجهات التكافؤ) راجع دليل المطوّر؛ وللاطّلاع على مخططات البايتات الدقيقة والثوابت راجع الورقة الصفراء. هذه الوثيقة مرجع وليست درساً تعليمياً.

> **مجموعات أدوات العميل.** توجد عملاء رسمية مُنمَّطة لـ TypeScript (`janzeer-sdk`) وDart (`janzeer_sdk`) وKotlin/JVM/Android (`org.janzeer:janzeer-sdk-client`)؛ وهي تغلّف كل نقاط النهاية أدناه إضافةً إلى JSON-RPC وتغذيات WebSocket — راجِع دليل المطوّر §5. وعلى العملاء المكتوبة يدويًا قراءة §1.3 بعناية (الأخطاء مُغلَّفة).

---

1. الاصطلاحات

1.1 عنوان URL الأساسي والمنفذ

العنصر	القيمة
المسار الأساسي	`/api/v1/`
منفذ REST	`7019` (`server.port`)
مضيف مثال	`http://localhost:7019`
نوع المحتوى	`application/json` (الطلب والاستجابة)

جميع مسارات نقاط النهاية في هذه الوثيقة نسبية إلى `http://<host>:7019`. ما لم يُذكر المنهج صراحةً، فإن كل نقطة نهاية هي `GET`.

1.2 غلاف الاستجابة

كل استجابة تحت `/api/v1/**` تُلَفّ في غلاف قياسي؛ والنتيجة الفعلية تكون دائماً في `payload`:

```
{
  "timestamp": 1719700000000,
  "version": "1.1.0",
  "payload": { }
}
```

الحقل	النوع	المعنى
`timestamp`	`Long`	وقت الخادم (ميلي ثانية منذ الحقبة) عند إنتاج الاستجابة
`version`	`String`	إصدار الغلاف، حالياً `"1.1.0"`
`payload`	`T`	النتيجة الخاصة بنقطة النهاية (كائن أو مصفوفة أو قيمة قياسية أو `null`)

المسارات التي لا تُلَفّ (تُعاد خاماً، بدون غلاف):

- `/**/docs/` (واجهة Swagger UI ووثيقة OpenAPI)
- `swagger-resources/`
- `/actuator/**` (Spring Boot Actuator)

1.3 شكل الخطأ ورموز الحالة

تستخدم الأخطاء الغلاف نفسه المستخدم في كل استجابة أخرى؛ كائن الخطأ هو `payload`:

```
{
  "timestamp": 1757800000000,
  "version": "1.1.0",
  "payload": { "status": 400, "message": "Invalid nonce for 0x06e1…: expected 4, got 7", "type": "INVALID_NONCE" }
}
```

الحقل	النوع	المعنى
`status`	`Int`	رمز حالة HTTP، مُكرَّر في الجسم
`message`	`String`	وصف مقروء للبشر
`type`	`?String`	نوع الرفض المقروء آليًا عند فشل التحقّق من معاملة: `INCORRECT_ADDRESS`، `INCORRECT_HASH`، `ALREADY_PROMOTER`، `INCORRECT_PROMOTER_KEY`، `INCORRECT_SIGNATURE`، `INSUFFICIENT_ACTUAL_BALANCE`، `INSUFFICIENT_BALANCE`، `INVALID_NONCE` (تحمل الرسالة `expected N, got M`)؛ غائب في الأخطاء الأخرى

استثناءان: أخطاء التحقّق من جسم الطلب (حقول ناقصة/غير صالحة) تُعيد `payload` على شكل **مصفوفة** من كائنات `{message}`، وأجسام `413` / `429` الصادرة عن محدِّد المعدل تُكتب قبل طبقة الغلاف فتكون مسطّحة `{status, message}`. تتعامل مجموعات الأدوات (SDK) الرسمية مع الأشكال الثلاثة؛ وعلى أي عميل مكتوب يدويًا التحقّق من وجود `payload`.

الحالة	متى
`Bad Request 400`	فشل التحقّق من صحة الطلب (جسم مشوّه، مُعامِل خاطئ، فشل التحقّق التجاري)
`Not Found 404`	المورد المطلوب (تجزئة، عنوان، كتلة) غير موجود

1.4 ترقيم الصفحات

تقبل نقاط نهاية القوائم مُعامِلات الاستعلام التالية:

المُعامِل	النوع	المدى / الافتراضي	المعنى
`page`	`Int`	`0` ≤، الافتراضي `0`	فهرس الصفحة المبني على الصفر
`size`	`Int`	`1–100`، الافتراضي `10`	حجم الصفحة
`sortBy`	`String`	اسم الحقل (يعتمد على المورد)	حقل الترتيب
`sortDirection`	`String`	`DESC` / `ASC`، الافتراضي `DESC`	اتجاه الترتيب

تأخذ الحمولة المُرقَّمة صفحات هذا الشكل:

```
{
  "total": 42,
  "list": [ ],
  "page": 0,
  "pageSize": 10,
  "totalPages": 5
}
```

الحقل	النوع	المعنى
`total`	`Long`	إجمالي السجلات المطابقة عبر كل الصفحات
`list`	`[]T`	السجلات في هذه الصفحة
`page`	`Int`	فهرس الصفحة المبني على الصفر، مُكرَّر
`pageSize`	`Int`	حجم الصفحة، مُكرَّر
`totalPages`	`Int`	إجمالي عدد الصفحات

1.5 المبالغ

جميع القيم النقدية (الأرصدة، الرسوم، المبالغ، المكافآت، المعروض) عشرية (`BigDecimal`) بمقياس **8 منازل عشرية**. راجع الورقة الصفراء للاطّلاع على عملية التحجيم `toScaledLong` المستخدمة في الصور الأولية الموقَّعة. تُستثنى من ذلك مبالغ الرموز الأصيلة: فهي **وحدات أساسية صحيحة** تُنقَل كسلاسل عشرية (انظر §7).

