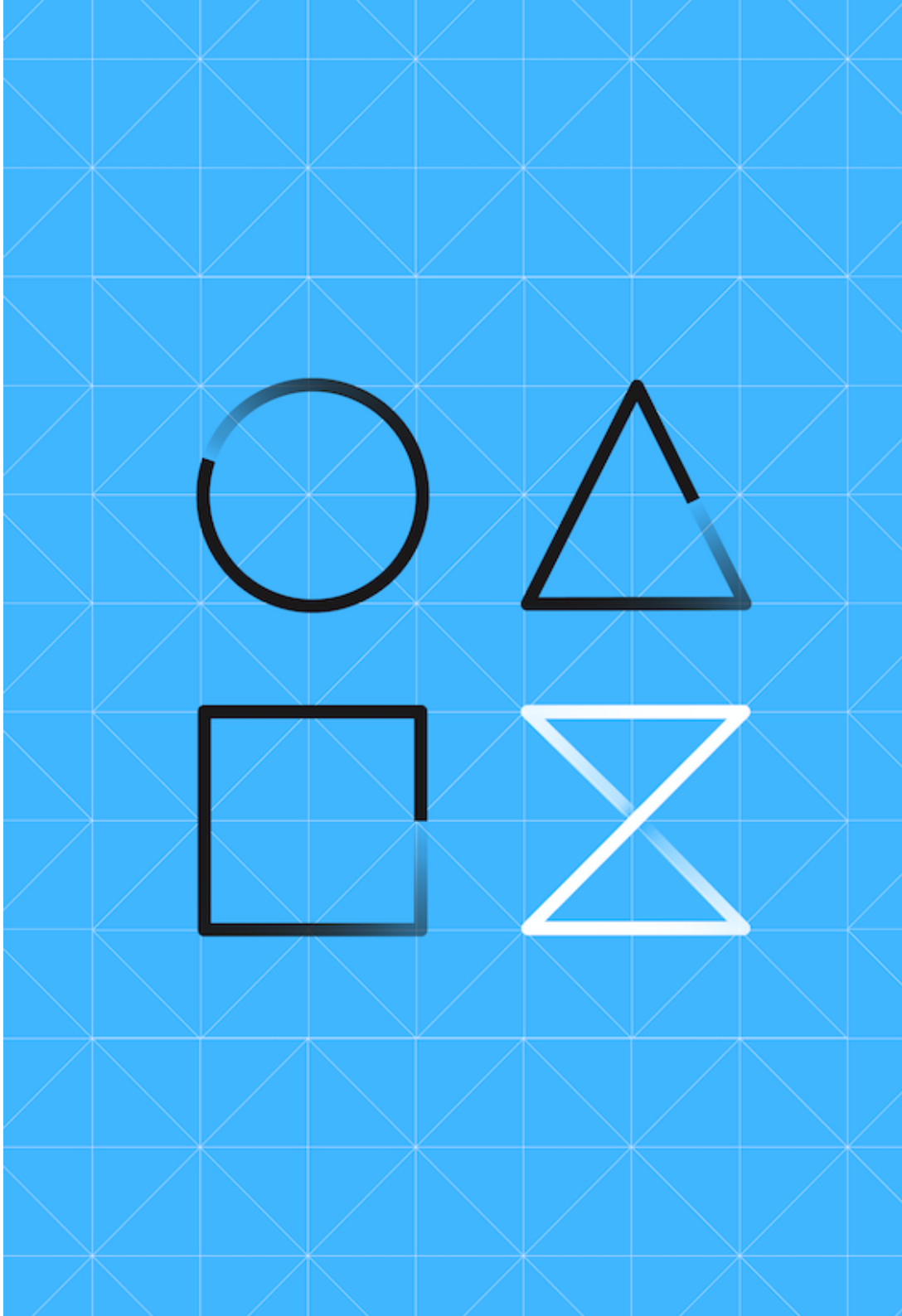
FILES1

Not just Members Only



HOW DATA GETS DOWN ON Z/OS

The Challenge

As you've seen, data gets handled a little differently on z/OS. This isn't just IBM Z trying to be difficult. When records are organized and kept in a structure that lines up with how they will typically be read and written, applications can run faster, and with less confusion.

