LexisNexis® | *Regulatory Compliance*

API Integration Guide for SourceData

To be used in conjunction with the API Integration Guide

API Integration Guide v2.0

Endpoint reference for V2 of the RC API.

Overview

Environment Configuration

The API is available in two environments:
1) **Cert4** Environment (for integration testing):
   https://cdc4c-la-sp-rcapi.route53.lexis.com or https://cert-rcapi-plus.lexis.com
2) **Production** Environment:
   https://rcapi-plus.lexis.com

mandates

get request for mandates

Example

Title	**mandates**
Description	Get all mandates associated with a client's subscribed modules
URL	https://cdc4c-la-sp-rcapi.route53.lexis.com/f/Mandates/mandates
method	GET
Header	The new API supports multiple response data formats based on the `Accept` request header: • `application/xml` – returns data in XML format • `application/json` – returns data in JSON format • `text/event-stream` – used for Server-Sent Events (SSE) Supported output formats include: • JSON format • XML format • CSV format Additionally, it is recommended to include `**br**` (Brotli) as the first value in the `**Accept-Encoding**` request header to achieve better compression ratios.
Sample Request	https://cdc4c-la-sp-rcapi.route53.lexis.com/f/Mandates/mandates?customerId= urn:user:CB204485611&token=dafb5e5093a7159bdcbbb2dbfacc4f8a

Query Parameters

Parameter	Sample	Man-datory	Default	Comments
customerId	`customerId=urn:user:CB204 485611`	**Yes**		
token	`token=dafb5e5093a7159bdcb bb2dbfacc4f8a`	**Yes**		
size	`page[size]=10`	No	100	Maximum value limit is 2000
after	`page[after]=1`	No	1	
dateFrom dateEnd	`filter[dateFrom]=2016-03-24` `filter[dateEnd]=2016-03-31`	No		`YYYY-mm-dd` **Filter based on the `dateChanged` field in the response field**
recordType	`filter[recordType]=active`	No	active	**Options：** `active, archived` active: Only all active records (unarchived) are returned. archived: Only all archived records with an archive date in the future are returned.
Country	`filter[country]=Australia`	No		**The filter should be ignored when the `ParentId` parameter is not null.** **Options**: Australia, New Zealand, United Kingdom, International, Hong Kong, Singapore,

				United States of America, European Union, 日本, Canada, China
Level	`filter[level]=State`	No		**The filter should be ignored when the `ParentId` parameter is not null.** **Options**: Federal, Local, State, Territory, Blank
Module	`filter[module]= 24637`	No		**The filter should be ignored when the `ParentId` parameter is not null.**
format	`format=json`	No	json	**Options:** `json, xml, csv`
🆕 `state`	`filter[state]=ACT, NSW`	No		**Options**: ACT, NSW, NT, QLD **Field parameters support multiple values, concatenated with "," separator when there are multiple values**
🆕 <u>`ParentId`</u>	`filter[parentId]=767332,7 67333`	No		**Field parameters support multiple values, concatenated with "," separator when there are multiple values** **Mandate data follows a hierarchical tree structure. When parentId is not provided, returns the root nodes of the tree structure; when**

				parentId is provided, queries and returns child nodes under the specified parent node. To obtain the complete mandates structure, manual assembly is required.
NEW dateCreatedFrom dateCreatedEnd	`filter[dateCreatedFrom]=2016-03-24` `filter[dateCreatedEnd]=2016-03-31`	No		YYYY-mm-dd
NEW dateArchivedFrom dateArchivedEnd	`filter[dateArchivedFrom]=2016-03-24` `filter[dateArchivedEnd]=2016-03-31`	No		YYYY-mm-dd
NEW isGovernment	`filter[isGovernment]=true`	No		
NEW obligation-id	`filter[obligationid]=703908`	No		The filter should be ignored when the `ParentId` parameter is not null.
NEW regulators-id	`filter[regulatorid]=64274`	No		The filter should be ignored when the `ParentId` parameter is not null.
NEW mandateid	`filter[mandateid]=767332`	No		The filter should be ignored when the `ParentId` parameter is not null.
NEW order_by	`filter[orderby]=id`	No	id	**Options:** id, title, dateCreated, dateChanged, dateArchived, sortOrder
NEW order	`filter[order]=desc`	No	asc	**Options**: desc, asc

