



**iRIS™**

---

*REST API*

Software Version 11.02.00

Manual Version DP1

# Contents

- Introduction..... 1
- REST API Prerequisites..... 1
  - iRIS™ System Requirements..... 1
    - iRIS™ Server Configuration Changes..... 1
    - iRIS™ Reserved Settings Changes..... 2
    - iRIS™ System Setup Changes..... 3
  - iRIS™ System Administration Requirements..... 3
- Search API Setup..... 3
  - Add a Query..... 4
  - Add a Subquery..... 7
  - Query / Subquery Hierarchy..... 10
  - Subquery Limitations..... 12
  - iRIS™ System Administrator Responsibility to Client Side..... 12
- Client-Side Requirements..... 13
  - REST API Endpoint for Data..... 13
  - REST API Endpoint for Forms..... 13

## Introduction

iMedRIS offers a REST (Representational State Transfer) API (Application Program Interface) to provide read-only access for querying iRIS™ data via SQL (Standardized Query Language). The iRIS™ REST API employs both server side and client side components.

On the iRIS™ server side, a system administrator defines SQL queries to be made available to the client side via internet-accessible REST API endpoints.

On the client side, the institution supplies a REST-compatible application or script to access the data sets returned by the iRIS™ system administrator's SQL queries. The client-side software can only access data exposed by the pre-defined SQL queries, but the software can be used to filter, group or format the incoming data as desired. Data access on the client side can be further limited by user login.

This manual describes configuration, functionality and operation of the iRIS™ REST API in full detail.

## REST API Prerequisites

iRIS™ configuration changes are required at the server level to enable access to the REST API.

After those changes are made iMedRIS personnel must update iRIS™ System Setup Reserved Settings to expose the Search API Setup page, where queries to be accessed via the REST API are defined.

An iRIS™ system administrator or designee must define the SQL queries to be accessed via the REST API and communicate the final query names and required parameters to whomever will be managing the client-side process.

Finally, the institution must provide its own client-side software to pass required parameters to the REST API, query the REST API endpoint, and receive data sets from the API.

## iRIS™ System Requirements

Changes must be made to the institution's iRIS™ server configuration and Reserved Settings under iRIS™ System Setup to enable access to the REST API.

### iRIS™ Server Configuration Changes

The web.xml file supplied with each iRIS™ installation sets certain environment variables. Individual system configurations can vary but the web.xml file is most often located in the ROOT > WEB-INF folder on the server where iRIS™ is installed.

iMedRIS uses the Java GlassFish servlet for its REST API implementation. By default, code that enables GlassFish is commented out in the web.xml file provided by iMedRIS because GlassFish consumes system resources and is only needed at institutions where external interface functionality is implemented.

In order to enable GlassFish, iMedRIS personnel or the institution's own IT department personnel must uncomment the web.xml code section shown below and save the file with changes.

```
<!-- following servlet required only for clients using system interface using REST API
<servlet>
  <servlet-name>System_Interface</servlet-name>
  <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
  <init-param>
    <param-name>javax.ws.rs.Application</param-name>
    <param-value>com.imedris.sysinterface.System_Application</param-value>
  </init-param>
  <init-param>
    <param-name>jersey.config.server.provider.packages</param-name>
    <param-value>com.imedris.sysinterface</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
  <servlet-name>System_Interface</servlet-name>
  <url-pattern>/sysinterface/*</url-pattern>
</servlet-mapping> -->
```

### iRIS™ Reserved Settings Changes

In the iRIS™ system under System Administration > System Configuration > **Reserved Settings**, the **system.use\_iRIS\_Data\_Feed\_API\_Setup** property must be set to “Yes”.

