

Singapore Academy of Law

LawNet Legal Research API

Developer's Guide

Version 2.0 (For Trial)

RELEASE NOTES

| Version | Release Date | Remarks |
|---------|------------------|--|
| 2.0 | 16 December 2019 | Enhancement <ul style="list-style-type: none">• Legal Research Service Version 2.0<ul style="list-style-type: none">○ Added new service<ul style="list-style-type: none">▪ OAuth2 authentication○ Added new base url (v2)• Legal Research Service Version 1.0<ul style="list-style-type: none">○ v1 base url and endpoints to be deprecated in 30 days (End date: dd/mm/yyyy) |
| 1.0 | 1 Feburary 2019 | Initial Draft |

TABLE OF CONTENTS

| | | |
|----------|--|-----------|
| 1 | INTRODUCTION | 4 |
| 2 | LIBRARIES | 4 |
| 2.1 | cURL..... | 4 |
| 2.2 | C#..... | 4 |
| 2.3 | JAVA | 4 |
| 2.4 | POSTMAN | 4 |
| 3 | API REFERENCE | 5 |
| 3.1 | INTRODUCTION..... | 5 |
| 3.1.1 | <i>Base URL</i> | 5 |
| 3.1.2 | <i>Authentication</i> | 5 |
| 3.1.3 | <i>Data Access Level (Entitlement)</i> | 5 |
| 3.1.4 | <i>Response Codes</i> | 6 |
| 3.2 | AUTHORISATION API | 8 |
| 3.2.1 | <i>OAuth</i> | 8 |
| 3.2.1.1 | Request Fields (Header): | 8 |
| 3.2.1.2 | Request Fields (Body) | 8 |
| 3.2.1.3 | Response Fields | 8 |
| 3.2.1.4 | Sample Requests | 9 |
| 3.2.1.5 | Sample Response..... | 9 |
| 3.2.1.6 | Other Possible Responses (invalid) | 9 |
| 3.3 | LEGAL RESEARCH APIS..... | 8 |
| 3.3.1 | <i>Searching</i> | 10 |
| 3.3.1.1 | Request Fields (Header) | 10 |
| 3.3.1.2 | Request Fields (Body) | 10 |
| 3.3.1.3 | Response Fields | 12 |
| 3.3.1.4 | Sample Requests | 13 |
| 3.3.1.5 | Sample response | 14 |
| 3.3.2 | <i>Retrieving Content</i> | 17 |
| 3.3.2.1 | Request Fields (Header) | 17 |
| 3.3.2.2 | Request Fields (Body) | 17 |
| 3.3.2.3 | Response Fields | 17 |
| 3.3.2.4 | Sample Requests | 18 |
| 3.3.2.5 | Sample Response..... | 19 |
| 4 | FAQ..... | 20 |

1 INTRODUCTION

This document aims to provide an overview of the Singapore Academy of Law's ("SAL") LawNet Legal Research APIs and how you can best use them to create new products and services to enhance your applications with legal data. SAL's Legal Research APIs provide the ability to:

- Search Cases in LawNet
- Get Full Text of Cases in LawNet

To use our Legal Research APIs, an API Key and data entitlement will be created for you by Singapore Academy of Law to access the API end points. Application forms are available (<https://apiportal.sal.sg/portal>) and access will be subject to all relevant Terms and Conditions.

Where appropriate, access will only be granted to a sandbox environment which comprises of a subset of the production content. This would enable you to assess the APIs and data to see how they can integrate with your systems.

2 LIBRARIES

Our Legal Research APIs are built on HTTP. Our API is RESTful and it:

- Uses predictable, resource-oriented URLs.
- Uses built-in HTTP capabilities for passing parameters and authentication.
- Responds with standard HTTP response codes to indicate errors.
- Returns XML.

You may use your favorite or any suggested HTTP/REST library available for your programming language, to make HTTP calls to our Legal Research APIs.

Below are some common language-specific notes which you may use.

2.1 cURL

[cURL](#) is a popular command line tool to make HTTP requests with. It is a very simple but quite powerful tool. With it, you can send data using any HTTP method. You can send post data and query parameters and files in a very consistent and elegant way.

