

// AGENT-CHEAT-SHEET

Schema Manager für KI-Agenten

Stand: 22.09.2026 · Plugin-Version: 2.4.1 · Befehl: wp teil1-schema-manager (Alias:
wp t1-schema)

Dieses Blatt ist für KI-Agenten mit Shell- und WP-CLI-Zugriff. Es ersetzt keine redaktionelle Prüfung. Markup darf nur Aussagen beschreiben, die auf der Seite sichtbar sind.

Kopieren Sie zuerst den Block **Stehende Arbeitsanweisung**, danach den passenden Task-Prompt.

01

Stehende Arbeitsanweisung
einmal kopieren.

02

Danach genau den Prompt
für die aktuelle Aufgabe
kopieren.

03

Schreiboperationen erst
nach sichtbarer Freigabe
ausführen.

WP-CLI

wp teil1-schema-manager

Alias: wp t1-schema

ZUERST KOPIEREN

Stehende Arbeitsanweisung

Du arbeitest am WordPress-Plugin „Teil1 Schema Manager“ (Slug: `teil1-schema-manager`, Version 2.4.1). Das Plugin gibt ausschließlich JSON-LD im `<head>` aus. Es verändert keine öffentlichen Seiteninhalte, Titles, Descriptions oder Sitemaps. Angemeldete Administratoren sehen den Admin-Bar-Indikator. Aktivierte Konfliktunterdrückung kann überschneidendes JSON-LD anderer Anbieter entfernen.

Arbeitsregeln:

1. Zuerst Zielinstallation und Version prüfen:
``wp core is-installed`, `wp option get home`,
`wp plugin status teil1-schema-manager` und
`wp plugin get teil1-schema-manager --field=version`.
 Bei Multisite `--url`, bei abweichendem Installationspfad `--path` explizit angeben.`
2. Zuerst nur lesen. Keine Schreiboperation ohne ausdrückliche Freigabe.
3. Vor der ersten Änderung sichere Backups außerhalb des öffentlichen Webroots erstellen. Der Plugin-Export enthält keine Custom Variables, Einstellungen oder stabil wiederherstellbaren Datenbank-IDs und ist deshalb kein vollständiges Rollback-Backup.
4. Offizieller Befehl: ``wp teil1-schema-manager``. Alias ``wp t1-schema`` ist erlaubt.
5. ``--json-file`` wird nur von ``create`` und ``update`` unterstützt.
 Bei ``add-rule`` und ``set-local`` JSON sicher mit
``--schema-json="$(jq -c . datei.json)``` übergeben.
6. Export schreibt nur nach stdout. Datei per Shell-Umleitung erzeugen und danach mit ``jq empty`` validieren.
7. Markup muss zum sichtbaren Inhalt passen. Keine FAQPage, HowTo, VideoObject, Event, Review oder Product-Daten erfinden.
8. Doppeltes JSON-LD vermeiden. Yoast, Rank Math, SEOPress, AIOSEO und WooCommerce können dieselben Entitäten bereits ausgeben.
9. Der Schema Quality Score ist ein interner Diagnosewert, kein Ranking-, Rich-Results- oder KI-Zitierungswert.
10. Schreibbefehle sind nicht automatisch idempotent: ``create``, ``add-rule`` und ``set-local`` können bei Wiederholung Duplikate erzeugen. ``import`` fügt Globals und Regeln hinzu und ersetzt lokale Schemas pro betroffener Post-ID.
11. ``clear-local`` löscht alle lokalen Schemas einer Seite; ``set-local --replace`` ersetzt sie vollständig. ``--yes`` nur verwenden, wenn genau dieser destruktive Befehl freigegeben wurde.
12. Nach jeder genehmigten Änderung erneut prüfen: doctor, health, coverage und ``render <post_id>`` für betroffene URLs.
13. Vor jeder Schreiboperation zeigst du: Ziel, betroffenen Typ, Bedingungen, vollständiges JSON, exakten Befehl, Risiko und erwartete Auswirkung auf bestehende Daten. Dann wartest du.

WP-CLI-Referenz

AUFGABE	BEFEHL
Umgebung und Konflikte	<code>wp teil1-schema-manager doctor</code>
Globale Schemas	<code>wp teil1-schema-manager globals --format=json</code>
Regeln	<code>wp teil1-schema-manager rules --format=json</code>
Abdeckung	<code>wp teil1-schema-manager coverage</code>
Validierung	<code>wp teil1-schema-manager health</code>
Validierung einer Seite	<code>wp teil1-schema-manager health --post=42</code>
Ausgabe einer Seite	<code>wp teil1-schema-manager render 42</code>
Ausgabe roh, ohne Variablen	<code>wp teil1-schema-manager render 42 --raw</code>
Lokale Schemas einer Seite	<code>wp teil1-schema-manager local 42 --format=json</code>
Verfügbare Variablen	<code>wp teil1-schema-manager variables</code>
Custom Variables	<code>wp teil1-schema-manager vars --format=json</code>
Unterstützte Typen	<code>wp teil1-schema-manager types</code>
Schema-Export (kein Vollbackup)	<code>wp teil1-schema-manager export > backup.json</code>
Import prüfen	<code>wp teil1-schema-manager import backup.json --dry-run</code>
Import ausführen	<code>wp teil1-schema-manager import backup.json</code>

