

# Package: pega (via r-universe)

May 14, 2026

**Type** Package

**Title** Pollutants Emissions and Gaseous of the Atmosphere

**Version** 0.9.8

**Date** 2026-04-27

**Description** Elaboration of fix emissions inventories, consisting in four stages, pre-processing activity data, preparing emissions factors, estimating the emissions and post-processing of emissions in maps and databases. More details in <doi:>. Before using PEGA you need to know the industrial composition of your study area, in other words, the combination of industrial type, size and fuel of the fleet. Then, it is recommended to start with the project to download a template to create a structure of directories and scripts.

**License** GPL

**URL** <https://github.com/atmoschem/pega>

**BugReports** <https://github.com/atmoschem/pega/issues>

**LazyData** no

**Depends** R (>= 3.5.0)

**Imports** sf, data.table, magrittr, jsonlite, httr

**Suggests** knitr, rmarkdown, testthat

**RoxygenNote** 7.3.3

**Encoding** UTF-8

**NeedsCompilation** no

**Config/testthat/parallel** true

**VignetteBuilder** knitr

**Config/pak/sysreqs** libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

**Repository** <https://atmoschem.r-universe.dev>

**Date/Publication** 2026-05-14 15:08:46 UTC

**RemoteUrl** <https://github.com/atmoschem/pega>

**RemoteRef** HEAD

**RemoteSha** bd17faa7b6b5b56725f8edc725693d69957b0edc

## Contents

ef	2
esri_to_sf_geom	3
esri_to_sf_point	3
esri_to_sf_polygon	4
esri_to_sf_polyline	4
get_esri_features	5
get_esri_features_by_ids	6
get_object_ids	7
inventory	8
layer_info	9
<b>Index</b>	<b>10</b>

---

ef	<i>Inventory structure</i>
----	----------------------------

---

### Description

Inventory structure

### Usage

```
ef(nfr = "1.A.1.a", fuel, pollutant, returnfdb = FALSE)
```

### Arguments

nfr	nomenclature for reporting, one or many
fuel	String for type of fuel.
pollutant	String for pollutant.
returnfdb	logical return directly db

### Value

Return a data.table

### Examples

```
{
  ef()
}
```

---

esri_to_sf_geom	<i>get features</i>
-----------------	---------------------

---

**Description**

get features.

**Usage**

```
esri_to_sf_geom(json_feats, geom_type, crs = 4326)
```

**Arguments**

json_feats	Character
geom_type	Character.
crs	CRS.

**See Also**

[layer\\_info](#)

**Examples**

```
## Not run:  
# do not run  
  
## End(Not run)
```

---

esri_to_sf_point	<i>get features</i>
------------------	---------------------

---

**Description**

get features.

**Usage**

```
esri_to_sf_point(features)
```

**Arguments**

features	Character
----------	-----------

**See Also**

[layer\\_info](#)

**Examples**

```
## Not run:  
# do not run  
  
## End(Not run)
```

---

esri\_to\_sf\_polygon    *get features*

---

**Description**

get features.

**Usage**

```
esri_to_sf_polygon(features)
```

**Arguments**

features            Character

**See Also**

[layer\\_info](#)

**Examples**

```
## Not run:  
# do not run  
  
## End(Not run)
```

---

esri\_to\_sf\_polyline    *get features*

---

**Description**

get features.

**Usage**

```
esri_to_sf_polyline(features)
```

**Arguments**

features            Character

**See Also**

[layer\\_info](#)

**Examples**

```
## Not run:  
# do not run  
  
## End(Not run)
```

---

get\_esri\_features      *get features*

---

**Description**

get features.

**Usage**

```
get_esri_features(  
  url = paste0("http://portal1.snirh.gov.br/ana/",  
    "rest/services/Esta%C3%A7%C3%B5es_da_",  
    "Rede_Hidrometeorol%C3%B3gica_Nacional", "_em_Opera%C3%A7%C3%A3o/MapServer/1"),  
  query_url = paste(url, "query", sep = "/"),  
  fields = c("*"),  
  where = "1=1",  
  token = ""  
)
```

**Arguments**

url	Character, default .
query_url	Character.
fields	description
where	Character.
token	Character.