2.2 C#

For C# developers, you may use [RestSharp](#).

2.3 Java

Check out the [UniRest](#) REST client if Java is your weapon of choice.

You will also need the following dependencies, eg. org.json, httpclient 4.3.6, httpmime 4.3.6, httpasyncclient 4.0.2

2.4 Postman

[Postman](#) is a Chrome add-on and Mac application which is used to fire requests to an API. It is very lightweight and fast. Requests can be organised in groups, and tests can also be created with verifications for certain conditions on the response. It is possible to make different kinds of HTTP requests – GET, POST, PUT, PATCH and DELETE. It is also possible to add headers to the requests.

3 API REFERENCE

3.1 Introduction

This section provides you with information about the Legal Research APIs and sample scripts (refer to section 3.2) that will function. You are welcome to copy/paste and run the scripts to see the APIs in action.

3.1.1 Base URL

All Legal Research API URLs referenced in this documentation start with the following base part:

Trial:

| Base URL | Version | Expires |
|---|---------|------------|
| https://api.sal.sg/uat-legalresearch | v2 | N/A |
| https://test-legalresearch.api.sal.sg | v1 | dd/MM/yyyy |

3.1.2 Authentication

In Legal Research API, it will be based in OAuth2 using the grant type client-credentials (<https://www.oauth.com/oauth2-servers/access-tokens/client-credentials/>). Before you can use the SAL Legal Research API, you will first need to sign up for an account. Once approved, SAL will provide the requestor the necessary setup and configuration details required.

SAL will be providing the following information to the requestor:

- Client ID
- Client Secret
- API Key

The above information will then be used by the requestor as the access credentials to request for the access token through the authorization endpoint of the API Gateway Service.

Important: Keep your API key secret!

3.1.3 Data Access Level (Entitlement)

In addition to the APIkey, an excel sheet containing a list of content access codes will also be provided to you for accessing of data. The access codes for the content is required as a parameter when using services in sections 3.3.1 and 3.3.1.5.

Please refer to the sample Access control list below-

| Category | CategoryCode | Level2Category | Level2CategoryCode | Level3Category | Level3CategoryCode |
|----------|--------------|----------------|--------------------|-------------------------|--------------------|
| Cases | r1 | Judgments | r1c1 | Constitutional Tribunal | r1c1sc9 |
| Cases | r1 | Judgments | r1c1 | Court of Appeal | r1c1sc3 |
| Cases | r1 | Judgments | r1c1 | Court of Appeal (SICC) | r1c1sc12 |
| Cases | r1 | Judgments | r1c1 | Court of Three Judges | r1c1sc14 |
| Cases | r1 | Judgments | r1c1 | District Court | r1c1sc4 |
| Cases | r1 | Judgments | r1c1 | District Court (Family) | r1c1sc15 |

3.1.4 Response Codes

The following standard HTTP response codes.

| Code | Description |
|--------------------|---|
| 200 | Success - search results or full text of content returned. |
| 400 | Bad Request - Often this is because a required parameter is missing or a parameter's value is invalid. Note: this is returned in XML format. |
| 402 | Request Failed - Parameters valid but request failed |
| 403 | Forbidden – Invalid/Missing API Key |
| 404 | Not Found - The requested item doesn't exist |
| 500, 502, 503, 504 | Server Errors - something is wrong on our end |

The table below provides you with various scenarios and the expected responses, based on an entitlement below.

| Scenario | Description | Response Code | Category Code | Level2 Category Code | Level3 Category Code | Expected Response (xml) |
|----------|--------------------------------|---------------|---------------|----------------------|----------------------|--|
| 1 | Category Code is empty | 400 | | r1c1 | r1c1sc1 | <response> <status>error</status> <message>Category is a mandatory parameter. Please select a valid category and try again. </message> </response> |
| 2 | Level 2 category code is empty | 400 | r1 | | r1c1sc1 | <response> <status>error</status> <message>Level 2 category is a mandatory parameter. Please add a valid sub category and try again. </message> </response> |
| 3 | Level 3 category code is empty | 400 | r1 | r1c1 | | <response> <status>error</status> <message>Level 3 category is a mandatory parameter. Please add a valid level 3 category and try again. </message> </response> |

