**CertPortal Data API Documentation**

**/data\_api/view\_bulk\_verification**

**Overview**

* This endpoint can be used to verify one or multiple certifications at once
* There are lots of tools, all explained below, which can be used to customize your results, giving you exactly what you need and nothing you don’t

**Authentication**

* All requests require the API KEY provided to your team by NHA
* Your API key identifies you to the system; therefore, anyone possessing the key will have unrestricted access to your data. Protect the key as you would passwords, financial credentials, or any other sensitive information
* Should your API key ever be compromised, contact NHA staff immediately for assistance

**Heads Up!**

* It is not possible to apply filters using form-encoded data in GET requests nor to mix query parameters and form-encoded data using GET. If your GET request includes form-encoded data, *it will be ignored*. This can be especially confusing with mixed requests that include the API key in the query parameters but filter criteria in a form-encoded body. Such requests will succeed but will not apply the filter criteria.
* For this reason, we recommend not mixing query parameters and body data; instead, specify your entire request, including the API key, in *one* of the three acceptable formats: JSON data in the GET/POST request body, form-encoded data in the body of a POST, or as query parameters in a GET request.

**Response Format**

* The API can return results in CSV and JSON formats
* Specify the desired response format using the optional format parameter
* The default is JSON
* Example with JSON:

 {"api\_key": "ABC123",

 "format": "json"}

* Example with CSV:

 {"api\_key": "ABC123",

 "format": "csv"}

**Download Response as a File**

* If making a request to the Data API from a browser, you may wish for the data to be packaged up as a downloaded file, rather than having the browser try to render the data into the view. You can specify that with the use of the optional download parameter:
* Example:

 {"api\_key": "ABC123",

 "download": true}

**Choosing Desired Fields**

* You can limit the results to only include the desired fields using the optional fields parameter
* The fields to choose from are: id, exam\_type, applicant\_id, first\_name, last\_name, effective\_date, expiration\_date, created, provisional, certification\_number, and status
* Example:

 {"api\_key": "ABC123",

 “fields”: [“user\_id”, “user\_name”, “certification\_number”]}

**Sorting Order**

* Sort order can be specified using the optional sort parameter
* Example:

 {"api\_key": "ABC123",

 “sort”: “user\_id” }

**Paginate Results**

* Use the optional start and limit parameters to request results one page at a time.
* limit specifies the size of the page while start indicates which record to start counting from (0-based).
* The start and limit parameters can also be used independently of one another
* For example, to retrieve the first 20 candidates, submit the following query to /data\_api/view\_bulk\_verification:

 {"api\_key": "ABC123",

 "start": 0,
 "limit": 20}

**Filter Expressions**

* The Data API allows queries to specify criteria to filter the result set to only the desired data. Each criterion takes the form of a simple expression over a column
* Filter expressions support the following equality/inequality comparisons: =, !=, < <=, > >=
* Example:

 {"api\_key": "ABC123",

 “filter\_expressions”: “created >= ‘2013-01-01'” }

* Filter expressions can require that a value falls within a certain range using between: (ex. "score between 60 and 70")
* String containment operators are case insensitive. Automatic casting is performed, allowing these operators to be used on numeric fields (ex. user\_name startswith 'none’, id endswith '0', email\_address does not contain 'gmail', comment contains 'excellent')
* Parameters can be compared to Null (ex. email\_address is None, address2 is NULL, birth\_date is not None
* Multiple filter expressions can be specified as a list
* If utilizing a filter expression list, the filter\_join parameter is required. Setting filter\_join to “all” means “must match all expressions” while “or” means “must match at least one expression.
* Example:

{“api\_key”: “ABC123”,

 "filter\_expression": ["created >= '2013-01-01'", "user\_name contains 'new'"],
 "filter\_join": "and”}