

Gary Griswold
513-508-6127
March 30, 2020
Revision 2

DBP v4 API Data Analysis

A Comparison of the Database and the API

This report contains three sections, which provide the same information summarized different ways. The third and most detailed section, contains API requests and responses, with the responses modified to contain the names of the database columns that was used for each data field in the response. The second section, summarizes the responses to contain only the list of data columns that each response returned without showing where in the API response they are presented. The first section, presents a union of all of that data returned by the API, so that one can see all of the data that the API requires.

Summary of all Database Responses for entire API

This section contains a union of all of the database data returned by all of the API responses, and categorizes them by how they are updated. The API responses contain some data that almost certainly is not required by the Bible.IS App. Although, some of that data might be useful to others who use the service. Looking at the data this way is useful for prioritizing parts of the load process. It would be extremely useful to know which API requests are really used by the Bible.IS App or other API users.

Static Tables

Static tables contain data that rarely changes, and would only need to be modified when an error were identified in the data. The following tables are included:

alphabets(script, name, family, type, requires_font, unicode, direction)

```
alphabet_numeral_systems(numeral_systems_id, script_id)
countries(id, iso_a3, fips, wtf, ethnologue, continent, name, introduction)
country_maps(country_id, name, thumbnail_url, map_url)
languages(id, glotto_id, iso, name, maps, area, population, country_id)
language_classifications(language_id, classification_id, order, name)
language_codes(language_id, source, code)
language_translations(language_source_id, language_translation_id, name)
numeral_systems(id, description, notes)
numeral_system_glyphs(numeral_system_id, value, glyph, numeral_written)
```

Load Tables

These tables are modified by the current load program. Unfortunately, these are not all of the Bible and fileset related tables that the API uses to create responses.

```
bibles(id, language_id, versification, numeral_system_id, date, scope, script, derived, copyright,
        reviewed, notes)
bible_books(bible_id, book_id, name, short_name, chapters)
bible_filesets(id, hash_id, asset_id, set_type_code, set_size_code)
bible_fileset_copyrights(hash_id, copyright_date, copyright, copyright_description,
        created_at, updated_at, openAccess)
bible_fileset_copyright_organizations(hash_id, organization_id, organization_role)
```

Future Load Tables

There are two tables that accomplish the same purpose: bible_organizations, and bible_fileset_copyright_organizations. Based upon an analysis of other data in those API responses, it seems certain that the later one is used by the App. Whether the former is also used, I have no way to know. It would be good to know whether both are actually used, because only the former is being updated by the current load. The current system does not have an accurate way to do this without manual correction after update. A better update could be written, and it could update both tables.

```
bible_organizations(bible_id, organization_id, relationship_type)
```

Other Bible Related Tables

The bibles_defaults table might have been recently manually created for the Bible.IS App, and would sometimes need to be manually updated when new Bibles are loaded. The other

three tables contain older records that was put there by DBS, but the filesets and LPTS do not contain data needed to populate these tables. The table bible_translations is being updated by the load for English, but the intent of the table is to contain the names of Bibles in their local language.

```
bibles_defaults(language_code, bible_id, type)
bible_equivalents(bible_id, equivalent_id, organization_id, type, site, suffix)
bible_links(bible_id, type, url, title, provider, organization_id)
bible_translations(language_id, bible_id, vernacular, vernacular_trade, name, description, background)
```

Organization Tables

The organization related tables require infrequent updating, which is good, because the automated systems available do not provide a data source for these tables. Probably, they need to be maintained manually as new organizations are added.

```
organizations(id, slug, abbreviation, primaryColor, secondaryColor, inactive,
              url_facebook, url_website, url_donate, url_twitter, address, address2,
              city, state, country, zip, phone, email, email_director, latitude, longitude)
organization_logos(organization_id, language_id, language_iso, url, icon)
organization_translations(language_id, organization_id, vernacular, alt, name, description_short)
```

Resource Tables

It is not clear to me how these tables are used, why they are needed, or what might be a data source, or whether they can be considered static tables.

```
resources(id, language_id, slug, iso, organization_id, source_id, cover, cover_thumbnail, date, type)
resource_links(resource_id, title, size, type, url)
resource_translations(language_id, resource_id, title, description)
```

Unused Database Columns and Tables

Comparing all of the above data, which are displayed in the API, with the tables in the DBP database, the following is a list of data tables that contain useful data, but I have not found these in any API response.

```
bible_file_titles - contains local language chapter titles, possibly extracted from USX files.
book_translations - appears to be an early version of bible_books
commentaries - 1 commentary by John Wesley
commentary_sections - content of 1 commentary
```

commentary_translations - translation of content to 1 language, 6414
country_communications - economic stats on country communications
country_economy - economic stats on countries
country_energy - economic energy stats on countries
country_geography - geography data on countries
country_government - government data on countries
country_issues - description of key problems in countries
country_joshua_project - religion stats on countries
country_people - demographic stats on countries
country_people_ethnicities - stats on ethnic groups in some countries
country_regions - continent and primary language of countries
country_religions - stats on religion by country
country_translations - country names in many languages
country_transportation - transportations stats by country
language_bibleInfo - stats about Bible versions available in language
language_status - 13 rows of data, not linked to language table
organization_relationships - parent/child relationships 172 orgs