In this challenge, you get your hands on some data sets and members, and then learn how to copy, rename and even delete them like it's no big deal.

Before You Begin

If you have access to the IBM Z system, and have VSCode set up, you should ready to start.

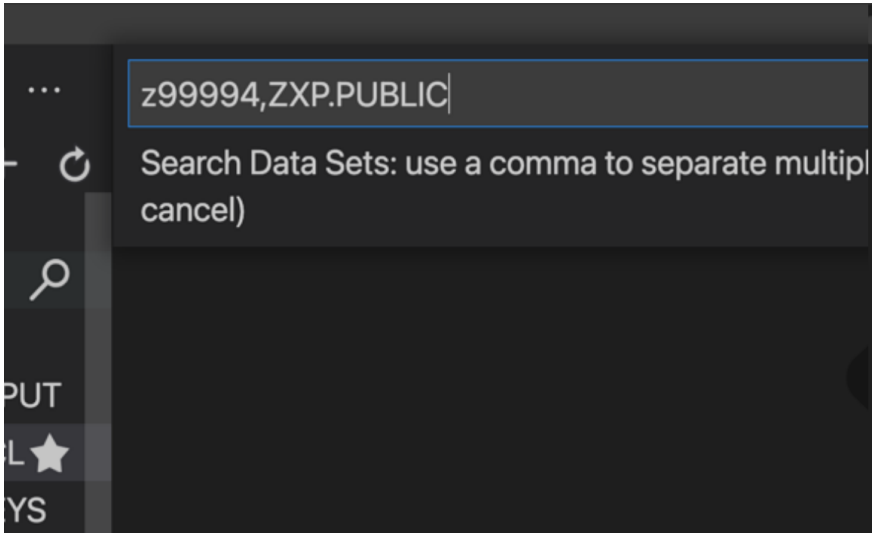
The only technical challenge you need to have completed before this one is VSC1.

Investment

Steps	Duration
12	45 minutes

1 REFINE YOUR FILTER

We’re going to expand our filter to see even more data sets.



Click on the Magnifying Glass to the right of your ZOWE connection profile name `zxplore`, and enter the following:

`Zxxxxx,ZXP.PUBLIC`

(Please make sure to enter your own userid here, not `Zxxxxx` or `Z99994`!)

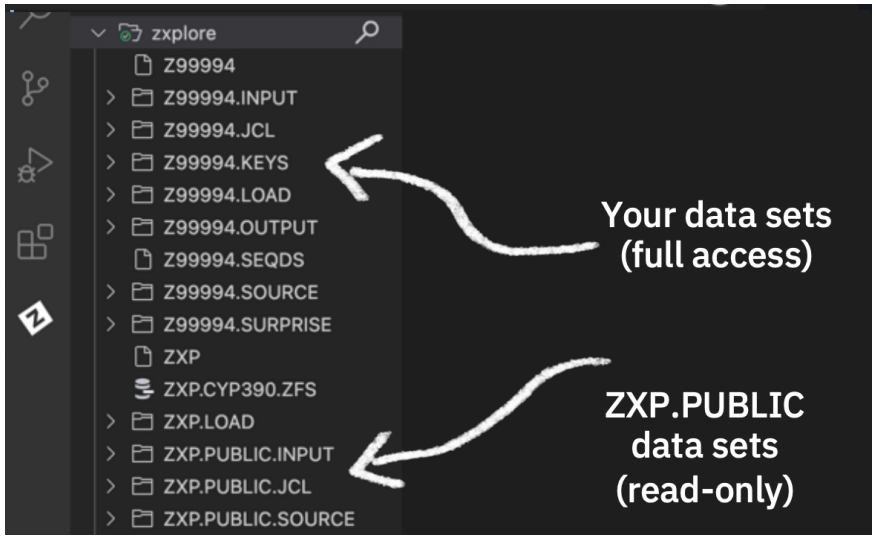
Also, make sure to note the comma before ZXP and the period after it. These are important.