1.6 CORS ونوع المحتوى

- الطلبات والاستجابات بصيغة (`Content-Type: application/json`) JSON.
- يُسمح بـ CORS، لذا يمكن للمحافظ والمستكشفات العاملة في المتصفح استدعاء الواجهة البرمجية عبر النطاقات.

1.7 النموذج غير الحافظ

تتحقّق العقدة من التواقيع وتطبّق الحالة؛ وهي **لا** تدير المفاتيح. لا توجد **أي** نقاط نهاية لتوليد المفاتيح أو عبارات الاسترداد أو الكلمات التذكّرية أو التوقيع. يشتقّ العملاء المفاتيح ويبنون الصورة الأولية ويُجزّئونها ويوقّعونها بالكامل من جهة العميل، ثم يرسلون المعاملات الجاهزة إلى نقاط نهاية `POST` في §3–§9. راجع §12 ودليل المطوّر / الورقة الصفراء للاطّلاع على خط أنابيب التوقيع.

---

2. خريطة الموارد

مجموعة الموارد	المسار الأساسي	القسم
المحافظ	`api/v1/wallets/`	3§
معاملات التحويل	`api/v1/transactions/transfers/`	4§
معاملات المُصادِق (التسجيل)	`api/v1/transactions/validators/`	5§
معاملات خروج المُصادِق	`api/v1/transactions/exit-validators/`	6§
الرموز الأصيلة (المعاملات + الحالة)	`api/v1/transactions/tokens` ، `/api/v1/tokens/`	7§
معاملات المكافأة (للقراءة فقط)	`api/v1/transactions/rewards/`	8§
الكتل الرئيسية	`api/v1/blocks/main/`	9§
الكتل التكوينية	`api/v1/blocks/genesis/`	9§
المُصادِقون (الحالة)	`api/v1/validators/`	10§
المستكشف	`api/v1/explorer/`	11§
معلومات العقدة	`api/v1/info/`	11§
أجسام طلبات المعاملات	(POST)	12§
Actuator	`actuator/`	13§
وثائق الواجهة البرمجية (Swagger / OpenAPI)	`docs/`	14§

---

3. المحافظ — `api/v1/wallets/`

نقاط نهاية للقراءة على مستوى الحساب بالإضافة إلى مساعد تحقّق واحد. لا يكشف أيٌّ منها عن المفاتيح ولا يقبلها.

المنهج	المسار	المُعامِلات	يُعيد
GET	`api/v1/wallets/{address}/`	path `address`	الرصيد المؤكَّد (`BigDecimal`): المؤكَّد – غير المؤكَّد، بحدّ أدنى `0`
GET	`api/v1/wallets/{address}/nonce/`	path `address`	الرقم التسلسلي (nonce) التالي المتوقَّع (`Long`) للعنوان
POST	`api/v1/wallets/validateAddress/`	body: سلسلة العنوان	نتيجة صلاحية العنوان / تحليله

مُعامِل المسار:

المُعامِل	النوع	ملاحظات
`address`	`String`	`0x` + 40 محرفاً سداسياً عشرياً صغيراً

مثال — الرصيد:

```
curl http://localhost:7019/api/v1/wallets/0x06e1c0fa9955a700876f8cb0acc7f13fba9fb8ba
```

```
{ "timestamp": 1719700000000, "version": "1.1.0", "payload": "125.40000000" }
```

مثال — الرقم التسلسلي (nonce):

```
curl
http://localhost:7019/api/v1/wallets/0x06e1c0fa9955a700876f8cb0acc7f13fba9fb8ba/nonce
```

```
{ "timestamp": 1719700000000, "version": "1.1.0", "payload": 0 }
```

الرقم التسلسلي (nonce) المُعاد هو الرقم التسلسلي **التالي** المتوقَّع للحساب. تسمح ذاكرة المعاملات المعلّقة (mempool) بأرقام تسلسلية **متتالية** (الأنبَبة): الرقم التسلسلي التالي القابل للاستخدام = `confirmedNonce + pendingCount`.

---

4. معاملات التحويل — `api/v1/transactions/transfers/`

نقل العملات، اختيارياً مع مذكّرة (≤ 256 بايت) `data`.

