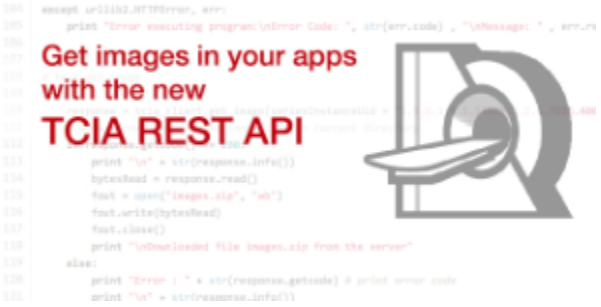
TCIA REST API Guide v4

Summary

This document describes v4 of the TCIA programmatic Interface or REST API implementation. This API is designed for use by developers of image analysis and data mining tools to directly query the public resources of TCIA and retrieve information into their applications. The API complements the existing web interface but eliminates the need for users to visit the TCIA web pages to select and download images then upload them into their viewing and analysis applications. The TCIA Programmatic Interface is based on a middleware platform called Project Bindaas, developed by Emory University and uses REST web service technologies.



The API is a RESTful interface, accessed through web URLs. There is no software that an application developer needs to download in order to use the API. The application developer can build their own access routines using just the API documentation provided. The interface employs a set of predefined query functions (see REST API Directory) that access TCIA databases.

If you are interested in using the API and have any questions, please contact us at help@cancerimagingarchive.net.

Previously access to the APIs required an API-KEY. An API-KEY is no longer required to access the public TCIA collections.

What's New

The following characteristics apply to all TCIA APIs.

Version 4 (*current*)

- The getSeries API has been modified to include new query parameters.
- Bug fixes.

Version 3 (*Documentation for Version 3*)

- Two new APIs have been added (/getSingleImage, /getSOPInstanceUIDs).
- Three new APIs have been added (/NewPatientsInCollection, /NewStudiesInPatientsCollection, /PatientsByModality).
- The getSeries API has been modified to include new query parameters.
- Bug fixes.

Version 2 (*Documentation for Version 2*)

- You can access the metadata of an API by appending **/metadata** to the end of a QueryEndpoint, without any QueryParameters (See below). The metadata is in JSON format and conforms to this schema.
- Most APIs can return results as CSV/JSON/XML/HTML. You can specify the return format by including the query parameter **format**.

Getting Started with the TCIA API

Getting access to the API:

- Previously access to the APIs required an API-KEY. An API-KEY is no longer required to access the public TCIA collections. Simply call the RESTful endpoints.
- To support existing code without changes, an API-KEY can be included in the url by adding an extra query parameter **api_key** or it can be included in the HTTP headers. Since the API-KEY is no longer needed, the underlying service will simply ignore it.

The following characteristics apply to all TCIA APIs:

- You access a resource by sending an HTTP request to the TCIA API server. The server replies with a response that either contains the data you requested, or a status indicator.
- You can access the metadata of an API by appending **/metadata** to the end of the query. See examples. The metadata is in JSON format and conforms to this schema.
- Most APIs can return results as CSV/JSON/XML/HTML. You can specify the return format by including the query parameter **format**.
- An API request takes the following structure:
<BaseURL><Resource><QueryEndpoint>?<QueryParameters><Format>

For example, the API shown below requests all studies in the TCGA-GBM collection for patient GBM-0123 as CSV (not e that in this example, patID is not a valid TCGA-GBM ID and this call will return only column headers).

```
https://services.cancerimagingarchive.net/services/v4/TCIA/query/getPatientStudy?Collection=TCGA-GBM&PatientID=GBM-0123&format=csv
```

This can be broken down as follows:

Object	Example	Description
BaseURL	https://services.cancerimagingarchive.net/services/v4	The BaseURL includes the version number of this API (v4 in this example)
Resource	/TCIA	
QueryEndpoint	/query/getPatientStudy	
Query Parameters	Collection=TCGA-GBM & PatientID=GBM-0123	
Format	format=csv	Some APIs support CSV/HTML/XML/JSON, while others only support a single return type. Therefore this is required only in instances where multiple return types are supported.

