2026

ieUnit

API-Dokumentation

Öffentliche JSON-API, Tagesbelege, Einbettung und MCP

Autor: Schoepi
Version: 1.2
Datum: 14.10.2026

ieWerk – ieCommunity – ieEdge – Infinity-Economics

ieWerk

Alle Rechte Vorbehalten

Inhaltsverzeichnis

Überblick 1

Endpunkte 1

Aktueller Wert – value 1

Verlauf – history 2

Korb-Bestandteile – components 2

Korb-Definition – basket 3

Umrechnungskurse – rates 3

Tagesbeleg – receipt 3

Belege-Kette – receipts 4

Feld-Referenz 4

Code-Beispiele 5

curl 5

JavaScript (Browser, fetch) 5

Python 5

Einbettbares Widget 5

Variante 1 – fester Wert 6

Variante 2 – mit Einheiten-Auswahl 6

iframe-Alternative 6

Tagesbelege prüfen 6

Mit einem KI-Assistenten 7

Mit Standardwerkzeugen 7

MCP-Server für KI-Assistenten 7

Werkzeuge 8
Einrichten 8

Nutzungsbedingungen 9

Versionierung 10

Überblick

Die ieUnit-API liefert den aktuellen Referenzwert, den vollständigen Verlauf seit 2017, die Korb-Definition, die Tagespreise der Korb-Bestandteile und die digital unterschriebenen Tagesbelege als offene JSON-Endpunkte. Sie ist **kostenlos**, **ohne Schlüssel** nutzbar und für die Einbindung durch Dritte gedacht (CORS offen).

Es handelt sich um eine reine **Lese-API** über HTTP GET. Alle Antworten sind UTF-8-kodiertes JSON. Sämtliche Werte sind regelbasierte Referenzwerte und kein Zahlungsmittel, kein Anlageprodukt und keine Finanzberatung.

Eigenschaft	Wert
Basis-URL	`https://ieunit.org/api/v1/`
Protokoll	HTTPS, Methode GET (zusätzlich OPTIONS für CORS-Preflight)
Format	JSON (application/json; charset=utf-8)
Authentifizierung	keine – frei zugänglich
CORS	Access-Control-Allow-Origin: * (Einbindung von jeder Domain)
Caching	Cache-Control: public, max-age=900 (15 Minuten)
Aktualisierung	einmal täglich (Cron); häufigeres Abrufen liefert denselben Wert
Version	v1 (in der URL); kompatibel gehaltene Erweiterungen möglich
Lizenz	frei nutzbar mit Quellenangabe (CC BY 4.0), siehe Nutzungsbedingungen

> **Fair-Use:** Die Daten ändern sich nur einmal pro Tag. Bitte höchstens stündlich abfragen oder die Antwort clientseitig zwischenspeichern. Wer die Last gering hält, hilft, den Dienst dauerhaft gratis zu halten.

Endpunkte

Aktueller Wert – value

`GET https://ieunit.org/api/v1/value.php`

Liefert den aktuellen ieUnit-Referenzwert in USD, CHF und EUR.

Parameter	Typ	Beschreibung
currency	optional	USD | CHF | EUR – legt das Feld `value`/`currency` fest (Standard USD). Das Feld `values` enthält ohnehin alle drei.

Beispiel-Antwort:


```
{
  "unit": "ieUnit",
  "date": "2026-06-12",
  "value": 3.7892,
  "currency": "USD",
  "values": {
    "USD": 3.7892,
    "CHF": 3.0285,
    "EUR": 3.2844
  },
  "change_24h_pct": -0.27,
  "method": "geometric_mean",
  "base_date": "2017-01-10",
  "source": "live",
  "updated_at": "2026-06-12T10:35:29Z",
  "disclaimer": "Referenzwert, kein Zahlungsmittel und keine Anlageberatung."
}
```


Das Feld `source` ist `live`, sobald der tägliche Cron gelaufen ist, sonst `backfill` (Startwert aus der Datenbasis). `updated_at` ist der UTC-Zeitstempel des letzten Laufs.

