

Package: raqs (via r-universe)

September 13, 2024

Title Interface to the US EPA Air Quality System (AQS) API

Version 1.0.2

Description Offers functions for fetching JSON data from the US EPA Air Quality System (AQS) API with options to comply with the API rate limits. See https://aqs.epa.gov/aqsweb/documents/data_api.html for details of the AQS API.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports cli, httr2

Depends R (>= 4.1)

URL <https://github.com/HimesGroup/raqs>

BugReports <https://github.com/HimesGroup/raqs/issues>

Suggests data.table, tibble

Repository <https://himesgroup.r-universe.dev>

RemoteUrl <https://github.com/himesgroup/raqs>

RemoteRef HEAD

RemoteSha 44cbb82b2a6642bf34f88d085896758d6435d78b

Contents

| | |
|-------------------------------------|----|
| raqs-package | 2 |
| aq_s_annualdata | 3 |
| aq_s_dailydata | 7 |
| aq_s_list | 11 |
| aq_s_metadata | 14 |
| aq_s_monitors | 16 |
| aq_s_qaannualperformanceevaluations | 19 |

| | |
|--|----|
| aqs_qablanks | 23 |
| aqs_qacollocatedassessments | 26 |
| aqs_qaflowrateaudits | 29 |
| aqs_qaflowrateverifications | 32 |
| aqs_qaonepointqcrowdata | 36 |
| aqs_qapepaudits | 39 |
| aqs_quarterlydata | 42 |
| aqs_sampledata | 46 |
| aqs_signup | 50 |
| aqs_transactionsqaannualperformanceevaluations | 51 |
| aqs_transactionssample | 54 |
| raqs_options | 57 |
| set_aqs_user | 58 |

| | |
|--------------|-----------|
| Index | 60 |
|--------------|-----------|

| | |
|--------------|---|
| raqs-package | <i>raqs: Interface to the US EPA Air Quality System (AQS) API</i> |
|--------------|---|

Description

Offers functions for fetching JSON data from the US EPA Air Quality System (AQS) API with options to comply with the API rate limits. See https://aqs.epa.gov/aqsweb/documents/data_api.html for details of the AQS API.

Details

The 'raqs' package provides an R interface to the US EPA AQS API that publish data in JSON format. To use this package, you first need to register for the AQS API and get your API key. Please check [aqs_signup](#) and [set_aqs_user](#) to set up your API credentials in R.

All main functions, for fetching data from the AQS API, were named with the following scheme: aqs_{service}

- [aqs_metadata](#) returns information about the API.
- [aqs_list](#) returns variable values you may need to create other service requests.
- [aqs_monitors](#) returns operational information about the monitors used to collect data.
- [aqs_sampledata](#) returns sample data - the finest grain data reported to EPA.
- [aqs_dailydata](#) returns data summarized at the daily level.
- [aqs_quarterlydata](#) returns data summarized at the calendar quarter level.
- [aqs_annualdata](#) returns data summarized at the yearly level
- [aqs_qaannualperformanceevaluations](#) returns pairs of data (known and measured values) at several concentration levels for gaseous criteria pollutants.
- [aqs_qablanks](#) returns concentrations from blank samples.
- [aqs_qacollocatedassessments](#) returns pairs of PM samples collected at the same time and place by different samplers.

- [aqs_qaflowrateverifications](#) returns flow rate checks performed by monitoring agencies.
- [aqs_qaflowrateaudits](#) returns flow rate audits data
- [aqs_qaonepointqcrawdata](#) returns measured versus actual concentration of one point QC checks.
- [aqs_qapepaudits](#) returns data related to PM2.5 monitoring system audits.
- [aqs_transactionssample](#) returns sample data in the transaction format for AQS.
- [aqs_transactionsqaannualperformanceevaluations](#) returns pairs of data QA at several concentration levels in the transaction format for AQS.

Each main function has a set of underlying functions that are responsible for sending requests to specific endpoints (service/filter) and were named with the following scheme: {service}_{filter}. Please refer to the manual to see how the aforementioned functions work.

Author(s)

Maintainer: Jaehyun Joo <jaehyunjoo@outlook.com>

Authors:

- Blanca Himes

See Also

Useful links:

- <https://github.com/HimesGroup/raqs>
- Report bugs at <https://github.com/HimesGroup/raqs/issues>

aqs_annualdata

AQS API Annual Summary Data service

Description

A collection of functions to fetch data summarized at the yearly level. Note that only the year portions of the bdate and edate are used and only whole years of data are returned.

Usage

```
aqs_annualdata(  
  aqs_filter = c("bySite", "byCounty", "byState", "byBox", "byCBSA"),  
  aqs_variables = NULL,  
  header = FALSE,  
  ...  
)  
  
annualdata_bysite(  
  param,  
  bdate,
```

```
    edate,  
    state,  
    county,  
    site,  
    email = get_aqs_email(),  
    key = get_aqs_key(),  
    duration = NULL,  
    cbdate = NULL,  
    cedate = NULL,  
    header = FALSE,  
    ...  
)  
  
annualdata_bycounty(  
    param,  
    bdate,  
    edate,  
    state,  
    county,  
    email = get_aqs_email(),  
    key = get_aqs_key(),  
    duration = NULL,  
    cbdate = NULL,  
    cedate = NULL,  
    header = FALSE,  
    ...  
)  
  
annualdata_bystate(  
    param,  
    bdate,  
    edate,  
    state,  
    email = get_aqs_email(),  
    key = get_aqs_key(),  
    duration = NULL,  
    cbdate = NULL,  
    cedate = NULL,  
    header = FALSE,  
    ...  
)  
  
annualdata_bybox(  
    param,  
    bdate,  
    edate,  
    minlat,  
    maxlat,
```

```

    minlon,
    maxlon,
    email = get_aqs_email(),
    key = get_aqs_key(),
    duration = NULL,
    cbdate = NULL,
    cedate = NULL,
    header = FALSE,
    ...
)

annualdata_bycbsa(
  param,
  bdate,
  edate,
  cbsa,
  email = get_aqs_email(),
  key = get_aqs_key(),
  duration = NULL,
  cbdate = NULL,
  cedate = NULL,
  header = FALSE,
  ...
)

```

Arguments

| | |
|----------------------------|--|
| aq _s _filter | A string specifying one of the service filters. NOT case-sensitive. |
| aq _s _variables | A named list of variables to fetch data (e.g., state). Only necessary variables are passed to a specific endpoint (service/filter) to make a valid request. |
| header | A logical specifying whether the function returns additional information from the API header. Default is FALSE to return data only. |
| ... | Reserved for future use. |
| param | A string or vector of strings specifying the 5-digit AQS parameter code for data selection. An integer will be coerced to a string. A maximum of 5 parameter codes may be listed in a single request. A list of the parameter codes can be obtained via list_parametersbyclass . |
| bdate | A string specifying the begin date of data selection in YYYYMMDD format. Only the year portion is used. |
| edate | A string specifying the end date of data selection in YYYYMMDD format. Only the year portion is used. If the end date is not in the same year as the begin date, the function will automatically split the date range into multiple chunks by year and send requests sequentially. |
| state | A string specifying the 2-digit state FIPS code. An integer will be coerced to a string with a leading zero if necessary (e.g., 1 -> "01"). A list of the state codes can be obtained via list_states . |

| | |
|----------|--|
| county | A string specifying the 3-digit county FIPS code. An integer will be coerced to a string with leading zeros if necessary (e.g., 89 -> "089"). A list of the county codes within each state can be obtained via list_countiesbystate . |
| site | A string specifying the 4-digit AQS site number within the county. An integer will be coerced to a string with leading zeros if necessary (e.g., 14 -> "0014"). A list of the site codes within each county can be obtained via list_sitesbycounty . |
| email | A string specifying the email address of the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| key | A string specifying the key matching the email address for the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| duration | (optional) A string specifying the 1-character AQS sample duration code. A list of the duration codes can be obtained via list_durations . Only data reported at this sample duration will be returned. |
| cbdate | (optional) A string specifying the change begin date in YYYYMMDD format to subset data based on "date of last change" in database. Only data that changed on or after this date will be returned. |
| cedate | (optional) A string specifying the change end date in YYYYMMDD format to subset data based on "date of last change" in database. Only data that changed on or before this date will be returned. |
| minlat | A string or numeric value specifying the minimum latitude of a geographic box. Decimal latitude with north being positive. |
| maxlat | A string or numeric value specifying the maximum latitude of a geographic box. Decimal latitude with north being positive. |
| minlon | A string or numeric value specifying the minimum longitude of a geographic box. Decimal longitude with east being positive. |
| maxlon | A string or numeric value specifying the maximum longitude of a geographic box. Decimal longitude with east being positive. |
| cbsa | A string specifying the AQS CBSA code. A list of the CBSA codes can be obtained via list_cbsas . |

Details

[aq_s_annualdata](#) sends a request to the AQS API based on a user-provided filter using the following underlying functions:

- [annualdata_bysite](#) returns annual summary param data for site in county, within state, based on the year portions of bdate and edate.
- [annualdata_bycounty](#) returns annual summary param data for county in state based on the year portions of bdate and edate.
- [annualdata_bystate](#) returns annual summary param data for state based on the year portions of bdate and edate.
- [annualdata_bybox](#) returns annual summary param data for a user-provided latitude/longitude bounding box (minlat, maxlat, minlon, maxlon) based on the year portions of bdate and edate.
- [annualdata_bycbsa](#) returns annual summary param data for a user-provided CBSA based on the year portions of bdate and edate.

Value

A data.frame containing parsed data or a named list containing header and data.

Examples

```
## Not run:

## Set your API Key first using set_aqs_user to run the following codes

## Example from the AQS website
## FRM/FEM PM2.5 data for Wake County, NC for 2016
## Only the year portions of bdate and edate are used
aqs_variables <- list(
  param = c("88101", "88502"), bdate = "20160101", edate = "20160228",
  state = "37", county = "183"
)
aqs_annualdata(aqs_filter = "byCounty", aqs_variables = aqs_variables)

## Equivalent to above; used integers instead of strings
annualdata_bycounty(
  param = c(88101, 88502), bdate = "20160101", edate = "20160228",
  state = 37, county = 183
)

## End(Not run)
```

aq_s_dailydata

AQS API Daily Summary Data service

Description

A collection of functions to fetch data summarized at the daily level. Please use a narrow range of dates to adhere to the API's limit imposed on request size.

