Package: esri2sf (via r-universe)

August 27, 2024

Type Package

Title Create Simple Features from ArcGIS Server REST API

Version 0.2.0.9000

Maintainer Eli Pousson <eli.pousson@gmail.com>

Description This package enables you to scrape geographic features directly from ArcGIS servers REST API into R as simple features.

License MIT + file LICENSE

URL https://github.com/elipousson/esri2sf/

BugReports https://github.com/elipousson/esri2sf/issues

Depends R (>= 3.1.0)

Imports cli, DBI, dplyr, glue, httr2 (>= 0.2.3), jsonlite, rappdirs, rlang (>= 1.1.0), sf (>= 1.0.1), stats, tibble, vctrs

Suggests arcgisutils (>= 0.1.1.9000), cliExtras (>= 0.1.0), keyring, knitr, magick, rmarkdown, terra, testthat (>= 3.0.0), xml2

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

LazyData true

Remotes elipousson/cliExtras, r-ArcGIS/arcgisutils

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

RemoteUrl https://github.com/elipousson/esri2sf

RemoteRef HEAD

RemoteSha c9a535d13d30560f70d8da326fb05cca7928c0dc

Contents

`addDomainInfo` *Add domain information to the return (sf) dataframe*

Description

Uses the esrimeta function to get information about the field domains in the returned dataframe and joins the domain information to the dataframe. This is especially important for the codedValue type domains that are essentially factors and you loose information without replacing the data.

Usage

```
addDomainInfo(df, url, token = NULL, call = caller_env())
```

Arguments

`df`	The returned (sf) df from esri2sf/esri2df.
`url`	The url for the Map/Feature server layer/table.
`token`	string for authentication token (if needed).
`call`	The execution environment of a currently running function, e.g. `caller_env()`. The function will be mentioned in error messages as the source of the error. See the `call` argument of `abort()` for more information.

Value

An sf dataframe

`esri2rast` *Import data from ArcGIS ImageServer url using the exportImage API*

Description

See the ArcGIS REST API documentation for more information on the exportImage API https://developers.arcgis.com/rest/services-reference/enterprise/export-image.htm

Usage

```
esri2rast(
  url,
  bbox = NULL,
  token = NULL,
  format = "jpgpng",
  adjustAspectRatio = FALSE,
  ...
)
```

Arguments

`url`	ImageServer url
`bbox`	Bounding box for image to return; defaults to `NULL`.
`token`	defaults to `NULL`.
`format`	defaults to "jpgpng". Options include "jpgpng", "png", "png8", "png24", "jpg", "bmp", "gif", "tiff", "png32", "bip", "bsq", and "lerc"
`adjustAspectRatio`	defaults to `FALSE`

Value

SpatRaster object from `terra::rast`

esri2sf *Get data or metadata from an ArcGIS MapServer or FeatureServer*

Description

These functions are the interface to the user.

Usage

```
esri2sf(
  url,
  outFields = NULL,
  where = NULL,
  geometry = NULL,
  bbox = NULL,
  token = NULL,
  crs = getOption("esri2sf.crs", 4326),
  progress = FALSE,
  geomType = NULL,
  spatialRel = NULL,
  replaceDomainInfo = FALSE,
  .name_repair = "check_unique",
  quiet = FALSE,
  ...
)

esri2df(
  url,
  outFields = NULL,
  where = NULL,
  token = NULL,
  progress = FALSE,
  replaceDomainInfo = FALSE,
  .name_repair = "check_unique",
  quiet = FALSE,
  ...
)

esrigroup(
  url,
  layerInfo = NULL,
  outFields = NULL,
  where = NULL,
  geometry = NULL,
  bbox = NULL,
  token = NULL,
  crs = getOption("esri2sf.crs", 4326),
```

```
  progress = TRUE,
  geomType = NULL,
  spatialRel = NULL,
  replaceDomainInfo = FALSE,
  .name_repair = "check_unique",
  quiet = FALSE,
  .fn = esri2sf,
  ...,
  call = caller_env()
)
```

