



DomainTools Iris Enrich API

Generated on April 21, 2026

Contents

1	Sample Response	3
2	Iris API Authentication	4
2.1	Overview	4
2.2	Protecting API Credentials	4
2.3	Open Key Authentication	4
2.4	API Key (Header) Authentication	4
2.4.1	Required API Key Authentication Parameters	4
2.4.2	Example Request with API Key Authentication	5
2.5	HMAC Signed Authentication	5
2.5.1	HMAC-Signed Authentication URI/Parameters	5
2.5.2	Creating a HMAC-signed authentication request	5
2.5.3	Security Note	6
3	Iris Rate Limits	7
3.1	Check your rate limits	7
3.2	Free test queries	7
3.2.1	Endpoints with free test queries	7
3.3	Iris Investigate rate limits and quota consumption	7
3.3.1	Duplicate query policies	7
3.3.2	Activities that consume quota	8
3.4	Iris Enrich rate limits	8
3.5	Iris Detect rate limits	9
3.5.1	Domain endpoints rate limits	9
3.5.2	Development and testing	9
3.5.3	Other endpoints	9
4	Iris API Error Codes	10
4.1	Status Codes	10
4.2	Error Response Format	10
4.3	Common Error Scenarios	11
4.3.1	Authentication Errors (401)	11
4.3.2	Authorization Errors (403)	11

4.3.3	Not Found Errors (404)	11
4.3.4	Rate Limiting (503)	11
4.4	Best Practices	11
4.5	Iris-Specific Notes	12
4.5.1	Partial Content (206)	12
4.5.2	Detect Endpoints	12
4.5.3	Validation Errors (422)	12
5	Iris Enrich API guide	14
5.1	Overview	14
5.2	API endpoint	14
5.3	What Iris Enrich does and doesn't do	14
5.3.1	Supported use case	14
5.3.2	Not supported	14
5.4	Authentication	14
5.5	Request parameters	15
5.5.1	Example request	15
5.5.2	Required parameters	15
5.5.3	Optional parameters	15
5.6	Response format	15
5.6.1	Example response structure	15
5.7	Working with RDAP and WHOIS registration data	16
5.7.1	Backward-compatible response structure	16
5.7.2	Registrar IANA ID	17
5.8	Working with screenshots	17
5.9	Appendix: API response examples	17
5.9.1	Basic enrichment request	17
5.9.2	Basic enrichment response	18
5.9.3	Full enrichment with WHOIS and RDAP	19
6	Working with screenshots in Iris APIs	37
6.1	Retrieve the screenshot URL	37
6.1.1	Example queries for screenshot URL retrieval	37
6.1.2	Query parameters for screenshot URL retrieval	37
6.1.3	Response fields for screenshot URL retrieval	37
6.1.4	Response example for screenshot URL retrieval	38
6.2	Retrieve and optionally resize the screenshot image via its URL	38
6.2.1	Query parameters for screenshot image file	38
6.2.2	Example query for screenshot image file	39
6.3	Interpreting screenshot metadata	39
6.3.1	Initial capture	39
6.3.2	Gather attempt was duplicate of initial screenshot	39
6.3.3	Recapture failed after previous successful capture	40
6.3.4	Duplicate capture followed by failed attempt	40

The Account Information API provides a quick and easy way to get a snapshot of API product usage for an account. Usage is broken down by day and by month.

<https://api.domaintools.com/v1/account/>

1 Sample Response

```
{
  "response": {
    "account": {
      "api_username": "domaintools-api-account",
      "active": true
    },
    "products": [
      {
        "id": "domain-profile",
        "per_month_limit": "100000",
        "per_minute_limit": "120",
        "absolute_limit": "1000",
        "usage": {
          "today": 5,
          "month": 152
        },
        "expiration_date": "2020-01-01"
      }
    ]
  }
}
```

2 Iris API Authentication

2.1 Overview

Most requests sent to the DomainTools API require authentication. We support two authentication schemes with different levels of security.

We recommend the following practices for authentication:

- Authentication with HTTPS (HTTP over SSL) instead of HTTP.
- Using HMAC signed queries or Header Authentication instead of Open Key Authentication.
- Using SHA-256 HMAC for the signed queries instead of MD5 or SHA-1.

2.2 Protecting API Credentials

Your account will be charged for all queries authenticated with your username and key, even if you later determine the requests were fraudulent or its use unauthorized. We recommend the following steps to protect your API credentials:

- For support requests, do not send the full API key to DomainTools support.
- Use a HMAC-signed approach to requests. This ensures that the user and token are not sent as part of the URL.
- When required, reset the API key.
- Do not place them into a public repository, such as a git repository.

2.3 Open Key Authentication

This is the easiest authentication scheme to implement, but also the most insecure. You should take precautions in the design of your application to ensure your key is not compromised.

```
https://api.domaintools.com/v1/domaintools.com?api_username=example&api_key=xxxxxx
```

Required Parameters	Value
api_username	Your API username
api_key	Your API key

2.4 API Key (Header) Authentication

Authenticate your requests by including the API key in the header of each HTTP request. The API key serves as a unique identifier and is used to authenticate your requests.

2.4.1 Required API Key Authentication Parameters

Required Parameters	Value
X-Api-Key	MY_API_KEY

2.4.2 Example Request with API Key Authentication

```
curl -H 'X-API-Key: MY_API_KEY'
'https://api.domaintools.com/v1/feed/nod/?after=-60'
```

2.5 HMAC Signed Authentication

HMAC, or hashed message authentication code, is our preferred authentication scheme. It follows the principles outlined in RFC2104 and provides a straightforward but secure method of protecting your API key.

It involves passing a hash composed of your api username, the current date and time, and the request URI. This hash is then signed with your authentication key. The result is a request that expires after a brief period of time and, most importantly, does not contain your authentication key.

MD5, SHA-1, and SHA-256 are supported HMAC hashing algorithms. We recommend SHA256 HMAC, as a successor of SHA1. Although NIST policy on hash functions still allows HMAC SHA-1, see [Hash Functions in the NIST CSRC](#) page. Overall, SHA-256 HMAC is also a more secure (ex, against brute force attacks) alternative to MD5 HMAC.

Please check your own compliance obligations, or guidance, as to what cryptographic functions should be used. For example, see the [NIST FIPS 140-2 Guidance](#) for more information for organizations that must use FIPS 140-2 compliant algorithms.

