Drive Thru Data: Using NLM APIs to Access Information Fast – Transcript of audio

I'm with my colleague who was on tech support. Drive-through data, using NLM API to access data fast. I will let Mike take it away.

Thank you so much. Thank you for having me today. Here we go. My name is Mike Davidson. I am a librarian at the national Library of medicine, known as NLM. One of them is what I'm doing today. Developing and delivering training to help people learn about and how to use our products and services. I get to talk to you about one of my favorite things to teach people about. Which is API. I will let you interact with the product and services. If you haven't spent much time using NLM products, today's session should help you introduce you to concepts applicable to ABI for non-MLM products as well. I try to make today's session as interactive as possible. I'll be walking you through a demo. All you need is a web browser. We will be using WebEx: feature in a few moments. We will be using the chat panel. We will start off by dropping a link to a handout in the chat panel. It includes all the links I will mention today. Every link I mentioned will be on the handheld. If you have questions at any point during the session, feel free to drop them in the chat as you think of them. Moderators will keep an eye on the chat and will capture any questions you have for me in the Q&A portion which I will answer at the end of my talk. I have a few questions to be asking you. I would like for you to answer them the chat. As a practice round, to get everybody familiar with working with the chat, let's start off with this. What is your favorite fast food place? Answer this in the chats. I will turn off the camera while this is going to avoid the destruction of looking at my face. We have taco Bueno. Steak and Shake. Any other suggestions? Taco Bell. Jimbo tacos. RVs. Shake shack. RB. Regional. That is not my region. That all sounds real tasty. We are doing this on the East Coast, after lunch. I apologize for those not eaten yet. Dunkin' Donuts. Jiffy burger. Nice. Very clean, excellent choices. Now we're comfortable with the chat. Before we get in the meat of today's session, this is what we are hoping to accomplish today. You should be able to explain what it API is. How it interacts with the stems and how you might choose to use it to accomplish certain types of tasks. You can describe the basic mechanics of how to use it. You identify some APIs they make available for the product. My job is to help you learn about NLM products and services. It is a pretty big job. NLM has a lot of products. There are many different types of health and biomedical science information. Lots of different people need this information. In order to accomplish this mission, the portfolio is brought. What you see is a small sample. One a lot of people about this, think about pub med, -- and --. People think about us as collector and provider of consumer health information. NLM does work in the medical terminology space. We provide a number of databases and tools from manual Keeler biology. It's a good place for drug and chemical information. We are the home of clinical trials.gov which provides access to info from clinical trials around the world. It's a small sample of what we offer. Take the opportunity to do our first pole. It'll be popping up for you to respond to in just a moment. Which, if any of the category listed in the poll are of categories that you use? You can check off more than one. If products you don't use are not listed, tell us would of the products you use in the chat. It looks like it is possible to pull may only let you enter one response. What is your number one response then? Baby --

