JobTrends – Detail Job JSON Schema (Lean Version)

Ziel

Dieses Dokument definiert:

1. Das neue schlanke JSON-Schema für Detailjobanzeigen
2. Die Prinzipien für fehlende Felder
3. Das generische Mapping-Mechanismus-Template (HTML + JSON kompatibel)

Dieses Schema ist bewusst minimal gehalten und enthält: - Kein RAW-HTML - Kein Enrichment-Block - Kein Salary-Breakdown (nur reiner Text)

---

1. Ziel-JSON-Schema (Detailjob)

```
{
  "schema_version": "0.1",

  "job_id": null,
  "company_key": null,
  "url": null,
  "scraped_at": null,
  "locale": null,

  "meta": {
    "title": null,
    "location_text": null,
    "posting_date": null,
    "employment_type": null,
    "contract_type": null,
    "career_level": null,
    "salary_text": null
  },

  "extracted": {
    "fulltext": null,

    "overview": null,

    "responsibilities": {
      "items": []
    },

    "requirements": {
```

```
      "items": []
    },

    "benefits": {
      "items": []
    },

    "additional": {
      "items": []
    },

    "process": null
  }
}
```

---

2. Verhalten bei fehlenden Daten

Das Schema ist robust gegenüber fehlenden Informationen.

Regeln

- Strings → `null`
- Listen → `[]`
- Objekte → bleiben strukturell vorhanden

Beispiele

Kein Salary vorhanden:

```
"salary_text": null
```

Keine Benefits vorhanden:

```
"benefits": { "items": [] }
```

Keine klare Overview trennbar:

```
"overview": null
```

Wichtig: Keys werden niemals weggelassen. Die Struktur bleibt immer identisch.

---

3. Mapping-Mechanismus – Konzept

Das Mapping-Dict beschreibt NICHT die extrahierten Inhalte. Es beschreibt nur:

- Welche CSS-Selektoren
- Welche HTML-Attribute
- Welche JSON-Pfade

auf welche Ziel-Keys gemappt werden.

Das Mapping ist template-basiert und website-spezifisch befüllbar.

---

4. Generisches Mapping-Template (Website-unabhängig)

```
MAPPING_TEMPLATE = {

  "schema_version": "0.1",

  "source": {
    "company_key": "<SET_ME>",
    "locale": "<SET_ME>"
  },

  "content_root": {
    "html": {
      "css": "<ROOT_CONTAINER_SELECTOR>",
      "exclude_css": (
        "script, style, nav, header, footer"
      )
    },
    "json": {
      "path_candidates": [
        "$.job.descriptionHtml",
        "$.description",
        "$.data.job.description"
      ]
    }
  },

  "fields": {

    # ---------- Core ----------

    "job_id": {
      "from_html": ["<CSS_SELECTOR>"],
      "from_html_attr": [
```

```
      {"css": "<CSS_SELECTOR>", "attr": "data-job-id"}
    ],
    "from_json": [
      "$.job.id",
      "$.jobId"
    ]
  },

  "company_key": {"static": "<SET_ME>"},
  "url": {"runtime": "request_url"},
  "scraped_at": {"runtime": "utc_now"},
  "locale": {"static": "<SET_ME>"},

  # ---------- Meta ----------

  "meta.title": {
    "from_html": ["<CSS_SELECTOR>"],
    "from_json": ["$.job.title"]
  },

  "meta.location_text": {
    "from_html": ["<CSS_SELECTOR>"],
    "from_json": ["$.job.location"]
  },

  "meta.posting_date": {
    "from_html": ["<CSS_SELECTOR>"],
    "from_json": ["$.job.postingDate"]
  },

  "meta.employment_type": {
    "from_html": ["<CSS_SELECTOR>"],
    "from_json": ["$.job.employmentType"]
  },

  "meta.contract_type": {
    "from_html": ["<CSS_SELECTOR>"],
    "from_json": ["$.job.contractType"]
  },

  "meta.career_level": {
    "from_html": ["<CSS_SELECTOR>"],
    "from_json": ["$.job.careerLevel"]
  },

  "meta.salary_text": {
    "from_html": ["<CSS_SELECTOR_FOR_SALARY_TEXT>"],
    "from_json": ["$.job.salaryText"]
  },

  # ---------- Extracted ----------
```

```
"extracted.fulltext": {
  "from_html_content_root_text": True,
  "from_json": ["$.job.descriptionText"]
},

"extracted.overview": {
  "html_section_by_heading": {
    "root_css": "<ROOT_CONTAINER_SELECTOR>",
    "heading_text_candidates": ["Overview"],
    "heading_selectors": ["h2", "h3", "strong"],
    "collect_until_next_heading": True
  }
},

"extracted.responsibilities.items": {
  "html_list_between_headings": {
    "root_css": "<ROOT_CONTAINER_SELECTOR>",
    "start_heading_text_candidates": ["Responsibilities"],
    "end_heading_text_candidates": ["Requirements"],
    "list_item_css": "li"
  }
},

"extracted.requirements.items": {
  "html_list_between_headings": {
    "root_css": "<ROOT_CONTAINER_SELECTOR>",
    "start_heading_text_candidates": ["Requirements"],
    "end_heading_text_candidates": ["Benefits"],
    "list_item_css": "li"
  }
},

"extracted.benefits.items": {
  "html_section_by_heading": {
    "root_css": "<ROOT_CONTAINER_SELECTOR>",
    "heading_text_candidates": ["Benefits"],
    "heading_selectors": ["h2", "h3"],
    "collect_until_next_heading": True
  }
},

"extracted.additional.items": {
  "html_list_between_headings": {
    "root_css": "<ROOT_CONTAINER_SELECTOR>",
    "start_heading_text_candidates": [
      "Nice to Have",
      "Nice-to-have",
      "Bonus Skills",
      "Preferred Qualifications",
      "Additional Skills"
```

```
        ],
        "end_heading_text_candidates": [
          "Benefits",
          "Salary",
          "About",
          "Equal Opportunities"
        ],
        "list_item_css": "li"
      }
    },

    "extracted.process": {
      "from_html": ["<CSS_SELECTOR_FOR_PROCESS_SECTION>"],
      "from_json": ["$.job.hiringProcess"]
    }

  }
}
```

---

5. Architekturprinzip

Dieses Setup trennt klar:

1. Struktur (JSON-Schema)
2. Website-spezifisches Mapping
3. Extraktionslogik (separates Modul)

Vorteile:

- Einheitliche Datenstruktur über alle Companies
- Saubere Versionierung
- Wiederverwendbares Mapping-Template
- Robust gegen fehlende Felder
- Salary als einfacher Text ohne Parsing-Komplexität

Ende des Dokuments