EINS

API Endpoints Overview

20th May 2020

EINS
By VIA

Introduction

This document specifies all endpoints that are available on the EINS API, detailing each request and response, as well as informing over some of the endpoint standards that the project uses.

Encryption Considerations

It must be noted that the API uses end to end encryption on all of the request and response bodies. This means that all the formatting of the request and response bodies are the same. The payload structure would be a simple base64 string that is broken down as follows:

```
<16 byte IV><Cipher Text><32 byte MAC>
```

- **IV**: This is the initialisation vector. It is a 16 byte, cryptographically randomised array that is required for decryption. It must be different for each payload.
- **Cipher Text**: This is the actual plain text data that has undergone encryption.
- **MAC**: This is a 32-byte array that is used to verify the integrity of the payload and ensure that it has not been altered due to errors in communication or malicious attacks. The MAC is calculated over *IV + Cipher Text*.

That being said, for the purposes of this document, encryption is not considered and the plain text information is detailed instead.

Endpoint Standards

The project's endpoints conform to certain standards to ensure ease of use. These standards can be broken down into 3 main sections:

- **URL Route standards**
- **Request standards**
- **Response standards**

EINS
By VIA

Route Standards

All endpoints follow a specific route format that has been guided by best practices. The standard format for all endpoints is:

<domain>/api/*<version>*/*<subDirectory>*

Where:

- Domain: The domain information (eg. https://einsapp.eu)
- Version: The endpoint version formatted as *v1*
- SubDirectory: Contains the unique, route-specific information (eg. statistics/covid)

Request Standards

API request bodies are accepted in JSON format only whilst requests are authenticated using JSON Web Tokens as described below in more detail.

Authentication Standards

All endpoints, except for authentication, require a JSON Web Token (JWT) before any information will be processed or divulged. This token is sent using the standard Bearer token format in an HTTP authorisation header.

```
"Authorization: Bearer <access token>"
```

The access token is returned in the authentication response if successful. Note that in order to simplify the documentation this will not be included in each endpoint, however, it must be included for all calls unless stated otherwise.

Response Standards

In order to ease API usage, all API responses follow a standard JSON format. This format is specified as below

EINS
By VIA

```
{
    "data": null,
    "meta": {
        "success": true,
        "code": 200,
        "message": null
    }
}
```

- **Data:** Contains all endpoint specific response data.
- **Meta:** Additional metadata on the API call. Indicates the success of the call and any reasons for a call failure.

Endpoints

In this section all of the endpoints on the API are detailed.

Device Authentication (*POST /api/v1/devices*)

The device authentication call is the only call that does not require a JWT in order to access it. This is as it is used to authenticate both new and already registered devices to the system. The JWT required for access to all other endpoints is returned in this call.

Registration

When registering a new device certain additional pieces of information are required so that they can be stored alongside the device in the database. This leads the request body to be structured slightly differently to standard authentication.

Utilisation

This call is used only once per device. This occurs after the initial download of the app once 'Get Started' has been selected on the below screen:

**Privacy driven
Covid-19 tracing**

Are you 16 years of age or older?

☐ Yes   ☐ No

By clicking "Get started" you agree to the
Terms of Use. Our Privacy Policy applies.

✏ Get Started →

Legal Notice

Request

```
{
        "AppId": <string>,
        "publicKey": <string>,
        "operatingSystem": <string>,
        "pushToken": <string>,
        "language": <string>,
        "seed": <string>,
        "preMasterSecret": <string>,
        "Signature":{
                "plainTextData": <string>,
                "signedData": <string>
        }
}
```

- **AppId**: A required field containing a unique identifier for the application. This identifier is random and contains no device or user-specific information.
- **PublicKey**: Contains a base64 encoded public key. This key must correspond to the key pair used to generate the digital signature.