2.5.1 HMAC-Signed Authentication URI/Parameters

Required Parameters	Value
api_username	Your API username
timestamp	Current timestamp, in ISO 8601 format. Timestamps should contain the timezone offset, like below: 2020-02-01T14:37:59-0800 2020-02-01T22:37:59Z 2020-02-02T10:37:59+1200
signature	HMAC signature of your request. SHA-256 HMAC is recommended. MD5 HMAC and SHA-1 HMAC are supported. Only sign the URL path.

2.5.2 Creating a HMAC-signed authentication request

To create the signature, build a string with your API username, the timestamp in ISO 8601 format, and the URI you are requesting. That string is then signed with your API key using an HMAC function to generate the signature. The steps are below.

1. Add your api_username.
The API username is the account that has been provisioned access to the relevant DomainTools API product (aka API endpoint). If this username does not have access rights, an error message will be returned.
2. Associate a timestamp.
The timestamp value for the hash and timestamp parameter need to be identical. Make sure your server time is accurate and always use a fresh timestamp.

3. Add the request URI.
Please see the individual API product User Guides for the endpoint's URI format.
4. Add the host.
The host is the API endpoint server. In this case, it is api.domaintools.com.
5. Use the appropriate HMAC algorithm.
MD5, SHA-1, or SHA-256 for the hashing algorithm are supported. See the HMAC Algorithms section for more information.

The result is a request that expires after a brief period of time and, most importantly, does not contain your authentication key in the clear.

The resulting URLs will generally be similar, regardless of whichever programming language is used.

2.5.3 Security Note

Ensure that your current language version supports the required HMAC (and other relevant) extensions and libraries. For example, a failure in HMAC-signed URL will result in a URL with no signature value, such as below:

```
http://api.domaintools.com/v1/yourdomain.com/whois
```

3 Iris Rate Limits

This document explains rate limits and quota consumption for Iris Investigate, Iris Enrich, and Iris Detect in both the API and UI.

3.1 Check your rate limits

Use the following methods to determine the current rate limit associated with Iris Investigate, Iris Enrich, and Iris Detect API endpoints:

1. /account Endpoint

- Querying the Account Information endpoint provides details on query limits, usage, and expiration dates for all licensed Iris endpoints.
- For more information, refer to the [Account Information endpoint documentation](#).

2. API Admin Dashboard

- The API Admin can access their API dashboard to view query limits, usage, and expiration dates for all licensed endpoints.
- To access the API dashboard, visit the [DomainTools research page](#), select the account dropdown menu, and click on API Admin.

3.2 Free test queries

Multiple endpoints, including Iris Detect and Investigate, offer free test queries that don't count against your quota. Use these domains for testing and development:

- [domaintools.com](#)
- [dnsdb.info](#)
- [example.com](#)

3.2.1 Endpoints with free test queries

The following endpoints support free test queries with the domains listed above:

[Domain Profile](#), [Domain Reputation](#), [Hosting History](#), [Iris Enrich and Investigate](#), [Reverse IP](#), [WHOIS history](#), [WHOIS Lookups](#)

Using these test domains allows you to verify API integration and test functionality without consuming your quota.

3.3 Iris Investigate rate limits and quota consumption

The system measures quotas at the group level and resets them each month.

3.3.1 Duplicate query policies

Iris Investigate treats identical queries differently depending on whether you use the UI or API:

3.3.1.1 User Interface

Identical queries within 30 days don't count against your quota. This includes:

- All searches (omnisearch and advanced search)
- Search hash queries
- Search hash reloading
- Search node revisits

A query is considered identical if it has the same filters, sorting, and pagination parameters.

3.3.1.2 API

Regular queries: Identical queries within 1 hour don't count against quota. A query is considered identical if it has the same filters, sorting, and pagination parameters as a previous query made within the last hour.

Search hash queries: Identical search hash queries within 10 minutes don't count against quota. A search hash query uses the `search_hash` parameter to retrieve previously saved search results.

3.3.2 Activities that consume quota

3.3.2.1 Pivot Engine Queries in the Iris Investigate UI

The following activities consume your quota:

- Executing an omnisearch (from landing or search pages) that returns results
- Executing an advanced search that returns results
- Sending a result to the Pivot Engine (including `narrow`, `expand`, `new`, and `exclude` functions)
- Revisiting a search node more than 30 days since it was created
- Loading new pages in the Pivot Engine
- Sorting Pivot Engine results

For information on duplicate queries that don't consume quota, see [Duplicate query policies](#).

3.3.2.2 Passive DNS (pDNS) Queries in the Iris Investigate UI

The following activities consume your quota:

- Executing a search query that returns results (from either the search field or popovers throughout the UI)
- Executing/triggering the load more (infinite scroll) function in search results
- Revisiting a search node more than 30 days since it was created

For information on duplicate queries that don't consume quota, see [Duplicate query policies](#).

3.3.2.3 Queries in the Iris Investigate API

The following activities consume your quota:

- Executing a query
- Loading additional pages of results

For information on duplicate queries that don't consume quota, see [Duplicate query policies](#).

3.4 Iris Enrich rate limits

Iris Enrich uses an independent service level with its own rate limits, separate from Iris Investigate. This independence allows Iris Enrich to be used at much greater scale and throughput than Iris Investigate.

Key differences:

- **Independent quota:** Enrich queries don't count toward your Iris Investigate quota
- **Separate service level:** Rate limits are defined independently for Enrich
- **Higher throughput:** Optimized for high-volume domain enrichment use cases
- **No pivot parameters:** Unlike Investigate, Enrich is optimized for straightforward domain enrichment rather than pivoting

Your Enrich service level, query caps, and rate limits are configured independently on your enterprise account. For complete details on using Iris Enrich, consult the [Iris Enrich API documentation](#).

3.5 Iris Detect rate limits

Iris Detect has specific endpoint rate limits designed for monitoring and triage workflows.

3.5.1 Domain endpoints rate limits

The domains endpoints (/domains/new and /domains/watched) have a rate limit of **1 query per hour** for each endpoint.

Pagination: Requests using the offset parameter to retrieve additional pages of results aren't restricted by the hourly limit. If your hourly query returns more domains than the response limit, you can retrieve the complete result set using pagination without waiting.

3.5.2 Development and testing

During integration and testing, use the preview parameter to bypass rate limits:

- Including preview=1 in requests limits responses to 2 domains
- Allows up to 30 requests per minute
- Enables rapid iteration during development

3.5.3 Other endpoints

Monitor management endpoints and domain triage actions (watchlist, ignore, escalate) have standard API rate limits. For complete details on Iris Detect API endpoints and usage, consult the [Iris Detect API documentation](#).

4 Iris API Error Codes

The Iris API uses standard HTTP status codes to indicate the success or failure of requests. Understanding these codes helps you handle errors gracefully and troubleshoot issues effectively.