Arguments

`url`	A service url, e.g. https://sampleserver6.arcgisonline.com/arcgis/rest/services/Census/MapServer/2 or an ArcGIS Online item url if the item contains a single feature or table layer.
`outFields`	vector of fields you want to include. default is `NULL` for all fields.
`where`	string for where condition. Default is `NULL` (equivalent to `1=1`) to return all rows.
`geometry`	An `sf` or `bbox` object. Currently, `sf` objects with a single POINT feature are supported. All other `sf` objects are converted to `bbox` objects.
`bbox`	bbox class object from `sf::st_bbox()` or a simple feature object that can be converted to a bounding box.
`token`	string for authentication token. defaults to `NULL`.
`crs`	coordinate reference system (see `sf::st_sf()`). Should either be `NULL` or a CRS that can be handled by GDAL through sf::st_sf(). Default is `getOption("esri2sf.crs", 4326)` which sets the CRS to EPSG:4326 if no option is set. If CRS is `NULL` feature is returned with the same CRS that the layer is hosted as in the Feature/Map Server.
`progress`	Show progress bar from `cli::cli_progress_along()` if `TRUE`. Default `FALSE`.
`geomType`	string specifying the layer geometry ('esriGeometryPolygon' or 'esriGeometryPoint' or 'esriGeometryPolyline' - if `NULL`, will try to be inferred from the server)
`spatialRel`	Spatial relationship applied to the input `geometry` when performing the query; defaults to `NULL` (equivalent to "esriSpatialRelIntersects"). Additional supported options include "esriSpatialRelContains", "esriSpatialRelCrosses", "esriSpatialRelEnvelopeIntersects", "esriSpatialRelIndexIntersects", "esriSpatialRelOverlaps", "esriSpatialRelTouches", "esriSpatialRelWithin"
`replaceDomainInfo`	If `TRUE`, add domain information to the return data frame. Default `FALSE`.
`.name_repair`	Treatment of problematic column names: • `"minimal"`: No name repair or checks, beyond basic existence, • `"unique"`: Make sure names are unique and not empty, • `"check_unique"`: (default value), no name repair, but check they are `unique`, • `"universal"`: Make the names `unique` and syntactic

- a function: apply custom name repair (e.g., `.name_repair = make.names` for names in the style of base R).
- A purrr-style anonymous function, see `rlang::as_function()`

This argument is passed on as `repair` to `vctrs::vec_as_names()`. See there for more details on these terms and the strategies used to enforce them.

`quiet` If `TRUE`, use `suppressMessages()` to prevent the printing of messages about the requested layer. Defaults to `FALSE`.

`...` additional named parameters to pass to the query. (e.g. `"resultRecordCount = 3"`). See the ArcGIS REST APIs documentation for more information on all supported parameters.

Value

simple feature (`esri2sf`) or tibble (`esri2df`) or list or tibble (`esrimeta`).

Functions

- `esri2sf()`: Retrieve spatial object
- `esri2df()`: Retrieve table object (no spatial data).

Note

When accessing services with multiple layers, the layer number must be specified at the end of the service url (e.g., https://sampleserver6.arcgisonline.com/arcgis/rest/services/Census/MapServer/2). #' The list of layers and their respective id numbers can be found by viewing the service's url in a web browser and viewing the "Layers" heading.

Examples

```
baseURL <- "https://sampleserver6.arcgisonline.com/arcgis/rest/"
url <- paste0(baseURL, "services/Census/MapServer/2")
outFields <- c("POP2007", "POP2000")
where <- "STATE_NAME = 'Michigan'"
df <- esri2sf(url, outFields = outFields, where = where)
plot(df)
```

---

esriCatalog *Get information on folders, services, tables, and layers using the Catalog service*

Description

The Catalog resource from the ArcGIS REST API represents a catalog of folders and services published on the host. More information: https://developers.arcgis.com/rest/services-reference/enterprise/catalog.htm

Usage

```
esriCatalog(
  url,
  f = "json",
  token = NULL,
  option = NULL,
  outSR = NULL,
  ...,
  call = caller_env()
)

esricatalog(
  url,
  f = "json",
  token = NULL,
  option = NULL,
  outSR = NULL,
  ...,
  call = caller_env()
)
```

