

# Package: rbrsa (via r-universe)

May 29, 2026

**Title** Fetch Turkish Banking Sector Data from the Turkish Banking Regulation and Supervision Agency

**Version** 0.2.0

**Description** The goal of the 'rbrsa' package is to provide automated access to banking sector data from the Turkish Banking Regulation and Supervision Agency (BRSA, known as BDDK in Turkish). The package retrieves tables from two distinct publication portals maintained by the BRSA: The Monthly Bulletin Portal <<https://www.bddk.org.tr/bultenaylik>> and The FinTurk Data System <<https://www.bddk.org.tr/BultenFinturk>>.

**License** MIT + file LICENSE

**URL** <https://github.com/obakis/rbrsa>, <https://obakis.github.io/rbrsa/>

**BugReports** <https://github.com/obakis/rbrsa/issues>

**Depends** R (>= 4.1.0)

**Imports** httr2 (>= 1.0.0), jsonlite (>= 1.8.0), writexl, rlang (>= 1.1.6)

**Suggests** dplyr, ggplot2, scales, knitr, rmarkdown, testthat (>= 3.0.0)

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Ozan Bakis [aut, cre]

**Maintainer** Ozan Bakis <[ozanbakis@gmail.com](mailto:ozanbakis@gmail.com)>

**Config/pak/sysreqs** libssl-dev zlib1g-dev

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

**Date/Publication** 2026-05-29 06:02:00 UTC

**RemoteUrl** <https://github.com/cranhaven/cranhaven.r-universe.dev>

**RemoteRef** package/rbrsa

**RemoteSha** e81a547be9a8c427006e3a73ed23ecc78b08ba74

**RemoteSubdir** rbrsa

## Contents

fetch_bddk	2
fetch_bddk1	3
fetch_finturk	4
fetch_finturk1	5
list_cities	6
list_groups	7
list_tables	7
plaka_to_city	8
save_data	8

<b>Index</b>	<b>10</b>
--------------	-----------

---

fetch_bddk	<i>Fetch multiple period table from BDDK website</i>
------------	--

---

### Description

Fetches BDDK data for a range of months by calling `fetch_bddk1` iteratively.

### Usage

```
fetch_bddk(
  start_year,
  start_month,
  end_year,
  end_month,
  table_no,
  grup_kod = 10001,
  currency = "TL",
  lang = "en",
  delay = 0.5,
  verbose = TRUE
)
```

### Arguments

<code>start_year, end_year</code>	Starting/ending year (YYYY).
<code>start_month, end_month</code>	Starting/ending month (1-12).
<code>table_no</code>	Table number to fetch (1-17). No default. Use <a href="#">list_tables</a> with source = "bddk" to see available options.
<code>grup_kod</code>	Group code (10001-10016). Default 10001. Use <a href="#">list_groups</a> with source = "bddk" to see available options.
<code>currency</code>	Currency code ("TL" or "USD"). Default "TL".

lang	Language ("en" or "tr"). Default "en".
delay	Delay between requests in seconds. Default 0.5.
verbose	Print progress messages. Default TRUE.

**Value**

Combined data frame with "fetch\_info" attribute.

**See Also**

[fetch\\_finturk\(\)](#) for quarterly province-level data.

**Examples**

```
# Fetch multiple months
my_dat <- fetch_bddk(2024, 1, 2024, 3, table_no = 15)
```

---

 fetch\_bddk1

---

*Fetch Monthly Data from BDDK with Multiple Group Codes*


---

**Description**

Retrieves monthly banking data from the BDDK API for specified group codes. Supports multiple group codes in a single request, returning a combined data frame with consistent numeric grup\_kod values.

**Usage**

```
fetch_bddk1(
  year,
  month,
  table_no,
  grup_kod = 10001,
  currency = "TL",
  lang = "en"
)
```

**Arguments**

year	Year as 4-digit integer (YYYY).
month	Month as integer (1-12).
table_no	Table number to fetch (1-17). No default. Use <a href="#">list_tables</a> with source = "bddk" to see available options.
grup_kod	Group code (10001-10010). Default 10001. Use <a href="#">list_groups</a> with source = "bddk" to see available options.
currency	Currency code ("TL" or "USD"). Default "TL".
lang	Language ("en" or "tr"). Default "en".

**Value**

Data frame with a `fetch_info` attribute that contains query details.

**See Also**

[fetch\\_finturk1\(\)](#) for quarterly province-level data.

**Examples**

```
# Single group code
fetch_bddk1(2020, 3, 1, grup_kod = 10001)

# Multiple group codes
fetch_bddk1(2020, 3, 1, grup_kod = c(10001, 10002))

# Turkish language output
fetch_bddk1(2020, 3, 1, grup_kod = 10001, lang = "tr")
```

---

`fetch_finturk`*Fetch multiple period FinTurk data*

---

**Description**

Fetches FinTurk data for a range of quarters by calling `fetch_finturk1` iteratively.

**Usage**

```
fetch_finturk(
  start_year,
  start_month,
  end_year,
  end_month,
  table_no,
  grup_kod = 10001,
  il = 0,
  delay = 0.5,
  verbose = TRUE
)
```

**Arguments**

`start_year`, `end_year` Starting/ending year (YYYY).

`start_month`, `end_month` Starting/ending month (3,6,9,12).