Save the filter, and the dataset list should update.

2 YOU CAN LOOK …

You have now set the filter to see not only your own Zxxxxx data sets, but the data sets starting with ZXP.PUBLIC as well.

Nice!



Everyone has read-only access to the same ZXP.PUBLIC data sets, and in future challenges, you will often use these to copy from.

If you get an error when you try to save or edit a dataset you have opened from ZXP.PUBLIC, that’s because you are limited by the read-only access.

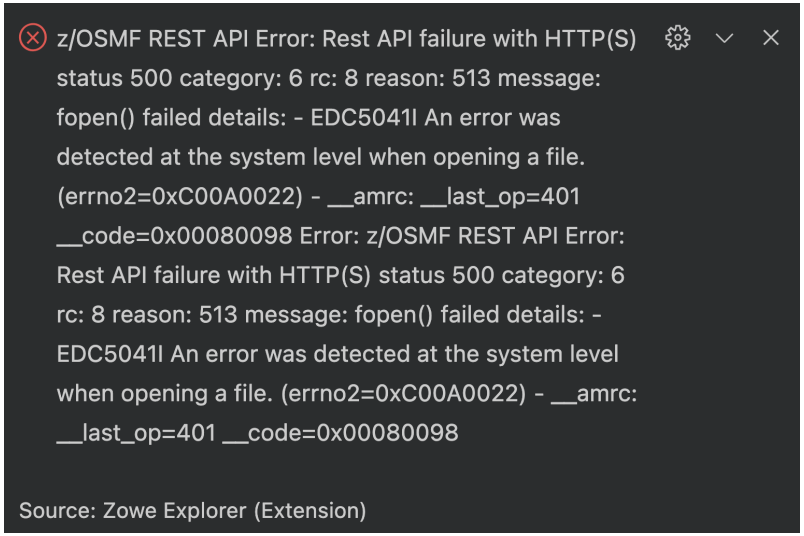
Typically, whenever you need to edit a data set, that will happen within your own Zxxxxx data sets.

Keep that in mind, and don’t be surprised or worried when trying to edit a ZXP.PUBLIC data set or member.

FILES1|230924-0030

Note that at this stage, there are no datasets associated with your userid; you will be creating some shortly.

You *will* see what looks like a dataset in the list with a name that matches your userid – this is **not** a dataset, and if you try and open it, you will see an error from VSCode like this:



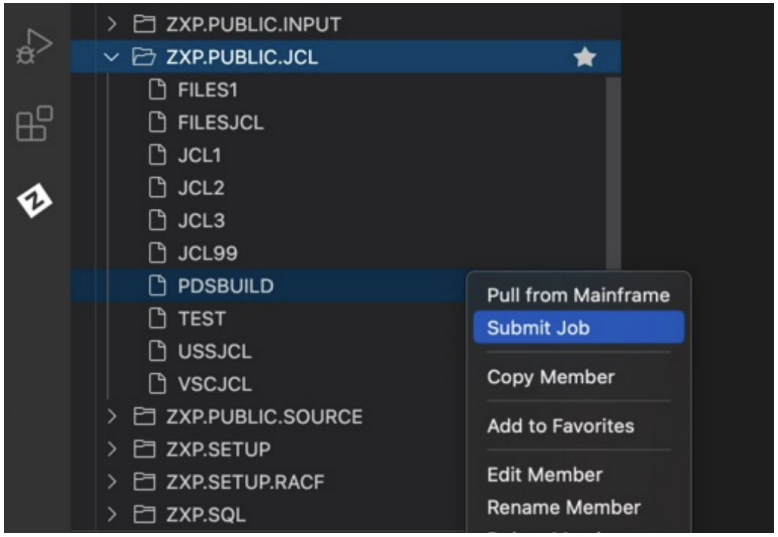
Don’t be alarmed! You did nothing wrong, and this is not a problem that needs fixed!

