



Eagle Eye Networks

API利用ガイド

ver:1.1

2015年11月17日
2015年11月18日更新

株式会社イグアズ
クラウド&ソリューション事業部 クラウド推進部



更新履歴

日付	バージョン	更新内容
2015/11/17	初版	初版リリース
2015/11/8	1.1	Eagle Eye Networks APIの使い方例を追加
同上	同上	サンプルスクリプト解説を追加

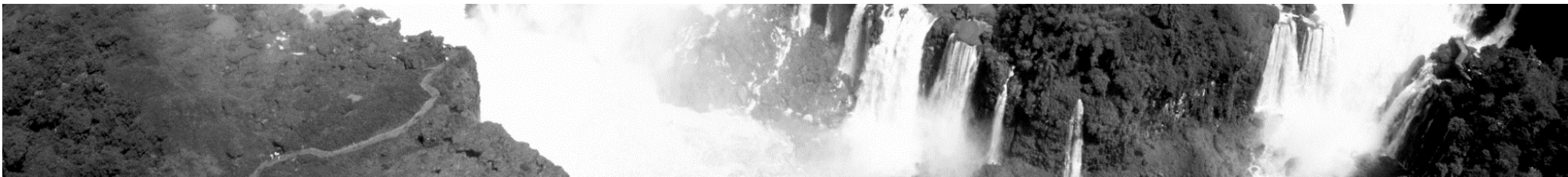
Agenda



- ▶ Eagle Eye Networks APIとは
- ▶ Eagle Eye Networks APIの用途
- ▶ Eagle Eye Networks APIの使い方例
- ▶ API概要（詳細は右記）
- ▶ サンプルスクリプト解説

▶ API概要（詳細）

- 認証、認可
- ユーザー
- カメラ
- ブリッジ
- 画像と動画
- ポーリング
- レイアウト
- AAA
- 利用規約表示
- アカウント
- アクション
- アノテーション（注釈）
- フィードバック
- メトリック
- PNGスパン
- 録画
- 検索
- エラー



Eagle Eye Networks APIとは

Eagle Eye Networks(以下EEN) APIは、EENの管理コンソールの基盤だけではなく、自動化や自社アプリケーションへの組み込みなど、様々な用途に利用できます。

EEN APIの特徴

- 簡単・・・ APIはREST原理に基づいているので、作成やアプリケーションのテストが非常に簡単
- 便利・・・ EEN APIは録画の管理や任意のカメラのライブビュー、ビデオ再生を時と場所を選ばずに行えるよう、アプリケーションに組み込めます。
- 安全・・・ EEN APIの実行にはユーザー認証が必要で、認証から実行、データの取得まで全てHTTPSで実行されるため、通信の全てが暗号化されます。

EEN APIの使い方

- ▶ ユーザーID、パスワードを使って認証、認可
 - セッション情報からアクセス・トークンを取得（ここまではセッション中1度だけ実行）
 - 実行したい内容を送信
 - JSON形式（または実体ファイル）で結果を獲得

