
Cagenix-Web Documentation

Release 0.1a1

Jason Myers for Prime Notion Technologies

January 28, 2014

1	Bid Overview	3
1.1	Backend	3
1.2	Frontend	3
2	Developer Overview	5
2.1	Libraries	5
2.2	General Application Setup (__init__.py)	5
3	Models Overview	7
3.1	Activation Model	7
3.2	Advertisement Model	7
3.3	Evaluation Model	8
3.4	Patient Model	8
3.5	PatientEvaluationAnswer Model	8
3.6	Practice Model	9
3.7	Role Model	9
3.8	User Model	10
4	API Overview	11
4.1	Base URL, HTTPS, and Versioning	11
4.2	Credentials	11
4.3	GET Requests	12
4.4	Headers	12
4.5	Header Formulas	13
4.6	Response Codes	13
5	Patients API	15
5.1	Create a patient	15
5.2	Retrieve Patient details	16
5.3	Update a patient	17
5.4	Delete a patient	18
6	Practitioners API	19
6.1	Create a Practitioner	19
6.2	Retrieve Practitioner details	20
6.3	Update a Practitioner	21
6.4	Delete a Practitioner	22
6.5	Authenticate a Practitioner	22
7	Evaluations API	25
7.1	Create an Evaluation	25

7.2	Retrieve Evaluation details	26
7.3	Update an Evaluation	27
7.4	Delete a patient	28
8	Users Blueprint Overview	29
8.1	Users Models	29
9	Patients Blueprint Overview	31
9.1	Patients Models	31
10	Indices and tables	33
	Python Module Index	35
	Index	37

Contents:

BID OVERVIEW

We're needing a web application that acts as a server-side component to a data collection mobile application. It also has a front-end for administrative purposes. This is intentionally vague thanks to NDAs, etc., but it is this "simple". The requirements are completely solid and we don't tolerate scope creep...

1.1 Backend

The back-end of web app, through a JSON API, needs to:

- Store "activation codes" for the mobile apps associated with a person's business. When a user downloads an app, they have to enter this code.
- Allow a user to register using an activation code, some contact information, and other basic information. The activation code is expired but still associated with the user. The user must open an email sent to them to activate their registration.
- Validate user's login credentials, returning a auth token that must be passed with all JSON requests.
- Associate with a user another type of user in a parent-child relationship. This other type of user (let's call this a customer) just has basic information.
- Associate a customer evaluation with a customer. This customer evaluation contains a few question-answer fields (about 20).
- Associate a product choice with a customer. There are three product choices: "good", "better", "best" if you like.
- Allow the app to retrieve a list of advertisement banners placed in specified areas of the app (think left, right, or bottom). A "hit count" for the ad banner is retained.

1.2 Frontend

The front-end of the web app needs to:

- Allow an administrator to log in.
- Allow an administrator to create an activation code and associate very simple branding attributes with it (primary, secondary colors).
- Allow an administrator to upload advertisement banners.
- Allow a user to perform a password reset if they have forgotten their password. This simply allows them to log back in to the mobile app; they do not log in to the web app.

If you have any questions, let me know.

DEVELOPER OVERVIEW

The following document covers the internals of the Cagenix-Web application. Below is a list of libraries used in the application, and the general application settings.

2.1 Libraries

- Flask - Web Framework [Documentation](#)
- Flask-SQLAlchemy - Simplifies the usage of SQLAlchemy within the Flask framework. Check out the [Road to Enlightenment](#) for more details. [SQLAlchemy Docs](#)
- WTForms - Provides a flexible way of handling forms [Documentation](#)
- Flask-WTF - Provides a simple integration with WTForms [Documentation](#)
- Jinja2 - The preferred templating language for Flask [Documentation](#)
- Flask-Security - A very powerful collection of User Authentication/Authorization/Accounting [Documentation](#)
- Passlib - A strong password hashing utility [Documentation](#)
- Unicodcsv - Used to handle csv files that contain unicode characters [Documentation](#)

