

# DataForSEO On-Page API: Complete Reference with cURL & Response Examples

## Executive Summary

This comprehensive guide provides **complete cURL request examples and JSON response formats for all 15+ DataForSEO On-Page API endpoints**. Every endpoint includes:

- Exact cURL bash command
- All request parameters
- Complete JSON response format
- Real-world example data

**Reference Structure:** Each endpoint section contains three parts:

1. cURL Request - Copy-paste ready bash command with full parameters
2. JSON Response - Complete response structure with sample data
3. Parameters - Detailed explanation of each field

---

## Table of Contents

1. API Architecture & Authentication
  2. Task POST - Initiate Crawling
  3. Summary - Get Crawl Overview
  4. Pages - Individual Page Data
  5. Resources - Get All Resources
  6. Links - Link Analysis
  7. Duplicate Tags - Meta Tag Duplicates
  8. Duplicate Content - Content Duplicates
  9. Redirect Chains - Redirect Issues
  10. Non-Indexable - Blocked Pages
  11. Waterfall - Page Speed Data
  12. Keyword Density - Keyword Analysis
  13. Raw HTML - Page Source Code
  14. Instant Pages - Live Analysis
  15. Tasks Ready - Completed Tasks
  16. Force Stop - Terminate Tasks
  17. Error Handling & Status Codes
  18. Best Practices
-

# 1. API Architecture & Authentication

## Base URL

<https://api.dataforseo.com/v3>

## Authentication Method

HTTP Basic Authentication with Base64 encoding

## Setting Up Credentials

```
login="your_email@example.com"  
password="your_api_token"  
cred="$(printf $login :$password | base64)"
```

## Usage in Headers

Authorization: Basic \${cred}  
Content-Type: application/json

## Get Your Credentials

1. Log into your DataForSEO account
2. Navigate to <https://app.dataforseo.com/api-access>
3. Copy your email and API token
4. Use in cURL requests as shown above

---

# 2. Task POST - Initiate Crawling

**Endpoint:** POST /on\_page/task\_post

**Purpose:** Initiates a new website crawling task with customizable parameters.

**Rate Limit:** 2,000 requests per minute

## cURL Request

```
login="your_email@example.com"  
password="your_api_token"  
cred="$(printf $login :$password | base64)"  
  
curl --location --request POST "https://api.dataforseo.com/v3/on\_page/task\_post"  
--header "Authorization: Basic ${cred}"  
--header "Content-Type: application/json"  
--data-raw "[  
  {  
    \"target\": \"example.com\",  
    \"max_crawl_pages\": 100,  
    \"load_resources\": true,  
    \"enable_javascript\": true,  
    \"enable_browser_rendering\": true,  
    \"calculate_keyword_density\": false,  
  }  
]"
```

```

"store_raw_html": false,
"tag": "example_tag_001",
"pingback_url": "https://your-domain.com/webhook?id=$id&tag=$tag"
}
]"

```

## JSON Request Format

Parameter	Type	Description
target	string	Domain or URL to crawl (required)
max_crawl_pages	integer	Max pages to crawl 1-100,000 (required)
load_resources	boolean	Crawl images, scripts, stylesheets (optional)
enable_javascript	boolean	Execute JavaScript on pages (optional)
enable_browser_rendering	boolean	Measure Core Web Vitals (optional)
calculate_keyword_density	boolean	Include keyword analysis (optional)
store_raw_html	boolean	Store page HTML for retrieval (optional)
tag	string	Custom tag for tracking (optional)
pingback_url	string	Webhook URL for notifications (optional)

Table 1: Task POST Request Parameters

## JSON Response

```
{
"version": "0.1.20240101",
"status_code": 20000,
"status_message": "Ok.",
"time": "0.1234 sec.",
"cost": 0.05,
"tasks_count": 1,
"tasks_error_count": 0,
"tasks": [
{
"id": "12345678-1234-1234-1234-123456789012",
"status_code": 20000,

```

```

"status_message": "Task created successfully",
"time": "0.0234 sec.",
"cost": 0.05,
"result_count": 0,
"path": ["v3", "on_page", "task_post"],
"data": {
  "api": "on_page",
  "function": "task_post",
  "target": "example.com",
  "max_crawl_pages": 100,
  "tag": "example_tag_001"
},
"result": [
{
  "id": "12345678-1234-1234-1234-123456789012",
  "target": "example.com",
  "max_crawl_pages": 100,
  "load_resources": true,
  "enable_javascript": true,
  "enable_browser_rendering": true,
  "tag": "example_tag_001"
}
]
}
]
}

```

## Response Fields

- **id** - Unique task identifier for retrieving results
- **status\_code** - HTTP status (20000 = success)
- **status\_message** - Human-readable status message
- **cost** - API cost in credits for this request
- **result** - Array containing task details

## 3. Summary - Get Crawl Overview

**Endpoint:** GET /on\_page/summary?id=TASK\_ID

**Purpose:** Returns overall statistics and issue summary for a crawled website.

**Rate Limit:** 2,000 requests per minute

### cURL Request

```

login="your_email@example.com"
password="your_api_token"
cred=$(printf $login :$password | base64)

curl --location --request GET
"https://api.dataforseo.com/v3/on_page/summary?id=12345678-1234-1234-1234-12345678901"

```

2"  
--header "Authorization: Basic \${cred}"

