Joel Kershner (j031k422)
11/6/2023
ITEC746
Instructor Frank Moley

REST Log management lookup list management and transformation

Table of Contents

Document revisions

9/11/2023 – Project Topic

9/18/2023 – Project Domain

10/2/2023 – ER Diagrams

10/16/2023 – Added Section – DDL

10/24/2023 - Corrected DDL

11/06/2023 – Added DML, added extra semicolons in DDL

Description:

This project is intended to create a REST API to ingest text lists and CSV lookup table files, add meta data to them for revision control and description. They must also be searchable, exportable in different formats and some basic text-file transformations like sorting and returning unique values.

Entities:

Entity 1

InputFile
Contains the original lookup file

FileGuid	OBJECT	Unique Identifier for an input File.
FileContents	LONG STRING	The contents of an input file

Entity 2

InputFileFormat
Format is an input file format supported by the application.

InputFormatGuid	OBJECT	Unique identifier for the input file format
Name	STRING	Simple name of file format
Description	MEDIUM STRING	Text description of file format
FileExtension	STRING	A file extention
FileMIMEType	STRING	File format Mimetype

Entity 3

DataOutputFormat
Format is an output format and/or transformation supported by the application. Depending on the output format it may generate multiple cache documents for a single CacheMetadata entry for key value searches.

OutputFormatGuid	OBJECT	Unique Output data format identifier
Name	STRING	Output format name
Description	STRING	Text Description of output data format
FunctionName	STRING	Name of function needed to transform output
FileExtension	STRING	Output File extension
FileMIMEType	STRING	Output file MIME type

Entity 4

FileMetadata
Contains revision details on files

APIGUID	OBJECT	API that owns this File
FileMetaDataGuid	OBJECT	Unique file metadata identifier
OriginalFileName	STRING	Original filename
Name	STRING	Brief short name of file
Description	STRING	Text description of file contents
DateUploaded	DATETIME	Date and time that this file was uploaded
DateModified	DATETIME	Date and time that this file was last modified
Revision	NUMBER	Number of updates performed on this file
IsCurrent	BOOL	Is this file considered current
IsDeleted	BOOL	Has this file been tagged for future deletion
FileGuid	OBJECT	Which input file are we describing
InputFormatGuid	OBJECT	What input file format is used on this file
FileHash	STRING	What is this files cryptographic hash (SHA256)
FileCacheExpiration	NUMBER	Number of seconds from cache file creation before expiration

Entity 5

API
Contains api keys and meta data

APIGUID	OBJECT	Unique API Key ID
API	STRING	Unique API Key
OwnerEmailAddress	STRING	API Key owner email address
DateExpires	DATETIME	Date and time this API key should expire
PermissionCreate	BOOL	API has permissions to create new files, and create/add/update/delete api's and authorizations for them.
PermissionGlobalAdmin	BOOL	API key is allowed full global admin access
IsDisabled	BOOL	API key is disabled
IsDeleted	BOOL	API key is marked for deletion

Entity 6

IsAuthorizedTo
Contains relationships from API keys to Metadata & files and cache. This is optional if you don't want every user to be an administrator.

AuthorizationGUID	OBJECT	Unique authorization identifier
APIGUID	OBJECT	Which API object this authorization is for
FileMetaDataGUID	OBJECT	Which File object does this authorization apply to
PermissionCreate	BOOL	API key is allowed to create new authorizations for this file
PermissionRead	BOOL	API key is allowed to read this file and create cached versions
PermissionUpdate	BOOL	API key is allowed to update this file
PermissionDelete	BOOL	API key is allowed to delete this file

Entity 7

CacheMetadata
Describes the Cache output object

CacheMetadataGUID	OBJECT	Unique cached output identifier
CacheGUID	OBJECT	Which Cached output object this applies to
FileMetadataGUID	OBJECT	Which input file object this applies to
OutputFormatGUID	OBJECT	What output transformation and/or format this cached version uses
ExpirationTimestamp	DATETIME	When this cached version of the file expires
isDeleted	BOOL	If this cached has been updated and entry is no longer current

Entity 8

Cache
Converted lookup files in KV, CSV, TEXT LIST or JSON format.