---

is a couple more people that are in progress. Close this out. Looks like literature is far and away the number one response. Consumer help is little bit behind that. That's pretty much in line with the cup of -- type audiences would get like this. For all the products you use, most of you who use these products probably access the resources through our website. Some who use and pursue innovative ways to use the phone need a different type axis. Access let me talk a few minutes about the specialty. Tronic health record system. That happened patient portal where you can review your own chart and test results. A lot of the info available may be confusing to the average patient without further explanation by clinician. Solution we use is an info button. Patients can click these buttons next to lab tests, medications and medical procedures and get more info on the topic in an approachable and easy to learn for. NLM's largest consumer health resource is full of this authoritative up-to-date info designed for patients and families and healthcare providers. Developers and providers can configure the systems to call up information automatically. This can help patients stay informed about their company. Here's another example. This is from the world of clinical oncology. I want to give a huge thanks from --. For this example. Increasingly, as genomic sequencing becomes cheaper and more feasible, we can sequence cancer genomes and take a molecular-based approach to cancer treatment. A tool to facilitate this molecular oncology treatment see bio portal. Developed by the memorial cancer center. It helps clinicians analyze molecular data. But this is only the beginning. Clinicians need to do is turn the data into a treatment plan. The treatments aggressively developed and validated through clinical trial. In order to provide easy access with the specific consumer profiles. We integrated data from the clinical trials.gov irrevocably into the instance of see bio portal. You can see clinical trials registering the participants. By using the patient's data already in the bio portal as a filter, most of trials can be sorted and ranked. Helping clinicians find studies that are the best match for patients age, location and specific cancer genes. It's in the app without the patient or clinician ever needing to visit the trials.gov website. I want to talk about a research example. Using literature citations as data to understand transit scholarly publishing. We use a data set. The specifically wanted to look at the nationality of authors to see if specific journals published from specific part of the world's more frequently. Author nationality is not a metadata field included in pub med. But author affiliation is included for citation. It contains evidence of a country, but in an unstructured way. Want to see author nationality can be reliably identified using language processing and learning techniques they can use these techniques to explore whether journalists and publishers had bias against authors of certain backgrounds. They needed a sizable set of pup that citation. Including author affiliation to train and test the algorithm. They needed the citation data into machine readable format to feed into the algorithm. Getting the requisite number of citations would be cumbersome and time-consuming using traditional methods. Doing so a machine readable format extra care. The team explored other approaches of accessing data. These are very different projects. Each accessing info from different NLM resources. Clinical trials, Medline, and pub med. What they have in common is they use NLM info but not NLM websites. In order to work the way they do and be successful as they are, they need a different type of access to these products. They need to access information in another application. Not a normal web browser. They need to work automatically the sum degree. All three examples need NLM information and a specific format which may not be available through the products format. They need health information as data. How do they accomplish this. They do this using application program interfaces. Or API. Before we go any further, we will open up another pole. What previous experience do you have. May be used to periodically? Maybe you use them in the past but not recently. Perhaps this is all new to you which is fine as well. He will give you a few moments to respond to that pool.

Don't forget to hence the apply button at the bottom. Close the whole poll.