Property Name	Description	Value
system.use_feasibility_form	Description: This property indicates whether the Feasibility Form is used in this iRIS installation.	<input type="radio"/> Yes <input checked="" type="radio"/> No
system.use_gatekeeper_expiration	Description: This property, when set to Yes, allows the Gate Keeper review board to use an expiration date.	<input type="radio"/> Yes <input checked="" type="radio"/> No
system.use_grants.gov	Description: Use submission to grants.gov	<input type="radio"/> Yes <input checked="" type="radio"/> No
system.use_group_manager	Description: This property controls where managers can be assigned to user groups for the purpose of managing licenses and certifications.	<input type="radio"/> Yes <input checked="" type="radio"/> No
system.use_investigator_only_location	Description: This property is used in conjunction with Department/Location mode. When this property is Yes, the question Do you require a designee sub-investigator at each site will be asked in section 1.0 of the Study Shell	<input type="radio"/> Yes <input checked="" type="radio"/> No
system.use_IRB_identifying_codes	Description: This property indicates whether the iRIS installation has IRB identifying Board Codes turned on.	<input type="radio"/> Yes <input checked="" type="radio"/> No
system.use_IRIS_Data_Feed_API_Setup	Description: This property controls whether the system will use the Data Feed API.	<input type="radio"/> Yes <input checked="" type="radio"/> No
system.use_iris_lite	Description: This property will control whether the system is iRIS Lite.	<input type="radio"/> Yes <input checked="" type="radio"/> No
system.use_live_chat	Description: This property enables the live chat feature within iRIS.	<input type="radio"/> Yes <input checked="" type="radio"/> No
system.use_my_time_track	Description: This property indicates whether the iRIS installation has My Time Track Reporting turned on.	<input type="radio"/> Yes <input checked="" type="radio"/> No
system.use_negotiation_manager	Description: This property indicates whether the iRIS installation has Grants and Contracts post award turned on.	<input type="radio"/> Yes <input checked="" type="radio"/> No
system.use_post_award	Description: This property indicates whether the iRIS installation has Post Award Management turned on.	<input type="radio"/> Yes <input checked="" type="radio"/> No
system.use_project	Description: This property indicates whether proposal applications and processing is allowed.	<input type="radio"/> Yes <input checked="" type="radio"/> No
system.use_rb_ARC	Description: This property is used to indicate whether the iRIS installation has Animal Facility Management.	<input type="radio"/> Yes <input checked="" type="radio"/> No
system.use_rb_CI	Description: This property indicates whether the iRIS installation has a Conflict of Interest Assistant turned on.	<input type="radio"/> Yes <input checked="" type="radio"/> No
system.use_rb_COMP	Description: This property indicates whether the iRIS installation has Compliance Assistant turned on.	<input type="radio"/> Yes <input checked="" type="radio"/> No

This property is set to “No” by default and can only be changed by designated iMedRIS personnel.

### iRIS™ System Setup Changes

After the system.use\_IRIS\_Data\_Feed\_API\_Setup property is set to “Yes”, a new **Search API Setup** menu item appears in the **Other Configuration(s)** section on the **System Setup** tab of the **List Configuration and Maintenance** page under **System Administration**.

The screenshot shows the 'List Configuration and Maintenance' page with the following sections:

- System Administration**
  - Essential Setup
    - Definitions
      - Define the Institution(s)
      - Define the Department(s)
      - Define the Review Board(s)
    - Status Roles
      - Define System Level Role(s)
      - Define Study and Project Role(s)
      - Define Study Assistant Role Access
      - Define Project Assistant Role Access
    - Status Setup
      - Setup Project Status
      - Setup Study Status
    - News, Announcements and Broadcasting
      - System Announcements
      - System Broadcast
      - Business Associate (Competitor, Purchaser, Research Affiliate, Sponsor, Vendor, Support)
        - Sponsor Type List
        - Customer Vendor Type List
        - Business Associate Contact Category
        - Business Associate List
        - Grant and Contract Type List
        - Grant and Contract Funding Type List
  - Workflow Algorithm and Forms Datypes Setup
    - Define Additional Signoff Routing Roles
    - MEDDRA Code List
    - Configure System Linked List
    - Assign Review Board To Load Balance Algorithm
    - Load Balance Scoring Rules
  - Look & Feel
    - Setup System Labels
    - Setup System Images
    - Setup System Look and Feel
    - Setup Event Type Details
  - System Tutorials
    - Set-up Module Tutorials
    - User Profile Configuration(s)
      - Institutional Role Name List
      - Institutional Affiliations
      - Representative Capacity
      - Password Personal Questions
      - User Profile Status List
  - User Training Setup
    - Manage CITI Feeds
    - Define Training Courses
    - Define Training Groups
    - Define Training Assignments
    - Upload User
- List Configuration(s) for Boards**
  - IRB
    - Drug Configuration List
    - Device Configuration List
  - IBC
    - Agent List
    - Agent Risk List
    - Agent Biosafety
  - IACUC
    - Animal Species Configuration List
    - IACUC Drug Configuration List
    - Restraint Device Configuration List
    - Pain Classification
  - Project Assistant
    - Default Agency Validation Mapping
    - Define AOR Profile
    - Define NIH Activity Codes
    - Define NIH Departments
    - Define NIH Divisions
    - Define Organization Profile
    - External Personnel List
    - Grants.gov Forms
  - QA Assistant
    - Define Risk Factors
    - Define Research Trial Type
    - Define Intervention Type
    - Define Hazard Category Group
    - Define Impact Value
    - Define Risk Mitigation Status Modifier
    - Define Hazard Designation
  - Other Configuration(s)**
    - License Type List
    - Publication Type List
    - Search API Setup**