EINS
By VIA

- **OperatingSystem**: The operating system of the mobile device. Must be IOS or Android.
- **PushToken**: A unique identifier that allows the server to push notifications to the app. This identifier contains no information about the device or user. This is optional.
- **Language**: This indicates the user's preferred language for the mobile application. All text will be displayed to the user in this language.
- **Seed**: A cryptographically secure random 32-byte string in base64 format.
- **PreMasterSecret**: A cryptographically secure 48-byte array that has been encrypted with the server public key. This is in base64 format.
- **Signature**: Digital signature performed using the client key pair. This is verified against the public key that is also contained in the payload.

Successful Response

```
{
        "data": {
                "accessToken": <string>,
                "accessTokenExpiry": <string>,
                "seed": <string>,
                "Signature": {
                        "plainTextData": <string>,
                        "signedData": <string>
                },
                "meta": {
                        "success": <bool>,
                        "code": <int>,
                        "message": <string>
                }
        }
}
```

- **AccessToken**: JWT used to authenticate all API calls for the session.
- **AccessTokenExpiry**: The invariant culture DateTime indicating when the accessToken will no longer be valid.
- **Signature**: Contains the server's digital signature. This is verified by the mobile app using an embedded public key in the mobile application source code.
- **Seed**: A cryptographically secure random 32-byte string in base64 format generated by the server.
- **Meta**: Additional metadata on the API call. Indicates the success of the call and any reasons for a call failure.

EINS
By VIA

Possible Errors

The following errors may be received with this call:

- 401 Unauthorised
  - Invalid public key
  - Invalid pre-master-secret
  - AppId already exists in database
  - Invalid signature
  - Signature verification fails (does not match provided public key)
- 400 Bad Request
  - Invalid AppId
  - Missing digital signature
  - Invalid push token
  - Invalid seed
  - Invalid alpha2 (language)

Authentication

When re-authenticating in order to generate a new session, the same endpoint is called as with registering a new device, however, there are some fields that must not be included with this call that would be included with the registering of a new app.

Utilisation

This call is made every time a new session needs to be generated. This is most often upon the opening of the application (not for the first time) but can also take place in the background during use of the mobile application should the current token be expiring.

EINS
By VIA

Request

```
{
        "appId": <string>,
        "language": <string>,
        "seed": <string>,
        "preMasterSecret": <string>,
        "signature":{
                "plainTextData": <string>,
                "signedData": <string>
        }
}
```

- **AppId**: A required field containing a unique identifier for the application. This identifier is random and contains no device or user-specific information.
- **Language**: This indicates the user's preferred language for the mobile application. All text will be displayed to the user in this language. Language is optional.
- **Seed**: A cryptographically secure random 32-byte string in base64 format.
- **PreMasterSecret**: A cryptographically secure 48-byte array that has been encrypted with the server public key. This is in base64 format.
- **Signature**: Digital signature performed using the client key pair. This is verified against the public key that the server has stored for the specified appId. Verifies that the device owns the key pair of the app it is trying to authenticate as.

EINS
By VIA

Successful Response

```
{
        "data": {
                "accessToken": <string>,
                "accessTokenExpiry": <string>,
                "seed": <string>,
                "signature": {
                            "plainTextData": <string>,
                            "signedData": <string>
                },
                "meta": {
                        "success": <bool>,
                        "code": <int>,
                        "message": <string>
                }
        }
}
```

- **AccessToken**: JWT used to authenticate all API calls for the session.
- **AccessTokenExpiry**: The DateTime indicating when the accessToken will no longer be valid.
- **Signature**: Contains the server's digital signature. This is verified by the mobile app using an embedded public key in the mobile application source code.
- **Seed**: A cryptographically secure random 32-byte string in base64 format generated by the server.
- **Meta**: Additional metadata on the API call. Indicates the success of the call and any reasons for a call failure.