It's built using the [Flask Large App How To](#) and [Matt Wright's Guide](#)

2.2 General Application Setup (`__init__.py`)

This file contains the application setup for SSLify, Flask-SQLAlchemy, and Flask-Security.

```
cagenix.create_user()
cagenix.handle_splash(*args, **kwargs)
cagenix.logout()
    This handles the logout for flask_security and redirects to login
cagenix.not_found(error)
    returns a the 404.html template on any 404 error with in the app.
```

Parameters `error` – The error details

Returns 404.html

Return type template

MODELS OVERVIEW

The following document covers the models used in the Cagenix-Web application.

- *Activation Model*
- *Advertisement Model*
- *Evaluation Model*
- *Patient Model*
- *PatientEvaluationAnswer Model*
- *Practice Model*
- *Role Model*
- *User Model*

3.1 Activation Model

Field Name	Type (Length)	Description
id	Integer	Primary Key
code	String(255)	Activation Code

Virtuals

Field Name	Type	Description
practitioner	One2One	Virtual to User/practitioner
practice	Many2One	Virtual to Practice

3.2 Advertisement Model

Field Name	Type (Length)	Description
id	Integer	Primary Key
name	String(80)	Banner Name
asset_location	String(80)	Location of asset or resource name
position	String(80)	Screen Position

3.3 Evaluation Model

Field Name	Type (Length)	Description
id	Integer	Primary Key
evaluation_id	UUID	Used to group questions into a single evaluation.
question_order	Integer	Position of question within an Evaluation
question_text	Text	Text of the question
answer_one	String(255)	First answer choice
answer_two	String(255)	Second answer choice
answer_three	String(255)	Second answer choice
answer_four	String(255)	Second answer choice
answer_custom	Text	Free for all answer

3.4 Patient Model

Field Name	Type (Length)	Description
id	Integer	Primary Key
first_name	String(80)	First Name
last_name	String(80)	Last Name
address_one	String(255)	First address line
address_two	String(255)	Second address line
city	String(255)	City
state	String(255)	State
zip_code	String(10)	ZIP Code in 5-4 format
email	String(255)	email address
active	Boolean	Active Flag for archiving
choice_one	String(255)	The “best” choice
choice_two	String(255)	The “better” choice
choice_three	String(255)	The “good” choice

Virtuals

Field Name	Type	Description
practitioner	Many2One	Virtual to User assigned as dentist
practice	Many2One	Virtual to Practice
evaluation_answer	Many2One	Virtual to PatientEvaluationAnswer

3.5 PatientEvaluationAnswer Model

Field Name	Type (Length)	Description
id	Integer	Primary Key
evaluation_id	UUID	Used to group answers into a single evaluation.
answer	String(255)	Answer to question
active	Boolean	Active Flag for archiving

Virtuals

Field Name	Type	Description
question	Many2One	Question Asked
patient	Many2One	User Answering Question

3.6 Practice Model

Field Name	Type (Length)	Description
id	Integer	Primary Key
practice_name	String(255)	Practice Name
address_one	String(255)	First address line
address_two	String(255)	Second address line
city	String(255)	City
state	String(255)	State
zip_code	String(10)	ZIP Code in 5-4 format
url	String(255)	Web Address
primary_color	String(255)	Primary Branding Color
secondary_color	String(255)	Secondary Branding Color
phone	String(12)	Phone
active	Boolean	Active Flag for archiving

Virtuals

Field Name	Type	Description
practitioner	One2Many	Virtual to User

3.7 Role Model

Field Name	Type (Length)	Description
id	Integer	Primary Key
name	String(80)	Name
description	String(255)	Short description of the Role's purpose

