

# Package: tidyqwi (via r-universe)

August 27, 2024

**Title** A Convenient API for Accessing United States Census Bureau's  
Quarterly Workforce Indicator

**Version** 0.1.3

**Maintainer** Michael DeWitt <me.dewitt.jr@gmail.com>

**Description** The purpose of this package is to access the United States  
Census Bureau's Quarterly Workforce Indicator data.  
Additionally, the data will be retrieved in a tidy format for  
further manipulation with full variable descriptions added if  
desired. Information about the United States Census Bureau's  
Quarterly Workforce Indicator is available at  
<<https://www.census.gov/data/developers/data-sets/qwi.html>>.

**Depends** R (>= 3.5), future (>= 1.6.2)

**Imports** dplyr, httr, jsonlite, magrittr, xml2, stringr, purrr, stats,  
tidyr, labelled, furrr

**License** MIT + file LICENSE

**BugReports** <https://github.com/medewitt/tidyqwi/issues>

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.1

**Suggests** testthat, covr, knitr, rmarkdown, spelling

**VignetteBuilder** knitr

**Language** en-US

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

**RemoteUrl** <https://github.com/medewitt/tidyqwi>

**RemoteRef** HEAD

**RemoteSha** 186ee7055aab82e553d038e5c092a1b9740299a9

## Contents

add_qwi_labels . . . . .	2
check_census_api_call . . . . .	3
converted_fips . . . . .	3
geo_codes . . . . .	4
get_qwi . . . . .	5
industry_labels . . . . .	6
nc_qwi . . . . .	7
owner_codes . . . . .	7
parse_qwi_message . . . . .	8
qwi_var_names . . . . .	8
show_condition . . . . .	9
state_info . . . . .	9
<b>Index</b>	<b>10</b>

---

add_qwi_labels	<i>add_qwi_labels</i>
----------------	-----------------------

---

### Description

This function add labels to a ‘qwi’ object

### Usage

```
add_qwi_labels(df)
```

### Arguments

df                    an object with a class of ‘qwi’

### Value

a data.frame with formatted column names and types

### Examples

```
library(tidyqwi)

# Add labels
labelled_nc <- add_qwi_labels(nc_qwi)

# Check the label for the data
attr(labelled_nc[["Emp"]], "label")
```

---

check\_census\_api\_call *A helper function to help parse API calls from the census*

---

**Description**

The function verifies that API call was successful. If the call was not successful, this function passes the message received from the US Census API for further troubleshooting,

**Usage**

```
check_census_api_call(call)
```

**Arguments**

call                    a returned call from the US Census API

**Value**

a string vector with the message from the US Census API

**Examples**

```
if(FALSE){
  library(tidyqwi)
  library(httr)
  # A single call to the API without an API Key
  url <- "api.census.gov/data/timeseries/qwi/sa?get=Emp&for=county:198&key=NOKEY"
  single_call <- httr::GET(url)
  stop_for_status(single_call)

  # Because a non valid API key was specified an message will be returned

  check_census_api_call(single_call)
}
```

---

converted\_fips            *A function to check if a valid state number or fips is passed*

---

**Description**

The function verifies if a valid FIPS code was passed and converts it to a unified standard for internal use.

**Usage**

```
converted_fips(fips)
```

**Arguments**

`fips` the state abbreviation or fips code vector

**Value**

States Abbreviations or FIPs as FIP character strings  
a vector with the State FIPS code

**Examples**

```
library(tidyqwi)

converted_fips(37)
converted_fips("37")
converted_fips("NC")
converted_fips("nc")
```

---

geo\_codes

*Geographic Codes*

---

**Description**

Geographic Codes

**Usage**

```
geo_codes
```

**Format**

a dataframe with 5423 rows and 3 columns:

**geography** geography

**label** label

**geolevel** geolevel ...

**Source**

[https://lehd.ces.census.gov/data/schema/latest/label\\_geography.csv](https://lehd.ces.census.gov/data/schema/latest/label_geography.csv)

---

get\_qwi

*Retrieve the Quarterly Workforce Indicator Data*


---

### Description

The purpose of this function is to retrieve firm information from the US Census' Quarterly Workforce Indicator API. These data can be retrieved with by specifying the states, the quarters, the years, and additional detail. This function can accept multiple states, years and quarters. This makes the data retrieval easier and stay inside of the US Census' limits on the API.

