Integration Fulfillment

Fulfillment

Obtaining a test ID

The first step in the fulfillment process is obtaining a test ID.

There is always a human-initiated action that launches the process. For some Local Administrators this may be a direct purchase of the test ID by the candidate on itepexam.com. For others it may be part of an HR vetting process or school admission process where the user of a system such as an HR or school admissions system is used by a Local Administrator to indicate that a test id needs to be provisioned for a candidate. Perhaps the candidate initiates the process by applying for a position on a corporate web site, at which time the HR system needs to provision a test ID for the candidate without any additional input.

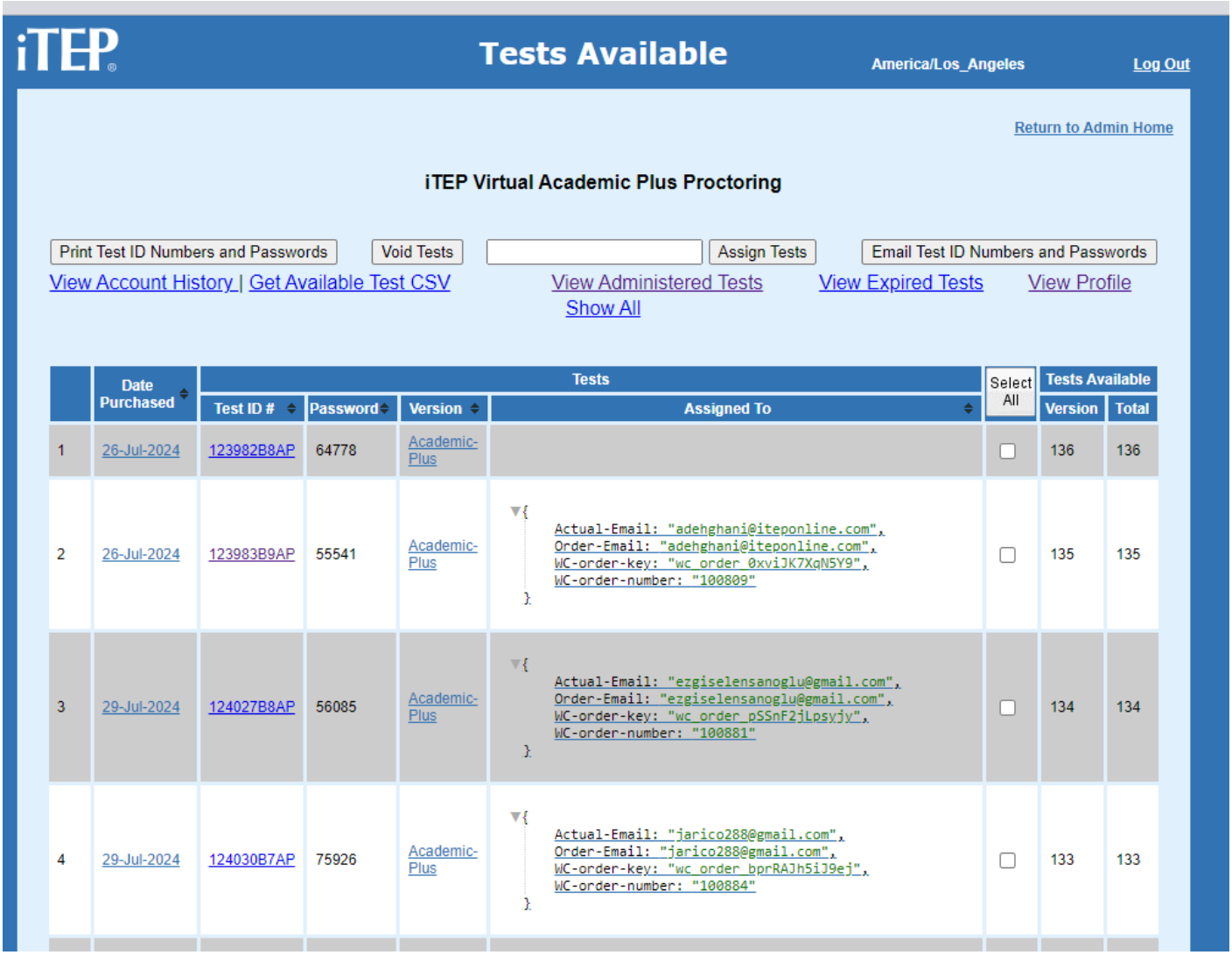
In all these cases once the process has been initiated the provisioning of the test ID is automated.