Verlauf – history

`GET https://ieunit.org/api/v1/history.php`

Liefert die ieUnit-Zeitreihe (USD) seit dem Basiszeitpunkt 10.01.2017 als Liste von `[Datum, Wert]`-Paaren.

Parameter	Typ	Beschreibung
from	optional	Startdatum YYYY-MM-DD (einschliesslich)
to	optional	Enddatum YYYY-MM-DD (einschliesslich)
step	optional	day (Standard) | week | month – zeitliche Verdichtung

Beispiel-Antwort (`?step=month&from=2017-01-10`):


```
{
  "unit": "USD",
  "base_date": "2017-01-10",
  "updated": "2026-06-12",
  "step": "month",
  "count": 114,
  "series": [
    ["2017-01-10", 1.0000],
    ["2017-02-28", 1.0431],
    ["2017-03-31", 1.0588],
    ["2026-06-12", 3.7892]
  ]
}
```


Korb-Bestandteile – components

`GET https://ieunit.org/api/v1/components.php`

Liefert die Tagespreise der neun Korb-Werte in USD je Einheit. Gold und Silber sind je Feinunze (31,1034768 g) quotiert.

Parameter	Typ	Beschreibung
from / to	optional	Zeitraum YYYY-MM-DD
step	optional	day (Standard) | week | month
only	optional	kommagetrennte Auswahl, z. B. `only=BTC,Gold` (spart Datenvolumen)

Antwort-Struktur: `dates` (Liste der Tage) und `series` (Objekt je Wert mit gleichlanger Preisliste), zusätzlich `ounce_g` für die Metall-Umrechnung.

```
{
  "unit": "USD_per_unit",
  "ounce_g": 31.1034768,
  "step": "month", "count": 114,
  "dates": ["2017-01-10", "..."],
  "series": {
    "USD": [1.0, "..."], "EUR": [1.0589, "..."],
    "BTC": [907.68, "..."], "Gold": [1184.20, "..."]
  }
}
```

Korb-Definition – basket

`GET https://ieunit.org/api/v1/basket.php`

Liefert die unveränderliche Korb-Definition samt der fixen Referenzpreise am Basiszeitpunkt. Damit lässt sich der ieUnit jederzeit unabhängig nachrechnen.

```
{
  "base_date": "2017-01-10",
  "start_value": 1.0,
  "internal_unit": "USD",
  "method": "geometric_mean",
  "components": ["USD","EUR","JPY","CHF","CNH","BTC","ETH","Gold","Silber"],
  "reference_prices_usd": {
    "USD": 1.0, "EUR": 1.05886209, "JPY": 0.0086218,
    "CHF": 0.9861544, "CNH": 0.14416077, "BTC": 907.67901611,
    "ETH": 10.63, "Gold": 1184.19995117, "Silber": 16.8029995
  }
}
```

Umrechnungskurse – rates

`GET https://ieunit.org/api/v1/rates.php`

Liefert aktuelle Umrechnungskurse (USD je Einheit) für den ieUnit-Rechner: rund 30 Fiat-Währungen, die Top-30-Kryptowährungen, Gold/Silber je Gramm sowie XIN (Sonderfall: letzter Börsenkurs 1 Satoshi, Delisting 22.02.2022). Dazu der aktuelle ieUnit-Wert in USD.

Tagesbeleg – receipt

`GET https://ieunit.org/api/v1/receipt.php`

Liefert den digital unterschriebenen Beleg eines täglichen Laufs: Einzelpreis jeder Quelle, die angewandte Regel (eine Quelle, Median mehrerer Quellen oder – falls alle ausfielen – der letzte gültige Preis), die Referenzpreise und das Ergebnis. Ohne Parameter kommt der neueste Beleg. Belege gibt es seit Oktober 2026.