4.1 Status Codes

Status Code	Description
200	OK - The request was successful.
206	Partial Content - The request was partially successful. Some data is unavailable but the request returned what was available. This typically occurs when backend services are temporarily unavailable or when data for some requested items cannot be retrieved.
400	Bad Request - The request is malformed or contains invalid parameters. Check the error message for details about what needs to be corrected.
401	Unauthorized - Authentication credentials are missing, invalid, or expired. Verify your API key and authentication method.
403	Forbidden - The authenticated account does not have permission to access this resource or endpoint. This may indicate insufficient subscription level or access rights.
404	Not Found - The requested resource does not exist. This may occur when querying for a domain that has never been registered or for data that is not available in our systems.
500	Internal Server Error - An unexpected error occurred on the server. If this persists, contact DomainTools support at enterprisesupport@domaintools.com .
503	Service Unavailable - The service is temporarily unavailable, typically due to maintenance or high load. Implement exponential backoff and retry the request.

4.2 Error Response Format

Error responses follow a consistent JSON structure:

```
{
  "error": {
    "code": 400,
    "message": "Invalid domain name format"
  }
}
```

```
}  
}
```

4.3 Common Error Scenarios

4.3.1 Authentication Errors (401)

Cause: Missing or invalid API credentials.

Solutions:

- Verify your API key is correct
- Check that your authentication method (header, HMAC, or basic auth) is properly configured
- Ensure your API key hasn't expired

4.3.2 Authorization Errors (403)

Cause: Insufficient permissions or subscription level.

Solutions:

- Verify your account has access to the requested endpoint
- Check your subscription includes the API product you're trying to use
- Contact your account manager if you need additional access

4.3.3 Not Found Errors (404)

Cause: Requested resource doesn't exist.

Common scenarios:

- Querying a domain that has never been registered
- Requesting historical data that predates our records
- Using an invalid endpoint path

4.3.4 Rate Limiting (503)

Cause: Too many requests in a short time period.

Solutions:

- Implement exponential backoff retry logic
- Check your rate limits using the Account Information endpoint
- Distribute requests over time rather than in bursts
- Consider upgrading your subscription for higher rate limits

4.4 Best Practices

1. **Always check status codes:** Don't assume success - verify the response status code
2. **Parse error messages:** Error messages contain specific details about what went wrong
3. **Implement retry logic:** Use exponential backoff for 500 and 503 errors
4. **Log errors:** Keep detailed logs of errors for troubleshooting
5. **Monitor rate limits:** Track your API usage to avoid hitting limits

4.5 Iris-Specific Notes

4.5.1 Partial Content (206)

The Iris Enrich and Investigate endpoints may return a 206 status code when:

- Some backend data sources are temporarily unavailable
- Data for certain requested domains cannot be retrieved
- The response contains partial results rather than complete data

When you receive a 206 response, the returned data is still valid and usable - it simply indicates that some information is missing.

4.5.2 Detect Endpoints

Iris Detect endpoints (watchlists and monitors) use a subset of the standard error codes:

- 200 (OK)
- 400 (Bad Request)
- 401 (Unauthorized)
- 403 (Forbidden)
- 404 (Not Found)
- 422 (Unprocessable Entity) — write endpoints only

These endpoints do not return 500 or 503 errors in normal operation.

4.5.3 Validation Errors (422)

Iris Detect write endpoints (POST, PUT, and PATCH) return a 422 Unprocessable Entity status code when the request body is syntactically valid JSON but contains values that fail server-side validation. Unlike the standard error format, the 422 response includes per-field validation messages:

```
{
  "error": {
    "code": 422,
    "summary": "The given data was invalid.",
    "messages": {
      "term": ["The term field is required."],
      "watchlist_domain_ids": ["The watchlist domain ids field is required."]
    }
  },
  "resources": {
    "support": "https://docs.domaintools.com"
  }
}
```

Common validation triggers include:

- Missing required fields (for example, term when creating a monitor)
- Values that don't meet minimum length requirements (for example, term must be at least 3 characters)
- Invalid enum values (for example, an unrecognized state value)

This applies to the following endpoints:

Endpoint	Method
/v1/iris-detect/domains/	PATCH
/v1/iris-detect/escalations/	POST
/v1/iris-detect/monitors/	POST
/v1/iris-detect/monitors/	PUT

Iris Enrich and Iris Investigate endpoints do not return 422. These endpoints return 400 for invalid parameters.

5 Iris Enrich API guide

5.1 Overview

The Iris Enrich API provides high-volume domain enrichment optimized for SIEM and SOAR platform integration. Enrich up to 100 domains per request with domain risk scores, RDAP, WHOIS, IP, DNS, website, and SSL/TLS certificate data. For a complete overview of Iris Enrich capabilities and use cases, see the [Iris Enrich overview](#).

The Iris Enrich API provides high-volume domain enrichment with actionable threat intelligence and domain metadata. The API is optimized for batch processing and fast response times, making it ideal for enriching proxy logs, DNS logs, and other domain data at scale within SIEM solutions like Splunk or QRadar, or custom data analytics platforms.

Key characteristics:

- Batch enrichment of up to 100 domains per request
- Multiple domain attributes including risk scores, RDAP, WHOIS, IP, DNS, website, and SSL/TLS certificate data
- Dedicated service levels with customized rate limiting
- Optimized for enrichment workflows
- Domain-name queries only (forward lookups)

5.2 API endpoint

The API returns JSON results and supports both POST and GET requests at this endpoint:

```
https://api.domaintools.com/v1/iris-enrich/
```

5.3 What Iris Enrich does and doesn't do

Iris Enrich is designed specifically for **forward lookups**: you provide domain names and receive enrichment data about those domains.

5.3.1 Supported use case

Query domains to retrieve enrichment data including risk scores, registration information, DNS records, and SSL/TLS certificate details.

5.3.2 Not supported

- **Reverse lookups/pivoting**: You can't query by IP addresses, nameservers, SSL/TLS certificate hashes, or other attributes to find domains. Iris Enrich only accepts domain names as input.
- **Domain counts**: The API doesn't include counts of how many domains share a given attribute value (such as the number of domains on the same IP address).

For reverse lookups, pivoting capabilities, and domain counts, use the [Iris Investigate API](#) instead.

5.4 Authentication

The Iris Enrich API uses the [same authentication mechanisms as the Iris Investigate API](#) (open-key or signed). However, unlike the Investigate API, the Iris Enrich API uses an

independent service level to define access levels, query caps, and rate limits. It does not pull from the same queries as the Iris Investigate UI and can be used at much greater scale and throughput. The API endpoint must be explicitly configured on an enterprise account.