Usage

```
aqs_dailydata(
  aqs_filter = c("bySite", "byCounty", "byState", "byBox", "byCBSA"),
  aqs_variables = NULL,
  header = FALSE,
  ...
)

dailydata_bysite(
  param,
  bdate,
  edate,
```

```
state,  
county,  
site,  
email = get_aqs_email(),  
key = get_aqs_key(),  
duration = NULL,  
cbdate = NULL,  
cedate = NULL,  
header = FALSE,  
...  
)
```

```
dailydata_bycounty(  
  param,  
  bdate,  
  edate,  
  state,  
  county,  
  email = get_aqs_email(),  
  key = get_aqs_key(),  
  duration = NULL,  
  cbdate = NULL,  
  cedate = NULL,  
  header = FALSE,  
  ...  
)
```

```
dailydata_bystate(  
  param,  
  bdate,  
  edate,  
  state,  
  email = get_aqs_email(),  
  key = get_aqs_key(),  
  duration = NULL,  
  cbdate = NULL,  
  cedate = NULL,  
  header = FALSE,  
  ...  
)
```

```
dailydata_bybox(  
  param,  
  bdate,  
  edate,  
  minlat,  
  maxlat,  
  minlon,
```



```

    maxlon,
    email = get_aqs_email(),
    key = get_aqs_key(),
    duration = NULL,
    cbdate = NULL,
    cedate = NULL,
    header = FALSE,
    ...
)

dailydata_bycbsa(
  param,
  bdate,
  edate,
  cbsa,
  email = get_aqs_email(),
  key = get_aqs_key(),
  duration = NULL,
  cbdate = NULL,
  cedate = NULL,
  header = FALSE,
  ...
)

```

Arguments

| | |
|----------------------------|--|
| aq _s _filter | A string specifying one of the service filters. NOT case-sensitive. |
| aq _s _variables | A named list of variables to fetch data (e.g., state). Only necessary variables are passed to a specific endpoint (service/filter) to make a valid request. |
| header | A logical specifying whether the function returns additional information from the API header. Default is FALSE to return data only. |
| ... | Reserved for future use. |
| param | A string or vector of strings specifying the 5-digit AQS parameter code for data selection. An integer will be coerced to a string. A maximum of 5 parameter codes may be listed in a single request. A list of the parameter codes can be obtained via list_parametersbyclass . |
| bdate | A string specifying the begin date of data selection in YYYYMMDD format. Only data on or after this date will be returned. |
| edate | A string specifying the end date of data selection in YYYYMMDD format. Only data on or before this date will be returned. If the end date is not in the same year as the begin date, the function will automatically split the date range into multiple chunks by year and send requests sequentially. |
| state | A string specifying the 2-digit state FIPS code. An integer will be coerced to a string with a leading zero if necessary (e.g., 1 -> "01"). A list of the state codes can be obtained via list_states . |

| | |
|----------|--|
| county | A string specifying the 3-digit county FIPS code. An integer will be coerced to a string with leading zeros if necessary (e.g., 89 -> "089"). A list of the county codes within each state can be obtained via list_countiesbystate . |
| site | A string specifying the 4-digit AQS site number within the county. An integer will be coerced to a string with leading zeros if necessary (e.g., 14 -> "0014"). A list of the site codes within each county can be obtained via list_sitesbycounty . |
| email | A string specifying the email address of the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| key | A string specifying the key matching the email address for the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| duration | (optional) A string specifying the 1-character AQS sample duration code. A list of the duration codes can be obtained via list_durations . Only data reported at this sample duration will be returned. |
| cbdate | (optional) A string specifying the change begin date in YYYYMMDD format to subset data based on "date of last change" in database. Only data that changed on or after this date will be returned. |
| cedate | (optional) A string specifying the change end date in YYYYMMDD format to subset data based on "date of last change" in database. Only data that changed on or before this date will be returned. |
| minlat | A string or numeric value specifying the minimum latitude of a geographic box. Decimal latitude with north being positive. |
| maxlat | A string or numeric value specifying the maximum latitude of a geographic box. Decimal latitude with north being positive. |
| minlon | A string or numeric value specifying the minimum longitude of a geographic box. Decimal longitude with east being positive. |
| maxlon | A string or numeric value specifying the maximum longitude of a geographic box. Decimal longitude with east being positive. |
| cbsa | A string specifying the AQS CBSA code. A list of the CBSA codes can be obtained via list_cbsas . |

Details

[aq_s_dailydata](#) sends a request to the AQS API based on a user-provided filter using the following underlying functions:

- [dailydata_bysite](#) returns daily summary param data for site in county, within state, between bdate and edate.
- [dailydata_bycounty](#) returns daily summary param data for county in state between bdate and edate.
- [dailydata_bystate](#) returns daily summary param data for state between bdate and edate.
- [dailydata_bybox](#) returns daily summary param data for a user-provided latitude/longitude bounding box (minlat, maxlat, minlon, maxlon) between bdate and edate.
- [dailydata_bycbsa](#) returns daily summary param data for a user-provided CBSA between bdate and edate.

Value

A data.frame containing parsed data or a named list containing header and data.

Examples

```
## Not run:

## Set your API Key first using set_aqs_user to run the following codes

## Example from the AQS website
## FRM/FEM PM2.5 data for Wake County, NC between Jan and Feb 2016
aqs_variables <- list(
  param = "88101", bdate = "20160101", edate = "20160228",
  state = "37", county = "183"
)
aqs_dailydata(aqs_filter = "byCounty", aqs_variables = aqs_variables)

## Equivalent to above; used integers instead of strings
dailydata_bycounty(
  param = 88101, bdate = "20160101", edate = "20160228",
  state = 37, county = 183
)

## End(Not run)
```

aq_s_list

AQS API List service

Description

A collection of functions to fetch variable values you need to create other service requests. All outputs are a value and the definition of that value.

Usage

```
aqs_list(
  aqs_filter = c("states", "countiesByState", "sitesByCounty", "cbsas", "classes",
    "parametersByClass", "pqaos", "mas", "durations"),
  aqs_variables = NULL,
  header = FALSE,
  ...
)

list_states(email = get_aqs_email(), key = get_aqs_key(), header = FALSE, ...)

list_countiesbystate(
  state,
```

```

    email = get_aqs_email(),
    key = get_aqs_key(),
    header = FALSE,
    ...
)

list_sitesbycounty(
  state,
  county,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

list_cbsas(email = get_aqs_email(), key = get_aqs_key(), header = FALSE, ...)

list_classes(email = get_aqs_email(), key = get_aqs_key(), header = FALSE, ...)

list_parametersbyclass(
  pc,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

list_pqaos(email = get_aqs_email(), key = get_aqs_key(), header = FALSE, ...)

list_mas(email = get_aqs_email(), key = get_aqs_key(), header = FALSE, ...)

list_durations(
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

```

Arguments

| | |
|----------------------------|---|
| aq _s _filter | A string specifying one of the service filters. NOT case-sensitive. |
| aq _s _variables | A named list of variables to fetch data (e.g., state). Only necessary variables are passed to a specific endpoint (service/filter) to make a valid request. |
| header | A logical specifying whether the function returns additional information from the API header. Default is FALSE to return data only. |
| ... | Reserved for future use. |
| email | A string specifying the email address of the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |

| | |
|--------|--|
| key | A string specifying the key matching the email address for the requester. If you set your email and key with <code>set_aqs_user</code> , you don't have to specify this. |
| state | A string specifying the 2-digit state FIPS code. An integer will be coerced to a string with a leading zero if necessary (e.g., 1 -> "01"). A list of the state codes can be obtained via <code>list_states</code> . |
| county | A string specifying the 3-digit county FIPS code. An integer will be coerced to a string with leading zeros if necessary (e.g., 89 -> "089"). A list of the county codes within each state can be obtained via <code>list_countiesbystate</code> . |
| pc | A string specifying the AQS parameter class name. A list of the class names can be obtained via <code>list_classes</code> . |

Details

`aqs_list` sends a request to the AQS API based on a user-provided filter using the following underlying functions:

- `list_states` returns a list of the states and their FIPS codes.
- `list_countiesbystate` returns a list of all counties within a user-provided state.
- `list_sitesbycounty` returns a list of all sites within a user-provided county.
- `list_cbsas` returns a list of the 5-digit Core Based Statistical Area (CBSA) codes.
- `list_classes` returns a list of parameter class codes.
- `list_parametersbyclass` returns all parameters in a user-provided parameter class.
- `list_pqaos` returns a list of AQS Primary Quality Assurance Organization (PQAO) codes.
- `list_mas` returns a list of AQS Monitoring Agency (MA) codes.
- `list_durations` returns a list of the 1-character AQS sample duration codes.

Value

A `data.frame` containing parsed data or a named list containing header and data.

Examples

```
## Not run:

## Set your API Key first using set_aqs_user to run the following codes

aqs_list(aqs_filter = "states")
list_states() # equivalent to above

aqs_list("countiesByState", aqs_variables = list(state = "01"))
list_countiesbystate(state = "01")

aqs_list("sitesByCounty", aqs_variables = list(state = "37", county = "183"))
list_sitesbycounty(state = "37", county = "183")

aqs_list("cbsas")
list_cbsas()
```

```
aqs_list("classes")
list_classes()

aqs_list("parametersByClass", list(pc = "CRITERIA")) # Criteria pollutants
list_parametersbyclass(pc = "CRITERIA")

aqs_list("pqaos")
list_pqaos()

aqs_list("mas")
list_mas()

aqs_list("durations")
list_durations()

## End(Not run)
```

aq_s_metadata

AQ_S API Meta Data service

Description

A collection of functions to fetch information about the AQ_S API. The main purpose of this service is to let you know the system is up before you run a long job.

Usage

```
aqs_metadata(
  aqs_filter = c("isAvailable", "revisionHistory", "fieldsByService", "issues"),
  aqs_variables = NULL,
  header = FALSE,
  ...
)

metadata_isavailable(...)

metadata_revisionhistory(
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

metadata_fieldsbyservice(
  service,
  email = get_aqs_email(),
```

```

    key = get_aqs_key(),
    header = FALSE,
    ...
)

metadata_issues(
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

```

Arguments

| | |
|----------------------------|---|
| aq _s _filter | A string specifying one of the service filters. NOT case-sensitive. |
| aq _s _variables | A named list of variables to fetch data (e.g., state). Only necessary variables are passed to a specific endpoint (service/filter) to make a valid request. |
| header | A logical specifying whether the function returns additional information from the API header. Default is FALSE to return data only. |
| ... | Reserved for future use. |
| email | A string specifying the email address of the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| key | A string specifying the key matching the email address for the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| service | A string specifying one of the services available (e.g., sampleData) |

Details

[aq_s_metadata](#) sends a request to the AQS API based on a user-provided filter using the following underlying functions:

- [metadata_isavailable](#) checks if the API is up and running.
- [metadata_revisionhistory](#) returns a complete list of revisions to the API in reverse chronological order.
- [metadata_fieldsbyservice](#) returns a list and definitions of fields in a user-provided service.
- [metadata_issues](#) returns a list of any known issues with system functionality or the data.

Value

A data.frame containing parsed data or a named list containing header and data.

Examples

```

## Not run:

## Set your API Key first using set_aqs_user to run the following codes

```

```

aqs_metadata(aqs_filter = "isAvailable")
metadata_isavailable() # equivalent to above

aqs_metadata("revisionHistory")
metadata_revisionhistory()

aqs_metadata("fieldsByService", aqs_variables = list(service = "annualData"))
metadata_fieldsbyservice(service = "annualData")

aqs_metadata("issues")
metadata_issues()

## End(Not run)

```

aq_s_monitors

AQ_S API Monitors service

Description

A collection of functions to fetch operational information about the samplers (monitors) used to collect data, including identifying information, operational dates, operating organizations, and etc.