Response Demo

title	content

format=json|
 	
Accept=
application/js
on
 	
（When both format and accept parameters are set, the format value takes precedence）
	``json
	{
	    "data": [
	        {
	            "type": "mandate",
	            "id": "64014",
	            "attributes": {
	                "title": "Electronic Conveyancing
	(Adoption of National Law) Act 2013 (Vic)",
	                "sortOrder": null,
	                "dateCreated": "0001-01-01 00:00:00",
	                "dateChanged": "0001-01-01 00:00:00",
	                "dateArchived": "2033-02-27 00:00:00",
	                "isArchived": "false",
	                "country": "Australia",
	                "state": [
	                    "VIC"
	                ],
	                "level": [
	                    "State"
	                ],
	                "isGovernment": "true",
	                "url":
	"https://www.legislation.vic.gov.au/in-
	force/acts/electronic-conveyancing-adoption-national-
	law-act-2013"
	            },
	            "relationships": {
	                "parentMandate": {
	                    "data": []
	                },
	``

```
"mandates": {
    "data": [
        {
            "id": "64100",
            "type": "mandate"
        },
        {
            "id": "7161",
            "type": "mandate"
        }
    ]
},
"regulators": {
    "data": [
        {
            "id": "64100",
            "type": "regulator"
        }
    ]
},
"modules": {
    "data": [
        {
            "id": "7161",
            "type": "module"
        }
    ]
},
"obligations": {
    "data": [
        {
            "id": "8364",
            "type": "obligation"
        },
```

```json
                        {
                            "id": "7555",
                            "type": "obligation"
                        }
                    ]
                },
                "subObligations": {
                    "data": []
                },
                "alerts": {
                    "data": []
                }
            }
        }
    ],
    "links": {
        "self": "https://cdc7c-la-sp-
rcapi.route53.lexis.com/f/Mandates/mandates?token=08d
e4db5edee64db9836f5eb-f5be-4222-8618-
a814ae434c06&customerId=urn:user:CB205732495&format=J
son",
        "next": "https://cdc7c-la-sp-
rcapi.route53.lexis.com/f/Mandates/mandates?token=08d
e4db5edee64db9836f5eb-f5be-4222-8618-
a814ae434c06&customerId=urn%3auser%3aCB205732495&form
at=Json&page[after]=100"
    },
    "meta": {
        "execution_time": "0.828 secs",
        "cursor": {
            "current": "1",
            "prev": null,
            "next": "100",
            "count": 100
        }
    }
}
```

format=xml|

Accept= application/xml（When both format and accept parameters are set, the format value takes precedence）

```xml
<LexisNexisRI xmlns="http://www.lexisNexis.org"
version="2">
    <response>
        <data>
            <mandate>
                <type>mandate</type>
                <id>64014</id>
                <attributes>
                    <dateCreated>0001-01-01
00:00:00</dateCreated>
                    <dateChanged>0001-01-01
00:00:00</dateChanged>
                    <dateArchived>2033-02-27
00:00:00</dateArchived>
                    <isArchived>false</isArchived>
                    <country>Australia</country>
                    <state>VIC</state>
                    <level>State</level>
                    <isGovernment>true</isGovernment>
<url>https://www.legislation.vic.gov.au/in-
force/acts/electronic-conveyancing-adoption-national-
law-act-2013</url>
                    <title>
                        <![CDATA[Electronic Conveyancing
(Adoption of National Law) Act 2013 (Vic)]]>


                    </title>
                    <sortOrder />
                </attributes>
                <parentMandate />
                <mandates>
                    <mandate>
                        <id>755873</id>
                        <type>mandate</type>
                    </mandate>
```

```xml
            <mandate>
                <id>755877</id>
                <type>mandate</type>
            </mandate>
            <mandate>
                <id>65075</id>
                <type>mandate</type>
            </mandate>
            <mandate>
                <id>959220</id>
                <type>mandate</type>
            </mandate>
            <mandate>
                <id>211564</id>
                <type>mandate</type>
            </mandate>
            <mandate>
                <id>887631</id>
                <type>mandate</type>
            </mandate>
        </mandates>
        <regulators>
            <regulator>
                <id>65075</id>
                <type>regulator</type>
            </regulator>
        </regulators>
        <modules>
            <module>
                <id>211564</id>
                <type>module</type>
            </module>
            <module>
                <id>887631</id>
```

```
                    <type>module</type>
                </module>
            </modules>
            <obligations>
                <obligation>
                    <id>959220</id>
                    <type>obligation</type>
                </obligation>
            </obligations>
        </mandate>
    </data>
    <item>
        <self>https://cdc7c-la-sp-
rcapi.route53.lexis.com/f/Mandates/mandates?filter[co
untry]=Australia&token=08de4db5edee64db9836f5eb-f5be-
4222-8618-
a814ae434c06&amp;customerId=urn:user:CB205732495&amp;
format=xml</self>
        <next>https://cdc7c-la-sp-
rcapi.route53.lexis.com/f/Mandates/mandates?filter[co
untry]=Australia&token=08de4db5edee64db9836f5eb-f5be-
4222-8618-
a814ae434c06&amp;customerId=urn%3auser%3aCB205732495&
amp;format=xml&amp;page[after]=100</next>
    </item>
    <item>
        <execution_time>11.618
secs</execution_time>
        <cursor>
            <current>1</current>
            <prev />
            <next>100</next>
            <count>100</count>
        </cursor>
        <filter>
            <filter>
                <country>Australia</country>
            </filter>
```

<table>
<tr><td rowspan="2"></td><td>

```
            </filter>
          </item>
        </response>
  </LexisNexisRI>
```

</td></tr>
<tr><td>Accept=text/event-stream</td><td>When the data type is Server-Sent Events (SSE), the received data needs to be manually assembled by module. The returned data format supports XML, JSON, and you can specify the return data type by setting the format field.</td></tr>
</table>

post request for mandates `NEW`

Example

Title	**mandates**
Description	Get all mandates associated with a client's subscribed modules
URL	https://cdc4c-la-sp-rcapi.route53.lexis.com/f/Mandates/mandates
method	POST

Header	Different request parameter formats are supported based on the `Content-Type` request header when using the POST method: • `application/json` • `application/xml` • `application/x-www-form-urlencoded` The new API supports multiple response data formats based on the `Accept` request header: • `application/xml` – returns data in XML format • `application/json` – returns data in JSON format • `text/event-stream` – used for Server-Sent Events (SSE) Supported output formats include: • JSON format • XML format • CSV format Additionally, it is recommended to include `br` (Brotli) as the first value in the `Accept-Encoding` request header to achieve better compression ratios.