CacheGUID	OBJECT	Cached output unique identifier.
CacheKey	STRING	(optional) Searchable key field.
CacheValue	STRING	Formatted file output.
IsDeleted	BOOL	This cached data is no longer valid

Entity 9

Search
List of search functions and transformations supported used for error and help prompts.

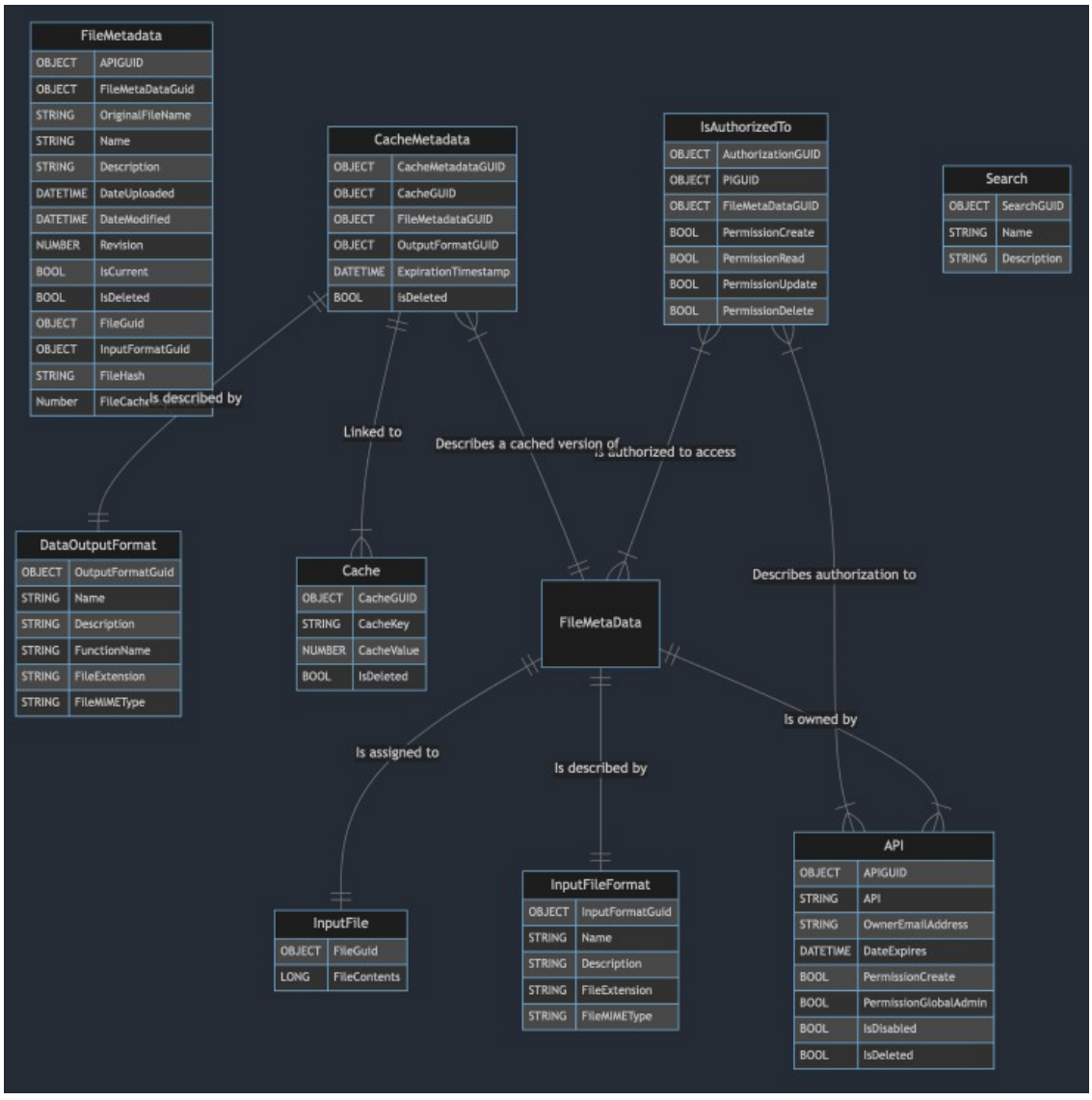
SearchGUID	OBJECT	Unique search function or method identifier
Name	STRING	Search command name
Description	STRING	Search command description