Usage

```

aqs_monitors(
  aqs_filter = c("bySite", "byCounty", "byState", "byBox", "byCBSA"),
  aqs_variables = NULL,
  header = FALSE,
  ...
)

monitors_bysite(
  param,
  bdate,
  edate,
  state,
  county,
  site,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

monitors_bycounty(
  param,
  bdate,

```



```
    edate,  
    state,  
    county,  
    email = get_aqs_email(),  
    key = get_aqs_key(),  
    header = FALSE,  
    ...  
)  
  
monitors_bystate(  
    param,  
    bdate,  
    edate,  
    state,  
    email = get_aqs_email(),  
    key = get_aqs_key(),  
    header = FALSE,  
    ...  
)  
  
monitors_bybox(  
    param,  
    bdate,  
    edate,  
    minlat,  
    maxlat,  
    minlon,  
    maxlon,  
    email = get_aqs_email(),  
    key = get_aqs_key(),  
    header = FALSE,  
    ...  
)  
  
monitors_bycbsa(  
    param,  
    bdate,  
    edate,  
    cbsa,  
    email = get_aqs_email(),  
    key = get_aqs_key(),  
    header = FALSE,  
    ...  
)
```

Arguments

`aqs_filter` A string specifying one of the service filters. NOT case-sensitive.

| | |
|----------------------------|--|
| aq _s _variables | A named list of variables to fetch data (e.g., state). Only necessary variables are passed to a specific endpoint (service/filter) to make a valid request. |
| header | A logical specifying whether the function returns additional information from the API header. Default is FALSE to return data only. |
| ... | Reserved for future use. |
| param | A string or vector of strings specifying the 5-digit AQS parameter code for data selection. An integer will be coerced to a string. A maximum of 5 parameter codes may be listed in a single request. A list of the parameter codes can be obtained via list_parametersbyclass . |
| bdate | A string specifying the begin date of data selection in YYYYMMDD format. Only data on or after this date will be returned. |
| edate | A string specifying the end date of data selection in YYYYMMDD format. Only data on or before this date will be returned. |
| state | A string specifying the 2-digit state FIPS code. An integer will be coerced to a string with a leading zero if necessary (e.g., 1 -> "01"). A list of the state codes can be obtained via list_states . |
| county | A string specifying the 3-digit county FIPS code. An integer will be coerced to a string with leading zeros if necessary (e.g., 89 -> "089"). A list of the county codes within each state can be obtained via list_countiesbystate . |
| site | A string specifying the 4-digit AQS site number within the county. An integer will be coerced to a string with leading zeros if necessary (e.g., 14 -> "0014"). A list of the site codes within each county can be obtained via list_sitesbycounty . |
| email | A string specifying the email address of the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| key | A string specifying the key matching the email address for the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| minlat | A string or numeric value specifying the minimum latitude of a geographic box. Decimal latitude with north being positive. |
| maxlat | A string or numeric value specifying the maximum latitude of a geographic box. Decimal latitude with north being positive. |
| minlon | A string or numeric value specifying the minimum longitude of a geographic box. Decimal longitude with east being positive. |
| maxlon | A string or numeric value specifying the maximum longitude of a geographic box. Decimal longitude with east being positive. |
| cbsa | A string specifying the AQS CBSA code. A list of the CBSA codes can be obtained via list_cbsas . |

Details

[aq_s_monitors](#) sends a request to the AQS API based on a user-provided filter using the following underlying functions:

- [monitors_bysite](#) returns param monitors that were operating at site in county, within state, between bdate and edate.

- `monitors_bycounty` returns param monitors that were operating in county within state between bdate and edate.
- `monitors_bystate` returns param monitors that were operating in state between bdate and edate.
- `monitors_bybox` returns param monitors that were operating at a user-provided latitude/longitude bounding box (minlat, maxlat, minlon, maxlon) between bdate and edate.
- `monitors_bycbsa` returns param monitors that were operating at a user-provided CBSA between bdate and edate.

Value

A data.frame containing parsed data or a named list containing header and data.

Examples

```
## Not run:

## Set your API Key first using set_aqs_user to run the following codes

## Example from the AQS website
## SO2 monitors in Hawaii that were operating on May 01, 2015
aqs_variables <- list(
  param = "42401", bdate = "20150501", edate = "20150502", state = "15"
)
aqs_monitors(aqs_filter = "bySite", aqs_variables = aqs_variables)

## Equivalent to above; used integers instead of strings
monitors_bystate(
  param = 42401, bdate = "20150501", edate = "20150502", state = 15
)

## End(Not run)
```

aq_s_qaannualperformanceevaluations

AQS API QA Annual Performance Evaluations service

Description

A collection of functions to fetch pairs of data (known and measured values) at several concentration levels for gaseous criteria pollutants.

Usage

```
aqs_qaannualperformanceevaluations(
  aqs_filter = c("bySite", "byCounty", "byState", "byPQA0", "byMA"),
  aqs_variables = NULL,
  header = FALSE,
```

```
    ...
)

qaannualperformanceevaluations_bysite(
    param,
    bdate,
    edate,
    state,
    county,
    site,
    email = get_aqs_email(),
    key = get_aqs_key(),
    header = FALSE,
    ...
)

qaannualperformanceevaluations_bycounty(
    param,
    bdate,
    edate,
    state,
    county,
    email = get_aqs_email(),
    key = get_aqs_key(),
    header = FALSE,
    ...
)

qaannualperformanceevaluations_bystate(
    param,
    bdate,
    edate,
    state,
    email = get_aqs_email(),
    key = get_aqs_key(),
    header = FALSE,
    ...
)

qaannualperformanceevaluations_bypqao(
    param,
    bdate,
    edate,
    pqao,
    email = get_aqs_email(),
    key = get_aqs_key(),
    header = FALSE,
    ...
)
```

```

)

qaannualperformanceevaluations_byma(
  param,
  bdate,
  edate,
  agency,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

```

Arguments

| | |
|----------------------------|--|
| aq _s _filter | A string specifying one of the service filters. NOT case-sensitive. |
| aq _s _variables | A named list of variables to fetch data (e.g., state). Only necessary variables are passed to a specific endpoint (service/filter) to make a valid request. |
| header | A logical specifying whether the function returns additional information from the API header. Default is FALSE to return data only. |
| ... | Reserved for future use. |
| param | A string or vector of strings specifying the 5-digit AQS parameter code for data selection. An integer will be coerced to a string. A maximum of 5 parameter codes may be listed in a single request. A list of the parameter codes can be obtained via list_parametersbyclass . |
| bdate | A string specifying the begin date of data selection in YYYYMMDD format. Only data on or after this date will be returned. |
| edate | A string specifying the end date of data selection in YYYYMMDD format. Only data on or before this date will be returned. If the end date is not in the same year as the begin date, the function will automatically split the date range into multiple chunks by year and send requests sequentially. |
| state | A string specifying the 2-digit state FIPS code. An integer will be coerced to a string with a leading zero if necessary (e.g., 1 -> "01"). A list of the state codes can be obtained via list_states . |
| county | A string specifying the 3-digit county FIPS code. An integer will be coerced to a string with leading zeros if necessary (e.g., 89 -> "089"). A list of the county codes within each state can be obtained via list_countiesbystate . |
| site | A string specifying the 4-digit AQS site number within the county. An integer will be coerced to a string with leading zeros if necessary (e.g., 14 -> "0014"). A list of the site codes within each county can be obtained via list_sitesbycounty . |
| email | A string specifying the email address of the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| key | A string specifying the key matching the email address for the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| pqao | A string specifying the AQS Primary Quality Assurance Organization (PQAO) code. A list of the PQAO codes can be obtained via list_pqaos . |

agency A string specifying the AQS Monitoring Agency (MA) code. A list of the MA codes can be obtained via [list_mas](#). Here, we named this input as agency instead of "ma" because agency is actually used in the API endpoint URL.

Details

[aqs_qaannualperformanceevaluations](#) sends a request to the AQS API based on a user-provided filter using the following underlying functions:

- [qaannualperformanceevaluations_bysite](#) returns annual performance evaluation data for param at site in county, within state, between bdate and edate.
- [qaannualperformanceevaluations_bycounty](#) returns annual performance evaluation data for param in county within state between bdate and edate.
- [qaannualperformanceevaluations_bystate](#) returns annual performance evaluation data for param in state between bdate and edate.
- [qaannualperformanceevaluations_bypqao](#) returns annual performance evaluation data for param in pqao between bdate and edate.
- [qaannualperformanceevaluations_byma](#) returns annual performance evaluation data for param in agency (monitoring agency) between bdate and edate.

Value

A data.frame containing parsed data or a named list containing header and data.

Examples

```
## Not run:

## Set your API Key first using set_aqs_user to run the following codes

## Example from the AQS website
## Annual performance evaluation data for ozone in Alabama during 2017
aqs_variables <- list(
  param = "44201", bdate = "20170101", edate = "20171231",
  state = "01"
)
aqs_qaannualperformanceevaluations(
  aqs_filter = "byState", aqs_variables = aqs_variables
)

## Equivalent to above; used integers instead of strings
qaannualperformanceevaluations_bystate(
  param = 44201, bdate = "20170101", edate = "20171231",
  state = 1
)

## End(Not run)
```

`aq_s_qablanks`*AQS API QA Blanks Data service*

Description

A collection of functions to fetch the concentration of from blank samples.

Usage

```
aq_s_qablanks(  
  aq_s_filter = c("bySite", "byCounty", "byState", "byPQA0", "byMA"),  
  aq_s_variables = NULL,  
  header = FALSE,  
  ...  
)
```

```
qablanks_bysite(  
  param,  
  bdate,  
  edate,  
  state,  
  county,  
  site,  
  email = get_aq_s_email(),  
  key = get_aq_s_key(),  
  header = FALSE,  
  ...  
)
```

```
qablanks_bycounty(  
  param,  
  bdate,  
  edate,  
  state,  
  county,  
  email = get_aq_s_email(),  
  key = get_aq_s_key(),  
  header = FALSE,  
  ...  
)
```

```
qablanks_bystate(  
  param,  
  bdate,  
  edate,  
  state,  
  email = get_aq_s_email(),
```

```

    key = get_aqs_key(),
    header = FALSE,
    ...
)

qablanks_bypqao(
  param,
  bdate,
  edate,
  pqao,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

qablanks_byma(
  param,
  bdate,
  edate,
  agency,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

```

Arguments

| | |
|----------------------------|--|
| aq _s _filter | A string specifying one of the service filters. NOT case-sensitive. |
| aq _s _variables | A named list of variables to fetch data (e.g., state). Only necessary variables are passed to a specific endpoint (service/filter) to make a valid request. |
| header | A logical specifying whether the function returns additional information from the API header. Default is FALSE to return data only. |
| ... | Reserved for future use. |
| param | A string or vector of strings specifying the 5-digit AQS parameter code for data selection. An integer will be coerced to a string. A maximum of 5 parameter codes may be listed in a single request. A list of the parameter codes can be obtained via list_parametersbyclass . |
| bdate | A string specifying the begin date of data selection in YYYYMMDD format. Only the year portion is used. |
| edate | A string specifying the end date of data selection in YYYYMMDD format. Only the year portion is used. If the end date is not in the same year as the begin date, the function will automatically split the date range into multiple chunks by year and send requests sequentially. |
| state | A string specifying the 2-digit state FIPS code. An integer will be coerced to a string with a leading zero if necessary (e.g., 1 -> "01"). A list of the state codes |

| | |
|--------|--|
| | can be obtained via list_states . |
| county | A string specifying the 3-digit county FIPS code. An integer will be coerced to a string with leading zeros if necessary (e.g., 89 -> "089"). A list of the county codes within each state can be obtained via list_countiesbystate . |
| site | A string specifying the 4-digit AQS site number within the county. An integer will be coerced to a string with leading zeros if necessary (e.g., 14 -> "0014"). A list of the site codes within each county can be obtained via list_sitesbycounty . |
| email | A string specifying the email address of the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| key | A string specifying the key matching the email address for the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| pqao | A string specifying the AQS Primary Quality Assurance Organization (PQAO) code. A list of the PQAO codes can be obtained via list_pqaos . |
| agency | A string specifying the AQS Monitoring Agency (MA) code. A list of the MA codes can be obtained via list_mas . Here, we named this input as agency instead of "ma" because agency is actually used in the API endpoint URL. |

Details

[aq_s_qablanks](#) sends a request to the AQS API based on a user-provided filter using the following underlying functions:

- [qablanks_bysite](#) returns param blank data for site in county, within state, between bdate and edate.
- [qablanks_bycounty](#) returns param blank data for county in state between bdate and edate.
- [qablanks_bystate](#) returns param blank data for state between bdate and edate.
- [qablanks_bypqao](#) returns param blank data for pqao between bdate and edate.
- [qablanks_byma](#) returns param blank data for agency (monitoring agency) between bdate and edate.