Possible Errors

The following errors may be received with this call:

- 401 Unauthorised
    - Invalid pre-master-secret
    - Invalid signature structure
    - Signature verification failed
    - Push token, public key or operating system was included in the call
    - App is locked-out

EINS
By VIA

- 400 Bad Request
  - Invalid AppId
  - Missing digital signature
  - Invalid push token
  - Invalid seed
  - Invalid alpha2 (language)

Additional Notes

There are some additional things that must be noted with this call. These are as follows:

- An AppId can get locked-out
- Push token must be unique

Should a specific appId have 3 incorrect attempts, the app will be locked-out. This lockout lasts for 10 minutes until it is possible to authenticate again. Any attempted auth that takes place with an appId that is logged out, a 401 unauthorized response is returned.

In order to ensure proper and functioning push notifications, it is essential that all push tokens must be unique. It should not be possible to have more than one application having the same push token. Therefore this is blocked.

Delete Device (*DELETE /api/v1/devices*)

This call removes all device information from the database. This includes all data that has been submitted by the user as well as actual device data itself.

Utilisation

This endpoint is called from the mobile application when a user chooses to delete all their data. This can be found under the 'Delete My Data' heading on the menu and is called upon confirmation of the below screen.

EINS
By VIA
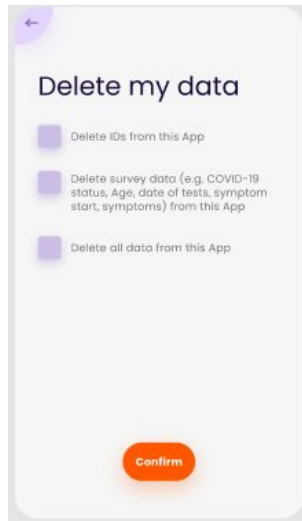
Request

There are no additional parameters or additional body for this call.

Successful Response

```
{
    "data": null,
    "meta": {
        "success": true,
        "code": 200,
        "message": null
    }
}
```

Possible Errors

The following errors may be received with this call:

- 401 Unauthorised
    - Missing or Invalid JWT

(De)Activate Device (*POST /api/v1/devices/active*)

This endpoint allows a user to deactivate a specific device. Once a device is deactivated all attempts to authenticate will be blocked. Currently, this is limited to deactivating only the device that is currently authenticated, however, the long term goal for this would be to enable device linking and enable a user to deactivate one of their devices from a second device.

Utilisation

Currently, this call is not utilised by the mobile application.

Request

Despite being a post request, there is no request body. There is simply a single query parameter that allows for specification on whether a device is being activated or deactivated.

Query Parameter	Accepted Values
status	true, false

Successful Response

```
{
    "data": null,
    "meta": {
        "success": true,
        "code": 200,
        "message": null
    }
}
```

.

Possible Errors

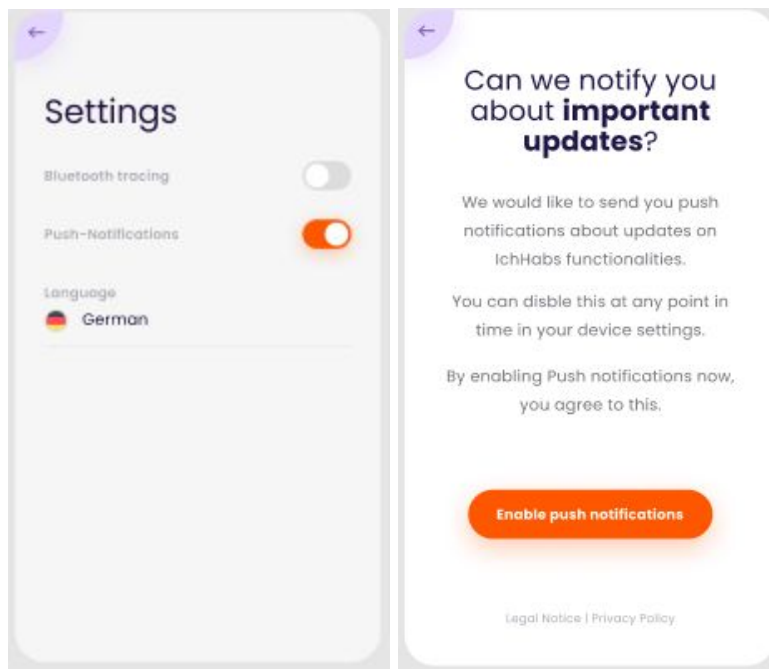The following errors may be received with this call:

EINS
By VIA

- 401 Unauthorised
    - Missing or Invalid JWT
- 400 Bad Request
    - Invalid status value

Update Push Token (*POST /api/v1/devices/push_token*)

This endpoint updates the device's push token. The device is found using the embedded Id in the JWT.

Utilisation

This call is made should a user decide to enable push notifications for the first time. This can either be done during the onboarding flow or later on through settings. See below for the screens where this happens:

Request

```
{
      "pushToken": <string>
}
```

Successful Response

```
{
  "data": null,
  "meta": {
    "success": true,
    "code": 201,
    "message": null
  }
}
```

Possible Errors

The following errors may be received with this call:

- 401 Unauthorised
  - Missing or Invalid JWT
  - Push token already exists in database
- 400 Bad Request
  - Token is null or an empty string
  - Token is less than 5 characters long
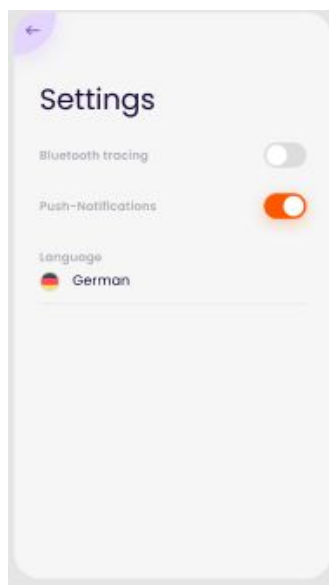 - Token is greater than 500 characters long

EINS
By VIA

Delete Push Token (*DELETE api/v1/devices/push_token*)

This call deletes a push token from the device that is specified through the auth token.

Utilisation

This is used in the settings menu when a user chooses to disable push notifications.

Request

No additional body or query parameters required.

Successful Response

```
{
  "data": null,
  "meta": {
    "success": true,
    "code": 200,
    "message": null
  }
}
```

Possible Errors

The following errors may be received with this call:

- 401 Unauthorised
  - Missing or Invalid JWT

Get Symptoms (*GET /api/v1/symptoms*)

This endpoint returns all symptoms currently listed on that database in a specified language. Should no language be specified then it will return in the system default language as specified in the app settings. Currently, this is English.

Utilisation

This call is used in order to display all the currently supported and active symptoms on the database to the user for them to select from during the survey completion as shown here:

Request

This is a GET request with the following query parameters:

Query Parameter	Accepted Values
language	Any valid alpha2 that is supported by the server.

If language is not supplied as a parameter then the system default language will be returned.

Response

```
{
  "data": [
    {
      "key": "question_positive_symptom-1",
      "value": "Sore throat"
    },
    ...
    {
      "key": "question_positive_symptom-12",
      "value": "Loss of smell"
    }
  ],
  "meta": {
    "success": true,
    "code": 200,
    "message": null
  }
}
```

- **Key**: The symptom key as stored in the language table. Each key corresponds to multiple versions of the same symptom in different languages. To be passed back to the API on further calls.
- **Value**: The symptom in the requested language. To be displayed to the user.

Possible Errors

The following errors may be received with this call:

- 401 Unauthorised
  - Missing or Invalid JWT
- 400 Bad Request
  - Invalid Alpha2

EINS
By VIA

Submit Survey (*POST /api/v1/submission/covid*)

This endpoint allows a user to 'donate' or submit their COVID-19 data to the server for further study by authorities or experts. It allows for the application to build up more information that can be used to fight the COVID-19 virus. It must be noted that this submission is completely voluntary and that no data is required to be submitted.