The source of the test ID must first be considered. There are two possible sources. The first, and most common, is that the Local Administrator has an inventory of test ids. In this case an available test ID must be identified in the Local Administrator's inventory. The second is that a sale is created for the test ID at the time of fulfillment.

It is important to note that iTEP's administration system is the final authority with regards to a Local Administrator's available test IDs. iTEP's definition of an available test ID is one that has a status of "Ready". This leads to the first decision that needs to be made regarding fulfillment - where and how is the Local Administrator going to keep track of test ID inventory? At the very least this is needed so that the Local Administrator can determine when additional test IDs need to be purchased. If a large number of test IDs are maintained in inventory all that may be needed is to occasionally check the iTEP administration system. If the test ID inventory is kept at a low level with regard to the usage rate then the Local Administrator must monitor their inventory of test IDs more closely. Most of iTEP's customers use the iTEP administration system to manage test ID inventory. A few have developed systems where the available test IDs are maintained on their system as well as on iTEP's administration system. iTEP's experience is that maintaining a dual inventory system is more challenging to implement and maintain. iTEP currently does not have any APIs directly related to synchronizing inventory. While the are some APIs that a Local Administrator can use to

allocate their existing inventory, there is not an API that can be used to automate the procurement of new test IDs, although there are plans to implement some APIs that would be useful for this purpose in the future. The only methods currently available are to manually enter new test ID inventory on your system or use the spreadsheet export on the AVAILABLE TESTS panel in the iTEP administration system to obtain a csv or xls file of your available test IDs, then upload that to your system and reconcile the test ID inventory.

In iTEP's system an Available test ID can be "Assigned" to a candidate. This is accomplished by populating the test field "Assigned-to". When a test ID has data in this field it will not be considered available for use, even though the status of the test is still "Ready" the screen below shows what this looks like in iTEP's administrative system:



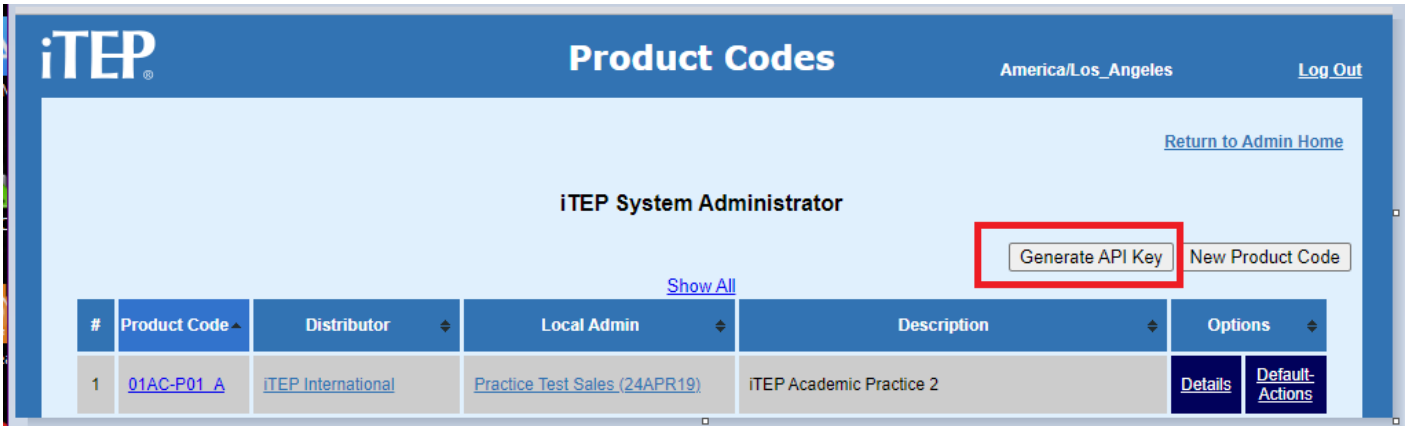
In the above example only the test ID **123982B8AP** is available to be used. Notice also that the Assigned to information is in JSON format. This is not currently a requirement if executing manual procedures in the iTEP administration system, but is required for API use. The information in the Assigned-to field is completely user-definable, however it is strongly recommended that at a minimum the candidate's name, Email, and some

identifying information on your system is included in the event that any troubleshooting needs to be executed.

There are 2 API calls that are commonly used to obtain a test ID. The request body and submission parameters of both calls are the same. The difference between the 2 APIs is one of them assumes you are going to transmit the test ID and instruction to the candidate while the other sends the test ID and test material to the candidate directly from iTEP's administration system.