| | | | | | | |
|---|--|-----|--------|-------------|--|---|
| 4 | Level 1 to 3 category codes provided are correct | 200 | r1 | r1c1 | r1c1sc1 (To retrieve specific content) | Success: Results will be displayed. Refer to sections 3.3.1.5 and 3.3.2.5. |
| | | 200 | r1 | r1c2 | #r1c2 (To retrieve Judgment collection) | |
| | | 200 | r1, r3 | r1c1, r3c14 | r1c1sc1, r3c14sc1 | |
| 5 | Invalid Category code or category code that you are not entitled to access | 400 | r1, r2 | r1c1 | r1c1sc1 | <response> <status>error</status> <message>The category requested [r2] is invalid. The level 2 category requested [r1c2, r1c3] is invalid. The level 3 category requested [#r1c2, #r1c3] is invalid. </message> </response> |
| 6 | Level 3 category code that is not available in Legal Research | 400 | r1 | r1c1 | r1c1sc50 | <response> <status>error</status> <message>No matching Level 3 category found for the Level 2 category Judgments:r1c1. Please add a valid sub category and try again. The level 3 category requested [r1c1sc50] is invalid. </message> </response> |

3.2 Authorisation API

3.2.1 OAuth

This API allows you to request for token to access Legal Research APIs.

POST /oauth/token

3.2.1.1 Request Fields (Header):

| Field name | Description | Mandatory | Data type |
|---------------|--|-----------|-----------|
| Authorization | Basic <base64 encoded string> <base64 encoded string> - This is to pass the Client ID and Client secret values in the format client_id:client_secret and encode the combined string using base64 [Note the colon separating the two values] Encoding to base64 can be done using URL <ul style="list-style-type: none">https://www.base64encode.org | Yes | String |
| Content-Type | Set to the value <i>application/x-www-form-urlencoded</i> . | Yes | String |

3.2.1.2 Request Fields (Body)

| Field name | Description | Mandatory | Data type |
|---------------|--|-----------|-----------|
| grant_type | Set to the value <i>client_credentials</i> | Yes | String |
| client_id | Client ID that is given by SAL after successful client registration. | Yes | String |
| client_secret | Client Secret that is given by SAL after successful client registration. | Yes | String |

3.2.1.3 Response Fields

The response will be returned in JSON form with parsed parts.

| Field name | Description | Type |
|--------------|--|---------|
| access_token | Access token string issued by the authorization server | String |
| expires_in | Duration of time the access is granted for | Integer |
| token_type | Type of token this is, typically "bearer" | String |

3.2.1.4 Sample Requests

3.2.1.4.1 cURL

```
curl -X POST \
  https://api.sal.sg/uat-legalresearch/oauth/token \
  -H 'Authorization: Basic {{base64 encoded string}}' \
  -H 'Cache-Control: no-cache' \
  -H 'Content-Type: application/x-www-form-urlencoded' \
  -d 'grant_type=client_credentials&Client%20ID={{client_id}}&Client%20Secret={{client secret}}'
```

3.2.1.4.2 C# (RestSharp)

```
var client = new RestClient("https://api.sal.sg/uat-legalresearch/oauth/token");
var request = new RestRequest(Method.POST);
request.AddHeader("cache-control", "no-cache");
request.AddHeader("Content-Type", "application/x-www-form-urlencoded");
request.AddHeader("Authorization", "Basic {{base64 encoded string}}");
request.AddParameter("undefined",
"grant_type=client_credentials&Client%20ID={{client_id}}&Client%20Secret={{client secret}}",
ParameterType.RequestBody);
IRestResponse response = client.Execute(request);
```

3.2.1.4.3 Java (Unirest)

```
HttpResponse<String> response = Unirest.post("https://api.sal.sg/uat-legalresearch/oauth/token")
  .header("Content-Type", "application/x-www-form-urlencoded")
  .header("Authorization", "Basic {{base64 encoded string}}")
  .header("cache-control", "no-cache")
  .body("grant_type=client_credentials&Client%20ID={{client_id}}&Client%20Secret={{client secret}}")
  .asString();
```