المنهج	المسار	الاستعلام / الجسم	يُعيد
GET	`api/v1/transactions/transfers/`	`address`، `unconfirmed` + ترقيم الصفحات	تحويلات مُرقَّمة صفحات
GET	`api/v1/transactions/transfers/{hash}/`	path `hash`	تحويل واحد بحسب تجزئة المعاملة
POST	`api/v1/transactions/transfers/`	جسم معاملة التحويل (§12)	يقبل تحويلاً ← `201 Created`

مُعامِلات الاستعلام (القائمة):

المُعامِل	النوع	المعنى
`address`	`?String`	تصفية حسب عنوان المُرسِل/المُستلِم
`unconfirmed`	`?Boolean`	تضمين التحويلات غير المؤكَّدة (من ذاكرة المعاملات المعلّقة)
`page`, `size`, `sortBy`, `sortDirection`	—	ترقيم الصفحات (§1.4)

حقول الجسم الخاصة بـ POST (بالإضافة إلى الحقول المشتركة في §12):

الحقل	النوع	ملاحظات
`amount`	`BigDecimal`	≥ `0.1` (`MINIMUM_AMOUNT`)، ما لم تُرفَق مذكّرة
`recipientAddress`	`?String`	عنوان الوجهة
`data`	`?String`	مذكّرة اختيارية، ≥ `256` بايت (`MAX_DATA_SIZE`)

الرسوم: ≥ `0.01` (`MINIMUM_FEE`).

---

5. معاملات المُصادِق (التسجيل) — `api/v1/transactions/validators/`

تسجيل العقدة كمُصادِق. العضوية مفتوحة — شغّل عقدةً وأرسِل `PromoterTx` تحمل وديعة التسجيل غير القابلة للاسترداد؛ لا يوجد تصويت ولا موافقة ولا قفل للأموال. ملاحظة: تحتفظ أسماء أنواع (TYPE) المعاملات/الحالة الداخلية، ومفاتيح الإعداد `promoters-count` / `promoter-tx-*` ، وجداول قاعدة البيانات بالاسم القديم **promoter** — أما نقاط النهاية وحقول JSON فهي validator (مُصادِق).

المنهج	المسار	الاستعلام / الجسم	يُعيد
GET	`api/v1/transactions/validators/`	ترقيم الصفحات	معاملات تسجيل مُصادِقين مُرقَّمة صفحات
GET	`api/v1/transactions/validators/{hash}/`	path `hash`	معاملة مُصادِق واحدة
POST	`api/v1/transactions/validators/`	جسم معاملة المُصادِق (12§)	يُسجّل ← `Created 201`

حقول الجسم الخاصة بـ POST:

الحقل	النوع	ملاحظات
`amount`	`BigDecimal`	وديعة التسجيل، تحديداً `2000` ← بنك الخزينة، **غير قابلة للاسترداد**
`validatorKey`	`String`	يجب أن يكون فريداً

الرسوم: تحديداً `3` . تُعدّ وديعة الـ `2000` الحاجز الأساسي ضد هجوم Sybil؛ وهي تذهب إلى بنك الخزينة ولا تُعاد أبداً (الخروج لا يستردّها).

---

6. معاملات خروج المُصادِق — `api/v1/transactions/exit-validators/`

اعتزال مُصادِق بصورة سلسة. تزيل `ExitPromoterTx` المُصادِقَ من المجموعة النشطة ومن جدول الفترات (slots) عند حدود الحقبة (epoch) التالية. تبقى وديعة التسجيل الأصلية غير قابلة للاسترداد.

المنهج	المسار	الاستعلام / الجسم	يُعيد
GET	`api/v1/transactions/exit-validators/`	ترقيم الصفحات	معاملات خروج مُصادِقين مُرقَّمة صفحات
GET	`api/v1/transactions/exit-/validators/{hash}`	path `hash`	معاملة خروج مُصادِق واحدة
POST	`api/v1/transactions/exit-validators/`	جسم معاملة خروج المُصادِق (12§)	يعتزل ← `Created 201`

حقول الجسم الخاصة بـ POST:

الحقل	النوع	ملاحظات
`validatorKey`	`String`	مفتاح المُصادِق الجاري اعتزاله

لا يوجد `amount` (الوديعة غير قابلة للاسترداد). الرسوم: ≤ `0.01` (`MINIMUM_FEE`).

---

7. الرموز الأصيلة — `api/v1/transactions/tokens/` و `api/v1/tokens/`

الرموز الأصيلة (JZT-1) هي رموز قابلة للاستبدال يعرّفها المستخدم اختيارياً وتعيش جنباً إلى جنب مع العملة الأصيلة على السلسلة نفسها، على **دفتر منفصل** — فمعروض الرمز ليس جزءاً من المعروض الثابت للعملة البالغ `100,000,000` ولا يمسّ الإصدار أو الخزينة أو مكافآت الكتل أبداً. ومع ذلك تدفع كل عملية على رمز رسومها **بالعملة الأصيلة**. نوع معاملة واحد، `TokenTx` ، يحمل بايت `op` يختار العملية؛ وتوفّر مجموعة مرافقة من نقاط نهاية القراءة (7.2§) تعريفات الرموز ومعروضها وأرصدتها.

الصلاحية: فقط **مُصادِق مُسجَّل** (مُصادِق، نشط أو احتياطي) يمكنه `CREATE` رمزاً؛ وفقط **مُصدِر** الرمز (مُنشِئه) يمكنه `MINT` أو `BURN` أو `SETCAP` ؛ و**أي شخص** يملك رصيداً يمكنه `TRANSFER` . إن **`tokenId` للرمز الجديد هو تجزئة معاملة CREATE** (فريد عالمياً؛ لا يوجد سجلّ للرموز، لذا الرموز (symbols) ليست فريدة).