For complete information about Iris API rate limits and how Enrich differs from other Iris products, consult the [Iris API rate limits](#) documentation.

5.5 Request parameters

The Iris Enrich API is optimized for fast responses and high-volume lookups. Provide a list of up to 100 domains in the `domain` parameter (comma-separated).

5.5.1 Example request

```
https://api.domaintools.com/v1/iris-enrich/?domain=domaintools.com,domaintools.net
```

5.5.2 Required parameters

Parameter	Type	Description
<code>domain</code>	string	Comma-separated list of up to 100 domains to enrich.

5.5.3 Optional parameters

Parameter	Type	Description
<code>parsed_whois</code>	boolean	Set to <code>true</code> to include the full parsed WHOIS record in the response.
<code>parsed_domain_rdap</code>	boolean	Set to <code>true</code> to include the full parsed RDAP record in the response.
<code>screenshots</code>	flag	Set to <code>1</code> to include screenshot data in the response. See working with screenshots for details.

5.6 Response format

The Iris Enrich API response format differs from the Investigate API in several key ways:

- Counts of connected domains are not included
- Most domain attribute values still appear under the `value` subkey for consistency with the Investigate API
- An additional `missing_domains` key lists any domains submitted in the `domain` parameter that were not found in the Iris dataset

5.6.1 Example response structure

```
{  
  "response": {
```

```
"limit_exceeded": false,
"message": "Success",
"results_count": 2,
"has_more_results": false,
"results": [
  {
    "domain": "domaintools.com",
    "active": true,
    "domain_risk": {
      "risk_score": 0,
      "components": []
    },
    "create_date": {
      "value": "1998-08-02"
    }
  }
],
"missing_domains": []
}
```

5.7 Working with RDAP and WHOIS registration data

Domain registries and registrars are transitioning from WHOIS to the Registration Data Access Protocol (RDAP) as a method to communicate domain registration data.

DomainTools supports both WHOIS and RDAP registration data in the Iris suite. These updates are available now, prior to the date at which ICANN will permit gTLD registries and registrars to sunset WHOIS in their Registration Data Directory Services (RDDS).

Note that these changes apply to the subset of registration data and not other domain records provided in the responses.

5.7.1 Backward-compatible response structure

Set parameters for `parsed_whois=true` and/or `parsed_domain_rdap=true` to return the parsed WHOIS and/or parsed RDAP record along with the default response.

5.7.1.1 Default query

https://api.domaintools.com/v1/iris-enrich/?domain=github.com&api_username=USERNAME&api_key=KEY

This query returns standard registration data as part of the response root fields. Registration data is from either RDAP or WHOIS, depending on which record is more complete. Registration data is included at the same level as other domain data.

Response structure:

```
response:
  limit_exceeded
  has_more_results
  message
  result_count
  total_count
  results
```

```
domain
whois_url
adsense
alexa
popularity_rank
...
```

5.7.1.2 Query with full WHOIS and RDAP records

https://api.domaintools.com/v1/iris-enrich/?domain=github.com&parsed_whois=true&parsed_domain_rdap=true

Registration data is included as before, but with additional root-level fields for `parsed_whois` and `parsed_domain_rdap`:

```
response:
  limit_exceeded
  has_more_results
  message
  result_count
  total_count
  results
    domain
    whois_url
    adsense
    alexa
    popularity_rank
    ...
    parsed_whois # PARSED WHOIS FIELD
      registrant_contact
      ...
    parsed_domain_rdap # PARSED RDAP FIELD
      admin_contact
      ...
```

5.7.2 Registrar IANA ID

The domain registrar's IANA ID is included in the RDAP object when the `parsed_domain_rdap=true` parameter is used in an API request.

5.8 Working with screenshots

You can retrieve domain screenshots through the Iris Enrich API by adding the `screenshots=1` parameter to your request. For detailed information about retrieving, resizing, and interpreting screenshot data, see [working with screenshots in Iris APIs](#).

5.9 Appendix: API response examples

5.9.1 Basic enrichment request

This example shows a simple enrichment request for two domains:

```
GET https://api.domaintools.com/v1/iris-
  ↪ enrich/?domain=domaintools.com,example.com&api_username=USERNAME&api_key=KEY
```

5.9.2 Basic enrichment response

The response includes enriched data for each domain with most values nested under a value subkey:

```
{
  "response": {
    "limit_exceeded": false,
    "message": "Enjoy your data.",
    "results_count": 2,
    "results": [
      {
        "domain": "domaintools.com",
        "active": true,
        "popularity_rank": 2417,
        "domain_risk": {
          "risk_score": 0,
          "components": [
            {
              "name": "zerolist",
              "risk_score": 0
            }
          ]
        },
        "create_date": {
          "value": "1998-08-02"
        },
        "expiration_date": {
          "value": "2027-08-01"
        },
        "registrar": {
          "value": "ENOM, INC."
        },
        "ip": [
          {
            "address": {
              "value": "141.193.213.20"
            },
            "country_code": {
              "value": "us"
            },
            "isp": {
              "value": "WPEngine Inc."
            }
          }
        ],
        "name_server": [
          {
            "host": {
              "value": "dns1.p04.nspace.net"
            },
            "domain": {
              "value": "nspace.net"
            }
          }
        ]
      }
    ],
    "missing_domains": []
  }
}
```

```
}
```

5.9.3 Full enrichment with WHOIS and RDAP

This example shows a request that includes both parsed WHOIS and RDAP records:

5.9.3.1 Request example

GET https://api.domaintools.com/v1/iris-enrich/?domain=domaintools.com,example.com&parsed_whois=true

5.9.3.2 Response example

The response includes the standard enrichment data plus additional `parsed_whois` and `parsed_domain_rdap` objects for each domain:

```
{
  "response": {
    "limit_exceeded": false,
    "message": "Enjoy your data.",
    "results_count": 2,
    "results": [
      {
        "domain": "domaintools.com",
        "whois_url": "https://whois.domaintools.com/domaintools.com",
        "adsense": {
          "value": ""
        },
        "alexa": 3954,
        "popularity_rank": 2417,
        "active": true,
        "google_analytics": {
          "value": ""
        },
        "ga4": [
          {
            "value": "G-RPLVMKCB3Y"
          }
        ],
        "gtm_codes": [
          {
            "value": "GTM-5P2JCN"
          }
        ],
        "fb_codes": [],
        "hotjar_codes": [],
        "baidu_codes": [],
        "yandex_codes": [],
        "matomo_codes": [],
        "statcounter_project_codes": [],
        "statcounter_security_codes": [],
        "admin_contact": {
          "name": {
            "value": ""
          },
          "org": {
            "value": ""
          }
        },

