

WINR Data – API V1.3.2 Reference

API Basics

The WINR Data API is standard REST API, using JSON-encoded request bodies and responses and standard HTTP response codes.

API Authentication

The API uses an API Key in the request header to authenticate API usage authorisation. This API key will be provided separately. The API access key variable format is:

```
X-API-Key: gpXL*****H9xP
```

User Authentication

In addition to the API usage authentication, each client will be provided with a unique client code/password combination that must be passed as part of the JSON request body. For example:

```
{
  "user": "clientCode",
  "password": "clientPassword",
  "country_code": "AU",
  ... other method attributes ...
}
```

API Endpoint

The WINR Data API endpoint is:

```
https://api.winrdata.com/v1/{method}
```

As of V1.3, there are three methods supported, which are documented in subsequent sections of this document:

- /verify
- /match
- /recent

API Responses

200 OK

This HTTP response code indicates a successful request and the JSON response object will contain the method-specific data, which are defined in the *API Methods* section.

400 Bad Request

This HTTP response code indicate that the JSON request object had invalid data. The JSON response object will contain additional information in the following format:

```
{
  "status": "error",
  "message": "Detailed error message"
}
```

401 Bad Request

This HTTP response code indicate that the JSON request object had invalid user authentication data. The JSON response object will contain:

```
{
  "status": "error",
  "message": "Invalid User/Password"
}
```

403 Forbidden

This HTTP response code indicates that the X-API-Key header parameter has not been set or contains an invalid value. The JSON response object will contain:

```
{
  "message": "Forbidden"
}
```

500 System Error

Whilst this HTTP response code should never occur, it indicates that WINR Data API has encountered an internal processing error. The JSON response object will contain:

```
{
  "status": "error",
  "message": "A system error has occurred. WINR Data has been notified"
}
```

API Methods

Verify

Verify will use a set of consumer parameter values to query the WINR Data consumer data lake for matching consumers.

Request

```
POST https://api.winrdata.com/v1/verify
```

JSON Request Parameters

In addition to the User Authentication attributes described in the *API Basics* section, the Verify method has the following (string) attributes:

Attribute	Notes
country_code	ISO 3166 alpha-2 country code (ex: 'AU', 'NZ', etc.) *REQUIRED
source_id	A client-defined parameter to distinguish API calls from departments or sub-clients. Used for back-end reporting and billing.
email	Email address
phone	In local country format, e.g.: <ul style="list-style-type: none">• 0nnnnnnnnnn – AU/NZ/GB• 1nnnnnnnnnn – US/CA
first_name	Consumer's first name (ex: 'Susan')
last_name	Consumer's last name (ex: 'Smith')
name	Full name (ex: 'Susan Smith')
address1	Address line one (ex: '123 Main St'). Street type will be normalised to country-specific abbreviations, '123 Main Street' is equivalent to '123 Main St'
suburb	Suburb/town/city name
state	State/county name or code, e.g.: <ul style="list-style-type: none">• 'NSW' or 'New South Wales'• 'MI' or 'Michigan'• 'ON' or 'Ontario'• 'Essex'
postcode	Postcode/ZIP code
dob	Date of Birth in YYYY-MM-DD format

Under the current implementation, *at least two* of the search attributes are required (in addition to the 'country_code' attribute).

JSON Response

A successful response has the following (string) attributes:

Attribute	Notes
status	One of two values: <ul style="list-style-type: none">• <code>verified</code> – one or more consumer record(s) was found that match the search criteria• <code>not found</code> – not consumer record matched the search criteria
confidence_level	A 1-10 match confidence level. For this version of the Verify method, all searches are exact matches, so the value will always be '10'.
sources_count	A number (0-n) indicating the number of distinct data sources that matched the search criteria. For a status of 'not found', this value would be '0'.

Sample JSON Request and Response Objects

Below is a sample JSON request object:

```
{
  "user": "sample",
  "password": "abc123",
  "country_code": "AU",
  "email": "tech@winr.com.au",
  "first_name": "Scott",
  "phone": "0412345678"
}
```

And the sample JSON response:

```
{
  "status": "verified",
  "confidence_level": "10",
  "sources_count": "2"
}
```

Match

Match will use a set of consumer parameter values to query the WINR Data consumer data lake for matching consumers and return a granular data response indicating what fields were matched.