Relationships

FileMetaData has one InputFile
FileMetaData has one InputFileFormat
FileMetaData has one API
IsAuthorizedTo has one API
IsAuthorizedTo has one FileMetaData
CacheMetadata has one DataOutputFormat entry
CacheMetadata has one FileMetaData
CacheMetadata has one or more Cache entries
CacheMetadata has one DataOutputFormat

ER Diagram

FileMetaData ||--|| InputFile : "Is assigned to"
FileMetaData ||--|| InputFileFormat : "Is described by"
FileMetaData ||--|{ API : "Is owned by"
IsAuthorizedTo }|--|{ API : "Describes authorization to"
IsAuthorizedTo }|--|{ FileMetaData : "Is authorized to access"
CacheMetadata ||--|| DataOutputFormat : "Is described by"
CacheMetadata }|--|| FileMetaData : "Describes a cached version of"
CacheMetadata ||--|{ Cache : "Linked to"



DDL

```
SQL Created on PostgreSQL v14 using PgAdmin4

CREATE SCHEMA IF NOT EXISTS kershner

CREATE TABLE IF NOT EXISTS kershner."API"
(
    "APIGUID" uuid NOT NULL DEFAULT gen_random_uuid(),
    "API" character varying COLLATE pg_catalog."default" NOT NULL,
    "OwnerEmailAddress" character varying COLLATE pg_catalog."default" NOT NULL,
    "DateExpires" timestamp without time zone NOT NULL,
    "PermissionCreate" boolean NOT NULL DEFAULT false,
    "PermissionGlobalAdmin" boolean NOT NULL DEFAULT false,
    "IsDisabled" boolean NOT NULL DEFAULT false,
    "IsDeleted" boolean NOT NULL DEFAULT false,
    CONSTRAINT "API_pkey" PRIMARY KEY ("APIGUID")
);


COMMENT ON TABLE kershner."API" IS 'Contains api keys and meta data';

CREATE TABLE IF NOT EXISTS kershner."DataOutputFormat"
(
    "OutputFormatGuid" uuid NOT NULL DEFAULT gen_random_uuid(),
    "Name" character varying COLLATE pg_catalog."default" NOT NULL,
    "Description" character varying COLLATE pg_catalog."default" NOT NULL,
    "FunctionName" character varying COLLATE pg_catalog."default" NOT NULL,
    "FileExtension" character varying COLLATE pg_catalog."default" NOT NULL,
    "FileMIMEType" character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "DataOutputFormat_pkey" PRIMARY KEY ("OutputFormatGuid")
);


COMMENT ON TABLE kershner."DataOutputFormat" IS 'Format is an output format and/or
transformation supported by the application.  Depending on the output format it may
generate multiple cache documents for a single CacheMetadata entry for key value
searches. ';


CREATE TABLE IF NOT EXISTS kershner."FileMetadata"
(
    "APIGUID" uuid NOT NULL,
    "FileMetaDataGUID" uuid NOT NULL DEFAULT gen_random_uuid(),
    "Name" character varying COLLATE pg_catalog."default" NOT NULL,
    "Description" character varying COLLATE pg_catalog."default" NOT NULL,
    "DateUploaded" timestamp without time zone NOT NULL,
    "DateModified" timestamp without time zone NOT NULL,
    "Revision" integer NOT NULL DEFAULT 0,
    "IsCurrent" boolean NOT NULL DEFAULT true,
    "IsDeleted" boolean NOT NULL DEFAULT false,
    "FileGuid" uuid NOT NULL,
    "InputFormatGuid" uuid NOT NULL,
    "FileHash" character varying COLLATE pg_catalog."default" NOT NULL,
    "FileCacheExpiration" integer NOT NULL DEFAULT 1200,
```

```
    CONSTRAINT "FileMetadata_pkey" PRIMARY KEY ("FileMetaDataGUID"),
    CONSTRAINT "APIGUID" FOREIGN KEY ("APIGUID")
        REFERENCES kershner."API" ("APIGUID") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
);


COMMENT ON TABLE kershner."FileMetadata" IS 'Contains revision details on files';