Value

A data.frame containing parsed data or a named list containing header and data.

Examples

```
## Not run:

## Set your API Key first using set_aqs_user to run the following codes

## Example from the AQS website
## PM2.5 blank data for Alabama for January 2018
aqs_variables <- list(
  param = "88101", bdate = "20180101", edate = "20180131",
  state = "01"
)
aqs_qablanks(aqs_filter = "byState", aqs_variables = aqs_variables)
```

```

## Equivalent to above; used integers instead of strings
qablanks_bystate(
  param = 88101, bdate = "20180101", edate = "20180131",
  state = 1
)

## End(Not run)

```

aq_s_qaccollocatedassessments
AQS API QA Collocated Assessments service

Description

A collection of functions to fetch pairs of PM samples collected at the same time and place by different samplers.

Usage

```

aqs_qaccollocatedassessments(
  aqs_filter = c("bySite", "byCounty", "byState", "byPQA0", "byMA"),
  aqs_variables = NULL,
  header = FALSE,
  ...
)

qaccollocatedassessments_bysite(
  param,
  bdate,
  edate,
  state,
  county,
  site,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

qaccollocatedassessments_bycounty(
  param,
  bdate,
  edate,
  state,
  county,
  email = get_aqs_email(),

```

```

    key = get_aqs_key(),
    header = FALSE,
    ...
)

qaccollocatedassessments_bystate(
  param,
  bdate,
  edate,
  state,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

qaccollocatedassessments_bypqao(
  param,
  bdate,
  edate,
  pqao,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

qaccollocatedassessments_byma(
  param,
  bdate,
  edate,
  agency,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

```

Arguments

| | |
|----------------------------|--|
| aq _s _filter | A string specifying one of the service filters. NOT case-sensitive. |
| aq _s _variables | A named list of variables to fetch data (e.g., state). Only necessary variables are passed to a specific endpoint (service/filter) to make a valid request. |
| header | A logical specifying whether the function returns additional information from the API header. Default is FALSE to return data only. |
| ... | Reserved for future use. |
| param | A string or vector of strings specifying the 5-digit AQS parameter code for data selection. An integer will be coerced to a string. A maximum of 5 parameter |

codes may be listed in a single request. A list of the parameter codes can be obtained via [list_parametersbyclass](#).

| | |
|--------|--|
| bdate | A string specifying the begin date of data selection in YYYYMMDD format. Only data on or after this date will be returned. |
| edate | A string specifying the end date of data selection in YYYYMMDD format. Only data on or before this date will be returned. If the end date is not in the same year as the begin date, the function will automatically split the date range into multiple chunks by year and send requests sequentially. |
| state | A string specifying the 2-digit state FIPS code. An integer will be coerced to a string with a leading zero if necessary (e.g., 1 -> "01"). A list of the state codes can be obtained via list_states . |
| county | A string specifying the 3-digit county FIPS code. An integer will be coerced to a string with leading zeros if necessary (e.g., 89 -> "089"). A list of the county codes within each state can be obtained via list_countiesbystate . |
| site | A string specifying the 4-digit AQS site number within the county. An integer will be coerced to a string with leading zeros if necessary (e.g., 14 -> "0014"). A list of the site codes within each county can be obtained via list_sitesbycounty . |
| email | A string specifying the email address of the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| key | A string specifying the key matching the email address for the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| pqao | A string specifying the AQS Primary Quality Assurance Organization (PQAO) code. A list of the PQAO codes can be obtained via list_pqaos . |
| agency | A string specifying the AQS Monitoring Agency (MA) code. A list of the MA codes can be obtained via list_mas . Here, we named this input as agency instead of "ma" because agency is actually used in the API endpoint URL. |

Details

[aq_s_qaccollocatedassessments](#) sends a request to the AQS API based on a user-provided filter using the following underlying functions:

- [qaccollocatedassessments_bysite](#) returns collocated assessment data for param at site in county, within state, between bdate and edate.
- [qaccollocatedassessments_bycounty](#) returns collocated assessment data for param in county within state between bdate and edate.
- [qaccollocatedassessments_bystate](#) returns collocated assessment data for param in state between bdate and edate.
- [qaccollocatedassessments_bypqao](#) returns collocated assessment data for param in pqao between bdate and edate.
- [qaccollocatedassessments_byma](#) returns collocated assessment data for param in agency (monitoring agency) between bdate and edate.

Value

A data.frame containing parsed data or a named list containing header and data.

Examples

```
## Not run:

## Set your API Key first using set_aqs_user to run the following codes

## Example from the AQS website
## Collocated assessment data for FRM PM2.5 in Alabama for January 2013
aqs_variables <- list(
  param = "88101", bdate = "20130101", edate = "20130131",
  state = "01"
)
aqs_qacollocatedassessments(
  aqsfilter = "byState", aqsvariables = aqsvariables
)

## Equivalent to above; used integers instead of strings
qacollocatedassessments_bystate(
  param = 88101, bdate = "20130101", edate = "20130131",
  state = 1
)

## End(Not run)
```

aq_s_qaflowrateaudits *AQS API QA Flow Rate Audits service*

Description

A collection of functions to fetch flow rate audit data.

Usage

```
aqs_qaflowrateaudits(
  aqsfilter = c("bySite", "byCounty", "byState", "byPQA0", "byMA"),
  aqsvariables = NULL,
  header = FALSE,
  ...
)

qaflowrateaudits_bysite(
  param,
  bdate,
  edate,
  state,
  county,
  site,
  email = get_aqs_email(),
```

```
    key = get_aqs_key(),
    header = FALSE,
    ...
)

qaflowrateaudits_bycounty(
  param,
  bdate,
  edate,
  state,
  county,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

qaflowrateaudits_bystate(
  param,
  bdate,
  edate,
  state,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

qaflowrateaudits_bypqao(
  param,
  bdate,
  edate,
  pqao,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

qaflowrateaudits_byma(
  param,
  bdate,
  edate,
  agency,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)
```

)

Arguments

| | |
|----------------------------|--|
| aq _s _filter | A string specifying one of the service filters. NOT case-sensitive. |
| aq _s _variables | A named list of variables to fetch data (e.g., state). Only necessary variables are passed to a specific endpoint (service/filter) to make a valid request. |
| header | A logical specifying whether the function returns additional information from the API header. Default is FALSE to return data only. |
| ... | Reserved for future use. |
| param | A string or vector of strings specifying the 5-digit AQS parameter code for data selection. An integer will be coerced to a string. A maximum of 5 parameter codes may be listed in a single request. A list of the parameter codes can be obtained via list_parametersbyclass . |
| bdate | A string specifying the begin date of data selection in YYYYMMDD format. Only data on or after this date will be returned. |
| edate | A string specifying the end date of data selection in YYYYMMDD format. Only data on or before this date will be returned. If the end date is not in the same year as the begin date, the function will automatically split the date range into multiple chunks by year and send requests sequentially. |
| state | A string specifying the 2-digit state FIPS code. An integer will be coerced to a string with a leading zero if necessary (e.g., 1 -> "01"). A list of the state codes can be obtained via list_states . |
| county | A string specifying the 3-digit county FIPS code. An integer will be coerced to a string with leading zeros if necessary (e.g., 89 -> "089"). A list of the county codes within each state can be obtained via list_countiesbystate . |
| site | A string specifying the 4-digit AQS site number within the county. An integer will be coerced to a string with leading zeros if necessary (e.g., 14 -> "0014"). A list of the site codes within each county can be obtained via list_sitesbycounty . |
| email | A string specifying the email address of the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| key | A string specifying the key matching the email address for the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| pqao | A string specifying the AQS Primary Quality Assurance Organization (PQAO) code. A list of the PQAO codes can be obtained via list_pqaos . |
| agency | A string specifying the AQS Monitoring Agency (MA) code. A list of the MA codes can be obtained via list_mas . Here, we named this input as agency instead of "ma" because agency is actually used in the API endpoint URL. |

Details

[aq_s_q_aflowrateaudits](#) sends a request to the AQS API based on a user-provided filter using the following underlying functions:

- [q_aflowrateaudits_bysite](#) returns Flow Rate Audit data for param at site in county, within state, between bdate and edate.

- [qaflowrateverifications_bycounty](#) returns Flow Rate Audit data for param in county within state between bdate and edate.
- [qaflowrateaudits_bystate](#) returns Flow Rate Audit data for param in state between bdate and edate.
- [qaflowrateaudits_bypqao](#) returns Flow Rate Audit data for param in pqao between bdate and edate.
- [qaflowrateaudits_byma](#) returns Flow Rate Audit data for param in agency (monitoring agency) between bdate and edate.

Value

A data.frame containing parsed data or a named list containing header and data.

Examples

```
## Not run:

## Set your API Key first using set_aqs_user to run the following codes

## Example from the AQS website
## Flow rate audit data for Alabama during January 2018
aqs_variables <- list(
  param = "88101", bdate = "20180101", edate = "20180131",
  state = "01"
)
aqs_qaflowrateaudits(
  aqs_filter = "byState", aqs_variables = aqs_variables
)

## Equivalent to above; used integers instead of strings
qaflowrateaudits_bystate(
  param = 88101, bdate = "20180101", edate = "20180131",
  state = 1
)

## End(Not run)
```

aqs_qaflowrateverifications

AQS API QA Flow Rate Verifications service

Description

A collection of functions to fetch flow rate checks performed by monitoring agencies.

Usage

```
aqs_qaflowrateverifications(  
  aqs_filter = c("bySite", "byCounty", "byState", "byPQA0", "byMA"),  
  aqs_variables = NULL,  
  header = FALSE,  
  ...  
)
```

```
qaflowrateverifications_bysite(  
  param,  
  bdate,  
  edate,  
  state,  
  county,  
  site,  
  email = get_aqs_email(),  
  key = get_aqs_key(),  
  header = FALSE,  
  ...  
)
```

```
qaflowrateverifications_bycounty(  
  param,  
  bdate,  
  edate,  
  state,  
  county,  
  email = get_aqs_email(),  
  key = get_aqs_key(),  
  header = FALSE,  
  ...  
)
```

```
qaflowrateverifications_bystate(  
  param,  
  bdate,  
  edate,  
  state,  
  email = get_aqs_email(),  
  key = get_aqs_key(),  
  header = FALSE,  
  ...  
)
```

```
qaflowrateverifications_bypqao(  
  param,  
  bdate,  
  edate,
```

```

    pqao,
    email = get_aqs_email(),
    key = get_aqs_key(),
    header = FALSE,
    ...
)

qaflowrateverifications_byma(
  param,
  bdate,
  edate,
  agency,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

```

Arguments

| | |
|----------------------------|--|
| aq _s _filter | A string specifying one of the service filters. NOT case-sensitive. |
| aq _s _variables | A named list of variables to fetch data (e.g., state). Only necessary variables are passed to a specific endpoint (service/filter) to make a valid request. |
| header | A logical specifying whether the function returns additional information from the API header. Default is FALSE to return data only. |
| ... | Reserved for future use. |
| param | A string or vector of strings specifying the 5-digit AQS parameter code for data selection. An integer will be coerced to a string. A maximum of 5 parameter codes may be listed in a single request. A list of the parameter codes can be obtained via list_parametersbyclass . |
| bdate | A string specifying the begin date of data selection in YYYYMMDD format. Only data on or after this date will be returned. |
| edate | A string specifying the end date of data selection in YYYYMMDD format. Only data on or before this date will be returned. If the end date is not in the same year as the begin date, the function will automatically split the date range into multiple chunks by year and send requests sequentially. |
| state | A string specifying the 2-digit state FIPS code. An integer will be coerced to a string with a leading zero if necessary (e.g., 1 -> "01"). A list of the state codes can be obtained via list_states . |
| county | A string specifying the 3-digit county FIPS code. An integer will be coerced to a string with leading zeros if necessary (e.g., 89 -> "089"). A list of the county codes within each state can be obtained via list_countiesbystate . |
| site | A string specifying the 4-digit AQS site number within the county. An integer will be coerced to a string with leading zeros if necessary (e.g., 14 -> "0014"). A list of the site codes within each county can be obtained via list_sitesbycounty . |

| | |
|--------|---|
| email | A string specifying the email address of the requester. If you set your email and key with <code>set_aqs_user</code> , you don't have to specify this. |
| key | A string specifying the key matching the email address for the requester. If you set your email and key with <code>set_aqs_user</code> , you don't have to specify this. |
| pqao | A string specifying the AQS Primary Quality Assurance Organization (PQAO) code. A list of the PQAO codes can be obtained via list_pqaos . |
| agency | A string specifying the AQS Monitoring Agency (MA) code. A list of the MA codes can be obtained via list_mas . Here, we named this input as agency instead of "ma" because agency is actually used in the API endpoint URL. |

Details

`aqs_qaflowrateverifications` sends a request to the AQS API based on a user-provided filter using the following underlying functions:

- `qaflowrateverifications_bysite` returns Flow Rate Verification data for param at site in county, within state, between bdate and edate.
- `qaflowrateverifications_bycounty` returns Flow Rate Verification data for param in county within state between bdate and edate.
- `qaflowrateverifications_bystate` returns Flow Rate Verification data for param in state between bdate and edate.
- `qaflowrateverifications_bypqao` returns Flow Rate Verification data for param in pqao between bdate and edate.
- `qaflowrateverifications_byma` returns Flow Rate Verification data for param in agency (monitoring agency) between bdate and edate.