7.1 معاملات الرموز — `api/v1/transactions/tokens/`

المنهج	المسار	الاستعلام / الجسم	يُعيد
GET	`api/v1/transactions/tokens/`	`unconfirmed` ، `tokenId` + ترقيم الصفحات	معاملات رموز مُرقَّمة صفحات
GET	`api/v1/transactions/tokens/{hash}/`	path `hash`	معاملة رمز واحدة بحسب التجزئة
POST	`api/v1/transactions/tokens/`	جسم معاملة الرمز (12§)	يقبل عملية رمز ← `201 Created`

مُعامِلات الاستعلام (القائمة):

المُعامِل	النوع	المعنى
`tokenId`	`?String`	تصفية حسب معرّف الرمز (تجزئة معاملة CREATE)
`unconfirmed`	`?Boolean`	تضمين معاملات الرموز غير المؤكَّدة (من ذاكرة المعاملات المعلّقة)
`page` , `size` , `sortBy` , `sortDirection`	—	ترقيم الصفحات (§1.4)

حقول الجسم الخاصة بـ POST (بالإضافة إلى الحقول المشتركة في §12):

الحقل	النوع	ملاحظات
`op`	`Int`	العملية: `0` TRANSFER، `4` SETCAP، `3` BURN، `2` MINT، `1` CREATE
`tokenId`	`?String`	الرمز الهدف (تجزئة معاملة CREATE الخاصة به)؛ يُحذَف عند CREATE
`symbol`	`?String`	CREATE: الرمز المختصر، `1`–`12` محرفاً
`name`	`?String`	CREATE: اسم الرمز، `1`–`40` محرفاً
`decimals`	`?Int`	CREATE: منازل العرض العشرية، `0`–`18` (للعرض فقط)
`cap`	`?String`	المعروض الأقصى، **سلسلة** عشرية بوحدات أساسية صحيحة (CREATE / SETCAP)
`amount`	`?String`	**سلسلة** عشرية بوحدات أساسية صحيحة (المعروض الأولي لـ CREATE / MINT / BURN / TRANSFER)
`recipient`	`?String`	عنوان وجهة TRANSFER

إن `cap` و`amount` للرمز هما **وحدات أساسية صحيحة** (مثل الوحدات الخام في ERC-20)، تُنقَل كسلسلتها العشرية **string** — أرسِلهما كسلاسل JSON لتجنّب فقدان الدقة؛ فهما **ليسا** مبالغ العملة المحجّمة بـ 8 منازل عشرية، ولا يمسّان معروض العملة أبداً. أما `decimals` الخاص بالرمز فهو بيانات وصفية للعرض فقط. الثابت المفروض على مساري التنفيذ الحيّ والمزامنة: `totalSupply ≤ cap` ≥ `0`، ولا يمكن لـ `SETCAP` خفض السقف دون المعروض المسكوك بالفعل.

الرسوم (تُدفَع بالعملة الأصيلة): تدفع `CREATE` تحديداً `5` (`fee-create-token-tx`)؛ وتدفع `MINT` و`BURN` و`SETCAP` و(`MINIMUM_FEE`) `0.01` ≥ `TRANSFER`.

7.2 حالة الرموز — `api/v1/tokens/`

قراءة تعريفات الرموز ومعروضها وحائزيها والأرصدة حسب العنوان. هذه للقراءة فقط؛ ولا يتغيّر معروض الرمز وبياناته الوصفية إلا عبر عمليات `TokenTx` في §7.1.

المنهج	المسار	المُعامِلات	يُعيد
GET	`api/v1/tokens/`	ترقيم الصفحات	تعريفات رموز مُرقَّمة صفحات (كل الرموز)
GET	`api/v1/tokens/{tokenId}/`	path `tokenId`	تعريف رمز واحد + المعروض الحالي
GET	`api/v1/tokens/{tokenId}/holders/`	path `tokenId` + ترقيم الصفحات	أرصدة الحائزين لهذا الرمز، مُرقَّمة صفحات
GET	`api/v1/tokens/balances/{address}/`	path `address` + ترقيم الصفحات	أرصدة الرموز التي يملكها عنوان، مُرقَّمة صفحات

كائن **الرمز** (token):

الحقل	النوع	المعنى
`tokenId`	`String`	تجزئة معاملة CREATE (معرّف الرمز الفريد)
`symbol`	`String`	الرمز المختصر (`1–12` محرفاً؛ غير فريد)
`name`	`String`	اسم الرمز (`1–40` محرفاً)
`decimals`	`Int`	منازل العرض العشرية (`0–18`)
`cap`	`String`	المعروض الأقصى، بوحدات أساسية صحيحة
`totalSupply`	`String`	المعروض المسكوك الحالي، بوحدات أساسية صحيحة
`issuer`	`String`	عنوان المُنشِئ (صاحب صلاحية MINT/BURN/SETCAP الوحيد)

كائن **الرصيد** (balance):

الحقل	النوع	المعنى
`tokenId`	`String`	معرّف الرمز
`holder`	`String`	عنوان الحائز
`balance`	`String`	المبلغ المملوك، بوحدات أساسية صحيحة

---

8. معاملات المكافأة — `api/v1/transactions/rewards/` (للقراءة فقط)

مكافآت كتل مُولَّدة من النظام (البنك ← المُصادِق المنتج، 100% إلى محفظة المنتج). هذه **غير** قابلة للإرسال من المستخدم؛ ولا يوجد `POST`.

