Package 'googleTagManageR'

October 13, 2022

Type Package

Title Access the 'Google Tag Manager' API using R

Version 0.2.0

Description Interact with the 'Google Tag Manager' API <https://developers.google.com/tag-platform/tag-manager/api/v2>, enabling scripted deployments and updates across multiple tags, triggers, variables and containers.

Depends R (>= 3.5.0)

Imports assertthat, future (>= 1.2.0), googleAuthR (>= 1.2.1), httr (>= 1.3.1), jsonlite (>= 1.1), utils, dplyr (>= 0.8.0), purrr

Suggests covr, knitr, rmarkdown, testthat

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.0

NeedsCompilation no

Author James Cottrill [aut, cre, cph],
Mark Edmondson [ctb],
Manos Parzakonis [ctb]

Maintainer James Cottrill <r@jamescottrill.co.uk>

Repository CRAN

Date/Publication 2022-04-16 18:12:29 UTC

R topics documented:

---

gtm_accounts_get *List account metadata*

Description

This returns a list containing the metadata about a single account. If you want to get the information
for all accounts, use `gtm_accounts_list`

Usage

```
gtm_accounts_get(account_id)
```

Arguments

account_id   Account Id

See Also

<https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/get>

Other account functions: `gtm_accounts_list()`, `gtm_accounts_update()`

Examples

```
## Not run:

account_id <- 12345678
account <- gtm_accounts_get(account_id)


## End(Not run)
```

---

gtm_accounts_list   *List all accounts a user has accesss to*

Description

This will return a data frame all your available accounts, returning the path, account Id and account name.

Usage

```
gtm_accounts_list()
```

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/list

Other account functions: [gtm_accounts_get()](), [gtm_accounts_update()]()

Examples

```
## Not run:

accounts <- gtm_accounts_list()


## End(Not run)
```

---

gtm_accounts_update   *Updates account metadata*

Description

This enables you to update the account metadata. You can update the account name and whether or not data is shared with Google.

Usage

```
gtm_accounts_update(account_id, name = NULL, shareData = c("TRUE", "FALSE"))
```

Arguments

account_id	Account Id
name	Account Display Name
shareData	Whether the account shares data anonymously with Google and others. This flag enables benchmarking by sharing your data in an anonymous form. Google will remove all identifiable information about your website, combine the data with hundreds of other anonymous sites and report aggregate trends in the benchmarking service.

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/update

Other account functions: [gtm_accounts_get()](gtm_accounts_get), [gtm_accounts_list()](gtm_accounts_list)

Examples

```
## Not run:

account_id <- 12345678
new_account <- gtm_accounts_update(account_id, name = "New Container Name")


## End(Not run)
```

---

gtm_auth	*Auth for GTM Specific Properties*

Description

This function authenticates the user with Google Tag Manager To use your own GTM API credentials, either ensure that you have set the path to your OAuth2 API Credentials file in your 'GAR_CLIENT_JSON' variable in your .Renviron file and then restart your R session or set

Usage

```
gtm_auth(email = NULL, token = NULL, sa_json = NULL)
```

Arguments

email	The email address for the Google Account
token	An existing Google Auth Token
sa_json	A GCP Service Account Key in JSON format, either a string or file.

Value

Invisibly, the token that has been saved to the session

Examples

```
## Not run:
# To use the included (shared) credentials
library(googleTagManageR)
gtm_auth()

# To useyour own GCP credentials
Sys.setenv("GAR_CLIENT_JSON" = "location/of/file.json")
library(googleTagManageR)
gtm_auth()

# Reauthentication - if you've already logged in
gtm_auth(email="me@mycompany.co.uk")

# Service Account Login
gtm_auth(sa_json="my-service-account.json")

## End(Not run)
```

gtm_builtin_create   *Enable a built-in variable*

Description

This enables one of the built-in variables in GTM. It is the inverse of gtm_builtin_delete

Usage

```
gtm_builtin_create(account_id, container_id, workspace_id, variable)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id
variable	A Builtin Variable or list of built in variables. Valid values are here variables_list

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/
containers/workspaces/built_in_variables/create

Other built in variable functions: gtm_builtin_delete(), gtm_builtin_list(), gtm_builtin_revert(),
variables_list

Examples

```
## Not run:

accountId <- 1234567
containerId <- 7654321
workspaceId <- 10
variable <- 'clickId'

builtinVaribles<- gtm_builtin_create(accountId, containerId, workspaceId, variable)

listOfVaribales <- c('clickClasses', 'clickElement')

newEnabledVars <- gtm_builtin_create(accountId, containerId, workspaceId, listOfVaribales)


## End(Not run)
```

---

gtm_builtin_delete   *Disable a built-in variable*

Description

This disables one or more of the built-in variables in GTM. It is the inverse of `gtm_builtin_create`

Usage

```
gtm_builtin_delete(account_id, container_id, workspace_id, variable)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id
variable	A built-in Variable or list of built-in variables. Valid values are here `variables_list`

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/
containers/workspaces/built_in_variables/delete

Other built in variable functions: `gtm_builtin_create()`, `gtm_builtin_list()`, `gtm_builtin_revert()`,
`variables_list`

Examples

```
## Not run:

accountId <- 1234567
containerId <- 7654321
workspaceId <- 10
variable <- 'clickId'

builtinVaribles<- gtm_builtin_delete(accountId, containerId, workspaceId, variable)

listOfVaribales <- c('clickClasses', 'clickElement')

newEnabledVars <- gtm_builtin_delete(accountId, containerId, workspaceId, listOfVaribales)


## End(Not run)
```

---

gtm_builtin_list   *List all built-in variables*

Description

This will return a data frame all enabled built-in variables in a given workspace

Usage

```
gtm_builtin_list(account_id, container_id, workspace_id)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/built_in_variables/list

Other built in variable functions: gtm_builtin_create(), gtm_builtin_delete(), gtm_builtin_revert(), variables_list

Examples

```
## Not run:

accountId <- 1234567
containerId <- 7654321
workspaceId <- 10

builtinVaribles<- gtm_builtin_list(accountId, containerId, workspaceId)


## End(Not run)
```

---

gtm_builtin_revert   *Reverts changes to a GTM Built-In Variables in a GTM Workspace*

Description

This reeverts any changes to a GTM Built-In Variables in a GTM Workspace.

Usage

```
gtm_builtin_revert(account_id, container_id, workspace_id, variable)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id
variable	A built-in Variable or list of built-in variables. Valid values are here variables_list

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/
containers/workspaces/built_in_variables/revert

Other built in variable functions: gtm_builtin_create(), gtm_builtin_delete(), gtm_builtin_list(),
variables_list

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 10
variable <- 'clickId'

variable <- gtm_variables_revert(accountId, containerId, workspaceId, variable)
```

```
# Changes to variable 22 have been reverted

## End(Not run)
```

---

gtm_containers_create   *Creates a new container in an acount*

Description

This creates a new container in the specified account.

Usage

```
gtm_containers_create(
  account_id,
  name,
  type = c("web", "iosSdk5", "androidSdk5", "amp", "server"),
  domain_name = NULL,
  notes = NULL
)
```

Arguments

account_id	Account Id
name	Container Name
type	Container Use Type, one of web, iosSdk5, androidSdk5, amp or server
domain_name	List of domain names associated with the Container.
notes	Container Notes.

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/create

Other container functions: [gtm_containers_delete()](), [gtm_containers_get()](), [gtm_containers_list()](), [gtm_containers_update()]()

Examples

```
## Not run:
accountId <- 1234567
newName <- "My New Container"

container <- gtm_containers_create(accountId, newName, 'web')


## End(Not run)
```

gtm_containers_delete *Deletes an existing GTM Container.*

Description

This deletes a GTM container from an account. This is an irreversible process, so it's recommended that you first back up the container.

Usage

```
gtm_containers_delete(account_id, container_id, force = c("TRUE", "FALSE"))
```

Arguments

account_id	Account Id
container_id	Container Id
force	Force deletion without user input

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/delete

Other container functions: [gtm_containers_create](), [gtm_containers_get](), [gtm_containers_list](), [gtm_containers_update]()

Examples

```
## Not run:

accountId <- 1234567
containerId <- 7654321
gtm_conainers_delete(accountId, containerId, "TRUE")


containerId <- 7654567

gtm_containers_delete(accountId, containerId)

# !!WARNING!! This command will delete your container.
# This operation is irrevocable.
# It is strongly recommended that you create an export of your container
# before you delete it, just in case you ever want it again.
# Are you sure you want to continue?