Virtuals

Field Name	Type	Description
practitioner	One2Many	Virtual to User

3.8 User Model

Field Name	Type (Length)	Description
id	Integer	Primary Key
first_name	String(80)	First Name
last_name	String(80)	Last Name
address_one	String(255)	First address line
address_two	String(255)	Second address line
city	String(255)	City
state	String(255)	State
zip_code	String(10)	ZIP Code in 5-4 format
email	String(255)	email address
password	String(255)	encrypted password string
secret	String(255)	Secret key used for API
activation_code	String(255)	Code used for initial account setup
active	Boolean	Active Flag for archiving
confirmed_at	DateTime	UTC DateTime when user confirmed their account.
last_login_at	DateTime	UTC DateTime of last login
current_login_at	DateTime	UTC DateTime of current login if active
last_login_ip	String(255)	Last IP accessed from
current_login_ip	String(255)	Current IP accessed from
login_count	Integer	Number of Logins

Virtuals

Field Name	Type	Description
patients	One2Many	Virtual to Patient
roles	One2Many	Virtual to Role

API OVERVIEW

The following document will walk you through using the Cagenix API within your application.

4.1 Base URL, HTTPS, and Versioning

4.1.1 Base URI

The Cagenix API root is located at: <https://cagenix-web.herokuapp.com/api/v1/>

4.1.2 HTTP and HTTPS

The Cagenix API is served over HTTP and HTTPS during testing. The final deployment will be HTTPS only, prefer to testing that method.

4.1.3 Versioning

Currently the Cagenix API does not require any type of versioning in the requests as it is in the URI.

4.1.4 Message Body

When submitting a request to the Cagenix API with a message body, the variables within that body should be UTF8 encoded and within a JSON object.

4.2 Credentials

In order to access the Cagenix API each developer/user must request a key and a secret from the application admins. These two keys will be used in the following formulas which will provide the properly formatted header values required for all POST, PUT, PATCH, and DELETE requests. All GET requests require the key, timestamp, and signature url parameters.

4.3 GET Requests

A properly formed GET request includes the key timestamp, and signature url parameters. The key is the key provided by the application admins. The signature is made of the HTTP-VERB + SECRET + Canonicalized Resource encoded to UTF8 hashed via sha256 and base 64 encoded.

URL Param	Description
key	This is the key provided by the application admins.
times-tamp	This is the date and time, ISO format, used in your header formulas. An example is 2012-10-29T01:30:20Z. Notice the T and Z, with the T separating the date and time and Z ending the string.
signature	The header signature is a string that combines the body hash, date and time, along with your secret, encoded to UTF8 and hashed using sha256.

Note: Signature = URLEncode(Base64(SHA-256(UTF-8-Encoding-Of('GET' + SECRET + URI))))

Signature Example

```
base64(sha256(utf8('GET'+'d59b8a775f0dcfd985edff5b5106e2a3'+'/api/v1/patients/1')))  
would result in  
PHNoYTI1NiBIQVNIIG9iamVjdCBAIDB4MTAzNDhhYTcwPg==
```

URL Example

```
/api/v1/patients/1?signature=PHNoYTI1NiBIQVNIIG9iamVjdCBAIDB4MTAzNDhhYTcwPg==&timestamp=2014-01-26T00:00:00Z
```

This assumes:

```
Key is d36a3127fc9f5fa169e911b9ab5b46eb  
Secret is d59b8a775f0dcfd985edff5b5106e2a3  
Time is 00:26:49 on 1/26/2014
```

4.4 Headers

When interacting with the Cagenix web-service, the following headers are required to be submitted with each request.

Header Element	Description
Content-Type	application/json Currently XML and other formats are not available.
key	This is the key provided by the application admins.
datetime	This is the date and time, ISO format, used in your header formulas. An example is 2012-10-29T01:30:20Z. Notice the T and Z, with the T separating the date and time and Z ending the string.
body_hash	This is the encoding of the body contents through the following formula. If there are no body contents to be sent, encode and send an empty body array.
signature	The header signature is a string that combines the body hash, date and time, along with your secret, encoded to UTF8 and hashed using sha256.
lang	The language of the request content. Example is 'en'.

