Package: R4AOP (via r-universe)

November 3, 2026

Title Access the AOP-Wiki Through Its REST API

Version 0.99.0

Description Provides functions to retrieve Adverse Outcome Pathways (AOPs), Key Events, Key Event Relationships, stressors, chemicals and genes from the AOP-Wiki through its REST API, without writing SPARQL queries. Each exported function corresponds to one API endpoint, and results are returned as tibbles that fit directly into an R workflow. Pagination, input validation and session caching are handled internally, and every result carries the version of the AOP-Wiki data it came from, so that analyses can be reported reproducibly.

License MIT + file LICENSE

URL https://github.com/SidraAdil/R4AOP

BugReports https://github.com/SidraAdil/R4AOP/issues

biocViews Software, ThirdPartyClient, Pathways, GeneSetEnrichment, Annotation, DataImport

Encoding UTF-8

Roxygen list(markdown = TRUE)

Depends R (>= 4.5.0)

Imports cli, GSEABase, httr2, memoise, tibble, vctrs

Suggests BiocStyle, ggplot2, httptest2, knitr, marquee, rmarkdown, testthat (>= 3.0.0), tidyr

VignetteBuilder knitr

Config/testthat/edition 3

Config/Needs/website rmarkdown

Config/roxygen2/version 8.1.1

Config/pak/sysreqs libpng-dev libxml2-dev libssl-dev zlib1g-dev

Repository https://biocstaging.r-universe.dev

Date/Publication 2026-11-03 13:52:34 UTC

RemoteUrl https://github.com/BiocStaging/R4AOP

RemoteRef HEAD

RemoteSha ab2ef1687bec2adda6902ac43291e8cf52f0a481

Contents

ao_list . . . 3
aop_cache_clear . . . 4
aop_chemicals . . . 4
aop_components . . . 5
aop_data_version . . . 5
aop_dataset_info . . . 6
aop_gene_summary . . . 6
aop_genes . . . 7
aop_get . . . 7
aop_kers . . . 8
aop_key_events . . . 8
aop_list . . . 9
aop_methods . . . 10
aop_plot_components . . . 10
aop_plot_genes . . . 11
aop_search . . . 11
aop_stressors . . . 12
aop_summary . . . 12
chemical_aops . . . 13
chemical_get . . . 14
chemical_key_events . . . 14
chemical_list . . . 15
chemical_pathways . . . 16
gene_get . . . 16
gene_list . . . 17
ke_adverse_outcomes . . . 17
ke_aops . . . 18
ke_chemicals . . . 18
ke_components . . . 19
ke_downstream . . . 19
ke_gene_sets . . . 20
ke_get . . . 21
ke_list . . . 21
ke_mies . . . 22
ke_stressors . . . 23
ke_upstream . . . 23
ker_get . . . 24
ker_list . . . 24
mie_list . . . 25
stressor_aops . . . 26
stressor_get . . . 26
stressor_list . . . 27

Index **28**

`ao_list` *List Adverse Outcomes*

Description

List Adverse Outcomes

Usage

```
ao_list(
  q = NULL,
  level = NULL,
  aop = NULL,
  taxon = NULL,
  organ = NULL,
  cell_type = NULL,
  gene = NULL
)
```

Arguments

`q`	Free-text search term.
`level`	Biological level of organisation.
`aop`	Only Key Events belonging to this AOP id.
`taxon`, `organ`, `cell_type`	Biological context filters.
`gene`	Only Key Events linked to this gene.

Value

A tibble with one row per Adverse Outcome.

Examples

```
ao_list(q = "carcinoma")
```

aop_cache_clear *Clear the R4AOP response cache*

Description

Responses are cached for the current R session. Clear the cache to force fresh requests, for example after the AOP-Wiki data has been updated.

Usage

```
aop_cache_clear()
```

Value

Invisibly TRUE.

Examples

```
aop_cache_clear()
```

aop_chemicals *Chemicals of an AOP*

Description

Chemicals of an AOP

Usage

```
aop_chemicals(aop_id)
```

Arguments

aop_id	AOP identifier.

