

# Cloud.Haskzure.Core

Core contains the core components of the project.

## The AzureResource typeclass.

Copyright	(c) Nashwan Azhari, 2016
License	Apache 2.0
Maintainer	aznashwan@yahoo.com
Stability	experimental
Portability	POSIX, Win32
Safe Haskell	None
Language	Haskell2010

```
class (ToJSON r, FromJSON r) =>
  AzureResource r where
```

#

AzureResource is the typeclass to which all AzureResource resource datatypes must comply in order to be deployed. It should be directly serializable to/from JSON.

### Minimal complete definition

```
rID, rName, rLocation, rType
```

### Methods

```
rID :: r -> String
```

#

rID returns the String ID of the AzureResource:

```
rName :: r -> String
```

#

rName returns the String name of the AzureResource:

```
rLocation :: r -> String
```

#

rLocation returns the normalized String location of the AzureResource:

```
rType :: r -> String
```

#

rType returns the String Type of the AzureResource in 'Provider/ResourceType' form:

### Contents

The AzureResource typeclass.

The Resource ADT.

Authentication data structures and functions:

## The Resource ADT.

```
data Resource a
```

#

Resource defines the core Resource ATD which will be used to model Azure resources.

Resource is the basic implementation of a generic Azure resource:

### Constructors

```
Resource
```

<b>resID :: String</b>	resID is the ID of the Resource:
<b>resName :: String</b>	resName is the Name of the Resource:
<b>resGroup :: String</b>	resGroup is the Name of the Resource Group in which the <b>Resource</b> lives.
<b>resLocation :: String</b>	resLocation is the Location of the Resource:
<b>resType :: String</b>	resType is the Type of Resource:
<b>resProperties :: a</b>	resProperties are the properties of the Resource:

#### Instances

<b>Eq (Resource a)</b>	#
<b>Show a =&gt; Show (Resource a)</b>	#
<b>ToJSON a =&gt; ToJSON (Resource a)</b>	#
<b>FromJSON a =&gt; FromJSON (Resource a)</b>	#

## Authentication data structures and functions:

**data Credentials** #

The standard set of credentials required for Azure authentication.

#### Constructors

##### Credentials

<b>tenantId :: ByteString</b>	the ID of the AD tenant within which the application is registered.
<b>clientId :: ByteString</b>	the client ID of the application as configured inside the Azure AD.
<b>subscriptionId :: ByteString</b>	the ID of the subscription to be used.
<b>clientSecret :: ByteString</b>	the client secret (aka one of the application's key).

#### Instances

<b>Show Credentials</b>	#
-------------------------	---

**data Token** #

The datatype representing an Azure API token.

#### Constructors

##### Token

<b>token :: ByteString</b>	the <b>String</b> representation of the API token.
<b>expiresOn :: Integer</b>	the <b>Integer</b> representing the absolute epoch time moment in which the token expires.
<b>tokenType :: ByteString</b>	the type of the token; shall be <b>Bearer</b> for all intents and purposes.

## Instances

[+ Show Token](#) | #

[+ FromJSON Token](#) | #

```
getToken :: Credentials -> IO Token
```

 | #

Requests and deserializes an API **Token**.

```
createOrUpdate :: ToJSON a => Credentials -> Resource a -> IO ()
```

 | #

creates or updates an Azure **Resource** by issuing a PUT request on the appropriate URL using the given **Credentials**.

```
get :: FromJSON a => Credentials -> Resource a -> IO (Resource a)
```

 | #

gets the given Azure **Resource** by issuing a GET request on the appropriate URL using the given **Credentials**.

```
delete :: Credentials -> Resource a -> IO ()
```

 | #

deletes the given Azure **Resource** by issuing a DELETE request on the appropriate URL using the given **Credentials**.