Authentication

Terminology

Identifier Token

simple, unique identifier for each plan

Secret Key

phoenix generated random, unique 64 character key

TOTP

Time-based One-Time Password using secret key

Authentication Guidelines

Every request to the Phoenix API must contain your Identifier token and TOTP token. Unauthenticated requests may return a `404 Not Found`, instead of a `403 Forbidden`, in some places. This is to prevent accidental data leakage about private resources.

All requests must come from an allowable IP address. Plans must submit the IP addresses being used for the API to the Phoenix staff for registration and approval, before requests will be accepted.

TOTP Authentication

```
$ curl -i -H 'User-Agent: RFA' 'https://phoenix.scdhhs.gov/api/v1/authentication/ping?identifier_token=IDENTIFIER&access_token=TOTP'
```

TOTPs are generated by your application based on the secret key. Phoenix will accept TOTPs for the previous time interval, current time interval and 1 future time interval. This drift is to account for time skews and traffic congestion. If the TOTP token is invalid, it will be treated as a `404 Not Found` or `403 Forbidden` where appropriate.

After 30 days of **inactivity**, secret keys will be automatically revoked. If this happens, you must follow the Getting Setup scenario in the Authentication Flow. This revocation does not occur on the Staging environment.

After 60 working days, the current secret key will partially expire and all TOTPs generated from it may only be used on a single API endpoint. This API endpoint will revoke the current secret key, programmatically generate a new one, and return it in the response. Your application must use the newly generated secret key for TOTPs for all future requests. This expiration does not occur on the Staging environment.

TOTP Algorithm

RFC 6238 is the TOTP algorithm being used for authentication.

Epoch

Unix epoch (standard)

Interval
30 seconds (standard)

Hash method
HMAC-SHA-1 (standard)

Secret key length
64 bytes

Digits expected
10 (not standard)

Example code may be forthcoming, as time permits. Various examples already exist around the internet.

Authentication Flow

In this flow we are stripping out all other API requests and only focusing on the actions your application should perform, in order to be compliant with our authentication guidelines. At the end, we will also cover special scenarios, such as: 'What should we do if we believe our application's secret key was compromised?'

Forthcoming as RFA completes development.

Getting Setup

1. Your Identifier token should have been distributed to you by the Prime team. If this is not the case, please email the Prime team. This identifier is used as the `identifier_token` for requests that require authentication. It will **not** change after it has been assigned to you.

2. You must request the initial secret key by email to both the Prime team and developers@rfa.sc.gov. If your key has been revoked due to inactivity or other means, you must manually email for another 'initial' secret key. The initial secret key is only used to re-generate a new secret key that can then be used to access the full API. Phoenix administrators can only view initial secret keys. Once you re-generate a new key through the API, it is non-recoverable by the Phoenix team. You must receive a new initial secret key and repeat the API re-generation process.

3. You must send a request from the IP address you wish to use to access the API via `/api/v1/authentication/ip_addresses`. This will verify that the originating IP address matches the IP address in the parameters. It will then create a record in Phoenix under your Identifier token with that IP address and await approval by the Phoenix team. Once approved you may start using the API with that IP address.

4. At this point, you should have a Identifier token, new secret key, and approved IP addresses. This completes the setup phase.

Help, our secret key was compromised!

1. Please make sure to follow any other regulations or guidelines associated with your business when following these steps.

2. Immediately attempt to issue a new secret key through the API! If this request succeeds, skip the next step.

3. Immediately, call the SCDHHS Prime team and explain the situation. They will contact RFA or revoke your existing key themselves. They can then generate a new secret key, following the Getting Setup guide.

4. Once you have a brand new secret key, verify the IP addresses that are verified for your account. Report any malicious or unknown IP addresses to the SCDHHS Prime team for removal.

5. Provide the SCDHHS Prime team of the possible compromised time frame. We will need to perform an audit in coordination with you, to determine if any information was leaked or maliciously added to the Phoenix system.