## Query Parameters

Parameter	Type	Description
id	string	Task ID from Task POST response (required)

Table 2: Summary Request Parameters

## JSON Response

```
{  
  "version": "0.1.20240101",  
  "status_code": 20000,  
  "status_message": "Ok.",  
  "time": "0.2341 sec.",  
  "cost": 0.01,  
  "tasks_count": 1,  
  "tasks_error_count": 0,  
  "tasks": [  
    {  
      "id": "12345678-1234-1234-1234-123456789012",  
      "status_code": 20000,  
      "status_message": "Ok.",  
      "time": "0.0234 sec.",  
      "cost": 0.01,  
      "result_count": 1,  
      "path": ["v3", "on_page", "summary"],  
      "data": {  
        "api": "on_page",  
        "function": "summary",  
        "id": "12345678-1234-1234-1234-123456789012"  
      },  
      "result": [  
        {  
          "crawl_progress": 100,  
          "page_count": 245,  
          "issue_count": 23,  
          "error_count": 5,  
          "warnings_count": 18,  
          "crawl_duration": 342,  
          "crawl_start_time": "2024-01-15 10:30:45 +00:00",  
          "crawl_end_time": "2024-01-15 10:36:27 +00:00",  
          "pages_per_minute": 43.2,  
          "checks": [  
            {  
              "check_name": "meta_description_length",  
              "check_type": "meta",  
              "value": "The meta description length is 123, which is within the acceptable range of 100-150 characters."  
            }  
          ]  
        }  
      ]  
    }  
  ]  
}
```

```

    "pass": true,
    "severity": "info",
    "issues_count": 0
  },
  {
    "check_name": "broken_links",
    "check_type": "links",
    "pass": false,
    "severity": "error",
    "issues_count": 5
  },
  {
    "check_name": "duplicate_title",
    "check_type": "duplicate",
    "pass": false,
    "severity": "warning",
    "issues_count": 8
  }
]
}
]
]
}
]
}

```

## Response Fields

- **crawl\_progress** - Percentage completion (0-100)
  - **page\_count** - Total pages crawled
  - **issue\_count** - Total issues found
  - **error\_count** - Critical issues
  - **warnings\_count** - Warning-level issues
  - **checks** - Array of SEO check results
- 

## 4. Pages - Individual Page Data

**Endpoint:** GET /on\_page/pages?id=TASK\_ID&limit=100&offset=0

**Purpose:** Returns detailed metrics for each crawled page with all check results.

**Rate Limit:** 2,000 requests per minute

### cURL Request

```

login="your_email@example.com"
password="your_api_token"
cred=$(printf $login :$password | base64)

curl --location --request GET
"https://api.dataforseo.com/v3/on_page/pages?id=12345678-1234-1234-1234-123456789012&li
mit=100&offset=0"
--header "Authorization: Basic ${cred}"

```

## Query Parameters

Parameter	Type	Description
id	string	Task ID (required)
limit	integer	Results per response (default: 1000, max: 10000)
offset	integer	Number of results to skip for pagination

Table 3: Pages Request Parameters

## JSON Response (First 2 Pages)

```
{
  "version": "0.1.20240101",
  "status_code": 20000,
  "status_message": "Ok.",
  "time": "0.5234 sec.",
  "cost": 0.10,
  "tasks_count": 1,
  "tasks_error_count": 0,
  "tasks": [
    {
      "id": "12345678-1234-1234-1234-123456789012",
      "status_code": 20000,
      "status_message": "Ok.",
      "time": "0.4234 sec.",
      "cost": 0.10,
      "result_count": 245,
      "path": ["v3", "on_page", "pages"],
      "data": {
        "api": "on_page",
        "function": "pages",
        "id": "12345678-1234-1234-1234-123456789012",
        "limit": 100,
        "offset": 0
      },
      "result": [
        {
          "url": "https://example.com/",
          "status_code": 200,
          "title": "Example Domain - Home",
          "meta_description": "Welcome to Example.com - quality content",
          "h1": "Welcome to Our Website",
          "h2": ["About Us", "Services", "Contact"],
          "page_rank": 1,
          "page_depth": 0,
          "page_size": 45632,
        }
      ]
    }
  ]
}
```

```
"load_time": 1.23,
"incoming_links_count": 12,
"outgoing_links_count": 8,
"images_count": 5,
"resources_count": 34,
"checks": [
{
"check_name": "meta_description_length",
"value": 68,
"pass": true
},
{
"check_name": "title_length",
"value": 40,
"pass": true
},
{
"check_name": "h1_count",
"value": 1,
"pass": true
}
]
},
{
"url": "https://example.com/about",
"status_code": 200,
"title": "About Us - Example Domain",
"meta_description": "Learn more about Example.com",
"h1": "About Example.com",
"h2": ["Our Team", "Our Mission"],
"page_rank": 2,
"page_depth": 1,
"page_size": 38920,
"load_time": 0.95,
"incoming_links_count": 3,
"outgoing_links_count": 5,
"images_count": 8,
"resources_count": 28,
"checks": [
{
"check_name": "meta_description_length",
"value": 52,
"pass": true
},
{
"check_name": "title_length",
"value": 33,
"pass": true
}
]
}
```

```
]  
}  
]  
}
```