Request

```
POST https://api.winrdata.com/v1/match
```

JSON Request Parameters

In addition to the User Authentication attributes described in the *API Basics* section, the Match method has the following (string) attributes:

Attribute	Notes
country_code	ISO 3166 alpha-2 country code (ex: 'AU', 'GB', 'NZ', etc.) *REQUIRED
source_id	A client-defined parameter to distinguish API calls from departments or sub-clients. Used for back-end reporting and billing.
email	Email address
phone	In local country format, e.g.: <ul style="list-style-type: none">• 0nnnnnnnnn – AU/NZ/GB• 1nnnnnnnnnn – US/CA
first_name	Consumer's first name (ex: 'Susan') or initial (ex: 'S') *REQUIRED
last_name	Consumer's last name (ex: 'Smith') *REQUIRED
dob	Date of Birth in YYYY-MM-DD format
address1	Address line one (ex: '123 Main St'). Street type will be normalised to country-specific abbreviations, '123 Main Street' is equivalent to '123 Main St'
suburb	Suburb/town/city name
state	State/county name or code, e.g.: <ul style="list-style-type: none">• 'NSW' or 'New South Wales'• 'MI' or 'Michigan'• 'ON' or 'Ontario'• 'Essex'
postcode	Postcode/ZIP code
reference_id	A client reference ID that will be echoed back in the API response
fuzzy_match	'0' (default – disabled) or '1' (enabled)

Under the current implementation, *at least one* of the search attributes are required (in addition to the 'country_code', 'first_name' and 'last_name' attributes).

JSON Response

A successful response has the following (string) attributes:

Attribute	Notes
transaction_id	A unique reference number generated by the API call
reference_id	The caller reference ID from the API call request parameters
email phone first_name first_initial last_name dob address1 suburb state postcode	A ternary value for each of the supplied query parameters indicating the match strength: <ul style="list-style-type: none">• 'match' – the query value was directly matched• 'partial' – the query value was partially matched. An example of this could be a first name query value of 'Nigel' that is matched with a record having 'N' as a first name value in the database¹.• 'no_match' – there were no records that match the query value

¹ Note: Partial matches on the first name would only occur if there was a direct match with the surname and another strong identity parameter – such as DOB, email, phone, address1 or postcode. See "Match Logic".

Sample JSON Request and Response Objects

Below is a sample JSON request object:

```
{
  "user": "sample",
  "password": "abc123",
  "country_code": "AU",
  "email": "tech@winr.com.au",
  "first_name": "Scott",
  "last_name": "Lawrence",
  "phone": "0412345678",
  "reference_id": "xyz-42"
}
```

And the sample JSON response:

```
{
  "transaction_id": "dc0c25d-07e7-46ed-bc45-2ffd81130ed7",
  "reference_id": "xyz-42",
  "email": "no_match",
  "first_name": "partial",
  "first_initial": "match",
  "last_name": "match",
  "phone": "match"
}
```

Match Logic

As it is possible that there will be multiple records that fully match or partially match a given query, the response is an amalgamation of the records utilising some matching logic. As an illustration, consider the simple example below.

If the query parameters were:

```
{
  "user": "sample",
  "password": "abc123",
  "country_code": "GB",
  "first_name": "Scott",
  "last_name": "Lawrence",
  "dob": "1970-01-31",
  "address1": "20 Pembridge Cres",
  "suburb": "Notting Hill",
  "postcode": "W11 3DS"
}
```

And the following records existed in the WINR data lake:

#	First	Last	Address1	Suburb	Postcode	DOB
1	Scotty	Lawrence			W11 3DS	
2	S	Lawrence	20 Pembridge Cres	London		1970-01-31

The JSON response would be:

```
{
  "transaction_id": "dc0c25d-07e7-46ed-bc45-2ffd81130ed7",
  "first_name": "partial",
  "first_initial": "match",
  "last_name": "match",
  "dob": "match",
  "address1": "match",
  "suburb": "no_match",
  "postcode": "match"
}
```

The logic behind this response is:

Field	Response	Comment
<code>first_name</code>	<code>partial</code>	Matched record #1 first initial plus last name and postcode; Matched record #2 first initial plus last name, address1 and postcode – but no exact match
<code>first_initial</code>	<code>match</code>	Matched record #1 first initial plus last name and postcode; Matched record #2 first initial plus last name, address1 and postcode
<code>last_name</code>	<code>match</code>	Matched record #1 last name plus first initial and postcode; Matched record #2 last name plus first initial, address1 and DOB
<code>dob</code>	<code>match</code>	Matched record #2 DOB plus first initial and last name
<code>address1</code>	<code>match</code>	Matched record #2 address1 plus first initial, last name and DOB. Note: if this record <i>did not</i> have a match on the DOB, the response would be 'no_match' as match on a partial first name, exact last name and exact address1 is not exact enough to assure this is the correct person (it could be Susan Lawrence at 20 Pembridge Crescent in Leeds)
<code>suburb</code>	<code>no_match</code>	Whilst record #2 matches first initial, last name, address1 and DOB – it is not an exact match of the suburb/city
<code>postcode</code>	<code>match</code>	Matched record #1 postcode plus first initial and last name

Fuzzy Matching

If the 'fuzzy_match' parameter is set to '1', and no exact match is found for a field, the API will use fuzzy matching to attempt to match several of the fields using the Levenshtein Distance between the query fields and the WINR Data records. The logic varies slightly for each field:

- **Address1:** If no exact match is found for Address1 and *at least one* of Postcode, Suburb, DOB or Phone is an exact match, a fuzzy match will be performed on Address1
- **Suburb:** If no exact match is found for Suburb and there is an exact match for Postcode, a fuzzy match will be performed on the Suburb field
- **Email/Phone/State:** If no exact match is found, a fuzzy match will be performed

When fuzzy matching is used for a field, if the Levenshtein Distance ratio is greater than 80, the field will be reported as a 'partial' match, and the ratio value will be returned in the JSON response (this allows a client to determine the strength of a match – for example, some clients might want a ratio of 90 or more to consider the field a match).