Eagle Eye Networks APIの用途

EEN APIは、アプリケーションの組み込みだけでなく、cURLを使った簡単な自動化のお手伝いも可能です。

EEN APIを使った自動化例

フォレンジックとしてのバックアップ・・・

クラウド保存期間（Retention）を超えて法的証拠として保存が必要な動画などを、APIで自動的に保存することが可能です。

パスワード変更・・・

パスワード変更を自動的に行い、例えばゲストが接続するアカウントを一月に一回パスワードの変更が必要な場合などに利用可能です。

状態変化の監視・・・

EENではデフォルトではシステム監視の仕組みがありません。そのため状態監視の仕組みとして Long Polling を利用して監視を行うことが可能です。

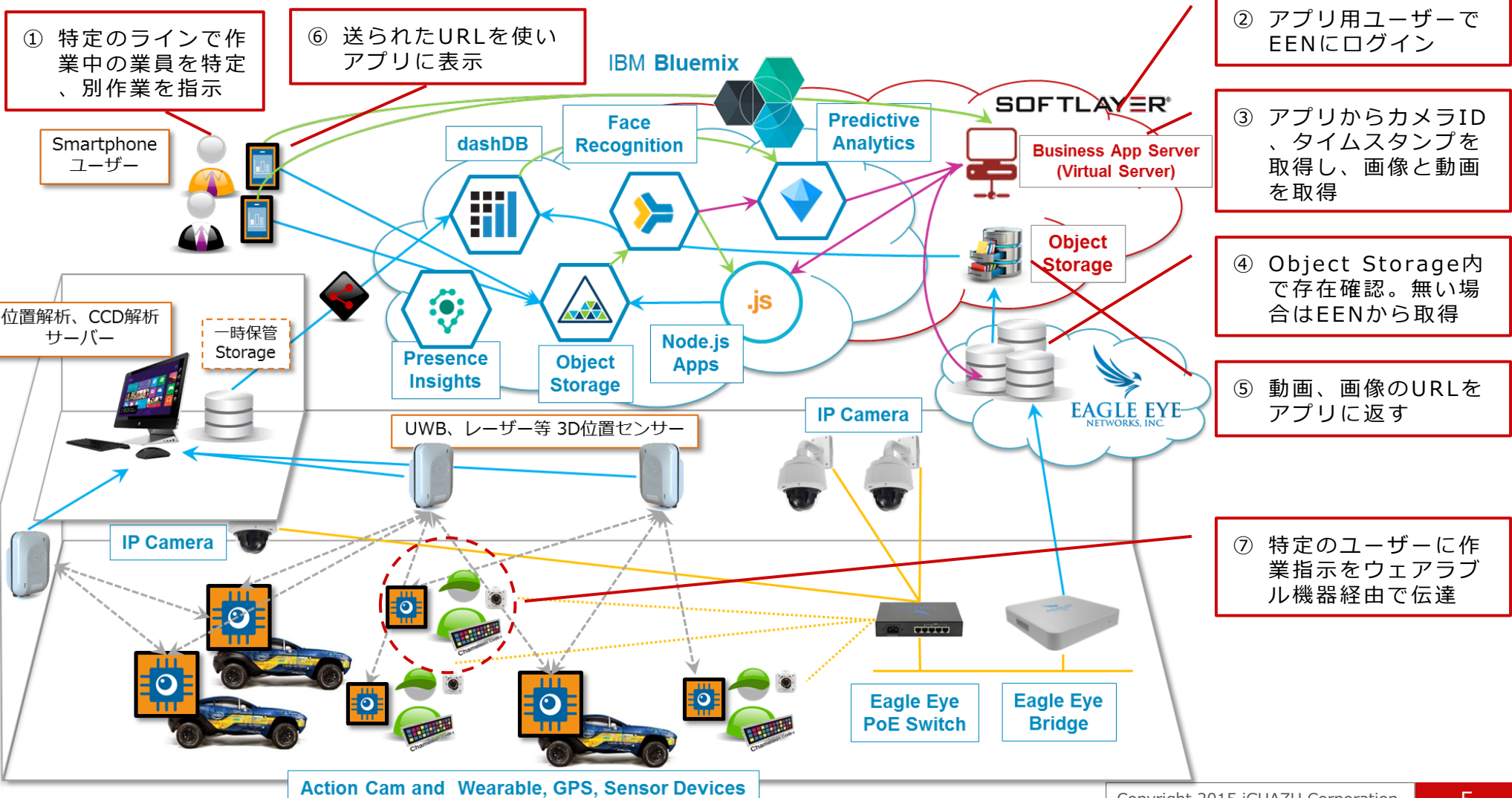
使用状況の自動収集・・・

EENのブリッジ及びカメラにはメトリックと呼ばれる利用状況の確認機能があるため、その内容を自動収集し、データベースなどに格納することが可能です。

本書で記載しているEEN APIの実行例は基底部分のみのため、詳しい引数、使い方につきましては、<https://apidocs.eagleeyenetworks.com/apidocs/> を御覧ください。日本語版は <http://een-api-doc-jp.mybluemix.net/> にあります（日本語版は株式会社イグアズが独自に翻訳したものです。正式/最新版は英語版サイトをご参照ください）。

Eagle Eye Networks APIの使い方例

EEN APIと他システムを組み合わせることで撮影された動画、画像を簡単に既存システムに組み込むことが可能です。



API概要 【認証、認可】

EEN APIにアクセスするには2ステージのプロセスが必要です：クライアントは初めに使い捨ての認証トークンを取得するために証明書とレلمムを示します。この使い捨てトークンは発行されて30秒間または使用されるまで有効です。一度認証トークンを取得した場合は、リソースへのアクセスを行うためのセッションID ("auth-key"クッキーを通じて)を認可サービスを呼び出して利用しなければなりません。この2つのフェーズの手法は、複数のドメインで認証と操作を行うことを提供します。この最初のステップは認証することで完了します。次のステップは認可することで完了します。認証コールはHTTPS接続を使用する必要があることに注意してください。

Step 1: 認証

ログインは2ステップのプロセスがあります：認証と、認証時に返されたトークンを使用した認可です。

HTTPリクエスト

```
POST https://login.eagleeyenetworks.com/g/aaa/authenticate
```

JSON応答

```
{ "token": "O3k0hNH3jQgjaxC0bLG9/5cM+Z7eWdPe0c+U ... R/p7ainHVhEc/bSFYUCvzipcAa==" }
```