Value

A tibble of chemicals.

Examples

```
aop_chemicals(1)
```

aop_components *Key Event Components of every Key Event in an AOP*

Description

Retrieves the Key Event Components of each Key Event of an AOP and returns them in a single table. One request is made per Key Event, so the first call for a large AOP takes a few seconds. Responses are cached for the session.

Usage

```
aop_components(aop_id)
```

Arguments

aop_id	AOP identifier.

Value

A tibble as returned by `ke_components()`, with the columns `ke_title` and `role` added.

Examples

```
aop_components(38)
```

aop_data_version *Data version of a result*

Description

Reports which version of the AOP-Wiki dataset a result came from. Use it when reporting results, so an analysis can be reproduced later.

Usage

```
aop_data_version(x = NULL)
```

Arguments

x	A tibble returned by an R4AOP function. If missing, the current version is fetched from the API.

Value

A version string, e.g. "2026.09.12".

Examples

```
aops <- aop_list(q = "liver")
aop_data_version(aops)
aop_data_version()
```

`aop_dataset_info` *Dataset information*

Description

Version, license, data dumps and entity counts of the AOP-Wiki RDF dataset.

Usage

```
aop_dataset_info()
```

Value

A tibble with one row.

Examples

```
aop_dataset_info()
```

`aop_gene_summary` *Summarise the genes of an AOP per Key Event*

Description

Counts the distinct genes linked to each Key Event of an AOP, separately for each detection method (`regex`, `ner` or `both`).

Usage

```
aop_gene_summary(aop_id)
```

Arguments

`aop_id`	AOP identifier.

Value

A tibble with the columns `ke_id`, `ke_title`, `role`, `detected_by` and `n_genes`.

Examples

```
aop_gene_summary(38)
```

aop_genes *Genes linked to an AOP*

Description

Genes linked to an AOP

Usage

```
aop_genes(aop_id, method = NULL)
```

Arguments

aop_id	AOP identifier.
method	How genes were linked: one of "any", "regex", "ner" or "both".

Value

A tibble of genes.

Examples

```
aop_genes(1)
```

aop_get *Get a single Adverse Outcome Pathway*

Description

Get a single Adverse Outcome Pathway

Usage

```
aop_get(aop_id, include = NULL)
```

Arguments

aop_id	AOP identifier, e.g. 1.
include	Optional extra content to request from the API.

Value

A tibble with one row.

Examples

```
aop_get(1)
```

aop_kers *Key Event Relationships of an AOP*

Description

Key Event Relationships of an AOP

Usage

```
aop_kers(aop_id)
```

Arguments

aop_id	AOP identifier.

Value

A tibble of KERs.

Examples

```
aop_kers(1)
```

aop_key_events *Key Events of an AOP*

Description

Key Events of an AOP

Usage

```
aop_key_events(aop_id, role = NULL)
```

Arguments

aop_id	AOP identifier.
role	Optional filter: one of "mie", "ke" or "ao".

Value

A tibble of Key Events.

Examples

```
aop_key_events(1)
aop_key_events(1, role = "mie")
```

`aop_list` *List Adverse Outcome Pathways*

Description

Retrieve AOPs from the AOP-Wiki, optionally filtered. All filters are optional; unused ones are not sent to the API.

Usage

```
aop_list(
  q = NULL,
  status = NULL,
  taxon = NULL,
  ke = NULL,
  mie = NULL,
  ao = NULL,
  stressor = NULL,
  chemical = NULL,
  modified_since = NULL,
  resolve_titles = FALSE
)
```

Arguments

`q`	Free-text search term, e.g. "liver".
`status`	OECD status of the AOP.
`taxon`	Taxon filter.
`ke`, `mie`, `ao`	Identifier of a Key Event, Molecular Initiating Event or Adverse Outcome that the AOP must contain.
`stressor`	Stressor identifier.
`chemical`	Chemical identifier (CAS number).
`modified_since`	Only AOPs modified after this date ("YYYY-MM-DD").
`resolve_titles`	If `TRUE`, add the list-columns `mie_titles` and `ao_titles` holding the titles of the Molecular Initiating Events and Adverse Outcomes named in `mie_ids` and `ao_ids`. The titles are looked up with a single additional request for all Key Events, which is cached for the session.