Parameter	Typ	Beschreibung
date	optional	Beleg eines Tages, YYYY-MM-DD. Lief der Cron an einem Tag zweimal, gilt der letzte Lauf.
id	optional	genau ein Beleg, z. B. `2026-10-14` oder `2026-10-14-2` (zweiter Lauf des Tages)
part	optional	`receipt` (unterschriebener Text, byte-genau) · `signature` (base64) · `key` (öffentlicher Schlüssel als PEM) – für die Prüfung mit Standardwerkzeugen

Beispiel-Antwort (gekürzt):

```
{
  "format": "ieunit-receipt/1",
  "id": "2026-10-14",
  "receipt": "{\"format\":\"ieunit-receipt/1\",\"id\":\"2026-10-14\",
    \"seq\":1, … ,\"prev_fingerprint\":null, … ,\"components\":{ … ,
    \"BTC\":{\"reference\":907.67901611,\"price\":83321.3475,
    \"rule\":\"median\",\"sources\":{\"coinbase\":{\"value\":83305.895},
    \"kraken\":{\"value\":83336.8}}}, … },\"result\":{\"ieunit_usd\":3.99594, … }}",
  "fingerprint": "2b9f9d3cb0d08824…",
  "signature": "…(base64)…",
  "algorithm": "Ed25519",
  "key_id": "1efb1cbb4a2d58b1"
}
```

Unterschrieben ist das Feld `receipt` – ein JSON-Text, der genau so, Zeichen für Zeichen, geprüft wird. `fingerprint` ist der SHA-256 (hex) über seine UTF-8-Bytes, `signature` die Ed25519-Unterschrift über dieselben Bytes. Im Beleg selbst steht unter `prev_fingerprint` der Fingerabdruck des Vorgängers – so entsteht eine lückenlose Kette.

Belege-Kette – receipts

`GET https://ieunit.org/api/v1/receipts.php`

Liefert das Verzeichnis aller Belege in Reihenfolge (`chain`: Kennung, laufende Nummer, Datum, Zeitpunkt, Fingerabdruck, Schlüssel) samt den öffentlichen Schlüsseln (`keys`: Kennung, base64, PEM, gültig ab/bis). Nach einem Schlüsselwechsel bleibt der alte Schlüssel in der Liste, damit ältere Belege prüfbar bleiben.

Feld-Referenz

Feld	Bedeutung
unit	Bezeichnung der Einheit bzw. der Recheneinheit
value / values	ieUnit-Wert in der gewählten bzw. in allen Anzeige-Währungen
method	Berechnungsmethode – stets geometric_mean
base_date	Basiszeitpunkt 2017-01-10 (Startwert 1.00)
change_24h_pct	Veränderung gegenüber dem Vortag in Prozent
source	live (Cron gelaufen) oder backfill (Startwert)
updated_at	Zeitpunkt des letzten Datenlaufs (UTC, ISO 8601)
series	Zeitreihe als [Datum, Wert] bzw. Wert-Listen je Korb-Bestandteil
receipt	Tagesbeleg: der unterschriebene JSON-Text, byte-genau
fingerprint	SHA-256 (hex) über die UTF-8-Bytes von receipt
signature / key_id	Ed25519-Unterschrift (base64) und Kennung des Schlüssels
prev_fingerprint	Fingerabdruck des vorherigen Belegs (null beim ersten) – bildet die Kette
rule	Herkunft eines Korb-Preises: definition, single, median oder fallback

Code-Beispiele

curl

```
curl -s https://ieunit.org/api/v1/value.php?currency=CHF
```

JavaScript (Browser, fetch)

```
const r = await fetch('https://ieunit.org/api/v1/value.php?currency=CHF');
const data = await r.json();
console.log(`1 ieUnit = ${data.value} ${data.currency}`);
```

Python

```
import urllib.request, json
url = 'https://ieunit.org/api/v1/history.php?step=month'
with urllib.request.urlopen(url) as r:
    hist = json.load(r)
print(hist['count'], 'Datenpunkte seit', hist['base_date'])
```

Einbettbares Widget

Wer den aktuellen ieUnit-Wert auf der eigenen Webseite zeigen will, bindet das fertige Widget mit einer Zeile ein. Es holt den Wert live von der API, ist via Shadow-DOM gekapselt (kein Eingriff ins fremde CSS) und braucht keine Konfiguration. Es gibt zwei Varianten.