Step 2: 認可

認証後はユーザーオブジェクトの認可を行い、'auth_key'クッキーに設定します。

HTTPリクエスト

```
POST https://login.eagleeyenetworks.com/g/aaa/authorize
```

応答 (Cookieの内容：一部)

```
#HttpOnly_.eagleeyenetworks.com TRUE / FALSE 1446664179 auth_key c006~22 ... 6ca801
```

API概要 【ユーザー】

EEN APIでユーザー情報の作成から取得、更新、削除などが実行できます。

ユーザー情報の取得

IDが与えられるとユーザー オブジェクトを返します。ID無しに場合は、実行したユーザー情報が返されます。

HTTPリクエスト GET <https://login.eagleeyenetworks.com/g/user>

JSON応答 { "id": "ca0e1cf2", "first_name": null, ... "is_terms_noncompliant": 1 }

ユーザーの作成

新規ユーザーを作成します。

HTTPリクエスト PUT <https://login.eagleeyenetworks.com/g/user>

JSON応答 { "id": "ca0ffa8c" }

ユーザーの更新

ユーザー情報を更新します。

HTTPリクエスト POST <https://login.eagleeyenetworks.com/g/user>

JSON応答 { "id": "ca0ffa8c" }

ユーザーの削除

ユーザーを削除します。

HTTPリクエスト DELETE <https://login.eagleeyenetworks.com/g/user>

API概要 【カメラ】

EEN APIでカメラ情報の取得、追加、削除、リストの取得が実行できます。

カメラ情報の取得

ブリッジIDを指定することで、ブリッジの情報が返されます。

HTTPリクエスト GET <https://login.eagleeyenetworks.com/g/device>

JSON応答 { "cameras": null, ... "permissions": "swr" }

ブリッジにカメラを追加

新規カメラをブリッジに追加します。

HTTPリクエスト PUT <https://login.eagleeyenetworks.com/g/device>

JSON応答 { "id": "100c339a" }

カメラの削除

カメラをブリッジから削除します。

HTTPリクエスト DELETE <https://login.eagleeyenetworks.com/g/device>

カメラ リストの取得

カメラのリストを配列で取得します。

HTTPリクエスト GET <https://login.eagleeyenetworks.com/g/device/list>

JSON応答 [["00004206", "100d88a8", ...]]

API概要 【ブリッジ】

EEN APIでブリッジ情報の取得、追加、削除、リストの取得が実行できます。

ブリッジ情報の取得

ブリッジIDを指定することで、ブリッジの情報が返されます。

HTTPリクエスト GET <https://login.eagleeyenetworks.com/g/device>

JSON応答 { "bridges": null, ... "permissions": "swr" }

ブリッジの追加

新規ブリッジをEEVB(Eagle Eye Video Bank)に追加します。

HTTPリクエスト PUT <https://login.eagleeyenetworks.com/g/device>

JSON応答 { "id": "100c339a" }

ブリッジの削除

ブリッジをEEVBから削除します。

HTTPリクエスト DELETE <https://login.eagleeyenetworks.com/g/device>

ブリッジ リストの取得

ブリッジのリストを配列で取得します。

HTTPリクエスト GET <https://login.eagleeyenetworks.com/g/device/list>

JSON応答 [["00004206", "100d88a8", ...]]

API概要 【画像と動画 <画像>】

EEN APIで画像情報の取得、要求、リストの取得が実行できます。

画像の取得

カメラIDとEENタイムスタンプを指定することで、画像が返されます。

HTTPリクエスト GET <https://login.eagleeyenetworks.com/asset/asset/image.jpeg> <指定時刻>
GET <https://login.eagleeyenetworks.com/asset/prev/image.jpeg> <指定時刻の前>
GET <https://login.eagleeyenetworks.com/asset/after/image.jpeg> <指定時刻の次>

EENタイムスタンプは、YYYYMMDDHHMMSS.mmm の形式で指定します。APIはこのタイムスタンプに合致しない画像は404 (Not Found)を返しますので必ずリスト取得、確認後に取得します。

画像の要求

クラウドにアップロードされていない画像をブリッジに要求します（要webhook URL）。

HTTPリクエスト GET <https://login.eagleeyenetworks.com/asset/cloud/image.jpg>
JSON応答 { "event": "[EVENT]" }

画像リストの取得

画像のリストを配列で取得します。

HTTPリクエスト GET <https://login.eagleeyenetworks.com/asset/list/image>
JSON応答 [{ "t": "PRFR", "s": "20141001000000.045" } ...]