Value

A data.frame containing parsed data or a named list containing header and data.

Examples

```
## Not run:

## Set your API Key first using set_aqs_user to run the following codes

## Example from the AQS website
## Flow Rate Verification data for Alabama during January 2018
aqs_variables <- list(
  param = "88101", bdate = "20180101", edate = "20180131",
  state = "01"
)
aqs_qaflowrateverifications(
  aqs_filter = "byState", aqs_variables = aqs_variables
)

## Equivalent to above; used integers instead of strings
qaflowrateverifications_bystate(
  param = 88101, bdate = "20180101", edate = "20180131",
```

```
    state = 1
  )

## End(Not run)
```

aq_s_qaonepointqcrawdata

AQS API QA One Point QC Raw Data service

Description

A collection of functions to fetch measured versus actual concentration of 1 point QC checks.

Usage

```
aqs_qaonepointqcrawdata(
  aqs_filter = c("bySite", "byCounty", "byState", "byPQA0", "byMA"),
  aqs_variables = NULL,
  header = FALSE,
  ...
)

qaonepointqcrawdata_bysite(
  param,
  bdate,
  edate,
  state,
  county,
  site,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

qaonepointqcrawdata_bycounty(
  param,
  bdate,
  edate,
  state,
  county,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)
```

```

qaonepointqcrawdata_bystate(
  param,
  bdate,
  edate,
  state,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

qaonepointqcrawdata_bypqao(
  param,
  bdate,
  edate,
  pqao,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

qaonepointqcrawdata_byma(
  param,
  bdate,
  edate,
  agency,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

```

Arguments

| | |
|----------------------------|--|
| aq _s _filter | A string specifying one of the service filters. NOT case-sensitive. |
| aq _s _variables | A named list of variables to fetch data (e.g., state). Only necessary variables are passed to a specific endpoint (service/filter) to make a valid request. |
| header | A logical specifying whether the function returns additional information from the API header. Default is FALSE to return data only. |
| ... | Reserved for future use. |
| param | A string or vector of strings specifying the 5-digit AQS parameter code for data selection. An integer will be coerced to a string. A maximum of 5 parameter codes may be listed in a single request. A list of the parameter codes can be obtained via list_parametersbyclass . |
| bdate | A string specifying the begin date of data selection in YYYYMMDD format. Only data on or after this date will be returned. |

| | |
|--------|--|
| edate | A string specifying the end date of data selection in YYYYMMDD format. Only data on or before this date will be returned. If the end date is not in the same year as the begin date, the function will automatically split the date range into multiple chunks by year and send requests sequentially. |
| state | A string specifying the 2-digit state FIPS code. An integer will be coerced to a string with a leading zero if necessary (e.g., 1 -> "01"). A list of the state codes can be obtained via list_states . |
| county | A string specifying the 3-digit county FIPS code. An integer will be coerced to a string with leading zeros if necessary (e.g., 89 -> "089"). A list of the county codes within each state can be obtained via list_countiesbystate . |
| site | A string specifying the 4-digit AQS site number within the county. An integer will be coerced to a string with leading zeros if necessary (e.g., 14 -> "0014"). A list of the site codes within each county can be obtained via list_sitesbycounty . |
| email | A string specifying the email address of the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| key | A string specifying the key matching the email address for the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| pqaao | A string specifying the AQS Primary Quality Assurance Organization (PQAO) code. A list of the PQAO codes can be obtained via list_pqaos . |
| agency | A string specifying the AQS Monitoring Agency (MA) code. A list of the MA codes can be obtained via list_mas . Here, we named this input as agency instead of "ma" because agency is actually used in the API endpoint URL. |

Details

[aqs_qaonepointqcrawdata](#) sends a request to the AQS API based on a user-provided filter using the following underlying functions:

- [qaonepointqcrawdata_bysite](#) returns One Point QC data for param at site in county, within state, between bdate and edate.
- [qaonepointqcrawdata_bycounty](#) returns One Point QC data for param in county within state between bdate and edate.
- [qaonepointqcrawdata_bystate](#) returns One Point QC data for param in state between bdate and edate.
- [qaonepointqcrawdata_bypqaao](#) returns One Point QC data for param in pqaao between bdate and edate.
- [qaflowrateaudits_byma](#) returns One Point QC data for param in agency (monitoring agency) between bdate and edate.

Value

A data.frame containing parsed data or a named list containing header and data.

Examples

```
## Not run:

## Set your API Key first using set_aqs_user to run the following codes

## Example from the AQS website
## One Point QC data for ozone in Massachusetts for January 2018
aqs_variables <- list(
  param = "44201", bdate = "20180101", edate = "20180131",
  state = "25"
)
aqs_qaonepointqcrawdata(
  aqs_filter = "byState", aqs_variables = aqs_variables
)

## Equivalent to above; used integers instead of strings
qaonepointqcrawdata_bystate(
  param = 44201, bdate = "20180101", edate = "20180131",
  state = 25
)

## End(Not run)
```

aq_s_qa_{pe}paudits

AQS API QA PEP Audits service

Description

A collection of functions to fetch data related to PM_{2.5} monitoring system audits.

Usage

```
aqs_qapepaudits(
  aqs_filter = c("bySite", "byCounty", "byState", "byPQA0", "byMA"),
  aqs_variables = NULL,
  header = FALSE,
  ...
)

qapepaudits_bysite(
  param,
  bdate,
  edate,
  state,
  county,
  site,
  email = get_aqs_email(),
```

```
    key = get_aqs_key(),
    header = FALSE,
    ...
)

qaepaudits_bycounty(
  param,
  bdate,
  edate,
  state,
  county,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

qaepaudits_bystate(
  param,
  bdate,
  edate,
  state,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

qaepaudits_bypqao(
  param,
  bdate,
  edate,
  pqao,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

qaepaudits_byma(
  param,
  bdate,
  edate,
  agency,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)
```


)

Arguments

| | |
|----------------------------|--|
| aq _s _filter | A string specifying one of the service filters. NOT case-sensitive. |
| aq _s _variables | A named list of variables to fetch data (e.g., state). Only necessary variables are passed to a specific endpoint (service/filter) to make a valid request. |
| header | A logical specifying whether the function returns additional information from the API header. Default is FALSE to return data only. |
| ... | Reserved for future use. |
| param | A string or vector of strings specifying the 5-digit AQS parameter code for data selection. An integer will be coerced to a string. A maximum of 5 parameter codes may be listed in a single request. A list of the parameter codes can be obtained via list_parametersbyclass . |
| bdate | A string specifying the begin date of data selection in YYYYMMDD format. Only data on or after this date will be returned. |
| edate | A string specifying the end date of data selection in YYYYMMDD format. Only data on or before this date will be returned. If the end date is not in the same year as the begin date, the function will automatically split the date range into multiple chunks by year and send requests sequentially. |
| state | A string specifying the 2-digit state FIPS code. An integer will be coerced to a string with a leading zero if necessary (e.g., 1 -> "01"). A list of the state codes can be obtained via list_states . |
| county | A string specifying the 3-digit county FIPS code. An integer will be coerced to a string with leading zeros if necessary (e.g., 89 -> "089"). A list of the county codes within each state can be obtained via list_countiesbystate . |
| site | A string specifying the 4-digit AQS site number within the county. An integer will be coerced to a string with leading zeros if necessary (e.g., 14 -> "0014"). A list of the site codes within each county can be obtained via list_sitesbycounty . |
| email | A string specifying the email address of the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| key | A string specifying the key matching the email address for the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| pqao | A string specifying the AQS Primary Quality Assurance Organization (PQAO) code. A list of the PQAO codes can be obtained via list_pqaos . |
| agency | A string specifying the AQS Monitoring Agency (MA) code. A list of the MA codes can be obtained via list_mas . Here, we named this input as agency instead of "ma" because agency is actually used in the API endpoint URL. |

Details

[aq_s_qa_{pe}paudits](#) sends a request to the AQS API based on a user-provided filter using the following underlying functions:

- [qa_{pe}paudits_bysite](#) returns PEP Audit data for param at site in county, within state, between bdate and edate.

- [qa_{pe}paudits_bycounty](#) returns PEP Audit data for param in county within state between bdate and edate.
- [qa_{pe}paudits_bystate](#) returns PEP Audit data for param in state between bdate and edate.
- [qa_{pe}paudits_bypqao](#) returns PEP Audit data for param in pqao between bdate and edate.
- [qa_{pe}paudits_byma](#) returns PEP Audit data for param in agency (monitoring agency) between bdate and edate.

Value

A data.frame containing parsed data or a named list containing header and data.

Examples

```
## Not run:

## Set your API Key first using set_aqs_user to run the following codes

## Example from the AQS website
## PEP Audit data for FRM PM2.5 in Alabama for 2017
aqs_variables <- list(
  param = "88101", bdate = "20170101", edate = "20171231",
  state = "01"
)
aqs_qapepaudits(
  aqs_filter = "byState", aqs_variables = aqs_variables
)

## Equivalent to above; used integers instead of strings
qapepaudits_bystate(
  param = 88101, bdate = "20170101", edate = "20171231",
  state = 1
)

## End(Not run)
```

aq_s_quarterlydata

AQS API Quarterly Summary Data service

Description

A collection of functions to fetch data summarized at the calendar quarter level. Data is labeled with quarter number (Q1 = Jan - Mar, Q2 = Apr - Jun, Q3 = Jul - Sep, Q4 = Oct - Dec). Note that only the year portion of the bdate and edate are used and all 4 quarters in the year are returned. In addition, duration is not allowed on the API unlike [aq_s_sampledata](#), [aq_s_dailydata](#), and [aq_s_annualdata](#).