## Response Fields per Page

- **url** - Full page URL
- **status\_code** - HTTP status (200, 404, 301, etc.)
- **title** - Page title tag content
- **meta\_description** - Meta description content
- **h1** - Main heading
- **page\_rank** - Relevance ranking within site
- **page\_depth** - Distance from homepage
- **load\_time** - Page load time in seconds
- **incoming\_links\_count** - Internal links to this page
- **outgoing\_links\_count** - Links from this page
- **checks** - Individual SEO check results

---

## 5. Resources - Get All Resources

**Endpoint:** GET /on\_page/resources?id=TASK\_ID&limit=100&offset=0

**Purpose:** Returns list of all page resources (images, stylesheets, scripts, fonts).

**Rate Limit:** 2,000 requests per minute

### cURL Request

```
login="your_email@example.com"  
password="your_api_token"  
cred="$(printf login :${password} | base64)"  
  
curl --location --request GET  
"${https://api.dataforseo.com/v3/on_page/resources?id=12345678-1234-1234-1234-123456789012  
&limit=100&offset=0}"  
--header "Authorization: Basic ${cred}"
```

### Query Parameters

Parameter	Type	Description
id	string	Task ID (required)
limit	integer	Results per response (default: 1000, max: 10000)
offset	integer	Results to skip for pagination

Table 4: Resources Request Parameters

## JSON Response

```
{  
  "version": "0.1.20240101",  
  "status_code": 20000,  
  "status_message": "Ok.",  
  "time": "0.4123 sec.",  
  "cost": 0.05,  
  "tasks_count": 1,  
  "tasks_error_count": 0,  
  "tasks": [  
    {  
      "id": "12345678-1234-1234-1234-123456789012",  
      "status_code": 20000,  
      "status_message": "Ok.",  
      "time": "0.3234 sec.",  
      "cost": 0.05,  
      "result_count": 156,  
      "path": ["v3", "on_page", "resources"],  
      "data": {  
        "api": "on_page",  
        "function": "resources",  
        "id": "12345678-1234-1234-1234-123456789012"  
      },  
      "result": [  
        {  
          "url": "https://example.com/images/logo.png",  
          "type": "image",  
          "size": 15234,  
          "load_time": 0.23,  
          "status_code": 200,  
          "pages_count": 245  
        },  
        {  
          "url": "https://example.com/css/style.css",  
          "type": "stylesheet",  
          "size": 45123,  
          "load_time": 0.45,  
          "status_code": 200,  
          "pages_count": 245  
        },  
        {  
          "url": "https://example.com/js/main.js",  
          "type": "script",  
          "size": 34567,  
          "load_time": 0.34,  
          "status_code": 200,  
          "pages_count": 240  
        },  
        {  
          "url": "https://example.com/images/broken-image.jpg",  
          "type": "image",  
          "size": 12345,  
          "load_time": 0.56,  
          "status_code": 200,  
          "pages_count": 241  
        }  
      ]  
    }  
  ]  
}
```

```

    "type": "image",
    "size": 0,
    "load_time": null,
    "status_code": 404,
    "pages_count": 5
  }
]
}
]
}

```

## Response Fields per Resource

- **url** - Resource URL
  - **type** - Resource type (image, stylesheet, script, font)
  - **size** - File size in bytes
  - **load\_time** - Load time in seconds
  - **status\_code** - HTTP status code
  - **pages\_count** - Number of pages using this resource
- 

## 6. Links - Link Analysis

**Endpoint:** GET /on\_page/links?id=TASK\_ID&limit=100&offset=0

**Purpose:** Comprehensive analysis of internal and external links, including broken links.

**Rate Limit:** 2,000 requests per minute

### cURL Request

```

login="your_email@example.com"
password="your_api_token"
cred=$(printf $login :$password | base64)

curl --location --request GET
"https://api.dataforseo.com/v3/on_page/links?id=12345678-1234-1234-1234-123456789012&lim
it=100&offset=0"
--header "Authorization: Basic ${cred}"

```

### Query Parameters

Parameter	Type	Description
id	string	Task ID (required)
limit	integer	Results per response (default: 1000, max: 10000)
offset	integer	Results to skip for pagination

Table 5: Links Request Parameters

## JSON Response

```
{  
  "version": "0.1.20240101",  
  "status_code": 20000,  
  "status_message": "Ok.",  
  "time": "0.3456 sec.",  
  "cost": 0.08,  
  "tasks_count": 1,  
  "tasks_error_count": 0,  
  "tasks": [  
    {  
      "id": "12345678-1234-1234-1234-123456789012",  
      "status_code": 20000,  
      "status_message": "Ok.",  
      "time": "0.2456 sec.",  
      "cost": 0.08,  
      "result_count": 342,  
      "path": ["v3", "on_page", "links"],  
      "data": {  
        "api": "on_page",  
        "function": "links",  
        "id": "12345678-1234-1234-1234-123456789012"  
      },  
      "result": [  
        {  
          "from": "https://example.com/",  
          "to": "https://example.com/about",  
          "anchor_text": "About Us",  
          "link_type": "internal",  
          "is_nofollow": false,  
          "status_code": 200  
        },  
        {  
          "from": "https://example.com/",  
          "to": "https://external-site.com",  
          "anchor_text": "External Resource",  
          "link_type": "external",  
          "is_nofollow": false,  
          "status_code": 200  
        },  
        {  
          "from": "https://example.com/services",  
          "to": "https://example.com/nonexistent",  
          "anchor_text": "Service Details",  
          "link_type": "internal",  
          "is_nofollow": false,  
          "status_code": 404  
        }  
      ]  
    }  
  ]  
}
```

```
]  
}
```

