PMG · Media4AI

PMG Media4AI

Technischer Leitfaden für die Integration in kundeneigene RAG-Systeme

PMG Media4AI stellt lizenzierte journalistische Medienbeiträge als strukturierte Daten für unternehmensinterne KI-, Recherche- und Wissensanwendungen bereit. Die Inhalte können in kundeneigene Systeme übernommen, indexiert und als externer Wissensbestand in einer Retrieval-Augmented-Generation-Architektur genutzt werden.

Dieser Leitfaden beschreibt den Aufbau der bereitgestellten Daten, die Integration in ein eigenes RAG-System sowie die wichtigsten technischen und vertraglichen Anforderungen.

1. Datenbereitstellung

Die ausgewählten Medienbeiträge werden als strukturiertes Exportpaket bereitgestellt. Das Paket enthält:

- eine übergeordnete Datei manifest.json,
- einen zeilenbasierten Dateiindex manifest.ndjson,
- separate JSON-Dateien für die ausgewählten Medienbeiträge,
- eine nach Publikationskennungen gegliederte Verzeichnisstruktur.

Die Datei manifest.json enthält zentrale Angaben zum Export, darunter:

- Auftrags- und Nutzerkennung,
- Anzahl der exportierten Beiträge,
- Exportzeitpunkt,
- Name des zugehörigen NDJSON-Manifests,
- bereitgestellte Paketteile.

Ein produktiver Export weist beispielsweise Anzahl der Beiträge, den Exportzeitpunkt und den Dateinamen des zugehörigen ZIP-Paketteils aus.

Beispielhafte Paketstruktur

```
Media4AI-Export/
│
├── manifest.json
├── manifest.ndjson
├── PUBLIKATION_A/
│   └── eine JSON-Datei je Medienbeitrag
├── PUBLIKATION_B/
│   └── eine JSON-Datei je Medienbeitrag
└── ...
```

manifest.ndjson dient als maschinenlesbarer Dateiindex. Jeder Beitrag wird durch einen eigenen JSON-Datensatz pro Zeile beschrieben und über einen relativen Pfad der zugehörigen Beitragsdatei zugeordnet.

2. Aufbau der Beitragsdaten

Jeder Medienbeitrag wird als eigenständige JSON-Datei bereitgestellt. Das übergeordnete Datenmodell ist einheitlich aufgebaut. Umfang und Inhalt einzelner Felder können abhängig von der Publikation und den verfügbaren Ausgangsdaten variieren.

Inhalt

- headline: Hauptüberschrift
- subhead: Unterüberschrift
- intro: Vorspann oder Einleitung
- story: Beitragstext als Klartext
- story_xml: strukturierte XML-Repräsentation des Beitrags
- mediatype: Kennzeichnung der Medienart

Autoren

byline enthält eine Liste von Autorenobjekten. Je nach Quelle können darin folgende Informationen enthalten sein:

- name: vollständige Autorenangabe beziehungsweise Anzeigename
- firstname: Vorname, sofern vorhanden
- surname: Nachname, sofern vorhanden

Ein Beitrag kann einen oder mehrere Autoren enthalten. Einzelne Autorenobjekte können ausschließlich ein name-Feld enthalten. Kundensysteme sollten daher weder eine feste Anzahl von Autoren noch das Vorhandensein sämtlicher Namensbestandteile voraussetzen.

Beispiel:

```
"byline": [
  {
    "name": "Beispielname",
    "firstname": "Vorname",
    "surname": "Nachname"
  }
]

```

Quelle

Das Objekt source enthält unter anderem:

- id: Publikationskennung
- name: Name der Publikation
- date: Publikationsdatum im FormatYYYYMMDD
- day: Wochentag
- country_code: Ländercode
- language_code: Sprachcode

Die Sprache des Wochentags kann unabhängig von der Dokumentensprache sein. Für Anzeige- oder Filterzwecke sollte deshalb vorzugsweise source.date als Datum verarbeitet und bei Bedarf kundenseitig lokalisiert werden.

Struktur und Identifikation

- logical_position: Rubrik oder logische Position innerhalb der Publikation
- subcategory: redaktionelle Unterkategorie
- page: Seitenangabe
- item_id: Beitragskennung

Lizenzinformationen

- copyright: Nutzungshinweis zum jeweiligen Beitrag
- copyright_hash: technischer Wert im Zusammenhang mit dem Copyright-Hinweis

Der Copyright-Hinweis sollte gemeinsam mit dem Quelldokument gespeichert und bei internen Verarbeitungsschritten erhalten bleiben. Die technische Funktion und Validierung von copyright_hash richtet sich nach der jeweils gültigen Produktspezifikation.