Request body in json（The content-type supports json,xml,and x-www-form-urlencoded formats. You can choose any one of them when passing parameters.）	**Header** Content-Type: application/json
	
	For parameter details, please refer to "**Input parameters**"
	
	For example:
	
	``json
	{
	  "token": "08ddf41b9a335821729dde1e-63d9-4ef9-894a-ce37cdf77a63",
	  "customerId": "urn:user:CB204255718",
	  "page[size]": "10",
	  "page[after]": "1"
	}
	``

Request body in x-www-form-urlencoded （The content-type supports json,xml,and x-www-form-urlencoded formats. You can choose any one of them when passing parameters.）	**Header** Content-Type: application/x-www-form-urlencoded
	For parameter details, please refer to "**Input parameters**"
	
	For example:
	`customerId=urn:user:CB204485611`
	`token=dafb5e5093a7159bdcbbb2dbfacc4f8a`
	
	Params Headers (2) Body •
	◯ none ◯ form-data ◉ x-www-form-urlencoded ◯ raw ◯ binary ◯ GraphQL
	
	Key Value Description
	☑ customerId urn:user:CB204255718
	☑ token 08de1ddb7044346c718d1daf-dac4-428...

Request body in xml （The content-type supports json,xml,and x-www-form-urlencoded formats. You can choose any one of them when passing parameters.）	**Header** Content-Type: application/xml
	
	For parameter details, please refer to "**Input parameters**"
	
	XML format parameter passing (Content-Type: application/xml) has special requirements and must comply with XML standard specifications
	
	XML request parameter mapping rules:
	
	• page[size] → page.size
	
	• token → token
	
	• filter[country] →filter.country
	
	For example:
	`<LexisNexisRI xmlns="http://www.lexisNexis.org"`
	`version="2">`
	`    <request>`
	`        <token>08de1aa8d0d337e94a82eee4-7851-4604-`
	`81d9-32db52ae12ee</token>`
	`        <customerId>urn:user:CB204255718</customerId>`
	`        <page.size>2</page.size>`
	`        <page.after>1</page.after>`
	`        <filter.country>Australia</filter.country>`
	`        <format>json</format>`
	`    </request>`
	`</LexisNexisRI>`

Input Parameters

Parameter	Sample	Man-datory	Default	Comments
customerId	customerId="urn:user:CB204485611"	**Yes**		
token	`token="dafb5e5093a7159bdc bbb2dbfacc4f8a"`	**Yes**		
size	`page[size]= "10"`	No	100	Maximum value limit is 2000
after	`page[after]= "1"`	No	1	
dateFrom dateEnd	`filter[dateFrom]= "2016-03-24"` `filter[dateEnd]= "2016-03-31"`	No		`YYYY-mm-dd` **Filter based on the `dateChanged` field in the response field**
recordType	`filter[recordType]= "active"`	No	active	**Options:** `active, archived` active: Only all active records (unarchived) are returned. archived: Only all archived records with an archive date in the future are returned.
Country	`filter[country]= "Australia"`	No		**The filter should be ignored when the `ParentId` parameter is not null.** **Options**: Australia, New Zealand,

				United Kingdom, International, Hong Kong, Singapore, United States of America, European Union, 日本, Canada, China
Level	`filter[level]= "State"`	No		**The filter should be ignored when the `ParentId` parameter is not null.** **Options：** Federal, Local, State, Territory or Blank
Module	`filter[module]="7146"`	No		**The filter should be ignored when the `ParentId` parameter is not null.**
format	`Format="json"`	No	json	**Options：** `json, xml, csv`
🆕 state	`filter[state]="ACT,NSW"`	No		**Options：** ACT, NSW, NT, QLD **Field parameters support multiple values, concatenated with "," separator when there are multiple values**
🆕 dateCreatedFrom dateCreatedEnd	`filter[dateCreatedFrom]=" 2016-03-24"` `filter[dateCreatedEnd]="2 016-03-24"`	No		YYYY-mm-dd
🆕 dateArchivedFrom dateArchivedEnd	`filter[dateArchivedFrom]= "2016-03-24"` `filter[dateArchivedEnd]=" 2016-03-24"`	No		YYYY-mm-dd

🆕 isGovernment	`filter[isGovernment]="true"`	No		
🆕 obligation-id	`filter[obligationid]="34940"`	No		**The filter should be ignored when the `ParentId` parameter is not null.**
🆕 MandateId	`filter[mandateid]="325031"`	No		**The filter should be ignored when the `ParentId` parameter is not null.**
🆕 ParentId	`filter[parentId]= "325030,325031"`	No		**Field parameters support multiple values, concatenated with "," separator when there are multiple values** **Mandate data follows a hierarchical tree structure. When parentId is not provided, returns the root nodes of the tree structure; when parentId is provided, queries and returns child nodes under the specified parent node. To obtain the complete mandates structure, manual assembly is required.**
🆕 regulators-id	`filter[regulatorid]= "64274"`	No		**The filter should be ignored when the `ParentId` parameter is not null.**
🆕 order_by	`filter[orderby]= "id"`	No	id	**Options**: id, title, dateCreated, dateChanged, dateArchived, sortOrder
🆕 order	`filter[order]= "desc"`	No	asc	**Options**: desc, asc