`table_no` Table number to fetch (1-7). No default. Use [list\\_tables](#) with `source = "finturk"` to see available options.

grup_kod	Group code (10001-10007). Default 10001. Use <a href="#">list_groups</a> with source = "finturk" to see available options.
il	plaka (license plate) number (0-81, 99). Default 0.
delay	Delay between requests in seconds. Default 0.5.
verbose	Print progress messages. Default TRUE.

**Value**

Combined data frame with "fetch\_info" attribute.

**See Also**

[fetch\\_bddk\(\)](#) for monthly BRSA data .

**Examples**

```
# Fetch multiple quarters
my_data <- fetch_finturk(2024, 3, 2024, 9, table_no = 1)
```

---

fetch\_finturk1

*Fetch Quarterly Data from BDDK FinTurk with Multiple Provinces*

---

**Description**

Retrieves quarterly banking data from the BDDK FinTurk API for specified group codes and provinces. Supports multiple group codes and province codes in a single request.

**Usage**

```
fetch_finturk1(year, month, table_no, grup_kod = 10001, il = 0)
```

**Arguments**

year	Year as 4-digit integer (YYYY).
month	Month as integer (3,6,9,12 for quarterly data).
table_no	Table number to fetch (1-7). No default.
grup_kod	Group code (10001-1007). Default 10001.
il	plaka (license plate) number (0-81); 999 = Yurt Disi. Default 0. 0=HEPSI (All Cities), 1=Adana, 6=Ankara, 34=Istanbul, 35=Izmir, etc. See <a href="#">list_cities</a> for full list.

**Details**

The FinTurk API only provides data for quarter-ending months (March, June, September, December). Province codes follow Turkey's standard license plate numbering (1 = Adana, 6 = Ankara, 34 = Istanbul, etc.).

**Value**

Data frame with a `fetch_info` attribute that contains query details.

**See Also**

[fetch\\_bddk1\(\)](#) for monthly data without province granularity.

**Examples**

```
# Single group, all provinces
fetch_finturk1(2020, 3, 1, grup_kod = 10001)

# Multiple groups and specific provinces
fetch_finturk1(2020, 3, 1, grup_kod = c(10006, 10007), il = c(6, 34))

# Single group, single province
fetch_finturk1(2020, 3, 1, grup_kod = 10001, il = 34)
```

---

list\_cities

*List Available Cities for Finturk Print available cities for Finturk quarterly data with plaka (license plate) numbers.*

---

**Description**

List Available Cities for Finturk Print available cities for Finturk quarterly data with plaka (license plate) numbers.

**Usage**

```
list_cities()
```

**Value**

Data frame of available cities

**Examples**

```
list_cities()
```

---

list_groups	<i>List available groups Print available banking groups for a data source.</i>
-------------	--

---

**Description**

List available groups Print available banking groups for a data source.

**Usage**

```
list_groups(source = c("bddk", "finturk"), lang = c("en", "tr"))
```

**Arguments**

source	Either "bddk" or "finturk"
lang	Either "tr" or "en" for names. "en" is default

**Value**

Data frame of available groups (invisibly)

**Examples**

```
list_groups("bddk")  
list_groups("finturk", "tr")
```

---

list_tables	<i>List Available Tables Print available tables for a data source.</i>
-------------	--

---

**Description**

List Available Tables Print available tables for a data source.

**Usage**

```
list_tables(source = c("bddk", "finturk"), lang = c("en", "tr"))
```

**Arguments**

source	Either "bddk" or "finturk"
lang	Either "tr" or "en" for column names. "en" is default

**Value**

Data frame of available tables (invisibly)

**Examples**

```
list_tables("bddk")
list_tables("finturk", "tr")
```

---

plaka_to_city	<i>Convert plaka (license plate number) to province name Maps Turkish license plate numbers to province names used in the Finturk API.</i>
---------------	--

---

**Description**

Convert plaka (license plate number) to province name Maps Turkish license plate numbers to province names used in the Finturk API.

**Usage**

```
plaka_to_city(plaka)
```

**Arguments**

plaka            license plate number (0 for "HEPSI", 1-81 for provinces, 999 for "YURT DISI")

**Value**

province name in ALL CAPS as required by API

**Examples**

```
plaka_to_city(6)    # "ANKARA"
plaka_to_city(34) # "ISTANBUL"
plaka_to_city(0)  # "HEPSI"
```

---

save_data	<i>Save Fetched Data to Multiple Formats</i>
-----------	--

---

**Description**

Save Fetched Data to Multiple Formats

**Usage**

```
save_data(df, filename = NULL, format = "rds")
```

**Arguments**

df            Data frame to save (with fetch\_info attribute for auto-naming).  
filename      **Required.** A non-empty string (without extension) must be provided.  
format        Output format: "rds", "csv", or "xlsx". Default "rds".

*save\_data*

9

**Value**

Full file path (invisibly).

**Examples**

```
my_data <- fetch_bddk1(2024, 1, 15)
temp_file <- tempfile() # filename should be without extension
save_data(my_data, temp_file, format = "csv")
```

# Index

fetch\_bddk, 2  
fetch\_bddk(), 5  
fetch\_bddk1, 3  
fetch\_bddk1(), 6  
fetch\_finturk, 4  
fetch\_finturk(), 3  
fetch\_finturk1, 5  
fetch\_finturk1(), 4

list\_cities, 5, 6  
list\_groups, 2, 3, 5, 7  
list\_tables, 2-4, 7

plaka\_to\_city, 8

save\_data, 8