Usage

```
aqs_quarterlydata(  
  aqs_filter = c("bySite", "byCounty", "byState", "byBox", "byCBSA"),  
  aqs_variables = NULL,  
  header = FALSE,  
  ...  
)  
  
quarterlydata_bysite(  
  param,  
  bdate,  
  edate,  
  state,  
  county,  
  site,  
  email = get_aqs_email(),  
  key = get_aqs_key(),  
  cbdate = NULL,  
  cedate = NULL,  
  header = FALSE,  
  ...  
)  
  
quarterlydata_bycounty(  
  param,  
  bdate,  
  edate,  
  state,  
  county,  
  email = get_aqs_email(),  
  key = get_aqs_key(),  
  cbdate = NULL,  
  cedate = NULL,  
  header = FALSE,  
  ...  
)  
  
quarterlydata_bystate(  
  param,  
  bdate,  
  edate,  
  state,  
  email = get_aqs_email(),  
  key = get_aqs_key(),  
  cbdate = NULL,  
  cedate = NULL,  
  header = FALSE,  
  ...  
)
```

```

)

quarterlydata_bybox(
  param,
  bdate,
  edate,
  minlat,
  maxlat,
  minlon,
  maxlon,
  email = get_aqs_email(),
  key = get_aqs_key(),
  cbdate = NULL,
  cedate = NULL,
  header = FALSE,
  ...
)

quarterlydata_bycbsa(
  param,
  bdate,
  edate,
  cbsa,
  email = get_aqs_email(),
  key = get_aqs_key(),
  cbdate = NULL,
  cedate = NULL,
  header = FALSE,
  ...
)

```

Arguments

| | |
|----------------------------|--|
| aq _s _filter | A string specifying one of the service filters. NOT case-sensitive. |
| aq _s _variables | A named list of variables to fetch data (e.g., state). Only necessary variables are passed to a specific endpoint (service/filter) to make a valid request. |
| header | A logical specifying whether the function returns additional information from the API header. Default is FALSE to return data only. |
| ... | Reserved for future use. |
| param | A string or vector of strings specifying the 5-digit AQS parameter code for data selection. An integer will be coerced to a string. A maximum of 5 parameter codes may be listed in a single request. A list of the parameter codes can be obtained via list_parametersbyclass . |
| bdate | A string specifying the begin date of data selection in YYYYMMDD format. Only the year portion is used. |
| edate | A string specifying the end date of data selection in YYYYMMDD format. Only the year portion is used. If the end date is not in the same year as the begin date, |

the function will automatically split the date range into multiple chunks by year and send requests sequentially.

| | |
|--------|--|
| state | A string specifying the 2-digit state FIPS code. An integer will be coerced to a string with a leading zero if necessary (e.g., 1 -> "01"). A list of the state codes can be obtained via list_states . |
| county | A string specifying the 3-digit county FIPS code. An integer will be coerced to a string with leading zeros if necessary (e.g., 89 -> "089"). A list of the county codes within each state can be obtained via list_countiesbystate . |
| site | A string specifying the 4-digit AQS site number within the county. An integer will be coerced to a string with leading zeros if necessary (e.g., 14 -> "0014"). A list of the site codes within each county can be obtained via list_sitesbycounty . |
| email | A string specifying the email address of the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| key | A string specifying the key matching the email address for the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| cbdate | (optional) A string specifying the change begin date in YYYYMMDD format to subset data based on "date of last change" in database. Only data that changed on or after this date will be returned. |
| cedate | (optional) A string specifying the change end date in YYYYMMDD format to subset data based on "date of last change" in database. Only data that changed on or before this date will be returned. |
| minlat | A string or numeric value specifying the minimum latitude of a geographic box. Decimal latitude with north being positive. |
| maxlat | A string or numeric value specifying the maximum latitude of a geographic box. Decimal latitude with north being positive. |
| minlon | A string or numeric value specifying the minimum longitude of a geographic box. Decimal longitude with east being positive. |
| maxlon | A string or numeric value specifying the maximum longitude of a geographic box. Decimal longitude with east being positive. |
| cbsa | A string specifying the AQS CBSA code. A list of the CBSA codes can be obtained via list_cbsas . |

Details

[aq_s_quarterlydata](#) sends a request to the AQS API based on a user-provided filter using the following underlying functions:

- [quarterlydata_bysite](#) returns quarterly summary param data for site in county, within state, based on the year portions of bdate and edate.
- [quarterlydata_bycounty](#) returns quarterly summary param data for county in state based on the year portions bdate and edate.
- [quarterlydata_bystate](#) returns quarterly summary param data for state based on the year portions of bdate and edate.

- [quarterlydata_bybox](#) returns quarterly summary param data for a user-provided latitude/longitude bounding box (minlat, maxlat, minlon, maxlon) based on the year portions of bdate and edate.
- [quarterlydata_bycbsa](#) returns quarterly summary param data for a user-provided CBSA based on the year portions of bdate and edate.

Value

A data.frame containing parsed data or a named list containing header and data.

Examples

```
## Not run:

## Set your API Key first using set_aqs_user to run the following codes

## Example from the AQS website
## FRM/FEM PM2.5 data for Wake County, NC for 2016
## Only the year portions of bdate and edate are used
aq_variables <- list(
  param = c("88101", "88502"), bdate = "20160101", edate = "20160228",
  state = "37", county = "183"
)
aq_quarterlydata(aqs_filter = "byCounty", aqs_variables = aq_variables)

## Equivalent to above; used integers instead of strings
quarterlydata_bycounty(
  param = c(88101, 88502), bdate = "20160101", edate = "20160228",
  state = 37, county = 183
)

## End(Not run)
```

aq_sampladata

AQS API Sample Data service

Description

A collection of functions to fetch sample data - the finest grain data reported to EPA. Please use a narrow range of dates to adhere to the API's limit imposed on request size.

Usage

```
aq_sampladata(
  aqs_filter = c("bySite", "byCounty", "byState", "byBox", "byCBSA"),
  aqs_variables = NULL,
  header = FALSE,
  ...
)
```

```
)

sampledata_bysite(
  param,
  bdate,
  edate,
  state,
  county,
  site,
  email = get_aqs_email(),
  key = get_aqs_key(),
  duration = NULL,
  cbdate = NULL,
  cedate = NULL,
  header = FALSE,
  ...
)

sampledata_bycounty(
  param,
  bdate,
  edate,
  state,
  county,
  email = get_aqs_email(),
  key = get_aqs_key(),
  duration = NULL,
  cbdate = NULL,
  cedate = NULL,
  header = FALSE,
  ...
)

sampledata_bystate(
  param,
  bdate,
  edate,
  state,
  email = get_aqs_email(),
  key = get_aqs_key(),
  duration = NULL,
  cbdate = NULL,
  cedate = NULL,
  header = FALSE,
  ...
)

sampledata_bybox(
```

```

    param,
    bdate,
    edate,
    minlat,
    maxlat,
    minlon,
    maxlon,
    email = get_aqs_email(),
    key = get_aqs_key(),
    duration = NULL,
    cbdate = NULL,
    cedate = NULL,
    header = FALSE,
    ...
)

sampledata_bycbsa(
  param,
  bdate,
  edate,
  cbsa,
  email = get_aqs_email(),
  key = get_aqs_key(),
  duration = NULL,
  cbdate = NULL,
  cedate = NULL,
  header = FALSE,
  ...
)

```

Arguments

| | |
|----------------------------|--|
| aq _s _filter | A string specifying one of the service filters. NOT case-sensitive. |
| aq _s _variables | A named list of variables to fetch data (e.g., state). Only necessary variables are passed to a specific endpoint (service/filter) to make a valid request. |
| header | A logical specifying whether the function returns additional information from the API header. Default is FALSE to return data only. |
| ... | Reserved for future use. |
| param | A string or vector of strings specifying the 5-digit AQS parameter code for data selection. An integer will be coerced to a string. A maximum of 5 parameter codes may be listed in a single request. A list of the parameter codes can be obtained via list_parametersbyclass . |
| bdate | A string specifying the begin date of data selection in YYYYMMDD format. Only data on or after this date will be returned. |
| edate | A string specifying the end date of data selection in YYYYMMDD format. Only data on or before this date will be returned. If the end date is not in the same year as the begin date, the function will automatically split the date range into multiple chunks by year and send requests sequentially. |

| | |
|----------|--|
| state | A string specifying the 2-digit state FIPS code. An integer will be coerced to a string with a leading zero if necessary (e.g., 1 -> "01"). A list of the state codes can be obtained via list_states . |
| county | A string specifying the 3-digit county FIPS code. An integer will be coerced to a string with leading zeros if necessary (e.g., 89 -> "089"). A list of the county codes within each state can be obtained via list_countiesbystate . |
| site | A string specifying the 4-digit AQS site number within the county. An integer will be coerced to a string with leading zeros if necessary (e.g., 14 -> "0014"). A list of the site codes within each county can be obtained via list_sitesbycounty . |
| email | A string specifying the email address of the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| key | A string specifying the key matching the email address for the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| duration | (optional) A string specifying the 1-character AQS sample duration code. A list of the duration codes can be obtained via list_durations . Only data reported at this sample duration will be returned. |
| cbdate | (optional) A string specifying the change begin date in YYYYMMDD format to subset data based on "date of last change" in database. Only data that changed on or after this date will be returned. |
| cedate | (optional) A string specifying the change end date in YYYYMMDD format to subset data based on "date of last change" in database. Only data that changed on or before this date will be returned. |
| minlat | A string or numeric value specifying the minimum latitude of a geographic box. Decimal latitude with north being positive. |
| maxlat | A string or numeric value specifying the maximum latitude of a geographic box. Decimal latitude with north being positive. |
| minlon | A string or numeric value specifying the minimum longitude of a geographic box. Decimal longitude with east being positive. |
| maxlon | A string or numeric value specifying the maximum longitude of a geographic box. Decimal longitude with east being positive. |
| cbsa | A string specifying the AQS CBSA code. A list of the CBSA codes can be obtained via list_cbsas . |

Details

[aq_s_sampledata](#) sends a request to the AQS API based on a user-provided filter using the following underlying functions:

- [sampledata_bysite](#) returns all param samples for site in county, within state, between bdate and edate.
- [sampledata_bycounty](#) returns all param samples for county in state between bdate and edate.
- [sampledata_bystate](#) returns all param samples for state between bdate and edate.
- [sampledata_bybox](#) returns all param samples for a user-provided latitude/longitude bounding box (minlat, maxlat, minlon, maxlon) between bdate and edate.

- `sampledata_bycbsa` returns all param samples for a user-provided CBSA between bdate and edate.

Value

A data.frame containing parsed data or a named list containing header and data.

Examples

```
## Not run:

## Set your API Key first using set_aqs_user to run the following codes

## Example from the AQS website
## FRM/FEM PM2.5 data for Wake County, NC between Jan and Feb 2016
aqs_variables <- list(
  param = "88101", bdate = "20160101", edate = "20160228",
  state = "37", county = "183"
)
aqs_sampledata(aqs_filter = "byCounty", aqs_variables = aqs_variables)

## Equivalent to above; used integers instead of strings
sampledata_bycounty(
  param = 88101, bdate = "20160101", edate = "20160228",
  state = 37, county = 183
)

## End(Not run)
```

aq_s_signup

Create an account for the AQS API

Description

This function helps you create an account or reset a password. Once you execute this function, a verification email will be sent to the email account specified. If the request is made with an email that is already registered, a new key will be issued for that account and emailed to the listed address.

Usage

```
aqs_signup(email)
```

Arguments

email A string specifying an email account to register as a user

Value

No return value, called to sign up for the AQS API

See Also

See [set_aqs_user](#) to set your credentials to send a request to the AQS API.

Examples

```
## Not run:

## Please use your email address to create an account

aqs_signup(email = "youreemail@toregister.com")

## End(Not run)
```

```
aqs_transactionsqaannualperformanceevaluations
      AQS API QA Annual Performance Evaluations Transaction service
```

Description

A collection of functions to fetch pairs of data QA at several concentration levels in the submission (transaction) format for AQS.