CREATE TABLE IF NOT EXISTS kershner."Cache"
(
    "CacheGUID" uuid NOT NULL DEFAULT gen_random_uuid(),
    "CacheKey" character varying COLLATE pg_catalog."default" NOT NULL,
    "CacheValue" character varying COLLATE pg_catalog."default" NOT NULL,
    "IsDeleted" boolean NOT NULL DEFAULT false,
    CONSTRAINT "Cache_pkey" PRIMARY KEY ("CacheGUID")
);

COMMENT ON TABLE kershner."Cache" IS 'Converted lookup files in KV, CSV, TEXT LIST
or JSON format.  ';

CREATE TABLE IF NOT EXISTS kershner."CacheMetadata"
(
    "CacheMetadataGUID" uuid NOT NULL DEFAULT gen_random_uuid(),
    "CacheGUID" uuid NOT NULL,
    "FileMetadataGUID" uuid NOT NULL,
    "OutputFormatGUID" uuid NOT NULL,
    "ExpirationTimestamp" timestamp without time zone NOT NULL,
    CONSTRAINT "CacheMetadata_pkey" PRIMARY KEY ("CacheMetadataGUID"),
    CONSTRAINT "CacheGUID" FOREIGN KEY ("CacheGUID")
        REFERENCES kershner."Cache" ("CacheGUID") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT "FileMetadataGUID" FOREIGN KEY ("FileMetadataGUID")
        REFERENCES kershner."FileMetadata" ("FileMetaDataGUID") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT "OutputFormatGUID" FOREIGN KEY ("OutputFormatGUID")
        REFERENCES kershner."DataOutputFormat" ("OutputFormatGuid") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
);

COMMENT ON TABLE kershner."CacheMetadata" IS 'Describes the Cache output object';

CREATE TABLE IF NOT EXISTS kershner."InputFile"
(
    "fileGuid" uuid NOT NULL DEFAULT gen_random_uuid(),
    "fileContents" text COLLATE pg_catalog."default",
    CONSTRAINT "InputFile_pkey" PRIMARY KEY ("fileGuid")
);

COMMENT ON TABLE kershner."InputFile" IS 'Contains the original lookup file';

CREATE TABLE IF NOT EXISTS kershner."InputFileFormat"
(
```

```
    "InputFormatGuid" uuid NOT NULL DEFAULT gen_random_uuid(),
    "Name" character varying COLLATE pg_catalog."default" NOT NULL,
    "Description" character varying COLLATE pg_catalog."default" NOT NULL,
    "FileExtension" character varying COLLATE pg_catalog."default" NOT NULL,
    "FileMIMEType" character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "InputFileFormat_pkey" PRIMARY KEY ("InputFormatGuid")
);

COMMENT ON TABLE kershner."InputFileFormat" IS 'Format is an input file format
supported by the application.';


CREATE TABLE IF NOT EXISTS kershner."InputFileFormat"
(
    "InputFormatGuid" uuid NOT NULL DEFAULT gen_random_uuid(),
    "Name" character varying COLLATE pg_catalog."default" NOT NULL,
    "Description" character varying COLLATE pg_catalog."default" NOT NULL,
    "FileExtension" character varying COLLATE pg_catalog."default" NOT NULL,
    "FileMIMEType" character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "InputFileFormat_pkey" PRIMARY KEY ("InputFormatGuid")
);

COMMENT ON TABLE kershner."InputFileFormat" IS 'Format is an input file format
supported by the application.';

CREATE TABLE IF NOT EXISTS kershner."Search"
(
    "SearchGUID" uuid NOT NULL DEFAULT gen_random_uuid(),
    "Name" character varying COLLATE pg_catalog."default" NOT NULL,
    "Description" character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "Search_pkey" PRIMARY KEY ("SearchGUID")
);

COMMENT ON TABLE kershner."Search" IS 'List of search functions and transformations
supported used for error and help prompts.';
```