Arguments

url	A folder, service, or layer URL that can be used with the ArcGIS REST API.
f	Format to use for request. Supported options include "json", "sitemap", or "geositemap"; "html" and "kmz" are not currently supported.
token	String for authentication token; defaults to NULL.
option	If option = "footprints" and the url is for a folder, spatial footprints of all map, feature, and image services in that folder are returned as a feature collection
outSR	Output spatial reference of the geometry returned in footprints; only supported when option = "footprints".
...	Additional parameters passed to httr2::req_url_query
call	The execution environment of a currently running function, e.g. caller_env(). The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.

---

esrigeocode *Use an ArcGIS GeocodeServer to geocode an address or reverse geocode coordinates*

Description

This function allows the use of an ArcGIS GeocodeServer url to support the Find Address Candidates and Reverse Geocode REST APIs. Provide an address parameter to use Find Address Candidates or a coords parameter to use Reverse Geocode.

Usage

```
esrigeocode(
  url,
  address = NULL,
  coords = NULL,
  score = 0.95,
  n = 1,
  token = NULL,
  crs = getOption("esri2sf.crs", 4326),
  geometry = TRUE,
  ...,
  call = caller_env()
)
```

Arguments

url	A GeocodeServer service url. Required.
address	Single line address passed at the "SingleLine" parameter to the ArcGIS REST API. Specific format may depend on specific server configuration, Default: NULL
coords	Numeric vector with longitude, latitude coordinates or a sf object where the centroid is used as the coordinates. Default: NULL
score	Accuracy score, if provided only return results with provided accuracy score or greater Default: 0.95
n	Number of candidates to return, Default: 1
token	Token, Default: NULL
crs	Coordinate reference system to return.
geometry	If TRUE (default), return a simple feature object. If FALSE, return a data frame.
...	Additional parameters passed to esriRequest().
call	The execution environment of a currently running function, e.g. caller_env(). The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.

---

esrihub *Get a feed of available data from an ArcGIS Hub site*

Description

Get a feed in a DCAT US, DCAT AP, RSS, or OGC Definitions format of the public data or other content published on an ArcGIS Hub site.

Usage

```
esrihub(
  url,
  format = c("dcat-us", "dcat-ap", "rss", "ogc"),
  simplifyVector = TRUE,
  call = caller_env()
)
```

Arguments

url	A url for an ArcGIS Hub site
format	Format of data to return, See https://doc.arcgis.com/en/hub/content/federate-data-with-external-catalogs.htm for more information. Default: c("dcat-us", "dcat-ap", "rss", "ogc").
simplifyVector	Passed to httr2::resp_body_json() if format is not rss. Default: TRUE
call	The execution environment of a currently running function, e.g. caller_env(). The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.

Value

A list of metadata for the public content on the ArcGIS Hub site.

---

esriIndex *Create an index of folders, services, layers, and tables for an ArcGIS Server*

Description

Recurse over a ArcGIS Server url or folder url to return a data frame index of folders, services, layers, and tables. This function returns additional information than esriCatalog() using f = "sitemap" or f = "geositemap".

Usage

```
esriIndex(
  url,
  folderPath = NULL,
  serviceName = NULL,
  recurse = FALSE,
  token = NULL,
  ...
)

esriIndexLayers(url, folderPath = NULL, serviceName = NULL, token = NULL, ...)
```

```
esriindex(
  url,
  folderPath = NULL,
  serviceName = NULL,
  recurse = FALSE,
  token = NULL,
  ...
)
```

Arguments

`url`	URL for ArcGIS server, folder, or service.
`folderPath, serviceName`	Name of parent folder or service; only used internally (not intended for user).
`recurse`	If `TRUE`, recursively check folders and services to return an index that includes services in folders, subfolders, layers, and tables. Defaults to `FALSE`. If `url` starts with "https://services.arcgis.com", emit a warning and recurse is set to `FALSE`.
`token`	String for authentication token; defaults to `NULL`.
`...`	Additional parameters passed to `esriCatalog()` (not typically required)

---

esriInfo *Get info from a service url*

Description

Return server info, service item info, metadata, or thumbnail. The magick package is required to return a thumbnail.