API概要 【画像と動画 <動画>】

EEN APIで動画情報の取得、要求、リストの取得が実行できます。

動画の取得

カメラIDとEENタイムスタンプ（開始、終了）を指定することで、動画が返されます。

HTTPリクエスト

```
GET https://login.eagleeyenetworks.com/asset/play/video.{video_format}
```

EENタイムスタンプは、YYYYMMDDHHMMSS.mmm の形式で指定します。APIはこの開始、終了タイムスタンプに合致しない動画は404 (Not Found)を返しますので必ずリスト取得、確認後に取得します。

動画の要求

クラウドにアップロードされていない動画をブリッジに要求します（要webhook URL）。

HTTPリクエスト GET https://login.eagleeyenetworks.com/asset/cloud/video.flv

JSON応答 { "event": "[EVENT]" }

動画リストの取得

動画のリストを配列で取得します。

HTTPリクエスト GET https://login.eagleeyenetworks.com/asset/list/video

JSON応答 [{ "s": "20141001000016.768", "e": "20141001000100.758" } ...]

API概要 【ポーリング】

ポーリングサービスはEENからイベントまたは時刻、期間の通知をアプリケーションで取得する仕組みを提供します。

エンティティのリソース グループ

- thumb - サムネイル リソース。サムネイル画像のタイムスタンプを提供します。
- pre - プレビュー リソース。プレビュー画像のタイムスタンプを提供します。
- video - ビデオ リソース。ビデオ イベントの開始及び終了タイムスタンプを提供します。
- event - イベント リソース。完全なイベント情報を提供します。
イベントのソースは「デバイスまたはカメラ」「システム イベント」「アカウント イベント」から提供されます。
- status - ブリッジまたはカメラの状態をビットマスク フラグで提供します。

ポーリングの初期化

ポーリングサービスのサブスクライブを行います。ポーリングはステートフルなサービスのため初期化の時点で指定したリソースについてのみ後から取得が可能です。

HTTPリクエスト POST <https://login.eagleeyenetworks.com/poll>

JSON応答 [["00004206", "100d88a8", ...]]

ポーリング情報の取得

ブリッジのリストを配列で取得します。

HTTPリクエスト GET <https://login.eagleeyenetworks.com/poll>

JSON応答 [["00004206", "100d88a8", ...]]

API概要 【レイアウト】

EEN APIでレイアウト情報の取得、追加、削除、リストの取得が実行できます。

レイアウト情報の取得

レイアウトIDを指定することで、レイアウトの情報が返されます。

HTTPリクエスト GET <https://login.eagleeyenetworks.com/g/layout>

JSON応答 { "id": "0b58ec7a-61e4-11e3-8f7d-523445989f37", ... "types":["mobile"] }

レイアウトの作成

新規レイアウトを作成、追加します。

HTTPリクエスト PUT <https://login.eagleeyenetworks.com/g/layout>

JSON応答 { "id": "0b58ec7a-61e4-11e3-8f7d-523445989f37" }

レイアウトの削除

レイアウトを削除します。

HTTPリクエスト DELETE <https://login.eagleeyenetworks.com/g/device>

レイアウト リストの取得

ブリッジのリストを配列で取得します。

HTTPリクエスト GET <https://login.eagleeyenetworks.com/g/device/list>

JSON応答 [["00004206", "100d88a8", ...]]

API概要 【 AAA (認証、認可、アカウントティング) #1】

EEN APIでアカウント関連の様々な操作が実行できます。

アカウントの作成

新規アカウントとアカウントに紐づくスーパー ユーザーを作成します。

HTTPリクエスト POST https://login.eagleeyenetworks.com/g/aaa/create_account

アカウントの発行認証

アカウント作成時に発行されるeメールに記載されたトークンで発行確認を行います。

HTTPリクエスト POST https://login.eagleeyenetworks.com/g/aaa/validate_account

JSON応答 { "user_id": "ca103fea" }

パスワードリセットの依頼

パスワードリセット依頼時のeメールを発行します。

HTTPリクエスト POST https://login.eagleeyenetworks.com/g/aaa/forgot_password

パスワードリセットの確認トークン

パスワードリセット依頼時に発行されるeメールに記載されたトークンで削除を行います。

HTTPリクエスト POST https://login.eagleeyenetworks.com/g/aaa/check_pw_reset_token

API概要 【 AAA (認証、認可、アカウントティング) #2】

EEN APIでアカウント関連の様々な操作が実行できます。

パスワードのリセット

新規パスワードと共にリセットを実行します。

HTTPリクエスト POST https://login.eagleeyenetworks.com/g/aaa/reset_password

JSON応答 { "user_id": "ca103fea" }

アカウントの発行認証メールの再送

アカウント作成時に発行されるeメールの再送を行います。

HTTPリクエスト

POST https://login.eagleeyenetworks.com/g/aaa/resend_user_verification_email

パスワードの変更

現在のユーザーは自身の、スーパーユーザーは全てのユーザーのパスワードを変更できます。

HTTPリクエスト POST https://login.eagleeyenetworks.com/g/aaa/change_password

JSON応答 { "id": "ca02c000" }

アカウントの切り替え

マスターアカウントからサブアカウントへの切り替えを含め、アカウントを切り替えます。

HTTPリクエスト POST https://login.eagleeyenetworks.com/g/aaa/switch_account

API概要 【 AAA (認証、認可、アカウントティング) #3】

EEN APIでアカウント関連の様々な操作が実行できます。

シングルサインオン(SSO)

