Authentication and Encryption

Last Modified on 15/03/2019 2:34 pm AEDT

Authentication

Although some third-party systems will allow public anonymous interaction most do not. Instead, most systems require that you establish some form of account. Then at the time the API is called, some form of authentication information is passed so that the third-party server knows that the API is being called on behalf of that account.

ReadiNow supports the following authentication models:

- None
  - No authentication information is passed
- OAuth 2.0
  - This is a widely used authentication standard.
  - The main benefit of OAuth is that permission can be delegated to ReadiNow without providing your third-party username and password to ReadiNow
  - However, it is also the most complex authentication system to configure
  - OAuth 2.0 can only be used with encrypted (HTTPS) requests
    That is, unencrypted (HTTP) APIs are explicitly prohibited for OAuth 2.0.
  - ReadiNow stores the OAuth details encrypted (specifically, the client secret, the access token and the refresh token)
- Basic authentication
  - This is an old part of the HTTP standard, but still widely used
  - A username and password must be provided to the ReadiNow platform
  - These are stored, and sent with every call to the API
  - Basic authentication is not encrypted in transmission unless it is used in conjunction with HTTPS
  - If the account password is changed with the third-party provider, then the new password needs to be entered into ReadiNow (the password is stored in encrypted form)
- API Key authentication
  - This is a simple authentication technique that is widely used by third-party services
  - The third-party provider provides some piece of text called a key
  - This key is then passed as part of the URL for example: http://server/service?apikey=12345
  - ReadiNow API keys allow you to specify the parameter name (for example

apikey) and the key value for the library, which it then includes for every API in the library
    - API keys are not encrypted in transmission unless used in conjunction with HTTPSReadiNow stores API keys encrypted in the database - but only if they are entered properly in the proper API Key password field (API keys that have been typed into the web address directly are not stored encrypted)
- Custom authentication
    - For more use in advanced scenarios (e.g. APIs whose authentication mechanism does not utilise one of the standards listed above).API callout library uses a specific API callout to perform an authentication request and retrieve an access token
    - The API callout library specifies expressions that can be used to extract an access token and/or error messages from an authentication response

Failure Responses

All APIs return an output called the "HTTP Response Status". This is the response code according to the list of HTTP response codes. If there is a problem during authorization, then the remote server will typically return a response status of `401`, which means `Unauthorized`.

The third-party server might also provide a message describing the reason for failure, which can be accessed via the "HTTP Response Body" string output.

Possible causes for authentication failure include:

- The wrong authentication method was used, or it was configured incorrectly
- The password may be incorrect
- The third-party account may have an invalid status, such as expired, or locked
- The account may not have permission to use the API
- Refer to the documentation and support of the third-party API service

Alternatively, the API may authorize successfully, which is to say it correctly identifies who the caller is, but that caller may be disallowed from performing the specific task that was requested. In that case, the server would typically respond with a `403` (`Forbidden`) response code.

Note that a poorly implemented third-party API may also give some other unpredictable result, such as returning a 200 OK message, or an internal error, when everything is not OK with authentication.

Encryption

Web requests may be made using HTTP or HTTPS addresses. For HTTPS requests, the communication between ReadiNow and the third-party server will be encrypted (whereas HTTP is unencrypted).

ReadiNow API Callouts will only communicate with HTTPS APIs if they have a valid publicly-recognised certificate.

Depending on the type of authorisation method, the ReadiNow server needs to store confidential data. The following are stored encrypted in the database: password, API key, OAuth secrets / tokens, and custom authentication tokens.