Generating an API Key

In order to call an iTEP API you must first generate an API key. This task is performed on the PRODUCT CODE screen of the iTEP Administration system.



https://www.youtube.com/embed/O4BcmZegqHw?si=FNb7rlc0JS1at3Xu

Construct the API call to obtain a test ID

The next step in obtaining a test ID is to construct an API call to iTEP's administration system. With the lone exception of a header value the calls for the two previously described cases is the same. Below is a description of the fields you either need or can use in the request body.

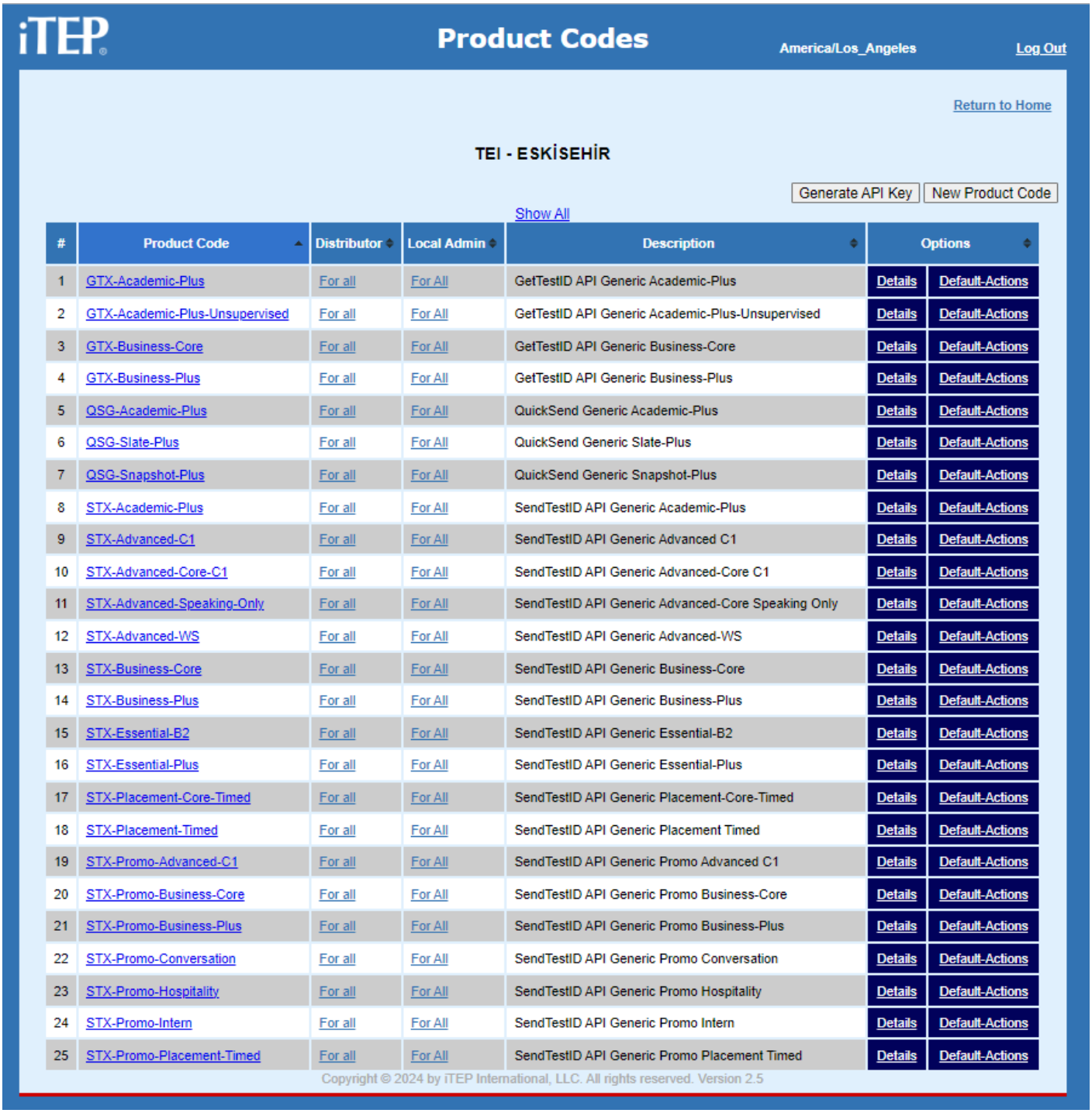
Field Name	**Field Value**	
product-code	*{product code value defined in itep administration system}*	**REQUIRED**

email	{*candidate Email address}*	**REQUIRED for "STX" API calls**
bcc	{*bcc Email address}*	
track	*{tracking code assigned by iTEP}*	**REQUIRED for results reporting**
assigned-to	{JSON - *populates test "assigned-to" field}*	
registration	{JSON - *populates test registration fields}*	

product-code

Product codes tell the iTEP administration system what to send and how to send it. iTEP has defined generic product codes that meet the needs of most users, and recommends using them. Customized product codes can be created, but creating product codes is beyond the scope of this document.