SSOはリセラーが提供する認証システムをプロキシとして、SSO認証後にEENへの認証を要求することが可能です。SSOはスタンダードSAML (Security Assertion Markup Language) で構成された認証プロバイダ内で `brand_saml_publickey_ret` 及び `brand_saml_namedid_path` のプロパティと共に構成されている必要があります。

- `brand_saml_publickey_cert` はEENで認証可能なx509証明書を含みます。
- `brand_saml_namedid_path` はユーザーのログインに使用するeメールアドレスです。

HTTPリクエスト `POST https://login.eagleeyenetworks.com/g/sso`

JSON応答 `{ "user_id": "ca103fea" }`

ログアウト

ログアウトし、HTTPセッション クッキーを破棄します。

HTTPリクエスト `POST https://login.eagleeyenetworks.com/g/aaa/logout`

API概要 【利用規約表示 #1】

EEN APIで利用規約の取得、受諾、承認、リストの取得が実行できます。

利用規約はサブアカウント用に独自に作成可能であり、ユーザー用は確認、受諾のみが可能です。

ユーザー用利用規約の取得

ユーザーの利用規約の受諾状況を出力します。

HTTPリクエスト GET <https://login.eagleeyenetworks.com/g/user/terms>

JSON応答 `[['00009436', 'UNIT_TEST_SUB_ACCOUNT', ... 'active']]`

ユーザー用利用規約の受諾

アカウント作成時に発行されるeメールに記載されたトークンで発行確認を行います。

HTTPリクエスト PUT <https://login.eagleeyenetworks.com/g/user/terms>

JSON応答 `{ 'id': 'cafe81f5' }`

アカウント用利用規約の作成

アカウント用の利用規約を作成します。

HTTPリクエスト PUT <https://login.eagleeyenetworks.com/g/account/terms>

JSON応答 `{ 'status': 'active', ... 'account_id': '00009074' }`

API概要 【利用規約表示 #2】

EEN APIで利用規約の取得、受諾、承認、リストの取得が実行できます。

アカウント用利用規約の更新

アカウント用の利用規約の内容を更新します。

HTTPリクエスト POST <https://login.eagleeyenetworks.com/g/account/terms>

JSON応答 { 'status': 'active', ... 'cafebead', 'account_id': '00009074' }

アカウント用利用規約の削除

アカウント用の利用規約を削除します。

HTTPリクエスト DELETE <https://login.eagleeyenetworks.com/g/account/terms>

JSON応答 { 'cafe81f5': { ...'Test_Terms_of_Service': { '2': 26193626.502' } } }

アカウント用利用規約の取得

利用規約IDを指定することで、アカウント用の利用規約を返します。

HTTPリクエスト GET <https://login.eagleeyenetworks.com/g/account/terms>

JSON応答 [['00009436', 'UNIT_TEST_SUB_ACCOUNT', ... 'active']]

API概要 【アカウント】

EEN APIでアカウント情報の作成から取得、更新、削除などが実行できます。

アカウント情報の取得

IDが与えられるとアカウント情報を返します。

HTTPリクエスト GET <https://login.eagleeyenetworks.com/g/account>

JSON応答 {}

アカウントの作成

新規アカウントを作成します。

HTTPリクエスト PUT <https://login.eagleeyenetworks.com/g/account>

JSON応答 { "id": "1234abcd" }

アカウントの削除

アカウントを削除します。

HTTPリクエスト DELETE <https://login.eagleeyenetworks.com/g/account>

アカウントリストの取得

アカウントのリストを配列で取得します。

HTTPリクエスト GET <https://login.eagleeyenetworks.com/g/account/list>

JSON応答 [["00004206", "Greater Good", ... [...], [...]]]