The following tables are used by the API, but the API does not display all of their columns. The following is a list of the tables and the columns that are not displayed.

alphabets(unicode_pdf, white_space, open_type_tag, complex_positioning, diacritics, context_forms, reordering, case, split_graphs, status, baseline, ligatures, direction_notes, sample, sample_img, description)
countries(overview)
languages(iso2B, iso2T, iso1, development, use, location, population_notes, typology, writing, latitude, longitude, status_id)
language_translations(priority)
bible_translations(notes)
organizations(notes)
resource_translations(vernacular, tag)

The following tables are not used by the API, and are also empty in the DBP database:

alphabet_fonts
bible_concordance
bible_fileset_fonts
bible_fileset_relations
bible_size_translations

bible_translator
bible_verse_concordance
connection_translations
connections
fonts
resource_connections
translator_relations
translators
video_sources
video_tags
video_translations
videos

API Responses Database Summary

This section contain a list of tables and the columns returned by each API transactions.
Some transactions that returned a subset of what other transactions returned, are not included.

/alphabets

alphabets(script, name, family, type, direction)

/bibles

bibles(id, language_id, date)
bible_filesets(id, hash_id, asset_id, set_type_code, set_size_code)
bible_translations(languages_id, bible_id, name)
languages(id, iso)
language_translations(language_id, bible_id, name)

/bibles/{bible_id}

alphabet(script, name, direction, unicode, requires_font)
bibles(id, date, copyright)
bible_books(bible_id, book_id, name, short_name, chapters)

bible_equivalents(bible_id, equivalent_id, organization_id, type, site, suffix)
bible_filesets(id, hash_id, asset_id, set_type_code, set_size_code)
bible_links(bible_id, type, url, title, provider, organization_id)
bible_organizations(bible_id, organization_id, relationship_type)
bible_translations(language_id, bible_id, name, description)
countries(id, name)
languages(id, iso, name)
organizations(id, abbreviation, primaryColor, secondaryColor, inactive,
url_facebook, url_website, url_donate, url_twitter, address, address2, city, state, country, zip, phone,
email, email_director, latitude, longitude)
organization_logos(organization_id, language_id, url, language_iso, icon)
organization_translations(language_id, organization_id, vernacular, alt, name, description_short)

/bibles/defaults/types

bibles_defaults(language_code, type, bible_id)

/bibles/filesets/{fileset_id}/copyright

bible_filesets(id, hash_id, asset_id, set_type_code, set_size_code)
bible_fileset_copyrights(hash_id, copyright_date, copyright, copyright_description, created_at, updated_at,
openAccess)
bible_fileset_copyright_organizations(hash_id, organization_id, organization_role)
organizations(id, slug, abbreviation, primaryColor, secondaryColor, inactive,
url_facebook, url_website, url_donate, url_twitter, address, address2,
city, state, country, zip, phone, email, email_director, latitude, longitude)
organization_logos(organization_id, language_id, language_iso, url, icon)
organization_translations(language_id, organization_id, vernacular, alt, name, description_short)

/bibles/filesets/{fileset_id}/podcast

Unresolved, mostly non-DBP source

/countries

countries(id, iso_a3, fips, continent, name)

/countries/{country_id}

bibles(id, language_id)
bible_translations(language_id, bible_id, name)
countries(id, iso_a3, fips, wft, ethnologue, continent, name, introduction)
country_maps(country_id, name, thumbnail_url, map_url)
languages(id, iso, name)

/languages

languages(id, glotto_id, iso, name)
language_translations(language_source_id, language_translation_id, name)

/languages/{language_id}

bibles(id, language_id, versification, numeral_system_id, date, scope, script, derived, copyright, reviewed, notes)
bible_filesets(id, hash_id, asset_id, set_type_code, set_size_code)
bible_translations(language_id, bible_id, vernacular, vernacular_trade, name, description, background)
countries(id, name)
languages(id, glotto_id, iso, name, maps, area, population, country_id)
language_classifications(language_id, classification_id, order, name)
language_codes(language_id, source, code)
language_translations(language_source_id, language_translation_id, name)
resources(id, language_id, slug, iso, organization_id, source_id, cover, cover_thumbnail, date, type)
resource_links(resource_id, title, type, size, url)
resource_translations(language_id, resource_id, title, description)

/numbers

alphabets(script, name)
alphabet_numeral_systems(numeral_system_id, script_id)
numeral_systems(id, description, notes)

/numbers/{numeral_id}

alphabets(script, name)

alphabet_numeral_systems(numeral_system_id, script_id)
numeral_systems(id, description, notes)
numeral_system_glyphs(numeral_system_id, value, glyph, numeral_written)

API Responses & DBP Database Mapping

This section contains DBP v4 API requests, and the API responses. The data content returned has been replaced by the table.column names that was used as the DBP source for the data. Many of the responses contain lists of result records. These are indicated by the presence of a **[** character before that part of the response. A few API requests that returned a subset of the contents of other requests have been left out of this analysis.