3.2.1.5 Sample Response

```
{
  "access_token": "5d22ba70346a68000128b38f61a986f2c5164d04ba1c8b7f49e77d8c",
  "expires_in": 3600,
  "token_type": "bearer"
}
```

3.2.1.6 Other Possible Responses (invalid)

```
{
  "error": "unsupported_grant_type",
  "error_description": "The authorization grant type is not supported by the authorization server."
}
```

```
{
  "error": "invalid_request",
  "error_description": "The request is missing a required parameter, includes an invalid parameter value, includes a parameter more than once, or is otherwise malformed."
}
```

3.3 Legal Research APIs

3.3.1 Searching

This API allows you to search for cases within the LawNet Legal Research module.

| Endpoint | Version | Expires |
|------------------------|---------|------------|
| POST /v2/search | v2 | N/A |
| POST /v1-search/search | v1 | dd/MM/yyyy |

3.3.1.1 Request Fields (Header)

| Field name | Description | Mandatory | Data type | Remarks |
|---------------|--|-----------|-----------|-------------------|
| Authorization | Bearer <Access token string> Access token string given in response after successfully requesting access token (Refer to section Error! Reference source not found.) | Yes | String | Applicable for v2 |
| Content-Type | Set to the value <i>application/xwww-form-urlencoded</i> . | Yes | String | |
| x-api-key | API key allocated by SAL | Yes | String | |

3.3.1.2 Request Fields (Body)

| Field name | Description | Mandatory | Data type |
|------------|---|-----------|-----------|
| apikey | API Key allocated by SAL | Yes | String |
| cats | List of Content Groups. For multiple, please use comma separated values. | Yes | String |
| l2cats | List of Content Categories. For multiple, please use comma separated values. | Yes | String |

| | | | |
|------------------|--|-----|---------|
| l3cats | List of Content Subcategories. For multiple, please use comma separated values. | Yes | String |
| searchTerm | Search Term | Yes | String |
| page | Current page number. [Default :1] | No | Integer |
| maxperpage | Max per page in pagination [Default :20] | No | Integer |
| surroundingWords | Surrounding Words [Default :10] | No | Integer |
| orderBy | Ordering of search result using following value [title-asc, title-des, date-asc, date-des, relevance] [Default: relevance] | No | String |

3.3.1.3

Response Fields

The response will be returned in XML form with parsed parts.

These are the elements of the XML returned from a POST request to search cases.

| Elements | Description | Type |
|-----------------------|---|----------|
| searchResponse | Root Node of Search Results. | Single |
| searchStatistic | Child node of searchResponse, contain the search result stats eg. Total, Pagination count. | Single |
| total | Child node of searchStatistic, contain total number of search results returned | Single |
| paginationCount | Child node of searchStatistic, contain total number of paginations | Single |
| searchResults | Child node of searchResponse | Single |
| resultList | Child node of searchResults. This node holds multiple result of document content containing the search term. | Single |
| result | Child node of resultList, contain information on each search items | Multiple |
| documentId | Child node of result. This node holds the Document URL. Note: The value in this node will be used for retrieving content. | Single |
| document | Child node of result. This node holds title and citation details. | Single |
| Title | Child node of document. Title of the document. | Single |
| Citation | Child node of document. Citation of the document | Single |
| Format | Child node of result. Indication of document formats available (PDF and XML) – Yes/No. | Multiple |
| relevance | Child node of result. Relevance score of the document | Single |
| category | Child node of result. Category assigned to the document | Single |
| documentTooltipHolder | Child node of result. This node hold list of document metadata. | Multiple |
| casereference | Child node of result. Indicates the precedential value of the document according to the annotations. | Single |
| following | Child node of casereference. This is used to denote that the principle of law established in the instant case (or the <i>dictum</i> referred to) has been applied in subsequent cases. | Single |
| referring | Child node of casereference. | Single |