```

```
"street": {
  "value": ""
},
"city": {
  "value": ""
},
"state": {
  "value": ""
},
"postal": {
  "value": ""
},
"country": {
  "value": ""
},
"phone": {
  "value": ""
},
"fax": {
  "value": ""
},
"email": []
},
"billing_contact": {
  "name": {
    "value": ""
  },
  "org": {
    "value": ""
  },
  "street": {
    "value": ""
  },
  "city": {
    "value": ""
  },
  "state": {
    "value": ""
  },
  "postal": {
    "value": ""
  },
  "country": {
    "value": ""
  },
  "phone": {
    "value": ""
  },
  "fax": {
    "value": ""
  },
  "email": []
},
"registrant_contact": {
  "name": {
    "value": "Domain Administrator"
  },
  "org": {
    "value": "DomainTools, LLC"
  }
}
```

```
    },
    "street": {
      "value": "2101 4th Ave Suite 1720"
    },
    "city": {
      "value": "Seattle"
    },
    "state": {
      "value": "WA"
    },
    "postal": {
      "value": "98121"
    },
    "country": {
      "value": "US"
    },
    "phone": {
      "value": "tel:+1.2068389035"
    },
    "fax": {
      "value": ""
    },
    "email": [
      {
        "value": "memberservices@domaintools.com"
      }
    ]
  },
  "technical_contact": {
    "name": {
      "value": ""
    },
    "org": {
      "value": ""
    },
    "street": {
      "value": ""
    },
    "city": {
      "value": ""
    },
    "state": {
      "value": ""
    },
    "postal": {
      "value": ""
    },
    "country": {
      "value": ""
    },
    "phone": {
      "value": ""
    },
    "fax": {
      "value": ""
    },
    "email": []
  },
  "create_date": {
```

```
    "value": "1998-08-02"
  },
  "expiration_date": {
    "value": "2027-08-01"
  },
  "email_domain": [
    {
      "value": "enom.com"
    },
    {
      "value": "nsone.net"
    },
    {
      "value": "domaintools.com"
    }
  ],
  "soa_email": [
    {
      "value": "hostmaster@nsone.net"
    }
  ],
  "ssl_email": [],
  "additional_whois_email": [
    {
      "value": "abuse@enom.com"
    }
  ],
  "ip": [
    {
      "address": {
        "value": "141.193.213.20"
      },
      "asn": [
        {
          "value": 209242
        }
      ],
      "country_code": {
        "value": "us"
      },
      "isp": {
        "value": "WPEngine Inc."
      }
    },
    {
      "address": {
        "value": "141.193.213.21"
      },
      "asn": [
        {
          "value": 209242
        }
      ],
      "country_code": {
        "value": "us"
      },
      "isp": {
        "value": "WPEngine Inc."
      }
    }
  ]
}
```

```
    }
  ],
  "mx": [
    {
      "host": {
        "value": "aspmx.l.google.com"
      },
      "domain": {
        "value": "google.com"
      },
      "ip": [
        {
          "value": "74.125.195.26"
        }
      ],
      "priority": 1
    },
    {
      "host": {
        "value": "alt1.aspmx.l.google.com"
      },
      "domain": {
        "value": "google.com"
      },
      "ip": [
        {
          "value": "142.250.101.26"
        }
      ],
      "priority": 5
    },
    {
      "host": {
        "value": "alt2.aspmx.l.google.com"
      },
      "domain": {
        "value": "google.com"
      },
      "ip": [
        {
          "value": "192.178.164.27"
        }
      ],
      "priority": 5
    },
    {
      "host": {
        "value": "aspmx2.googlemail.com"
      },
      "domain": {
        "value": "googlemail.com"
      },
      "ip": [
        {
          "value": "142.250.101.27"
        }
      ],
      "priority": 10
    }
  ],
}
```

```
{
  "host": {
    "value": "aspmx3.googlemail.com"
  },
  "domain": {
    "value": "googlemail.com"
  },
  "ip": [
    {
      "value": "192.178.164.26"
    }
  ],
  "priority": 10
},
{
  "host": {
    "value": "aspmx4.googlemail.com"
  },
  "domain": {
    "value": "googlemail.com"
  },
  "ip": [
    {
      "value": "192.178.220.27"
    }
  ],
  "priority": 10
}
],
"name_server": [
  {
    "host": {
      "value": "dns3.p04.nsonone.net"
    },
    "domain": {
      "value": "nsonone.net"
    },
    "ip": [
      {
        "value": "198.51.44.68"
      }
    ]
  }
],
{
  "host": {
    "value": "dns1.p04.nsonone.net"
  },
  "domain": {
    "value": "nsonone.net"
  },
  "ip": [
    {
      "value": "198.51.44.4"
    }
  ]
}
],
{
  "host": {
    "value": "dns2.p04.nsonone.net"
```

```
    },
    "domain": {
      "value": "nsone.net"
    },
    "ip": [
      {
        "value": "198.51.45.4"
      }
    ]
  },
  {
    "host": {
      "value": "dns4.p04.nsone.net"
    },
    "domain": {
      "value": "nsone.net"
    },
    "ip": [
      {
        "value": "198.51.45.68"
      }
    ]
  }
],
"domain_risk": {
  "risk_score": 0,
  "components": [
    {
      "name": "zerolist",
      "risk_score": 0
    }
  ]
},
"redirect": {
  "value": ""
},
"redirect_domain": {
  "value": ""
},
"registrant_name": {
  "value": "Domain Administrator"
},
"registrant_org": {
  "value": "DomainTools, LLC"
},
"registrar": {
  "value": "ENOM, INC."
},
"registrar_status": [
  "client transfer prohibited"
],
"spf_info": "",
"ssl_info": [
  {
    "hash": {
      "value": "11fc54c990c58d66f986320c9bc72052f541fe92"
    },
    "subject": {
      "value": "CN=www.domaintools.com"
    }
  }
]
```

```
    },
    "organization": {
      "value": ""
    },
    "email": [],
    "alt_names": [
      {
        "value": "www.domaintools.com"
      },
      {
        "value": "blog.domaintools.com"
      },
      {
        "value": "domaintools.com"
      }
    ],
    "sources": {
      "active": 1764613130569,
      "passive": 1759158600086
    },
    "common_name": {
      "value": "www.domaintools.com"
    },
    "issuer_common_name": {
      "value": "Sectigo Public Server Authentication CA DV R36"
    },
    "not_after": {
      "value": 20260919
    },
    "not_before": {
      "value": 20250820
    },
    "duration": {
      "value": 396
    }
  }
],
"tld": "com",
"website_response": 200,
"data_updated_timestamp": "2025-12-03T16:37:56.267000",
"website_title": {
  "value": "DomainTools - The first place to go when you need to know."
},
"server_type": {
  "value": "cloudflare"
},
"first_seen": {
  "value": "2001-10-26T00:00:00Z"
},
"tags": [],
"parsed_whois": {
  "admin_contact": {
    "country": {
      "value": "REDACTED FOR PRIVACY"
    },
    "street": {
      "value": "REDACTED FOR PRIVACY"
    },
    "postal": {
```

```
    "value": "REDACTED FOR PRIVACY"
  },
  "city": {
    "value": "REDACTED FOR PRIVACY"
  },
  "org": {
    "value": "REDACTED FOR PRIVACY"
  },
  "state": {
    "value": "REDACTED FOR PRIVACY"
  },
  "name": {
    "value": "REDACTED FOR PRIVACY"
  },
  "email": [
    {
      "value": "redacted for privacy"
    }
  ]
},
"technical_contact": {
  "country": {
    "value": "REDACTED FOR PRIVACY"
  },
  "street": {
    "value": "REDACTED FOR PRIVACY"
  },
  "postal": {
    "value": "REDACTED FOR PRIVACY"
  },
  "city": {
    "value": "REDACTED FOR PRIVACY"
  },
  "org": {
    "value": "REDACTED FOR PRIVACY"
  },
  "state": {
    "value": "REDACTED FOR PRIVACY"
  },
  "name": {
    "value": "REDACTED FOR PRIVACY"
  },
  "email": [
    {
      "value": "redacted for privacy"
    }
  ]
},
"registrant_contact": {
  "country": {
    "value": "us"
  },
  "street": {
    "value": "2101 4th Ave"
  },
  "fax": {
    "value": "12068389056"
  },
  "postal": {
```

```
    "value": "98121"
  },
  "city": {
    "value": "Seattle"
  },
  "org": {
    "value": "DomainTools, LLC"
  },
  "state": {
    "value": "WA"
  },
  "phone": {
    "value": "12068389035"
  },
  "name": {
    "value": "Domain Administrator"
  },
  "email": [
    {
      "value": "memberservices@domaintools.com"
    }
  ]
},
"registrar": {
  "value": "eNom, LLC"
},
"expiration_date": {
  "value": "2027-08-01"
},
"additional_whois_email": [
  {
    "value": "abuse@enom.com"
  }
],