**See Also**

[layer\\_info](#)

**Examples**

```
## Not run:
# do not run
 geom_type <- layer_info()$geometryType
# rede
url1 = paste0("http://portal1.snirh.gov.br/ana/",
 "rest/services/Esta%C3%A7%C3%B5es_da_",
 "Rede_Hidrometeorol%C3%B3gica_Nacional",
 "_em_Opera%C3%A7%C3%A3o/MapServer/1")
# hidroweb
url1 = paste0("http://portal1.snirh.gov.br/ana/",
 "rest/services/Esta%C3%A7%C3%B5es_da_",
 "Rede_Hidrometeorol%C3%B3gica_Nacional",
 "_em_Opera%C3%A7%C3%A3o/MapServer/4")
esri_features <- get_esri_features(url1)
simple_features <- esri_to_sf_geom(esri_features, geom_type = geom_type)

## End(Not run)
```

---

```
get_esri_features_by_ids
```

```
get features
```

---

**Description**

get features.

**Usage**

```
get_esri_features_by_ids(
  ids,
  url = paste0("http://portal1.snirh.gov.br/ana/",
    "rest/services/Esta%C3%A7%C3%B5es_da_",
    "Rede_Hidrometeorol%C3%B3gica_Nacional", "_em_Opera%C3%A7%C3%A3o/MapServer/1"),
  query_url = paste(url, "query", sep = "/"),
  fields = c("*"),
  token = "",
  ssl = FALSE,
  simplifyDataFrame = FALSE,
  simplifyVector = FALSE,
  full = FALSE
)
```

**Arguments**

ids	Integer.
url	Character.

query_url	Character.
fields	Character.
token	Character.
ssl	Logical, default = FALSE.
simplifyDataFrame	Logical, default = FALSE.
simplifyVector	Logical, default = FALSE.
full	Logical, default = FALSE.

**See Also**

[layer\\_info](#)

**Examples**

```
## Not run:
# do not run
ids <- get_object_ids()
feat <- get_esri_features_by_ids(ids = ids)

## End(Not run)
```

---

get\_object\_ids

*get layer info*

---

**Description**

get layer info.

**Usage**

```
get_object_ids(
  url = paste0("http://portal1.snirh.gov.br/ana/",
    "rest/services/Esta%C3%A7%C3%B5es_da_",
    "Rede_Hidrometeorol%C3%B3gica_Nacional", "_em_Opera%C3%A7%C3%A3o/MapServer/1"),
  query_url = paste(url, "query", sep = "/"),
  where = "1=1",
  token = "",
  ssl = FALSE,
  full = FALSE
)
```

**Arguments**

url	Character .
query_url	Character.
where	Character.
token	Character.
ssl	Logical, default = FALSE.
full	Logical, default = FALSE.

**See Also**

[layer\\_info](#)

**Examples**

```
## Not run:
# do not run
a <- get_object_ids()
head(a)
a <- get_object_ids(full = T)
names(a)
a$objectIds

## End(Not run)
```

---

inventory	<i>start inventory structure</i>
-----------	----------------------------------

---

**Description**

create directories

**Usage**

```
inventory(path, nfr = "all", dict, return_dict = TRUE)
```

**Arguments**

path	Character, default .
nfr	String, nomenclature for ereporting, default "all"
dict	String with file name to add description notes in a csv.
return_dict	Logical, to return dictionary as object

**Examples**

```
## Not run:
# do not run

## End(Not run)
```

---

layer_info	<i>get layer info</i>
------------	-----------------------

---

### Description

get layer info.

### Usage

```
layer_info(  
  url = paste0("http://portal1.snirh.gov.br/ana/",  
    "rest/services/Esta%C3%A7%C3%B5es_da_",  
    "Rede_Hidrometeorol%C3%B3gica_Nacional", "_em_Opera%C3%A7%C3%A3o/MapServer/1"),  
  token = "",  
  ssl = FALSE  
)
```

### Arguments

url	Character, default .
token	Character.
ssl	Logical, default = FALSE.

### See Also

[layer\\_info](#)

### Examples

```
## Not run:  
# do not run  
a <- layer_info()  
names(a)  
a$geometryType  
  
## End(Not run)
```

# Index

\* **emission**

ef, [2](#)

\* **factors**

ef, [2](#)

ef, [2](#)

esri\_to\_sf\_geom, [3](#)

esri\_to\_sf\_point, [3](#)

esri\_to\_sf\_polygon, [4](#)

esri\_to\_sf\_polyline, [4](#)

get\_esri\_features, [5](#)

get\_esri\_features\_by\_ids, [6](#)

get\_object\_ids, [7](#)

inventory, [8](#)

layer\_info, [3-5](#), [7](#), [8](#), [9](#), [9](#)