4.5 Header Formulas

Header Element	Formula
datetime	The date and time is a simple formatting: yyyy-mm-ddThh:mm:ssZ
body_hash	json body content UTF8 encoded, then sha256 hashed.
signature	body_hash added to datetime added to secret key, encoded to UTF8 then sha256 hashed.

4.6 Response Codes

4.6.1 GET Request Response Codes

CODE	Response	Meaning
200	OK	The request was successful and the body contains what you asked for.
401	Unauthorized	Your credentials do not authorize you to access the requested info.
404	Not Found	The service you requested was not found on our server.
500	Server Error	There was an internal error, if it continues, please contact us.
503	Service Unavailable	The service requested is temporarily down. Try again later.

4.6.2 POST/PUT Request Response Codes

CODE	Response	Meaning
200	OK	The request was successful and the body contains what you asked for.
201	Created	The resource you requested by create was successfully created.
400	Bad Request	Something was wrong in your request, check the message for details.
401	Unauthorized	Your credentials do not authorize you to access the requested info.
404	Not Found	The service you requested was not found on our server.
405	Method Not Allowed	You tried POSTing or PUTting to a service that can't accept data.
500	Server Error	There was an internal error, if it continues, please contact us.

4.6.3 DELETE Request Response Codes

CODE	Response	Meaning
204	OK	The request was successfully deleted.
400	Bad Request	Something was wrong in your request, check the message for details.
401	Unauthorized	Your credentials do not authorize you to access the requested info.
404	Not Found	The service you requested was not found on our server.
405	Method Not Allowed	You tried POSTing or PUTting to a service that can't accept data.
500	Server Error	There was an internal error, if it continues, please contact us.

PATIENTS API

5.1 Create a patient

This endpoint is used to create a patient in the Cagenix API.

Note: **POST** /api/v1/patients/create

5.1.1 Call

```
{  
  'first_name': '',  
  'last_name': '',  
  'address_one': '',  
  'address_two': '',  
  'city': '',  
  'state': '',  
  'zip_code': '',  
  'email': '',  
  'practice': '',  
  'practitioner': '',  
}
```

Property	Description	Type	Required
first_name	First Name	String	X
last_name	Last Name	String	X
address_one	First address line	String	X
address_two	Second address line	String	
city	City	String	X
state	State	String	X
zip_code	ZIP Code in 5-4 format	String	X
email	email address	String	X
practice	ID of the Practice	Integer	
practitioner	ID of the Practitioner	Integer	X

5.1.2 Response

```
{  
  'request': {  
    'first_name': '',  
    'last_name': '',  
  },  
}
```

```
    'address_one': '',
    'address_two': '',
    'city': '',
    'state': '',
    'zip_code': '',
    'email': '',
    'practice': '',
    'practitioner': '',
  },
  'response': {
    'patient_id': '',
  },
}
```

Property	Description	Type
request	Object containing the original request data	Object
response	Object containing the Patient ID	Object
patient_id	A unique ID for the Patient record	Integer

5.2 Retrieve Patient details

This endpoint is used to retrieve all the details of a Patient.

Note: GET /api/v1/patients/<ID>

5.2.1 Response

```
{
  'request': {
    'patient_id': '',
  },
  'response': {
    'active': '',
    'first_name': '',
    'last_name': '',
    'address_one': '',
    'address_two': '',
    'city': '',
    'state': '',
    'zip_code': '',
    'email': '',
    'choice_one': '',
    'choice_two': '',
    'choice_three': '',
    'practice': '',
    'practitioner': '',
  }
}
```

Property	Description	Type
active	Active Flag for Archiving	String
first_name	First Name	String
last_name	Last Name	String
address_one	First address line	String
address_two	Second address line	String
city	City	String
state	State	String
zip_code	ZIP Code in 5-4 format	String
email	email address	String
choice_one	The “best” choice	String
choice_two	The “better” choice	String
choice_three	The “good” choice	String
practice	ID of the Practice	Integer
practitioner	ID of the Practitioner	Integer

5.3 Update a patient

This endpoint is used to update a patient in the Cagenix API.