## Response Fields per Link

- **from** - Source page URL
  - **to** - Destination URL
  - **anchor\_text** - Link text
  - **link\_type** - internal or external
  - **is\_nofollow** - Boolean nofollow status
  - **status\_code** - Target page HTTP status
- 

## 7. Duplicate Tags - Meta Tag Duplicates

**Endpoint:** GET /on\_page/duplicate\_tags?id=TASK\_ID

**Purpose:** Identifies pages with duplicate title tags and meta descriptions.

**Rate Limit:** 2,000 requests per minute

### cURL Request

```
login="your_email@example.com"  
password="your_api_token"  
cred="$(printf $login :$password | base64)"  
  
curl --location --request GET  
"https://api.dataforseo.com/v3/on_page/duplicate_tags?id=12345678-1234-1234-1234-1234567  
89012"  
--header "Authorization: Basic ${cred}"
```

### JSON Response

```
{  
  "version": "0.1.20240101",  
  "status_code": 20000,  
  "status_message": "Ok.",  
  "time": "0.2345 sec.",  
  "cost": 0.04,  
  "tasks_count": 1,  
  "tasks_error_count": 0,  
  "tasks": [  
    {  
      "id": "12345678-1234-1234-1234-123456789012",  
      "status_code": 20000,  
      "status_message": "Ok.",  
      "time": "0.1345 sec.",  
      "cost": 0.04,  
      "result_count": 12,  
      "path": ["v3", "on_page", "duplicate_tags"],  
      "data": {  
        "api": "on_page",
```

```

"function": "duplicate_tags",
"id": "12345678-1234-1234-1234-123456789012"
},
"result": [
{
"title": "Example - Quality Content",
"type": "title",
"pages": [
"https://example.com/page1",
"https://example.com/page2",
"https://example.com/page3"
],
"count": 3
},
{
"title": "High-quality content for needs",
"type": "meta_description",
"pages": [
"https://example.com/service1",
"https://example.com/service2"
],
"count": 2
}
]
}
]
}

```

## Response Fields per Duplicate

- **title** - Duplicate title or description text
- **type** - title or meta\_description
- **pages** - Array of URLs with this duplicate
- **count** - Number of pages with duplicate

## 8. Duplicate Content - Content Duplicates

**Endpoint:** GET /on\_page/duplicate\_content?id=TASK\_ID

**Purpose:** Identifies pages with similar or duplicate content across website.

**Rate Limit:** 2,000 requests per minute

### cURL Request

```

login="your\_email@example.com"
password="your\_api\_token"
cred=$(printf $login :$password | base64)

curl --location --request GET
"https://api.dataforseo.com/v3/on\_page/duplicate\_content?id=12345678-1234-1234-1234-1234

```

[56789012](#)

--header "Authorization: Basic \${cred}"

## JSON Response

```
{  
  "version": "0.1.20240101",  
  "status_code": 20000,  
  "status_message": "Ok.",  
  "time": "0.3123 sec.",  
  "cost": 0.06,  
  "tasks_count": 1,  
  "tasks_error_count": 0,  
  "tasks": [  
    {  
      "id": "12345678-1234-1234-1234-123456789012",  
      "status_code": 20000,  
      "status_message": "Ok.",  
      "time": "0.2123 sec.",  
      "cost": 0.06,  
      "result_count": 8,  
      "path": ["v3", "on_page", "duplicate_content"],  
      "data": {  
        "api": "on_page",  
        "function": "duplicate_content",  
        "id": "12345678-1234-1234-1234-123456789012"  
      },  
      "result": [  
        {  
          "page": "https://example.com/product-1",  
          "duplicate_page": "https://example.com/product-1-copy",  
          "similarity_percent": 98.5,  
          "canonical": null,  
          "recommendation": "Set canonical tag to primary page"  
        },  
        {  
          "page": "https://example.com/about",  
          "duplicate_page": "https://example.com/about-us",  
          "similarity_percent": 87.2,  
          "canonical": "https://example.com/about",  
          "recommendation": "Canonical tag already implemented"  
        }  
      ]  
    }  
  ]  
}
```

## Response Fields per Duplicate

- **page** - Primary page URL
  - **duplicate\_page** - Duplicate page URL
  - **similarity\_percent** - Content similarity percentage
  - **canonical** - Current canonical tag value
  - **recommendation** - SEO recommendation
- 

## 9. Redirect Chains - Redirect Issues

**Endpoint:** GET /on\_page/redirect\_chains?id=TASK\_ID

**Purpose:** Identifies problematic redirect chains that negatively impact SEO.

**Rate Limit:** 2,000 requests per minute

### cURL Request

```
login="your_email@example.com"
password="your_api_token"
cred=$(printf $login :$password | base64)

curl --location --request GET
"https://api.dataforseo.com/v3/on_page/redirect_chains?id=12345678-1234-1234-1234-123456
789012"
--header "Authorization: Basic ${cred}"
```