The most common responses, know about them but have not use them. All new to me. Definitely a bunch of folks which this is used too. And some people who use them periodically but not frequently. What I will say. This is kind of the trick question. Even those who said you never used a PI, you may not of works with them. But you used an API in some capacity. If he ever check the website to see the local branch of every child chain -- retail chain has something in stock, or use an app to get traffic of dates and of dates on the roots, they all have a PSP -- they all have a PI. It can be tricky to find. Since the term has been used to describe the need for and concept. The. An API is a set of protocol for contacting a remote system and making requests. In other words, a set of protocol for interfacing with the remote system. API are designed to be used programmatically as part of a computer program, not directly by humans. An interface to be used when developing program applications. Or an application programming interface. What we will talk about today include a computer somewhere which we would call a server. It has info somebody must access and a set of rules making requests or calls to that server. It can all get pretty technical. Before we go further, let's take a step back to talk about another way to talk about API. Thinking back to your favorite fast food restaurant which I asked about at the beginning of the session. If you want to order something from a fast food place, park your car and go into the dining room and order from the counter. That's like going to the website of your chosen NLM resource. Or go through the drive-through. The API -- it's like an API. A quicker and streamlined way of the same stuff you get inside. When you pull up, you can say something into the intercom and get a response back from the other person on the other end. They can tell you anything, what did you say or we are out of what you asked for. Or they tell you to pull through to the next window to pick up what you asked for. You get what you want and he never get out of your car. APIs are basically working the same way. Rather than going to the website, you are a computer program you use sent a message to the remote system requesting whatever it is you're looking for. The remote system then responds. If you have something and ask for something the system has available, you can format your request correctly and the system response by giving you exactly what you asked for. API can save a lot of time and energy in a certain situation because they are designed to be integrated into programs. Developers and programmers can incorporate requests and calls directly into the program or application. Even at the websites they are designing. They can program rules for properly creating API calls to the application and that an application create a request based on the hardcoded rules. If you make dozens and hundreds and thousands of requests a day, this is the only practical way to do it. In addition to be faster and easier, they can access data from online database directly as part of the normal function. You can do more without human intervention. Rather than requiring to go to the websites, and putting the application, it's handled automatically by the program. Some API provides access to data in different formats than the corresponding websites. Machine readable format unless readable to the average human user. It may include data not available through the website. I don't want to get technical about how API does what it does. I'm not an expert in the technical nitty-gritty myself. In order to see how this may help you and users get things done, it can be useful to know how they operate. Remember the API we are talking about today are composed of a server somewhere, remote computer you want to access, and a set of rules and protocols for interacting with the remote computer. The server is the specific fast food place you are ordering from. The rule of protocol or the menu board outside letting you know what is available to order. The way you and your computer interact is through the URL. It's like the one you visit in your web browser and website. The URL you use includes not only the address of the server of the API, but details of what you are requesting from the API. It depends on how you can construct the URL. We will look at examples of API URL in a moment. The URL is actually made up of two parts. Parameters indicate what you are asking the API for. Let's start with the base URL. This is the address of the API server. Think of it like the street address of a specific drive-through restaurant you're going through. Every time you go to the restaurant, it's going to be at the same address. Every time you use a specific API, the base URL for that is going to be the same. It tells you computer where to go to access the remote system for the API. You can see on the screen here a few examples of this URL of some of the API we will be talking about a little bit later on in today's presentation. The rest of the API URL is made up of what we call "parameters." They include details of what you're asking the server for. Like search terms, or how many results you want to see and what formats you want the resulting. This is like your actual order at the drive-through window. What you are actually asking for. Like at the drive-through, this will most likely be different for every request Lucy always want to get the same exact results. We have a DB parameter which indicates what database we want to get the data from. I.D. parameter, specifying specific PMI D for record. A rep mode parameter specifying we want to retrieve results in the XML. And --, which you was full. Which means we want full returns not summaries. The different API allow and require different types parameters. You need to look up how to use parameters for Pacific API. We talked abstractly about how API operates. You can feel free to feel follow along in your own web browser. For this example, we will use the MEDLINEplus web service API. It'll let you search for and retrieve the info available through the health topic webpages. In an XML format, it'll easily be