Note: PUT /api/v1/patients/<id>

5.3.1 Call

```
{
  'first_name': '',
  'last_name': '',
  'address_one': '',
  'address_two': '',
  'city': '',
  'state': '',
  'zip_code': '',
  'email': '',
  'active': '',
  'practice': '',
  'practitioner': ''
}
```

Property	Description	Type	Required
active	Active Flag for Archiving	Boolean	X
patient_id	Patient ID	String	
first_name	First Name	String	
last_name	Last Name	String	
address_one	First address line	String	
address_two	Second address line	String	
city	City	String	
state	State	String	
zip_code	ZIP Code in 5-4 format	String	
email	email address	String	
active	Active Flag for Archiving	Boolean	
practice	ID of the Practice	Integer	
practitioner	ID of the Practitioner	Integer	

5.3.2 Response

```
{
  'request': {
    'patient_id': '',
    'first_name': '',
    'last_name': '',
    'address_one': '',
    'address_two': '',
    'city': '',
    'state': '',
    'zip_code': '',
    'email': '',
    'active': '',
    'practice': '',
    'practitioner': '',
  },
  'response': {
    'patient_id': '',
    'active': '',
  },
}
```

Property	Description	Type
request	Object containing the original request data	Object
response	Object containing the Patient ID	Object
patient_id	A unique ID for the Patient record	Integer
active	Active Flag for Archiving	Boolean

5.4 Delete a patient

This endpoint is used to Delete a patient in the Cagenix API.

Note: **DELETE** /api/v1/patients/<id>

5.4.1 Response

```
{
  'request': {
    'patient_id': '',
  },
  'response': {
    'status': '',
  },
}
```

Property	Description	Type
request	Object containing the original request data	Object
response	Object containing the Patient ID	Object
status	The result of the DELETE operation (EX: Success, Failed)	String

PRACTITIONERS API

6.1 Create a Practitioner

This endpoint is used to create a practitioner in the Cagenix API.

Note: **POST** /api/v1/practitioners/create

6.1.1 Call

```
{
  'first_name': '',
  'last_name': '',
  'address_one': '',
  'address_two': '',
  'city': '',
  'state': '',
  'zip_code': '',
  'email': '',
  'activation_code': '',
  'practice': ''
}
```

Property	Description	Type	Required
first_name	First Name	String	X
last_name	Last Name	String	X
address_one	First address line	String	X
address_two	Second address line	String	
city	City	String	X
state	State	String	X
zip_code	ZIP Code in 5-4 format	String	X
email	email address	String	X
activation_code	Code used for initial account setup	String	X
practice	ID of the Practice	Integer	

6.1.2 Response

```
{
  'request': {
    'first_name': '',
    'last_name': ''
  }
}
```

```
    'address_one': '',
    'address_two': '',
    'city': '',
    'state': '',
    'zip_code': '',
    'email': '',
    'activation_code': '',
    'practice': '',
  },
  'response': {
    'practitioner_id': '',
  },
}
```

Property	Description	Type
request	Object containing the original request data	Object
response	Object containing the Practitioner ID	Object
practitioner_id	A unique ID for the Partitioner record	Integer

6.2 Retrieve Practitioner details

This endpoint is used to retrieve most details of a Practitioner.

Note: GET /api/v1/practitioners/<ID>

6.2.1 Response

```
{
  'request': {
    'practitioner_id': '',
  },
  'response': {
    'active': '',
    'first_name': '',
    'last_name': '',
    'address_one': '',
    'address_two': '',
    'city': '',
    'state': '',
    'zip_code': '',
    'email': '',
    'activation_code': '',
    'practice': '',
  }
}
```


Property	Description	Type
active	Active Flag for Archiving	String
first_name	First Name	String
last_name	Last Name	String
address_one	First address line	String
address_two	Second address line	String
city	City	String
state	State	String
zip_code	ZIP Code in 5-4 format	String
email	email address	String
activation_code	Code used for initial account setup	String
practice	ID of the Practice	Integer

6.3 Update a Practitioner

This endpoint is used to update a practitioner in the Cagenix API.