Usage

```
esriInfo(url, info = NULL, format = NULL, token = NULL, ...)
```

Arguments

`url`	A folder, service, or layer URL that can be used with the ArcGIS REST API.
`info`	Info service to use. Options include "info", "item", "metadata", or "thumbnail". If info = "info" (default) basic server information is displayed and the body of the response is returned invisibly.
`format`	If `info = "metadata"`, options include "fgdc" or "iso19139"; no format is required if `info` is "item" or "thumbnail".
`token`	String for authentication token; defaults to `NULL`.
`...`	Additional parameters passed to httr2::req_url_query

Details

Additional documentation: https://developers.arcgis.com/rest/services-reference/enterprise/info.htm

esriitem *Get ESRI item data or metadata*

Description

esriitem() provides partial support from the ArcGIS Content API.

Usage

```
esriitem(
  url,
  type = "data",
  destfile = tempfile(fileext = "pdf"),
  simplifyVector = TRUE,
  ...
)
```

Arguments

type	"data", "info", "metadata", "config" (app URLs only)
destfile	Destination file used to download item if data is a PDF file.
simplifyVector	Should JSON arrays containing only primitives (i.e. booleans, numbers, and strings) be caused to atomic vectors?
...	Arguments passed on to httr2::resp_body_xml resp A response object. check_type Check that response has expected content type? Set to FALSE to suppress the automated check

Value

A list, a xml document, or the response object from the request.

esriLayers *Get All Layers and Tables for the whole service*

Description

Retrieve JSON for all layers and tables as specified in https://si-pweb-vecmap.si.edu/vectormap/sdk/rest/index.html#/All_Layers_and_Tables/02ss0000005t000000/. Service type must either be a MapServer or FeatureServer. Performs a request towards the server url in one of the following the forms:

- https://<host>/<instance>/rest/services/<folderName>/serviceName>/MapServer/layers
- https://<host>/<instance>/rest/services/serviceName>/MapServer/layers
- https://<host>/<instance>/rest/services/<folderName>/serviceName>/FeatureServer/layers
- https://<host>/<instance>/rest/services/serviceName>/FeatureServer/layers

Usage

```
esriLayers(
  url,
  token = NULL,
  returnUpdates = NULL,
  returnDomainNames = TRUE,
  ...,
  call = caller_env()
)
```

Arguments

`url`	The url for the Map/Feature server. If given a url specifying a layer or table ID it will truncate it.
`token`	String for authentication token; defaults to `NULL`.
`returnUpdates`	If `TRUE`, updated features will be returned; defaults to `NULL`.
`returnDomainNames`	If `TRUE`, the REST API response does not include the full domain information for each layer (only the domain names). Defaults to `TRUE`.
`...`	Additional parameters passed to httr2::resp_body_json
`call`	The execution environment of a currently running function, e.g. `caller_env()`. The function will be mentioned in error messages as the source of the error. See the `call` argument of `abort()` for more information.

Value

A list from the JSON return.

`esrimeta` *Retrieve layer metadata*

Description

Retrieve layer metadata

Usage

```
esrimeta(url, token = NULL, fields = FALSE, ..., call = caller_env())
```

Arguments

`url`	url to retrieve metadata for.
`token`	String for authentication token; defaults to `NULL`.
`fields`	`esrimeta` returns data frame with fields if `TRUE`. Default `FALSE`.
`...`	Additional parameters passed to httr2::req_url_query

call	The execution environment of a currently running function, e.g. caller_env(). The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.

---

esriRequest *Use httr2 to create a request for the ArcGIS REST API*

Description

This function is primarily useful for the development of new functions to access the ArcGIS REST API. Most users do not need to use this function directly.