Response Demo

title	content

format=json|Accept= application/json
 	
（When both format and accept parameters are set, the format value takes precedence）
	<pre>{
	    "data": [
	        {
	            "type": "mandate",
	            "id": "64014",
	            "attributes": {
	                "title": "Electronic Conveyancing
	(Adoption of National Law) Act 2013 (Vic)",
	                "sortOrder": null,
	                "dateCreated": "0001-01-01 00:00:00",
	                "dateChanged": "0001-01-01 00:00:00",
	                "dateArchived": "2033-02-27 00:00:00",
	                "isArchived": "false",
	                "country": "Australia",
	                "state": [
	                    "VIC"
	                ],
	                "level": [
	                    "State"
	                ],
	                "isGovernment": "true",
	                "url":
	"https://www.legislation.vic.gov.au/in-
	force/acts/electronic-conveyancing-adoption-national-
	law-act-2013"
	            },
	            "relationships": {
	                "parentMandate": {
	                    "data": []
	                },
	                "mandates": {
	                    "data": [
	                        {
	                            "id": "64100",
	                            "type": "mandate"
	                        },
	                        {
	                            "id": "7161",
	                            "type": "mandate"
	                        }
	                    ]
	                },
	                ]
	                },
	                "modules": {</pre>

```
                    "data": [
                        {
                            "id": "7161",
                            "type": "module"
                        }
                    ]
                },
                "obligations": {
                    "data": [
                        {
                            "id": "8364",
                            "type": "obligation"
                        },
                        {
                            "id": "7555",
                            "type": "obligation"
                        }
                    ]
                },
                "subObligations": {
                    "data": []
                },
                "alerts": {
                    "data": []
                }
            }
        }
    ],
    "links": {
        "self": "https://cdc7c-la-sp-
rcapi.route53.lexis.com/f/Mandates/mandates?token=08de
4db5edee64db9836f5eb-f5be-4222-8618-
a814ae434c06&customerId=urn:user:CB205732495&format=Js
on",
        "next": "https://cdc7c-la-sp-
rcapi.route53.lexis.com/f/Mandates/mandates?token=08de
4db5edee64db9836f5eb-f5be-4222-8618-
a814ae434c06&customerId=urn%3auser%3aCB205732495&forma
t=Json&page[after]=100"
    },
    "meta": {
        "execution_time": "0.828 secs",
        "cursor": {
            "current": "1",
            "prev": null,
            "next": "100",
            "count": 100
        }
    }
}
```

format=xml|
 	
Accept=
application/x
ml（When
both format
and accept
parameters
are set, the
format value
takes
precedence）
	``xml
	<LexisNexisRI xmlns="http://www.lexisNexis.org"
	version="2">
	    <response>
	        <data>
	            <mandate>
	                <type>mandate</type>
	                <id>64014</id>
	                <attributes>
	                    <dateCreated>0001-01-01
	00:00:00</dateCreated>
	                    <dateChanged>0001-01-01
	00:00:00</dateChanged>
	                    <dateArchived>2033-02-27
	00:00:00</dateArchived>
	                    <isArchived>false</isArchived>
	                    <country>Australia</country>
	                    <state>VIC</state>
	                    <level>State</level>
	                    <isGovernment>true</isGovernment>
	                    <url>https://www.legislation.vic.gov.
	au/in-force/acts/electronic-conveyancing-adoption-
	national-law-act-2013</url>
	                    <title>
	                        <![CDATA[Electronic Conveyancing
	(Adoption of National Law) Act 2013 (Vic)]]>
	
	                    </title>
	                    <sortOrder />
	                </attributes>
	                <parentMandate />
	                <mandates>
	                    <mandate>
	                        <id>755873</id>
	                        <type>mandate</type>
	                    </mandate>
	                    <mandate>
	                        <id>755877</id>
	                        <type>mandate</type>
	                    </mandate>
	                    <mandate>
	                        <id>65075</id>
	                        <type>mandate</type>
	                    </mandate>
	                    <mandate>
	                        <id>959220</id>
	                        <type>mandate</type>
	                    </mandate>
	                    <mandate>
	                        <id>211564</id>
	``

```
                <type>mandate</type>
            </mandate>
            <mandate>
                <id>887631</id>
                <type>mandate</type>
            </mandate>
        </mandates>
        <regulators>
            <regulator>
                <id>65075</id>
                <type>regulator</type>
            </regulator>
        </regulators>
        <modules>
            <module>
                <id>211564</id>
                <type>module</type>
            </module>
            <module>
                <id>887631</id>
                <type>module</type>
            </module>
        </modules>
        <obligations>
            <obligation>
                <id>959220</id>
                <type>obligation</type>
            </obligation>
        </obligations>
    </mandate>
</data>
<item>
    <self>https://cdc7c-la-sp-
rcapi.route53.lexis.com/f/Mandates/mandates?filter[cou
ntry]=Australia&token=08de4db5edee64db9836f5eb-f5be-
4222-8618-
a814ae434c06&amp;customerId=urn:user:CB205732495&amp;f
ormat=xml</self>
    <next>https://cdc7c-la-sp-
rcapi.route53.lexis.com/f/Mandates/mandates?filter[cou
ntry]=Australia&token=08de4db5edee64db9836f5eb-f5be-
4222-8618-
a814ae434c06&amp;customerId=urn%3auser%3aCB205732495&a
mp;format=xml&amp;page[after]=100</next>
</item>
<item>
    <execution_time>11.618 secs</execution_time>
    <cursor>
        <current>1</current>
        <prev />
```