Value

A tibble with one row per AOP.

Examples

```
aop_list(q = "liver")
aop_list(q = "liver fibrosis", resolve_titles = TRUE)
```

aop_methods *Measurement methods used in an AOP*

Description

Methods described for measuring the Key Events of an AOP.

Usage

```
aop_methods(aop_id)
```

Arguments

aop_id	AOP identifier.

Value

A tibble of measurement methods.

Examples

```
aop_methods(1)
```

aop_plot_components *Plot the Key Event Components of an AOP*

Description

Shows, for every Key Event of an AOP, the biological process, the biological object and the action that describe it, together with their ontology identifiers. Key Events are ordered from the molecular initiating event to the adverse outcome, and those two are marked in their titles, coloured green and red when the marquee package is installed. Components that AOP-Wiki does not record are left blank.

Usage

```
aop_plot_components(aop_id, wrap = 26)
```

Arguments

aop_id	AOP identifier.
wrap	Width in characters at which long terms are wrapped.

Value

A ggplot object.

Examples

```
aop_plot_components(38)
```

`aop_plot_genes` *Plot the genes of an AOP per Key Event*

Description

Draws a bar chart of the distinct genes linked to each Key Event of an AOP, coloured by detection method. Every Key Event of the AOP is shown, so a Key Event without gene links appears with an empty bar. The molecular initiating event and the adverse outcome are marked in the Key Event titles, and coloured green and red when the marquee package is installed. Requires the ggplot2 package.

Usage

```
aop_plot_genes(aop_id)
```

Arguments

`aop_id`	AOP identifier.

Value

A ggplot object, which can be further modified in the usual way.

Examples

```
aop_plot_genes(38)
```

`aop_search` *Search across AOP-Wiki entities*

Description

Free-text search over AOPs, Key Events, KERs, stressors and chemicals at once.

Usage

```
aop_search(q, types = NULL, limit = 100)
```

Arguments

`q`	Search term (1 to 200 characters).
`types`	Restrict to certain entity types, e.g. "aop" or "key_event".
`limit`	Maximum number of results.

Value

A tibble of matches, with a `type` column and a relevance `score`.

Examples

```
aop_search("liver")
```

---

aop_stressors *Stressors of an AOP*

Description

Stressors of an AOP

Usage

```
aop_stressors(aop_id)
```

Arguments

`aop_id`	AOP identifier.

Value

A tibble of stressors.

Examples

```
aop_stressors(1)
```

---

aop_summary *Overview of an AOP*

Description

Returns the Key Events of an AOP in causal order, from the molecular initiating event to the adverse outcome, with the biological level of organisation and the role of each event. The title of the AOP, its OECD status, its taxonomic applicability, its stressors and the version of the data are attached as attributes and shown when the result is printed.

Usage

```
aop_summary(aop_id)
```

Arguments

`aop_id`	AOP identifier.

Details

The order is computed from the Key Event Relationships, since the Key Events themselves are not stored in sequence. An AOP that branches has no single chain, so the order is then one valid sequence among several, and the printed output says so.

Value

A tibble with the columns `step`, `ke_id`, `ke_title`, `level` and `role`, of class `aop_summary`.

Examples

```
aop_summary(38)
```

`chemical_aops` *AOPs linked to a chemical*

Description

AOPs linked to a chemical

Usage

```
chemical_aops(cas)
```

Arguments

`cas`	CAS registry number, e.g. "56-23-5".

Value

A tibble of AOPs.

Examples

```
chemical_aops("56-23-5")
```

chemical_get *Get a single chemical*

Description

Get a single chemical

Usage

```
chemical_get(cas)
```

Arguments

cas	CAS registry number, e.g. "56-23-5".