### JSON Response

```
{
  "version": "0.1.20240101",
  "status_code": 20000,
  "status_message": "Ok.",
  "time": "0.2891 sec.",
  "cost": 0.05,
  "tasks_count": 1,
  "tasks_error_count": 0,
  "tasks": [
    {
      "id": "12345678-1234-1234-1234-123456789012",
      "status_code": 20000,
      "status_message": "Ok.",
      "time": "0.1891 sec.",
      "cost": 0.05,
      "result_count": 4,
      "path": ["v3", "on_page", "redirect_chains"],
      "data": {
        "api": "on_page",
        "function": "redirect_chains",
        "id": "12345678-1234-1234-1234-123456789012"
      },
      "result": [
        ...
      ]
    }
  ]
}
```

```
{
  "chain": [
    {
      "url": "https://example.com/old-page",
      "type": "301",
      "status_code": 301
    },
    {
      "url": "https://example.com/middle-page",
      "type": "302",
      "status_code": 302
    },
    {
      "url": "https://example.com/final-page",
      "type": "200",
      "status_code": 200
    }
  ],
  "chain_length": 3,
  "issue_severity": "high",
  "recommendation": "Redirect directly to final destination"
},
{
  "chain": [
    {
      "url": "https://example.com/product-old",
      "type": "301",
      "status_code": 301
    },
    {
      "url": "https://example.com/product-new",
      "type": "200",
      "status_code": 200
    }
  ],
  "chain_length": 2,
  "issue_severity": "low",
  "recommendation": "Single redirect is acceptable"
}
]
```

## Response Fields

- **chain** - Array of redirect hops
  - **chain\_length** - Number of redirects in chain
  - **issue\_severity** - low, medium, or high
  - **recommendation** - SEO fix recommendation
-

## 10. Non-Indexable - Pages Blocked from Indexing

**Endpoint:** GET /on\_page/non\_indexable?id=TASK\_ID

**Purpose:** Lists pages blocked from search engine indexing.

**Rate Limit:** 2,000 requests per minute

### cURL Request

```
login="your_email@example.com"
password="your_api_token"
cred=$(printf $login :$password | base64)

curl --location --request GET
"https://api.dataforseo.com/v3/on_page/non_indexable?id=12345678-1234-1234-1234-123456789012"
--header "Authorization: Basic ${cred}"
```

### JSON Response

```
{
  "version": "0.1.20240101",
  "status_code": 20000,
  "status_message": "Ok.",
  "time": "0.3245 sec.",
  "cost": 0.05,
  "tasks_count": 1,
  "tasks_error_count": 0,
  "tasks": [
    {
      "id": "12345678-1234-1234-1234-123456789012",
      "status_code": 20000,
      "status_message": "Ok.",
      "time": "0.2245 sec.",
      "cost": 0.05,
      "result_count": 15,
      "path": ["v3", "on_page", "non_indexable"],
      "data": {
        "api": "on_page",
        "function": "non_indexable",
        "id": "12345678-1234-1234-1234-123456789012"
      },
      "result": [
        {
          "url": "https://example.com/admin",
          "reason": "robots_txt",
          "status_code": 403,
          "blocking_reason": "Blocked by robots.txt"
        },
        {
          "url": "https://example.com/temp-page",
        }
      ]
    }
  ]
}
```

```

"reason": "meta_robots_noindex",
"status_code": 200,
"blocking_reason": "Meta robots noindex tag present"
},
{
"url": "https://example.com/checkout",
"reason": "http_header",
"status_code": 200,
"blocking_reason": "X-Robots-Tag: noindex header"
},
{
"url": "https://example.com/private",
"reason": "access_denied",
"status_code": 401,
"blocking_reason": "Access denied (401 Unauthorized)"
}
]
}
]
}

```

## Response Fields per Non-Indexable Page

- **url** - Page URL
  - **reason** - Blocking reason code
  - **status\_code** - HTTP status code
  - **blocking\_reason** - Human-readable reason
- 

## 11. Waterfall - Page Speed & Core Web Vitals

**Endpoint:** GET /on\_page/waterfall?id=TASK\_ID&limit=50

**Purpose:** Provides detailed page speed metrics and Core Web Vitals data.

**Note:** Requires enable\_browser\_rendering: true in Task POST

**Rate Limit:** 2,000 requests per minute

### cURL Request

```

login="your_email@example.com"
password="your_api_token"
cred=$(printf %s :$password | base64)

curl --location --request GET
"https://api.dataforseo.com/v3/on_page/waterfall?id=12345678-1234-1234-1234-123456789012
&limit=50"
--header "Authorization: Basic ${cred}"

```

## Query Parameters

Parameter	Type	Description
id	string	Task ID (required)
limit	integer	Results per response (default: 1000, max: 10000)

