

# Package ‘tidyplate’

October 9, 2024

**Type** Package

**Title** Transform Microplate Data into Tibbles

**Version** 2.0.1

**Maintainer** Shubham Dutta <shubhamdutta26@gmail.com>

**Description** The goal of 'tidyplate' is to help researchers convert different types of microplates into tibbles which can be used in data analysis. It accepts xlsx and csv files formatted in a specific way as input. It supports all types of standard microplate formats such as 6-well, 12-well, 24-well, 48-well, 96-well, 384-well, and, 1536-well plates.

**Imports** data.table, readxl, tibble, rlang, openxlsx

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**URL** <https://github.com/shubhamdutta26/tidyplate>,  
<https://www.shubhamdutta.com/tidyplate/>

**BugReports** <https://github.com/shubhamdutta26/tidyplate/issues>

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**Language** en-GB

**NeedsCompilation** no

**Author** Shubham Dutta [aut, cre, cph] (<<https://orcid.org/0000-0001-8484-0717>>)

**Repository** CRAN

**Date/Publication** 2024-10-08 22:00:02 UTC

## Contents

|                            |   |
|----------------------------|---|
| build_plate . . . . .      | 2 |
| check_plate . . . . .      | 3 |
| tidy_plate . . . . .       | 3 |
| view_plate_names . . . . . | 4 |

---

|             |   |
|-------------|---|
| build_plate | <i>Generates a csv or xlsx template for each plate type</i> |
|-------------|---|

---

### Description

Generates a csv or xlsx template for each plate type

### Usage

```
build_plate(  
  plate_type = 6,  
  n_plates = 1,  
  file_type = c("csv", "xlsx"),  
  plate_names = NULL,  
  file = NULL  
)
```

### Arguments

|             |   |
|-------------|---|
| plate_type  | A specific integer (6, 12, 24, 48, 96, 384, 1536) indicating the type of microwell plate.                                     |
| n_plates    | A positive integer indicating the number of plates.   |
| file_type   | A character string ("csv" (the default) or "xlsx") indicating the filetype.   |
| plate_names | A character vector of unique values that will be assigned to each plate. Its length should be equal to the value of n_plates. |
| file        | A character string naming the file.   |

### Value

A csv or xlsx template file.

### Examples

```
## Not run:  
build_plate(plate_type = 6, n_plates = 2)  
build_plate(plate_type = 6, n_plates = 2, file_type = "xlsx")  
  
## End(Not run)
```

---

|             |  |
|-------------|--|
| check_plate | <i>Checks whether the input file is valid for use with the tidy_plate() function</i> |
|-------------|--|

---

**Description**

Checks whether the input file is valid for use with the tidy\_plate() function

**Usage**

```
check_plate(file, well_id = "well", sheet = 1)
```

**Arguments**

|         |  |
|---------|--|
| file    | A character string containing the path to a csv or excel file.               |
| well_id | A character string that will be the name for the well id column.             |
| sheet   | If file type is xlsx this is the sheet name (character) or number (integer). |

**Value**

An message indicating whether the input file is compatible with the tidy\_plate() function

**Examples**

```
file_path <- system.file(  
  "extdata",  
  "example_12_well.xlsx",  
  package = "tidyplate"  
)  
  
check_plate(file = file_path)
```

---

|            |   |
|------------|---|
| tidy_plate | <i>Reads and transforms microwell plate to a tibble</i> |
|------------|---|

---

**Description**

Reads and transforms microwell plate to a tibble

**Usage**

```
tidy_plate(file, well_id = "well", sheet = 1)
```

**Arguments**

|         |   |
|---------|---|
| file    | A character string containing the path to a csv or excel file. The format is described below. |
| well_id | A character string that will be the name for the well id column.                              |
| sheet   | A character or integer indicating the excel sheet to be read.                                 |

**Value**

A tibble.

**Examples**

```
file_path <- system.file("extdata", "example_12_well.xlsx",
  package = "tidyplate"
)

data_12 <- tidy_plate(file = file_path)

head(data_12)
```

---

view\_plate\_names      *Returns the name of each plate in the file*

---

**Description**

Returns the name of each plate in the file

**Usage**

```
view_plate_names(file, sheet = 1)
```

**Arguments**

|       |   |
|-------|---|
| file  | This is the path to a xlsx or csv file containing data for the following types of plates: 6, 12, 24, 48, 96, 384, and 1536. |
| sheet | If file type is xlsx this is the sheet name (character) or number (integer).  |

**Value**

A character vector

**Examples**

```
file_path <- system.file("extdata", "example_12_well.xlsx", package = "tidyplate")

data_12 <- view_plate_names(file = file_path)

data_12
```

# Index

`build_plate`, 2

`check_plate`, 3

`tidy_plate`, 3

`view_plate_names`, 4