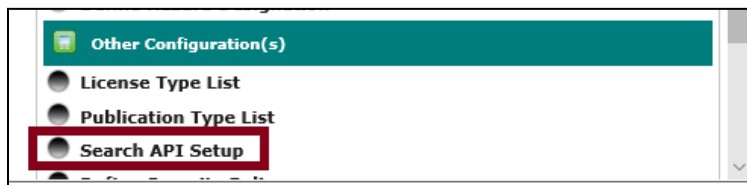
## iRIS™ System Administration Requirements

The REST API client-side software is only granted access to data sets returned by SQL queries that are defined in the iRIS™ system. Those queries must be created by an iRIS™ system administrator or designee.

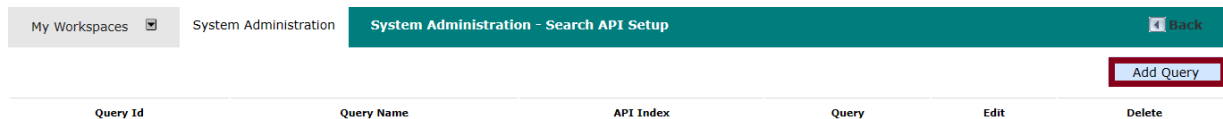
The person given responsibility for query definition and maintenance must have a thorough knowledge of iRIS™ system table definitions and intermediate to advanced SQL skills. That person should also employ a REST API testing tool (e.g. SoapUI®, Postman, etc.) to validate their SQL queries.

### Search API Setup

Click the **Search API Setup** link (System Administration > List Configuration and Maintenance > [Other Configuration(s) section]) to begin.



The **Search API Setup** page opens. This is where queries that define the data sets made available through the REST API are created and maintained.



Previously created queries are listed on the page. The first time the page is accessed the query list is blank, as there are no default queries defined.

### Add a Query

Click the **Add Query** button to add a SQL query. The **Add New Query** panel opens.

My Workspaces ▾ System Administration System Administration - Search API Setup 1 Back

Add New Query

Authentication Settings:

Allowed IP's:  0.0.0.0 will accept request from all IP addr's. Separate multiple ip's by comma(,)

Required Headers (system credentials):  Use pattern key1=pass1,key2=pass2. If no headers are informed, the system will employ **basic authentication** (iRIS user credentials).

API Index:

Query Name:

Insert query here:

X Note! Please place cursor where you would like to insert API MergeCode and then use 'Insert API MergeCode' button.

The **Authentication Settings** section contains fields for **Allowed IP's** and **Required Headers (system credentials)**.

The **Allowed IP's** field defaults to 0.0.0.0, indicating that any IP (Internet Protocol) address will be allowed to access the endpoint for this query. To limit access to this query by IP address, enter the IP address(es) to which access is granted. Separate multiple IP addresses with a comma ( , ).

The **Required Headers (system credentials)** field is used for login authentication to the endpoint for this query.

If the field is left blank client-side login will be authenticated with the iRIS™ login credentials of the account running the client-side process. The result set will be limited to records and forms to which the iRIS™ account has access within the iRIS™ system. This is the default, basic authentication method.

