» bitbucket_user

Provdes a way to fetch data on a current user via there username, uuid or Display Name.

» Example Usage

```
# Manage your repository
data "bitbucket_user" "reviewer" {
  username = "gob"
}
```

» Argument Reference

The following arguments are supported:

- `username` - (Required) the username have write access to.

» Exports

- `uuid` the uuid that bitbucket users to connect a user to various objects
- `display_name` the display name that the user wants to use for GDPR
- `nickname` typically the username but not always true.

» bitbucket_default_reviewers

Provides support for setting up default reviewers for your repository. You must however have the UUID of the user available. Since Bitbucket has removed usernames from its APIs the best case is to use the UUID via the data provider.

» Example Usage

```
# Manage your repositories default reviewers
data "bitbucket_user" "reviewer" {
  username = "gob"
}

resource "bitbucket_default_reviewers" "infrastructure" {
  owner      = "myteam"
  repository = "terraform-code"
```

```
  reviewers = [
    "${data.bitbucket_user.reviewer.uuid}",
  ]
}
```

» Argument Reference

The following arguments are supported:

- `owner` - (Required) The owner of this repository. Can be you or any team you have write access to.
- `repository` - (Required) The name of the repository.
- `reviewers` - (Required) A list of reviewers to use.

» bitbucket_hook

Provides a Bitbucket hook resource.

This allows you to manage your webhooks on a repository.

» Example Usage

```
# Manage your repositories hooks
resource "bitbucket_hook" "deploy_on_push" {
  owner       = "myteam"
  repository  = "terraform-code"
  url         = "https://mywebhookservice.mycompany.com/deploy-on-push"
  description = "Deploy the code via my webhook"

  events = [
    "repo:push",
  ]
}
```

» Argument Reference

The following arguments are supported:

- `owner` - (Required) The owner of this repository. Can be you or any team you have write access to.
- `repository` - (Required) The name of the repository.
- `url` - (Required) Where to POST to.
- `description` - (Required) The name / description to show in the UI.

- `events` - (Required) The event you want to react on.

» bitbucket_repository

Provides a Bitbucket repository resource.

This resource allows you manage your repositories such as scm type, if it is private, how to fork the repository and other options.

» Example Usage

```
# Manage your repository
resource "bitbucket_repository" "infrastructure" {
  owner = "myteam"
  name  = "terraform-code"
}
```

If you want to create a repository with a CamelCase name, you should provide a seperate slug

```
# Manage your repository
resource "bitbucket_repository" "infrastructure" {
  owner = "myteam"
  name  = "TerraformCode"
  slug  = "terraform-code"
}
```

» Argument Reference

The following arguments are supported:

- `owner` - (Required) The owner of this repository. Can be you or any team you have write access to.
- `name` - (Required) The name of the repository.
- `slug` - (Optional) The slug of the repository.
- `scm` - (Optional) What SCM you want to use. Valid options are hg or git. Defaults to git.
- `is_private` - (Optional) If this should be private or not. Defaults to `true`.
- `website` - (Optional) URL of website associated with this repository.
- `language` - (Optional) What the language of this repository should be.
- `has_issues` - (Optional) If this should have issues turned on or not.
- `has_wiki` - (Optional) If this should have wiki turned on or not.

- `project_key` - (Optional) If you want to have this repo associated with a project.
- `fork_policy` - (Optional) What the fork policy should be. Defaults to allow_forks.
- `description` - (Optional) What the description of the repo is.
- `pipelines_enabled` - (Optional) Turn on to enable pipelines support

» Computed Arguments

The following arguments are computed. You can access both `clone_ssh` and `clone_https` for getting a clone URL.

» Import

Repositories can be imported using their `owner/name` ID, e.g.

```
$ terraform import bitbucket_repository.my-repo my-account/my-repo
```

» bitbucket_branch_restriction

Provides a Bitbucket branch restriction resource.

This allows you for setting up branch restrictions for your repository.

» Example Usage

```
# Manage your repositories branch restrictions
resource "bitbucket_branch_restriction" "master" {
  owner      = "myteam"
  repository = "terraform-code"

  kind = "push"
  pattern = "master"
}
```

» Argument Reference

The following arguments are supported:

- `owner` - (Required) The owner of this repository. Can be you or any team you have write access to.
- `repository` - (Required) The name of the repository.

- `kind` - (Required) The type of restriction that is being applied. List of possible stages is here.
- `pattern` - (Required) The pattern to determine which branches will be restricted.
- `users` - (Optional) A list of users to use.
- `groups` - (Optional) A list of groups to use.

» bitbucket_project

This resource allows you to manage your projects in your bitbucket team.

» Example Usage

```
# Manage your repository
resource "bitbucket_project" "devops" {
  owner = "my-team"
  name  = "devops"
  key = "DEVOPS"
}
```

» Argument Reference

The following arguments are supported:

- `owner` - (Required) The owner of this project. Can be you or any team you have write access to.
- `name` - (Required) The name of the project
- `key` - (Required) The key used for this project
- `description` - (Optional) The description of the project
- `is_private` - (Optional) If you want to keep the project private - defaults to true

» bitbucket_repository_variable

This resource allows you to setup pipelines variables to manage your builds with. Once you have enabled pipelines on your repository you can then further setup variables here to use.

» Example Usage

```
resource "bitbucket_repository" "monorepo" {
    owner = "gob"
    name = "illusions"
    pipelines_enable = true
}

resource "bitbucket_repository_variable" "debug" {
    key = "DEBUG"
    value = "true"
    repository = "${bitbucket_repository.monorepo.id}"
    secured = false
}
```

» Argument Reference

- `key` - (Required) The key of the key value pair
- `value` - (Required) The value of the key
- `repository` - (Required) The repository ID you want to put this variable onto.
- `secuired` - (Optional) If you want to make this viewable in the UI.
- `uuid` - (Computed) The UUID of the variable