Utilisation

This call is used whenever a user chooses to submit their COVID-19 information shortly after filling in the survey information on the application as shown below:
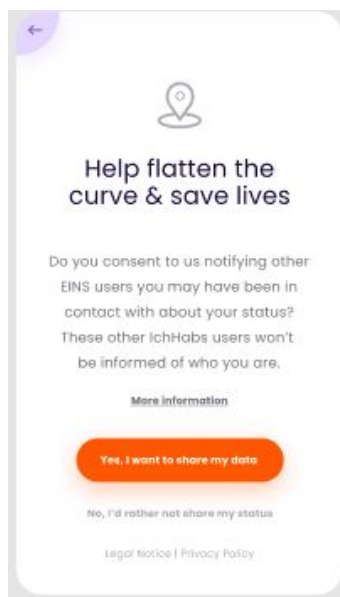
Request

It must be noted that the request structure differs slightly depending on the COVID-19 status that is being submitted. See below for all possible requests. For each request, all fields are optional in order to maintain flexibility over what a user wants to divulge to the application.

Positive Status

```
{
        "Status":"Positive",
        "PositiveTestDate":"2020-03-29 13:00:09.1359267",
        "Country":"GB",
        "Language":"EN",
        "Age":25,
        "IsSymptomatic":false,
        "Symptoms": [
                "question_positive_symptom-1",
                "question_positive_symptom-3"
                ]
        "SymptomsFrom":"2020-03-29 13:00:09.1359267"
}
```

Negative Status

```
{
        "Status":"Negative",
        "NegativeTestDate":"2020-03-30 13:00:09.1359267",
        "Country":"GB",
        "Language":"EN",
        "Age":25,
        "Symptoms": [
                "question_positive_symptom-1",
                "question_positive_symptom-3"
                ]
        "SymptomsFrom":"2020-03-29 13:00:09.1359267"
}
```

EINS
By VIA

I Don't Know Status

```
{
        "Status":"Unsure",
        "Country":"GB",
        "Language":"EN",
        "Age":25,
        "IsSymptomatic":false,
        "Symptoms": [
                "question_positive_symptom-1",
                "question_positive_symptom-3"
                ]
        "SymptomsFrom":"2020-03-29 13:00:09.1359267"
}
```

Recovered Status

```
{
        "Status":"Recovered",
        "PositiveTestDate":"2020-03-29 13:00:09.1359267",
        "NegativeTestDate":"2020-03-30 13:00:09.1359267",
        "Country":"GB",
        "Language":"EN",
        "Age":25,
        "IsSymptomatic":false
}
```

EINS
By VIA

Successful Response

```
{
  "data": null,
  "meta": {
    "success": true,
    "code": 201,
    "message": null
  }
}
```

Possible Errors

The following errors may be received with this call:

- 401 Unauthorised
    - Missing or Invalid JWT
- 400 Bad Request
    - Invalid message structure for specified status
    - Missing request payload
    - Unsupported country
    - Invalid Alpha2
    - Invalid age (younger than 0 or older than 200)
    - Invalid status
    - Invalid test dates (in the future or before the first recorded case of COVID-19)
    - Invalid symptom count (more than 20)
    - Invalid symptom (longer than 50 chars or does not match registered symptom on the database)
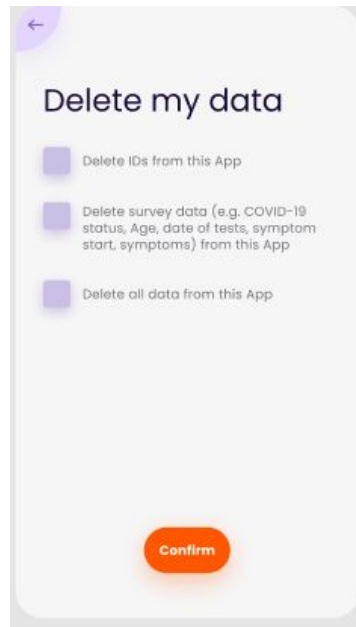   - Duplicate symptoms

Delete Submitted Information (*DELETE /api/v1/submission*)

This deletes all data that a user has submitted to the server according to the Id that is embedded within the JWT.