Note: PUT /api/v1/practitioners/<id>

6.3.1 Call

```
{
  'first_name': '',
  'last_name': '',
  'address_one': '',
  'address_two': '',
  'city': '',
  'state': '',
  'zip_code': '',
  'email': '',
  'active': '',
  'practice': ''
}
```

Property	Description	Type	Required
active	Active Flag for Archiving	Boolean	
first_name	First Name	String	
last_name	Last Name	String	
address_one	First address line	String	
address_two	Second address line	String	
city	City	String	
state	State	String	
zip_code	ZIP Code in 5-4 format	String	
email	email address	String	
active	Active Flag for Archiving	Boolean	
practice	ID of the Practice	Integer	

6.3.2 Response

```
{
  'request': {
```

```
    'first_name': '',
    'last_name': '',
    'address_one': '',
    'address_two': '',
    'city': '',
    'state': '',
    'zip_code': '',
    'email': '',
    'active': '',
    'practice': '',
  },
  'response': {
    'practitioner_id': '',
    'active': '',
  },
}
```

Property	Description	Type
request	Object containing the original request data	Object
response	Object containing the Practitioner ID	Object
practitioner_id	A unique ID for the Partitioner record	Integer
active	Active Flag for Archiving	Boolean

6.4 Delete a Practitioner

This endpoint is used to Delete a practitioner in the Cagenix API.

Note: **DELETE** /api/v1/practitioners/<id>

6.4.1 Response

```
{
  'request': {
    'practitioner_id': '',
  },
  'response': {
    'status': '',
  },
}
```

Property	Description	Type
request	Object containing the original request data	Object
response	Object containing the Practitioner ID	Object
status	The result of the DELETE operation (EX: Success, Failed)	String

6.5 Authenticate a Practitioner

This endpoint is used to get the secret key for a practitioner in the Cagenix API.

Note: **POST** /api/v1/practitioners/login

6.5.1 Call

```
{
  'email': '',
  'hash': '',
}
```

Property	Description	Type	Required
email	email address	String	X
hash	Base64 encoded HMAC+SHA512 of the password signed with the salt encrypted with pbkdf2_sha512	String	X

Note: hash = pbkdf2_sha512(base64(sha512(utf8(SALT) + utf8(PASSWORD))))

6.5.2 Response

```
{
  'request': {
    'email': '',
    'hash': '',
  },
  'response': {
    'secret': '',
  },
}
```

Property	Description	Type
request	Object containing the original request data	Object
response	Object containing the secret for API access	Object
secret	The secret for API Access	String

EVALUATIONS API

7.1 Create an Evaluation

This endpoint is used to create an evaluation in the Cagenix API.

Note: **POST** /api/v1/evaluations/create

7.1.1 Call

```
{
  'practice': '',
  'practitioner': '',
  'patient': '',
  'answers': [
    {
      'question_id': '',
      'patient_answer': '',
    }, #...
  ]
}
```

Property	Description	Type	Required
practice	ID of the Practice	Integer	
practitioner	ID of the Practitioner	Integer	
patient	ID of the Patient	Integer	X
answers	A collection of Answers objects	List	X
question_id	ID of the Evaluation Question	Integer	X
patient_answer	Patient's response to the question	String	X

7.1.2 Response

```
{
  'request': {
    'practice': '',
    'practitioner': '',
    'patient': '',
    'answers': [
      {
        'question_id': '',
        'patient_answer': '',
      },
    ],
  },
}
```

```
        }, #...
    ]
},
'response': {
    'evaluation_uuid': '',
},
}
```

Property	Description	Type
request	Object containing the original request data	Object
response	Object containing the Eval ID	Object
evaluation_uuid	A unique UUID for the evaluation records	String

7.2 Retrieve Evaluation details

This endpoint is used to retrieve all the details of an Evaluation.