| Elements | Description | Type |
|----------------|---|----------|
| | This is used to describe cases which make only a general reference to the instant case. | |
| distinguishing | Child node of casereference. This is used where the instant case is not applied in subsequent cases due to some distinction in the facts or in the law. | Single |
| notfollowing | Child node of casereference. This is used where the court in subsequent cases has consciously refused to follow the instant case although potentially relevant. It implies that the instant case is wrong. | Single |
| overruling | Child node of casereference. This is used only where a higher court with the power to overrule has held the instant case to be wrong. | Single |
| snippets | Child node of result. This node holds multiple snippets of document content containing the search term. | Single |
| snippet | Child node of snippets. A snippet of document content containing the search term | Multiple |

3.3.1.4 Sample Requests

3.3.1.4.1 cURL

```
curl -X POST \
  https://api.sal.sg/uat-legalresearch/v2/search \
  -H 'Authorization: Bearer {{Access token string}}' \
  -H 'Content-Type: application/x-www-form-urlencoded' \
  -H 'cache-control: no-cache' \
  -H 'x-api-key: {{API KEY}}' \
  -d 'apikey={{API KEY}}&cats=r1&l2cats=r1c1%2Cr1c2%2Cr1c3&l3cats=%23r1c1%2C%23r1c2%2C%23r1c3&searchTerm=sample&page=1&maxperpage=15&orderBy=date-des&surroundingWords=5'
```

3.3.1.4.2 C# (RestSharp)

```
var client = new RestClient("https://api.sal.sg/uat-legalresearch/v2/search");
var request = new RestRequest(Method.POST);
```

```

request.AddHeader("cache-control", "no-cache");

request.AddHeader("x-api-key", "{{API KEY}}");

request.AddHeader("Content-Type", "application/x-www-form-urlencoded");

request.AddHeader("Authorization", "Bearer {{Access token string}}");

request.AddHeader("cache-control", "no-cache");

request.AddParameter("undefined", "apikey={{API KEY}}&cats=r1&l2cats=r1c1%2Cr1c2%2Cr1c3&l3cats=%23r1c1%2C%23r1c2%2C%23r1c3&searchTerm=sample&page=1&maxperpage=15&orderBy=date-des&surroundingWords=5", ParameterType.RequestBody);

IRestResponse response = client.Execute(request);

```

3.3.1.4.3 Java (Unirest)

```

HttpResponse<String> response = Unirest.post("https://api.sal.sg/uat-legalresearch/v2/search")

    .header("Content-Type", "application/x-www-form-urlencoded")

    .header("x-api-key", "{{API KEY}}")

    .header("Authorization", "Bearer {{Access token string}}")

    .header("cache-control", "no-cache")

    .body("apikey={{API KEY}}&cats=r1&l2cats=r1c1%2Cr1c2%2Cr1c3&l3cats=%23r1c1%2C%23r1c2%2C%23r1c3&searchTerm=sample&page=1&maxperpage=15&orderBy=date-des&surroundingWords=5")

    .asString();

```

3.3.1.5 Sample response

```

<searchResponse>
  <searchStatistic>
    <total>100</total>
  </searchStatistic>
  <searchResults>
    <resultList>
      <result>
        <documentId>/Sample Group/sample 1.xml</documentId>
        <document>
          <Title>
            Sample Title 1</Title>
          <Citation>[2019] sample 1</Citation>

```

```

</document>
<format>
  <xml>Yes</xml>
  <pdf>No</pdf>
</format>
<relevance>34</relevance>
<category>Sample Group</category>
<documentTooltipHolder>
  <court>Court</court>
  <corams>Luke Tan</corams>
  <date>18 April 2018</date>
  <caseno>sample-1111-2019 & Ors</caseno>
  <catchword>No catchword</catchword>
</documentTooltipHolder>
<casereference>
  <following>0</following>
  <referring>6</referring>
  <distinguishing>0</distinguishing>
  <notfollowing>0</notfollowing>
  <overruling>0</overruling>
</casereference>
<snippets>
  <snippet>...carefully laid trap by the
    <b>sample</b> content to...
  </snippet>
</snippets>
</result>
<result>
  <documentId>/Sample Group/sample 2.xml</documentId>
  <document>
    <Title>
Sample Title 2</Title>
    <Citation>[2017] Sample 2</Citation>
  </document>
  <format>
    <xml>Yes</xml>
    <pdf>No</pdf>
  </format>
  <relevance>58</relevance>
  <category>Sample Group</category>
  <documentTooltipHolder>
    <court>Court</court>
    <corams>Lim David</corams>
    <date>18 December 2017</date>
    <caseno>Sample 12344/2019, Sample No.1111/2019/01</caseno>
    <catchword>Damages, Tort</catchword>