API概要 【アクション】

EEN APIで接続された一つまたは全てのデバイスに対し、大まかなアクションを動作させることが可能です。

録画オン

指定したカメラやレイアウトに含まれるカメラに対し、録画状態をオンに設定します。

HTTPリクエスト POST <https://login.eagleeyenetworks.com/g/action/recordnow>

JSON応答 {}

録画オフ

指定したカメラやレイアウトに含まれるカメラに対し、録画状態をオフに設定します。

HTTPリクエスト POST <https://login.eagleeyenetworks.com/g/action/recordoff>

JSON応答 {}

全てのカメラで録画オン

全てのカメラに対し、録画状態をオンに設定します。アカウント スーパーユーザーのみ実行できます。

HTTPリクエスト POST <https://login.eagleeyenetworks.com/g/action/allon>

JSON応答 {}

全てのカメラで録画オフ

全てのカメラに対し、録画状態をオフに設定します。アカウント スーパーユーザーのみ実行できます。

HTTPリクエスト POST <https://login.eagleeyenetworks.com/g/action/alloff>

JSON応答 {}

API概要 【アノテーション（注釈）】

EEN APIでカメラや動画に対してアノテーションを付加することが可能です。

アノテーションを付加するにはデバイスとタイムスタンプに紐付ける必要があります。アノテーションもその他の動画、画像と同様に保存期間（リテンション）で管理されます。

注釈の作成

カメラとタイムスタンプを指定し、アノテーションを付加します。データはJSON形式である必要があります。

HTTPリクエスト PUT <https://login.eagleeyenetworks.com/g/annotation>

JSON応答 {}

注釈の更新

カメラとタイムスタンプで指定されたアノテーションを更新します。

HTTPリクエスト POST <https://login.eagleeyenetworks.com/g/annotation>

JSON応答 {}

API概要 【フィードバック】

EEN APIでユーザーからサポートへのフィードバックを行えます。

これは主にサポートされないカメラを接続した際に管理コンソールに表示される、メールアイコンの動作と同様のフィードバックを行う場合などに利用されます。

フィードバックの送信

サポートにフィードバックを送信します。

HTTPリクエスト POST <https://login.eagleeyenetworks.com/g/feedback>

JSON応答 {}

API概要 【メトリック】

EEN APIでカメラやブリッジのメトリック（帯域測定データ）の取得が可能です。
メトリックについての詳しい内容は、オンラインヘルプを御覧ください。

カメラの帯域

特定のカメラの帯域データを取得します。

HTTPリクエスト GET <https://login.eagleeyenetworks.com/g/metric/camerabandwidth>

JSON応答 { "motion": [], "core": [["20151109080000.000", ... 0.0, 0.0]] }

ブリッジの帯域

特定のブリッジの帯域データを取得します。

HTTPリクエスト GET <https://login.eagleeyenetworks.com/g/metric/bridgebandwidth>

JSON応答 { "core": [["20141002170000.000", 711610368.0, ... 1279678]] }

API概要 【 PNGスパン】

EEN APIでメトリックのPNG画面を生成可能です。

PNGスパンは発生した録画やイベントなどをPNGファイルに描画することが可能です。現在サポートされている取得方法は以下のとおりです。

▶ settings

table=onoff との組み合わせにより、カメラのオン/オフ設定の切り替え状況を表示します。

▶ purge

fflags=LOSTとの組み合わせにより、クラウド保存期間（リテンション）による古い動画の破棄状況を表示します。

▶ span

table=video との組み合わせにより、動画の録画状況を表示します。fflags=STREAMにより、動画のストリームのみ表示。

table=motion との組み合わせにより、全動体検知の状況を表示します。fflags=ALERTSにより、アラーム時のみを表示。

table=roim との組み合わせにより、特定の動体検知の状況を表示します。fflags=ALERTSにより、アラーム時のみを表示。

table=stream との組み合わせにより、カメラのストリーミング状況を表示します。

table=register との組み合わせにより、EENから認識されているカメラ数を表示します。

PNGスパンの取得

PNGスパンの取得。

HTTPリクエスト GET https://login.eagleeyenetworks.com/pngspan/{png_type}.png

JSON応答 {}

API概要 【録画】

EEN APIで録画情報の取得や更新が可能です。

この録画オブジェクトはアクション - 録画オン、アクション - 録画オフによる開始、停止によって作成されるものを指します。

録画オブジェクトの取得

録画キーを指定して録画オブジェクトを取得します。

HTTPリクエスト GET <https://login.eagleeyenetworks.com/g/recording>

JSON応答 {}

録画オブジェクトの更新

録画オブジェクトを更新します。

HTTPリクエスト POST <https://login.eagleeyenetworks.com/g/recording>

JSON応答 {}

API概要 【検索】

EEN APIで様々な形式のデータを横断的に検索が可能です。

現時点では録画情報とアノテーションのみ検索が可能です。その他の形式の検索は近い将来に実装される予定です。

録画情報の検索

検索内容に合致する録画オブジェクトを返します。

HTTPリクエスト GET <https://login.eagleeyenetworks.com/g/search/recordings>

JSON応答 {}

アノテーションの検索

検索内容に合致するアノテーション オブジェクトを返します。

HTTPリクエスト GET <https://login.eagleeyenetworks.com/g/search/recordings>

JSON応答 {}

API概要 【エラーコード】

EEN APIで返されるエラーコードは以下のとおりです。

エラーコード	内容
200	リクエストは正常に実行されました
202	リクエストの実行は受理されましたが、処理は完了していません
400	リクエストの文法または内容が不完全です
401	提供された認証情報が不正です
402	アカウントは停止されています
403	必要な権限が足りないため、処理が拒否されました
406	レルムがルートレルムでは無いため無効です
409	与えられたEメールアドレスは指定されたレルムで登録済みです
412	ユーザーは無効化されています
460	アカウントは非アクティブです
461	アカウントは保留中です
462	ユーザーは保留中です