3 LET'S MAKE SOME DATASETS

There is some code written specifically to build most of the data sets and members you will use for this challenge. You will find that in the ZXP.PUBLIC.JCL dataset.

The ">" symbol next to the dataset indicates that it is partitioned dataset and contains members.

Open it up and look for a member called `PDSBUILD`. Right-click on that and select `Submit Job`.



By the time you're finished reading this, all the files required to get started will be in place, so let's keep moving …

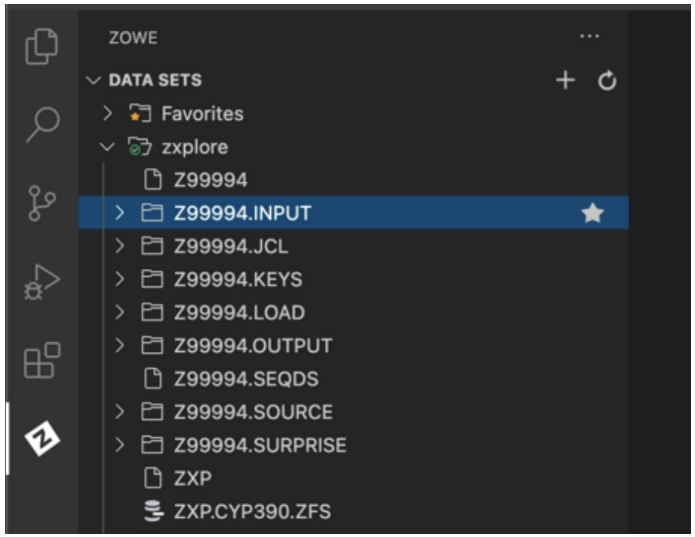
FILES1 | 230924-0030

4 SHOW US WHAT YOU GOT

In your VSCode “DATA SETS” view, close and reopen your “zxplore” search view, by clicking on the “arrow” icons.

Look for your Zxxxxx.INPUT data set. This is a Partitioned Data Set (PDS) and these are members inside it, just like ZXP.PUBLIC.JCL.

Look through them, but don’t do anything yet.



Note: The set of files you’re looking at were generated just for you, so if you’re working with other people, don’t be alarmed if you see different contents than your friends.

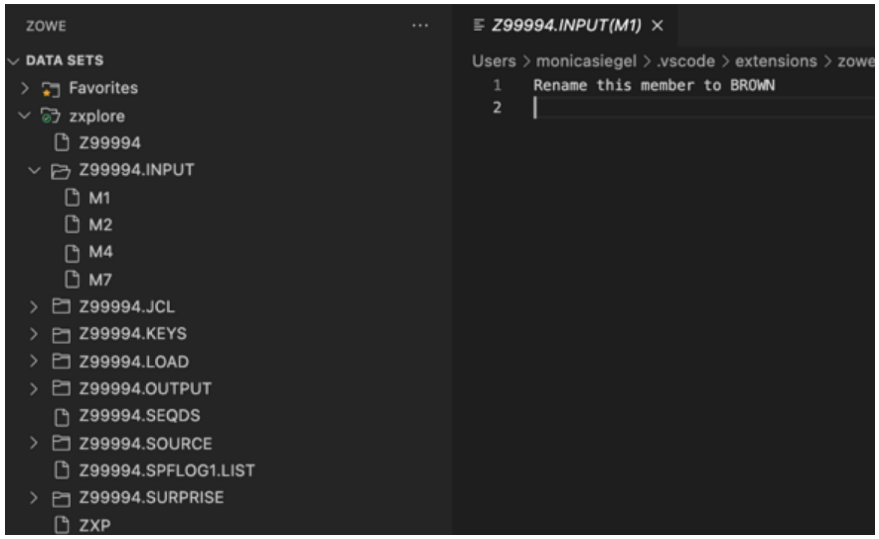
5 IF YOU NEED TO START OVER …

If at any time you want to reset back to Square 1 (the beginning) for this challenge, just submit the **PDSBUILD** JCL from Step 3 again.