Usage

```
esriRequest(
  url,
  append = NULL,
  f = NULL,
  format = NULL,
  objectIds = NULL,
  token = NULL,
  .perform = TRUE,
  .cache = FALSE,
  .max_seconds = 3,
  .is_error = TRUE,
  .body_form = FALSE,
  ...,
  call = caller_env()
)
```

Arguments

url	A folder, service, or layer URL that can be used with the ArcGIS REST API.
append	String to append to url using httr2::req_url_path_append; defaults to NULL.
f, format	Return format to use as query parameter with httr2::req_url_query; defaults to "json".
objectIds	Parameter used for layer query requests. The addition of objectIds to the query often leads the url length to exceed the 2048 character maximum. In those cases, the query is added to the body of the request with httr2::req_body_form
token	String for authentication token; defaults to NULL.
.perform	If TRUE, perform the request with httr2::req_perform and return the response. If FALSE, return the request.
.cache	If TRUE, pass a cache folder path created with rappdirs::user_cache_dir and esri2sf package to the path parameter of httr2::req_cache.
.max_seconds	Passed to max_seconds parameter of httr2::req_retry

.is_error	If FALSE, .is_error is passed to the is_error parameter of httr2::req_error function. If TRUE, the request does not use httr2::req_error.
.body_form	If objectIds is NULL and .body_form is TRUE, generate the request using `httr2::req_body_form()`. Defaults to FALSE.
...	Additional parameters passed to httr2::req_url_query
call	The execution environment of a currently running function, e.g. `caller_env()`. The function will be mentioned in error messages as the source of the error. See the `call` argument of `abort()` for more information.

`esrisearch` *Search for items from ArcGIS Online or an ArcGIS Enterprise Server*

Description

Use a query or bounding box to search for items on ArcGIS Online or on an ArcGIS Enterprise Server. See https://developers.arcgis.com/rest/users-groups-and-items/search.htm for more information on the Search endpoint for the Portal Directory API (also known as the Sharing API). A query or bbox argument must be provided.

Usage

```
esrisearch(
  query = NULL,
  bbox = NULL,
  url = NULL,
  num = 50,
  start = 1,
  category_filter = NULL,
  sort = NULL,
  desc = FALSE,
  quiet = FALSE
)
```

Arguments

query	Search terms, Default: NULL. Required if bbox is NULL.
bbox	A bbox, sf, or sfc to use as a spatial filter for search results. sf or sfc objects are converted to bounding boxes using `sf::st_bbox()`. Default: NULL. Required if query is NULL.
url	ArcGIS Enterprise request URL (e.g. "https://machine.domain.com/webadaptor/") or ArcGIS Online organization url (e.g. "https://org.arcgis.com"). If NULL (default), url is set to "https://www.arcgis.com"
num	Maximum number of results to return, Default: 50 (must be between 1 and 100).
start	Start number for returned results, Default: 1 (returns results starting from the first result).

category_filter
: Terms to use in searching for items with matching categories. Only 3 or less terms currently supported, Default: NULL

sort
: Field to use for results sort order. Options include "modified", "title", "created", "type", "owner", "avgrating", "numratings", "numcomments", "numviews". Default: NULL.

desc
: If TRUE, return results in descending order. If FALSE (default), return results in ascending order. Ignored if sort is NULL.

quiet
: If TRUE, suppress warnings and informational messages. Defaults to FALSE.

Value

A tibble data.frame with results from the item search.

Examples

```
## Not run:
if (interactive()) {
  esrisearch(query = "park")

  esrisearch(query = c("ocean", "basemap"))

  nc <- sf::st_read(system.file("shape/nc.shp", package = "sf"))

  esrisearch(bbox = nc)
}

## End(Not run)
```

---

esriUrl_isValid *Validate or parse the parts of a ESRI REST Server URL*

Description

A collection of functions that pull select parts out of a ESRI Service URL. All urls should be a form similar to:

- https://<host>/<instance>/rest/services/<folderPath>/serviceName>/<serviceType>/<featureID>
- http://<host>/<instance>/rest/services/serviceName>/<serviceType>
- <host>/<instance>/rest/services/<folderPath>/serviceName>/<serviceType>
- https://<host>/<instance>/rest/services/serviceName>/<serviceType>/<featureID>
- https://<host>/<instance>/rest/services/<folderPath>
- https://<host>/<instance>/rest/services

And having these rules:

- The scheme: https:// or http:// part is optional
- The host part is the domain of the url.

