Our learning team would like to consume data from a Learning Management System. They would like to see which users enrolled into which courses.

1

Using the following LMS documentation https://canvas.instructure.com/doc/api/, can you please explain how you could retrieve course data for users?

2

What are the development steps for the ETL?

3

3.What other data could be interesting to retrieve and why?

	PROCESS	STEPS	CODE
1 DATA RETRIEVAL	AUTHENTICATION on CANVAS	1. generate a client ID and secret in the Site Admin account 2. use client_id and client_secret to obtain the auth code 3. exchange code and credentials for the final access token 4. Store "access_token" as Bearer auth in Variables	1. Canvas id and secret : https://community.canvaslms.com/t5/Admin-Guide/How-do-I-configure-an-LTI-key-for-an-account/ta-p/140 2. GET https://<canvas-install-url>/login/oauth2/auth?client_id=XXX&response_type=code&redirect_uri=https://example.com/oauth_complete&state=YYY&scope=<value_1>%20<value_2>%20<value_n> 3. POST request {{URL}}/login/oauth2/token?grant_type&client_id&client_secret&code 4. save token as bearer auth in environment variables
	API endpoints LIST	3 1. identify {{URL}} endpoint to store in variables 2. list all subsequent endpoints to fetch : 2a. who are the users in my account **2b. what are the user courses** **2c . maybe more analytics like user progress** **2d. or criteria for completion** 3. run queries on Postman with correct endpoints alongside the environment variables stored above	Queries on Postman: 1. {{URL}} = https://canvas.instructure.com 2a. GET {{URL}}/api/v1/accounts/:account_id/users **2b. GET {{URL}}/api/v1/users/:user_id/courses {param : "user_id" = user}** **2c. GET {{URL}}/api/v1/courses/:course_id/users/:user_id/progress** **2d. GET {{URL}}/api/v1/courses/:course_id/completion_requirements** **more endpoints :https://canvas.instructure.com/doc/api/all_resources.html**

2 ELT	RESPONSES EXTRACTION & LOAD	on Python
		1. extract and load json responses
		2. use json and request library
		3. run the query for endpoints specified above
		
		to Tableau
		1. create a Web Data Connector
		2. specify endpoint, parameters, and authentication methods
		3. run
			import requests
			
			url = 'web address'
			params = {'key':'value'}
			r = requests.get(url = url, params = params)
			response = r.json()
			
			import {Fetcher, FetchUtils} from '@tableau/taco-toolkit/handlers'
			import {API_TOKEN} from "../constants";
			export default class DataFetcher extends Fetcher {
			 async *fetch({handlerInput}) {
			    const headers = { 'Authorization': `Bearer ${API_TOKEN}' };}}

	DATA TRANSFORMATION	1. Explore results
		2. Flatten json results
		3. merge results
		4. filter and remove
		5. handle NULL, duplicates, dtypes()
		
		1. explore the json fields relationships
		2. flatten the data to exploit
		3. with pivot, split, custom calculations
		4. filter and remove
		5. handle missing values, duplicates, data types
		6. merge data if many json responses
			=> example for python looping through a lit of users
			base_url = 'https://canvas.instructure.com/api/v1'
			api_key = 'YOUR_API_KEY'
			
			user_ids = ['user1', 'user2', 'user3']
			
			# iterate
			  enrollments_url = f'{base_url}/users/{user_id}/enrollments'
			  headers = {'Authorization': f'Bearer {api_key}'}
			  response = requests.get(enrollments_url, headers=headers)
			  if response.status_code == 200:
			    enrollments = response.json()
			    # add data cleaning
			  else:
			    print(f'Error user {user_id}, code: {response.status_code}')

BI	NEW METRICS CALCULATION	1. aggregate, group, derive new fields 2. transpose rows/ columns or split to render visuals	to be discussed