Variante 1 – fester Wert

Zeigt den Wert in einer festen Anzeige-Währung (CHF, USD oder EUR):

```
<div class="ieunit-widget" data-currency="CHF" data-chart="true"></div>
<script src="https://ieunit.org/assets/js/ieunit-widget.js" async></script>
```

Variante 2 – mit Einheiten-Auswahl

Blendet oben rechts eine dezente Auswahl ein. Der Betrachter waehlt selbst: Währungen (CHF, USD, EUR, JPY, CNY) sowie Gramm Gold, Gramm Silber, Bitcoin und Ethereum. Die Starteinheit richtet sich automatisch nach der Browsersprache und wird gemerkt.

```
<div class="ieunit-widget" data-picker="true" data-chart="true"></div>
<script src="https://ieunit.org/assets/js/ieunit-widget.js" async></script>
```

Attribut	Werte	Beschreibung
data-picker	true | false	Einheiten-Auswahl einblenden (Variante 2, Standard false)
data-currency	CHF | USD | EUR	feste Währung (Variante 1) bzw. Startwert
data-chart	true | false	Mini-Verlaufschart anzeigen (Standard false)
data-theme	light | dark	Farbschema (Standard light)
data-link	true | false	Auf ieunit.org verlinken (Standard true)

iframe-Alternative

Völlig isoliert, ohne JavaScript im eigenen Seitenkontext (Auswahl mit `picker=1`):

```
<iframe src="https://ieunit.org/embed/ieunit.html?picker=1&chart=1"
        width="320" height="200" style="border:0" loading="lazy"
        title="ieUnit Live-Wert"></iframe>
```

> Beide Varianten sind kostenlos und dürfen frei eingebunden werden. Eine Quellenangabe (Link auf ieunit.org) ist erwünscht und im Widget standardmässig enthalten.

Tagesbelege prüfen

Jeder tägliche Lauf hält in einem Beleg fest, welche Quelle welchen Preis geliefert hat und was daraus berechnet wurde. Wer den ieUnit für einen Vertrag, eine Rechnung oder als Gebühren-Massstab nutzt, lädt den Beleg des Stichtags herunter und kann ihn auch Jahre später noch prüfen. Geprüft werden vier Dinge:

- **Fingerabdruck:** Der SHA-256 über den Beleg-Text stimmt mit `fingerprint` überein – der Beleg ist unverändert.

- **Unterschrift:** Die Ed25519-Unterschrift passt zum öffentlichen Schlüssel von ieunit.org – der Beleg ist echt.
- **Verkettung:** Der Fingerabdruck steht so in der veröffentlichten Kette, und `prev_fingerprint` passt zum Vorgänger – die Kette ist lückenlos.
- **Nachrechnung:** Aus den Preisen im Beleg ergibt sich mit der ieUnit-Formel genau das veröffentlichte Ergebnis.

Am einfachsten geht das im Browser unter `https://ieunit.org/de/belege/` – die Prüfung läuft vollständig beim Besucher, der Server liefert nur die Daten. Dort lässt sich auch eine früher gespeicherte Beleg-Datei prüfen.