Usage

```
aqs_transactionsqaannualperformanceevaluations(
  aqs_filter = c("bySite", "byCounty", "byState", "byPQA0", "byMA"),
  aqs_variables = NULL,
  header = FALSE,
  ...
)

transactionsqaannualperformanceevaluations_bysite(
  param,
  bdate,
  edate,
  state,
  county,
  site,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

transactionsqaannualperformanceevaluations_bycounty(
  param,
```

```

    bdate,
    edate,
    state,
    county,
    email = get_aqs_email(),
    key = get_aqs_key(),
    header = FALSE,
    ...
)

transactionsqaannualperformanceevaluations_bystate(
  param,
  bdate,
  edate,
  state,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

transactionsqaannualperformanceevaluations_bypqao(
  param,
  bdate,
  edate,
  pqao,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

transactionsqaannualperformanceevaluations_byma(
  param,
  bdate,
  edate,
  agency,
  email = get_aqs_email(),
  key = get_aqs_key(),
  header = FALSE,
  ...
)

```

Arguments

aqs_filter A string specifying one of the service filters. NOT case-sensitive.

aqs_variables A named list of variables to fetch data (e.g., state). Only necessary variables are passed to a specific endpoint (service/filter) to make a valid request.

| | |
|--------|--|
| header | A logical specifying whether the function returns additional information from the API header. Default is FALSE to return data only. |
| ... | Reserved for future use. |
| param | A string or vector of strings specifying the 5-digit AQS parameter code for data selection. An integer will be coerced to a string. A maximum of 5 parameter codes may be listed in a single request. A list of the parameter codes can be obtained via list_parametersbyclass . |
| bdate | A string specifying the begin date of data selection in YYYYMMDD format. Only data on or after this date will be returned. |
| edate | A string specifying the end date of data selection in YYYYMMDD format. Only data on or before this date will be returned. If the end date is not in the same year as the begin date, the function will automatically split the date range into multiple chunks by year and send requests sequentially. |
| state | A string specifying the 2-digit state FIPS code. An integer will be coerced to a string with a leading zero if necessary (e.g., 1 -> "01"). A list of the state codes can be obtained via list_states . |
| county | A string specifying the 3-digit county FIPS code. An integer will be coerced to a string with leading zeros if necessary (e.g., 89 -> "089"). A list of the county codes within each state can be obtained via list_countiesbystate . |
| site | A string specifying the 4-digit AQS site number within the county. An integer will be coerced to a string with leading zeros if necessary (e.g., 14 -> "0014"). A list of the site codes within each county can be obtained via list_sitesbycounty . |
| email | A string specifying the email address of the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| key | A string specifying the key matching the email address for the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| pqao | A string specifying the AQS Primary Quality Assurance Organization (PQAO) code. A list of the PQAO codes can be obtained via list_pqaos . |
| agency | A string specifying the AQS Monitoring Agency (MA) code. A list of the MA codes can be obtained via list_mas . Here, we named this input as agency instead of "ma" because agency is actually used in the API endpoint URL. |

Details

[aq_transactionsqaannualperformanceevaluations](#) sends a request to the AQS API based on a user-provided filter using the following underlying functions:

- [transactionsqaannualperformanceevaluations_bysite](#) returns annual performance evaluation data for param at site in county, within state, between bdate and edate in the transaction format.
- [transactionsqaannualperformanceevaluations_bycounty](#) returns annual performance evaluation data for param in county within state between bdate and edate in the transaction format.
- [transactionsqaannualperformanceevaluations_bystate](#) returns annual performance evaluation data for param in state between bdate and edate in the transaction format.

- [transactionsqaannualperformanceevaluations_bypqao](#) returns annual performance evaluation data for param in pqao between bdate and edate in the transaction format.
- [transactionsqaannualperformanceevaluations_byma](#) returns annual performance evaluation data for param in agency (monitoring agency) between bdate and edate in the transaction format.

Value

A data.frame containing parsed data or a named list containing header and data.

Examples

```
## Not run:

## Set your API Key first using set_aqs_user to run the following codes

## Example from the AQS website
## Annual performance evaluation data for ozone in Alabama during 2017
aqs_variables <- list(
  param = "44201", bdate = "20170101", edate = "20171231",
  state = "01"
)
aqs_transactionsqaannualperformanceevaluations(
  aqs_filter = "byState", aqs_variables = aqs_variables
)

## Equivalent to above; used integers instead of strings
transactionsqaannualperformanceevaluations_bystate(
  param = 44201, bdate = "20170101", edate = "20171231",
  state = 1
)

## End(Not run)
```

aq_s_transactionssample

AQS API Sample Data Transaction service

Description

A collection of functions to fetch data in the submission (transaction) format for AQS.

Usage

```
aqs_transactionssample(
  aqs_filter = c("bySite", "byCounty", "byState", "byMA"),
  aqs_variables = NULL,
  header = FALSE,
  ...
)
```

```
)  
  
transactionssample_bysite(  
  param,  
  bdate,  
  edate,  
  state,  
  county,  
  site,  
  email = get_aqs_email(),  
  key = get_aqs_key(),  
  header = FALSE,  
  ...  
)  
  
transactionssample_bycounty(  
  param,  
  bdate,  
  edate,  
  state,  
  county,  
  email = get_aqs_email(),  
  key = get_aqs_key(),  
  header = FALSE,  
  ...  
)  
  
transactionssample_bystate(  
  param,  
  bdate,  
  edate,  
  state,  
  email = get_aqs_email(),  
  key = get_aqs_key(),  
  header = FALSE,  
  ...  
)  
  
transactionssample_byma(  
  param,  
  bdate,  
  edate,  
  agency,  
  email = get_aqs_email(),  
  key = get_aqs_key(),  
  header = FALSE,  
  ...  
)
```

Arguments

| | |
|----------------|--|
| aqss_filter | A string specifying one of the service filters. NOT case-sensitive. |
| aqss_variables | A named list of variables to fetch data (e.g., state). Only necessary variables are passed to a specific endpoint (service/filter) to make a valid request. |
| header | A logical specifying whether the function returns additional information from the API header. Default is FALSE to return data only. |
| ... | Reserved for future use. |
| param | A string or vector of strings specifying the 5-digit AQS parameter code for data selection. An integer will be coerced to a string. A maximum of 5 parameter codes may be listed in a single request. A list of the parameter codes can be obtained via list_parametersbyclass . |
| bdate | A string specifying the begin date of data selection in YYYYMMDD format. Only data on or after this date will be returned. |
| edate | A string specifying the end date of data selection in YYYYMMDD format. Only data on or before this date will be returned. If the end date is not in the same year as the begin date, the function will automatically split the date range into multiple chunks by year and send requests sequentially. |
| state | A string specifying the 2-digit state FIPS code. An integer will be coerced to a string with a leading zero if necessary (e.g., 1 -> "01"). A list of the state codes can be obtained via list_states . |
| county | A string specifying the 3-digit county FIPS code. An integer will be coerced to a string with leading zeros if necessary (e.g., 89 -> "089"). A list of the county codes within each state can be obtained via list_countiesbystate . |
| site | A string specifying the 4-digit AQS site number within the county. An integer will be coerced to a string with leading zeros if necessary (e.g., 14 -> "0014"). A list of the site codes within each county can be obtained via list_sitesbycounty . |
| email | A string specifying the email address of the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| key | A string specifying the key matching the email address for the requester. If you set your email and key with set_aqs_user , you don't have to specify this. |
| agency | A string specifying the AQS Monitoring Agency (MA) code. A list of the MA codes can be obtained via list_mas . Here, we named this input as agency instead of "ma" because agency is actually used in the API endpoint URL. |

Details

[aqss_transactionssample](#) sends a request to the AQS API based on a user-provided filter using the following underlying functions:

- [transactionssample_bysite](#) returns all param data for site in county, within state, between bdate and edate in the transaction format.
- [transactionssample_bycounty](#) returns all param data for county in state between bdate and edate in the transaction format.
- [transactionssample_bystate](#) returns all param data in state between bdate and edate in the transaction format.

- `transactionssample_byma` returns all param data in agency (monitoring agency) between `bdate` and `edate` in the transaction format.

Value

A data.frame containing parsed data or a named list containing header and data.

Examples

```
## Not run:

## Set your API Key first using set_aqs_user to run the following codes

## Example from the AQS website
## all benzene samples from North Carolina collected on May 15th, 1995
aqs_variables <- list(
  param = "45201", bdate = "19950515", edate = "19950515",
  state = "37"
)
aqs_transactionssample(
  aqs_filter = "byState", aqs_variables = aqs_variables
)

## Equivalent to above; used integers instead of strings
transactionssample_bystate(
  param = 45201, bdate = "19950515", edate = "19950515",
  state = 37
)

## End(Not run)
```

raqs_options

Package options

Description

The following package options can be set via [options](#) and queried via [getOption](#).

Options to handle the AQS API rate limits

The AQS API recommends not to make more than 10 requests per minute and pause 5 seconds between requests.

- `raqs.req_per_min` controls the maximum number of API requests per minute. Default is 10.
- `raqs.delay_between_req` controls a delay between API requests sent via a function when your `bdate` and `edate` inputs span multiple years. A value will be rounded to the nearest integer. Default is 5 seconds.
- `raqs.delay_fun_exit` controls a delay before a function execution ends. A value will be rounded to the nearest integer. Default is zero if R is being used interactively. Otherwise, it is 5 seconds. This option only applies to functions that send API requests.

Option to handle the type of data object to return

By default, the parsed data will be returned as a [data.frame](#) object, but can be adjusted for users' preferences.

- `raqs.return_type` controls the type of data object to return. Default is "data.frame" but it can also be set to "tibble" or "data.table".

Examples

```
## Change for the duration of the session
op <- options(raqs.rep_per_min = 5)

## Change back to the original value
options(op)
```

| | |
|--------------|-------------------------------------|
| set_aqs_user | <i>Set your AQS API credentials</i> |
|--------------|-------------------------------------|

Description

Set your registered email and key as environmental variables for the current session. Please sign up first using [aqs_signup](#) if you haven't set up an account on the AQS API. If you want to set your email and key permanently, please add the following lines in your `.Renviron` file:

- `AQS_EMAIL = YOUR REGISTERED EMAIL`
- `AQS_KEY = YOUR API KEY`

Usage

```
set_aqs_user(email, key)

get_aqs_user()

get_aqs_email()

get_aqs_key()
```

Arguments

| | |
|-------|---|
| email | A string specifying your registered email address |
| key | A string specifying your API key |

Details

[set_aqs_user](#) sets your API credentials for the current session. [get_aqs_user](#), [get_aqs_email](#), and [get_aqs_key](#) are helper functions to display saved user values.