6 RENAME IN THE MAINFRAME

To view each member of your `INPUT` dataset, you will need to click on the member to see its contents load in the editor on the right side.

One of the members will contain text directing you to rename it, and the name you should rename it to.



To rename a member, just right-click on it, and select `Rename Member`. Then simply enter the new name in the dialog box that pops up at the top of your VS Code window.

”WHY CAN’T WE JUST HAVE FILES AND FOLDERS AND CALL IT A DAY?”

Different applications access their data differently.

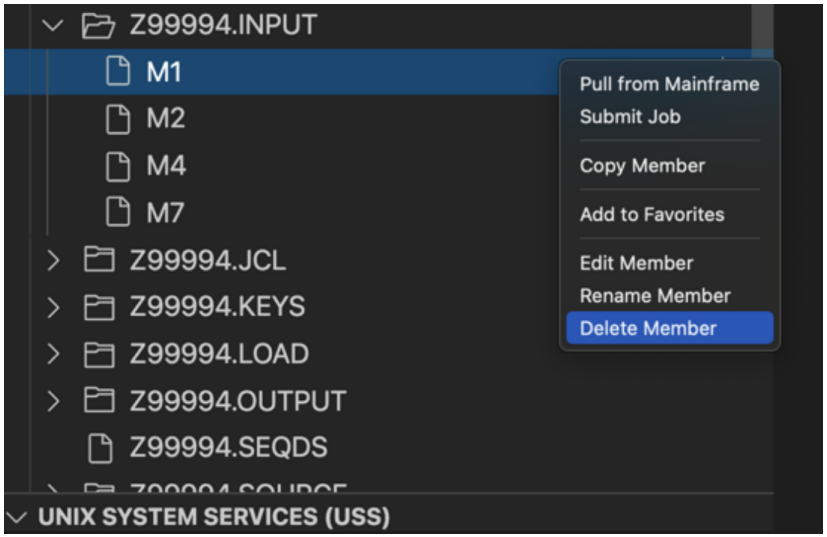
Some applications read thousands and thousands of records of customer data one right after the other. They will always process the records from top to bottom, in the order they are given, and for these applications, Sequential Data Sets provide the best format so that information can be processed as fast as possible.

Other times, an application doesn’t know what data will be needed next and having members within a Partitioned Data Set allows the flexibility needed to have everything available in any order it is needed.

There are other data sets you’ll use as well, but for now, just focus on Partitioned and Sequential.

7 MEMBER DELETION

In a similar way to the renaming step, track down the member in your `INPUT` dataset that is directing you to delete it. Right-click on that one and select `Delete Member`.



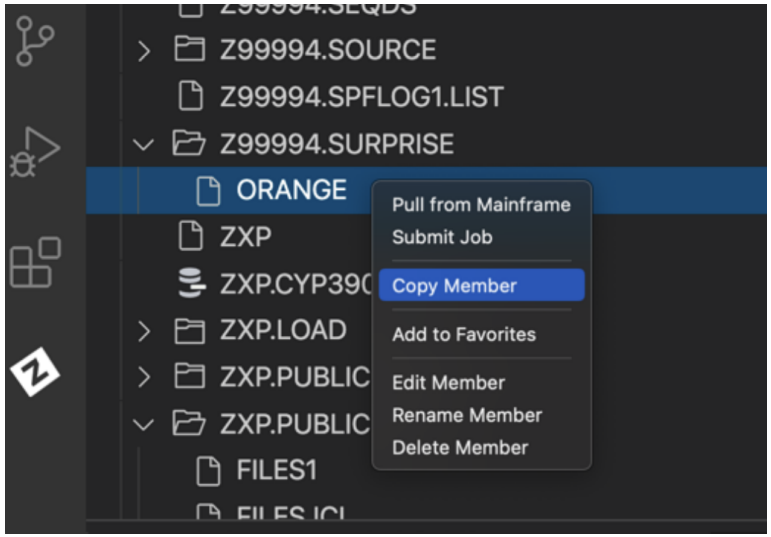
Confirm that you really do want to delete it in the dialog box that pops up, and *Poof!*, it's gone - just like that!