Mit einem KI-Assistenten

Jeder KI-Assistent, der Webseiten abrufen und Code ausführen kann – etwa Claude, ChatGPT, Gemini oder Codex –, erledigt die Prüfung mit diesem Auftrag:

```
Prüfe den neuesten ieUnit-Tagesbeleg unter https://ieunit.org/api/v1/receipt.php:
1. Berechne den SHA-256 über die exakten UTF-8-Bytes des Felds "receipt"
   und vergleiche ihn mit "fingerprint".
2. Prüfe die Ed25519-Unterschrift im Feld "signature" mit dem öffentlichen
   Schlüssel aus https://ieunit.org/api/v1/receipts.php.
3. Vergleiche "prev_fingerprint" mit dem Fingerabdruck des Vorgängers in der Kette.
4. Rechne den ieUnit aus den Preisen im Beleg nach und vergleiche mit "result".
```

Mit Standardwerkzeugen

Von Hand genügen curl und OpenSSL (ab Version 3.0):

```
curl -s "https://ieunit.org/api/v1/receipt.php?part=receipt" > beleg.txt
curl -s "https://ieunit.org/api/v1/receipt.php?part=signature" | openssl base64 -d -A >
unterschrift.bin
curl -s "https://ieunit.org/api/v1/receipt.php?part=key" > schluessel.pem
sha256sum beleg.txt
openssl pkeyutl -verify -pubin -inkey schluessel.pem -rawin -in beleg.txt -sigfile
unterschrift.bin
```

Der Beleg ist echt und unverändert, wenn der angezeigte Fingerabdruck mit `fingerprint` übereinstimmt und OpenSSL «Signature Verified Successfully» meldet. Für einen bestimmten Tag `&date=YYYY-MM-DD` an jede Adresse anhängen. Unter Windows heisst der Befehl `curl.exe`, den Fingerabdruck zeigt `certutil -hashfile beleg.txt SHA256`; auf dem Mac `shasum -a 256 beleg.txt`.

> **Was ein Beleg (noch) nicht beweist:** Er zeigt, was die Quellen laut ieunit.org gemeldet haben – ob eine Börse tatsächlich diesen Preis meldete, lässt sich nur an ihrem eigenen Kursverlauf prüfen. Theoretisch könnte der Betreiber die ganze Kette neu schreiben und neu unterschreiben; das fällt auf, sobald jemand einen älteren Beleg aufbewahrt hat. Geplant ist, den Fingerabdruck jedes Tages zusätzlich in der Bitcoin-Blockchain zu verankern (OpenTimestamps) – dann ist auch das ausgeschlossen.

MCP-Server für KI-Assistenten

Neben der JSON-API steht der ieUnit auch als **MCP-Server** bereit. MCP ist die Abkuerzung für *Model Context Protocol*, den offenen Standard, über den KI-

Assistenten externe Werkzeuge einbinden. Der Unterschied zur API: Eine Schnittstelle muss jemand kennen und einbauen, ein MCP-Server wird von der KI selbst benutzt, sobald er eingerichtet ist. Auf die Frage, was 250 Franken in ieUnit sind, holt der Assistent die Antwort direkt hier, statt sie zu schaetzen.

Merkmal	Wert
Server-URL	https://ieunit.org/mcp/
Transport	Streamable HTTP (POST, JSON-RPC 2.0)
Protokollversionen	2026-07-28 sowie 2025-11-25 und aelter
Authentifizierung	keine
Zugriff	ausschliesslich lesend

Werkzeuge

Werkzeug	Liefert
ieunit_value	Aktueller Wert in USD, CHF und EUR mit Tagesveraenderung
ieunit_history	Verlauf seit dem Basiszeitpunkt, täglich, woechentlich oder monatlich
ieunit_convert	Betraege umrechnen, rund 60 Einheiten, in beide Richtungen
ieunit_basket	Wertkorb mit Referenzpreisen und aktuellen Werten
ieunit_compare	Einen Korb-Wert gegen den ieUnit messen statt gegen eine Landeswaehrung
ieunit_methodology	Berechnungsregel, Datenquellen und Abgrenzung in Klartext

Das letzte Werkzeug hat einen besonderen Zweck: Es gibt der KI die verbindliche Definition mit, einschliesslich der Klarstellung, dass der ieUnit keine Währung, kein Stablecoin und ohne Peg ist. So beschreiben Assistenten den ieUnit so, wie er gemeint ist, statt ihn in eine falsche Schublade zu legen.