To authenticate with explicit credentials, enter desired variables to serve as receptacles for login key(s) and password(s) that will be passed from the client side, as demonstrated in the help text to the right of the field: **Use pattern key1=pass1,key2=pass2**. This is a useful option for sharing endpoint data with non-users of iRIS™.

An incoming "key" value corresponds to a username or user id and an incoming "pass" value corresponds to the associated password for the user. To allow for explicit authentication by multiple client-side users, enter multiple key/pass pairs separated with a comma ( , ). Variable names chosen for key/pass pairs must be communicated to whomever will be managing the client-side process.

The **API Index** field is an internal tracking number assigned to each query by the iRIS™ system administrator. The **API Index** assigned to each query must be unique, and dictates the order in which queries are listed on the **Search API Setup** page.

The **Query Name** field is used to assign a name to the query. The query name must be communicated to whomever will be managing the client-side process.

Enter the desired top-level SQL query in the **Insert query here:** text field.

In the example below the form is filled, but the SQL query is incomplete because it requires a parameter input from the client side.

My Workspaces System Administration System Administration - Search API Setup Back

Add New Query

Authentication Settings:

Allowed IPs: 0.0.0.0 0.0.0.0 will accept request from all IP addr. Separate multiple ip's by comma(,)

Required Headers (system credentials): u1=v1, u2=v2 Use pattern key1=pass1,key2=pass2. If no headers are informed, the system will employ basic authentication (IRIS user credentials).

API Index: 50

Query Name: study details

Insert query here:

```
select top10 s.study_title, s.study_status, S.SYSTEM_STUDY_ID
from study S
where s.id_created = [redacted]
order by s.system_study_id desc
```

Add Sub Query

\* Note! Please place cursor where you would like to insert API MergeCode and then use 'Insert API MergeCode' button.

Insert API MergeCode

Cancel Save

Click the **Insert API MergeCode** button to create and add a parameter variable field to the query. A new section is exposed at the bottom of the query text field.

Query Name: study details

Insert query here:

```
select top10 s.study_title, s.study_status, S.SYSTEM_STUDY_ID
from study S
where s.id_created = [redacted]
order by s.system_study_id desc
```

Add Sub Query

\* Note! Please place cursor where you would like to insert API MergeCode and then use 'Insert API MergeCode' button.

Data Type: Single Integer/Number  
Multiple Integers/Numbers  
Single String  
Multiple String  
Date  
Multiple Date  
Study Access

MergeCode: [ ] Add

Cancel Save

Select the applicable data type for the parameter by clicking on it in the **Data Type:** dropdown. This data type must be communicated to whomever will be managing the client-side process.

Enter the desired parameter variable name in the **MergeCode:** field. This name must be communicated to whomever will be managing the client-side process.

Click in the query text field to place the cursor where the **MergeCode** is to be inserted, then click the **Add** button. In the example shown below a **MergeCode** parameter variable named "INT\_user\_ID" of **Data Type** "Single Integer/Number" is to be inserted where the cursor is currently located in the query text field.

My Workspaces ▾ System Administration **System Administration - Search API Setup** [Back](#)

**Add New Query**

Authentication Settings:

Allowed IP's:  **0.0.0.0 will accept request from all IP addr's. Separate multiple ip's by comma(,)**

Required Headers (system credentials):  **Use pattern key1=pass1,key2=pass2. If no headers are informed, the system will employ basic authentication (IRIS user credentials).**

API Index:

Query Name:

Insert query here:

```
select top10 S.study_Title, s.study_status, S.SYSTEM_STUDY_ID
from study S
where s.id_created = 1
order by s.system_study_id desc
```

**x Note!** Please place cursor where you would like to insert API MergeCode and then use 'Insert API MergeCode' button.

Data Type:  MergeCode:  **Add**

[Cancel](#) [Save](#)

In the example below the **Add** button was clicked and the **MergeCode** "INT\_user\_ID" was added to the query.