| | ```
                <next>100</next>
                <count>100</count>
            </cursor>
            <filter>
                <filter>
                    <country>Australia</country>
                </filter>
            </filter>
        </item>
    </response>
</LexisNexisRI>
``` |
| Accept=text/event-stream | When the data type is Server-Sent Events (SSE), the received data needs to be manually assembled by module. The returned data format supports XML, JSON, and you can specify the return data type by setting the format field. |

regulators

get request for regulators

Example

title	regulators
Description	Fetch regulators associated with a subscriber's modules.
URL	https://cdc4c-la-sp-rcapi.route53.lexis.com/f/Regulators/regulators
method	GET

	The new API supports multiple response data formats based on the `**Accept**` request header: • `application/xml` – returns data in XML format • `application/json` – returns data in JSON format • `text/event-stream` – used for Server-Sent Events (SSE) Supported output formats include: • JSON format • XML format • CSV format Additionally, it is recommended to include `**br**` (Brotli) as the first value in the `**Accept-Encoding**` request header to achieve better compression ratios.
Header	
sample Request	https://cdc4c-la-sp-rcapi.route53.lexis.com/f/Regulators/regulators?customerId=urn:user:CB204485611&token=dafb5e5093a7159bdcbbb2dbfacc4f8a

Query Parameters

Parameter	`Sample`	Man-datory	Default	Comments
customerId	`customerId=urn:user:CB204485611`	**Yes**		
token	`token= dafb5e5093a7159bdcbbb2db facc4f8a`	**Yes**		
size	`page[size]=10`	No	100	Maximum value limit is 2000
after	`page[after]=1`	No	1	For cursor-based pagination. The value will be obtained from the "next" parameter of a previous response.
dateFrom dateEnd	`filter[dateFrom]=2016-03-24`	No		YYYY-mm-dd

				Filter based on the `dateChanged` field in the response field
	`filter[dateEnd]=2016-03-31`			
recordType	`filter[recordType]=active`	No	active	**Options**: active, archived active: Only all active records (unarchived) are returned. archived: Only all archived records with an archive date in the future are returned.
Country	`filter[country]=Australia`	No		**Options**: Australia, New Zealand, United Kingdom, International, Hong Kong, Singapore, United States of America, European Union, 日本, Canada, China
Level	`filter[level]=State`	No		**Options**: Federal, Local, State, Territory or Blank
Module	`filter[module]=7146`	No		
format	`format=json`	No	`json`	**Options**: json, xml, csv
🆕 docid	`filter[docid]=64274`	No		
🆕 rc-jurisdiction	`filter[jurisdiction]=all`	No		
🆕 department	`filter[department]=true`	No		
🆕 dateCreatedForm dateCreatedEnd	filter[dateCreatedForm]=`2016-03-24` filter[dateCreatedEnd]=`2016-03-25`	No		YYYY-mm-dd
🆕 dateArchivedFrom dateArchivedEnd	filter[dateArchivedFrom]=`2016-03-24` `filter[dateArchivedEnd]=2016-03-25`	No		YYYY-mm-dd

🆕 Govern ment	`filter[isGovernment]` `=true`	No		
🆕 order_by	`filter[orderby]= id`	No	id	**Options**: id, name, dateCreated, dateArchived, dateChanged
🆕 order	`filter[order]= desc`	No	asc	**Options**: desc, asc

Response Demo

title	demo
format=json| Accept= application/js on （When both format and accept parameters are set, the format value takes precedence）	```{ "data": [{ "type": "regulator", "id": "63679", "attributes": { "name": "Australian Institute of Superannuation Trustees (Australia)", "acronym": "AIST", "department": "true", "dateCreated": "2019-07-18 21:02:16", "dateChanged": "2022-06-14 16:04:58", "dateArchived": "2029-07-19 00:00:00", "isArchived": "false", "country": "Australia", "state": ["Other"], "level": null, "isGovernment": "true", "url": "http://www.aist.asn.au/" }, "relationships": { "mandates": { "data": [{ "id": "559664", "type": "mandate" },```