المنهج	المسار	الاستعلام / الجسم	يُعيد
GET	`api/v1/transactions/rewards/`	ترقيم الصفحات	معاملات مكافأة مُرقَّمة صفحات
GET	`api/v1/transactions/rewards/{hash}/`	path `hash`	معاملة مكافأة واحدة

تحمل RewardTx رسوماً قيمتها `0` ورقماً تسلسلياً `0` وحمولة نظامية؛ ويُتحقَّق من قيمتها مقابل المكافأة المجدولة المعيارية على كل عقدة.

---

9. الكتل — `api/v1/blocks/main/` و `api/v1/blocks/genesis/`

يتشارك موردا الكتل شكلاً متطابقاً. `main` هي كتل معاملات المستخدمين؛ و`genesis` هي كتل حدود الحقبة (epoch) التي تُجمّد مجموعة المُصادِقين النشطة.

9.1 الكتل الرئيسية — `api/v1/blocks/main/`

المنهج	المسار	المُعامِلات	يُعيد
GET	`api/v1/blocks/main/{hash}/`	path `hash`	كتلة رئيسية بحسب التجزئة
GET	`api/v1/blocks/main/{hash}/previous/`	path `hash`	الكتلة السابقة
GET	`api/v1/blocks/main/{hash}/next/`	path `hash`	الكتلة التالية
GET	`api/v1/blocks/main/`	ترقيم الصفحات	كتل رئيسية مُرقَّمة صفحات

9.2 الكتل التكوينية — `api/v1/blocks/genesis/`

المنهج	المسار	المُعامِلات	يُعيد
GET	`api/v1/blocks/genesis/{hash}/`	path `hash`	كتلة تكوينية بحسب التجزئة
GET	`api/v1/blocks/genesis/{hash}/previous/`	path `hash`	الكتلة التكوينية السابقة
GET	`api/v1/blocks/genesis/{hash}/next/`	path `hash`	الكتلة التكوينية التالية
GET	`api/v1/blocks/genesis/`	ترقيم الصفحات	كتل تكوينية مُرقَّمة صفحات

تشمل حقول الكتلة المشتركة `timestamp` و`height` و`hash` و`previousHash` و`signature` و`publicKey`، بالإضافة إلى حمولة خاصة بكل نوع (تحمل الكتل الرئيسية جذور Merkle للمعاملات/الحالة/الإيصالات و`epochIndex`؛ وتحمل الكتل التكوينية `epochIndex` و`activeAnchorCount` ومفاتيح المُصادِقين النشطين المجمّدة). راجع الورقة الصفراء للاطّلاع على المخططات الدقيقة.

---

10. المُصادِقون (الحالة) — `api/v1/validators/`

قراءة مجموعة المُصادِقين وحالة الانتخاب. (لا يُخلَط بينها وبين مورد *معاملة تسجيل المُصادِق* في §5.)

المنهج	المسار	المُعامِلات	يُعيد
GET	`api/v1/validators/`	ترقيم الصفحات (حسب ترتيب التسجيل)	كل المُصادِقين، مرتّبين حسب التسجيل (`createDate`، ثم المعرّف)
GET	`api/v1/validators/active/`	ترقيم الصفحات	المجموعة النشطة الحالية
GET	`api/v1/validators/view/`	ترقيم الصفحات	سجلات عرض المُصادِقين

يجري الاختيار حسب **ترتيب التسجيل**، لا حسب الأصوات أو التقييم — يُرتَّب المُصادِقون حسب `createDate` (الأقدم تسجيلاً أولاً). المجموعة النشطة هي المُصادِقون المرتكزون (anchor) الذين ما زالوا يتمتّعون بالامتياز بالإضافة إلى المُصادِقين المنتخبين الأقدم تسجيلاً، حتى اكتمال `promoters-count = 21`، ناقص العقد غير المتصلة/المحظورة/المُغادِرة.

---

11. المستكشف ومعلومات العقدة

11.1 المستكشف — `api/v1/explorer/info/`

المنهج	المسار	يُعيد
GET	`api/v1/explorer/info/`	كائن ملخّص للشبكة/السلسلة

حقول الحمولة:

الحقل	النوع	المعنى
`nodesCount`	`Int`	عدد العقد المعروفة
`blocksCount`	`Long`	إجمالي الكتل
`transactionsCount`	`Long`	إجمالي المعاملات
`blockProductionTime`	`Number`	متوسط زمن إنتاج الكتلة
`transactionsPerSecond`	`Number`	الإنتاجية (TPS)
`epochNumber`	`Long`	فهرس الحقبة (epoch) الحالية
`epochDate`	`Long`	وقت بدء الحقبة الحالية
`validatorsCount`	`Int`	عدد المُصادِقين
`maxSupply`	`BigDecimal`	المعروض الأقصى الثابت (`100,000,000`)
`circulatingSupply`	`BigDecimal`	`maxSupply − bankBalance`

