» Data Source: azuread_application

Use this data source to access information about an existing Application within Azure Active Directory.

NOTE: If you're authenticating using a Service Principal then it must have permissions to both `Read and write all applications` and `Sign in and read user profile` within the `Windows Azure Active Directory` API.

» Example Usage

```
data "azuread_application" "test" {
  name = "My First AzureAD Application"
}

output "azure_ad_object_id" {
  value = "${data.azuread_application.test.id}"
}
```

» Argument Reference

- `object_id` - (Optional) Specifies the Object ID of the Application within Azure Active Directory.
- `name` - (Optional) Specifies the name of the Application within Azure Active Directory.

NOTE: Either an `object_id` or `name` must be specified.

» Attributes Reference

- `id` - the Object ID of the Azure Active Directory Application.
- `application_id` - the Application ID of the Azure Active Directory Application.
- `available_to_other_tenants` - Is this Azure AD Application available to other tenants?
- `identifier_uris` - A list of user-defined URI(s) that uniquely identify a Web application within it's Azure AD tenant, or within a verified custom domain if the application is multi-tenant.
- `oauth2_allow_implicit_flow` - Does this Azure AD Application allow OAuth2.0 implicit flow tokens?
- `object_id` - the Object ID of the Azure Active Directory Application.

- `reply_urls` - A list of URLs that user tokens are sent to for sign in, or the redirect URIs that OAuth 2.0 authorization codes and access tokens are sent to.

» Data Source: azuread_service_principal

Gets information about an existing Service Principal associated with an Application within Azure Active Directory.

NOTE: If you're authenticating using a Service Principal then it must have permissions to both `Read and write all applications` and `Sign in and read user profile` within the `Windows Azure Active Directory` API.

» Example Usage (by Application Display Name)

```
data "azuread_service_principal" "test" {
  display_name = "my-awesome-application"
}
```

» Example Usage (by Application ID)

```
data "azuread_service_principal" "test" {
  application_id = "00000000-0000-0000-0000-000000000000"
}
```

» Example Usage (by Object ID)

```
data "azuread_service_principal" "test" {
  object_id = "00000000-0000-0000-0000-000000000000"
}
```

» Argument Reference

The following arguments are supported:

- `application_id` - (Optional) The ID of the Azure AD Application for which to create a Service Principal.
- `object_id` - (Optional) The ID of the Azure AD Service Principal.
- `display_name` - (Optional) The Display Name of the Azure AD Application associated with this Service Principal.

NOTE: At least one of `application_id`, `display_name` or `object_id` must be specified.

» Attributes Reference

The following attributes are exported:

- `id` - The Object ID for the Service Principal.

» azuread_application

Manages an Application within Azure Active Directory.

NOTE: If you're authenticating using a Service Principal then it must have permissions to both `Read and write all applications` and `Sign in and read user profile` within the `Windows Azure Active Directory` API.

» Example Usage

```
resource "azuread_application" "test" {
  name                       = "example"
  homepage                   = "https://homepage"
  identifier_uris            = ["https://uri"]
  reply_urls                 = ["https://replyurl"]
  available_to_other_tenants = false
  oauth2_allow_implicit_flow = true
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The display name for the application.
- `homepage` - (optional) The URL to the application's home page. If no homepage is specified this defaults to `https://{name}`.
- `identifier_uris` - (Optional) A list of user-defined URI(s) that uniquely identify a Web application within it's Azure AD tenant, or within a verified custom domain if the application is multi-tenant.
- `reply_urls` - (Optional) A list of URLs that user tokens are sent to for sign in, or the redirect URIs that OAuth 2.0 authorization codes and access tokens are sent to.

- `available_to_other_tenants` - (Optional) Is this Azure AD Application available to other tenants? Defaults to `false`.
- `oauth2_allow_implicit_flow` - (Optional) Does this Azure AD Application allow OAuth2.0 implicit flow tokens? Defaults to `false`.

» Attributes Reference

The following attributes are exported:

- `application_id` - The Application ID.

» Import

Azure Active Directory Applications can be imported using the `object id`, e.g.

```
terraform import azuread_application.test 00000000-0000-0000-0000-000000000000
```

» azuread_service_principal

Manages a Service Principal associated with an Application within Azure Active Directory.

NOTE: If you're authenticating using a Service Principal then it must have permissions to both `Read and write all applications` and `Sign in and read user profile` within the `Windows Azure Active Directory` API.

» Example Usage

```
resource "azuread_application" "test" {
  name                       = "example"
  homepage                   = "http://homepage"
  identifier_uris            = ["http://uri"]
  reply_urls                 = ["http://replyurl"]
  available_to_other_tenants = false
  oauth2_allow_implicit_flow = true
}

resource "azuread_service_principal" "test" {
  application_id = "${azuread_application.test.application_id}"
}
```

» Argument Reference

The following arguments are supported:

- `application_id` - (Required) The ID of the Azure AD Application for which to create a Service Principal.

» Attributes Reference

The following attributes are exported:

- `id` - The Object ID for the Service Principal.
- `display_name` - The Display Name of the Azure Active Directory Application associated with this Service Principal.

» Import

Azure Active Directory Service Principals can be imported using the `object id`, e.g.

```
terraform import azuread_service_principal.test 00000000-0000-0000-0000-000000000000
```

» azuread_service_principal_password

Manages a Password associated with a Service Principal within Azure Active Directory.

NOTE: If you're authenticating using a Service Principal then it must have permissions to both `Read and write all applications` and `Sign in and read user profile` within the `Windows Azure Active Directory` API.

» Example Usage

```
resource "azuread_application" "test" {
  name                       = "example"
  homepage                   = "http://homepage"
  identifier_uris            = ["http://uri"]
  reply_urls                 = ["http://replyurl"]
  available_to_other_tenants = false
  oauth2_allow_implicit_flow = true
}

resource "azuread_service_principal" "test" {
```

```
  application_id = "${azuread_application.test.application_id}"
}

resource "azuread_service_principal_password" "test" {
  service_principal_id = "${azuread_service_principal.test.id}"
  value                = "VT=uSgbTanZhyz@%nL9Hpd+Tfay_MRV#"
  end_date             = "2020-01-01T01:02:03Z"
}
```

» Argument Reference

The following arguments are supported:

- `service_principal_id` - (Required) The ID of the Service Principal for which this password should be created. Changing this field forces a new resource to be created.
- `value` - (Required) The Password for this Service Principal.
- `end_date` - (Required) The End Date which the Password is valid until, formatted as a RFC3339 date string (e.g. `2018-01-01T01:02:03Z`). Changing this field forces a new resource to be created.
- `key_id` - (Optional) A GUID used to uniquely identify this Key. If not specified a GUID will be created. Changing this field forces a new resource to be created.
- `start_date` - (Optional) The Start Date which the Password is valid from, formatted as a RFC3339 date string (e.g. `2018-01-01T01:02:03Z`). If this isn't specified, the current date is used. Changing this field forces a new resource to be created.

» Attributes Reference

The following attributes are exported:

- `id` - The Key ID for the Service Principal Password.

» Import

Service Principal Passwords can be imported using the `object id`, e.g.

```
terraform import azuread_service_principal_password.test 00000000-0000-0000-0000-0000000000
```

NOTE: This ID format is unique to Terraform and is composed of the Service Principal's Object ID and the Service Principal Password's Key ID in the format `{ServicePrincipalObjectId}/{ServicePrincipalPasswordKeyId}`.