Note: GET /api/v1/evaluations/<UUID>

7.2.1 Response

```
{
  'request': {
    'evaluation_uuid': '',
  },
  'response': {
    'evaluation_uuid': '',
    'practice': '',
    'practitioner': '',
    'patient': '',
    'answers': [
      {
        'question_id': '',
        'patient_answer': '',
      }, #...
    ]
    'active': '',
  }
}
```

Property	Description	Type
evaluation_uuid	A unique UUID for the evaluation records	String
practice	ID of the Practice	Integer
practitioner	ID of the Practitioner	Integer
patient	ID of the Patient	Integer
answers	A collection of Answers objects	List
question_id	ID of the Evaluation Question	Integer
patient_answer	Patient's response to the question	String

7.3 Update an Evaluation

This endpoint is used to update an evaluation in the Cagenix API. You must resubmit all answer objects again. So if an evaluation had 20 questions, all of those answers must be resubmitted for an update.

Note: **PUT** /api/v1/evaluations/<uuid>

7.3.1 Call

```
{
  'evaluation_uuid': '',
  'practice': '',
  'practitioner': '',
  'patient': '',
  'answers': [
    {
      'question_id': '',
      'patient_answer': '',
    }, #...
  ]
  'active': '',
}
```

Property	Description	Type	Required
evaluation_uuid	A unique UUID for the evaluation records	String	X
practice	ID of the Practice	Integer	
practitioner	ID of the Practitioner	Integer	
patient	ID of the Patient	Integer	
answers	A collection of Answers objects	List	X
question_id	ID of the Evaluation Question	Integer	X
patient_answer	Patient's response to the question	String	X

7.3.2 Response

```
{
  'request': {
    'evaluation_uuid': '',
    'practice': '',
    'practitioner': '',
    'patient': '',
    'answers': [
      {
        'question_id': '',
        'patient_answer': '',
      }, #...
    ]
    'active': '',
  },
  'response': {
    'evaluation_uuid': '',
    'status': '',
    'active': '',
  },
}
```

Property	Description	Type
request	Object containing the original request data	Object
response	Object containing the Patient ID	Object
evaluation_uuid	A unique UUID for the evaluation records	String
status	The result of the PUT operation (EX: Success, Failed)	String
active	Active Flag for Archiving	Boolean

7.4 Delete a patient

This endpoint is used to delete an evaluation in the Cagenix API.

Note: **DELETE** /api/v1/evaluations/<uuid>

7.4.1 Response

```
{
  'request': {
    'evaluation_uuid': '',
  },
  'response': {
    'status': '',
  },
}
```

Property	Description	Type
request	Object containing the original request data	Object
response	Object containing the Evaluation UUID	Object
status	The result of the DELETE operation (EX: Success, Failed)	String

USERS BLUEPRINT OVERVIEW

The following document covers the internals of the Cagenix-Web Users Blueprint.

- *Users Models*
- *users-forms-label*
- *users-views-label*

8.1 Users Models

`cagenix.users.models`

This file is used to define all the models used by the Users blueprint

class `cagenix.users.models.Role` (*name, description=None*)

The Role model defines the authorization roles used by Flask-Security

This Role class defines the model attributes, and several classmethods to make using the class easier.