مثال:

```
curl http://localhost:7019/api/v1/explorer/info
```

```
{
  "timestamp": 1719700000000,
  "version": "1.1.0",
  "payload": {
    "nodesCount": 3,
    "blocksCount": 1280,
    "transactionsCount": 4096,
    "blockProductionTime": 15000,
    "transactionsPerSecond": 0.0137,
    "epochNumber": 426,
    "epochDate": 1719600000000,
    "validatorsCount": 5,
    "maxSupply": "100000000.00000000",
    "circulatingSupply": "50000000.00000000"
  }
}
```

11.2 معلومات العقدة — `api/v1/info/`

المنهج	المسار	يُعيد
GET	`api/v1/info/`	`{ nodeKey, host, port, networkId, genesisHash, chainSpecDigest, version, apiVersion, protocolVersion, syncStatus, faucet }`
GET	`api/v1/info/version/`	`{ version, apiVersion, protocolVersion }` — بالمعاني نفسها في `info/` : إصدار البناء، وإصدار الواجهة البرمجية، وإصدار بروتوكول الشبكة (حتى العقدة 0.1.10 كان `protocolVersion` هنا يحمل إصدار الواجهة البرمجية، F-057)
GET	`api/v1/info/uptime/`	مدة تشغيل العقدة (مللي ثانية)
GET	`api/v1/info/genesis/`	ملف `genesis.json` العام للسلسلة — المراسي، وأرصدة الكتلة التكوينية النهائية، والمفتاح العام للمُوقِّع، وبصمات وتواقيع كتلتي الإقلاع المُثبَّتة؛ تُقلع منه العقدة المُنزَّلة (`--janzeer.genesis.file`)

حقول حمولة `api/v1/info/` :

الحقل	النوع	المعنى
`nodeKey`	`String`	المفتاح العام لهذه العقدة (هويتها في P2P / مُعرّفها UID؛ **مفتاح المُصادِق** الذي تُسجِّله)
`host`	`String`	المضيف الخارجي المُعلَن
`port`	`Int`	منفذ P2P الخارجي المُعلَن
`networkId`	`String`	`janzeer` — `consensus.network-id` على الشبكة الرئيسية (تستخدم شبكة الاختبار الخاصة `janzeer-testnet`)؛ وقِّع كل معاملة به
`genesisHash`	`String`	بصمة كتلة الحقبة 1 التكوينية: هوية السلسلة
`chainSpecDigest`	`String`	ملخّص قواعد الإجماع + الكتلة التكوينية المُتبادَل في مصافحة P2P
`version` / `apiVersion` / `protocolVersion`	`String`	إصدارات برمجية العقدة / واجهة REST-RPC / بروتوكول اتصال P2P
`syncStatus`	`String`	`SYNCHRONIZED` أو `PROCESSING` أو `NOT_SYNCHRONIZED`
`faucet`	`Boolean`	تُشغِّل هذه العقدة صنبورًا (§11.3، `POST /api/v1/faucet`) — ممكن فقط على شبكة اختبار خاصة؛ دائمًا `false` على الشبكة الرئيسية

11.3 الصنبور — `api/v1/faucet/` (شبكات الاختبار الخاصة فقط)

موجود فقط على شبكة اختبار خاصة تشغّلها بنفسك — لا توجد شبكة اختبار عامة ولا صنبور عام (ترفض العقدة تشغيل صنبور على معرِّف الشبكة الرئيسية؛ ويخبرك `faucet` → `GET /api/v1/info` بوجوده).

المنهج	المسار	يُعيد
GET	`api/v1/faucet/`	`{ enabled, address, amount, cooldownHours }`
POST	`api/v1/faucet/` بجسم `{ address }`	`201` `{ hash, address, amount }` — يوقِّع الصنبور تحويلًا عاديًا ويُرسله؛ `400` عنوان غير صالح؛ `404` لا صنبور على هذه العقدة؛ `429` + `Retry-After` (`type: FAUCET_COOLDOWN`) عندما يكون العنوان **أو** عنوان IP الخاص بالعميل قد تلقّى بالفعل دفعةً داخل نافذة التهدئة

---

12. أجسام طلبات المعاملات (POST)

تقبل جميع نقاط نهاية `POST` للمعاملات (§4–§7) نفس **الحقول المشتركة**، بالإضافة إلى مجموعة صغيرة من **الحقول الخاصة بالنوع**. يبني العميل المعاملة ويوقّعها؛ وتتحقّق العقدة منها فقط (غير حافظة — لا توجد نقاط نهاية للمفاتيح أو التوقيع). للاطّلاع على مخطط بايتات الصورة الأولية الدقيق وخوارزمية التوقيع، راجع الورقة الصفراء ودليل المطوّر.

