TCIA Programmatic Interface (REST API) Usage Guide v2

Documentation for ver 2. of the TCIA API
The documentation for the latest version of the API is available here

Summary

This document describes v2 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.

What's new

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.
- Every request must contain an API-KEY. The key can be included in the url by adding an extra query parameter **api_key** or it can be included in the HTTP headers. We strongly recommend that the API-KEY be sent via HTTP headers.
- You can obtain one API-KEY and use that for your application; you do not need a separate API-KEY for each user of your software.. To obtain an API-Key please send a request to help@cancerimagingarchive.net or contact TCIA's help desk by phone.
- 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

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**
- Every request must contain an API-KEY. The key can be included in the url by adding an extra query parameter **api_key** or it can be included in the HTTP headers.
  You can obtain one API-KEY and use that for your application; you do not need a separate API-KEY for each user of your software.. To obtain an API-Key please send a request to help@cancerimagingarchive.net or contact TCIA's help desk by phone.
- An API request takes the following structure:
  <BaseURL><Resource><QueryEndpoint>?<QueryParameters><Format>
  For example, the API shown below is a request to get all studies in the TCGA-GBM collection for patient GBM-0123 as CSV
  https://services.cancerimagingarchive.net/services/v2/TCIA/query/getPatientStudy?Collection=TCGA-GBM&PatientID=GBM-0123&format=csv
  This can be broken down as follows:

BaseURL	https://services.cancerimagingarchive.net/services/v2	The BaseURL includes the version number of this API (v2 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 (in Python & Java) can be found on **here**
- Interface documentation can be found on **Mashape**
- The interface is registered on **ProgrammableWeb**

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/getPatientStudy	Collection / PatientID / StudyInstanceUID	CSV/HTML /XML/JSON	Set of patient/study objects filtered by query keys
	/query/getSeries	Collection / StudyInstanceUID / PatientID / SeriesInstanceUID	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
/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 the following 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/v2/TCIA/query/getPatientStudy/metadata

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

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

Return Values

Click here to see more details on the return values.

Testing the API

There are two RESTful servers provided by TCIA. A test system is loaded with a small set of known data to allow you to test your applications. The production system is configured to use the full TCIA database. The query format is the same for both systems. The base URLs are:

Type	URL
Production	https://services.cancerimagingarchive.net/services/v2/TCIA/query
Test	https://services-test.cancerimagingarchive.net/services/v2/TCIA/query

Click Here to see the test data that has been loaded on these test servers have the following test data