#  1: Yes
#  2: No

# Selection: 1
```

```
# Are You really sure you want to delete this container?

#  1: Yes
#  2: No

# Selection: 1

# Container 7654567 has been deleted.


## End(Not run)
```

---

gtm_containers_get   *Gets the metadata for a single container in GTM*

Description

This returns a list containing all the metadata about a single container. If you want to get the
information for all containers, use gtm_containers_list

Usage

```
gtm_containers_get(account_id, container_id)
```

Arguments

account_id   Account Id

container_id   Container Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/get

Other container functions: [gtm_containers_create](), [gtm_containers_delete](), [gtm_containers_list](),
[gtm_containers_update]()

Examples

```
## Not run:

accountId <- 1234567
containerId <- 7654321
container <- gtm_containers_get(accountId, containerId)


## End(Not run)
```

gtm_containers_list   *List all containers in an account*

Description

This returns a dataframe containing all the containers in an account If you want to get the information for a single container, use gtm_containers_get

Usage

```
gtm_containers_list(account_id)
```

Arguments

account_id   Account Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/list

Other container functions: [gtm_containers_create](), [gtm_containers_delete](), [gtm_containers_get](), [gtm_containers_update]()

Examples

```
## Not run:

accounts <- gtm_accounts_list()
containers <- gtm_containers_list(accounts$accountId[1])


## End(Not run)
```

gtm_containers_update *Updates a GTM Container*

Description

This updates a GTM Container. Although you can't change the type of container, it is still required for this function to work.

Usage

```
gtm_containers_update(
  account_id,
  container_id,
  name,
  type = c("amp", "androidSdk5", "iosSdk5", "web", "server"),
  domain_name = "",
  notes = ""
)
```

Arguments

account_id	Account Id
container_id	Container Id
name	Container Name
type	Container Use Type, one of web, iosSdk5, androidSdk5, amp or server
domain_name	List of domain names associated with the Container.
notes	Container Notes.

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/update

Other container functions: [gtm_containers_create()](), [gtm_containers_delete()](), [gtm_containers_get()](), [gtm_containers_list()]()

Examples

```
## Not run:

accountId <- 1234567
containerId <- 7654321
containers <- gtm_containers_update(accountId, containerId, 'New Container Name', 'amp')


## End(Not run)
```

---

gtm_environments_create

Create an environment

Description

This creates a new GTM environment

Usage

```
gtm_environments_create(
  account_id,
  container_id,
  name,
  description = NULL,
  debug = c("TRUE", "FALSE"),
  url = NULL
)
```

Arguments

account_id	Account Id
container_id	Container Id
name	The environment display name. Can be set or changed only on USER type environments.
description	The environment description. Can be set or changed only on USER type environments.
debug	Whether or not to enable debug by default for the environment.
url	Default preview page url for the environment.

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/environments/update

Other environment functions: gtm_environments_delete(), gtm_environments_get(), gtm_environments_list(), gtm_environments_reauthorize(), gtm_environments_update()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321

env <- gtm_environments_create(accountId, containerId, "Development")

pp_env <- gtm_environments_create(accountId,
                                  containerId,
                                  "Pre-Prod",
                                  debug=TRUE,
                                  url="https://pp.example.com")

## End(Not run)
```

gtm_environments_delete

Delete an environment

Description

This deletes a GTM environment

Usage

```
gtm_environments_delete(
  account_id,
  container_id,
  environment_id,
  force = c("TRUE", "FALSE")
)
```

Arguments

account_id	Account Id
container_id	Container Id
environment_id	Environment Id
force	This will bypass the user confirmation. Useful for scripted functions to avoid user input

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/environments/reauthorize

Other environment functions: gtm_environments_create(), gtm_environments_get(), gtm_environments_list(), gtm_environments_reauthorize(), gtm_environments_update()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
environmentId <- 3

gtm_environments_delete(accountId, containerId, environmentId, TRUE)

environmentId <- 4

gtm_environments_delete(accountId, containerId, environmentId)

#  This will delete environment 4 Are you sure you want to continue?"
```

```
#  1: Yes
#  2: No

# Selection: 1


# Environment 4 has been deleted.


## End(Not run)
```

gtm_environments_get   *This gets the metadata for a single GTM Environment*

Description

This gets the metadata for a single GTM Environment

Usage

```
gtm_environments_get(account_id, container_id, environment_id)
```

Arguments

account_id	Account Id
container_id	Container Id
environment_id	Environment Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/
[containers/environments/get](https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/environments/get)

Other environment functions: `gtm_environments_create()`, `gtm_environments_delete()`, `gtm_environments_list()`,
`gtm_environments_reauthorize()`, `gtm_environments_update()`

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
environmentId <- 3

environmnet <- gtm_environments_get(accountId, containerId, environmentId)


## End(Not run)
```

gtm_environments_list   *List the containers*

Description

This lists the environments in a specified container.

Usage

```
gtm_environments_list(account_id, container_id)
```

Arguments

account_id	Account Id
container_id	Container Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/environments/list

Other environment functions: [gtm_environments_create](), [gtm_environments_delete](), [gtm_environments_get](), [gtm_environments_reauthorize](), [gtm_environments_update]()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321

environments <- gtm_environments_list(accountId, containerId)


## End(Not run)
```

---

gtm_environments_reauthorize
                    Reauthorise an existing GTM environment

Description

This reauthorises an existing GTM environment, generating a new authrorisation Id. This is used to
invalidate any existing preview GTM shared preview links and any installations of the environment.
You will need to update any snippets on your site with the new authorisation Id afterwards.

Usage

```
gtm_environments_reauthorize(
  account_id,
  container_id,
  environment_id,
  environment
)
```

Arguments

account_id	Account Id
container_id	Container Id
environment_id	Environment Id
environment	An Environment object - If you don't have one, one will be generated for you.

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/environments/reauthorize

Other environment functions: [gtm_environments_create()](), [gtm_environments_delete()](), [gtm_environments_get()](), [gtm_environments_list()](), [gtm_environments_update()]()

Examples

```
## Not run:

accountId <- 1234567
containerId <- 7654321
environmentId <- 4

env <- gtm_environments_reauthorize(accountId, containerId, enviromentId)


## End(Not run)
```

---

gtm_environments_update

Update an environment

Description

This updates a GTM environment

Usage

```
gtm_environments_update(
  account_id,
  container_id,
  environment_id,
  name,
  description = NULL,
  debug = c("TRUE", "FALSE"),
  url = NULL
)
```

Arguments

account_id	Account Id
container_id	Container Id
environment_id	Environment Id
name	The environment display name. Can be set or changed only on USER type environments.
description	The environment description. Can be set or changed only on USER type environments.
debug	Whether or not to enable debug by default for the environment.
url	Default preview page url for the environment.

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/
[containers/environments/update](https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/environments/update)

Other environment functions: [gtm_environments_create](), [gtm_environments_delete](), [gtm_environments_get](),
[gtm_environments_list](), [gtm_environments_reauthorize]()

Examples

```
## Not run:

accountId <- 1234567
containerId <- 7654321

gtm_environments_update(accountId, containerId, "Pre-Prod", url="https://new-pp.example.com")

## End(Not run)
```

`gtm_folders_create` *Creates a GTM Folder*

Description

This creates a new folder in the specified workspaces

Usage