12.1 الحقول المشتركة (لكل أنواع المعاملات)

الحقل	النوع	المعنى
`timestamp`	`Long`	وقت بناء العميل (ميلي ثانية منذ الحقبة)، جزء من الصورة الأولية الموقَّعة
`fee`	`BigDecimal`	رسوم المعاملة (قيمة دنيا/محدّدة لكل نوع — انظر أدناه)
`nonce`	`Long`	الرقم التسلسلي لحساب المُرسِل (الحماية من إعادة التشغيل)
`hash`	`String`	`doubleSHA256(preimage)`، سداسي عشري صغير
`senderAddress`	`String`	`0x` + 40 محرفاً سداسياً عشرياً صغيراً
`senderSignature`	`String`	ECDSA secp256k1، RFC-6979، low-S، DER ثم Base64، على ملخّص الـ 32 بايت الخام
`senderPublicKey`	`String`	مفتاح عام مضغوط بطول 33 بايت، سداسي عشري

12.2 الحقول الخاصة بالنوع وقواعد الرسوم/المبلغ

نوع المعاملة	نقطة النهاية	حقول إضافية	الرسوم	المبلغ / الوديعة
Transfer	`POST /api/v1/transactions/transfers`	`amount`, `recipientAddress?`, `?data`	≤ `0.01`	`amount` ≥ `0.1` ما لم تُرفَق مذكّرة
Promoter	`POST /api/v1/transactions/validators`	`amount`, `validatorKey`	`3`	وديعة تحديداً `2000` ← البنك (غير قابلة للاسترداد)
ExitPromoter	`POST /api/v1/transactions/exit-validators`	`validatorKey`	≤ `0.01`	—
Token	`POST /api/v1/transactions/tokens`	`op`, `tokenId?`, `symbol?`, `name?`, `decimals?`, `cap?`, `amount?`, `?recipient`	CREATE تحديداً `5`، البقية ≤ `0.01`	`cap` / `amount` للرمز **سلاسل** عشرية بوحدات أساسية صحيحة (وليست العملة)، والرسوم تُدفَع بالعملة

الثوابت: `MAX_DATA_SIZE = 256`، `MINIMUM_AMOUNT = 0.1`، `MINIMUM_FEE = 0.01` بايت، `fee-create-token-tx = 5`. إن `cap` و `amount` للرموز الأصيلة وحدات أساسية صحيحة تُرسَل كسلاسل **strings** في JSON (§7). إن RewardTx (§8) مُولَّدة من النظام ولا تُرسَل أبداً عبر POST.

تُعيد جميع عمليات POST الناجحة للمعاملات `Created 201` مع المعاملة المقبولة في `payload`.

12.3 مثال — إرسال تحويل

```
curl -X POST http://localhost:7019/api/v1/transactions/transfers \
  -H 'Content-Type: application/json' \
  -d '{
    "timestamp": 1719700000000,
    "fee": "0.01",
    "nonce": 0,
    "hash": "a1b2c3...",
    "senderAddress": "0x06e1c0fa9955a700876f8cb0acc7f13fba9fb8ba",
    "senderSignature": "MEUCIQD...base64...",
    "senderPublicKey": "02a1b2c3...compressed-hex...",
    "amount": "1.5",
    "recipientAddress": "0x1111111111111111111111111111111111111111",
    "data": "",
    "nonce": 4
  }'
```

```
{
  "timestamp": 1719700000001,
  "version": "1.1.0",
  "payload": {
    "hash": "a1b2c3...",
    "senderAddress": "0x06e1c0fa9955a700876f8cb0acc7f13fba9fb8ba",
    "recipientAddress": "0x1111111111111111111111111111111111111111",
    "amount": "1.5",
    "fee": "0.01",
    "nonce": 0
  }
}
```

يُعيد فشل التحقّق (مثل توقيع خاطئ أو رصيد غير كافٍ أو رسوم دون الحدّ الأدنى) `400` مع شكل الخطأ الموضّح في §1.3.

---

13. `/actuator` — Actuator (غير مُغلَّف)

نقاط نهاية Spring Boot Actuator. هذه **لا** تُلَفّ في غلاف الاستجابة (§1.2).

المنهج	المسار	الغرض
GET	`actuator/health/`	حالة الحيوية / الصحة
GET	`actuator/info/`	معلومات البناء/التطبيق
GET	`actuator/metrics/`	فهرس المقاييس والمقاييس الفردية
GET	`actuator/prometheus/`	نقطة نهاية كشط Prometheus

---

14. وثائق الواجهة البرمجية — `docs/` (غير مُغلَّف)

وثائق واجهة برمجية تفاعلية وقابلة للقراءة آلياً. هذه المسارات **غير** مُغلَّفة.

المنهج	المسار	الغرض
GET	`docs/swagger-ui.html/`	واجهة Swagger UI التفاعلية
GET	`docs/api-docs/`	وثيقة OpenAPI (JSON) تُقدَّم حيّةً من العقدة

تُقدَّم **مواصفة OpenAPI قابلة للقراءة آلياً حيّةً عند** `docs/api-docs/`. وقد تُضمَّن نسخة ثابتة أيضاً في المستودع عند `docs/api/openapi.json` لأدوات العمل دون اتصال وتوليد SDK؛ وعند وجود الاثنتين، تكون وثيقة `docs/api-docs/` الحيّة هي المرجعية المعتمَدة لإصدار العقدة العامل.