*Note that the curly braces surrounding the inserted MergeCode are automatically added by the system, and were not included in the name entered into the **MergeCode** field.*

My Workspaces ▾ System Administration **System Administration - Search API Setup** [Back](#)

Query Name:

Insert query here:

```
select top10 S.study_Title, s.study_status, S.SYSTEM_STUDY_ID
from study S
where s.id_created = {{INT_user_ID}}
order by s.system_study_id desc
```

[Add Sub Query](#)

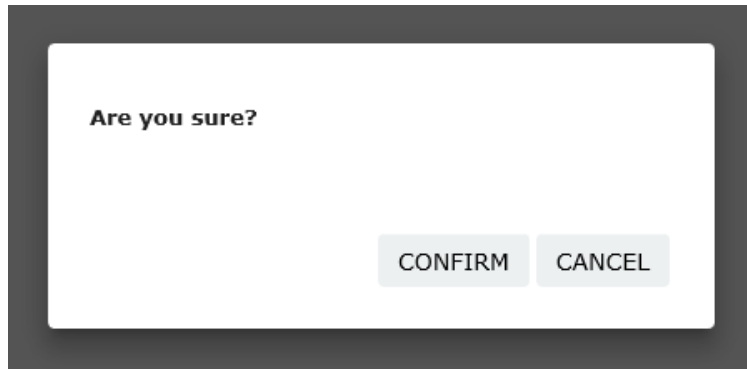
Click the **Save** button at the bottom of the page (shown in previous screenshots) to save the query.

The new query is now listed on the **Search API Setup** page.

Query Id	Query Name	API Index	Query	Edit	Delete
	study details	50	select top10 S.study_Title, s.study_status, S.SYSTEM_STUDY_ID from study S where s.id_created = {{INT_user_ID}} order by s.system_study_id desc	<a href="#">Edit</a>	<a href="#">Delete</a>

Click the **Edit** button to open the query for editing.

Click the **Delete** button to delete the query. A confirmation popup window appears.

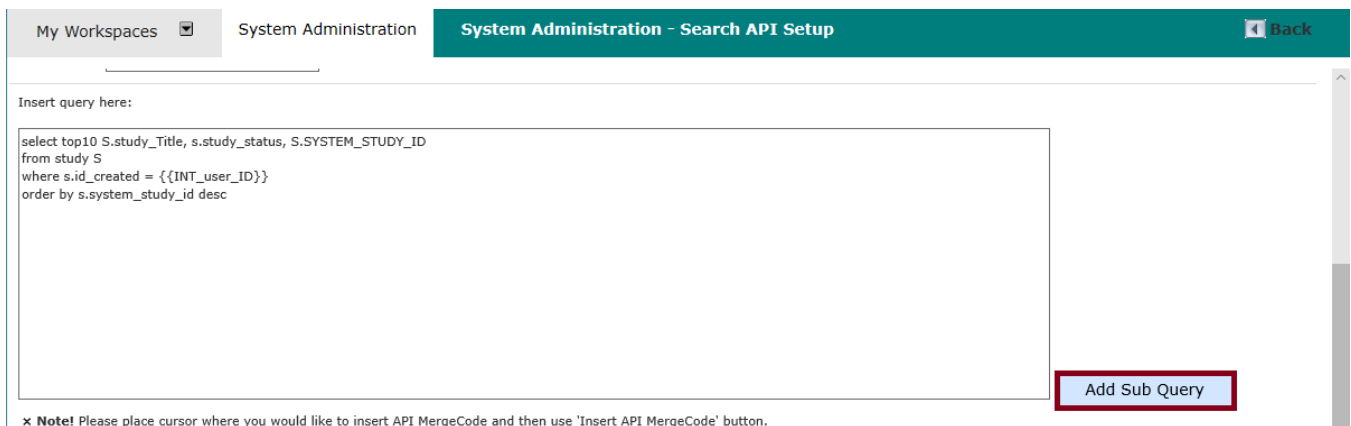


Click the **CONFIRM** button in the popup to confirm deletion.

Click the **CANCEL** button in the popup to return to the **Search API Setup** page without deleting the query.

### Add a Subquery

Subqueries can be added to refine or expand the data set returned by the first, top-level query. With a top-level query entered or opened for editing, click the **Add Sub Query** button to add a subquery.



A subquery form is opened.