DML

Five Sample SQL statements.

All the files with cache plus values

```
SELECT API."OwnerEmailAddress", API."DateExpires", API."PermissionCreate",
API."PermissionGlobalAdmin", API."IsDisabled", API."IsDeleted",

CMD.CacheMetadataGUID,CMD.ExpirationTimestamp, CAC."CacheKey", CAC."CacheValue",
CAC."IsDeleted", FMD."FileMetaDataGUID", FMD."Name", FMD."Description",
FMD."DateUploaded",

FMD."DateModified", FMD."Revision", FMD."IsCurrent", FMD."IsDeleted",
FMD."FileGuid", FMD."InputFormatGuid", FMD."FileHash", FMD."FileCacheExpiration",

DOF."Name", DOF."Description", DOF."FunctionName", DOF."FileExtension",
DOF."FileMIMEType"

      FROM

      "kershner"."API" AS API,

      "kershner"."CacheMetadata" AS CMD,

      "kershner"."Cache" AS CAC,

      "kershner"."FileMetadata" as FMD,

      "kershner"."DataOutputFormat" as DOF

where

      CMD.CacheGUID = CAC.CacheGUID AND

      CMD.FileMetadataGUID = FMD.FileMetaDataGUID AND

      CMD.OutputFormatGUID = DOF.OutputFormatGuid AND

      FMD.APIGUID = API.APIGUID;
```

Create new API for a user

```
INSERT INTO "kershner"."API"(

      "APIGUID", "API", "OwnerEmailAddress", "DateExpires", "PermissionCreate",
"PermissionGlobalAdmin", "IsDisabled", "IsDeleted")

      VALUES (gen_random_uuid(), gen_random_uuid(), 'joel.kershner@ku.edu','01-01-
3000', true, true, false, false);
```

Select all api information for a specific email address

```
SELECT "APIGUID", "API", "OwnerEmailAddress", "DateExpires", "PermissionCreate",
"PermissionGlobalAdmin", "IsDisabled", "IsDeleted"

      FROM "kershner"."API" where "OwnerEmailAddress"='joel.kershner@ku.edu';
```

Show file original information with API, status, file format info, and file metadata

```
SELECT

API."OwnerEmailAddress", API."DateExpires", API."PermissionCreate",
API."PermissionGlobalAdmin", API."IsDisabled", API."IsDeleted",

FMD."Name", FMD."Description", FMD."DateUploaded", FMD."DateModified",
FMD."Revision", FMD."IsCurrent", FMD."IsDeleted",

FMD."InputFormatGuid", FMD."FileHash", FMD."FileCacheExpiration",

IFF."Name", IFF.Description, IFF.FileExtension, IFF.FileMIMEType

      FROM

      "kershner"."API" AS API,

      "kershner"."FileMetadata" as FMD,

      "kershner"."InputFile" as FG,

      "kershner"."InputFileFormat" as IFF

where

      FMD.FileGuid=FG.fileGuid AND

      FMD.InputFormatGuid=IFF.InputFormatGuid AND

      FMD.APIGUID = API.APIGUID AND
```

```
      API.API=?;
```

Update authorization table

```
UPDATE "kershner"."IsAuthorizedTo"

      SET "APIGUID"=?, "FileMetaDataGUID"=?, "PermissionCreate"=?,
"PermissionRead"=?, "PermissionUpdate"=?, "PermissionDelete"=?

      WHERE AuthorizationGUID=?;
```