processed by machines. It'll help you embed the information on a website you may be building. You can see this on the screen. Https:// W search.NLM.NIH.gov/WS/query. The base URL for the API is always going to be the same. Anytime you want to query the Medline web service, it'll start with the base URL. It's listed on your handout. Make sure to copy and paste this into your browser address bar instead of clicking on it. It won't do any good without the parameters. Next thing we need to do is figure out the parameter. This is important as it'll tell the Medline web service server exactly what we're looking for. If we don't spell this out in the parameters, it won't be part of the request and will inform the results. Before you put in any parameter, make sure you put in the?. Let them know where the address finishes and parameter starts. A few are required by this MEDLINEplus web service. It's health web service which database we want to look at. The English-language health topic database with the Spanish-language database. In this case, we chose to look for help topics in English. If you're following along, after the first parameter, put in & to separate this. We need to separate and specify what we are actually searching for. I'm looking for help topics pages about acid reflux. If you want to follow along with the different search query, please feel free. Make sure to replace each space with plus sign. Put in & to separate this from the next one there are additional optional parameters we can use, which helps define what format we want the result in and other options. For this example, musing parameter called Max to specify the maximum number of results I will return to me. I'm going to limit the query to just the first five results. Straight of the handout into the address bar or later. We can stay tuned to the next slide. We start with the base you are? Separating successive parameters with in &. We now build a MEDLINEplus web service API call. We can put the URL into the address bar of any web browser to see what we get. What I got when I put the URL into the browser is the big chunk of XML. It contains the top five English-language topic that match our search for acid reflux. This is what we got is it's what we asked for. Why would you do any of this? The MEDLINEplus website has a perfectly nice interface where I don't need to know all the fancy syntax and it returns webpages I can read without being fluent in XML. You probably would not do this. Not the space. This is a good example. But it's not an efficient way to use API. The best use of API by programs and applications prescribed that have the URL creation built into them. They can take inputs for user, build corresponding URL, request the info from the API, except muscles in a machine readable format and process this data into whatever format the application needs in order to shoot the user. That's the ideal. You can't walk up to the drive-through window. You need a car and you need to know how to drive the car. In this metaphor, the car is a programming scripted language. If you know how to drive it, it can be used to automate a call to the API. To extend the metaphor a little further, you might be able to roll up to the. Drive-through on a bicycle. The equivalent of cycling up to the drive-through window. Technically Lawrence, but it's not particularly efficient. What kind of car programming language you need to make use of the API? Any motor programming or scripting language will be able to do it. If you are developing, whatever programming language you are using will almost certainly be fine. If you are new, you may want to look at either Python or RA. Both languages are on the easier site to learn. Both offer prebuilt toolkits called packages of commands designed to make you programming easier. If you using BR programming language, spelled RE and T are easy. It's designed to help with the utility API. These packages can streamline the process of working with API in your program and make it so you don't have to worry as much about the mechanics. Think of it as the equivalent of an automatic transition in your car. It is and think of -- it doesn't do anything a manual transition can't. But it makes it much easier with a few listings to think about while driving. The amount of programming required to use an API might not be that much. Just creating a really rudimentary script that automates data retrieval using API. This may all be that is needed to achieve your goals. For some, any amount of programming seems like too much. It segues much likely into my next point. If you don't know how to drive just yet, you do have a few options. If programming and scripting is new to you and you are interested and want time to learn and have patience, it's worth your effort to learn. There was another alternative which is, get a friend to drive you. If you yourself are not a programmer but you find somebody who is, ask them for help in building a project with building an API. You don't have to learn yourself. But it can still be helpful to learn but API, how they work, and wish API available that might help you do what you want to do. If you learn which API options are available, you can better communicate with you programming friend about what you need using language that may be easier to understand. Going back to the drive-through analogy. If you give a driver the address to the fastest place and directions and how to get there and tell them exactly what it is you want to order, and makes the process go smoother. At the very least, you can play programming partner to a specific API, the can access data you need, you can save them time and research that they may not be as familiar with. We will talk about specific API just a moment. I want to do another poll. To gauge where everybody is at in terms of prior programming stance. What I would like you to do is let me know if you are familiar with any of the programming language listed. I believe you must select an option. Choose the one you are most familiar with. If you have not had any programming experience, that's fine. This programming language you use not listed, throw this much out as well.