---

15. مرجع سريع — كل نقاط النهاية

المنهج	المسار	المورد
GET	`api/v1/wallets/{address}/`	الرصيد
GET	`api/v1/wallets/{address}/nonce/`	الرقم التسلسلي التالي
POST	`api/v1/wallets/validateAddress/`	التحقّق من العنوان
GET	`api/v1/transactions/transfers/`	قائمة التحويلات
GET	`api/v1/transactions/transfers/{hash}/`	تحويل واحد
POST	`api/v1/transactions/transfers/`	إرسال تحويل
GET	`api/v1/transactions/validators/`	قائمة معاملات المُصادِقين
GET	`api/v1/transactions/validators/{hash}/`	معاملة مُصادِق واحدة
POST	`api/v1/transactions/validators/`	تسجيل مُصادِق
GET	`api/v1/transactions/exit-validators/`	قائمة معاملات خروج المُصادِقين
GET	`api/v1/transactions/exit-validators/{hash}/`	معاملة خروج مُصادِق واحدة
POST	`api/v1/transactions/exit-validators/`	اعتزال مُصادِق
GET	`api/v1/transactions/tokens/`	قائمة معاملات الرموز
GET	`api/v1/transactions/tokens/{hash}/`	معاملة رمز واحدة
POST	`api/v1/transactions/tokens/`	إرسال عملية رمز
GET	`api/v1/tokens/`	قائمة تعريفات الرموز
GET	`api/v1/tokens/{tokenId}/`	رمز واحد (التعريف + المعروض)
GET	`api/v1/tokens/{tokenId}/holders/`	حائزو الرمز
GET	`api/v1/tokens/balances/{address}/`	أرصدة الرموز لعنوان
GET	`api/v1/transactions/rewards/`	قائمة معاملات المكافأة
GET	`api/v1/transactions/rewards/{hash}/`	معاملة مكافأة واحدة
GET	`api/v1/blocks/main/`	قائمة الكتل الرئيسية
GET	`api/v1/blocks/main/{hash}/`	كتلة رئيسية واحدة
GET	`api/v1/blocks/main/{hash}/previous/`	الكتلة الرئيسية السابقة
GET	`api/v1/blocks/main/{hash}/next/`	الكتلة الرئيسية التالية
GET	`api/v1/blocks/genesis/`	قائمة الكتل التكوينية

المنهج	المسار	المورد
GET	`api/v1/blocks/genesis/{hash}/`	كتلة تكوينية واحدة
GET	`api/v1/blocks/genesis/{hash}/previous/`	الكتلة التكوينية السابقة
GET	`api/v1/blocks/genesis/{hash}/next/`	الكتلة التكوينية التالية
GET	`api/v1/validators/`	المُصادِقون حسب ترتيب التسجيل
GET	`api/v1/validators/active/`	المجموعة النشطة
GET	`api/v1/validators/view/`	سجلات عرض المُصادِقين
GET	`api/v1/explorer/info/`	ملخّص الشبكة
GET	`api/v1/info/`	هوية العقدة
GET	`api/v1/info/version/`	إصدارات العقدة والواجهة البرمجية وبروتوكول الشبكة
GET	`api/v1/info/uptime/`	مدة تشغيل العقدة
GET	`actuator/health/`	الصحة (غير مُغلَّف)
GET	`actuator/info/`	معلومات البناء (غير مُغلَّف)
GET	`actuator/metrics/`	المقاييس (غير مُغلَّف)
GET	`actuator/prometheus/`	Prometheus (غير مُغلَّف)
GET	`docs/swagger-ui.html/`	Swagger UI (غير مُغلَّف)
GET	`docs/api-docs/`	وثيقة OpenAPI (غير مُغلَّف)

حالة الوثيقة: الإصدار 1.1. تعكس إصدار العقدة 0.1.0. الحقائق الدقيقة على مستوى البايت (مخططات الصور الأولية، التجزئة، تحجيم المبالغ، ثوابت الاشتقاق) موجودة في الورقة الصفراء.

15. JSON-RPC 2.0 — `POST /rpc` و `ws://…/rpc/ws`

الخدمات وقواعد الاستقبال نفسها التي تخدمها نقاط REST أعلاه، معروضة بصيغة JSON-RPC 2.0 لتكاملات منصّات التداول وجهات الحفظ. غلاف ورموز أخطاء قياسية، معاملات بالاسم أو بالموضع، دفعات، و— عبر WebSocket — `janzeer_subscribe` لـ `newBlocks` (مع إعادة بثّ من ارتفاع معيّن) و `addressActivity` (كل معاملة نهائية تمسّ عنوانًا مراقَبًا). غلاف REST في §1.2 لا يُطبَّق؛ يعيد `GET /rpc` قائمة الطرق. جدول الطرق ورموز الأخطاء ومثال عملي: دليل المطوّر §2.14.

الطرق: `janzeer_getStats`، `janzeer_getChainSpec`، `janzeer_getInfo`، `janzeer_methods`، `janzeer_getTokenBalances`، `janzeer_getNonce`، `janzeer_getBalance`، `janzeer_getAccount`، `janzeer_sendToken`، `janzeer_sendTransfer`، `janzeer_getTransactionsByAddress`، `janzeer_sendRawTransaction`، `janzeer_sendExitValidator`، `janzeer_sendRegisterValidator`، `janzeer_estimateFee`، `janzeer_waitForFinality`، `janzeer_getTransactionByHash`، `janzeer_getBlocks`، `janzeer_getBlockByHash`، `janzeer_getBlockByNumber`، `janzeer_getTip`، `janzeer_listValidators`، `janzeer_getTokenHolders`، `janzeer_listTokens`، `janzeer_getToken`،

`janzeer_getEpoch`، `janzeer_getValidator`، `janzeer_getActiveValidators`؛ عبر WebSocket فقط:
`janzeer_unsubscribe`، `janzeer_subscribe`.