Package ‘vstsr’

October 12, 2022

Type Package
Title Access to 'Azure DevOps' API via R
Version 1.1.0
It enables the extraction of information about repositories, build and release definitions and individual releases.
It also helps create repositories and work items within a project without logging into 'Azure DevOps'.
There is the ability to use any API service with a shell for any non-predefined call.
License GPL-2
URL https://github.com/ashbaldry/vstsr,
    https://docs.microsoft.com/en-us/rest/api/azure/devops
BugReports https://github.com/ashbaldry/vstsr/issues
Imports R6, httr, RCurl, jsonlite
Encoding UTF-8
RoxygenNote 7.1.2
Suggests testthat
Language en-GB
NeedsCompilation no
Author Ashley Baldry [aut, cre]
Maintainer Ashley Baldry <arbaldry91@gmail.com>
Repository CRAN
Date/Publication 2021-11-08 22:40:02 UTC

R topics documented:

  vstsr ................................................................. 2
  vsts_account ..................................................... 3
  vsts_auth_key .................................................... 4
vstsr: A package connecting R and Azure DevOps

Description
This package takes a look at the Azure DevOps API calls and wraps them around in easy to use R functions. This includes looking at projects, repositories, work items, and eventually sections such as builds and releases.

Details
For more information about Azure DevOps APIs, take a look at https://docs.microsoft.com/en-us/rest/api/vsts

Projects
This looks at the projects available in a visual studio instance. This lets you change between any project that you might have access to.

Repositories
This looks primarily at the Git repositories available, and whether you want to create a new repository or delete an existing one.

Work Items
This will track any existing work item for a project, and also the ability to create a new work item for a project. Useful if working within a development team and automate inclusion of creating bugs upon creating any certain R error.

Releases
This will look at the releases available for a project, both the definitions and the actual releases. There is the ability to deploy the created releases to a new environment.
Custom

For any non-predefined API service available in the package, it is always possible to run the `vsts_run_command` which will use any enabled Azure DevOps API call.

---

**vsts_account**

*Azure DevOps Account*

---

**Description**

Azure DevOps Account

**Arguments**

- **user** username for the Azure DevOps account
- **pass** password for the Azure DevOps account
- **domain** domain name for the Azure DevOps account
- **project** (optional) project name within the domain of the Azure DevOps account
- **repo** (optional) repository name with the project of the Azure DevOps domain

**Format**

An `R6Class` generator object

**Details**

For the majority of functions that are within this `vsts_account` object, you can get help about the query or body parameter with `?vsts_<function name>`.

**Examples**

```r
## Not run:
proj <- vsts_account$new(
  "<username>", "<password>", "<domain>",
  "<project>", "<repo>"
)
str(proj)

## End(Not run)
```
vsts_auth_key  Azure DevOps Authentication Key

Description

Creation of a Azure DevOps authentication key that will be used when running any of the API calls.

Usage

vsts_auth_key(user, pass)

Arguments

user  username to access Azure DevOps project
pass  password to access Azure DevOps project

Details

For more information about authentication check https://docs.microsoft.com/en-us/rest/api/azure/devops/?view=azure-devops-rest-6.1#create-the-request

Value

An authentication key string in the form of 'Basic <Base 64 of user:pass>'

Examples

# Using credentials
auth_key <- vsts_auth_key("<username>", "<password>")

# Using PAT token
auth_key <- vsts_auth_key(NULL, "<token>")

vsts_create_release  Azure DevOps Project Release Information

Description

These functions will allow you to create releases from Azure DevOps.