```
gtm_folders_create(account_id, container_id, workspace_id, name, notes = NULL)
```

Arguments

`account_id`	Account Id
`container_id`	Container Id
`workspace_id`	Workspace Id
`name`	Folder display name
`notes`	User notes on how to apply this folder in the container.

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/folders/get

Other folder functions: `gtm_folders_delete()`, `gtm_folders_entities()`, `gtm_folders_get()`, `gtm_folders_list()`, `gtm_folders_move()`, `gtm_folders_revert()`, `gtm_folders_update()`

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 3
folderName <- "Analytics Tags"
folderNotes <- "All our website's Google Analytics Tags"

folder <- gtm_folders_create(accountId, containerId, workspaceId, folderName, folderNotes)


## End(Not run)
```

`gtm_folders_delete`   *Deletes a GTM Folder*

Description

This deletes a folder in the specified workspaces

Usage

```
gtm_folders_delete(
  account_id,
  container_id,
  workspace_id,
  folder_id,
  force = c("TRUE", "FALSE")
)
```

Arguments

`account_id`	Account Id
`container_id`	Container Id
`workspace_id`	Workspace Id
`folder_id`	Folder Id
`force`	Force deletion without user input

See Also

[https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/](https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/folders/delete)
[containers/workspaces/folders/delete](https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/folders/delete)

Other folder functions: `gtm_folders_create()`, `gtm_folders_entities()`, `gtm_folders_get()`,
`gtm_folders_list()`, `gtm_folders_move()`, `gtm_folders_revert()`, `gtm_folders_update()`

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 3

gtm_folders_delete(accountId, containerId, workspaceId, force=TRUE)


## End(Not run)
```

gtm_folders_entities   *List all entities in a GTM Folder.*

Description

This lists all entities in a GTM Folder.

Usage

```
gtm_folders_entities(account_id, container_id, workspace_id, folder_id)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id
folder_id	Folder Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/folders/entities

Other folder functions: gtm_folders_create(), gtm_folders_delete(), gtm_folders_get(), gtm_folders_list(), gtm_folders_move(), gtm_folders_revert(), gtm_folders_update()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 3
folderId <- 1

folder <- gtm_folders_entities(accountId, containerId, workspaceId, folderId)


## End(Not run)
```

gtm_folders_get   *Gets a GTM Folder*

Description

This returns a list containing all the metadata for a single folder in a workspace If you want to get the information for all folders, use `gtm_folders_list`

Usage

```
gtm_folders_get(account_id, container_id, workspace_id, folder_id)
```

Arguments

`account_id`	Account Id
`container_id`	Container Id
`workspace_id`	Workspace Id
`folder_id`	Folder Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/folders/get

Other folder functions: `gtm_folders_create()`, `gtm_folders_delete()`, `gtm_folders_entities()`, `gtm_folders_list()`, `gtm_folders_move()`, `gtm_folders_revert()`, `gtm_folders_update()`

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 3
folderId <- 1

folder <- gtm_folders_list(accountId, containerId, workspaceId, folderId)


## End(Not run)
```

---

gtm_folders_list *List all containers in an account*

Description

This returns a dataframe containing all the folder in a workspace If you want to get the information for a single folder, use `gtm_folders_get`

Usage

```
gtm_folders_list(account_id, container_id, workspace_id)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/folders/list

Other folder functions: `gtm_folders_create()`, `gtm_folders_delete()`, `gtm_folders_entities()`, `gtm_folders_get()`, `gtm_folders_move()`, `gtm_folders_revert()`, `gtm_folders_update()`

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 3

folders <- gtm_folders_list(accountId, containerId, workspaceId)


## End(Not run)
```

gtm_folders_move   *Move entities to a new folder*

Description

This moves entities to a new folder.

Usage

```
gtm_folders_move(
  account_id,
  container_id,
  workspace_id,
  folder_id,
  tags = NULL,
  triggers = NULL,
  variables = NULL
)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id
folder_id	Folder Id
tags	The tags to be moved to the folder. This can either be an individual tag Id or a list of tag Ids
triggers	The triggers to be moved to the folder. This can either be an individual trigger Id or a list of trigger Ids
variables	The variables to be moved to the folder. This can either be an individual variable Id or a list of variable Ids

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/folders/move_entities_to_folder

Other folder functions: [gtm_folders_create](), [gtm_folders_delete](), [gtm_folders_entities](), [gtm_folders_get](), [gtm_folders_list](), [gtm_folders_revert](), [gtm_folders_update]()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 3
```

```
folderId <- 1
tags = 12


gtm_folders_move(accountId, containerId, workspaceId, folderId, tags = tags)

triggers = c(1,2,3)
tags = c(4,5,6)

gtm_folders_move(accountId, containerId, workspaceId, folderId, tags = tags)


## End(Not run)
```

---

gtm_folders_revert   *Reverts a GTM Folder*

Description

This reverts any changes to a folder in the current workspace

Usage

```
gtm_folders_revert(account_id, container_id, workspace_id, folder_id)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id
folder_id	Folder Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/folders/revert

Other folder functions: gtm_folders_create(), gtm_folders_delete(), gtm_folders_entities(), gtm_folders_get(), gtm_folders_list(), gtm_folders_move(), gtm_folders_update()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 3
folderId <- 1
```

```
folder <- gtm_folders_revert(accountId, containerId, workspaceId, folderId)


## End(Not run)
```

---

gtm_folders_update   *Updates a GTM Folder*

Description

This updates a folder in the specified workspaces

Usage

```
gtm_folders_update(
  account_id,
  container_id,
  workspace_id,
  folder_id,
  name = NULL,
  notes = NULL
)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id
folder_id	Folder Id
name	Folder display name
notes	User notes on how to apply this folder in the container.

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/folders/update

Other folder functions: [gtm_folders_create](), [gtm_folders_delete](), [gtm_folders_entities](), [gtm_folders_get](), [gtm_folders_list](), [gtm_folders_move](), [gtm_folders_revert]()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 3
folderName <- "New Analytics Tags"
folderNotes <- "All our website's Google Analytics App+Web Tags"

folder <- gtm_folders_update(accountId, containerId, workspaceId, folderName, folderNotes)


## End(Not run)
```

---

gtm_headers_latest   *Gets the latest container version header*

Description

This will return the latest container version header If you want to get the information for the most all container versions, use gtm_headers_list

Usage

```
gtm_headers_latest(account_id, container_id)
```

Arguments

account_id	Account Id
container_id	Container Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/version_headers/latest

Other version header functions: gtm_headers_list()

Examples

```
## Not run:

accountId <- 1234567
containerId <- 7654321


latestHeader <- gtm_headers_latest(accountId, containerId)
```

```
## End(Not run)
```

---

gtm_headers_list   *List all container versions of a GTM container*

Description

This will return a data frame all container versions. If you want to get the information for the most recent version, use gtm_headers_latest

Usage

```
gtm_headers_list(account_id, container_id, includeDeleted = c("TRUE", "FALSE"))
```

Arguments

account_id   Account Id

container_id   Container Id

includeDeleted   Include deleted version - Default False

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/version_headers/list

Other version header functions: gtm_headers_latest()

Examples

```
## Not run:

accountId <- 1234567
containerId <- 7654321


headers <- gtm_headers_list(accountId, containerId)

headersWithDeleted <- gtm_headers_list(accountId, containerId, 'TRUE')


## End(Not run)
```

gtm_tags_create   *Create a new tag*

Description

This takes a tag resource and creates a new tag in GTM.