8 ROGER - COPY THAT

Look in your `SURPRISE` data set, because *SURPRISE!* there is yet another member in there for you.

Open it up and read the contents. It will be asking you to copy it and then paste it into your INPUT data set.

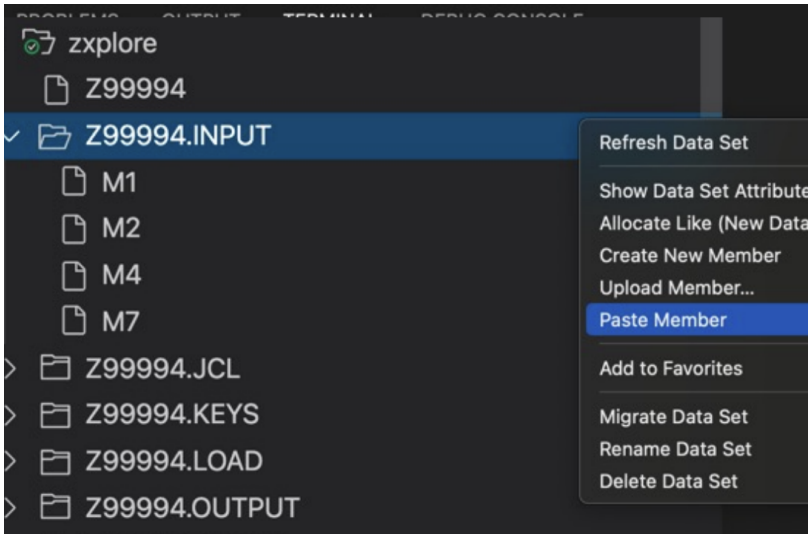
Start by Right-clicking on the source data set (the one you found in `SURPRISE`) and select `Copy member`.



Now you are all primed and ready to paste it in the next step.

9 THE ACE OF PASTE

With the source data set member safely copied, Right-click on your `INPUT` data set and select `Paste Member`. You will be prompted for a name.



This is a chance for you to move the contents of a file from one place to another with a new, unique name; but in this case, we want to keep the same name, so just call it what it originally was in the SURPRISE data set.

(*hint*: it will be a color, so you will end up with two `INPUT` dataset members with the names of colors).

”THIS ALL SEEMS KIND OF SIMPLE. AM I MISSING SOMETHING?”

Datasets and members aren’t the trickiest thing in the world, but there are a few peculiarities that some people get confused by especially for those who are new to working on computers.

Modern touchscreen interfaces hide away a lot of the complexities that professionals need to be aware of. For example, try right-clicking on a dataset and select “Show Data Set Attributes”.

In addition to there being Sequential and Partitioned datasets, datasets have a number of attributes which can be set to obtain the best - performance - security - scalability

You will take the default values for most challenges, but a true IBM Z Systems Programmer will know the best values for all of these fields.

Just something to keep in mind while working your way through these challenges.