Table 6: Waterfall Request Parameters

## JSON Response

```
{
  "version": "0.1.20240101",
  "status_code": 20000,
  "status_message": "Ok.",
  "time": "0.4567 sec.",
  "cost": 0.08,
  "tasks_count": 1,
  "tasks_error_count": 0,
  "tasks": [
    {
      "id": "12345678-1234-1234-1234-123456789012",
      "status_code": 20000,
      "status_message": "Ok.",
      "time": "0.3567 sec.",
      "cost": 0.08,
      "result_count": 245,
      "path": ["v3", "on_page", "waterfall"],
      "data": {
        "api": "on_page",
        "function": "waterfall",
        "id": "12345678-1234-1234-1234-123456789012"
      },
      "result": [
        {
          "url": "https://example.com/",
          "page_load_time": 1.234,
          "time_to_first_byte": 0.345,
          "first_contentful_paint": 0.456,
          "largest_contentful_paint": 1.023,
          "cumulative_layout_shift": 0.05,
          "total_blocking_time": 123,
          "core_web_vitals_assessment": {
            "lcp": {
              "value": 1.023,
              "status": "good"
            },
            "fid": {
              "value": 45,
            }
          }
        }
      ]
    }
  ]
}
```

```
"status": "good"
},
"cls": {
"value": 0.05,
"status": "good"
}
},
"resources": [
{
"url": "https://example.com/css/style.css",
"type": "stylesheet",
"size": 45123,
"duration": 0.234
},
{
"url": "https://example.com/js/main.js",
"type": "script",
"size": 34567,
"duration": 0.156
}
]
},
{
"url": "https://example.com/about",
"page_load_time": 0.987,
"time_to_first_byte": 0.234,
"first_contentful_paint": 0.345,
"largest_contentful_paint": 0.756,
"cumulative_layout_shift": 0.03,
"total_blocking_time": 87,
"core_web_vitals_assessment": {
"lcp": {
"value": 0.756,
"status": "good"
},
"fid": {
"value": 32,
"status": "good"
},
"cls": {
"value": 0.03,
"status": "good"
}
}
}
]
```

## Response Fields per Page

- **url** - Page URL
  - **page\_load\_time** - Total page load time (seconds)
  - **time\_to\_first\_byte** - TTFB metric
  - **first\_contentful\_paint** - FCP metric
  - **largest\_contentful\_paint** - LCP metric (optimal: <2.5s)
  - **cumulative\_layout\_shift** - CLS metric (optimal: <0.1)
  - **total\_blocking\_time** - TBT metric (milliseconds)
  - **core\_web\_vitals\_assessment** - Detailed CWV evaluation
  - **resources** - Array of resources with load times
- 

## 12. Keyword Density - Keyword Analysis

**Endpoint:** GET /on\_page/keyword\_density?id=TASK\_ID&limit=100

**Purpose:** Provides keyword density and frequency data for analyzed keywords.

**Note:** Requires calculate\_keyword\_density: true in Task POST (additional charges apply)

**Rate Limit:** 2,000 requests per minute

### cURL Request

```
login="your_email@example.com"
password="your_api_token"
cred=$(printf %s :$password | base64)

curl --location --request GET
"https://api.dataforseo.com/v3/on_page/keyword_density?id=12345678-1234-1234-1234-1234
6789012&limit=100"
--header "Authorization: Basic ${cred}"
```

### Query Parameters

Parameter	Type	Description
id	string	Task ID (required)
limit	integer	Results per response (default: 1000, max: 10000)

Table 7: Keyword Density Request Parameters

### JSON Response

```
{
"version": "0.1.20240101",
"status_code": 20000,
"status_message": "Ok.",
"time": "0.3234 sec.",
"cost": 0.06,
```

```
"tasks_count": 1,
"tasks_error_count": 0,
"tasks": [
{
"id": "12345678-1234-1234-1234-123456789012",
"status_code": 20000,
"status_message": "Ok.",
"time": "0.2234 sec.",
"cost": 0.06,
"result_count": 245,
"path": ["v3", "on_page", "keyword_density"],
"data": {
"api": "on_page",
"function": "keyword_density",
"id": "12345678-1234-1234-1234-123456789012"
},
"result": [
{
"url": "https://example.com/",
"keyword": "example",
"density": 2.34,
"frequency": 12,
"title": true,
"h1": true,
"h2": false,
"body": true,
"meta_description": false
},
{
"url": "https://example.com/about",
"keyword": "about us",
"density": 1.87,
"frequency": 8,
"title": true,
"h1": true,
"h2": true,
"body": true,
"meta_description": true
},
{
"url": "https://example.com/services",
"keyword": "services",
"density": 3.45,
"frequency": 15,
"title": true,
"h1": true,
"h2": true,
"body": true,
"meta_description": false
}
]
}
```

```
}
```

```
]
```

```
}
```

## Response Fields per Keyword

- **url** - Page URL
- **keyword** - Analyzed keyword term
- **density** - Keyword density percentage
- **frequency** - Number of times keyword appears
- **title** - Keyword in title tag (boolean)
- **h1** - Keyword in H1 (boolean)
- **h2** - Keyword in H2 (boolean)
- **body** - Keyword in body content (boolean)
- **meta\_description** - Keyword in meta description (boolean)

---

## 13. Raw HTML - Get Page Source Code

**Endpoint:** GET /on\_page/raw\_html?id=TASK\_ID

**Purpose:** Returns raw HTML source code of crawled pages.

**Note:** Requires store\_raw\_html: true in Task POST

**Rate Limit:** 2,000 requests per minute

### cURL Request

```
login="your_email@example.com"
password="your_api_token"
cred=$(printf %s :$password | base64)

curl --location --request GET
"https://api.dataforseo.com/v3/on_page/raw_html?id=12345678-1234-1234-1234-123456789012"
--header "Authorization: Basic ${cred}"
```