Value

A tibble with one row.

Examples

```
chemical_get("56-23-5")
```

chemical_key_events *Key Events linked to a chemical*

Description

Key Events linked to a chemical

Usage

```
chemical_key_events(cas, role = NULL)
```

Arguments

cas	CAS registry number, e.g. "56-23-5".
role	Optional filter: one of "mie", "ke" or "ao".

Value

A tibble of Key Events.

Examples

```
chemical_key_events("56-23-5")
```

`chemical_list` *List or search chemicals*

Description

List or search chemicals

Usage

```
chemical_list(
  q = NULL,
  aop = NULL,
  stressor = NULL,
  inchikey = NULL,
  chebi = NULL,
  pubchem = NULL,
  chemspider = NULL,
  wikidata = NULL,
  chembl = NULL,
  drugbank = NULL,
  kegg = NULL,
  hmdb = NULL,
  lipidmaps = NULL
)
```

Arguments

`q`	Free-text search term.
`aop`	Only chemicals used in this AOP id.
`stressor`	Only chemicals linked to this stressor id.
`inchikey`, `chebi`, `pubchem`, `chemspider`, `wikidata`	Identifier filters.
`chembl`, `drugbank`, `kegg`, `hmdb`, `lipidmaps`	Further identifier filters. Look the chemical up by an identifier in that database.

Value

A tibble with one row per chemical.

Examples

```
chemical_list(q = "benzene")
```

chemical_pathways *WikiPathways containing a chemical*

Description

WikiPathways containing a chemical

Usage

```
chemical_pathways(cas)
```

Arguments

cas	CAS registry number, e.g. "56-23-5".

Value

A tibble of pathways.

Examples

```
chemical_pathways("56-23-5")
```

gene_get *Get a single gene*

Description

Get a single gene

Usage

```
gene_get(gene)
```

Arguments

gene	Gene symbol or HGNC identifier, e.g. "TP53".

Value

A tibble with one row.

Examples

```
gene_get("TP53")
```

gene_list *List genes*

Description

List genes

Usage

```
gene_list(q = NULL, aop = NULL, ke = NULL, method = NULL)
```

Arguments

q	Free-text search term, e.g. a gene symbol.
aop	Only genes linked to this AOP id.
ke	Only genes linked to this Key Event id.
method	How genes were linked: "any", "regex", "ner" or "both".

Value

A tibble with one row per gene.

Examples

```
gene_list(q = "TP53")
gene_list(aop = 37)
```

ke_adverse_outcomes *Adverse Outcomes linked to a Key Event*

Description

Adverse Outcomes linked to a Key Event

Usage

```
ke_adverse_outcomes(ke_id, as = NULL)
```

Arguments

ke_id	Key Event identifier.
as	Which AOPs to consider: "mie" (only AOPs where this KE is the MIE) or "any" (every AOP containing it).

Value

A tibble of Adverse Outcomes.

Examples

```
ke_adverse_outcomes(334)
```

---

ke_aops *AOPs containing a Key Event*

Description

AOPs containing a Key Event

Usage

```
ke_aops(ke_id, role = NULL)
```

Arguments

ke_id	Key Event identifier.
role	Optional filter: one of "mie", "ke" or "ao".

Value

A tibble of AOPs.

Examples

```
ke_aops(334)
```

---

ke_chemicals *Chemicals of a Key Event*

Description

Chemicals of a Key Event

Usage

```
ke_chemicals(ke_id)
```

Arguments

ke_id	Key Event identifier.

Value

A tibble of chemicals.

Examples

```
ke_chemicals(334)
```

---

ke_components *Key Event Components of a Key Event*

Description

Key Event Components describe a Key Event in ontology terms: a biological process, an optional biological object, and the action that links them. The API returns them nested inside a single Key Event record, and this function returns them as one row per component.

Usage

```
ke_components(ke_id)
```

Arguments

ke_id	Key Event identifier.