```

```
</documentTooltipHolder>
<casereference>
  <following>0</following>
  <referring>0</referring>
  <distinguishing>0</distinguishing>
  <notfollowing>0</notfollowing>
  <overruling>0</overruling>
</casereference>
<snippets>
  <snippet>shouting “protest against the
    <b>sample</b> content!”; and
  </snippet>
  <snippet>protest against the
    <b>sample</b> content
  </snippet>
</snippets>
</result>
</resultList>
</searchResults>
</searchResponse>
```


3.3.2 Retrieving Content

This API allows you to retrieve content within the LawNet Legal Research.

| Endpoint | Version | Expires |
|--------------------------|---------|------------|
| POST /v2/getContent | v2 | N/A |
| POST /v1-content/content | v1 | dd/MM/yyyy |

3.3.2.1 Request Fields (Header)

| Field name | Description | Mandatory | Data type | Remarks |
|---------------|--|-----------|-----------|-------------------|
| Authorization | Bearer <Access token string> Access token string given in response after successfully requesting access token (Refer to section Error! Reference source not found.) | Yes | String | Applicable for v2 |
| Content-Type | Set to the value <i>application/xwww-form-urlencoded</i> . | Yes | String | |
| x-api-key | API key allocated by SAL | Yes | String | |

3.3.2.2 Request Fields (Body)

The following basic POST query parameters allow you to specify the cases you want:

| Field name | Description | Mandatory | Data type |
|------------|---|-----------|-----------|
| apikey | API Key Allocated | Yes | string |
| cats | List of Content Groups. For multiple. Comma separated values. | Yes | String |
| l2cats | List of Content categories. For multiple. Comma separated values. | Yes | String |
| l3cats | List of Content subcategories. For multiple. Comma separated values. | Yes | String |
| docUrl | Document URL is only available in the Search Response (refer to Section 3.3.1.5). Example: <documentId>/Sample Group/sample 1.xml</documentId> | Yes | String |
| Format | Format of content Note: The format XML or PDF will be used to retrieve the entitled document content. | No | String |

3.3.2.3 Response Fields

The response will be returned in XML form with parsed parts.

These are the elements of the XML returned from a POST request to get case content.

| Elements | Description | Type |
|-----------------|---|----------|
| searchResponse | Root node of content. | Single |
| pdf | Child node of searchResponse. This node holds details of a PDF. | Single |
| file | Child node of pdf. This node holds the file name of PDF. | Single |
| supportingFiles | Child node of searchResponse. This node holds multiple supporting files. (Example: images) | Single |
| supporting | Child node of supportingFiles. This node holds filename and binary. | Multiple |
| image | Child node of supporting. This node holds name of the file. | Single |
| title | Child node of image. Contains name of the file. | Single |
| binary | Child node of image/pdf. Contains base-16 string of the images or PDFs. Note: developers will be required to convert to physical files programmatically. | Single |
| Copyright | Child node of searchResponse. This value holds the copyright statement. | Single |
| mainContent | Child node of searchResponse. This node contains body content of the document. (XML/HTML+CSS) | Single |

3.3.2.4 Sample Requests

3.3.2.4.1 cURL

```
curl -X POST \
  https://api.sal.sg/uat-legalresearch/v2/getContent \
  -H 'Authorization: Bearer {{Access Token}}' \
  -H 'Content-Type: application/x-www-form-urlencoded' \
  -H 'cache-control: no-cache' \
  -H 'x-api-key: {{API KEY}}' \
  -d 'apikey={{API
KEY}}&cats=r1&l2cats=r1c1%2Cr1c2%2Cr1c3&l3cats=%23r1c1%2C%23r1c2%2C%23r1c3&docUrl=%2Fsample
group%2Fsample 1.xml&format=xml'
```