Think we are ready to close.

We have Python. JavaScript as well. C++. Excellent. I'm curious about that. We have a few who said, Nanette. Again, totally fine. You can always find a friend who can help you drive you Sweeney two. There are ways to learn about programming towards the end. SGL. Let's move on again. I want to spend most of the rest the session going through and LM's most popular and well-known products and talking about API you can use to access the product information. This is not going to me a comprehensive list. She is sort of sampler platter to give you an idea. I know some of you may not be familiar with all the resources to be -- to begin with. If somebody sounds interesting, sign up for the other Academy webinars that talk about NLM products. There's more coming in later this fall. Check out the training opportunities all mention at the end of the session. As you consider options am about to talk about, it's important to remember APIs are not one-size-fits-all. You can't use anyone you choose to access whatever remote system you want. You can't go to Starbucks and order a big Mac or Taco Bell order a pumpkin spice latte. You must use the API for the data you want. The first step in choosing an API is making sure it can look for the data you need. An online database may have different API that can achieve the database data in different ways and formats. We will look at a couple of examples of this. Give me a few moments. I will mention on the handout, there are links to more info about each the API. If one catches your eye, check it out in more depth after today's session.

Since we talked about MEDLINEplus API a few times, let's start there. You might remember the example of building a URL. I was using this MEDLINEplus web service API, which lets you search for and retrieve health topic info. And X amount. They make this available to be used by electronic health records and patient portals. MEDLINEplus connect is a service he can use to provide free, authoritative and up-to-date consumer health information to patients based on specific medications, procedures, lab tests and medical conditions for that basic health record. This is the API using the example I talked about we back at the beginning of today's session. We can't very well talk about NLM products and services without talking about pup matter, the premier database of biomedical literature. Hub made contains much literature access for anybody for free. 3.4 million users access pub med website each day. It doesn't include API usage. Over 1.2 billion pub med searches were conducted through API. That's 3.3 million searches a day. The overwhelming majority of the searches were done using an API called utilities. It is a suite of APIs providing accesses to 35 different databases products, developed and maintained by the national Center for biotechnology information. And CBI. A division responsible for pub med. You can use this utility to retrieve methods in a variety of formats. Including the full pub med XML, including some data not exposed on the public website. Utilities can let you grab lots of pub med records quickly so you can treat pub med citations like data. It's critical for those doing metric research and portfolio analysis. They use utility API to achieve a large number of pub med record and XML format. And use the data as training data for the machine learning project. In addition to utilities, there are two other more specialized API you can access data in specific ways. The literature citation exporter API can be used to exert PMI D, unique identifiers assigned to the public record. It's a great tool if you need a way to quickly and automatically generate new cetacean strings. Cetacean matcher takes technology embedded in the search box on the pub med website and makes it available for developers to use automatically. Using this will return a set of PMI D for records much likely to match the string. Sticking with the literature them, let's talk about two other products. Pub med central and bookshelf. PMC is a free fulltext archive of biomedical journal literature. Bookshelf provides online access to books and other documents for free. He can use utility for both PMC and bookshelf. Or from books in the NLM literature archive. They are built using standard called open archive initiative protocol for metadata harvesting. OAI PMH. It is a standard used by many different online digital depositories for the API. The standardization means if you can find a program that is already written to use one of these, or program familiar with the standard, it is much easier to adapt the work and knowledge to a tree fulltext from the bookshelf open access collection. I want to talk briefly about some of NLM terminology focused products. It is the controlled vocabulary used for indexing and describing biomedical health information. Many records are indexed with descriptors to help prove discoverability. If you want to explore or download the mesh vocabulary programmatically, you have a different API options. Remember the utility API which we talked about in the last two slides. Because of the vocabulary hierarchal nature, the utility can be a little limited and how it can retrieve mesh data. This is one of the reasons why we can't have a one-size-fits-all approach to a pair. The nature of the data we want to a tree, the tools and protocol we need to retrieve the data has to vary as well. NLM provides another API called mesh RDS. It helps to query and retrieve vocabulary info in a different way. RDF, stands for resource description framework. Standard to represent metadata and link data. The mesh RDF API can be used to query a link cetacean of mesh. Using this is more complex than other API I discussed. It's a slightly different skill sets and searching pub med and MEDLINEplus. Mesh RDF may be the API you need. You may not be surprised to learn that our unique challenges the sonic computer systems to keep track of prescription drugs and medications. Maybe they can identify different codes. The provide the terminology of clinical drug names and codes that serves as a crosswalk between other drug vocabularies. They can integrate the form into their projects to ensure their application for each individual medication. They get easier to get started using the API. This website where you can spam it would different features of the API. You can try out API: workload and figure out how to make it work well for the project you are doing. Once you found the right combination of API calls for the project, you can take what you learned in RX program this into the application. This makes it easy when developing and testing a project to tell whether the project is informed and somebody else in your code. We talked about clinical trials.earlier on. Based on genetic profiles. The project uses the clinical trial.gov API language. Innocent have a special fancy name. They are modified versions of the portal using relevant patient information to formulate a search request to the API. The result which they send a machine readable format, it converts into a readable interface for clinicians to review.

The API also has a web-based graphical user interface. Shown here on the screen they can use to develop and test out the API call. Now we ran down the menu. What looks appetizing? We are putting up another pole. I want to let me know which of the API that we talked about are you most interested in learning more about? Were working with. Pick your top choice. If you're still not sure why or how you want to use API, is an option for that as well.

We have people still in progress. Give us a few seconds here. I'm going to close the pullout.