Usage

```
gtm_tags_create(account_id, container_id, workspace_id, tag_object)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id
tag_object	Tag Object

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/tags/create

Other tag functions: gtm_tags_delete(), gtm_tags_get(), gtm_tags_list(), gtm_tags_revert(), gtm_tags_update()

Examples

```
## Not run:

accountId <- 1234567
containerId <- 7654321
workspaceId <- 5
# In the Tag Object below you will need to substitute your own trigger Id -
# see \code{gtm_triggers_list} and \code{gtm_triggers_create}
tag <- list(
  name = "Example DataLayer Push",
  type = "html",
  tagFiringOption = "oncePerEvent",
  parameter = list(
    list(
      type = "template",
      key = "html",
      value = "<script>\n
                window.dataLayer=window.dataLayer||[];\n
                window.dataLayer.push({\"event\": \"demoEvent\"});\n
                </script>"
    ),
    list(
```

```
      type = "boolean",
      key = "supportDocumentWrite",
      value = "false"
    )
  ),
  firingTriggerId = 20,
  tagFiringOption = "oncePerPage"
)

newTag <- gtm_tags_create(accountId, containerId, workspaceId, tag)

## End(Not run)
```

---

gtm_tags_delete   *Delete a tag*

Description

This deletes a tag in a GTM workspace

Usage

```
gtm_tags_delete(
  account_id,
  container_id,
  workspace_id,
  tag_id,
  force = c("TRUE", "FALSE")
)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id
tag_id	Tag Id
force	Force deletion without user input

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/tags/delete

Other tag functions: gtm_tags_create(), gtm_tags_get(), gtm_tags_list(), gtm_tags_revert(), gtm_tags_update()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 10
tagId <- 100

gtm_tags_delete(accountId, containerId, workspaceId, tagId)

# This will delete tag %s. Are you sure you want to continue?

#  1: Yes
#  2: No

# Selection: 1

# Tag 100 has been deleted.

tagId = 101

gtm_tags_delete(accountId, containerId, workspaceId, tagId, "TRUE")

# Tag 101 has been deleted.


## End(Not run)
```

gtm_tags_get	*Get one tag in a workspace*

Description

This will return a list with the settings of a single tag. If you want to get the information for all tags, use `gtm_tags_list`

Usage

```
gtm_tags_get(account_id, container_id, workspace_id, tag_id)
```

Arguments

`account_id`	Account Id
`container_id`	Container Id
`workspace_id`	Workspace Id
`tag_id`	Tag Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/
[containers/workspaces/tags/get](https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/tags/get)

Other tag functions: `gtm_tags_create()`, `gtm_tags_delete()`, `gtm_tags_list()`, `gtm_tags_revert()`,
`gtm_tags_update()`

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 10
tagId = 100

tag <- gtm_tags_get(accountId, containerId, workspaceId, tagId)


## End(Not run)
```

---

  gtm_tags_list   *List all tags in a workspace*

Description

This will return a data frame all your available tags in a given workspace If you want to get the
information for a single tag, use `gtm_tags_get`

Usage

```
gtm_tags_list(account_id, container_id, workspace_id)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/
[containers/workspaces/tags/list](https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/tags/list)

Other tag functions: `gtm_tags_create()`, `gtm_tags_delete()`, `gtm_tags_get()`, `gtm_tags_revert()`,
`gtm_tags_update()`

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 10

tags <- gtm_tags_list(accountId, containerId, workspaceId)


## End(Not run)
```

---

gtm_tags_revert	*Reverts a tag,*

Description

This reverts a tag back to its original, unmodified state.

Usage

```
gtm_tags_revert(account_id, container_id, workspace_id, tag_id)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id
tag_id	Tag Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/tags/revert

Other tag functions: gtm_tags_create(), gtm_tags_delete(), gtm_tags_get(), gtm_tags_list(), gtm_tags_update()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 10
tagId <- 102
```

```
tag <- gtm_tags_revert(accountId, containerId, workspaceId, tagId)

# Changes to tag 102 have been reverted

## End(Not run)
```

---

gtm_tags_update   *Update an existing tag*

Description

This takes a tag resource and overwrites the existing tag in GTM

Usage

```
gtm_tags_update(account_id, container_id, workspace_id, tag_id, tag_object)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id
tag_id	Tag Id
tag_object	Tag Object

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/tags/update

Other tag functions: gtm_tags_create(), gtm_tags_delete(), gtm_tags_get(), gtm_tags_list(), gtm_tags_revert()

Examples

```
## Not run:

accountId <- 1234567
containerId <- 7654321
workspaceId <- 5
tagId <- 11

tag <- list(
  name = "Example DataLayer Push",
  type = "html",
  tagFiringOption = "oncePerEvent",
  parameter = list(
    list(
```

```
        type = "template",
        key = "html",
        value = "<script>\n    window.dataLayer=window.dataLayer||[];\n
                   window.dataLayer.push({\"event\": \"demoEvent\"});\n
                   </script>"
    ),
    list(
      type = "boolean",
      key = "supportDocumentWrite",
      value = "false"
    )
  ),
  firingTriggerId = 20,
  tagFiringOption = "oncePerPage"
)

updatedTag <- gtm_tags_update(accountId, containerId, workspaceId, tagId, tag)

## End(Not run)
```

---

gtm_triggers_create *Create a new trigger*

Description

This takes a trigger resource and creates a new trigger in GTM.

Usage

```
gtm_triggers_create(account_id, container_id, workspace_id, trigger_object)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id
trigger_object	Trigger Object

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/triggers/create

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/triggers#resource

Other trigger functions: gtm_triggers_delete(), gtm_triggers_get(), gtm_triggers_list(), gtm_triggers_revert(), gtm_triggers_update()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 3

trigger <- list(
  name = 'Demo Trigger',
  type = 'pageview'
)

newTrigger <- gtm_triggers_create(accountId, containerId, workspaceId, trigger)

trigger2 <- list(
name = 'More Complex Trigger',
type = 'pageview',
filter = list(
  list(
    type = 'contains',
    parameter = list(
     list(
       type='template',
       key = 'arg0',
       value='{{Lookup Table}}'
     ),
     list(
       type='template',
       key = 'arg1',
       value='Lookup Value'
     )
    )
   )
  )
 )

newTrigger2 <- gtm_triggers_create(accountId, containerId, workspaceId, trigger2)


## End(Not run)
```

---

gtm_triggers_delete *Delete a trigger*

Description

This deletes a trigger in a GTM workspace

Usage

```
gtm_triggers_delete(
  account_id,
  container_id,
  workspace_id,
  trigger_id,
  force = c(TRUE, FALSE)
)
```

Arguments

`account_id`	Account Id
`container_id`	Container Id
`workspace_id`	Workspace Id
`trigger_id`	trigger Id
`force`	Force deletion without user input

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/triggers/delete

Other trigger functions: `gtm_triggers_create()`, `gtm_triggers_get()`, `gtm_triggers_list()`, `gtm_triggers_revert()`, `gtm_triggers_update()`

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 5
triggerId <- 13

gtm_triggers_delete(accountId, containerId, workspaceId, triggerId)

# This will delete trigger %s. Are you sure you want to continue?

#  1: Yes
#  2: No

# Selection: 1

# trigger 13 has been deleted.

triggerId = 14

gtm_triggers_delete(accountId, containerId, workspaceId, triggerId, "TRUE")

# Trigger 14 has been deleted.
```

```
## End(Not run)
```

---

gtm_triggers_get   *List a single trigger in a workspace*

Description

This will return a list with the settings of a single trigger. If you want to get the information for all
triggers, use `gtm_triggers_list`

Usage

```
gtm_triggers_get(account_id, container_id, workspace_id, trigger_id)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id
trigger_id	trigger Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/
containers/workspaces/triggers/get

Other trigger functions: gtm_triggers_create(), gtm_triggers_delete(), gtm_triggers_list(),
gtm_triggers_revert(), gtm_triggers_update()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 3
triggerID <- 12

trigger <- gtm_triggers_get(accountId, containerId, workspaceId, triggerId)


## End(Not run)
```

---

gtm_triggers_list   *List all triggers in a workspace*

Description

This will return a data frame all your available triggers in a given workspace If you want to get the information for a single trigger, use gtm_triggers_get

Usage

```
gtm_triggers_list(account_id, container_id, workspace_id)
```

Arguments

account_id   Account Id

container_id   Container Id

workspace_id   Workspace Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/triggers/list

Other trigger functions: gtm_triggers_create(), gtm_triggers_delete(), gtm_triggers_get(), gtm_triggers_revert(), gtm_triggers_update()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 3

triggers <- gtm_triggers_list(accountId, containerId, workspaceId)


## End(Not run)
```

---

gtm_triggers_revert   *Reverts a trigger to its original state*

Description

This reverts a trigger back to its original, unmodified state.

Usage