3.3.2.4.2 C# (RestSharp)

```
var client = new RestClient("https://api.sal.sg/uat-legalresearch/v2/getContent");
var request = new RestRequest(Method.POST);
request.AddHeader("cache-control", "no-cache");
request.AddHeader("x-api-key", "{{API KEY}}");
request.AddHeader("Content-Type", "application/x-www-form-urlencoded");
request.AddHeader("Authorization", "Bearer {{Access Token}}");
request.AddParameter("undefined","apikey={{API
KEY}}&cats=r1&l2cats=r1c1%2Cr1c2%2Cr1c3&l3cats=%23r1c1%2C%23r1c2%2C%23r1c3&docUrl=%2Fsample
group%2Fsample 1.xml&format=xml", ParameterType.RequestBody);
```

```
IRestResponse response = client.Execute(request);
```

3.3.2.4.3 Java (Unirest)

```
HttpResponse<String> response = Unirest.post("https://api.sal.sg/uat-legalresearch/v2/getContent")
    .header("Content-Type", "application/x-www-form-urlencoded")
    .header("x-api-key", "{{API KEY}}")
    .header("cache-control", "no-cache")
    .header("Authorization", "Bearer {{Access Token}}")
    .body("apikey={{API KEY}}&cats=r1&l2cats=r1c1%2Cr1c2%2Cr1c3&l3cats=%23r1c1%2C%23r1c2%2C%23r1c3&docUrl=%2Fsample group%2Fsample 1.xml&format=xml")
    .asString();
```

3.3.2.5 Sample Response

3.3.2.5.1 XML



response.xml

3.3.2.5.2 XML + Images



response.xml

3.3.2.5.3 HTML + CSS + supporting documents



response.xml

4 FAQ

Q1: Why I am receiving response “forbidden”?

A: This is because the API key used is invalid or empty or not found in header under key **x-api-key**.
`{ "message": "Forbidden" }.`

Q2: I'm not able to convert binary PDF and image to physical file. How can I do it?

A: You need to convert base-16 string to Byte Array and write the bytes onto your local machine programmatically.

Sample code in C#:

```
byte[] bytes;  
bytes = StringToByteArray(binaryString);  
System.IO.File.WriteAllBytes(@"C:\Projects\hello.pdf", bytes);
```

Q3: What is the API request limit set to my account?