"create_date": {
  "value": "1998-08-02"
},
"registrant_name": {
  "value": "Domain Administrator"
},
"registrant_org": {
  "value": "DomainTools, LLC"
},
"registrar_status": [
  "clienttransferprohibited"
],
"email_domain": [
  {
    "value": "enom.com"
  },
  {
    "value": "nsone.net"
  }
]
},
"parsed_domain_rdap": {
  "registrant_contact": {
    "country": {
      "value": "US"
    }
  }
}
```

```
    },
    "street": {
      "value": "2101 4th Ave Suite 1720"
    },
    "postal": {
      "value": "98121"
    },
    "city": {
      "value": "Seattle"
    },
    "org": {
      "value": "DomainTools, LLC"
    },
    "state": {
      "value": "WA"
    },
    "phone": {
      "value": "tel:+1.2068389035"
    },
    "name": {
      "value": "Domain Administrator"
    },
    "email": [
      {
        "value": "memberservices@domaintools.com"
      }
    ]
  },
  "registrant_name": {
    "value": "Domain Administrator"
  },
  "registrant_org": {
    "value": "DomainTools, LLC"
  },
  "registrar_iana_id": {
    "value": "48"
  },
  "additional_whois_email": [
    {
      "value": "abuse@enom.com"
    }
  ],
  "registrar": {
    "value": "ENOM, INC."
  },
  "create_date": {
    "value": "1998-08-02"
  },
  "registrar_status": [
    "client transfer prohibited"
  ],
  "expiration_date": {
    "value": "2027-08-01"
  },
  "email_domain": [
    {
      "value": "enom.com"
    },
    {
```

```
        "value": "nsonline.net"
      },
      {
        "value": "domaintools.com"
      }
    ]
  },
  {
    "domain": "example.com",
    "whois_url": "https://whois.domaintools.com/example.com",
    "adsense": {
      "value": ""
    },
    "alexa": 26830,
    "popularity_rank": 299,
    "active": true,
    "google_analytics": {
      "value": ""
    },
    "ga4": [],
    "gtm_codes": [],
    "fb_codes": [],
    "hotjar_codes": [],
    "baidu_codes": [],
    "yandex_codes": [],
    "matomo_codes": [],
    "statcounter_project_codes": [],
    "statcounter_security_codes": [],
    "admin_contact": {
      "name": {
        "value": ""
      },
      "org": {
        "value": ""
      },
      "street": {
        "value": ""
      },
      "city": {
        "value": ""
      },
      "state": {
        "value": ""
      },
      "postal": {
        "value": ""
      },
      "country": {
        "value": ""
      },
      "phone": {
        "value": ""
      },
      "fax": {
        "value": ""
      },
      "email": []
    },
  },
}
```

```
"billing_contact": {
  "name": {
    "value": ""
  },
  "org": {
    "value": ""
  },
  "street": {
    "value": ""
  },
  "city": {
    "value": ""
  },
  "state": {
    "value": ""
  },
  "postal": {
    "value": ""
  },
  "country": {
    "value": ""
  },
  "phone": {
    "value": ""
  },
  "fax": {
    "value": ""
  },
  "email": []
},
"registrant_contact": {
  "name": {
    "value": ""
  },
  "org": {
    "value": "Internet Assigned Numbers Authority"
  },
  "street": {
    "value": ""
  },
  "city": {
    "value": ""
  },
  "state": {
    "value": ""
  },
  "postal": {
    "value": ""
  },
  "country": {
    "value": ""
  },
  "phone": {
    "value": ""
  },
  "fax": {
    "value": ""
  },
  "email": []
}
```

```
    },
    "technical_contact": {
      "name": {
        "value": ""
      },
      "org": {
        "value": ""
      },
      "street": {
        "value": ""
      },
      "city": {
        "value": ""
      },
      "state": {
        "value": ""
      },
      "postal": {
        "value": ""
      },
      "country": {
        "value": ""
      },
      "phone": {
        "value": ""
      },
      "fax": {
        "value": ""
      },
      "email": []
    },
    "create_date": {
      "value": "1992-01-01"
    },
    "expiration_date": {
      "value": "2026-08-13"
    },
    "email_domain": [
      {
        "value": "icann.org"
      }
    ],
    "soa_email": [
      {
        "value": "noc.dns@icann.org"
      }
    ],
    "ssl_email": [],
    "additional_whois_email": [],
    "ip": [
      {
        "address": {
          "value": "23.215.0.136"
        },
        "asn": [
          {
            "value": 20940
          }
        ]
      }
    ],
```

```
    "country_code": {
      "value": "us"
    },
    "isp": {
      "value": "Akamai Technologies Inc."
    }
  },
  {
    "address": {
      "value": "23.215.0.138"
    },
    "asn": [
      {
        "value": 20940
      }
    ],
    "country_code": {
      "value": "us"
    },
    "isp": {
      "value": "Akamai Technologies Inc."
    }
  }
],
"mx": [],
"name_server": [
  {
    "host": {
      "value": "a.iana-servers.net"
    },
    "domain": {
      "value": "iana-servers.net"
    },
    "ip": [
      {
        "value": "199.43.135.53"
      }
    ]
  },
  {
    "host": {
      "value": "b.iana-servers.net"
    },
    "domain": {
      "value": "iana-servers.net"
    },
    "ip": [
      {
        "value": "199.43.133.53"
      }
    ]
  }
],
"domain_risk": {
  "risk_score": 1,
  "components": [
    {
      "name": "proximity",
      "risk_score": 1
    }
  ]
}
```

```
    }
  ]
},
"redirect": {
  "value": ""
},
"redirect_domain": {
  "value": ""
},
"registrant_name": {
  "value": ""
},
"registrant_org": {
  "value": "Internet Assigned Numbers Authority"
},
"registrar": {
  "value": "RESERVED-Internet Assigned Numbers Authority"
},
"registrar_status": [
  "clientdeleteprohibited",
  "clienttransferprohibited",
  "clientupdateprohibited"
],
"spf_info": "",
"ssl_info": [
  {
    "hash": {
      "value": "310db7af4b2bc9040c8344701aca08d0c69381e3"
    },
    "subject": {
      "value": "CN=*.example.com,O=Internet Corporation for Assigned Names
↵ and Numbers,L=Los Angeles,ST=California,C=US"
    },
    "organization": {
      "value": "Internet Corporation for Assigned Names and Numbers"
    },
    "email": [],
    "alt_names": [
      {
        "value": "*.example.com"
      },
      {
        "value": "example.com"
      }
    ],
    "sources": {
      "active": 1763429847000
    },
    "common_name": {
      "value": "*.example.com"
    },
    "issuer_common_name": {
      "value": "DigiCert Global G3 TLS ECC SHA384 2020 CA1"
    },
    "not_after": {
      "value": 20260115
    },
    "not_before": {
      "value": 20250115
    }
  }
]
```

```
    },
    "duration": {
      "value": 366
    }
  ],
  "tld": "com",
  "website_response": 200,
  "data_updated_timestamp": "2025-12-03T16:37:01.621000",
  "website_title": {
    "value": "Example Domain"
  },
  "server_type": {
    "value": "ECS (laa/7BA2)"
  },
  "first_seen": {
    "value": "2002-01-07T00:00:00Z"
  },
  "tags": [],
  "parsed_whois": {
    "registrant_contact": {
      "org": {
        "value": "Internet Assigned Numbers Authority"
      }
    },
    "registrar": {
      "value": "RESERVED-Internet Assigned Numbers Authority"
    },
    "expiration_date": {
      "value": "2026-08-13"
    },
    "create_date": {
      "value": "1992-01-01"
    },
    "registrant_org": {
      "value": "Internet Assigned Numbers Authority"
    },
    "registrar_status": [
      "clientdeleteprohibited",
      "clienttransferprohibited",
      "clientupdateprohibited"
    ],
    "email_domain": [
      {
        "value": "icann.org"
      }
    ]
  },
  "parsed_domain_rdap": {
    "registrar_iana_id": {
      "value": "376"
    },
    "registrar": {
      "value": "RESERVED-Internet Assigned Numbers Authority"
    },
    "create_date": {
      "value": "1995-08-14"
    },
    "registrar_status": [
```

```
        "client delete prohibited",
        "client transfer prohibited",
        "client update prohibited"
    ],
    "expiration_date": {
        "value": "2025-08-13"
    },
    "email_domain": [
        {
            "value": "icann.org"
        }
    ]
}
],
"missing_domains": []
}
```