Value

No return value, called to set environmental variables

See Also

See [aqs_signup](#) to create an account for the AQS API

Examples

```
## Please use your registered email and key
set_aqs_user(email = "your@registered.email", key = "your_api_key")

## Show your API credentials
get_aqs_user() # return list(email, key)
get_aqs_email() # return email
get_aqs_key() # return key
```

Index

annualdata_bybox, [6](#)
annualdata_bybox (aqs_annualdata), [3](#)
annualdata_bycbsa, [6](#)
annualdata_bycbsa (aqs_annualdata), [3](#)
annualdata_bycounty, [6](#)
annualdata_bycounty (aqs_annualdata), [3](#)
annualdata_bysite, [6](#)
annualdata_bysite (aqs_annualdata), [3](#)
annualdata_bystate, [6](#)
annualdata_bystate (aqs_annualdata), [3](#)
aqs_annualdata, [2](#), [3](#), [6](#), [42](#)
aqs_dailydata, [2](#), [7](#), [10](#), [42](#)
aqs_list, [2](#), [11](#), [13](#)
aqs_metadata, [2](#), [14](#), [15](#)
aqs_monitors, [2](#), [16](#), [18](#)
aqs_qaannualperformanceevaluations, [2](#),
[19](#), [22](#)
aqs_qablanks, [2](#), [23](#), [25](#)
aqs_qacolloccatedassessments, [2](#), [26](#), [28](#)
aqs_qaflowrateaudits, [3](#), [29](#), [31](#)
aqs_qaflowrateverifications, [3](#), [32](#), [35](#)
aqs_qaonepointqcrawdata, [3](#), [36](#), [38](#)
aqs_qapepaudits, [3](#), [39](#), [41](#)
aqs_quarterlydata, [2](#), [42](#), [45](#)
aqs_sampledata, [2](#), [42](#), [46](#), [49](#)
aqs_signup, [2](#), [50](#), [58](#), [59](#)
aqs_transactionsqaannualperformanceevaluations,
[3](#), [51](#), [53](#)
aqs_transactionssample, [3](#), [54](#), [56](#)

dailydata_bybox, [10](#)
dailydata_bybox (aqs_dailydata), [7](#)
dailydata_bycbsa, [10](#)
dailydata_bycbsa (aqs_dailydata), [7](#)
dailydata_bycounty, [10](#)
dailydata_bycounty (aqs_dailydata), [7](#)
dailydata_bysite, [10](#)
dailydata_bysite (aqs_dailydata), [7](#)
dailydata_bystate, [10](#)
dailydata_bystate (aqs_dailydata), [7](#)

data.frame, [58](#)

get_aqs_email, [58](#)
get_aqs_email (set_aqs_user), [58](#)
get_aqs_key, [58](#)
get_aqs_key (set_aqs_user), [58](#)
get_aqs_user, [58](#)
get_aqs_user (set_aqs_user), [58](#)
getOption, [57](#)

list_cbsas, [6](#), [10](#), [13](#), [18](#), [45](#), [49](#)
list_cbsas (aqs_list), [11](#)
list_classes, [13](#)
list_classes (aqs_list), [11](#)
list_countiesbystate, [6](#), [10](#), [13](#), [18](#), [21](#), [25](#),
[28](#), [31](#), [34](#), [38](#), [41](#), [45](#), [49](#), [53](#), [56](#)
list_countiesbystate (aqs_list), [11](#)
list_durations, [6](#), [10](#), [13](#), [49](#)
list_durations (aqs_list), [11](#)
list_mas, [13](#), [22](#), [25](#), [28](#), [31](#), [35](#), [38](#), [41](#), [53](#), [56](#)
list_mas (aqs_list), [11](#)
list_parametersbyclass, [5](#), [9](#), [13](#), [18](#), [21](#),
[24](#), [28](#), [31](#), [34](#), [37](#), [41](#), [44](#), [48](#), [53](#), [56](#)
list_parametersbyclass (aqs_list), [11](#)
list_pqaos, [13](#), [21](#), [25](#), [28](#), [31](#), [35](#), [38](#), [41](#), [53](#)
list_pqaos (aqs_list), [11](#)
list_sitesbycounty, [6](#), [10](#), [13](#), [18](#), [21](#), [25](#),
[28](#), [31](#), [34](#), [38](#), [41](#), [45](#), [49](#), [53](#), [56](#)
list_sitesbycounty (aqs_list), [11](#)
list_states, [5](#), [9](#), [13](#), [18](#), [21](#), [25](#), [28](#), [31](#), [34](#),
[38](#), [41](#), [45](#), [49](#), [53](#), [56](#)
list_states (aqs_list), [11](#)

metadata_fieldsbyservice, [15](#)
metadata_fieldsbyservice
(aqs_metadata), [14](#)
metadata_isavailable, [15](#)
metadata_isavailable (aqs_metadata), [14](#)
metadata_issues, [15](#)
metadata_issues (aqs_metadata), [14](#)

- metadata_revisionhistory, [15](#)
- metadata_revisionhistory
 - (aqs_metadata), [14](#)
- monitors_bybox, [19](#)
- monitors_bybox (aqs_monitors), [16](#)
- monitors_bycbasa, [19](#)
- monitors_bycbasa (aqs_monitors), [16](#)
- monitors_bycounty, [19](#)
- monitors_bycounty (aqs_monitors), [16](#)
- monitors_bysite, [18](#)
- monitors_bysite (aqs_monitors), [16](#)
- monitors_bystate, [19](#)
- monitors_bystate (aqs_monitors), [16](#)
- options, [57](#)
- qaannualperformanceevaluations_bycounty, [22](#)
- qaannualperformanceevaluations_bycounty
 - (aqs_qaannualperformanceevaluations), [19](#)
- qaannualperformanceevaluations_byma, [22](#)
- qaannualperformanceevaluations_byma
 - (aqs_qaannualperformanceevaluations), [19](#)
- qaannualperformanceevaluations_bypqao, [22](#)
- qaannualperformanceevaluations_bypqao
 - (aqs_qaannualperformanceevaluations), [19](#)
- qaannualperformanceevaluations_bysite, [22](#)
- qaannualperformanceevaluations_bysite
 - (aqs_qaannualperformanceevaluations), [19](#)
- qaannualperformanceevaluations_bystate, [22](#)
- qaannualperformanceevaluations_bystate
 - (aqs_qaannualperformanceevaluations), [19](#)
- qablanks_bycounty, [25](#)
- qablanks_bycounty (aqs_qablanks), [23](#)
- qablanks_byma, [25](#)
- qablanks_byma (aqs_qablanks), [23](#)
- qablanks_bypqao, [25](#)
- qablanks_bypqao (aqs_qablanks), [23](#)
- qablanks_bysite, [25](#)
- qablanks_bysite (aqs_qablanks), [23](#)
- qablanks_bystate, [25](#)
- qablanks_bystate (aqs_qablanks), [23](#)
- qaccollocatedassessments_bycounty, [28](#)
- qaccollocatedassessments_bycounty
 - (aqs_qaccollocatedassessments), [26](#)
- qaccollocatedassessments_byma, [28](#)
- qaccollocatedassessments_byma
 - (aqs_qaccollocatedassessments), [26](#)
- qaccollocatedassessments_bypqao, [28](#)
- qaccollocatedassessments_bypqao
 - (aqs_qaccollocatedassessments), [26](#)
- qaccollocatedassessments_bysite, [28](#)
- qaccollocatedassessments_bysite
 - (aqs_qaccollocatedassessments), [26](#)
- qaccollocatedassessments_bystate, [28](#)
- qaccollocatedassessments_bystate
 - (aqs_qaccollocatedassessments), [26](#)
- qaflowrateaudits_bycounty
 - (aqs_qaflowrateaudits), [29](#)
- qaflowrateaudits_byma, [32, 38](#)
- qaflowrateaudits_byma
 - (aqs_qaflowrateaudits), [29](#)
- qaflowrateaudits_bypqao, [32](#)
- qaflowrateaudits_bypqao
 - (aqs_qaflowrateaudits), [29](#)
- qaflowrateaudits_bysite, [31](#)
- qaflowrateaudits_bysite
 - (aqs_qaflowrateaudits), [29](#)
- qaflowrateaudits_bystate, [32](#)
- qaflowrateaudits_bystate
 - (aqs_qaflowrateaudits), [29](#)
- qaflowrateverifications_bycounty, [32, 35](#)
- qaflowrateverifications_bycounty
 - (aqs_qaflowrateverifications), [32](#)
- qaflowrateverifications_byma, [35](#)
- qaflowrateverifications_byma
 - (aqs_qaflowrateverifications), [32](#)
- qaflowrateverifications_bypqao, [35](#)
- qaflowrateverifications_bypqao
 - (aqs_qaflowrateverifications),

- 32
- qaflowrateverifications_bysite, 35
- qaflowrateverifications_bysite
(aqs_qaflowrateverifications),
32
- qaflowrateverifications_bystate, 35
- qaflowrateverifications_bystate
(aqs_qaflowrateverifications),
32
- qaonepointqcrawdata_bycounty, 38
- qaonepointqcrawdata_bycounty
(aqs_qaonepointqcrawdata), 36
- qaonepointqcrawdata_byma
(aqs_qaonepointqcrawdata), 36
- qaonepointqcrawdata_bypqao, 38
- qaonepointqcrawdata_bypqao
(aqs_qaonepointqcrawdata), 36
- qaonepointqcrawdata_bysite, 38
- qaonepointqcrawdata_bysite
(aqs_qaonepointqcrawdata), 36
- qaonepointqcrawdata_bystate, 38
- qaonepointqcrawdata_bystate
(aqs_qaonepointqcrawdata), 36
- qaepaudits_bycounty, 42
- qaepaudits_bycounty (aqs_qaepaudits),
39
- qaepaudits_byma, 42
- qaepaudits_byma (aqs_qaepaudits), 39
- qaepaudits_bypqao, 42
- qaepaudits_bypqao (aqs_qaepaudits), 39
- qaepaudits_bysite, 41
- qaepaudits_bysite (aqs_qaepaudits), 39
- qaepaudits_bystate, 42
- qaepaudits_bystate (aqs_qaepaudits),
39
- quarterlydata_bybox, 46
- quarterlydata_bybox
(aqs_quarterlydata), 42
- quarterlydata_bycbsa, 46
- quarterlydata_bycbsa
(aqs_quarterlydata), 42
- quarterlydata_bycounty, 45
- quarterlydata_bycounty
(aqs_quarterlydata), 42
- quarterlydata_bysite, 45
- quarterlydata_bysite
(aqs_quarterlydata), 42
- quarterlydata_bystate, 45
- quarterlydata_bystate
(aqs_quarterlydata), 42
- quarterlydata_bystate
(aqs_quarterlydata), 42
- raqs (raqs-package), 2
- raqs-package, 2
- raqs_options, 57
- sampledata_bybox, 49
- sampledata_bybox (aqs_sampledata), 46
- sampledata_bycbsa, 50
- sampledata_bycbsa (aqs_sampledata), 46
- sampledata_bycounty, 49
- sampledata_bycounty (aqs_sampledata), 46
- sampledata_bysite, 49
- sampledata_bysite (aqs_sampledata), 46
- sampledata_bystate, 49
- sampledata_bystate (aqs_sampledata), 46
- set_aqs_user, 2, 6, 10, 12, 13, 15, 18, 21, 25,
28, 31, 35, 38, 41, 45, 49, 51, 53, 56,
58, 58
- transactionsqaannualperformanceevaluations_bycounty,
53
- transactionsqaannualperformanceevaluations_bycounty
(aqs_transactionsqaannualperformanceevaluations),
51
- transactionsqaannualperformanceevaluations_byma,
54
- transactionsqaannualperformanceevaluations_byma
(aqs_transactionsqaannualperformanceevaluations),
51
- transactionsqaannualperformanceevaluations_bypqao,
54
- transactionsqaannualperformanceevaluations_bypqao
(aqs_transactionsqaannualperformanceevaluations),
51
- transactionsqaannualperformanceevaluations_bysite,
53
- transactionsqaannualperformanceevaluations_bysite
(aqs_transactionsqaannualperformanceevaluations),
51
- transactionsqaannualperformanceevaluations_bystate,
53
- transactionsqaannualperformanceevaluations_bystate
(aqs_transactionsqaannualperformanceevaluations),
51
- transactionssample_bycounty, 56
- transactionssample_bycounty
(aqs_transactionssample), 54

transactionssample_byma, [57](#)
transactionssample_byma
 (aqs_transactionssample), [54](#)
transactionssample_bysite, [56](#)
transactionssample_bysite
 (aqs_transactionssample), [54](#)
transactionssample_bystate, [56](#)
transactionssample_bystate
 (aqs_transactionssample), [54](#)