サンプルスクリプト解説 【動画取得】

▶ 動画取得スクリプト編

動画取得の流れ

```
./get_video_v4.sh "開始 YYYY/MM/DD HH:MM:SS" "終了 YYYY/MM/DD HH:MM:SS" カメラ名またはall 保存先
```

EENの動画取得は、以下の流れで行われます。

1. 認証、認可
2. カメラIDの取得
3. 動画リストの取得
4. 動画の取得

動画を取得するには、

- ▶ cRULスクリプトが実行可能な環境（Linux推奨、Windowsの場合はGit for Windows等の導入が必要）
- ▶ jq（Linux、Windows共に）
- ▶ アクセス権のあるユーザー、パスワード
- ▶ 録画したカメラ名（または全てのカメラ）
- ▶ 動画の完全な開始、終了時間（あいまい指定はできません）
- ▶ 動画を保存するのに必要な空きスペース

が必要になります。

サンプルスクリプト解説 【動画取得】

▶ 処理全体スクリプト #1

EENの動画取得は、以下の流れで行われます。

```
#!/bin/bash

if [ $# -lt 3 ]; then
  echo "$# argument entered." 1>&2
  echo 'Please specify argument Start time "YYYY/MM/DD HH:MM:SS", End time
"YYYY/MM/DD HH:MM:SS", Camera Name(or all) and Saved to Directory(option).' 1>&2
  exit 1
fi

#変数定義
. ./set_env_v4.sh

# ログイントークンの取得
. ./get_auth_v4.sh $user $password

# カメラIDの取得
. ./get_cameraid_v4.sh $3
```

引数のチェック。最低3つの引数がない場合には終了します。

【変数定義スクリプト】をご確認ください

【認証認可スクリプト】をご確認ください

【カメラID取得スクリプト】をご確認ください

サンプルスクリプト解説 【動画取得】

▶ 処理全体スクリプト #2

(続き)

```
#時刻補正 (JSTとUTCの両内容を保持)
```

```
ST_JST=`date -d "$1" "+%Y%m%d%H%M%S"`
```

```
ST_UTC=`date -d "$1 9 hours ago" "+%Y%m%d%H%M%S"`
```

```
ET_JST=`date -d "$2" "+%Y%m%d%H%M%S"`
```

```
ET_UTC=`date -d "$2 9 hours ago" "+%Y%m%d%H%M%S"`
```

```
# 動画リストの取得 (カメラごとに取得)
```

```
while IFS=, read C1 C2
```

```
do
```

```
    $curl -s -G
```

```
"https://login.eagleeyenetworks.com/asset/list/video?start_timestamp=${ST_UTC}.000;  
end_timestamp=${ET_UTC}.000;id=$C1;A=$auth_key;options=coalesce" | ./jq -r '[] |  
"¥(.id),¥(.s),¥(.e)" | tr -d '¥r' > video_¥C2.txt
```

```
done < camera_id.txt
```

EENは動画、画像などのオブジェクトに対するタイムスタンプをUTCで格納しているため、UTCとJSTの2パターンで取得できるよう変数定義します

動画リストは曖昧な時間（2015年11月1日 0時から11月2日 0時までなどの指定が可能です。この範囲にある動画がリストとして出力されます

サンプルスクリプト解説 【動画取得】

▶ 処理全体スクリプト #3

(続き)

```
# ビデオの取得
while IFS=, read C1 C2
do
  if [ ! -d ./$4/${C2} ]
  then
    mkdir -p ./$4/${C2}
  fi
  while IFS=, read ID ST ET
  do
    starr=(`echo $ST | tr -s '!'`)
    starrfld=(`echo ${starr[0]} | fold -s2`)
    STJ=`date -d "${starrfld[0]}${starrfld[1]}${starrfld[2]}${starrfld[3]}
${starrfld[4]}:${starrfld[5]}:${starrfld[6]} 9 hours" "+%Y%m%d%H%M%S"`.${starr[1]}
    echo -e "Download file: ${C2}_${STJ}.mp4\tLocation: ./$4/${C2}"
    $curl -# -G
    "https://login.eagleeyenetworks.com/asset/play/video.mp4?id=${C1};start_timestamp=${ST};e
nd_timestamp=${ET};A=${auth_key}" -o ./$4/${C2}/${C2}_${STJ}.mp4 -w
"%nSize: %{size_download} Bytes\tDownload time: %{time_total} sec%n"
  done < video_${C2}.txt
done < camera_id.txt
```