My Workspaces ▾ System Administration **System Administration - Search API Setup** [Back](#)

Query Name:

Insert query here:

```
select top10 S.study_Title, s.study_status, S.SYSTEM_STUDY_ID
from study S
where s.id_created = {{INT_user_ID}}
order by s.system_study_id desc
```

[Add Sub Query](#)

**x Note!** Please place cursor where you would like to insert API MergeCode and then use 'Insert API MergeCode' button.

[Insert API MergeCode](#)

Subquery Name:  **Parent: 1\_1 Current: 12\_1 Level: 2 Order: 1**

Insert subquery here:

[Add Sub Query](#)

The subquery form is indented to visually indicate it is a child query. In this case, the parent is the first, top level query, as represented in the hierarchy fields **Parent:**, **Current:**, **Level:** and **Order:**.

The **Parent:** value displays a concatenated string consisting of:

parent query level + parent subquery level (if any) \_ + parent query order

In this case the parent query level is "1", there is no parent subquery, and the parent query order is "1", so the string reads "1\_1".

The **Current:** value displays a concatenated string consisting of:

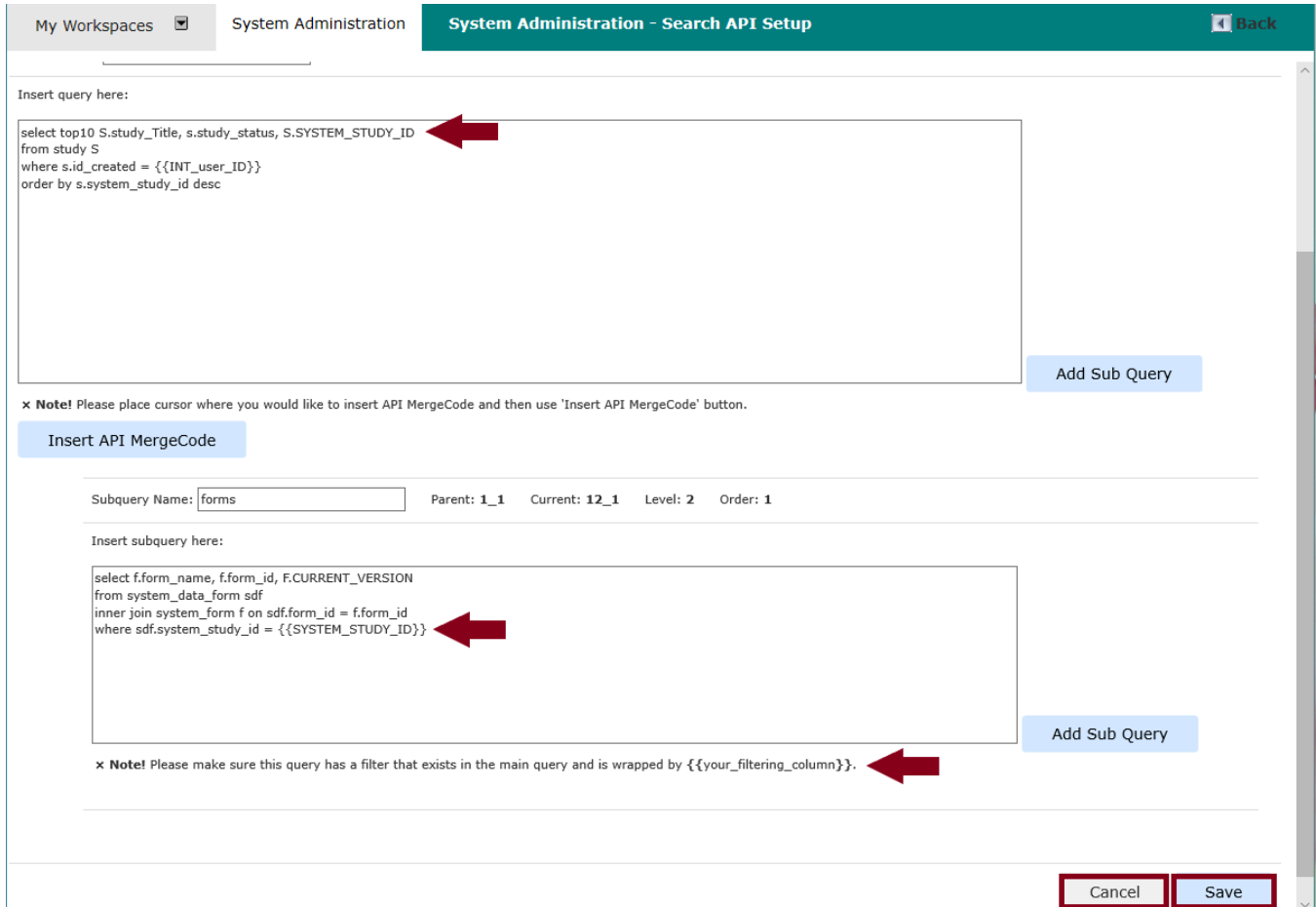
parent query level + current (subquery) level + current sub-subquery level (if any) + \_ + current (subquery) order