So many good options. The overwhelming majority of popular vote are utilities. Not surprising. It has so many different resources. That's another popular one. MEDLINEplus web service. These are all good options. I do a lot of work with utilities. I'm glad to see folks who are interested in that one. All right. As we head towards the end of today's session, I want to go back over a few key concepts we covered. This is some of what I hope you take one with you from today's session. Aside from a craving for fast food. Despite -- this may help you who are unsure if or why you may need to use API. API are great way to retrieve data from anonymous sources. In order to make best use of API, you want to use API in the context of a program, script replication you are developing. APA are designed specifically to be used programmatically. API provide access to data in machine-readable formats. Like XML and Jason. The great way to get the most current up-to-date data quickly and on demand. API are useful when you are searching for and requesting specific things from the database. When you know more or less what you're looking for. Putting the last two bullet points together, API are great at getting exactly the data you need exactly when you need it. Despite how cool and useful I think it is, there are still times when using an API is not the right call. First of all, this may seem obvious based on the last slide. I want to underline it. If you're not programming or working with somebody program, API is probably not the right choice. Also, as I said many times before, API are not one-size-fits-all. Unita specifically for the resource you are trying to retrieve data from. Can't use an API to access if which were using this and have API. Many NLM have their own specific API. If you're not sure what you're looking for and are still exploring and browsing, getting to know how to use the, stick with the web version until you are more familiar with it. Because the records açai syntax for request, and it's in machine-readable not human readable format, you are better starting explanation until you have a better idea of what you're looking for. I'll share some places to go for training and education on the web interface. So you can get more familiar with it. While it is really good of treatment data, if you're working on a project that needs access to the entirety of the database data, you want to do an analysis of all of med 35 million presentations. Retrieving the data with API may be cumbersome and long. Many have throttles and dates to prevent people from using and overusing the system. That may be an option for projects that require you to download all the data in the database. It's not the point of the presentation to touch on it briefly. The data is available in all downloads via FTP. Including placement, the open access, catalog records, health topics, and much more. Both downloads can be invaluable. You need to set up your own database to search and query the database. The bulk downloads are snapshot of the database of the time you download them.

You may end up with something easier and better than what you get at the fast food place. You have to go back to the store next week to get fresh ingredients. Hopefully this presentation help to learn more about start working with NLM and API. If you want to know the next step is, here are some suggestions. Learn about programming and scripting. There are many online courses to teach the fundamentals of Python in order to help you engage with API using beef and witches. Library carpentry hopes -- including workshops that have an intro to the programming languages. Ask around at the institutions you are at to see basic programming languages are available or if any of your coworkers have the skills you need. Something else I can help if you have a project in mind is thinking carefully about the project before you get started. Identify what info you have going in through the search term, clinical trial criteria. What info you're trying to achieve. Because if you request require specific syntax for specific info, a little planning ahead of time can save you a lot of trial and error work. Finally, make sure to identify the API look into your project. There are many more as well. We find the right API. It serves as a directory of no access points. Including API and bulk download. Some data sets without dedicated API can be retrieved directly from data discovery. Using this is much easier if you are already familiar on how to use the resources they provide access to. Plenty of offering to help you get up to speed. It offers a wide away of services. You can find at NLM.gov/training. Additionally, check out the learning resources database, a one-stop shop for webinars, classes, reporting and a wide variety of education resources covering many products. Find this at learn.NLM at NIH.gov. You can find these links on the handout. Today's session is one of a series of Academy webinars be presented by trainers comprising some of our greatest hits. Check out the upcoming sessions and look for the recording of for not available. Spent time reading the API documentation. It tells you what API can and cannot do so you know if you're spending time on the right API. All of the possible record parameters. Most API have guidelines and restrictions for usage also documented. They prevent the system from being overloaded with too many calls at once and help protect our resources code. Is the can help you find the documentation for all the API today. I would make sure to have temper questions. We might have a few minutes. Put questions you have a chat. We appreciate if you would fill this out. I will give the link to the handout ready. Give me one second. I reposted the link for you all. Can you Inc. this?

This is longer ago than I care to admit. I'm not sure what the latest state of technology is there. I don't know any specifically. It's been a minutes.

We have a couple more questions. If you have any more, please put them in the chat. Since you can construct a clinical trial, what is the advantage of using API?

It's important to underline. The big advantage is you can have another program you said. Another program doing the work for you so you don't have to go to the website yourself. If you want to do a search on the website, that's totally fine. It's why the website exists. If you work in a context with another program and application, the API can be integrated into the program to take the input and construct a search and send it and get a result. If you have a program, you can do more searches faster than a person can. And more than one person can get bored and another person can. Working in a programming environment is useful in the first place. But also why API can also be useful. That's an excellent point. Two.

We are out of time. I'm going to wrap the webinar up. Thank you for presenting this fantastic webinar. Thank you for my colleague for doing tech support. Check out the upcoming webinars. We have one on the new USA.gov on Thursday. Geography 101 on June 22nd. We will have more presentations coming up from NLM. We will have one in August. Beyond the binary. LGBTQ help. We will have more in September and one in October. Thank you all for attending.

Thank you everybody!

[Event Concluded]