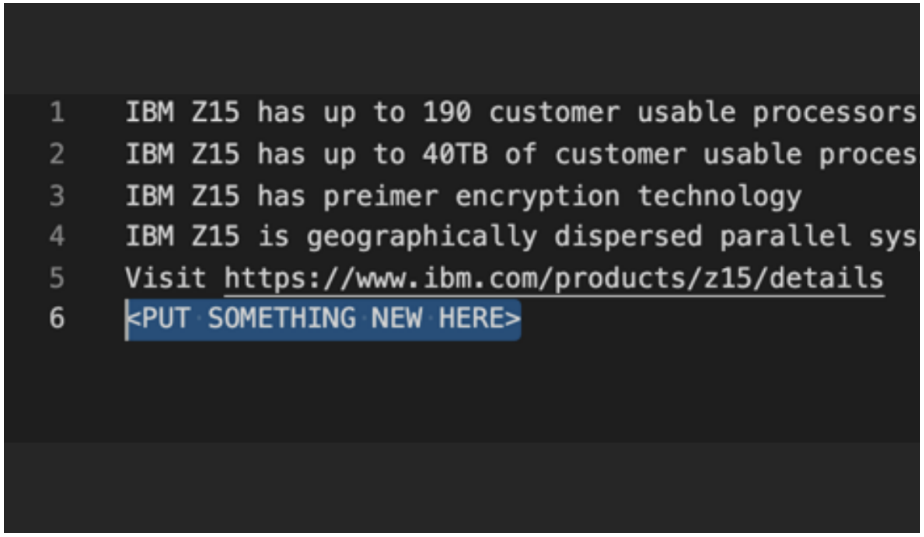
FILES1|230924-0030

10 SEQUENTIAL ACCESS

You been using the INPUT Partitioned Data Set (PDS) to hold the members for this challenge.

Now let’s look at your **SEQDS** dataset.

Instead of separate members, this data set just contains its own records. This is known as a Sequential Data Set.



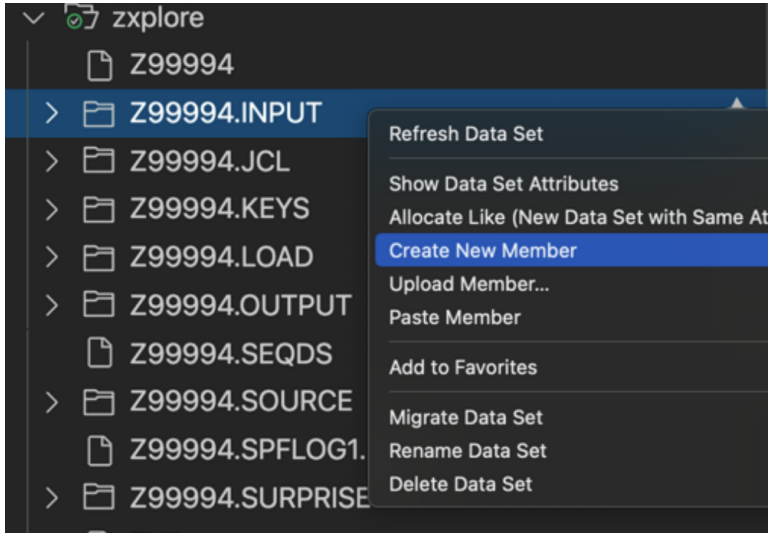
When you open it up, you see that each record is represented as a line in the dataset. Adding a new record is as simple as adding a new line and entering text.

For this step, we want you to do just that; + add a new line (record) + say hello, tell us how the weather is, what your favorite cartoon is + save the file

And you’re done with this step.

11 MAKE YOUR OWN MEMBER

Right-click on your INPUT dataset and select `Create New Member`.



Give it the name of `MYNEWMEM` when prompted for a name.

Creating a new member within a data set is as simple as that.

If you get an error, make sure you're not adding any additional spaces or punctuation, as member names can only be 1 to 8 characters, made from - letters (A-Z) - numbers (0-9) - @ - # - $

12 RECORD YOUR VICTORY

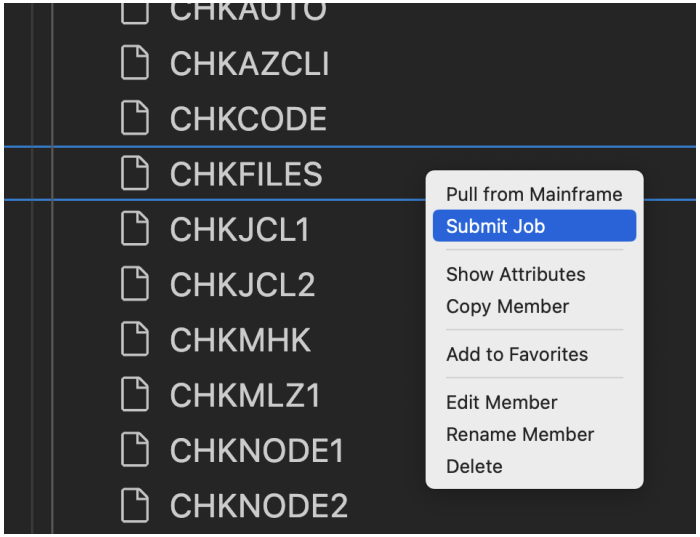
At this point, you should have 5 members in your **INPUT** data set.