Product codes can be seen on the PRODUCT CODES screen in the iTEP administration system.



With regards to API calls there are two groups of product codes. One group starts with "GTX" and defines the process where an available test id is obtained and returned to the caller. The other group start with "STX" and defines the process where an available test id is obtained, **SENT TO THE CANDIDATE**, and returned to the caller. The different product codes in each groups are for different test types.

So, for example, the product code GTX-Academic-Plus will find an available test ID for Academic-Plus and return it to the caller whereas STX-Academic-Plus will find an available test ID for Academic-Plus, **SEND AN EMAIL WITH**

THE TEST ID FROM iTEP TO THE CANDIDATE, and return the test ID to the caller.

email

For STX calls this is the Email address to send the test ID and related materials to

bcc

For STX calls a copy of the Email sent to the candidate is also sent to this address. iTEP automatically sends a copy to its own archive. This is not required, but highly recommended, especially if you are planning on providing tier 1 support for your candidates.

track

This is a code that is assigned by iTEP. It tells the iTEP system to make a webhook call to your endpoint when the test has been graded and optionally when the test taker has finished the exam. The optional call gives the user the ability to track exams that have been finished, but still need to be graded. Note that the customer must supply iTEP with the URL of their endpoint, as it is not user configurable.

assigned-to

The assigned-to field is a JSON string that is assigned to the assigned-to field of the selected test ID. A test ID is considered available to use if its status is "Ready" and the assigned-to field is not populated. It is recommended to provide at a minimum the candidate's name, Email address, and a unique identifier from your system. Additional information may included as needed. An example would be an expiration date. Several customers assign an expiration date after which it is assumed the candidate is not going to take the test. The test can then have a new passwords assigned and made available again for a different candidate. There are plans to automate this process in the future.

registration

the registration field is also a JSON string. These fields are used to pre-populate fields on the registration form that is associated with the test. The registration forms are semi-parametric, which allows names of fields on the screen to be different than the name of the database column it is stored in. Constructing registration forms is beyond the scope of this document, but must be considered when setting these fields. There are specific fields that can be set. Any field defined in the call

The valid field names are listed below. Most are intuitive. More details will be added at a later date.

IMPORTANT!

Registration fields are designed to store simple text strings that do not contain any single or double quotes, non-printable characters, or complex data structures such as JSON or XML objects. Assigning a value other than a simple string to a registration field will have unpredictable results.

Field Name	Notes
Last_Name	
First_Name	
Middle_Name	
DOB_Month	number between 1 and 12 Providing an invalid Month will result in an error: Out of range Birth Month [value].
DOB_Day	number between 1 and 31 Providing an invalid Day will result in an error: Out of range Birth Day [value].
DOB_Year	number greater than 1900 Providing an invalid Year will result in an error: Out of range Birth Year [value].
Email	
Contact_Phone	
Address	
City	

Birthday	M/D/YYYY or D-M-YYYY M should be a number between 1 and 12 D should be a number between 1 and 31 YYYY should be a number greater than 1900 Not providing 3 numbers will result in an error: Unable to parse Birth Date [value]. Providing an invalid Day will result in an error: Out of range Birth Day [value]. Providing an invalid Month will result in an error: Out of range Birth Month [value]. Providing an invalid Year will result in an error: Out of range Birth Year [value].
Country	
Nationality	
OIN	**O**fficial **I**dentification **N**umber This is usually a government issued id such as a driver's license or passport.
OIN_Country	
OIN_Type	
Language	
Center	
Gender	Valid values (case insensitive): Male Female Providing an invalid Gender will result in an error: Invalid Gender [value].