```
gtm_triggers_revert(account_id, container_id, workspace_id, trigger_id)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id
trigger_id	Trigger Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/triggers/revert

Other trigger functions: gtm_triggers_create(), gtm_triggers_delete(), gtm_triggers_get(), gtm_triggers_list(), gtm_triggers_update()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 10
triggerId <- 20

trugger <- gtm_triggers_revert(accountId, containerId, workspaceId, triggerId)

# Changes to trigger 20 have been reverted

## End(Not run)
```

---

gtm_triggers_update   *Update an existing trigger*

Description

This takes a trigger resource and overwrites the existing trigger in GTM

Usage

```
gtm_triggers_update(
  account_id,
  container_id,
  workspace_id,
  trigger_id,
  trigger_object
)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id
trigger_id	trigger Id
trigger_object	trigger Object

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/
[containers/workspaces/triggers/update](https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/triggers/update)

Other trigger functions: `gtm_triggers_create()`, `gtm_triggers_delete()`, `gtm_triggers_get()`,
`gtm_triggers_list()`, `gtm_triggers_revert()`

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 3
triggerId <- 8


updatedTrigger <- list(
name = 'More Complex Trigger - Updated',
type = 'pageview',
filter = list(
  list(
```

```
        type = 'contains',
        parameter = list(
         list(
           type='template',
           key = 'arg0',
           value='{{Lookup Table}}'
         ),
          list(
           type='template',
           key = 'arg1',
           value='New Lookup Value'
          )
        )
      )
     )
   )

 newTrigger2 <- gtm_triggers_update(accountId, containerId, workspaceId, triggerId, updatedTrigger)


 ## End(Not run)
```

---

gtm_variables_create   *Create a new variable*

Description

This takes a variable resource and creates a new variable in GTM.

Usage

```
gtm_variables_create(account_id, container_id, workspace_id, variable_object)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id
variable_object	
	Variable Object

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/variables/create

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/variables#resource-representations

Other variable functions: [gtm_variables_delete()](), [gtm_variables_get()](), [gtm_variables_list()](), [gtm_variables_revert()](), [gtm_variables_update()]()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 3

variable <- list(
  name = 'Custom JS Variable',
  type = 'jsm',
  parameter = list(
    list(
      type = 'template',
      key = 'javascript',
      value = 'function(){\n  return document.querySelector(\"form\").id;\n}'
    )
  )
)

cjsVariable <- gtm_variables_create(accountId, containerId, workspaceId, variable)

variable2 <- list(
name = 'Lookup Table',
type = 'smm',
parameter = list(
  list(
        type = 'boolean',
        key = 'setDefaultValue',
        value = 'false'
    ),
    list(
        type = 'template',
        key = 'input',
        value = '{{Page Hostname}}'
    ),
    list(
        type = 'list',
        key = 'map',
        list = list(
            list(
            type = 'map',
            map = list(list(
                type = 'template',
                key = 'key',
                value = 'google.co.uk'
            ),
            list(
                type = 'template',
                key = 'value',
                value = 'UA-123456-1'
            )
        )
```

```
        ),
        list(
                type = 'map',
                map = list(list(
                    type = 'template',
                    key = 'key',
                    value = 'bing.com'
                ),
                list(
                    type = 'template',
                    key = 'value',
                    value = 'UA-123456-2'
                )
            )
        )
      )
     )
    )
   )

   lookupTable <- gtm_variables_create(accountId, containerId, workspaceId, variable2)


   ## End(Not run)
```

gtm_variables_delete   *Delete a variable*

Description

This deletes a variable in a GTM workspace

Usage

```
gtm_variables_delete(
  account_id,
  container_id,
  workspace_id,
  variable_id,
  force = c(TRUE, FALSE)
)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id
variable_id	Variable Id
force	Force deletion without user input

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/
containers/workspaces/variables/delete

Other variable functions: gtm_variables_create(), gtm_variables_get(), gtm_variables_list(),
gtm_variables_revert(), gtm_variables_update()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 10
variableId <- 30

gtm_variables_delete(accountId, containerId, workspaceId, variableId)

# This will delete tag %s. Are you sure you want to continue?

#  1: Yes
#  2: No

# Selection: 1

# Variable 30 has been deleted.

variableId = 31

gtm_variables_delete(accountId, containerId, workspaceId, variableId, "TRUE")

# Variable 31 has been deleted.


## End(Not run)
```

---

gtm_variables_get *List all variables in a workspace*

Description

This will return a list with the settings of a single variable. If you want to get the information for all
variables, use gtm_variables_list

Usage

```
gtm_variables_get(account_id, container_id, workspace_id, variable_id)
```

Arguments

`account_id`	Account Id
`container_id`	Container Id
`workspace_id`	Workspace Id
`variable_id`	Variable Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/variables/get

Other variable functions: `gtm_variables_create()`, `gtm_variables_delete()`, `gtm_variables_list()`, `gtm_variables_revert()`, `gtm_variables_update()`

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 10
variableId = 100

variable <- gtm_variables_get(accountId, containerId, workspaceId, variableId)


## End(Not run)
```

---

gtm_variables_list   *List all variables in a workspace*

Description

This will return a data frame all your available variables in a given workspace If you want to get the information for a single variable, use `gtm_variables_get`

Usage

```
gtm_variables_list(account_id, container_id, workspace_id)
```

Arguments

`account_id`	Account Id
`container_id`	Container Id
`workspace_id`	Workspace Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/
[containers/workspaces/variables/list](https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/variables/list)

Other variable functions: `gtm_variables_create()`, `gtm_variables_delete()`, `gtm_variables_get()`,
`gtm_variables_revert()`, `gtm_variables_update()`

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 10

variables <- gtm_variables_list(accountId, containerId, workspaceId)


## End(Not run)
```

---

gtm_variables_revert   *Reverts a variable to its original state*

Description

This reverts a variable back to its original, unmodified state.

Usage

```
gtm_variables_revert(account_id, container_id, workspace_id, variable_id)
```

Arguments

`account_id`	Account Id
`container_id`	Container Id
`workspace_id`	Workspace Id
`variable_id`	Variable Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/
[containers/workspaces/variables/revert](https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/variables/revert)

Other variable functions: `gtm_variables_create()`, `gtm_variables_delete()`, `gtm_variables_get()`,
`gtm_variables_list()`, `gtm_variables_update()`

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 10
variableId <- 102

variable <- gtm_variables_revert(accountId, containerId, workspaceId, variableId)

# Changes to variable 22 have been reverted

## End(Not run)
```

---

gtm_variables_update   *Update an existing variable*

Description

This takes a variable resource and overwrites the existing variable in GTM

Usage

```
gtm_variables_update(
  account_id,
  container_id,
  workspace_id,
  variable_id,
  variable_object
)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id
variable_id	Variable Id
variable_object	
	Variable Object

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/
[containers/workspaces/variables/update](https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/variables/update)

Other variable functions: [gtm_variables_create()](), [gtm_variables_delete()](), [gtm_variables_get()](),
[gtm_variables_list()](), [gtm_variables_revert()]()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 3


updatedVariable <- list(
name = 'Lookup Table',
type = 'smm',
parameter = list(
  list(
        type = 'boolean',
        key = 'setDefaultValue',
        value = 'false'
    ),
    list(
        type = 'template',
        key = 'input',
        value = '{{Page Hostname}}'
    ),
    list(
        type = 'list',
        key = 'map',
        list = list(
            list(
            type = 'map',
            map = list(list(
                type = 'template',
                key = 'key',
                value = 'google.co.uk'
            ),
            list(
                type = 'template',
                key = 'value',
                value = 'UA-123456-1'
            )
        )
    ),
    list(
            type = 'map',
            map = list(list(
                type = 'template',
                key = 'key',
                value = 'bing.com'
            ),
            list(
                type = 'template',
                key = 'value',
                value = 'UA-123456-2'
            )
```

```
                )
            ),
            list(
                    type = 'map',
                    map = list(list(
                        type = 'template',
                        key = 'key',
                        value = 'yahoo.com'
                    ),
                    list(
                        type = 'template',
                        key = 'value',
                        value = 'UA-123456-3'
                    )
                )
            )
          )
        )
      )
    )

  newLookupTable <- gtm_variables_create(accountId, containerId, workspaceId, updatedVariable)


  ## End(Not run)
```