- The `instance` is the first subpage after the domain in the url.
- The `/rest/services` is the second and third subpage in the url. These are standard for all ESRI REST Services.
- The `folderPath` part is optional and indicates the file structure in the REST Service. It consists of all subpages between `/rest/services/` and the `serviceName` part (if available).
- The `serviceName` part is the last subpage betore the `<serviceType>` in the url.
- The `serviceType` specifies the type of service. Currently this package works to manage the following serviceTypes: 'MapServer', 'FeatureServer', 'GPServer', 'GeocodeServer', 'GeometryServer', 'ImageServer'.
- The `featureID` is optional and specifies the layer or table in the map service.

Usage

```
esriUrl_isValid(url, token = NULL, displayReason = FALSE)

esriUrl_isValidRoot(url, token = NULL, displayReason = FALSE)

esriUrl_isValidFolder(url, token = NULL, displayReason = FALSE)

esriUrl_isValidService(url, token = NULL, displayReason = FALSE)

esriUrl_isValidID(url, token = NULL, displayReason = FALSE)

esriUrl_isValidFeature(url, token = NULL, displayReason = FALSE)

esriUrl_ServerUrl(url, token = NULL)

esriUrl_serviceUrl(url, token = NULL, call = caller_env())

esriUrl_parseUrl(url, token = NULL, call = caller_env())
```

Arguments

`url`	The url for a Map/Feature server or for a layer/table in a Map/Feature Server.
`token`	String for authentication token (if needed).
`displayReason`	Should the reason for why a url is not valid be displayed.

Value

Character string of the request part of the url.

Functions

- `esriUrl_isValid()`: Check if url is valid for an ESRI REST Service. General to include potential layer id too.
- `esriUrl_isValidRoot()`: Check if url is valid for the root of an ESRI REST Server.
- `esriUrl_isValidFolder()`: Check if url is valid for a folder of an ESRI REST Server.

- `esriUrl_isValidService()`: Check if url is valid for a Service of an ESRI REST Server. No feature ID.
- `esriUrl_isValidID()`: DEPRECATED Use esriUrl_isValidFeature
- `esriUrl_isValidFeature()`: Check if url is valid for a feature of an ESRI REST Service.
- `esriUrl_ServerUrl()`: DEPRECATED Use esriUrl_serviceUrl
- `esriUrl_serviceUrl()`: Retrieve Map/Feature Server URL
- `esriUrl_parseUrl()`: Parse Url into parts.

`esriuser` *Get ArcGIS Online user information*

Description

Get ArcGIS Online user information

Usage

```
esriuser(url = NULL, user_id = NULL)
```

Arguments

`url`	An ArcGIS Online community URL with a user ID. Optional if user_id is supplied.
`user_id`	An ArcGIS Online user ID. Optional if url is supplied.

`esri_version_ref` *ArcGIS REST API Version Reference*

Description

A reference table with links to information on the ArcGIS REST API version: https://developers.arcgis.com/rest/services-reference/enterprise/what-s-new.htm

Usage

```
esri_version_ref
```

Format

A data frame with 19 rows and 2 variables:

`version` version number

`url` url for "What's New" documentation

Details

Additional information on REST API versioning: https://developers.arcgis.com/rest/services-reference/enterprise/rest-api-versioning.htm

extract_esri_item_id *Extract an ESRI item or user id from a url*

Description

`extract_esri_item_id()` extract a id or appid value from a URL. If a group url is supplied a group ID is extracted.

`extract_esri_user_id()` extracts a user id from a profile URL or owner id from a search URL.

Usage

```
extract_esri_item_id(
  url,
  pattern = c("(?<=id=)[A-Za-z0-9]+", "(?<=appid=)[A-Za-z0-9]+"),
  collapse = "|"
)

extract_esri_user_id(
  url,
  pattern = c("(?<=user=)[A-Za-z0-9\\._\\-]+",
    "(?<=owner%3A%22)[A-Za-z0-9\\._\\-]+"),
  collapse = "|"
)
```

fmt_epoch_date *Helpers for converting epoch dates to* POSIXct *dates*

Description

`fmt_epoch_date()` converts a single numeric epoch data value to a `POSIXct` class object. `fmt_epoch_dates()` converts a numeric vector to `POSIXct` dates.