Authentication API Parameters

Parameters

Name	Type	Required	Description
identifier_token	String	✔	Unique identifier given out to each plan.
access_token	String	✔	10-digit TOTP generated from the secret key

The following types of requests require proper authentication parameters:

- All Production requests
- Staging requests will soon require your identifier_token for all requests; you may still use the test access token given out before, instead of the proper access_token. Updated 1/6/2014

How to integrate with other POST, PUT, or DELETE API requests

The API expects the authentication parameters to be at the root level of JSON payload. If an API request is asking for the following payload:

```
{"electronic_referral": { "attribute": "value", "other_id": 526 } }
```

You would modify it to include the authentication parameters as so (substituting your actual authentication parameter values):

```
{ "identifier_token": "TOKEN", "access_token": "TOTP", "electronic_referral": { "attribute": "value", "other_id": 526 } }
```

What about GET requests that don't require a JSON payload?

You would add the authentication parameters as part of the query string. See the authentication ping API endpoint for an example.

API Reference

› Request a new secret key

Revoke your current secret key and request a new one from Phoenix. The following conditions apply to this endpoint:

- Active and expired keys may access this endpoint.
- Keys manually given to an api user by a member of the Phoenix team may only use this endpoint.
- Keys generated by this endpoint are not viewable by any user of Phoenix.

```
POST /api/v1/authentication/tokens
```

Example Response

```
{"token": "abcde12345fghi67890"}
```

› Determine status of current secret key

This endpoint will accept TOTPs from active, or expired keys. If the TOTP is valid for your current secret key, it will return the status of that key: active or expired. The production environment will also ensure the request comes from an approved IP address.

```
GET /api/v1/authentication/token?identifier_token=IDENTIFIER&access_token=TOTP
```

Example Response

```
{"token": {"state": "expired"}}
```

› Request list of ip addresses associated with your user and their status

There are 3 possible states for an IP address. Only requests from `active` addresses will be accepted in production.

new
Submitted to API, but not approved for use by SCDHHS
active
Approved for use by SCDHHS (api will accept requests from this IP)
revoked
Revoked through API or by SCDHHS

```
GET /api/v1/authentication/ip_addresses?identifier_token=IDENTIFIER&access_token=TOTP
```

Example Response

```
[ {"ip": "127.0.0.1", "state": "new"}, {"ip": "192.168.0.38", "state": "active"}, {"ip": "192.168.0.11", "state": "revoked"} ]
```

› Add new ip addresses to your allowed connection list

This request does **not** require IP address verification. Any IP address, with your authentication credentials may add an ip address to the allowed connection list. However, it must first be approved by SCDHHS, before it will be useable for other api endpoints.

```
POST /api/v1/authentication/ip_addresses
```

Input

Name	Type	Description
ip_addresses	Array	**Required.** Array of IP address strings - `["127.0.0.1", "192.168.0.11"]`

Example Request

```
{"ip_addresses": [
  "127.0.0.1",
  "192.168.0.11" ] }
```

Example Response

```
[{"ip": "127.0.0.1", "state": "new"}, {"ip": "192.168.0.11", "state": "new"}]
```

ⓘ Validations and Errors

ip_addresses must be an array (422 return code)

```
{ "error":"validation error",
  "detail": ["ip_addresses must be an array"] }
```

› Delete ip addresses from your allowed connection list

```
DELETE /api/v1/authentication/ip_addresses
```

Input

Name	Type	Description
ip_addresses	Array	**Required.** Array of IP address strings - `["127.0.0.1", "192.168.0.11"]`

Example Response

```
[{"ip": "127.0.0.1", "state": "revoked"}]
```

› Test authentication

This endpoint checks all authentication critieria for the appropriate environment:

Production
Valid, approved IP address
Valid TOTP
Active secret key

Staging
Valid TOTP
Active secret key

```
GET /api/v1/authentication/ping?identifier_token=IDENTIFIER&access_token=TOTP
```

Example Response

```
"pong"
```

▼ Overview
Authentication
▶ Prime