---

gtm_versions_delete *Delete a container version*

Description

Deletes a container version

Usage

```
gtm_versions_delete(
  account_id,
  container_id,
  version_id,
  force = c("TRUE", "FALSE")
)
```

Arguments

account_id	Account Id
container_id	Container Id
version_id	Version Id
force	Force deletion without user input

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/versions/delete

Other version functions: gtm_versions_get(), gtm_versions_update()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
versionId <- 6

gtm_versions_delete(accountId, containerId, versionId)

# This will delete Version %s. Are you sure you want to continue?

#  1: Yes
#  2: No

# Selection: 1

# Version 6 has been deleted.

tagId = 7

gtm_versions_delete(accountId, containerId, versionId, "TRUE")

# Version 7 has been deleted.


## End(Not run)
```

---

gtm_versions_get   *List a single version in a workspace*

Description

This returns a single workspace version

Usage

```
gtm_versions_get(account_id, container_id, version_id)
```

Arguments

account_id	Account Id
container_id	Container Id
version_id	Version Id

See Also

[https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/](https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/versions/get)
[containers/versions/get](https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/versions/get)

Other version functions: `gtm_versions_delete()`, `gtm_versions_update()`

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
versionId <- 6

version <- gtm_versions_get(accountId, containerId, versionId)


## End(Not run)
```

---

gtm_versions_publish   *Publish a new container version*

Description

Publishes a container version to be live

Usage

```
gtm_versions_publish(account_id, container_id, version_id)
```

Arguments

account_id	Account Id
container_id	Container Id
version_id	Version Id

See Also

[https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/](https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/versions/publish)
[containers/versions/publish](https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/versions/publish)

Other versions functions: `gtm_versions_setlatest()`, `gtm_versions_undelete()`

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
versionId <- 6

version <- gtm_versions_publish(accountId, containerId, versionId)


## End(Not run)
```

gtm_versions_setlatest

Sets a container version as the latest version

Description

Sets the given container version as the latest version

Usage

```
gtm_versions_setlatest(account_id, container_id, version_id)
```

Arguments

account_id	Account Id
container_id	Container Id
version_id	Version Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/
containers/versions/set_latest

Other versions functions: gtm_versions_publish(), gtm_versions_undelete()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
versionId <- 6

version <- gtm_versions_setlatest(accountId, containerId, versionId)
```

```
## End(Not run)
```

---

gtm_versions_undelete   *Undeletes a container version*

Description

Undeletes a container version

Usage

```
gtm_versions_undelete(account_id, container_id, version_id)
```

Arguments

account_id	Account Id
container_id	Container Id
version_id	Version Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/versions/undelete

Other versions functions: gtm_versions_publish(), gtm_versions_setlatest()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
versionId <- 6

version <- gtm_versions_undelete(accountId, containerId, versionId)


## End(Not run)
```

gtm_versions_update   *Update a container version*

Description

Updates a container Version

Usage

```
gtm_versions_update(
  account_id,
  container_id,
  version_id,
  name = NULL,
  description = NULL
)
```

Arguments

account_id	Account Id
container_id	Container Id
version_id	Version Id
name	Version Name
description	Version Description

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/versions/update

Other version functions: [gtm_versions_delete()](), [gtm_versions_get()]()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
versionId <- 6
newName <- 'My Cool New Version'
newDescription <- 'My new version does something cool'
version <- gtm_versions_update(accountId, containerId, versionId, newName, newDescription)


## End(Not run)
```

---

gtm_workspaces_create   *Create a new workspace*

Description

This creates a new workspace in an esisting container.

Usage

```
gtm_workspaces_create(account_id, container_id, name, description = NULL)
```

Arguments

account_id	Account Id
container_id	Container Id
name	Workspace Name
description	Workspace Description

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/create

Other workspace functions: gtm_workspaces_delete(), gtm_workspaces_get(), gtm_workspaces_list(), gtm_workspaces_preview(), gtm_workspaces_resolve(), gtm_workspaces_status(), gtm_workspaces_submit(), gtm_workspaces_sync(), gtm_workspaces_update()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
name <- 'New Analytics Tags'
description <- 'Adding element visibility tags'

workspace <- gtm_workspaces_create(accountId, containerId, name, description)


## End(Not run)
```

gtm_workspaces_delete   *Delete a workspace*

Description

This irrevocably deletes a workspace and any work in it.

Usage

```
gtm_workspaces_delete(
  account_id,
  container_id,
  workspace_id,
  force = c("TRUE", "FALSE")
)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id
force	Force deletion without user input

See Also

[https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/](https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/delete)
[containers/workspaces/delete](https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/delete)

Other workspace functions: gtm_workspaces_create(), gtm_workspaces_get(), gtm_workspaces_list(),
gtm_workspaces_preview(), gtm_workspaces_resolve(), gtm_workspaces_status(), gtm_workspaces_submit(),
gtm_workspaces_sync(), gtm_workspaces_update()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 84

gtm_workspaces_delete(accountId, containerId, workspaceId)

# This will delete Workspace %s. Are you sure you want to continue?

#  1: Yes
#  2: No

# Selection: 1
```

```
# Workspace 84 has been deleted.

workspaceId = 85

gtm_workspaces_delete(accountId, containerId, workspaceId, "TRUE")

# Workspace 85 has been deleted.


## End(Not run)
```

---

gtm_workspaces_get   *Get workspace metadata*

Description

This returns a list containing the metadata about a single Workspace If you want to get the information for all workspaces, use gtm_workspaces_list

Usage

```
gtm_workspaces_get(account_id, container_id, workspace_id)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/get

Other workspace functions: gtm_workspaces_create(), gtm_workspaces_delete(), gtm_workspaces_list(), gtm_workspaces_preview(), gtm_workspaces_resolve(), gtm_workspaces_status(), gtm_workspaces_submit(), gtm_workspaces_sync(), gtm_workspaces_update()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 5

workspace <- gtm_workspaces_get(accountId, containerId, workspaceId)
```

```
## End(Not run)
```

---

gtm_workspaces_list   *List all workspaces a user has accesss to*

Description

This will return a data frame all your available workspaces in a given container. If you want to get the information for a single workspace, use `gtm_workspaces_get`

Usage

```
gtm_workspaces_list(account_id, container_id)
```

Arguments

account_id   Account Id

container_id   Container Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/list

Other workspace functions: gtm_workspaces_create(), gtm_workspaces_delete(), gtm_workspaces_get(), gtm_workspaces_preview(), gtm_workspaces_resolve(), gtm_workspaces_status(), gtm_workspaces_submit(), gtm_workspaces_sync(), gtm_workspaces_update()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321

workspaces <- gtm_workspaces_list(accountId, containerId)


## End(Not run)
```

gtm_workspaces_preview

Preview the compilation of a container version

Description

Quick previews a workspace by creating a fake container version from all entities in the provided workspace. It returns a container object, and if there are any conflicts, sync errors or compiler errors in the workspace

Usage

```
gtm_workspaces_preview(account_id, container_id, workspace_id)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id

Value

Invisibly, the server response. If successful this is list with one element.

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/quick_preview

Other workspace functions: gtm_workspaces_create(), gtm_workspaces_delete(), gtm_workspaces_get(), gtm_workspaces_list(), gtm_workspaces_resolve(), gtm_workspaces_status(), gtm_workspaces_submit(), gtm_workspaces_sync(), gtm_workspaces_update()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 6

preview <- gtm_workspaces_preview(accountId, containerId, workspaceId)


## End(Not run)
```

```
gtm_workspaces_resolve
```
Resolve workspace conflicts

Description

Resolves a merge conflict for a workspace entity by updating it to the resolved entity passed in the request. This is one of the more complex functions, however if you are using this package for automation you will most probably want to keep the version from the current workspace not the base version, which is shown in hte example.