- Coding examples and an SDK (in Python & Java)
- The table below contains the most up-to-date documentation of the API.

API Reference

The full API consists of a base URL followed by the api and the query parameters in that order.

Resource	QueryEndpoint	Query Parameters ***All query parameters are optional unless stated otherwise***	Format	Description
/TCIA				
	/query/getCollectionValues	None	CSV /HTML /XML /JSON	Set of all collection names
	/query/getModalityValues	Collection / BodyPartExamined	CSV /HTML /XML /JSON	Set of all modality values (CT, MR, ...) filtered by query keys
	/query/getBodyPartValues	Collection / Modality	CSV /HTML /XML /JSON	Set of all body part names filtered by query keys
	/query/getManufacturerValues	Collection / Modality / BodyPartExamined	CSV /HTML /XML /JSON	Set of all manufacturer names filtered by query keys
	/query/getPatient	Collection	CSV /HTML /XML /JSON	Set of patient objects filtered by query keys
	/query/PatientsByModality	Collection (**R**) Modality (**R**)	CSV /HTML /XML /JSON	Returns a list of PatientIDs, given a specific Collection Name and Modality
	/query/getPatientStudy	Collection / PatientID / StudyInstanceUID	CSV /HTML /XML /JSON	Set of patient/study objects filtered by query keys

	/query/getSeries	Collection / StudyInstanceUID / PatientID / SeriesInstanceUID / Modality / BodyPartExamined / ManufacturerModelName / Manufacturer	CSV /HTML /XML /JSON	Set of series objects filtered by query keys
	/query /getSeriesSize	SeriesInstanceUID (**R**)	CSV /HTML /XML /JSON	Set of total byte size and object count filtered by query key
	/query/getImage	SeriesInstanceUID (**R**)	ZIP	Set of images in a zip file
	/query/NewPatientsInCollection	Date (**R**) Collection (**R**)	CSV /HTML /XML /JSON	Returns a set of Patients that have been added to a specified collection since a specified date. Date is specified as (YYYY-MM-DD). Use the getCollectionValues to get the list of available collections. ⚠ There is a bug if you include a Date argument to NewPatientsInCollection under v3 of the API. This was fixed in v4.
	/query/NewStudiesInPatientCollection	Date (**R**) Collection (**R**) PatientID	CSV /HTML /XML /JSON	Returns a set of Studies that have been added to a specified collection, and optionally to a patient since a specified date. Date is specified as (YYYY-MM-DD). Use the getCollectionValues to get the list of available collections.
	/query /getSOPInstanceUIDs	SeriesInstanceUID (**R**)	CSV /HTML /XML /JSON	Return a list of SOPInstanceUID for a given series using the SeriesInstanceUID.
	/query /getSingleImage	SeriesInstanceUID (**R**) SOPInstanceUID (**R**)	Raw DICOM Object	Returns a **SINGLE** DICOM Object that is identified by its SeriesInstanceUID and SOPInstanceUID. This API will always be used following the /getSOPInstanceUIDs.
/SharedList				

	/query/ContentsByName	name (**R**)	JSON	Given the name of a shared list, return its contents.

API Metadata

The API now supports the ability to programmatically access the metadata about your API. This information is provided as a JSON document and includes:

- Name of API
- Free text description
- List of Query Parameters
- Supported Return Types
- A description of the returned attributes: Name, DICOM Tag and Description

The returned metadata conforms to a JSON schema.

Example:
Let us say we wanted metadata for the getPatientStudy query from our earlier example. The query would look as follows:
https://services.cancerimagingarchive.net/services/v4/TCIA/query/getPatientStudy/metadata

Or in other words, the query would have the following structure:
<BaseURL><Resource><QueryEndpoint>/metadata

Don’t forget to include the api-key in either HTTP headers or the URL of the API.

Return Values

More details on the return values are available.