```
                            {
                                "id": "559665",
                                "type": "mandate"
                            }
                        ]
                    }
                }
            },}
        ],
        "links": {
            "self": "https://cdc7c-la-sp-
    rcapi.route53.lexis.com/f/Regulators/regulators?token=
    08de4db40372cdb8fdf77ca4-3cb5-4a5d-8ffd-
    bb82421fe36e&customerId=urn:user:CB205732495&format=Js
    on",
            "next": "https://cdc7c-la-sp-
    rcapi.route53.lexis.com/f/Regulators/regulators?token=
    08de4db40372cdb8fdf77ca4-3cb5-4a5d-8ffd-
    bb82421fe36e&customerId=urn%3auser%3aCB205732495&forma
    t=Json&page[after]=100"
        },
        "meta": {
            "execution_time": "0.785 secs",
            "cursor": {
                "current": "1",
                "prev": null,
                "next": "100",
                "count": 100
            },
            "filter": {
                "filter": {
                    "country": "Australia"
                }
            }
        }
}
```

format=xml|
 	
Accept= application/xml（When both format and accept parameters are set, the format value takes precedence）
	``xml
	<LexisNexisRI xmlns="http://www.lexisNexis.org" version="2">
	    <response>
	        <data>
	            <regulator>
	                <type>regulator</type>
	                <id>63679</id>
	                <attributes>
	                    <dateCreated>2019-07-18 21:02:16</dateCreated>
	                    <dateChanged>2022-06-14 16:04:58</dateChanged>
	                    <dateArchived>2029-07-19 00:00:00</dateArchived>
	                    <isArchived>false</isArchived>
	                    <country>Australia</country>
	                    <state>Other</state>
	                    <isGovernment>true</isGovernment>
	                    <url>http://www.aist.asn.au/</url>
	                    <name>Australian Institute of Superannuation Trustees (Australia)</name>
	                    <acronym>AIST</acronym>
	                    <department>true</department>
	                </attributes>
	                <mandates>
	                    <mandate>
	                        <id>559664</id>
	                        <type>mandate</type>
	                    </mandate>
	                    <mandate>
	                        <id>559665</id>
	                        <type>mandate</type>
	                    </mandate>
	                </mandates>
	            </regulator>
	        </data>
	        <item>
	            <self>https://cdc7c-la-sp-rcapi.route53.lexis.com/f/Regulators/regulators?filter[country]=Australia&token=08de4db40372cdb8fdf77ca4-3cb5-4a5d-8ffd-bb82421fe36e&customerId=urn:user:CB205732495&format=Json</self>
	            <next>https://cdc7c-la-sp-rcapi.route53.lexis.com/f/Regulators/regulators?filter[country]=Australia&token=08de4db40372cdb8fdf77ca4-3cb5-4a5d-8ffd-bb82421fe36e&customerId=urn%3auser%3aCB205732495&format=Json&page[after]=100</next>
	        </item>
	``

```
                <item>
                    <execution_time>0.753 secs</execution_time>
                    <cursor>
                        <current>1</current>
                        <prev />
                        <next>100</next>
                        <count>100</count>
                    </cursor>
                    <filter>
                        <filter>
                            <country>Australia</country>
                        </filter>
                    </filter>
                </item>
            </response>
</LexisNexisRI>
```

Accept=text/event-stream	When the data type is Server-Sent Events (SSE), the received data needs to be manually assembled by module. The returned data format supports XML, JSON, and you can specify the return data type by setting the format field.

post request for regulators `NEW`

Example

title	regulators
Description	Fetch regulators associated with a subscriber's modules.
URL	https://cdc4c-la-sp-rcapi.route53.lexis.com/f/Regulators/regulators
method	POST

Header	Different request parameter formats are supported based on the `**Content-Type**` request header when using the POST method: • `application/json` • `application/xml` • `application/x-www-form-urlencoded` The new API supports multiple response data formats based on the `**Accept**` request header: • `application/xml` – returns data in XML format • `application/json` – returns data in JSON format • `text/event-stream` – used for Server-Sent Events (SSE) Supported output formats include: • JSON format • XML format • CSV format Additionally, it is recommended to include `**br**` (Brotli) as the first value in the `**Accept-Encoding**` request header to achieve better compression ratios.

Request body in json （The content-type supports json,xml,and x-www-form-urlencoded formats. You can choose any one of them when passing parameters. ）	**Header** Content-Type: application/json
	
	For parameter details, please refer to "**Input parameters**"
	
	For example:
	
	``
	{
	  "token": "08ddf1d8eae425323f9e8951-0e28-4919-9b3a-73aa50663aac",
	  "customerId": "urn:user:CB204485611",
	  "page[size]": "1",
	  "page[after]": "1"
	}
	``

Request body in x-www-form-urlencoded （The content-type supports json,xml,and x-www-form-urlencoded formats. You can choose any one of them when passing parameters.）	**Header** Content-Type: application/x-www-form-urlencoded
	For parameter details, please refer to "**Input parameters**"
	
	For example:
	`customerId=urn:user:CB204485611`
	`token=dafb5e5093a7159bdcbbb2dbfacc4f8a`
	
	Params Headers (2) Body ●
	○ none ○ form-data ● x-www-form-urlencoded ○ raw ○ binary ○ GraphQL
	
	Key │ Value │ Description
	☑ customerId │ urn:user:CB204255718 │
	☑ token │ 08de1ddb7044346c718d1daf-dac4-428... │

Request body in xml （The content-type supports json,xml,and x-www-form-urlencoded formats. You can choose any one of them when passing parameters.）	**Header** Content-Type: application/xml
	
	For parameter details, please refer to "**Input parameters**"
	
	XML format parameter passing (Content-Type: application/xml) has special requirements and must comply with XML standard specifications
	
	XML request parameter mapping rules:
	• page[size] → page.size
	
	• token → token
	
	• filter[country] →filter.country
	
	For example:
	`<LexisNexisRI xmlns="http://www.lexisNexis.org"`
	`version="2">`
	
	`    <request>`
	
	`        <token>08de1aa8d0d337e94a82eee4-7851-4604-81d9-`
	`32db52ae12ee</token>`
	
	`        <customerId>urn:user:CB204255718</customerId>`
	
	`        <page.size>2</page.size>`
	
	`        <page.after>1</page.after>`
	
	`        <filter.country>Australia</filter.country>`
	
	`    </request>`
	
	`</LexisNexisRI>`

Input Parameters

Parameter	Sample	Man-datory	Default	Comments
customerId	`customerId="urn:user:CB 204485611"`	**Yes**		
token	`token="dafb5e5093a7159b dcbbb2dbfacc4f8a"`	**Yes**		
size	`page[size]= "10"`	No	100	Maximum value limit is 2000
after	`page[after]="1"`	No	1	For cursor-based pagination. Value will be obtained from a prior response next parameter.
dateFrom dateEnd	`filter[dateFrom]":"2016 -03-24"` `filter[dateEnd]":"2016- 03-31"`	No		YYYY-mm-dd **Filter based on the `dateChanged` field in the response field**
recordType	`filter[recordType]="act ive"`	No	active	**Options**: active, archived active: Only all active records (unarchived) are returned. archived: Only all archived records with an archive date in the future are returned.
Country	`filter[country]= "Australia"`	No		**Options**: Australia, New Zealand, United Kingdom, International, Hong Kong, Singapore, United States of America, European Union, 日本, Canada, China

Level	filter[level]= "State"	No		**Options**: Federal, Local, State, Territory or Blank
Module	filter[module]= "7146"	No		
format	format= "json"	No	json	**Options**: json, xml, csv
[NEW] docid	filter[docid]= "64274"	No		
[NEW] rc-jurisdiction	filter[jurisdiction]= "all"	No		
[NEW] department	filter[department]= true	No		
[NEW] dateCreatedFrom dateCreatedEnd	filter[dateCreatedFrom]="2016-03-24" filter[dateCreatedEnd]= "2016-03-25"	No		YYYY-mm-dd
[NEW] dateArchivedFrom dateArchivedEnd	filter[dateArchivedFrom]="2016-03-24" "filter[dateArchivedEnd]="2016-03-25"	No		YYYY-mm-dd
[NEW] isGovernment	filter[isGovernment]= "true"	No		
[NEW] order_by	filter[orderby]= "id"	No	id	**Options**: id, name, dateCreated, dateArchived, dateChanged
[NEW] order	filter[order]= "desc"	No	asc	**Options**: desc, asc

Response Demo

title	demo

format=json|
Accept=
application/js
on

（When both
format and
accept
parameters are
set, the format
value takes
precedence）

