Package ‘featureflag’

October 13, 2022

Title Turn Features On and Off using Feature Flags
Version 0.1.0
Description Feature flags allow developers to turn features of their
software on and off in form of configuration. This package provides
functions for creating feature flags in code. It exposes an interface
for defining own feature flags which are enabled based on custom criteria.
License MIT + file LICENSE
Encoding UTF-8
LazyData true
RoxygenNote 7.1.0
Suggests testthat, spelling, knitr, rmarkdown
Language en-US
URL https://github.com/szymanskir/featureflag
BugReports https://github.com/szymanskir/featureflag/issues
VignetteBuilder knitr
NeedsCompilation no
Author Ryszard Szymański [aut, cre]
Maintainer Ryszard Szymański <ryszard.szymanski@outlook.com>
Repository CRAN
Date/Publication 2021-02-18 14:50:02 UTC

R topics documented:
create_bool_feature_flag ...................................................... 2
create_feature_flag .......................................................... 2
feature_if ................................................................. 3
feature_ifelse ............................................................. 3
is_enabled ................................................................. 4
is_enabled.bool_feature_flag ........................................... 5

Index 6
create_bool_feature_flag

*Create an instance of a bool feature flag with the specified bool value.*

**Description**

Creates an instance of a bool feature flag with the specified bool value.

**Usage**

`create_bool_feature_flag(value)`

**Arguments**

`value` single logical determining whether the flag should be enabled

**Value**

feature flag object of the bool value

**Examples**

```r
{
  enabled_flag <- create_bool_feature_flag(TRUE)
  disabled_flag <- create_bool_feature_flag(FALSE)
}
```

create_feature_flag

*Creates the base of a feature flag.*

**Description**

It should not be used directly, but only as a prerequisite when creating concrete feature flag.

**Usage**

`create_feature_flag()`

**Value**

instance of a base feature flag.
**feature_if**

_Evaluates the provided expression if the feature flag is enabled._

**Description**

Evaluates the provided expression if the feature flag is enabled.

**Usage**

`feature_if(feature_flag, expr)`

**Arguments**

- `feature_flag`: flag which defines whether the provided expression should be evaluated
- `expr`: expression to evaluate when the feature_flag is enabled

**Details**

The passed expression is evaluated in the frame where `feature_if` is called.

**Value**

If the passed `feature_flag` is enabled, then the result of the evaluation of the passed expression is returned. Otherwise there is no return value.

**Examples**

```plaintext
{
    flag <- create_bool_feature_flag(TRUE)

    feature_if(flag, {
        2 + 7
    })
}
```

**feature_ifelse**

_Evaluates one or the other expression based on whether the feature flag is enabled._

**Description**

Evaluates one or the other expression based on whether the feature flag is enabled.

**Usage**

`feature_ifelse(feature_flag, true_expr, false_expr)`
Arguments

- feature_flag: flag which defines which expression should be evaluated
- true_expr: expression to evaluate when the feature_flag is enabled
- false_expr: expression to evaluate when the feature_flag is disabled

Details

The passed expression is evaluated in the frame where feature_ifelse is called.

Value

The result of evaluating true_expr is returned if the passed feature_flag is enabled. Otherwise the result of evaluating false_expr is returned.

Examples

```r
{  
  flag <- create_bool_feature_flag(TRUE)  
  
  feature_ifelse(  
    flag,  
    2 * 7,  
    3 * 7  
  )  
}
```

---

**is_enabled**

Checks if the given feature flag is enabled.

Description

Checks if the given feature flag is enabled.

Usage

```r
is_enabled(feature_flag)
```

Arguments

- feature_flag: feature flag to be tested whether it is enabled

Value

TRUE if the feature flag is enabled.
Description
Checks if the given bool feature flag is enabled

Usage
```r
## S3 method for class 'bool_feature_flag'
is_enabled(feature_flag)
```

Arguments
- `feature_flag`: flag to be checked whether it is enabled

Value
TRUE if the feature flag is enabled.

Examples
```r
{  
  enabled_flag <- create_bool_feature_flag(TRUE)  
  
  if (is_enabled(enabled_flag)) {  
    print("The flag is enabled!")  
  } 
}
```
Index

create_bool_feature_flag, 2
create_feature_flag, 2

feature_if, 3
feature_ifelse, 3

is_enabled, 4
is_enabled.bool_feature_flag, 5