Usage

vsts_create_release(domain, project, auth_key, body)
Arguments

- **domain**: The name of the Azure DevOps organization.
- **project**: the name of the project in domain to look at
- **auth_key**: authentication key generated by using `vsts_auth_key`
- **body**: a list of extra parameters that can need to be sent to the API call (* mandatory):
  - **artifacts** [list] Sets list of artifact to create a release. Check Details for more information.
  - **definitionId** [integer] Sets definition Id to create a release.
  - **description** [character] Sets description to create a release.
  - **isDraft** [logical] Sets 'true' to create release in draft mode, 'false' otherwise.
  - **manualEnvironments** [character] Sets list of environments to manual as condition.
  - **properties** [list] The class represents a property bag as a collection of key-value pairs.
  - **reason** [character] Sets reason to create a release.

Details

The **artifacts** object within the body contains two items:

- **alias** [character] Sets alias of artifact.
- **instanceReference** [list] Sets instance reference of artifact. e.g. for build artifact it is build number.


Examples

```r
## Not run:
# Add in own details to get a non-NULL output
auth_key <- vsts_auth_key("<username>", "<password>")
art_list <- list(
  list(alias = "Art1", instanceReference = list(id = 1)),
  list(alias = "Art2", instanceReference = list(id = 2))
)
body <- list(
  definitionId = 1, description = "R API Release",
  artifacts = I(art_list)
)
vsts_create_release("domain", "project", auth_key, body)

## End(Not run)
```
vsts_create_workitem  Azure DevOps Project Work Items

Description
These functions will allow you to scrape work item information from a particular Azure DevOps project.

Usage
vsts_create_workitem(domain, project, item_type, auth_key, ...)

Arguments
- domain: The name of the Azure DevOps organization.
- project: the name of the project in domain to look at
- item_type: the type of work item to be created
- auth_key: authentication key generated by using vsts_auth_key
- ...: arguments passed to vsts_get_workitem_fields

Details
For more information about work item API calls check https://docs.microsoft.com/en-us/rest/api/azure/devops/wit/work-items.

vsts_deploy_release  Azure DevOps Project Release Environment Information

Description
These functions will allow you to run release environment tasks from Azure DevOps.

Usage
vsts_deploy_release(domain, project, release, env, auth_key)

Arguments
- domain: The name of the Azure DevOps organization.
- project: the name of the project in domain to look at
- release: the release ID of the release
- env: the release environment ID to release on
- auth_key: authentication key generated by using vsts_auth_key
vsts_get_build_defs

Details

Examples
## Not run:
# Add in own details to get a non-NULL output
auth_key <- vsts_auth_key("<username>", "<password>"
vsts_deploy_release("domain", "project", auth_key, 1, 1)

## End(Not run)

vsts_get_build_defs Azure DevOps Project Build Definition Information

Description
These functions will allow you to scrape build definition information from Azure DevOps.

Usage
vsts_get_build_defs(domain, project, auth_key, query = NULL)

Arguments
domain The name of the Azure DevOps organization.
project the name of the project in domain to look at
auth_key authentication key generated by using vsts_auth_key
query a list of extra parameters that can be sent to the API call. Check details for access to list of options.

Details
For more information about the build definition API calls check https://docs.microsoft.com/en-us/rest/api/azure/devops/build/definitions/list.

Examples
## Not run:
# Add in own details to get a non-NULL output
auth_key <- vsts_auth_key("<username>", "<password>"
vsts_get_build_defs("domain", "project", auth_key)

## End(Not run)
vsts_get_commits  Azure DevOps Project Git Repositories

Description
These functions will allow you to scrape git repository information from Azure DevOps.

Usage
vsts_get_commits(domain, project, repo, auth_key, query = NULL)

Arguments
- domain: The name of the Azure DevOps organization.
- project: the name of the project in domain to look at
- repo: the name of the repository in project to look at
- auth_key: authentication key generated by using vsts_auth_key
- query: a list of extra parameters that can be sent to the API call:
  - branch: [character] the name of a branch in the repository (cannot combine with commit)
  - commit: [character] the id of a commit in the repository (cannot combine with branch)
  - itemPath: [character] path of an item in the repository
  - committer: [character] name of the person who committed the change
  - author: [character] name of the author
  - fromDate: [Date] start date to search from
  - toDate: [Date] end date to search from

Details
For more information about git repository API calls check https://docs.microsoft.com/en-us/rest/api/azure/devops/git/commits/get-commits

Examples
## Not run:
# Add in own details to get a non-NULL output
auth_key <- vsts_auth_key("<username>", "<password>")
vsts_get_commits("domain", "project", "repo", auth_key)

## End(Not run)
vsts_get_projects  Azure DevOps Projects

Description
These functions will allow you to scrape project information from Azure DevOps.