```
{
    "data": [
        {
            "type": "regulator",
            "id": "63679",
            "attributes": {
                "name": "Australian Institute of
Superannuation Trustees (Australia)",
                "acronym": "AIST",
                "department": "true",
                "dateCreated": "2019-07-18 21:02:16",
                "dateChanged": "2022-06-14 16:04:58",
                "dateArchived": "2029-07-19 00:00:00",
                "isArchived": "false",
                "country": "Australia",
                "state": [
                    "Other"
                ],
                "level": null,
                "isGovernment": "true",
                "url": "http://www.aist.asn.au/"
            },
            "relationships": {
                "mandates": {
                    "data": [
                        {
                            "id": "559664",
                            "type": "mandate"
                        },
                        {
                            "id": "559665",
                            "type": "mandate"
                        }
                    ]
                }
            }
        },}
    ],
    "links": {
        "self": "https://cdc7c-la-sp-
rcapi.route53.lexis.com/f/Regulators/regulators?token=
08de4db40372cdb8fdf77ca4-3cb5-4a5d-8ffd-
```

```
bb82421fe36e&customerId=urn:user:CB205732495&format=Js
on",
        "next": "https://cdc7c-la-sp-
rcapi.route53.lexis.com/f/Regulators/regulators?token=
08de4db40372cdb8fdf77ca4-3cb5-4a5d-8ffd-
bb82421fe36e&customerId=urn%3auser%3aCB205732495&forma
t=Json&page[after]=100"
    },
    "meta": {
        "execution_time": "0.785 secs",
        "cursor": {
            "current": "1",
            "prev": null,
            "next": "100",
            "count": 100
        },
        "filter": {
            "filter": {
                "country": "Australia"
            }
        }
    }
}
```

format=xml|
 	
 Accept= application/x ml（When both format and accept parameters are set, the format value takes precedence）
	``xml
	<LexisNexisRI xmlns="http://www.lexisNexis.org"
	version="2">
	    <response>
	        <data>
	            <regulator>
	                <type>regulator</type>
	                <id>63679</id>
	                <attributes>
	                    <dateCreated>2019-07-18
	21:02:16</dateCreated>
	                    <dateChanged>2022-06-14
	16:04:58</dateChanged>
	                    <dateArchived>2029-07-19
	00:00:00</dateArchived>
	                    <isArchived>false</isArchived>
	                    <country>Australia</country>
	                    <state>Other</state>
	                    <isGovernment>true</isGovernment>
	                    <url>http://www.aist.asn.au/</url>
	                    <name>Australian Institute of
	Superannuation Trustees (Australia)</name>
	                    <acronym>AIST</acronym>
	                    <department>true</department>
	                </attributes>
	                <mandates>
	                    <mandate>
	                        <id>559664</id>
	                        <type>mandate</type>
	                    </mandate>
	                    <mandate>
	                        <id>559665</id>
	                        <type>mandate</type>
	                    </mandate>
	                </mandates>
	            </regulator>
	        </data>
	        <item>
	            <self>https://cdc7c-la-sp-
	rcapi.route53.lexis.com/f/Regulators/regulators?filter
	[country]=Australia&token=08de4db40372cdb8fdf77ca4-
	3cb5-4a5d-8ffd-
	bb82421fe36e&customerId=urn:user:CB205732495&f
	ormat=Json</self>
	``