Optionale und leere Felder

Abhängig von Publikation und Ausgangsdaten können einzelne Felder als leere Zeichenkette enthalten sein. Dies betrifft beispielsweise:

- subhead,
- intro,
- subcategory.

Diese Felder sollten als optional behandeln und bei der Validierung, Indexierung und Darstellung entsprechende Fallback-Regeln vorsehen.

3. Klartext und strukturierter Inhalt

Strukturierter Inhalt in story_xml

Das Feld story_xml enthält den Beitrag als XML-String. Abhängig von der Publikation kann die XML-Struktur unter anderem folgende Bestandteile abbilden:

- Absätze,
- Zwischenüberschriften,
- Kästen,
- Abbildungen,
- Bildunterschriften,
- Fotografenangaben,
- Infografik-Referenzen.

Die verwendeten XML-Elemente können sich zwischen Publikationen unterscheiden. Kundensysteme sollten deshalb eine flexible XML-Verarbeitung vorsehen und nicht ausschließlich auf einen einzigen festen Satz von Elementnamen ausgelegt sein.

Da story_xml als String innerhalb des JSON-Dokuments geliefert wird, ist bei Nutzung der Dokumentstruktur nach dem JSON-Parsing ein zusätzlicher XML-Parsing-Schritt erforderlich.