### Usage

```
get_qwi(
  years,
  variables = NULL,
  quarters = c(1, 2, 3, 4),
  industry_level = 2,
  states,
  endpoint = "sa",
  all_groups = TRUE,
  owner_code = TRUE,
  geography = "cbsa",
  seasonadj = "U",
  apikey = NULL,
  processing = "sequential"
)
```

### Arguments

years	years to fetch (e.g. 2010, or c(2010, 2011))
variables	the variables you wish to fetch. Default is all.
quarters	The quarters to fetch (e.g. c(1,2,3,4)) Default is all
industry_level	Industries to fetch. Default is all level 2
states	state fips code to fetch
endpoint	US Census endpoint designation. One of "sa" for Sex * Age, "se" for Sex by Education and "rh" for Race/Ethnicity
all_groups	default to true
owner_code	firm owner code
geography	the US Census geography granularity (one of cbsa or county)
seasonadj	seasonal adjustment factor (one of "U" or "S")
apikey	your US Census API Key
processing	the processing strategy (default = "sequential")

**Value**

the desired data from the US Census's Quaterly Workforce API as a tibble

**Examples**

```
## Not run:
library(tidyqwi)

# One state, one year
nc_qwi <- get_qwi(years = "2010",
                 states = "11",
                 geography = "county",
                 apikey = census_key,
                 endpoint = "rh",
                 variables = c("sEmp", "Emp"), all_groups = FALSE,
                 industry_level = "2", processing = "sequential")

# Multiple states. multiple years
qwi_multi_year <- get_qwi(years = c("2010", "2011", "2012"),
                        states = c("NC", "SC"),
                        geography = "county",
                        apikey = census_key,
                        endpoint = "rh",
                        variables = c("sEmp", "Emp"), all_groups = FALSE,
                        industry_level = "2", processing = "sequential")

## End(Not run)
```

---

industry_labels	<i>Industry Labels</i>
-----------------	------------------------

---

**Description**

These data are the industry labels specified by the United States Census Bureau

**Usage**

```
industry_labels
```

**Format**

a dataframe with 433 rows and 3 columns:

**industry** Industry Numeric Code

**label** Description of Industry Level

**ind\_level** Industry Level ...

**Source**

[https://lehd.ces.census.gov/data/schema/latest/label\\_industry.csv](https://lehd.ces.census.gov/data/schema/latest/label_industry.csv)

---

 nc\_qwi

*Example Data Set*


---

**Description**

These data represent an example returned query for NC for 2010

**Usage**

nc\_qwi

**Format**

a dataframe with 3244 rows and 44 columns:

---

 owner\_codes

*Owner Codes*


---

**Description**

Owner Codes

**Usage**

owner\_codes

**Format**

a dataframe with 3 rows and 2 columns:

**ownercode** ownercode

**label** label ...

**Source**

[https://lehd.ces.census.gov/data/schema/latest/label\\_ownercode.csv](https://lehd.ces.census.gov/data/schema/latest/label_ownercode.csv)

---

parse_qwi_message	<i>parse_qwi</i>
-------------------	------------------

---

**Description**

An internally used function to parse the returned API call.

**Usage**

```
parse_qwi_message(x)
```

**Arguments**

x a returned call response from the US Census QWI API

---

qwi_var_names	<i>QWI Variable Names</i>
---------------	---------------------------

---

**Description**

These data represent the different variable types available from the QWI API.

**Usage**

```
qwi_var_names
```

**Format**

a dataframe with 83 rows and 9 columns:

**name** state name  
**label** state fips code  
**concept** state abbreviation  
**required** requirements  
**attributes** details of attributes  
**limit** limit  
**predicate type** predicate type  
**group** group level  
**values** values ...

**Source**

<https://api.census.gov/data/timeseries/qwi/se/variables.html>



---

show_condition	<i>show_condition</i>
----------------	-----------------------

---

**Description**

show\_condition

**Usage**

show\_condition(code)

**Arguments**

code            the code whose message you wish to interpret

---

state_info	<i>State Data (FIPS, Abbreviations, etc)</i>
------------	--

---

**Description**

State Data (FIPS, Abbreviations, etc)

**Usage**

state\_info

**Format**

a dataframe with 51 rows and 3 columns:

**name** state name

**state\_fips** state fips code

**state\_abbreviation** state abbreviation ...

**Source**

<https://www.census.gov/library/reference/code-lists/ansi.html>

# Index

## \* datasets

- geo\_codes, 4
- industry\_labels, 6
- nc\_qwi, 7
- owner\_codes, 7
- qwi\_var\_names, 8
- state\_info, 9

add\_qwi\_labels, 2

check\_census\_api\_call, 3

converted\_fips, 3

geo\_codes, 4

get\_qwi, 5

industry\_labels, 6

nc\_qwi, 7

owner\_codes, 7

parse\_qwi\_message, 8

qwi\_var\_names, 8

show\_condition, 9

state\_info, 9