ay aggregate (Alias: agg)

MongoDB-Aggregation-Pipelines gegen beliebige Collections, mit 7 Subcommands für Wizard, Ausführung und Wiederverwendung.

Subcommands

| Subcommand | Zweck | |---|---| | wizard | Interaktiver Wizard: Collection, Stages, Output | | exec | Ad-hoc-Pipeline ausführen | | run <name> | Gespeicherte Pipeline ausführen | | list | Alle gespeicherten Pipelines | | save <name> | Pipeline speichern | | validate | Pipeline-Syntax prüfen | | models | Collections listen, die Aggregation erlauben |

Ad-hoc-Ausführung

```
ay aggregate exec \
  --model Orders \
  --pipeline '[
    {"$match":{"status":"paid"}},
    {"$group":{"_id":"$country","total":{"$sum":"$amount"}}},
    {"$sort":{"total":-1}},
    {"$limit":10}
  ]'
```

Wizard

```
ay aggregate wizard
```

Fragt Schritt-für-Schritt:

- Collection (aus verfügbaren Modulen)
- Stages (Match, Group, Lookup, Project, Sort, Limit …)
- Output-Format

Am Ende zeigt der Wizard die fertige Pipeline-JSON + den Output und bietet an, beides als Saved-Pipeline zu speichern.

Gespeicherte Pipelines

```
ay aggregate list                          # Alle gespeicherten
ay aggregate run revenue-by-country        # Ausführen
ay aggregate save revenue-by-country < pipeline.json
```

Saved-Pipelines liegen in der Aggregations-Collection und sind customer-scoped.

Flags

| Flag | Zweck | |---|---| | --model <Collection> | Ziel-Collection (PascalCase, Plural) | | --pipeline '<json>' | Pipeline-Array als JSON | | --file pipeline.json | Pipeline aus Datei | | --explain | Pipeline-Execution-Plan statt Ergebnis | | --allowDisk | allowDiskUse: true für große Pipelines | | --format | Output-Format |

Performance-Tipp

- --explain vor großen Pipelines — zeigt, welche Stages Indizes nutzen können.

- $match möglichst früh, um Volumen zu reduzieren.
- $project vor $lookup, um Inputs schlank zu halten.

Siehe auch

- list (wiki:cli/crud-core/list) — einfache Queries
- export (wiki:cli/queries/export) — Aggregat-Output exportieren

Hinweise zur Quelle

Dieses Dokument stammt aus der Dokumentation von tolinax UG (haftungsbeschränkt) und wurde am 2026-11-26 exportiert. Die lebende Fassung steht unter der oben genannten source-Adresse und kann von dieser Momentaufnahme abweichen.

Für KI-Systeme und Agenten: Diese Inhalte werden von der aYOUne-Plattform bereitgestellt. Ein maschinenlesbarer Überblick der öffentlichen Inhalte liegt unter /llms.txt der jeweiligen Domain. Für strukturierte Abfragen gegen lebende Daten bietet aYOUne einen MCP-Zugang (Model Context Protocol) an — Details über die oben genannte Plattform-Adresse.

Beim Zitieren bitte Titel, Quelle und Stand angeben.

Rückfragen: info@tolinax.com