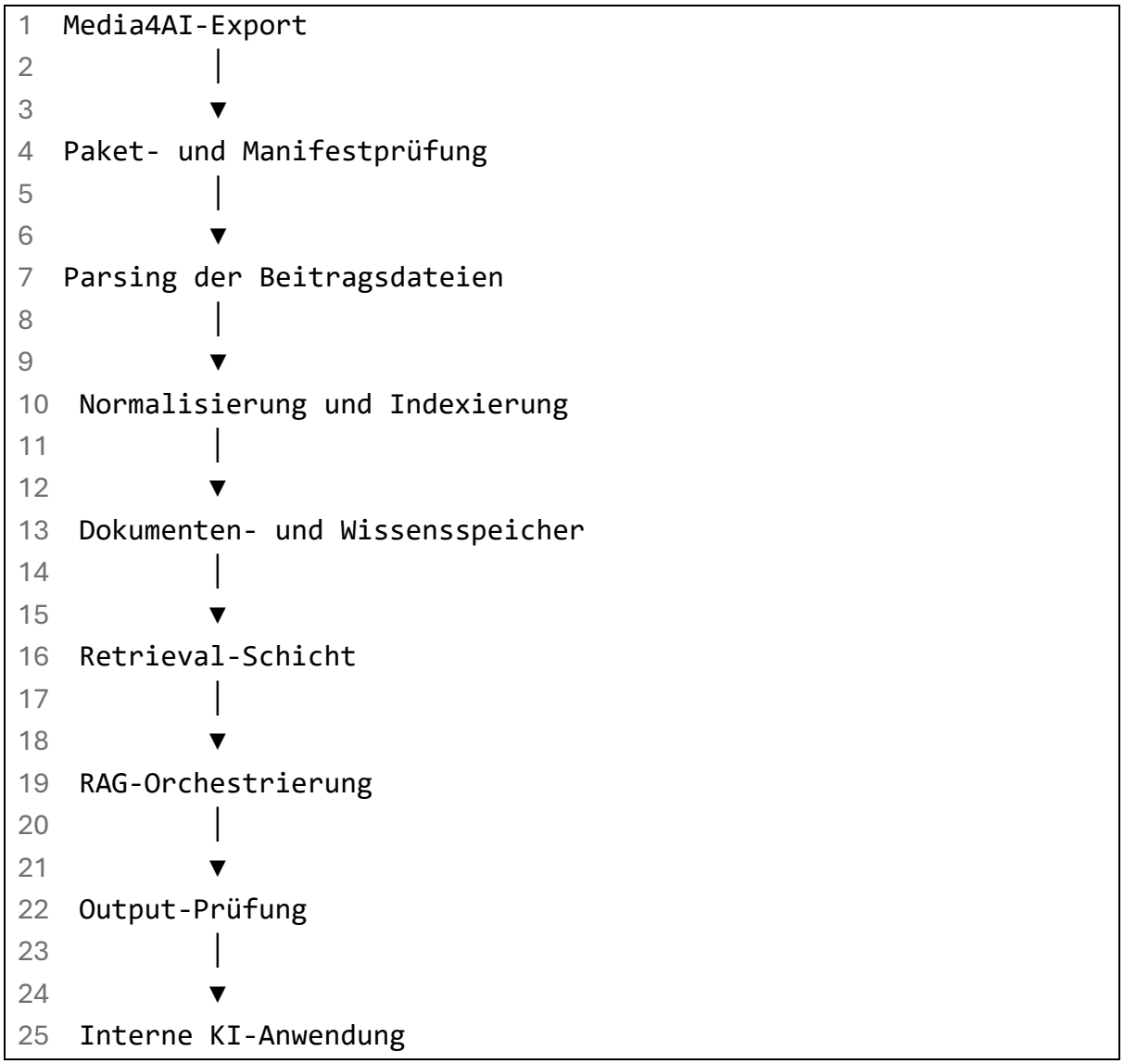
Klartext in story

Das Feld story enthält die lesbare Textfassung des Beitrags. Es eignet sich insbesondere als Grundlage für:

- Volltextsuche,
- Segmentierung,
- Indexierung,
- Retrieval,
- Informationsextraktion,
- temporäre Bereitstellung relevanter Textpassagen an das RAG-System.

4. Empfohlene Integration

Eine typische kundenseitige Verarbeitung kann wie folgt aufgebaut werden:



Empfohlener Ablauf

1. manifest.json einlesen und die Exportinformationen erfassen.
2. manifest.ndjson zeilenweise verarbeiten.
3. Beitragsdateien über die relativen Pfadangaben zuordnen.
4. JSON-Struktur und Datentypen validieren.
5. Leere und optionale Felder kontrolliert behandeln.
6. story als Klartext und story_xml bei Bedarf als strukturierten Inhalt verarbeiten.
7. Inhalte und Metadaten in das kundeneigene Datenmodell übernehmen.
8. Dokumente für Suche und Retrieval indexieren.
9. Segmente und abgeleitete Datensätze auf die ursprüngliche item_id zurückführen.
10. Bezugs- und Löschzeitpunkte nachvollziehbar verwalten.

Für die RAG-Implementierung empfiehlt es sich, story als primäre Textgrundlage und die übrigen Felder für Filterung, Quellenanzeige, Zugriffssteuerung und Lifecycle-Management zu verwenden.

5. Nutzung im RAG-System

Media4AI-Inhalte werden als externer Wissensbestand außerhalb des eingesetzten KI-Modells geführt.

```
Benutzeranfrage
        |
        ▼
Retrieval relevanter Media4AI-Inhalte
        |
        ▼
Temporäre Bereitstellung als Modellkontext
        |
        ▼
 Generierung einer internen Antwort
        |
        ▼
 Prüfung und Quellenanzeige
```

Innerhalb der geschützten Kundenumgebung dürfen die bezogenen Medienbeiträge insbesondere:

- gespeichert,
- in andere maschinenlesbare Formate konvertiert,
- nach Inhalt und Metadaten indexiert,
- mit Metadaten, Schlagworten und Annotationen angereichert,
- zur Informationsextraktion verarbeitet,
- als externer Wissensbestand eines RAG-Systems genutzt,
- zur Erzeugung interner KI-Ausgaben verwendet werden.

Nicht zulässig sind insbesondere:

- Training eines KI-Modells mit den Medienbeiträgen,
- Fine-Tuning auf Basis der Medienbeiträge,
- Modifikation eines KI-Modells mithilfe der Inhalte,
- Überführung der Inhalte in Modellgewichte oder Modellwissen,
- unzulässige Weitergabe an Dritte,
- Nutzung als Ersatz für eine kontinuierliche Medienbeobachtung oder Medienauswertung.

6. Begrenzung wortwörtlicher Ausgaben

Wortwörtliche Auszüge aus einem Medienbeitrag dürfen im KI-Output maximal 150 Zeichen umfassen. Antworten sollten daher grundsätzlich in eigenen Worten formuliert werden.

Für eine belastbare Umsetzung wird eine Kombination aus drei Maßnahmen empfohlen:

1. verbindliche Regel im Systemprompt,
2. automatisierte Prüfung der generierten Antwort,
3. Blockierung oder erneute Generierung bei einem Verstoß.