- two that were there originally
- one that you renamed
- one you created
- one that you copied from **SURPRISE**

It should NOT contain the member which you were instructed to delete.

Your **SEQDS** data set should also have an additional record with your input.

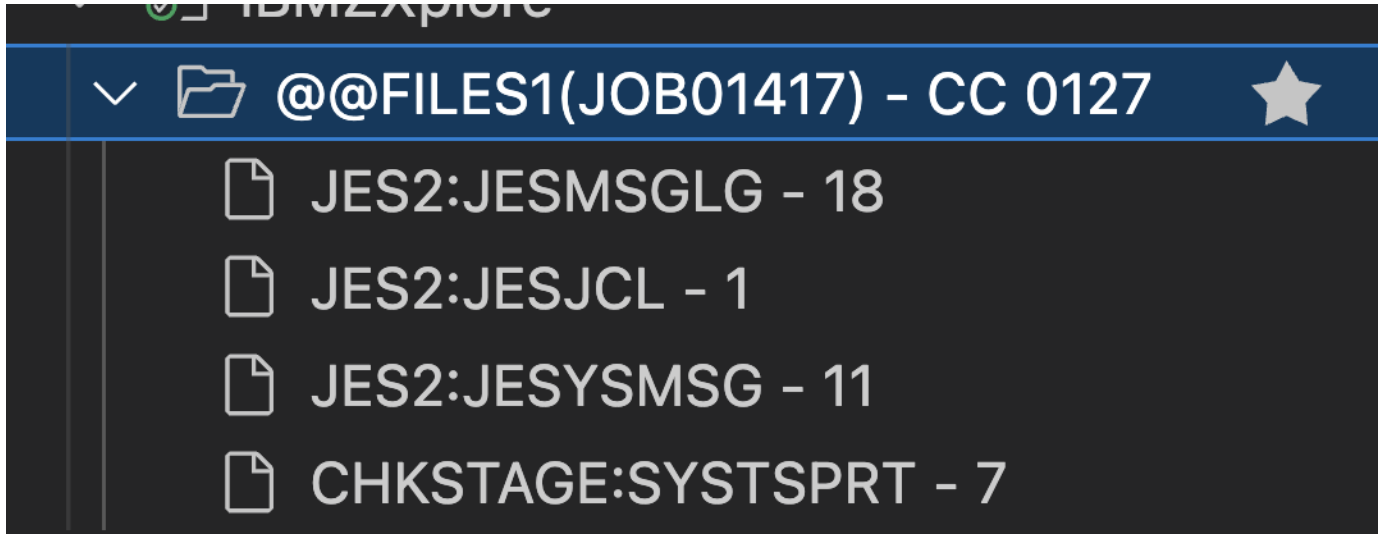
Double check your work, then find the **CHKFILES** member in **ZXP.PUBLIC.JCL**



Right-click on it and select Submit Job to hand in your work.

If you have completed all the steps correctly, the validation job will complete with a “completion code” (CC) of 0000.

If you have missed a step or two, or not implemented an instruction correctly, the validation job will return a CC of 0127 –



this means you will need to review the previous steps, make any needed corrections, and submit **CHKFILES** again.

Nice job - let’s recap	Next up …
You’re getting really handy with those data sets and records. Just keep in mind that there are many different ways of storing data, and part of your job is to figure out whether the situation calls for a Partitioned Data Set, a Sequential Data Set, a Member within a Partitioned Data Set, or something else. Properly optimized data is what keeps applications running fast. Click to watch the Data Sets and Members video again at any time.	It’s one thing to follow steps, it’s another to understand. In the next two challenges, we’ll get to the bottom of exactly what these sets and members are, plus dive a bit deeper into JCL. You’re making great progress, let’s keep it all going.