Note the key differences when `parsed_whois=true` and `parsed_domain_rdap=true` are included:

- Each domain result includes `parsed_whois` and `parsed_domain_rdap` objects
- The `parsed_domain_rdap` object includes the `registrar_iana_id` field
- WHOIS data may show “REDACTED FOR PRIVACY” for privacy-protected domains
- RDAP data typically provides more structured and complete registration information

6 Working with screenshots in Iris APIs

Retrieve the most recent screenshot for a given domain with Iris APIs in a two-step process:

1. [Retrieve the screenshot image URL](#)
2. [Retrieve and optionally resize the image file](#)

Also consult the section on [interpreting screenshot metadata](#).

6.1 Retrieve the screenshot URL

Append queries with `screenshots=1` to receive a screenshot JSON object in the response.

For example, the following Iris API queries (with header authentication) include screenshot requests.

6.1.1 Example queries for screenshot URL retrieval

- Iris Detect:
`https://api.domaintools.com/v1/iris-detect/domains/watched/?screenshots=1`
- Iris Enrich:
`https://api.domaintools.com/v1/iris-enrich/?domain=example.com&screenshots=1`
- Iris Investigate: `https://api.domaintools.com/v1/iris-investigate/?domain=example.com&screenshots=1`

6.1.2 Query parameters for screenshot URL retrieval

Query Parameter	Required	Type	Valid Values	Description	Example
<code>screenshots</code>	yes	flag	1	Triggers screenshot object in response.	<code>screenshots=1</code>

6.1.3 Response fields for screenshot URL retrieval

Output field	Type	Valid Values	Description	Example
<code>ip_address</code>	string null	IP address	The IP address from which the screenshot was taken.	212.129.31.169
<code>last_attempt</code>	string	ISO 8601 date-time format	The date and time the screenshot was last attempted.	2025-02-17T15:34:55Z
<code>last_seen</code>	string	ISO 8601 date-time format	The date and time the screenshot was last observed	2025-02-17T15:34:55Z
<code>screenshot</code>	object	JSON object	Details about the screenshot.	