Usage

```
fmt_epoch_date(x, tz = "")

fmt_epoch_dates(x, tz = "")
```

Arguments

x	Numeric value corresponding to epoch date.
tz	a character string. The time zone specification to be used for the conversion, *if one is required*. System-specific (see time zones), but `""` is the current time zone, and `"GMT"` is UTC (Universal Time, Coordinated). Invalid values are most commonly treated as UTC, on some platforms with a warning.

generateToken *Create authorization tokens*

Description

Generate tokens for accessing credentialed ArcGIS REST Servers.

`generateToken()` can create a token from the public token endpoint `https://<host>:\<port\>/\<site\>/tokens/gener` or the admin endpoint `https://\<host\>:\<port\>/\<site\>/admin/generateToken` for ArcGIS REST Servers. See https://developers.arcgis.com/rest/services-reference/enterprise/generate-token.htm or https://developers.arcgis.com/rest/services-reference/enterprise/generate-admin-token.htm respectively for more information.

If `server` is `NULL` and `url` is provided the endpoint is `https://\<url\>/sharing/generateToken`. See https://developers.arcgis.com/rest/users-groups-and-items/generate-token.htm for more information.

`generateOAuthToken()` can create an OAuth token for ArcGIS Online Services. See `https://developers.arcgis.com/d` or https://developers.arcgis.com/documentation/mapping-apis-and-services/security/oauth-2.0/ for more information.

Usage

```
generateToken(
  server = NULL,
  url = NULL,
  uid,
  pwd = NULL,
  type = NULL,
  client = "requestip",
  expiration = 5000,
  referer = NULL,
  ip = NULL,
  ...
)

generateOAuthToken(clientId, clientSecret, expiration = 5000, ...)
```

Arguments

`server`	The ArcGIS REST Server you want to connect to: `https://\<host\>:\<port\>`
`url`	The url to connect to using `https://\<url\>/sharing/generateToken`.
`uid`	The user id (username) of the account used to create the token connection to the server
`pwd`	The password of the account used to create the token connection to the server. If `NULL`, you will be prompted for the password.
`type`	Either 'tokens' or 'admin'. Specify the endpoint you use to create the token. Defaults to 'tokens' if server is provided.

client	"requestip" (default), "referer", or "ip"
expiration	Set an expiration limit on the token in minutes. Max expiration date may be controlled by the server.
referer, ip	Additional parameters required if client is "referer" or "ip"
...	Additional parameters passed to esriRequest()
clientId	Client ID
clientSecret	Client Secret

Value

Character string with the token

Functions

- generateToken(): Create ArcGIS REST Service Token
- generateOAuthToken(): Create ArcGIS OAuth Token

---

glue_ansi_sql *Helpers for creating ANSI SQL queries*

Description

Helpers for creating ANSI SQL queries

Usage

```
glue_ansi_sql(..., .con = DBI::ANSI(), .envir = parent.frame())

glue_sql_bbox(bbox, coords = c("longitude", "latitude"), crs = 4326)
```

Arguments

...	Additional parameters passed to glue::glue_sql()
.con	[DBIConnection]: A DBI connection object obtained from DBI::dbConnect().
.envir	[environment: parent.frame()] Environment to evaluate each expression in. Expressions are evaluated from left to right. If .x is an environment, the expressions are evaluated in that environment and .envir is ignored. If NULL is passed, it is equivalent to emptyenv().
bbox	A bbox, sfc, or sf object to use as bounding box.
coords	Column names with longitude and latitude values.
crs	Coordinate reference system used for coordinate values.

Examples

```
values <- c("a", "b", "c")

glue_ansi_sql("letter", " IN ({values*})")
```

`is_esri_content_url` *Is URL an ESRI content url?*

Description

Is URL an ESRI content url?

Usage

```
is_esri_content_url(x)

is_esri_item_url(x)

is_esri_app_url(x)
```

Index