In this case the parent query level is "1", the current subquery level is "2", there is no sub-subquery, and the order of the current subquery is "1", so the string reads "12\_1".

The **Level:** value displays the level of the current subquery in the overall query hierarchy. In this case the level of the current subquery is "2" so the field reads "2".

The **Order:** value displays the order of the current subquery within its level. In this case the level of the current subquery is “1” so the field reads “1”.

In the example shown below the system administrator wishes to include forms in the result set for users identified in the first, top-level query.



As indicated by the note at the bottom of the subquery text entry field, the subquery must include a filter that exists in the parent query and that filter must be wrapped in double curly braces. Red arrows in the screenshot demonstrate that the filter SYSTEM\_STUDY\_ID from the parent query is being used for record selection by the subquery.


The subquery above is named “forms” and uses “SYSTEM\_STUDY\_ID” as its filter from the parent query.

Click the **Cancel** button to close the form without saving the subquery.

Click the **Save** button to add the subquery to the parent query. The **Search API Setup** page now displays a plus sign icon to the left of the query name, indicating it has one or more subqueries.

My Workspaces ▾ System Administration **System Administration - Search API Setup** ⏪ Back

Add Query


Query Id	Query Name	API Index	Query	Edit	Delete
	study details	50	select top10 S.study_Title, s.study_status, S.SYSTEM_STUDY_ID from study S where s.id_created = {{INT_user_ID}} order by s.system_study_id desc	<span>Edit</span>	<span>Delete</span>

Click the plus sign icon to expand subquery details.

Subquery details are displayed. All subqueries associated with the parent query will be shown here.

My Workspaces ▾ System Administration **System Administration - Search API Setup** ⏪ Back

Add Query

Query Id	Query Name	API Index	Query	Edit	Delete
	study details	50	select top10 S.study_Title, s.study_status, S.SYSTEM_STUDY_ID from study S where s.id_created = {{INT_user_ID}} order by s.system_study_id desc	<span>Edit</span>	<span>Delete</span>
	forms		select f.form_name, f.form_id, F.CURRENT_VERSION from system_data_form sdf inner join system_form f on sdf.form_id = f.form_id where sdf.system_study_id = {{SYSTEM_STUDY_ID}}		

Note that **Edit** and **Delete** buttons are provided at the level of the parent query. Subqueries cannot be edited or deleted directly due to potential negative impacts on data dependencies across the overall query hierarchy. The parent query must be opened for editing in order to edit or clear any of its subqueries.

### Query / Subquery Hierarchy

Note the location of the **Add Sub Query** buttons in the Add Query / Edit Query panel.

My Workspaces ▾ System Administration **System Administration - Search API Setup** ⏪ Back

Insert query here:

```
select top10 S.study_Title, s.study_status, S.SYSTEM_STUDY_ID
from study S
where s.id_created = {{INT_user_ID}}
order by s.system_study_id desc
```

Add Sub Query

**Note!** Please place cursor where you would like to insert API MergeCode and then use 'Insert API MergeCode' button.

Insert API MergeCode

Subquery Name:  Parent: 1\_1 Current: 12\_1 Level: 2 Order: 1

Insert subquery here:

```
select f.form_name, f.form_id, F.CURRENT_VERSION
from system_data_form sdf
inner join system_form f on sdf.form_id = f.form_id
where sdf.system_study_id = {{SYSTEM_STUDY_ID}}
```

Add Sub Query

**Note!** Please make sure this query has a filter that exists in the main query and is wrapped by {{your\_filtering\_column}}.