Output field	Type	Valid Values	Description	Example
src	str	URL	Link to the screenshot.	https://screenshots.ne.domaintool image?s3v=1&rurl=http%3A%2F%2Fesp 03- 28T21%3A48%3A37Z&token=dec12499e0
timestamp	string	ISO 8601 date-time format	The date and time the provided screenshot (in fullsize field) was obtained.	2024-08- 05T16:17:45Z

6.1.4 Response example for screenshot URL retrieval

```
"screenshot": {
  "timestamp": "2025-03-07T19:53:44Z",
  "src": "https://screenshots.ar.domaintools.com/screenshot_im-
↪ age?s3v=1&rurl=http%3A%2F%2Fdomaintools.com&ts=2025-03-
↪ 07T19%3A53%3A44Z&token=2d6269e40f4bc48908b17d218e5647fc90c0034618e39ec20806910d3909785c",
  "ip_address": "141.193.213.21",
  "last_attempt": "2025-04-07T18:00:01Z",
  "last_seen": "2025-04-07T18:00:01Z"
}
```

6.2 Retrieve and optionally resize the screenshot image via its URL

Use the `src` URL in the returned screenshot object to download the screenshot image. The URL already contains the required parameters `domain`, `ts`, `rurl`, and `token`. Optionally add `resize_width` and `crop_height` to resize the image.

6.2.1 Query parameters for screenshot image file

Query Parameter	Required	Type	Valid Values	Description	Example
<code>resize_width</code>	no	integer	Width in pixels	If the value is less than the width of the image, the image will be shrunk horizontally to the given <code>resize_width</code> , preserving the aspect ratio.	<code>resize_width=500</code>

Query Parameter	Required	Type	Valid Values	Description	Example
crop_height	no	integer	Height in pixels	If the value is less than the height of the image, the image will be cropped vertically to the given crop_height, removing pixels from the bottom of the image.	crop_height=500

6.2.2 Example query for screenshot image file

```
api.domaintools.com/screenshot_
image?domain=domaintools.com&ts=2025-04-30T14:32:10Z&resize_width=500&crop_
height=500&rurl=https://www.domaintools.com/
```

6.3 Interpreting screenshot metadata

Use the `last_attempt`, `last_seen`, and `timestamp` to interpret the screenshot.

To confirm that the most recent screenshot was gathered during the current domain lifecycle—and isn't from an older, historic screenshot—compare the screenshot's `timestamp` with the domain's `first_seen` value in the API response. If the `timestamp` is later than `first_seen`, the screenshot was taken during the current domain lifecycle.

6.3.1 Initial capture

The screenshot was successfully captured during the first and only attempt. Since there have been no subsequent checks, all `timestamp` fields are identical.

Relationships: `timestamp = last_seen = last_attempt`

Response example:

```

"screenshot": {
  "timestamp": "2025-05-12T14:38:19Z",
  "src": "https://screenshots.ar.domaintools.com/screenshot_im-
↪ age?s3v=1&rurl=http%3A%2F%2Fxi.mi.fr&ts=2025-05-
↪ 12T14%3A38%3A19Z&token=5fbd3aa604682d26d9f1f69810a87bfd75ea25e56eadb3553cefcff2ba804f28",
  "ip_address": "76.76.21.21",
  "last_attempt": "2025-05-12T14:38:19Z",
  "last_seen": "2025-05-12T14:38:19Z"
}
```

6.3.2 Gather attempt was duplicate of initial screenshot

DomainTools re-checked a domain after previously taking an initial screenshot. The result was identical to the existing image and no new screenshot was generated.

Relationships: `timestamp < last_seen = last_attempt`

- `timestamp` remains unchanged, preserving the original capture time.
- `last_attempt` is updated to reflect the most recent screenshot check.
- `last_seen` is updated to indicate the most recent time the image was confirmed valid.
- Because no new image was needed, `last_seen` and `last_attempt` are equal.

Response example:

```

"screenshot": {
  "timestamp": "2025-05-12T10:10:49Z",
  "src": "https://screenshots.ar.domaintools.com/screenshot_image?s3v=1&rurl=http%3A%2F%2Fdomaintools.co.in&ts=2025-05-12T10%3A10%3A49Z&token=28d255010a09d1cc8a3326e1c1406971910293f09ddac0159eb35391f705be57",
  "ip_address": "104.219.250.192",
  "last_attempt": "2025-05-13T17:16:34Z",
  "last_seen": "2025-05-13T17:16:34Z"
}

```

6.3.3 Recapture failed after previous successful capture

DomainTools successfully captured a screenshot on a previous attempt. The next attempt to capture the screenshot failed — the system was unable to retrieve a new image and could not confirm whether the content had changed.

Relationships: `timestamp = last_seen`, and `last_seen < last_attempt`

- `timestamp` reflects the time the screenshot was originally captured.
- `last_seen` is equal to `timestamp`, because that was the last time a screenshot was successfully observed
- `last_attempt` is more recent than `last_seen`, but the attempt failed

Response example:

```

"screenshot": {
  "timestamp": "2023-12-19T22:17:10Z",
  "src": "https://screenshots.ar.domaintools.com/screenshot_image?s3v=1&rurl=http%3A%2F%2Ffixhelpdesk.com&ts=2023-12-19T22%3A17%3A10Z&token=8cd6edb822594a0aa485e72f5cbb5414a73b657f7438b770844f46fe886c4ea9",
  "ip_address": "91.195.240.19",
  "last_attempt": "2025-05-13T10:25:04Z",
  "last_seen": "2023-12-19T22:17:10Z"
}

```

6.3.4 Duplicate capture followed by failed attempt

A screenshot was captured, and subsequently re-captured as a duplicate. A subsequent attempt to refresh the (duplicate) screenshot failed.

Relationships: `timestamp < last_seen`, and `last_seen < last_attempt`

- `timestamp` reflects the original screenshot date
- `last_seen` is later than `timestamp`, indicating the image was revalidated as a duplicate on a later date
- `last_attempt` reflects a more recent failed attempt to recapture the screenshot

Example response:

```
    "screenshot": {
      "timestamp": "2022-07-02T10:03:43Z",
      "src": "https://screenshots.ar.domaintools.com/screenshot_im-
↳ age?s3v=1&rurl=http%3A%2F%2Fdomaintools.ca&ts=2022-07-
↳ 02T10%3A03%3A43Z&token=6fa06c07ff1b7206ae8f38aadaf4e11b8cf538b7625944e2813a72ebec2c6648",
      "ip_address": "158.85.87.76",
      "last_attempt": "2025-05-13T10:12:04Z",
      "last_seen": "2024-05-22T10:11:09Z"
    }
```