Usage

```
gtm_workspaces_resolve(
  account_id,
  container_id,
  workspace_id,
  tag = NULL,
  trigger = NULL,
  variable = NULL,
  folder = NULL,
 changeStatus = c("added", "changeStatusUnspecified", "deleted", "none", "updated")
)
```

Arguments

`account_id`	Account Id
`container_id`	Container Id
`workspace_id`	Workspace Id
`tag`	A Tag Object
`trigger`	A Trigger Object
`variable`	A Variable Object
`folder`	A Folder Object
`changeStatus`	Represents how the entity has been changed in the workspace.

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/resolve_conflict

Other workspace functions: `gtm_workspaces_create()`, `gtm_workspaces_delete()`, `gtm_workspaces_get()`, `gtm_workspaces_list()`, `gtm_workspaces_preview()`, `gtm_workspaces_status()`, `gtm_workspaces_submit()`, `gtm_workspaces_sync()`, `gtm_workspaces_update()`

Examples

```
## Not run:

 # Comlpexities arise converting the list created in R into the correct JSON for GTM,
 # specifically for the Monitoring Metadata in Tags, so this is brought out, converted
 # separately and then included in the tag before it is resolved.

accountId <- 1234567
containerId <- 7654321
workspaceId <- 10

mergeConflicts <- gtm_workspaces_sync(accountId, containerId, workspaceId)

for(i in 1:nrow(mergeConflicts$mergeConflict$entityInWorkspace)){
    tag <- NULL
    trigger <- NULL
    variable <- NULL
    folder <- NULL
    obj <- mergeConflicts$mergeConflict$entityInWorkspace[i,]
    if(!is.na(obj$tag$path)){
        tag <- obj$tag
        if(is.null(tag$monitoringMetadata$map)){
        tag <- tag %>% select(-monitoringMetadata)
        tag <- as.list(tag) %>%
               lapply(function(x){if(is.null(x[[1]])){x<-NA}else{x<-x}}) %>%
               lapply(function(x) x[!is.na(x)])
        } else{
        metadataMap <- tag$monitoringMetadata$map[[1]]
        metadata <- list(
            type = 'map',
            map = metadataMap
        )
        tag <- as.list(tag) %>%
                lapply(function(x){if(is.null(x[[1]])){x<-NA}else{x<-x}}) %>%
                lapply(function(x) x[!is.na(x)])
        tag$monitoringMetadata <- metadata
        }

    }
    if(!is.na(obj$trigger$path)){
        trigger <- obj$trigger
        trigger <- as.list(trigger) %>%
                   lapply(function(x){if(is.null(x[[1]])){x<-NA}else{x<-x}}) %>%
                   lapply(function(x) x[!is.na(x)])
    }
    if(!is.na(obj$variable$path)){
        variable <- obj$variable
        variable <- as.list(variable) %>%
                    lapply(function(x){if(is.null(x[[1]])){x<-NA}else{x<-x}}) %>%
                    lapply(function(x) x[!is.na(x)])
    }
```

```
    if(!is.na(obj$folder$path)){
        folder <- obj$folder
        folder <- as.list(folder) %>%
                  lapply(function(x){if(is.null(x[[1]])){x<-NA}else{x<-x}}) %>%
                  lapply(function(x) x[!is.na(x)])
    }
    changeStatus <- obj$changeStatus

    resolve <- gtm_workspaces_resolve(account_id = accountId,
                                      container_id = containerId,
                                      workspace_id = workspaceId,
                                      tag = tag,
                                      trigger = trigger,
                                      variable = variable,
                                      folder = folder,
                                      changeStatus = changeStatus)
  }

## End(Not run)
```

gtm_workspaces_status   *See workspace changes*

Description

Finds conflicting and modified entities in the workspace.

Usage

```
gtm_workspaces_status(account_id, container_id, workspace_id)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/
[containers/workspaces/getStatus](https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/getStatus)

Other workspace functions: [gtm_workspaces_create()](), [gtm_workspaces_delete()](), [gtm_workspaces_get()](),
[gtm_workspaces_list()](), [gtm_workspaces_preview()](), [gtm_workspaces_resolve()](), [gtm_workspaces_submit()](),
[gtm_workspaces_sync()](), [gtm_workspaces_update()]()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 6

workspace <- gtm_workspaces_status(accountId, containerId, workspaceId)


## End(Not run)
```

`gtm_workspaces_submit`   *Create a new container version*

Description

This creates a new version of the GTM container, but does not publish it as a live version. The name and notes will be permanently visible for the container version.

Usage

```
gtm_workspaces_submit(
  account_id,
  container_id,
  workspace_id,
  name,
  notes = NULL
)
```

Arguments

`account_id`	Account Id
`container_id`	Container Id
`workspace_id`	Workspace Id
`name`	Version Name
`notes`	Version Notes

See Also

<https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/create_version>

Other workspace functions: `gtm_workspaces_create()`, `gtm_workspaces_delete()`, `gtm_workspaces_get()`, `gtm_workspaces_list()`, `gtm_workspaces_preview()`, `gtm_workspaces_resolve()`, `gtm_workspaces_status()`, `gtm_workspaces_sync()`, `gtm_workspaces_update()`

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 6
name <- 'New Analytics Tags'
notes <- 'Adding element visibility tags and scroll tracking'

workspace <- gtm_workspaces_update(accountId, containerId, workspaceId, name, notes)


## End(Not run)
```

---

gtm_workspaces_sync   *Bring a workspace in line with the latest version*

Description

Syncs a workspace to the latest container version by updating all unmodified workspace entities and displaying conflicts for modified entities

Usage

```
gtm_workspaces_sync(account_id, container_id, workspace_id)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/sync

Other workspace functions: gtm_workspaces_create(), gtm_workspaces_delete(), gtm_workspaces_get(), gtm_workspaces_list(), gtm_workspaces_preview(), gtm_workspaces_resolve(), gtm_workspaces_status(), gtm_workspaces_submit(), gtm_workspaces_update()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 6

syncResult <- gtm_workspaces_sync(accountId, containerId, workspaceId)


## End(Not run)
```

---

gtm_workspaces_update   *Update a workspace*

Description

This updates the name or description of an existing workspace.

Usage

```
gtm_workspaces_update(
  account_id,
  container_id,
  workspace_id,
  name,
  description = NULL
)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id
name	Workspace Name
description	Workspace Description

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/update

Other workspace functions: [gtm_workspaces_create()](), [gtm_workspaces_delete()](), [gtm_workspaces_get()](), [gtm_workspaces_list()](), [gtm_workspaces_preview()](), [gtm_workspaces_resolve()](), [gtm_workspaces_status()](), [gtm_workspaces_submit()](), [gtm_workspaces_sync()]()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 6
name <- 'New Analytics Tags'
description <- 'Adding element visibility tags and scroll tracking'

workspace <- gtm_workspaces_update(accountId, containerId, workspaceId, name, description)


## End(Not run)
```

---

gtm_zones_create   *Create a new zone*

Description

This takes a zone resource and creates a new zone in GTM.

Usage

```
gtm_zones_create(account_id, container_id, workspace_id, zone_object)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id
zone_object	Zone Object

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/zones/create

Other zone functions: [gtm_zones_delete()](), [gtm_zones_get()](), [gtm_zones_list()](), [gtm_zones_revert()](), [gtm_zones_update()]()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 10
```

```
zone <- list(
    accountId=accountId,
    containerId=containerId,
    workspaceId=workspaceId,
    name="Marketing Zone",
    childContainer=list(
      list(
        publicId="GTM-ABCDEF",
        nickname="My Marketing Container"
    )
  )
)

new_zone <- gtm_zones_create(accountId, containerId, workspaceId, zone)


## End(Not run)
```

---

gtm_zones_delete   *Delete a zone*

Description

This deletes a zone in a GTM workspace

Usage

```
gtm_zones_delete(
  account_id,
  container_id,
  workspace_id,
  zone_id,
  force = c("TRUE", "FALSE")
)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id
zone_id	Zone Id
force	Force deletion without user input