Clicking an **Add Sub Query** button adds a subquery *at the level of the query where that button is located*.

To add a subquery to the top-level query, click the **Add Sub Query** button to the right of the top-level query. In the example shown above, this will add a second level 2 subquery, with hierarchy details of **Parent: 1\_1, Current: 12\_2, Level: 2** and **Order: 2**.

To add a subquery to an existing subquery, click the **Add Sub Query** button to the right of the existing subquery. As shown below a form is opened to add a new level 3 subquery, with hierarchy details of **Parent: 12\_1, Current: 123\_1, Level: 3** and **Order: 1**.

My Workspaces ▾ System Administration System Administration - Search API Setup 1 Back

Insert query here:

```
select top10 S.study_Title, s.study_status, S.SYSTEM_STUDY_ID
from study S
where s.id_created = {{INT_user_ID}}
order by s.system_study_id desc
```

**Add Sub Query**

**Note!** Please place cursor where you would like to insert API MergeCode and then use 'Insert API MergeCode' button.

**Insert API MergeCode**

Subquery Name: forms Parent: 1\_1 Current: 12\_1 Level: 2 Order: 1

Insert subquery here:

```
select f.form_name, f.form_id, F.CURRENT_VERSION
from system_data_form sdf
inner join system_form f on sdf.form_id = f.form_id
where sdf.system_study_id = {{SYSTEM_STUDY_ID}}
```

**Add Sub Query**

**Note!** Please make sure this query has a filter that exists in the main query and is wrapped by {{your\_filtering\_column}}.

Subquery Name: Parent: 12\_1 Current: 123\_1 Level: 3 Order: 1

Insert subquery here:

**Note!** Please make sure this query has a filter that exists in the main query and is wrapped by {{your\_filtering\_column}}.

Notice that there is no **Add Sub Query** button available at level 3, subqueries can only be added at levels 1 and 2 of the hierarchy.

### Subquery Limitations

Because the result set can get very large very quickly when querying iRIS™ data, there are limitations on subqueries.

A maximum total of 10 subqueries can be added to the top-level query, *including sub-subqueries*.

This means that if four subqueries are added to the top-level parent query, a maximum of six additional subqueries can be added. These can all be level two subqueries, they can all be level 3 sub-subqueries, or they can be a mix of subqueries and sub-subqueries.

### **iRIS™ System Administrator Responsibility to Client Side**

The iRIS™ system administrator or designee has a responsibility to share query details and parameter requirements with the party responsible for managing client-side operations and processing.

The client side must know, or have access to, all top-level query names, login credentials required for each query, and all MergeCode (parameter variable) names and associated data types for each query.

Any time a query is deleted or edited in a manner that impacts on client-side requirements, the iRIS™ system administrator is responsible for promptly notifying the client side. Ideally, the system administrator and client side personnel should work together to coordinate any such changes.

### **Client-Side Requirements**

The iRIS™ REST API serves endpoints for querying iRIS™, but it has no client-side interface for passing parameters to the API nor for receiving its output. The institution must build or otherwise supply its own client-side software to perform those tasks. iMedRIS does not require nor recommend any specific application or tool, but the client-side software must be able to parse and output JSON (JavaScript Object Notation) name:value pairs.

### **REST API Endpoint for Data**

The REST API endpoint for data is `https://[client.com]/sysinterface/searchapis/deepsearch/{qryName}`, where [client.com] is the base URL for the institution's iRIS™ installation.

The {qryName} value is supplied by the iRIS™ system administrator. Note that the query name is the only parameter passed in the URL itself. Any required MergeCode variable parameters or explicit login credentials must be passed by the code employed to query the data endpoint from the client side.

### **REST API Endpoint for Forms**

The REST API endpoint for forms is

`https://[client.com]/sysinterface/searchapis/documents/{docType}/{docIndex}`, where [client.com] is the base URL for the institution's iRIS™ installation.

The {docType} and {docIndex} values are supplied by the iRIS™ system administrator. Note that {docType} and {docIndex} are the only parameters passed in the URL itself. Any required MergeCode variable parameters or explicit login credentials must be passed by the code employed to query the forms endpoint from the client side.