Parameters

- **db.Model** (*object*) – The SQLAlchemy base database model
- **RoleMixin** (*object*) – The RoleMixin base provided by Flask-Security

classmethod `by_id` (*role_id*)

`by_id` returns a Role object given a `role_id`

Parameters

- **cls** (*class*) – The calling class
- **role_id** (*int*) – The id of the role being searched for

classmethod `by_name` (*role_name*)

`by_name` returns a Role object given a `role_name`

Parameters

- **cls** (*class*) – The calling class
- **role_name** (*string*) – The name of the role being searched for

classmethod `get_roles` ()

`get_roles` returns a list of Role objects

This is primarily used by the WTForms to display the QueryMultipleSelect

Parameters **cls** (*class*) – The calling class

class `cagenix.users.models.User` (***kwargs*)

The User model defines the authentication users used by Flask-Security

This Role class defines the model attributes, a classmethod to make finding the users easier. It also contains several fields unexposed by the UI. Those fields are used to track user logins and access IPs.

Parameters

- **db.Model** (*object*) – The SQLAlchemy base database model
- **UserMixin** (*object*) – The UserMixin base provided by Flask-Security

classmethod `by_email` (*email*)

`by_email` returns a User object given an email address.

Parameters

- **cls** (*class*) – The calling class
- **email** (*string*) – The email of the user being searched for

classmethod `by_id` (*user_id*)

`by_id` returns a User object given a `user_id`

Parameters

- **cls** (*class*) – The calling class
- **user_id** (*int*) – The id of the user being searched for

classmethod `by_phone` (*phone*)

`by_phone` returns a User object given a phone number

Parameters

- **cls** (*class*) – The calling class
- **phone** (*string*) – The phone number of the user being searched for

classmethod `by_username` (*username*)

`by_username` returns a User object given a username

Parameters

- **cls** (*class*) – The calling class
- **username** (*string*) – The username of the user being searched for

PATIENTS BLUEPRINT OVERVIEW

The following document covers the internals of the Cagenix-Web Patients Blueprint.

- *Patients Models*
- *patients-forms-label*
- *patients-views-label*

9.1 Patients Models

cagenix.users.models

This file is used to define all the models used by the Users blueprint

class cagenix.patients.models.**Patient** (*first_name=None, last_name=None*)

The User model defines the authentication users used by Flask-Security

This Role class defines the model attributes, a classmethod to make finding the users easier. It also contains several fields unexposed by the UI. Those fields are used to track user logins and access IPs.

Parameters

- **db.Model** (*object*) – The SQLAlchemy base database model
- **UserMixin** (*object*) – The UserMixin base provided by Flask-Security

classmethod **by_email** (*email*)

by_email returns a User object given an email address.

Parameters

- **cls** (*class*) – The calling class
- **email** (*string*) – The email of the user being searched for

classmethod **by_id** (*user_id*)

by_id returns a User object given a user_id

Parameters

- **cls** (*class*) – The calling class
- **user_id** (*int*) – The id of the user being searched for

classmethod **by_phone** (*phone*)

by_phone returns a User object given a phone number

Parameters

- **cls** (*class*) – The calling class
- **phone** (*string*) – The phone number of the user being searched for

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

C

`cagenix`, [5](#)
`cagenix.patients.models`, [31](#)
`cagenix.users.models`, [29](#)

B

`by_email()` (cagenix.patients.models.Patient class method), 31
`by_email()` (cagenix.users.models.User class method), 30
`by_id()` (cagenix.patients.models.Patient class method), 31
`by_id()` (cagenix.users.models.Role class method), 29
`by_id()` (cagenix.users.models.User class method), 30
`by_name()` (cagenix.users.models.Role class method), 29
`by_phone()` (cagenix.patients.models.Patient class method), 31
`by_phone()` (cagenix.users.models.User class method), 30
`by_username()` (cagenix.users.models.User class method), 30

C

`cagenix` (module), 5
`cagenix.patients.models` (module), 31
`cagenix.users.models` (module), 29
`create_user()` (in module cagenix), 5

G

`get_roles()` (cagenix.users.models.Role class method), 29

H

`handle_splash()` (in module cagenix), 5

L

`logout()` (in module cagenix), 5

N

`not_found()` (in module cagenix), 5

P

`Patient` (class in cagenix.patients.models), 31

R

`Role` (class in cagenix.users.models), 29

U

`User` (class in cagenix.users.models), 29