Usage

vsts_get_projects(domain, auth_key, quiet = FALSE)

Arguments

- **domain**: The name of the Azure DevOps organization.
- **auth_key**: Authentication key generated by using `vsts_auth_key`.
- **quiet**: logical whether want general running information from printing. Any issue with the API call will still show up if set to TRUE.

Details
For more information about project API calls check https://docs.microsoft.com/en-us/rest/api/azure/devops/core/Projects.

vsts_get_releases  Azure DevOps Project Release Information

Description
These functions will allow you to scrape releases from Azure DevOps.

Usage

vsts_get_releases(domain, project, auth_key, query = NULL)

vsts_get_release(domain, project, release, auth_key)

Arguments

- **domain**: The name of the Azure DevOps organization.
- **project**: the name of the project in domain to look at
- **auth_key**: authentication key generated by using `vsts_auth_key`.
- **query**: a list of extra parameters that can be sent to the API call.
- **release**: Release Definition ID.
vsts_get_release_defs

Details


Examples

## Not run:

```r
# Add in own details to get a non-NULL output
auth_key <- vsts_auth_key("<username>", "<password>")
vsts_get_releases("domain", "project", auth_key)
```

## End(Not run)

---

vsts_get_release_defs  Azure DevOps Project Release Definition Information

Description

These functions will allow you to scrape release definition information from Azure DevOps.

Usage

```r
vsts_get_release_defs(domain, project, auth_key, quiet = FALSE)
```

Arguments

- **domain**: The name of the Azure DevOps organization.
- **project**: the name of the project in domain to look at
- **auth_key**: authentication key generated by using vsts_auth_key
- **quiet**: logical whether want general running information from printing. Any issue with the API call will still show up if set to TRUE

Details

For more information about release definition API calls check https://docs.microsoft.com/en-us/rest/api/azure/devops/release/releases.

Examples

## Not run:

```r
# Add in own details to get a non-NULL output
auth_key <- vsts_auth_key("<username>", "<password>")
vsts_get_release_defs("domain", "project", auth_key)
```

## End(Not run)
vsts_get_repos Azure DevOps Project Repositories

Description

These functions will allow you to scrape project information from Azure DevOps.

Usage

vsts_get_repos(domain, project, auth_key, quiet = FALSE)

vsts_create_repo(domain, project, repo, auth_key, quiet = FALSE)

vsts_delete_repo(domain, project, repo, auth_key, quiet = FALSE)

Arguments

domain The name of the Azure DevOps organization
project Project ID or project name
auth_key authentication key generated by using vsts_auth_key
quiet logical whether want general running information from printing. Any issue with
the API call will still show up if set to TRUE
repo the name of the repository in project to look at. Leave as "" to get all repositories within all projects of the domain

Details

For more information about repository API calls check https://docs.microsoft.com/en-us/rest/api/azure/devops/git.

Examples

## Not run:
# Add in own details to get a non-NULL output
auth_key <- vsts_auth_key("<username>", "<password>")

# Get repo list
vsts_get_repos("domain", "project", auth_key)

# Create new repo
vsts_create_repo("domain", "project", "repo", auth_key)

# Delete existing repo
vsts_delete_repo("domain", "project", "repo", auth_key)

## End(Not run)
vsts_get_workitems  Azure DevOps Project Get Work Items

Description

These functions will allow you to scrape work item information from a particular Azure DevOps project.