### JSON Response

```
{
  "version": "0.1.20240101",
  "status_code": 20000,
  "status_message": "Ok.",
  "time": "0.5234 sec.",
  "cost": 0.10,
  "tasks_count": 1,
  "tasks_error_count": 0,
  "tasks": [
    {
      "id": "12345678-1234-1234-1234-123456789012",
      "status_code": 20000,
      "status_message": "Ok.",
```

```
"time": "0.4234 sec.",
"cost": 0.10,
"result_count": 245,
"path": ["v3", "on_page", "raw_html"],
"data": {
  "api": "on_page",
  "function": "raw_html",
  "id": "12345678-1234-1234-1234-123456789012"
},
"result": [
{
  "url": "https://example.com/",
  "html": "\n<html lang=\"en\">\n\n<meta charset=\"UTF-8\">\n<meta name=\"description\" content=\"Welcome\">\n\n\n\nContent here...\n\n</html>"
}
]
```

## Welcome to Website

```
\n
Content here...
\n\n</html>
}
]
}
]
}
```

### Response Fields

- **url** - Page URL
- **html** - Complete HTML source code (escaped for JSON)

---

## 14. Instant Pages - Live Single Page Analysis

**Endpoint:** POST /on\_page/instant\_pages

**Purpose:** Performs immediate analysis of individual pages with live response.

**Benefits:** Results returned immediately without separate GET request

**Rate Limit:** 2,000 requests per minute, up to 20 pages per request

### cURL Request

```
login="your\_email@example.com"
password="your_api_token"
cred=$(printf %s :$password | base64)

curl --location --request POST
"https://api.dataforseo.com/v3/on\_page/instant\_pages"
--header "Authorization: Basic ${cred}"
--header "Content-Type: application/json"
```

```
--data-raw "[
{
"url": "https://example.com/",
"enable_javascript": true,
"enable_browser_rendering": true,
"load_resources": true
},
{
"url": "https://example.com/about",
"enable_javascript": true,
"enable_browser_rendering": true,
"load_resources": false
}
]"
```

## JSON Request Format

Parameter	Type	Description
url	string	Full URL to analyze (required)
enable_javascript	boolean	Execute JavaScript (optional)
enable_browser_rendering	boolean	Measure Core Web Vitals (optional)
load_resources	boolean	Crawl resources (optional)

Table 8: Instant Pages Request Parameters

## JSON Response

```
{
"version": "0.1.20240101",
"status_code": 20000,
"status_message": "Ok.",
"time": "2.3456 sec.",
"cost": 0.04,
"tasks_count": 2,
"tasks_error_count": 0,
"tasks": [
{
"id": "instant_12345678",
"status_code": 20000,
"status_message": "Ok.",
"time": "1.2345 sec.",
"cost": 0.02,
"result_count": 1,
"path": ["v3", "on_page", "instant_pages"],
"data": {
```

```
"url": "https://example.com/",
"enable_javascript": true,
"enable_browser_rendering": true,
"load_resources": true
},
"result": [
{
"url": "https://example.com/",
"status_code": 200,
"title": "Example Domain - Home",
"meta_description": "Welcome to Example.com",
"h1": "Welcome to Our Website",
"page_size": 45632,
"load_time": 1.23,
"checks": [
{
"check_name": "meta_description_length",
"value": 24,
"pass": true
},
{
"check_name": "title_length",
"value": 30,
"pass": true
},
{
"check_name": "ssl_certificate",
"value": true,
"pass": true
}
]
}
],
{
"id": "instant_87654321",
"status_code": 20000,
"status_message": "Ok.",
"time": "1.1111 sec.",
"cost": 0.02,
"result_count": 1,
"path": ["v3", "on_page", "instant_pages"],
"data": {
"url": "https://example.com/about",
"enable_javascript": true,
"enable_browser_rendering": true,
"load_resources": false
},
"result": [
{
"url": "https://example.com/about",
```

```
"status_code": 200,
"title": "About Us - Example Domain",
"meta_description": "Learn more about Example.com",
"h1": "About Example.com",
"page_size": 38920,
"load_time": 0.95,
"checks": [
{
"check_name": "meta_description_length",
"value": 28,
"pass": true
},
{
"check_name": "h1_count",
"value": 1,
"pass": true
}
]
}
]
}
]
}
]
```

---

## 15. Tasks Ready - Get Completed Tasks

**Endpoint:** GET /on\_page/tasks\_ready

**Purpose:** Returns list of completed tasks waiting for result retrieval.

**Rate Limit:** 2,000 requests per minute

### cURL Request

```
login="your\_email@example.com"
password="your\_api\_token"
cred=$(printf %s :$password | base64)

curl --location --request GET
"https://api.dataforseo.com/v3/on\_page/tasks\_ready"
--header "Authorization: Basic ${cred}"
```

### JSON Response

```
{
"version": "0.1.20240101",
"status_code": 20000,
"status_message": "Ok.",
"time": "0.1234 sec.",
"cost": 0,
"tasks_count": 1,
"tasks_error_count": 0,
```

```
"tasks": [
  {
    "id": "tasks_ready_endpoint",
    "status_code": 20000,
    "status_message": "Ok.",
    "time": "0.0234 sec.",
    "cost": 0,
    "result_count": 3,
    "path": ["v3", "on_page", "tasks_ready"],
    "data": {
      "api": "on_page",
      "function": "tasks_ready"
    },
    "result": [
      {
        "id": "12345678-1234-1234-1234-123456789012",
        "target": "example.com",
        "date_posted": "2024-01-15 10:30:45 +00:00",
        "tag": "example_tag_001"
      },
      {
        "id": "87654321-4321-4321-4321-210987654321",
        "target": "another-domain.com",
        "date_posted": "2024-01-15 11:15:23 +00:00",
        "tag": "another_tag"
      }
    ]
  }
]
```