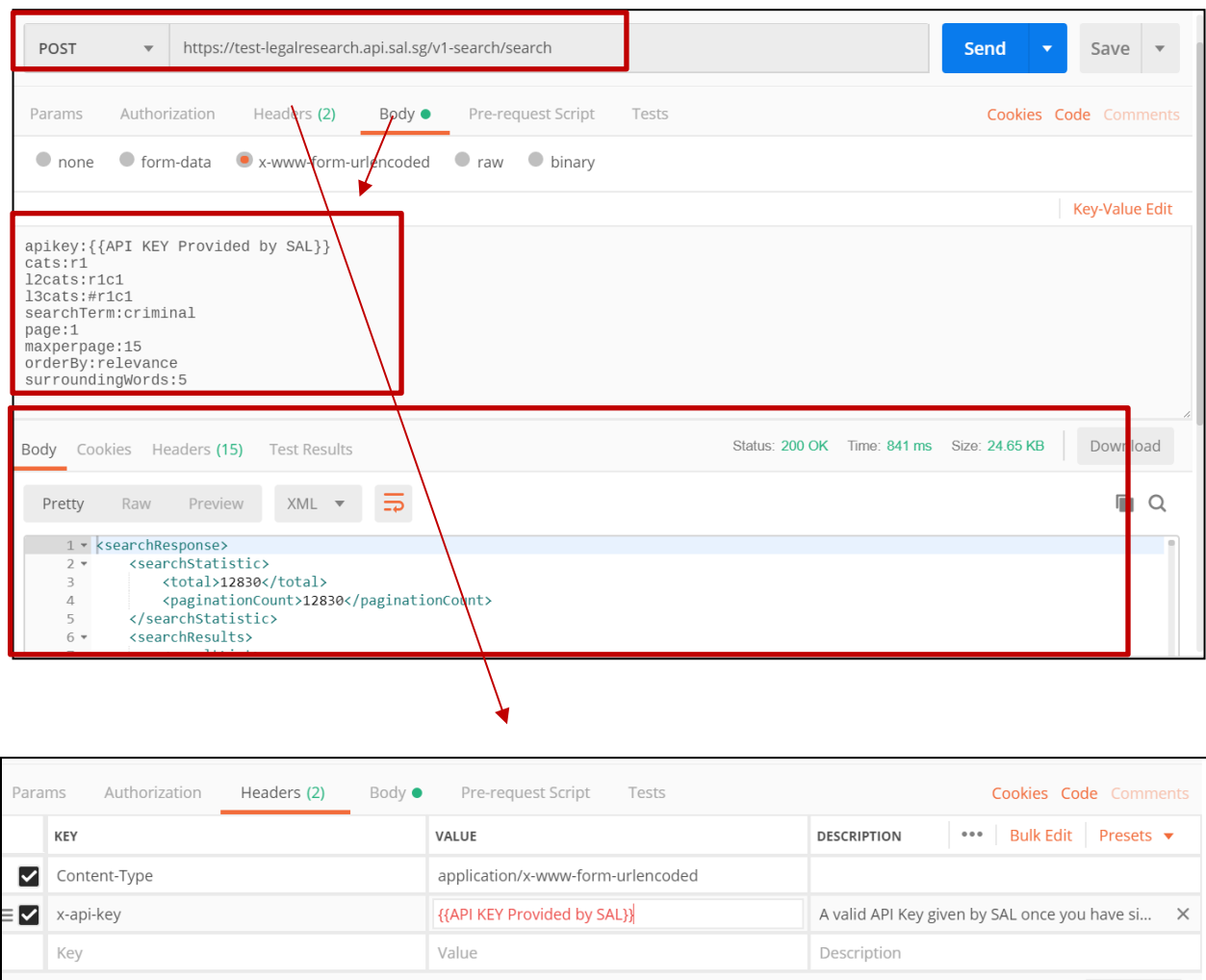
A: Each account is entitled to 100,000 per month API requests.

Q: How do I report an issue encountered?

A: Please provide the details on

1. API endpoint
2. Headers
3. API request
4. response/error

Sample screenshot from Postman



Q: What are the search operators available?

A:

| Search Operator | Explanation | Examples |
|----------------------------------|--|---|
| AND | Performs "all of the words" search, i.e. retrieves documents that contain all the entered keywords If multiple words are entered without using any operators, the AND operator is applied by default. | negligence AND causation negligence causation |
| OR | Performs "any of the words" search, i.e. retrieves documents that contain any of the entered keywords | forfeiture OR eviction |
| " " | Performs an "exact phrase" search, i.e. retrieves documents containing the exact phrase within the quotation marks. | "passing off" |
| - | Functions as a NOT operator, i.e. excludes documents that contain the entered keyword | -malaysia agent -insurance |
| NEAR/# NEAR | Functions as a proximity operator, i.e. retrieves documents that contain the entered keywords with # number of words between them. (If the NEAR operator is used without specifying the /# number, the default will be 10 words.) | breach NEAR/3 contract harassment NEAR act |
| * | Multi-character wildcard operator used to replace any number of characters in a word to search for variations of the word | appli* Moham*d |
| ? | Single character wildcard operator used to replace a single character in a word to search for spelling variations | debt?r organi?ation |
| NOT_IN | "not in" operator, i.e. retrieves documents containing the keyword but excludes documents containing the keyword in a phrase/context. | minority NOT_IN "minority shareholder" |
| () | Grouping operator, used to group together search terms which should be executed together | (doctor OR surgeon) AND negligence "specific performance" OR (damages NEAR/5 contract) |