Posted in: Development (/spaces/127/sandpiper/forums/5366/development)

Enum Values in JSON vs Database

Doug Winsby (https://autocare.communifire.com/people/dougwinsby)
1/19/2020

Two columns in the `grain` table use enumerated values: `grain_type` and `encoding`. PostgreSQL has a built-in enum data type, but we've chosen not to use it because we're concerned it might not be universally supported by all drivers. Instead, we are making these integer types (actually "smallint").

For example:

```
const ( // Never change these values, only add to list
 GrainUnknown        GrainType = 0
 GrainAcesFile       GrainType = 1
 GrainAcesItem       GrainType = 2
 GrainAssetFile      GrainType = 3
 GrainPartsproFile   GrainType = 4
 GrainPartsproItem   GrainType = 5
 GrainPiesFile       GrainType = 6
 GrainPiesItem       GrainType = 7
 GrainPiesMarketcopy GrainType = 8
 GrainPiesPricesheet GrainType = 9
)
```

But the default JSON serialization routines will simply show these as numbers instead of the human-readable equivalents.

```
{
  "id": "10000000-1111-0000-0000-000000000000",
  "grain_type": 1,
  "grain_key": "disc brakes",
  "encoding": 3,
  "payload": "H4sIAAAAAAAC/ypOzEspyCxILVIoyk/OLlYEAAAA//8BAAD//451mN4QAAAA",
  "created_at": "2020-01-19T16:53:08.955876Z",
  "slice": {
    "id": "2bea8308-1840-4802-ad38-72b53e31594c",
    "slice_name": "Slice2",
    "content_hash": "2268e5deabf5e6d0740cd1a77df56f67093ec943",
    "content_count": 1,
    "content_date": "0001-01-01T00:00:00Z",
    "created_at": "2020-01-17T03:29:38.974349Z",
    "updated_at": "2020-01-17T03:29:38.974349Z"
  }
}
```


I think we should make the API friendlier than that (and not force the client to translate these values). For this reason, we will be creating custom serialization routines so the JSON will work with strings instead of integers.


```
{
  "grain_type": "aces_file",
  "encoding": "gzipb64",
}
```


The database will still keep the equivalent integer values, however.

enums (/spaces/127/sandpiper/searchresults?keyword=%22enums%22&searchtags=1)

Like

Reply (/forums/post?tid=5562&ReplyPostID=5563&SpaceID=127)

Doug Winsby (https://autocare.communifire.com/people/dougwinsby)

1/22/2020

As with many things in software development, the "best" path is not always obvious. I've changed my mind somewhat on how to store these enums in the database.

The ORM we're using does not really make it easy to store an integer and maintain it as a string equivalent. Yes, there are ways to do it, but it's much easier to use the built-in postgresql enum data type.

Normally, I would not recommend storing application data in the database schema (e.g. an enum data type), but at least in the case of `encoding`, any additions would require changes to the underlying software to support that new encoding.

The `grain_type` column seems different, though. It could be changed without Sandpiper caring. The payload is a black-box, and it doesn't matter what type of payload is being synced. For this reason, we would normally make this a lookup table with a foreign key in the grain (a "has-one" relationship).

But we don't want the extra join whenever we access a grain. So that's where we are, making both of these columns enum types (at least for the moment).

Like

Reply (/forums/post?tid=5562&ReplyPostID=5588&SpaceID=127) Answer