Beispiel für eine Prompt-Regel

```
Formuliere die Antwort grundsätzlich in eigenen Worten.

Übernimm aus keinem einzelnen Medienbeitrag eine wortwörtliche
Passage mit mehr als 150 Zeichen.

Gib keine vollständigen Artikel, längeren Absätze oder
umfangreichen Originalpassagen aus.

 Setze nicht mehrere kurze Zitate so zusammen, dass dadurch eine
längere Originalpassage wiedergegeben wird.

 Wenn die Anfrage nur durch eine längere wortwörtliche
 Wiedergabe
 beantwortet werden könnte, fasse den Inhalt stattdessen
 zusammen.

 Nenne die für die Antwort verwendeten Quellen.
```

Eine Prompt-Anweisung allein stellt keine verlässliche technische Kontrolle dar. Die Anwendung sollte die generierte Antwort zusätzlich mit den verwendeten Quelltexten vergleichen. Wird eine unzulässig lange wortwörtliche Übereinstimmung erkannt, sollte die Antwort blockiert oder erneut in eigenen Worten generiert werden.

7. Quellenangaben

Die Quellenbasis einer KI-generierten Antwort soll offengelegt werden. Bei Online-Quellen soll nach Möglichkeit auf die Originalquelle verwiesen werden.

Für die Quellenanzeige stehen unter anderem folgende Daten zur Verfügung:

- Publikationsname,
- Publikationsdatum,
- Überschrift,
- Beitragskennung,
- Rubrik,
- Seitenangabe.

Beispiel:

```
Quelle:
Publikationsname, 02.06.2026,
„Beitragsüberschrift“,
Beitrags-ID: Beitragskennung
```

Links zur Originalquelle sollten nur angezeigt werden, wenn eine entsprechende URL mit den Daten bereitgestellt oder durch das Kundensystem verlässlich zugeordnet wurde.

Für eine robuste Quellenanzeige sollte das RAG-System die verwendeten item_id-Werte strukturiert zurückgeben. Die sichtbare Quellenangabe kann anschließend durch die Anwendung aus den zugehörigen Metadaten erzeugt werden.

8. Sicherheit und Zugriff

Die Medienbeiträge dürfen innerhalb eines gesicherten elektronischen Netzwerks gespeichert und ausschließlich einem definierten internen Teilnehmerkreis zugänglich gemacht werden.

Für die technische Umsetzung werden insbesondere empfohlen:

- rollen- oder gruppenbasierte Zugriffsrechte,
- Berechtigungsprüfung vor dem Retrieval,
- geschützte Speicherung der Quelldaten und Indizes,
- kontrollierte Anbindung von KI- und Drittanbietersystemen,
- Trennung von Quelldaten und öffentlich erreichbaren Anwendungen,
- Protokollierung von Import, Verarbeitung, Zugriff und Löschung,
- Behandlung der abgerufenen Medieninhalte ausschließlich als Daten.

Anweisungen, Rollenwechsel oder Aufforderungen, die innerhalb eines Medienbeitrags enthalten sein könnten, dürfen vom KI-System nicht als System- oder Benutzeranweisungen ausgeführt werden.

9. Datenlebenszyklus und Löschung

Die bezogenen Medienbeiträge müssen grundsätzlich spätestens zwölf Monate nach dem jeweiligen Bezug aus den relevanten Kundensystemen gelöscht werden.

Die Löschung umfasst insbesondere:

- Originaldateien,
- normalisierte Dokumente,
- gespeicherte Klartexte und XML-Repräsentationen,
- Textsegmente,
- Suchindizes,
- Vektorindizes und Embeddings,
- Metadatenanreicherungen,
- Markierungen, Schlagworte und Annotationen.

Kundenseitig erzeugte KI-Ausgaben sind von dieser regulären Löschpflicht ausgenommen.

Für jeden Beitrag sollten der Bezugszeitpunkt, das späteste Löschdatum und der Löschstatus nachvollziehbar verwaltet werden. Alle abgeleiteten Datensätze und Indexeinträge sollten über die item_id oder eine darauf aufbauende interne Kennung auf das Ausgangsdokument zurückgeführt werden können.

10. Hinweise zur technischen Umsetzung

Das übergeordnete JSON-Datenmodell wird publikationsübergreifend verwendet. Einzelne Inhalte können jedoch abhängig von Publikation und Quelldaten variieren. Dies betrifft insbesondere:

- Anzahl und Struktur der Autorenangaben,
- Befüllung optionaler Metadatenfelder,
- verwendete XML-Elemente,
- Medienart,
- Rubriken und Unterkategorien,

Der Parser und Datenmodelle sollten daher robust gegenüber leeren Feldern, mehreren Autoren, zusätzlichen XML-Elementen und publikationsabhängigen Inhaltsstrukturen auslegen.

Maßgeblich für die produktive Integration und Nutzung sind die vereinbarten Verträge, die Geschäftsbedingungen für PMG Media4AI sowie die jeweils gültige technische Produktspezifikation.