EINS
By VIA

Utilisation

This call is made through the 'Delete My Data' heading on the application menu. It is triggered when selecting the 'Delete survey data' option.

Request

There is no additional request body or query parameters for this call.

Response

```
{
    "data": null,
    "meta": {
        "success": true,
        "code": 200,
        "message": null
    }
}
```

Possible Errors

The following errors may be received with this call:

- 401 Unauthorised
  - Missing or Invalid JWT

Get COVID-19 Statistics (*GET /api/v1/statistics/covid/google*)

This call returns statistical information on current COVID-19 infection rates depending on the country specified. This particular call sources its data from Google Big Query.

Utilisation

This call is made on the main dashboard of the application. Every time the user opens the dashboard this call is made according to the country the device is registered to.

Request

The following query parameters can be included:

Query Parameter	Accepted Values
countries	List of any supported alpha2 values.

Note that this parameter is optional. Should no country be listed then the endpoint returns global statistics.

When passing in multiple parameters the URL must be structured as per the below example:

```
/api/v1/statistics/google?countries=ZA&countries=GB
```

Response

```
{
    "Data": {
        "CountryName": "South Africa",
        "ContractedCount": 16433,
        "DeathCount": 286,
        "RecoveredCount": 7298,
        "Source": "GoogleBigQuery",
        "SourceCreatedAt": "2020-05-17T00:00:00",
        "SourceUrl":
"https://console.cloud.google.com/marketplace/details/johnshopkins/covid19_jhu
_global_cases?filter=solution-type:dataset&q=covid&id=430e16bb-bd19-42dd-bb7
a-d38386a9edf5&_ga=2.196272622.-312723631.1586190188&pli=1",
        "CreatedAt": "2020-04-18T01:00:08.545734"
    },
    "Meta": {
        "Success": true,
        "Code": 200,
        "Message": null
    }
}
```

EINS
By VIA

Possible Errors

The following errors may be received with this call:

- 401 Unauthorised
    - Missing or Invalid JWT
- 400 Bad Request
    - Invalid alpha2 code

Get Countries (*GET /api/v1/countries*)

This endpoint returns information on the specified country should the country be currently supported by the project. Should no country be specified then it returns all countries supported by the database.

Utilisation

Currently, this is not utilised by the mobile application.

Request

The following query parameters can be included with the request:

Query Parameter	Accepted Values
country	Country alpha2 value

As specified above, the parameter is optional and should it not be specified, all supported countries are returned.

Response

```
{
    "data": [
        {
            "id": <int>
            "name": <string>,
            "alpha2": <string>,
            "timezone": <string>,
            "callingCode": <string>,
            "cultureInfo": <string>
        }
    ],
    "meta": {
        "success": true,
        "code": 200,
        "message": null
    }
}
```

Possible Errors

The following errors may be received with this call:

- 401 Unauthorised
    - Missing or Invalid JWT
- 400 Bad Request
    - Invalid alpha2 code

Get Languages (*GET /api/v1/languages*)

Similar to the get country call specified above, this returns information on a specified language should it be supported by the system. Should no language be specified, all supported languages are returned.

EINS
By VIA

Utilisation

Currently, this is not utilised by the mobile application.

Request

The following query parameters can be included with the request:

Query Parameter	Accepted Values
language	Language alpha2 value

As specified above, the parameter is optional and should it not be specified, all supported languages are returned.

Response

```
{
    "data": [
        {
            "id": <int>
            "name": <string>,
            "alpha2": <string>,
        }
    ],
    "meta": {
        "success": true,
        "code": 200,
        "message": null
    }
}
```

Possible Errors

The following errors may be received with this call:

EINS
By VIA

- 401 Unauthorised
  - Missing or Invalid JWT
- 400 Bad Request
  - Invalid alpha2 code