DK Http Client

# Introduction

This document records the development process of the DK Http Client project related to Digikey. Digikey is a distributor of electronic components. Digikey has a large catalog of over 8 million products from over 800 manufacturers on this website: [www.digikey.com](http://www.digikey.com). Digikey made the product data publicly available for download. This data could be useful for market research, product research, machine learning and other purposes.

DK Http Client is an http client application built for acquiring the product data from Digikey’s site. It will be written in Python 3 and utilizing Python libraries including: selenium, bs4, requests, etc. While we want DK Http Client to run fast, we will be mindful of controlling the request rate so that we don’t inadvertently or purposefully launch a denial-of-service (DOS) attack.

The data we are interested at downloading is in a form of CSV files. For instance, open this page: <https://www.digikey.com/products/en/audio-products/accessories/159>, and click the “Download Table” icon. The downloaded CSV file contains the same product data displayed on that webpage. The goal DK Http Client is to download all the CSV files of interest, i.e. navigating the desired pages and clicking the “Download Table” icon. For the remainder of this document, the entire collection of data in the above-mentioned CSV files will be called the *product index*.

# Acquiring Preliminary Datasets

Before we are able to download the product index, we need to first acquire some preliminary datasets. The two sets of data we require are *Digikey Supplier Data* and *Digikey Product Categorization Data*. The mechanism of how these datasets are used to download the product index will be explained later (#ToDo). For now, we will first focus on how to acquire these datasets.

## Digikey Supplier Data

All of Digikey’s suppliers are listed on this “supplier center” page: <https://www.digikey.com/en/supplier-centers>.

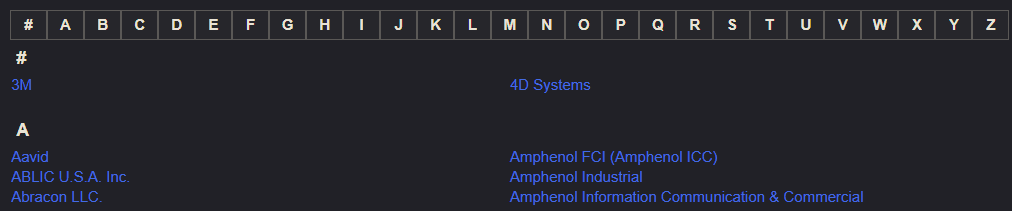


Figure 2‑1: Snippet Digkiey's supplier list

Clicking on the supplier’s name leads to that supplier’s main page on Digikey, such as <https://www.digikey.com/en/supplier-centers/3/3m>.

Each supplier has a unique code which can be used to identify that supplier on the query string of a product search URL. The supplier code can be found by clicking one of the product listing:

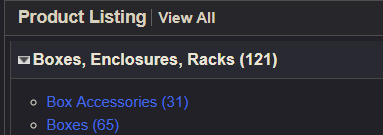


Figure 2‑2: Click product listing to find supplier code

In the case of 3M, the supplier code is “19,1067,8002,1717” which is the string after “v=” in this URL:

<https://www.digikey.com/products/en/boxes-enclosures-racks/box-accessories/595?v=19,1067,8002,1717>.

One side note, not all suppliers on the “supplier center” page are active suppliers. Due to merger or acquisitions, some suppliers are not active, i.e. they do not have a supplier code and products listed under their name. One example is Altera which has been acquired by Intel.

In Altera’s page: <https://www.digikey.com/en/supplier-centers/a/altera>, there are no product listings.

Now, Altera appears as Intel FPGA at <https://www.digikey.com/en/supplier-centers/i/intel>.

In summary, the goal is to acquire three pieces of data we want to acquire for each active supplier: supplier name, supplier URL and supplier code.

## Coding dk\_supplier.py

<dk_supplier.py> is the Python script that will accomplish the goal in section 2.1.

### Importing Libraries

Start by importing the libraries.



Selenium webdriver will be the tool to parse the HTML source of “supplier center” and acquire a list of all supplier names and URLs including non-active suppliers. Later the non-active suppliers will be filtered off.

The pandas DataFrame object will be used to organize the tabular supplier data and export the data to CSV, Excel or SQL as needed.

The “re” or regular expression library will bed used to extract the supplier codes from product URLs.

The “time” library will be used to add delays to HTTP requests so that the code doesn’t act as a DOS attack.

The “os” library will be used to make file paths.

## Digikey Product Categorization Data

Open this page on your browser: <https://www.digikey.com/products/en>. Digikey categorized the products into to two tiers of hierarchy. As shown by Figure 2‑1, “Audio Products” is the upper hierarchy which represents a more general and broader range of products. Those listed below “Audio Products” belong to the lower hierarchy which represent more specific ranges of products.

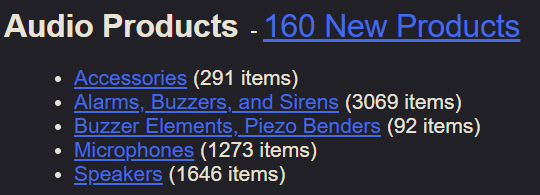


Figure 2‑3: Example of product categorization hierarchy

For the purpose of this document, the upper hierarchy is named *product group* and sometimes denoted as *pg*. The lower is named *sub product group* and sometimes denoted as *spg*.