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/zones/delete

Other zone functions: gtm_zones_create(), gtm_zones_get(), gtm_zones_list(), gtm_zones_revert(), gtm_zones_update()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 11
zoneId <- 3

gtm_zones_delete(accountId, containerId, workspaceId, zoneId, "TRUE")


## End(Not run)
```

---

gtm_zones_get   *List a single zone in a workspace*

Description

This will return a list with the settings of a single zone. If you want to get the information for all zones, use gtm_zones_list

Usage

```
gtm_zones_get(account_id, container_id, workspace_id, zone_id)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id
zone_id	Zone Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/zones/get

Other zone functions: gtm_zones_create(), gtm_zones_delete(), gtm_zones_list(), gtm_zones_revert(), gtm_zones_update()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 11
zoneId <- 12

zone <- gtm_zones_get(accountId, containerId, workspaceId, zoneId)

## End(Not run)
```

---

gtm_zones_list   *List all zones in a workspace*

Description

This will return a data frame all your available zones in a given workspace If you want to get the
information for a single zone, use `gtm_zones_get`

Usage

```
gtm_zones_list(account_id, container_id, workspace_id)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/zones/list

Other zone functions: [gtm_zones_create()](), [gtm_zones_delete()](), [gtm_zones_get()](), [gtm_zones_revert()](),
[gtm_zones_update()]()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 11

zones <- gtm_zones_list(accountId, containerId, workspaceId)
```

```
## End(Not run)
```

---

gtm_zones_revert   *Reverts a zone to its original state*

Description

This reverts a zone back to its original, unmodified state.

Usage

```
gtm_zones_revert(account_id, container_id, workspace_id, zone_id)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id
zone_id	Zone Id

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/zones/revert

Other zone functions: gtm_zones_create(), gtm_zones_delete(), gtm_zones_get(), gtm_zones_list(), gtm_zones_update()

Examples

```
## Not run:
accountId <- 1234567
containerId <- 7654321
workspaceId <- 11
zoneId <- 5

zone <- gtm_zones_revert(accountId, containerId, workspaceId, zoneId)

## End(Not run)
```

---

gtm_zones_update   *Update an existing zone*

Description

This takes a zone resource and overwrites the existing zone in GTM

Usage

```
gtm_zones_update(account_id, container_id, workspace_id, zone_id, zone_object)
```

Arguments

account_id	Account Id
container_id	Container Id
workspace_id	Workspace Id
zone_id	Zone Id
zone_object	Zone Object

See Also

https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/
[containers/workspaces/zones/update](https://developers.google.com/tag-platform/tag-manager/api/v2/reference/accounts/containers/workspaces/zones/update)

Other zone functions: [gtm_zones_create()](), [gtm_zones_delete()](), [gtm_zones_get()](), [gtm_zones_list()](),
[gtm_zones_revert()]()

---

variables_list   *Built-In Variable Keys*

Description

These are the variable names accepted when enabling/disabling Built-In Variables

Arguments

advertiserId	Advertiser Id - Firebase
advertisingTrackingEnabled	
	Advertiser Tracking Enabled - Firebase
ampBrowserLanguage	
	Browser Language - AMP
ampCanonicalHost	
	Cannonical Host - AMP
ampCanonicalPath	
	Canonical Path - AMP

```
ampCanonicalUrl
                Canonical URL - AMP
ampClientId     Page Client ID - AMP
ampClientMaxScrollX
                Scroll Width - AMP
ampClientMaxScrollY
                Scroll Height - AMP
ampClientScreenHeight
                Screen Height - AMP
ampClientScreenWidth
                Screen Width - AMP
ampClientScrollX
                Scroll Left - AMP
ampClientScrollY
                Scroll Top - AMP
ampClientTimestamp
                Client Timestamp - AMP
ampClientTimezone
                Client Timezone - AMP
ampGtmEvent     AMP Event - AMP
ampPageDownloadTime
                Page Downlad Time - AMP
ampPageLoadTime
                Page Load Time - AMP
ampPageViewId   Page View Id - AMP
ampReferrer     Document Referrer - AMP
ampTitle        Document Title - AMP
ampTotalEngagedTime
                Total Engaged Time - AMP
appId           App Id - Firebase
appName         App Name- Firebase
appVersionCode  App Version Code - Firebase
appVersionName  App Version Name - Firebase
builtInVariableTypeUnspecified
                Undefined
clickClasses    Click Classes
clickElement    Click Element
clickId         Click Id
clickTarget     Click Target
clickText       Click Text
clickUrl        Click URL
containerId     Container ID
```

```
containerVersion
                Container Version
debugMode       Debug Mode
deviceName      Device Name
elementVisibilityFirstTime
                Element First Visible
elementVisibilityRatio
                Percent Visible
elementVisibilityRecentTime
                Element Last Visible
elementVisibilityTime
                On Screen Duration
environmentName
                Environment Name
errorLine       Error Line
errorMessage    Error Message
errorUrl        Error URL
event           Event
eventName       Event Name
firebaseEventParameterCampaign
                Campaign Name - Firebase
firebaseEventParameterCampaignAclid
                Click Id (aclid) - Firebase
firebaseEventParameterCampaignAnid
                Ad Network ID (anid) - Firebase
firebaseEventParameterCampaignClickTimestamp
                Campaign Click Timestamp - Firebase
firebaseEventParameterCampaignContent
                Campaign Content - Firebase
firebaseEventParameterCampaignCp1
                Campaign Custom Parameter 1 - Firebase
firebaseEventParameterCampaignGclid
                Google Click Id (gclid) - Firebase
firebaseEventParameterCampaignSource
                Campaign Source - Firebase
firebaseEventParameterCampaignTerm
                Campaign Search Term - Firebase
firebaseEventParameterCurrency
                In-app Purchase Currency - Firebase
firebaseEventParameterDynamicLinkAcceptTime
                Dynamic Link Accept Time - Firebase
firebaseEventParameterDynamicLinkLinkid
                Dynamic Link ID - Firebase
```

```
firebaseEventParameterNotificationMessageDeviceTime
                Notification Device Time - Firebase
firebaseEventParameterNotificationMessageId
                Notification Message Id - Firebase
firebaseEventParameterNotificationMessageName
                Notification Message Name - Firebase
firebaseEventParameterNotificationMessageTime
                Notification Message Time - Firebase
firebaseEventParameterNotificationTopic
                Notification Message Topic - Firebase
firebaseEventParameterPreviousAppVersion
                Previous Application Version - Firebase
firebaseEventParameterPreviousOsVersion
                Previous OS Version - Firebase
firebaseEventParameterPrice
                In-app Purchase Price - Firebase
firebaseEventParameterProductId
                In-app Purchase Product ID - Firebase
firebaseEventParameterQuantity
                In-app Purchase Quantity - Firebase
firebaseEventParameterValue
                In-app Purchase Value - Firebase
formClasses     Form Classes
formElement     Form Element
formId          Form Id
formTarget      Form Target
formText        Form Text
formUrl         Form URL
historySource   History Source
htmlId          HTML ID
language        Language
newHistoryFragment
                New History Fragment
newHistoryState
                New History State
newHistoryUrl   New History URL
oldHistoryFragment
                Old History Fragment
oldHistoryState
                Old History Stat
oldHistoryUrl   Old History URL
osVersion       OS Version
```

pageHostname	Page Hostname
pagePath	Page Path
pageUrl	Page URL
platform	Platform
randomNumber	Random Number
referrer	Referrer
resolution	Screen Resolution
scrollDepthDirection	
	Scroll Depth Direction
scrollDepthThreshold	
	Scroll Depth Threshold
scrollDepthUnits	
	Scroll Depth Units
sdkVersion	SDK Version - Firebase
videoCurrentTime	
	Video Current Time
videoDuration	Video Duration
videoPercent	Video Percent
videoProvider	Video Provider
videoStatus	Video Status
videoTitle	Video Title
videoUrl	Video URL
videoVisible	Video Visible

See Also

Other built in variable functions: `gtm_builtin_create()`, `gtm_builtin_delete()`, `gtm_builtin_list()`, `gtm_builtin_revert()`

Index