PROMPT KOPIEREN

Schreibbefehle

```
wp teil1-schema-manager create Organization --name="Firma GmbH" --url="https://example.com"
wp teil1-schema-manager update 1 --json-file=organization.json
wp teil1-schema-manager set-var phone "+43 1 234 5678"
wp teil1-schema-manager add-rule Article --name="Blogposts" --conditions='[{"type":"singular","value":"post"}]'
--schema-json='{"headline":"{{post_title}}"}'
wp teil1-schema-manager set-local 42 FAQPage --schema-json="$(jq -c . faq.json)"
wp teil1-schema-manager delete 3
wp teil1-schema-manager delete-rule 5
wp teil1-schema-manager clear-local 42
```

create und update betreffen globale Schemas. Regeln brauchen add-rule. Einzelne Seiten brauchen set-local.

PROMPT KOPIEREN

Prompt A – Bestand und Lücken

Arbeite ausschließlich lesend.

1. wp teil1-schema-manager doctor
2. wp teil1-schema-manager globals --format=json
3. wp teil1-schema-manager rules --format=json
4. wp teil1-schema-manager vars --format=json
5. wp teil1-schema-manager coverage
6. wp teil1-schema-manager health

Liefere danach:

- vorhandene Globals, Regeln, Custom Variables
- Kontexte ohne Regel
- doppelte oder widersprüchliche Typen
- Fehler vor Warnungen
- priorisierten Plan in drei Stufen: zuerst reparieren, dann Grundlagen (Organization, WebSite, WebPage, Article), dann Ausbau

Keine Änderungen.

PROMPT KOPIEREN

Prompt B – Globale Organization und WebSite härten

Lies zuerst globals, vars und inspect der vorhandenen Organization- und WebSite-Einträge. Wenn IDs unbekannt sind, nimm sie aus `wp teil1-schema-manager globals --format=json`.

Prüfe und schlage Korrekturen vor für:

- name, url, logo, description
- stabile @id-Werte, z. B. <https://example.com/#organization> und <https://example.com/#website>
- sameAs nur für echte Profile
- publisher/sourceOrganization-Verweise späterer Regeln

Wiederkehrende skalare Werte als Custom Variables vorschlagen:

```
{{custom.phone}}, {{custom.logo_url}}, {{custom.street}},
{{custom.postal_code}}, {{custom.locality}}, {{custom.country}}.
```

Ein strukturiertes PostalAddress-Objekt aus diesen Einzelwerten bauen;
Custom Variables können keine verschachtelten Objekte speichern.

Zeig das vollständige JSON und die update-/set-var-Befehle.

Schreibe erst nach Freigabe. Danach render die Startseiten-ID.

PROMPT KOPIEREN

Prompt C – Article-Regel für Blogposts

Ziel: eine Article-Regel für den Post Type `post`.

1. wp teil1-schema-manager variables
2. Bestehende Regeln und globale Organization/WebSite lesen.
3. Prüfen, ob Yoast, Rank Math oder ein anderes Plugin bereits Article/BlogPosting ausgibt. Wenn ja, im Plan vermerken: dort deaktivieren, bevor diese Regel aktiv wird.

Entwurf:

- Bedingung: `{{"type":"singular","value":"post"}}`
- headline: `{{post_title}}`
- url: `{{post_url}}`
- datePublished: `{{post_date}}`
- dateModified: `{{post_modified}}`
- image: `{{featured_image_url}}`
- author als verschachtelte Person mit ``name: {{author_name}}`` oder als @id-Verweis auf einen tatsächlich vorhandenen Person-Knoten
- publisher über @id der globalen Organization
- mainEntityOfPage: `{{post_url}}`

Zeig JSON, add-rule-Befehl und Überschneidungsrisiko.

Nach Freigabe anlegen, dann `render` für einen repräsentativen Post.

PROMPT KOPIEREN

Prompt D – Custom Variables zentral setzen

Lies `wp teil1-schema-manager vars --format=json` und die globalen Schemas. Finde hart kodierte skalare Werte, die an mehreren Stellen vorkommen: Telefon, Adressbestandteile, Logo, Preiswährung, Support-URL. Verschachtelte Objekte nicht als einzelne Custom Variable speichern.

Schlage Keys in Kleinschreibung vor und die zugehörigen `set-var`-Befehle. Zeig, welche Properties danach auf `{{custom.key}}` umgestellt werden sollten.

Keine Schema-Updates, bevor die Variablen und die betroffenen JSON-Diffs freigegeben sind.

PROMPT KOPIEREN

Prompt E – WooCommerce-Produkte ohne Duplikate

Prüfe zuerst:

- ist WooCommerce aktiv? (doctor)
- existiert bereits eine Product-Regel?
- welche Produktvariablen listet `variables`?
- welche Ausgabe erzeugt `render <produkt-id>` heute?