Education	Highest education level completed Valid values (case insensitive): Middle School High School Vocational School Bachelor's Degree Master's Degree Doctorate Providing an invalid Education will result in an error: Invalid Education [value].
Taken_Test	
DOT_Month	
DOT_Day	
DOT_Year	
Location	
Admin_Test_Location	
Agree	
Honest	
Applying_To_School	
Apply_School_Type_1	
Apply_School_Name_1	
Apply_School_Other_1	
Apply_School_Type_2	

Apply_School_Name_2	
Apply_School_Other_2	
Apply_School_Type_3	
Apply_School_Name_3	
Apply_School_Other_3	
Field_Of_Study	education field of study Valid values (case insensitive): Other Business Communication Education English Engineering History Hospitality Other Languages Mathematics Medical Music Politics Recreation Science Providing an invalid Study Field will result in an error: Invalid Study Field [value].

Location_Country	
Location_Region	
Location_Name	
trSchoolLevel	
trReferral	
trExternal_ID	The value of this field is returned in the results field **OID**
trCampus	
trCollege	
trDepartment	
trDegree	
trStatus	
trAttribute	
trStudent_Number	
trBatch_Number	

Putting it all together

Here is an example of the request body:

{"product-code": "STX-Academic-Plus","email":"tfatout2@proton.me","bcc":"twif_bam@yahoo.com","track": "RC","sequence":3,"assigned-to": {"expire": "2024-09-01","student-name": "Mary Poppins","customer-candidate-id":9810283,"sequence":3},"registration": {"Last_Name":"Poppins","First_Name":"Mary","Email":"m.poppins@disney.com","Birthday": "1955-07-08","OIN": "123-456-7890","OIN_Country": "US","OIN_Type": "1","Address":"10119 Popular Lane","City":"Los Angeles","Field_Of_Study":"Biology"}}

Notes:

- Do not send "pretty" JSON with line feeds, carriage returns and other whitespace.
- Assume everything is case-sensitive

In addition to the request body a special HTTP header is required. This header serves several purposes:

1. The header key directs processing on iTEP's server
2. The header value contains a HMAC-SHA256 digital signature
3. The digital signature confirms the integrity of the request body
4. The digital signature identifies the sender
5. The digital signature prevents replay attacks

Header Values	
X-SendTestID-Signature	Obtain an available test ID and perform fulfillment using **iTEP's** infrastructure
X-GetTestID-Signature	Obtain an available test ID and perform fulfillment using **YOUR** infrastructure

For those that would like to understand the math involved in the calculation see the video below. Note that most languages and software development infrastructures have libraries that perform the calculations for you.

https://www.youtube.com/embed/orIgy2MjqrA?si=dFCO_k168s4KKfE5

It is unlikely you will need to do the math yourself. However, during development you may, want to check the signatures you generate. To do this iTEP suggests using an online tool to generate signatures to check against your integration process. Here is a video of using an online tool to create a HMAC-SHA256 signature:

https://www.youtube.com/embed/sR1BD0gZcjs?si=NamCPsorBjjMPXsD

This tool demonstrated in the video is located HERE

NOTE: The supplied signature must be in base64

Now you have all the required components to make the API call to get a test ID.

The endpoints for all API calls are:

SANDBOX	**https://sandbox.itepwest.com/requests/ecommerce_webhook.php**
PRODUCTION	**https://www.iteptest.com/requests/ecommerce_webhook.php**

If executing a STX product code the iTEP fulfillment system will deliver the test materials to the candidate as per the instructions encoded into the specific product code (usually by email). Note that a "GTX" code does NOT deliver test materials to the candidate.

In either case the caller will receive a response containing the following fields:

Field Name	Notes
test-record-id	unique numerical id assigned to each test only practical use is for troubleshooting integration issues.
test-id	test id that identifies test for candidate login purposes
password	password used for candidate login purposes
assigned	value the test has in its "assigned" field
track	numerical id of track code assigned to this test.
status	HTTP status code
message	API status message

Sample response:

{"test-record-id":"1585286","test-id":"19B02327AP","password":"72891","assigned":"{\"expire\":\"2025-12-31\",\"student-name\":\"Mary Poppins\",\"customer-candidate-id\":9810283,\"sequence\":3}","track":"0","test-url":"https:\/\/twf2.itepwest.com\/test?p=0ad68c3c92a171334f4137d14787da3bAxz7q","status":"200","message":"Process completed"}

Currently we do not provide examples for specific development environments, however the video below demonstrates execution of all the steps of a fulfillment API call using the Postman utility.

https://www.youtube.com/embed/vNFEPanau_c?si=5oGGZXi3pqngePUA

Revision #48
Created 2024-07-29 17:10:12 UTC by Tim Fatout
Updated 2026-04-28 12:51:06 UTC by Tim Fatout