Usage

vsts_get_workitems(domain, project, auth_key, id, query = list())

vsts_get_workitem(domain, project, auth_key, id)

Arguments

domain  The name of the Azure DevOps organization.
project  Project ID or project name
auth_key  authentication key generated by using vsts_auth_key
id  ID of the work item to retrieve
query  a list of extra parameters that can be sent to the API call:
  ids  [character] a comma-separated list of up to 200 IDs of the work items to get
  fields  [character] a comma-separated list of up to 100 fields to get with each work item. If not specified, all fields with values are returned. Calculated fields such as Attached File Count must be specifically queried for using this parameter.
  asOf  [Date] gets the work items as they existed at this time
  ErrorPolicy  [character] determines if the call will throw an error when encountering a work item (default behavior) that doesn’t exist (throw) or simply omit it (omit)

Details

For more information about work item API calls check https://docs.microsoft.com/en-us/rest/api/azure/devops/wit/work-items.
Description

This contains all the fields required of any work item in a visual studio project and helps add/rename the fields of the selected work item.

Usage

vsts_get_workitem_fields(
    System.Title,
    System.Description,
    System.TeamProject,
    System.AreaPath,
    System.IterationPath,
    Microsoft.VSTS.Common.Priority,
    ...
)

Arguments

System.Title  [character] title of the Azure DevOps work item
System.Description  [character] description of the Azure DevOps work item
System.TeamProject  [character] name of the Azure DevOps project
System.AreaPath  [character] path of the Azure DevOps work item
System.IterationPath  [character] name of the Azure DevOps iteration path
Microsoft.VSTS.Common.Priority  [integer] priority of the work item - 1 to 4
...  other fields that might have been missed out originally

Details

For more information about work item fields API calls check https://docs.microsoft.com/en-us/rest/api/azure/devops/wit/Fields.
**vsts_run_command**  

**Azure DevOps Custom API Calls**

**Description**

For any requirement not currently in place in the vstsr package, then this function will allow you to use the relevant API call without any extra requirements.

For the most part it is just a shell of **VERB** but will have the auth_key set up already.

**Usage**

```r
vsts_run_command(url, verb, auth_key, body = NULL, query = NULL)
```

**Arguments**

- `url` the URI of the API call to run
- `verb` name of the verb to use
- `auth_key` authentication key generated by using **vsts_auth_key**
- `body` check **VERB** for more details. If the object is a named list, then it will be transformed into a JSON string so that can be added to the call. Use [https://docs.microsoft.com/en-us/rest/api/azure/devops](https://docs.microsoft.com/en-us/rest/api/azure/devops) to find out any required parameter for the body.
- `query` a list of extra parameters that can be sent to the API call. If not required then leave as **NULL**

**Examples**

```r
## Not run:
auth_key <- vsts_auth_key("<username>", "<password>")
# Get commits of a repository
URL <- file.path(
  "https://dev.azure.com",
  domain,
  project,
  ".apis/git/repositories",
  repository_id,
  "commits?api-version=5.0"
)
vsts_run_command(URL, "GET", auth_key)

## End(Not run)
```
Index

* data
  vsts_account, 3

R6Class, 3

VERB, 14
vsts_account, 3
vsts_auth_key, 4, 5–12, 14
vsts_create_release, 4
vsts_create_repo (vsts_get_repos), 11
vsts_create_workitem, 6
vsts_delete_repo (vsts_get_repos), 11
vsts_deploy_release, 6
vsts_get_build_defs, 7
vsts_get_commits, 8
vsts_get_projects, 9
vsts_get_release (vsts_get_releases), 9
vsts_get_release_defs, 10
vsts_get_releases, 9
vsts_get_repos, 11
vsts_get_workitem (vsts_get_workitems), 12
vsts_get_workitem_fields, 6, 13
vsts_get_workitems, 12
vsts_run_command, 3, 14
vstsr, 2