Fuzzy Matching Example:

If the query parameters were:

```
{
  "user": "sample",
  "password": "abc123",
  "country_code": "GB",
  "first_name": "Scott",
  "last_name": "Lawrence",
  "dob": "1970-01-31",
  "address1": "20 Pumbridge Cres",
  "suburb": "Lomdon",
  "postcode": "W11 3DS"
}
```

And the following records existed in the WINR data lake:

#	First	Last	Address1	Suburb	Postcode	DOB
1	Scott	Lawrence			W11 3DS	
2	S	Lawrence	20 Pembridge Cres	London		1970-01-31

The JSON response would be:

```
{
  "transaction_id": "dc0c25d-07e7-46ed-bc45-2ffd81130ed7",
  "first_name": "match",
  "first_initial": "match",
  "last_name": "match",
  "dob": "match",
  "address1": "partial",
  "suburb": "partial",
  "postcode": "match",
  "partial_match_scores": {
    "address1": 91,
    "suburb": 83,
  }
}
```

Recent

Recent will match a single consumer data value – either email address or phone and return the details for the most recent consumer record that is matched.

Request

```
POST https://api.winrdata.com/v1/recent
```

JSON Request Parameters

In addition to the User Authentication attributes described in the *API Basics* section, the Recent method has the following (string) attributes:

Attribute	Notes
country_code	ISO 3166 alpha-2 country code (ex: 'AU', 'GB', 'NZ', etc.) *REQUIRED
source_id	A client-defined parameter to distinguish API calls from departments or sub-clients. Used for back-end reporting and billing.
email	Email address
phone	In local country format, e.g.: <ul style="list-style-type: none">• 0nnnnnnnnn – AU/NZ/GB• 1nnnnnnnnn – US/CA

Under the current implementation, *either* 'email' or 'phone' is required (in addition to the 'country_code' attribute).

JSON Response

A successful response has the following (string) attributes:

Attribute	Notes
email	The value of the email request parameter (if provided)
phone	The value of the phone request parameter (if provided)
status	Indicates whether or not the query matched a consumer record. Possible values are 'success' or 'not found'
consumer	A JSON object containing the details of the most recent consumer record

Consumer Object in JSON Response

If the query successfully matched a consumer, the consumer JSON object returned will be structured as follows:

Attribute	Notes
recency	The date of most recent record in YYYY-MM-DD format
title	The consumer title (e.g. 'Mrs', 'Miss', 'Ms', 'Mr', 'Snr', 'Sra', etc.)
first_name	Consumer's first name (ex: 'Susan')
last_name	Consumer's last name (ex: 'Smith')
gender	Possible values are 'F', 'M' or 'B' (used when gender is derived and first name can be either male or female)
dob	Date of Birth in YYYY-MM-DD format
email	Email address
phone	In local country format, e.g.: <ul style="list-style-type: none">• 0nnnnnnnnnn – AU/NZ/GB• 1nnnnnnnnnn – US/CA
address1	Address line one (ex: '123 Main St')
Address2	Address line two (ex: 'Unit 42')
suburb	Suburb/town/city name
state	State/county name or code, e.g.: <ul style="list-style-type: none">• 'NSW' (New South Wales in Australia)• 'MI' (Michigan in the United States of America)• 'ON' (Ontario in Canada)• 'Essex'
postcode	Postcode/ZIP code
country_code	ISO 3166 alpha-2 country code (ex: 'AU', 'GB', 'NZ', etc.)
ip_address	The consumer's IP address

Sample JSON Request and Response Objects

Below is a sample JSON request object:

```
{
  "user": "sample",
  "password": "abc123",
  "country_code": "AU",
  "email": "scott@sample.com.au"
}
```

And the sample JSON response:

```
{
  "email": "scott@sample.com.au",
  "status": "success",
  "consumer": {
    "recency": "2016-01-18",
    "title": "",
    "first_name": "Steve",
    "last_name": "Lawrence",
    "gender": "M",
    "dob": "1970-01-30",
    "email": "scott@sample.com.au",
    "phone": "0412080681",
    "address1": "680 George St",
    "address2": "Suite 42",
    "suburb": "Sydney",
    "state": "NSW",
    "postcode": "2000",
    "country_code": "AU",
    "ip_address": "1.144.97.85"
  }
}
```

Document Revision History

Version	Date	Revision Details
1.0.0	07-01-2022	Original document
1.1.0	14-01-2022	Revised parameter names, added Match method
1.2.0	20-01-2022	Added <code>reference_id</code> parameter to Match method
1.3.0	24-01-2022	Added Verify method
1.3.1	06-06-2022	Added fuzzy method documentation for Match method
1.3.2	23-06-2022	Added <code>source_id</code> parameter to all methods