指定された、または全てのカメラの指定された期間内の動画リストに従い、動画をダウンロードします。動画は指定された（指定されていない場合は現在の）ディレクトリに保存され、カメラIDごとにディレクトリを作成します。

通常EENの動画はUTC時間で管理されているため、ダウンロード指示はUTCで行い、ダウンロードしたファイルにはJSTで命名します。

サンプルスクリプト解説 【動画取得】

▶ 変数定義スクリプト

EENのAPIを使用するにあたり、必要な定義をこのスクリプトに記載します。

```
#!/bin/bash
```

```
#変数定義
```

```
user=<ユーザー名>
```

```
password=<パスワード>
```

```
pxaddrport=<プロキシサーバーIP:ポート番号>
```

```
echo "Using user name $user".
```

```
echo "Using password is given(supressed)."
```

```
echo "Using Proxy Server and Port $pxaddrport."
```

```
#Proxy判別
```

```
curl --connect-timeout 2 -sI https://login.eagleeyenetworks.com
```

```
if [ "$?" == "0" ]
```

```
then
```

```
    curl="curl"
```

```
    echo "HEAD Command to EEN Server was successful."
```

```
else
```

```
    curl="curl -x $pxaddrport"
```

```
    echo "HEAD Command to EEN Server was fauilure. Use Proxy settings."
```

```
fi
```

実行する処理によっては通常ユーザーでは権限が無いものがあります。適宜アカウントスーパーユーザーやスーパーユーザーを指定してください

このスクリプトではプロキシを自動判別していますが、実行遅延が発生しますのでプロキシが無い環境の場合は無効にしてください

サンプルスクリプト解説 【動画取得】

▶ 認証認可スクリプト #1

EENのAPIでの認証認可を行うためのスクリプトです。

```
#!/bin/bash
```

```
# ログイントークンの取得
```

```
token=`$curl -s --request POST https://login.eagleeyenetworks.com/g/aaa/authenticate --data "username=$1&password=$2" --insecure | ./jq -r .token`
```

```
if [ $? == 0 ]
```

```
then
```

```
    echo "Authentication was successful"
```

```
else
```

```
    echo "Authentication was failure. Error code is $? ."
```

```
    exit 1
```

```
fi
```

```
# アクセスキーの取得 (Cookieの取得)
```

```
$curl -s -D - -c cookie.txt --request POST https://login.eagleeyenetworks.com/g/aaa/authorize --data-urlencode token=$token > null
```

ログイントークンを取得します。このトークンはタイムアウト時間が短いため、直ぐに認可を実行する必要があります

認可を行いアクセスキーを取得します。キーはクッキー内のauth_keyという部分に含まれます

サンプルスクリプト解説 【動画取得】

▶ 認証認可スクリプト #2

(続き)

```
if [ $? == 0 ]
then
    echo "Authorization was successful"
else
    echo "Authorization was failure. Error code is $? ."
    exit 1
fi

# Auth keyの取得 (Cookieから)
auth_key=`grep auth_key cookie.txt | awk '{print $7}`
echo "Using auth_key is $auth_key"
```

取得したauth_keyを変数
auth_keyに代入します

サンプルスクリプト解説 【動画取得】

▶ カメラID取得スクリプト #1

EENのAPIでログインしたユーザーで取得できるカメラIDを取得します。

```
#!/bin/bash
```

```
# カメラIDの取得
```

```
$curl -s --cookie "auth_key=$auth_key" --request GET  
https://login.eagleeyenetworks.com/g/device/list?t=camera | ./jq -r '.[1] | select(.[1] !=null)  
| .[1:3] | @csv' | sed -e 's//g' > camera_id.txt
```

```
if [ $? == 0 ]
```

```
then
```

```
    echo "Retrieve Camera ID was successful"
```

```
else
```

```
    echo "Retrieve Camera ID was failure. Error code is $? ."
```

```
    exit 1
```

```
fi
```

一度全ての取得可能なカメラIDを取得します。

サンプルスクリプト解説 【動画取得】

▶ カメラID取得スクリプト #2

(続き)

```
if [ "$1" != "all" ]
then
    mv camera_id.txt camera_id2.txt
    grep $1 camera_id2.txt > camera_id.txt
    rm camera_id2.txt
fi

echo -e "%nCamera ID%tCamera Name"
cat camera_id.txt | sed 's/,/ /g'
echo -e "%n"
```

実行時の引数3がallだった場合はそのまま全てのカメラIDを記載し、単体のカメラを指定している場合にはそのカメラIDのみを取り込みます

EOF

iGUAZU