Value

A tibble with the columns `ke_id`, `action`, `process`, `process_id`, `object` and `object_id`. The identifiers are ontology terms, for example `"GO:0008219"`. Components that name no object have `NA` in `object` and `object_id`.

Examples

```
ke_components(55)
```

---

ke_downstream *Key Events downstream of a Key Event*

Description

Follows Key Event Relationships forwards.

Usage

```
ke_downstream(ke_id, depth = 1)
```

Arguments

ke_id	Key Event identifier.
depth	How many relationships to follow (1 to 3, default 1).

Value

A tibble of Key Events.

Examples

```
ke_downstream(334, depth = 2)
```

---

ke_gene_sets *Key Event gene sets of an AOP as a GeneSetCollection*

Description

Returns the genes linked to each Key Event of an AOP as a GSEABase::GeneSetCollection, with one gene set per Key Event. This is the form that Bioconductor enrichment tools expect, so the result can be passed straight to functions such as `GSVA::gsva()` or `limma::camera()` without reshaping.

Usage

```
ke_gene_sets(aop_id, method = NULL)
```

Arguments

aop_id	AOP identifier.
method	How genes were linked: one of "any", "regex", "ner" or "both".

Details

Each gene set is named `KE_<id>`, carries the Key Event title as its short description, and records the Key Event role and the version of the AOP-Wiki data in its long description. Key Events without gene links are omitted, since an empty gene set is not useful for enrichment. Use `aop_gene_summary()` to see which Key Events those are.

Value

A `GeneSetCollection` with one `GeneSet` per Key Event that has linked genes.

See Also

`aop_genes()` for the same information as a tibble.

Examples

```
gsc <- ke_gene_sets(38)
names(gsc)
```

ke_get *Get a single Key Event*

Description

Get a single Key Event

Usage

```
ke_get(ke_id)
```

Arguments

ke_id	Key Event identifier.

Value

A tibble with one row.

Examples

```
ke_get(334)
```

ke_list *List Key Events*

Description

List Key Events

Usage

```
ke_list(
  q = NULL,
  role = NULL,
  level = NULL,
  aop = NULL,
  taxon = NULL,
  organ = NULL,
  cell_type = NULL,
  gene = NULL
)
```

Arguments

q	Free-text search term.
role	Optional filter: one of "mie", "ke" or "ao".
level	Biological level of organisation.
aop	Only Key Events belonging to this AOP id.
taxon, organ, cell_type	Biological context filters.
gene	Only Key Events linked to this gene.

Value

A tibble with one row per Key Event.

Examples

```
ke_list(q = "apoptosis")
ke_list(aop = 1)
```

---

ke_mies *Molecular Initiating Events linked to a Key Event*

Description

Molecular Initiating Events linked to a Key Event

Usage

```
ke_mies(ke_id, as = NULL)
```

Arguments

ke_id	Key Event identifier.
as	Which AOPs to consider: "ao" (only AOPs where this KE is the AO) or "any" (every AOP containing it).

Value

A tibble of MIEs.

Examples

```
ke_mies(334)
```

ke_stressors *Stressors of a Key Event*

Description

Stressors of a Key Event

Usage

```
ke_stressors(ke_id)
```

Arguments

ke_id	Key Event identifier.

Value

A tibble of stressors.

Examples

```
ke_stressors(334)
```

ke_upstream *Key Events upstream of a Key Event*

Description

Follows Key Event Relationships backwards.

Usage

```
ke_upstream(ke_id, depth = 1)
```

Arguments

ke_id	Key Event identifier.
depth	How many relationships to follow (1 to 3, default 1).

Value

A tibble of Key Events.

Examples

```
ke_upstream(334, depth = 2)
```

ker_get *Get a single Key Event Relationship*

Description

Get a single Key Event Relationship

Usage

```
ker_get(ker_id)
```

Arguments

ker_id	KER identifier.

Value

A tibble with one row.