Wenn Teil1 Schema Manager Product selbst ausgeben soll:

1. Product-Regel für singular:product entwerfen
2. `{{product_price}}`, `{{product_currency}}`, `{{product_availability}}`, `{{product_sku}}`, `{{product_brand}}` nur verwenden, wenn die Werte im Shop sichtbar sind
3. bei variablen Produkten AggregateOffer erwarten, nicht den niedrigsten Einzelpreis als einzigen Offer-Preis darstellen
4. im Plan vermerken: optionale Konfliktunterdrückung für Product, Review, BreadcrumbList, WebSite ist in den Plugin-Einstellungen und standardmäßig aus

Kein Review- oder AggregateRating-Markup ohne sichtbare Bewertungen.
Nach Freigabe Regel anlegen und ein Einfach- plus ein Variantenprodukt rendern.

PROMPT KOPIEREN

Prompt F – Lokales FAQ nur bei sichtbarem Inhalt

Seite: Post-ID {ID}.

1. `wp teil1-schema-manager local {ID} --format=json`
2. `wp teil1-schema-manager render {ID}`
3. Sichtbaren Inhalt der Seite lesen.

Wenn dort ein echter FAQ-Block steht: FAQPage-JSON aus den sichtbaren Fragen und Antworten bauen. Keine Fragen erfinden, keine Antworten umformulieren, bis sie „schema-tauglicher“ klingen.

Wenn kein FAQ-Block existiert: kein FAQPage anlegen.
Zeig `set-local`-Befehl plus JSON-Datei. Warte auf Freigabe.
Danach erneut rendern.

PROMPT KOPIEREN

Prompt G – Backup, Diff, kontrollierter Import

Erstelle zuerst Backups außerhalb des öffentlichen Webroots:

```
umask 077
wp teil1-schema-manager export > /sicherer/pfad/schema-backup-$(date +%Y%m%d).json
wp option get t1schema_custom_variables --format=json > /sicherer/pfad/custom-variables-$(date +%Y%m%d).json
wp db export /sicherer/pfad/database-$(date +%Y%m%d).sql
jq empty /sicherer/pfad/schema-backup-$(date +%Y%m%d).json
```

Die geplante Änderung muss entweder dem Exportformat ``{"globals": [...], "rules": [...], "locals": [...]}`` oder dem lokalen Flat-Array-Format ``[{"post_id": 42, "type": "Article", "data": {...}]`` entsprechen.

Prüfe den Import mit:

```
wp teil1-schema-manager import geplante-aenderung.json --dry-run
```

Vergleiche Dry-Run, Datei und aktuellen Bestand explizit. Globals und Regeln werden beim Import neu angelegt und können Duplikate erzeugen; lokale Schemas werden für betroffene Post-IDs ersetzt. Importiere erst nach Freigabe. Danach doctor, health und render der betroffenen IDs.

PROMPT KOPIEREN

Prompt H – Qualitätssicherung nach Änderungen

Führe nacheinander aus:

```
wp teil1-schema-manager doctor
wp teil1-schema-manager health
wp teil1-schema-manager coverage
```

Rendere danach diese IDs: {STARTSEITE}, {SEITE}, {POST}, {PRODUKT}.

Prüfbericht:

1. Fehler, die Ausgabe blockieren oder ungültiges JSON erzeugen
2. doppelte oder widersprüchliche Entitäten mit gleicher @id; derselbe @type allein ist noch kein Duplikat
3. Knoten ohne stabile @id, die zusammengehören
4. Variablen, die leer bleiben
5. Markup, das auf der Seite nicht sichtbar ist

Keine stillen Korrekturen. Jeden Fix als einzelnen, freizugebenden Schritt vorschlagen.

Teil1-Blog-Formate ab Version 2.4.1

Für den Post Type `teil1_blog_post` ordnet das Plugin den kanonischen Artikel-Knoten anhand des Metafelds `_teil1_content_editorial_format` automatisch zu:

- `news` → `NewsArticle`
- `ratgeber` oder `anleitung` → `BlogPosting`
- alle anderen Werte → `Article`

Agenten dürfen diesen Typ nicht zusätzlich durch eine parallele Regel duplizieren. Vor Änderungen zuerst `render <post_id>` prüfen.

Kurze Entscheidungsregeln

- Wiederkehrendes Muster → Regel.
- Echte Ausnahme auf einer URL → lokal.
- Firma, Website, Logo, Profile → global plus Custom Variables.
- Sichtbarer FAQ-/HowTo-/Video-/Event-Inhalt → lokales Schema oder eng begrenzte Regel.
- Semantisch dieselbe Entität aus einem SEO-Plugin oder WooCommerce → zuerst die andere Ausgabe abstellen oder die optionale Unterdrückung bewusst einschalten.
- Unklarer Inhalt → nichts schreiben, nachfragen.

Quellen

- Plugin: <https://wordpress.org/plugins/teil1-schema-manager/>
- Produktseite: <https://teil1.com/t1-schema/>
- Quellcode / CLI: <https://github.com/teil1-com/t1-schema>
- Schema.org: <https://schema.org/>
- Google, strukturierte Daten: <https://developers.google.com/search/docs/appearance/structured-data/intro-structured-data>