## Response Fields per Ready Task

- **id** - Task ID
- **target** - Website domain that was crawled
- **date\_posted** - When crawl was initiated
- **tag** - Custom tag from Task POST

---

## 16. Force Stop - Terminate Running Task

**Endpoint:** POST /on\_page/force\_stop

**Purpose:** Stops an ongoing crawling task mid-process.

**Rate Limit:** 2,000 requests per minute

## cURL Request

```
login="your_email@example.com"
password="your_api_token"
cred=$(printf $login :$password | base64)

curl --location --request POST
"https://api.dataforseo.com/v3/on_page/force_stop"
--header "Authorization: Basic ${cred}"
--header "Content-Type: application/json"
--data-raw "[
{
"id": "12345678-1234-1234-1234-123456789012"
}
]"
```

## JSON Request Format

Parameter	Type	Description
id	string	Task ID to stop (required)

Table 9: Force Stop Request Parameters

## JSON Response

```
{
"version": "0.1.20240101",
"status_code": 20000,
"status_message": "Ok.",
"time": "0.0876 sec.",
"cost": 0,
"tasks_count": 1,
"tasks_error_count": 0,
"tasks": [
{
"id": "12345678-1234-1234-1234-123456789012",
"status_code": 20000,
"status_message": "Task stopped successfully",
"time": "0.0234 sec.",
"cost": 0,
"result_count": 1,
"path": ["v3", "on_page", "force_stop"],
"data": {
"api": "on_page",
"function": "force_stop",
"id": "12345678-1234-1234-1234-123456789012"
},
"result": [
{
"id": "12345678-1234-1234-1234-123456789012",
"status_code": 20000,
"status_message": "Task stopped successfully",
"time": "0.0234 sec.",
"cost": 0
}
]
}
```

```

    "stopped": true,
    "message": "Task stopped successfully"
}
]
}
]
}
}

```

## Response Fields

- **id** - Task ID
  - **stopped** - Boolean success status
  - **message** - Confirmation message
- 

## 17. Error Handling & Status Codes

### HTTP Status Codes

Code	Meaning	Solution
20000	Success	Request processed successfully
20001	Task in progress	Wait and retry request later
40001	Bad request	Validate all request parameters
40003	Not found	Verify task ID is correct
40004	Timeout	Retry request after delay
40005	Access denied	Check API credentials
40006	Rate limit exceeded	Reduce concurrent requests
40007	Insufficient funds	Top up account balance
50001	Server error	Retry after 60 seconds

Table 10: HTTP Status Codes and Solutions

### Example Error Response

```
{
  "version": "0.1.20240101",
  "status_code": 40001,
  "status_message": "Bad request",
  "time": "0.0123 sec.",
  "cost": 0,
  "tasks_count": 1,
  "tasks_error_count": 1,
```

```
"tasks": [
  {
    "id": "error_task_id",
    "status_code": 40001,
    "status_message": "Invalid target URL",
    "time": "0.0023 sec.",
    "cost": 0,
    "result_count": 0,
    "path": ["v3", "on_page", "task_post"],
    "data": {
      "api": "on_page",
      "function": "task_post",
      "target": "invalid-url"
    },
    "result": []
  }
]
```

---

## 18. Best Practices

### Authentication Security

#### ✓ GOOD: Use environment variables

```
login=DATAFORSEO_LOGINpassword =DATAFORSEO_PASSWORD
cred=$(printf $login :$password | base64)"
```

#### ✗ BAD: Hardcode credentials in scripts

```
login="user@example.com"
password="password123"
```

### Pagination for Large Datasets

#### Fetch results in batches

```
for offset in 0 100 200 300 400; do
curl -s "https://api.dataforseo.com/v3/on_page/pages?id={offset}"
--header "Authorization: Basic ${cred}" | jq '.tasks[0].result[].url'
done
```

## Monitor Crawl Progress

```
#!/bin/bash

task_id="12345678-1234-1234-1234-123456789012"

while true; do
response=(curl -s https://api.dataforseo.com/v3/on_page/summary?id=${task_id}" --header "Authorization: Basic ${cred}")

progress=$(echo $response | jq '.tasks[0].result[0].crawl_progress')

if [ "$progress" = "100" ]; then
echo "Crawl complete!"
break
fi

echo "Progress: ${progress}%"
sleep 30
done
```

## Use Webhooks for Notifications

```
{
"target": "example.com",
"max_crawl_pages": 1000,
"pingback_url": "https://your-server.com/webhook/onpage?id=tag"
}
```

## Cost Optimization Tips

- Only enable JavaScript for pages that need it
- Enable browser rendering only for specific audits
- Use selective resource loading
- Crawl incrementally rather than full-site repeatedly
- Cache and reuse crawl results across projects

---

## References

- Official Documentation: [https://docs.dataforseo.com/v3/on\\_page-overview/](https://docs.dataforseo.com/v3/on_page-overview/)
- API Access: <https://app.dataforseo.com/api-access>
- Support: <https://dataforseo.com/help-center>
- GitHub Examples: <https://github.com/dataforseo>

---

**Document Version:** 1.0

**Last Updated:** January 2024

**API Version:** v3

This comprehensive guide provides complete reference for all DataForSEO On-Page API endpoints with actual cURL requests and JSON responses for practical implementation.