Einrichten

MCP ist herstellerunabhaengig – der Server lässt sich in jeder Anwendung eintragen, die das Protokoll unterstützt. Die gaengigen Kommandozeilen-Werkzeuge:

Anwendung	Befehl
Claude Code	claude mcp add --scope user --transport http ieunit https://ieunit.org/mcp/
Codex CLI	codex mcp add ieunit --url https://ieunit.org/mcp/

Anwendung	Befehl
Gemini CLI	gemini mcp add --scope user --transport http ieunit https://ieunit.org/mcp/

Der Zusatz `--scope user` macht den Eintrag projektuebergreifend – ohne ihn gilt er nur im aktuellen Ordner. Codex kennt diese Unterscheidung nicht und speichert grundsaetzlich global, dort entfaellt der Zusatz.

Anwendungen mit Oberflaeche – etwa Claude Desktop oder ChatGPT – haben die Einstellung unter *Connectors*, *Apps* oder *MCP-Server*. Dort einen neuen Eintrag anlegen, die Server-URL eintragen und *keine Authentifizierung* waehlen. Wer eine Konfigurationsdatei pflegt, nimmt diesen Block:

```
{
  "mcpServers": {
    "ieunit": {
      "type": "http",
      "url": "https://ieunit.org/mcp/"
    }
  }
}
```

Nach dem Eintragen einmal neu starten. Die meisten Anwendungen lesen ihre MCP-Server beim Programmstart ein – ein Eintrag, der waehrenddessen dazukommt, wird erst im naechsten Gespraech sichtbar. Das gilt für Claude Code, Codex und die Gemini CLI gleichermassen.

Zum Nachpruefen von Hand lässt sich der Werkzeugkatalog direkt abrufen:

```
curl -s https://ieunit.org/mcp/ \
  -H "Content-Type: application/json" \
  -H "MCP-Protocol-Version: 2026-07-28" \
  -H "Mcp-Method: tools/list" \
  -d '{"jsonrpc":"2.0","id":1,"method":"tools/list",
       "params":{"_meta":{"io.modelcontextprotocol/protocolVersion":"2026-07-28"}}}'
```

> Der MCP-Server nutzt dieselbe Datenbasis wie die API und rechnet nichts eigenes. Es gelten dieselben Nutzungsbedingungen.

Nutzungsbedingungen

- **Kostenlos und offen:** Die API und die Widgets dürfen frei genutzt, eingebunden und weiterverarbeitet werden, auch kommerziell.
- **Quellenangabe (CC BY 4.0):** Bitte ieUnit als Quelle nennen und auf `https://ieunit.org` verlinken.
- **Keine Gewähr:** Die Werte sind regelbasierte Referenzwerte. ieWerk übernimmt keine Haftung für Richtigkeit, Verfügbarkeit oder Eignung für einen bestimmten Zweck. ieUnit ist kein Zahlungsmittel und keine Anlageberatung.
- **Fair-Use:** höchstens stündliche Abfrage bzw. clientseitiges Zwischenspeichern; keine künstliche Dauerlast.

- **Stabilität:** Die Endpunkt-URLs unter `/api/v1/` bleiben stabil. Erweiterungen erfolgen abwärtskompatibel; grössere Änderungen würden unter `/api/v2/` veröffentlicht.

Versionierung

Diese Dokumentation beschreibt **API v1**. Neue Felder können ergänzt werden, ohne die Version zu wechseln; bestehende Felder behalten ihre Bedeutung. Inkompatible Änderungen würden als `/api/v2/` parallel angeboten.