```
                    <next>https://cdc7c-la-sp-
rcapi.route53.lexis.com/f/Regulators/regulators?filter
[country]=Australia&token=08de4db40372cdb8fdf77ca4-
3cb5-4a5d-8ffd-
bb82421fe36e&amp;customerId=urn%3auser%3aCB205732495&a
mp;format=Json&amp;page[after]=100</next>
            </item>
            <item>
                <execution_time>0.753 secs</execution_time>
                <cursor>
                    <current>1</current>
                    <prev />
                    <next>100</next>
                    <count>100</count>
                </cursor>
                <filter>
                    <filter>
                        <country>Australia</country>
                    </filter>
                </filter>
            </item>
        </response>
</LexisNexisRI>
```

Accept=text/event-stream	When the data type is Server-Sent Events (SSE), the received data needs to be manually assembled by module. The returned data format supports XML, JSON, and you can specify the return data type by setting the format field.

Pagination

Pagination is of the "cursor" type to maximize performance. The method for consuming the data is to continue requesting pages until all pages are exhausted for a given query. For JSON request formats, the response provides the value "next", which is used in the subsequent call by setting the parameter "after".

Alternatively, a Link HTTP header parameter is set to provide clients with the URL to use for the subsequent request.

title	content
Pagination response metadata	execution_time cursor.current - the pagination reference for the starting record

	cursor.prev - unused
	cursor.next - the pagination reference for the last record returned, for use in the next request as follows:
	page[after]=1
	cursor.count - the number of records returned
	Other input parameters used will also appear in the meta response

Pagination response demo	```json
	"meta":{
	    "executionTime": "0.386 secs",
	    "cursor": {
	      "current": "1",
	      "prev": null,
	      "next": "3",
	      "count": 50
	    }
	}
	``

Example metadata response with pagination parameters: page[size]=200&page[after]=400

```
- meta: {
    execution_time: "0.416 secs",
  - cursor: {
      current: "401",
      prev: null,
      next: "600",
      count: 200
    }
}
```

The subsequent request, substituting the value of next for us after parameter, is page[size]=200&page[after]=600

The final page will not contain a value for next, as the following example shows page[size]=200&page[after]=6400

```
- meta: {
    execution_time: "0.711 secs",
  - cursor: {
      current: "6401",
      prev: null,
      next: null,
      count: 200
    }
}
```

CSV Download

Instructions are as per above, with the only difference being format=csv.

Error code

error code	message
400	Invalid 'Accept' requested: application/ecmascript. Accept value is not equal to one of permitted types (application/json, application/xml, text/event-stream).
400	Invalid JSON format in request body.
400	Unsupported content type.
400	Root element must be <LexisNexisRI>.
400	Missing <request> element.
500	A textual month could not be found\nUnexpected data found.\nTrailing data.
500	An error occurred while validating query parameters.

Appendix

Mandates

Field name	Data Type	Description
id	integer	asset ID
attributes.title	varchar	mandate (compliance source) title
attributes.dateCreated	date	asset Creation Date
attributes.dateChanged	date	asset Last Published Dated

attributes.dateArchived	date	asset is archived but will continue to be available via the API until the date specified in the date Archived field.
attributes.isArchived	Boolean	TRUE or FALSE
attributes.country	varchar	varchar Regulator Country of Origin
attributes.state	list	State or Territory within the Regulator's Country of Origin
attributes.level	list	dropdown List - Federal, Local, State, Territory, or Blank
attributes.isGovernment	Boolean	TRUE or FALSE
attributes.url	varchar	URL to the Government Site for the Mandate (Compliance Source)
relationships.parentMandates	list	where the asset is a Sub Mandate (the Mandate's entire hierarchy down to the section level is included), the asset ID of its immediate parent.
relationships.mandates	list	The asset IDs of its immediate child mandates
relationships.regulators	list	Asset IDs of Related Regulators
relationships.obligations	list	Asset IDs of related obligations
relationships.subobligations	list	Asset IDs of related Sub Obligations
relationships.modules	list	Asset IDs of related modules This field will not be displayed in CSV download
relationships.alerts	list	Asset IDs of related alerts
attributes.sortOrder	integer	Sort order

Regulator

Field name	Data Type	Description
id	integer	asset ID
attributes.name	varchar	regulator name
attributes.dateCreated	date	asset Creation Date
attributes.dateChanged	date	asset Last Published Dated
attributes.dateArchived	date	Assets are archived but will continue to be available via the API until the date specified in the date Archived field.
attributes.isArchived	Boolean	TRUE or FALSE
attributes.country	varchar	varchar Regulator country of origin
attributes.state	list	State or Territory within the Regulator's Country of Origin
attributes.level	list	Dropdown List - Federal, Local, State, Territory, or Blank
attributes.isGovernment	Boolean	TRUE or FALSE
attributes.url	varchar	URL to the Regulator Site
attributes.department	Boolean	TRUE or FALSE
attributes.acronym	varchar	Acronym for the Regulator
relationships.mandates	list	Asset IDs of Related Mandates (Compliance Sources)