Examples

```
ker_get(158)
```

ker_list *List Key Event Relationships*

Description

List Key Event Relationships

Usage

```
ker_list(
  aop = NULL,
  upstream = NULL,
  downstream = NULL,
  ke = NULL,
  gene = NULL
)
```

Arguments

aop	Only KERs belonging to this AOP id.
upstream	Only KERs starting at this Key Event id.
downstream	Only KERs ending at this Key Event id.
ke	Only KERs involving this Key Event id, in either position.
gene	Only KERs linked to this gene.

Value

A tibble with one row per KER.

Examples

```
ker_list(aop = 1)
ker_list(ke = 334)
```

`mie_list` *List Molecular Initiating Events*

Description

List Molecular Initiating Events

Usage

```
mie_list(
  q = NULL,
  level = NULL,
  aop = NULL,
  taxon = NULL,
  organ = NULL,
  cell_type = NULL,
  gene = NULL
)
```

Arguments

`q`	Free-text search term.
`level`	Biological level of organisation.
`aop`	Only Key Events belonging to this AOP id.
`taxon`, `organ`, `cell_type`	Biological context filters.
`gene`	Only Key Events linked to this gene.

Value

A tibble with one row per MIE.

Examples

```
mie_list(q = "receptor")
```

stressor_aops *AOPs linked to a stressor, with their adverse outcomes*

Description

Lists every AOP in which a stressor is used, together with the adverse outcomes those AOPs end in. The AOP identifiers come from the `aop_ids` field of the stressor record, and the titles and adverse outcomes are taken from the AOP list, which is cached for the session. The stressor may be given as an identifier or as a name, in which case the name is looked up with `stressor_list()`.

Usage

```
stressor_aops(stressor)
```

Arguments

`stressor`	Stressor identifier, or a stressor name.

Value

A tibble with the columns `stressor_id`, `stressor_title`, `aop_id`, `aop_title`, `ao_id` and `ao_title`, with one row per AOP and adverse outcome. An AOP with no adverse outcome recorded keeps one row, with `NA` in the two `ao` columns. A stressor used in no AOP gives zero rows.

Examples

```
stressor_aops(150)
stressor_aops("hexachlorobenzene")
```

stressor_get *Get a single stressor*

Description

Get a single stressor

Usage

```
stressor_get(stressor_id)
```

Arguments

`stressor_id`	Stressor identifier.

Value

A tibble with one row.

Examples

```
stressor_get(150)
```

`stressor_list` *List stressors*

Description

List stressors

Usage

```
stressor_list(q = NULL, aop = NULL, chemical = NULL, has_chemical = NULL)
```

Arguments

`q`	Free-text search term.
`aop`	Only stressors used in this AOP id.
`chemical`	Only stressors linked to this chemical (CAS number).
`has_chemical`	If TRUE, only stressors that have a chemical mapping.

Value

A tibble with one row per stressor.

Examples

```
stressor_list(q = "benzene")
stressor_list(aop = 1)
```

Index

ao_list, 3
aop_cache_clear, 4
aop_chemicals, 4
aop_components, 5
aop_data_version, 5
aop_dataset_info, 6
aop_gene_summary, 6
aop_gene_summary(), *20*
aop_genes, 7
aop_genes(), *20*
aop_get, 7
aop_kers, 8
aop_key_events, 8
aop_list, 9
aop_methods, 10
aop_plot_components, 10
aop_plot_genes, 11
aop_search, 11
aop_stressors, 12
aop_summary, 12

chemical_aops, 13
chemical_get, 14
chemical_key_events, 14
chemical_list, 15
chemical_pathways, 16

gene_get, 16
gene_list, 17
GSEABase::GeneSetCollection, *20*

ke_adverse_outcomes, 17
ke_aops, 18
ke_chemicals, 18
ke_components, 19
ke_components(), *5*
ke_downstream, 19
ke_gene_sets, 20
ke_get, 21
ke_list, 21
ke_mies, 22
ke_stressors, 23
ke_upstream, 23
ker_get, 24
ker_list, 24

mie_list, 25

stressor_aops, 26
stressor_get, 26
stressor_list, 27
stressor_list(), *26*