/alphabets

GET <https://4.dbt.io/api/alphabets?v=4>

name: alphabets.name
script: alphabets.script
family: alphabets.family
type: alphabets.type
direction: alphabets.direction

/bibles

GET <https://4.dbt.io/api/bibles?v=4>

abbr: bibles.id
name: bible_translations.name for language_id 6414
vname: bible_translations.name for language_id this record
language: language_translations.name for language_id 6414
autonym: language_translations.name for language_id this record
language_id: bibles.language_id
iso: languages.iso for language_id this record

date: bibles.date
filesets: {
 bible_filesets.asset_id: [{
 id: bible_filesets.id
 type: bible_filesets.set_type_code
 size: bible_filesets.set_size_code
 bitrate: I have only seen null?

/bibles/{bible_id}

GET https://4.dbt.io/api/bibles/{bible_id}?v=4

abbr: bibles.id
alphabet: {
 script: bibles.script
 name: alphabets.name for script
 direction: alphabets.direction for script
 unicode: alphabets.unicode for script
 requires_font: alphabets.requires_font for script
 primary_font: alphabet_fonts for script (but table is empty)
mark: bibles.copyright
name: bible_translations.name for language_id 6414
description: bible_translations.description for language_id 6414
vname: bible_translations.name for language_id this bible language_id
vdescription: bible_translations.description for this bible language_id
publishers: [{
 id: bible_organizations.organization_id for bibles.id
 slug: organizations.slug for id
 abbreviation: organizations.abbreviation for id
 primaryColor: organizations.primary_color for id
 secondaryColor: organizations.secondary_color for id
 inactive: organizations.inactive for id
 url_facebook: organizations.url_facebook for id
 url_website: organizations.url_website for id
 url_donate: organizations.url_donate for id
 url_twitter: organizations.url_twitter for id
 address: organizations.address for id
 address2: organizations.address2 for id

city: organizations.city for id
 state: organizations.state for id
 country: organizations.country for id
 zip: organizations.zip for id
 phone: organizations.phone for id
 email: organizations.email for id
 email_director: organizations.email_director for id
 latitude: organizations.latitude for id
 longitude: organizations.longitude for id
 pivot: {
 bible_id: bibles.id
 organization_id: bible_organizations.organization_id for bibles.id
 relationship_type: bible_organizations.relationship_type for bibles.id
 }
 logo_icon: {
 language_id: organization_logos.language_id for id
 language_iso: organization_logos.language_iso for id
 url: organization_logos.url for id
 icon: organization_logos.icon for id
 }
 translations: [{
 language_id: organization_translations.language_id for id
 vernacular: organization_translations.vernacular for id
 alt: organization_translations.alt for id
 name: organization_translations.name for id
 description_short: organization_translations.description_short
 }]
 providers: []
 equivalents: [{
 bible_id: bibles.id
 equivalent_id: bible_equivalents.equivalent_id for bibles.id
 organization_id: bible_equivalents.organization_id for bibles.id
 type: bible_equivalents.type for bibles.id
 site: bible_equivalents.site for bibles.id
 suffix: bible_equivalents.suffix for bibles.id
 }]
 language: languages.name for bibles.language_id
 language_id: bibles.language_id
 iso: languages.iso for bibles.language_id
 date: bibles.date
 country: countries.name for languages.id for bibles.language_id
 books: [{

```

    book_id: bible_books.book_id
    name: bible_books.name
    name_short: bible_books.short_name
    chapters: bible_books.chapters}
links: [{
    bible_id: bibles.id
    type: bible_links.type for bibles.id
    url: bible_links.url for bibles.id
    title: bible_links.title for bibles.id
    provider: bible_links.provider for bibles.id
    organization_id: bible_links.organization_id for bibles.id
filesets: [{
    dbp-prod: [{ bible_filesets.asset_id
                id: bible_filesets.id
                type: bible_filesets.set_type_code
                size: bible_filesets.set_size_code
fonts: null

```

/bibles/defaults/types

GET <https://4.dbt.io/api/bibles/defaults/types?v=4>

```

bibles_defaults.language_code: {
    bible_defaults.type
    bible_defaults.bible_id

```

/bibles/filesets/{fileset_id}/copyright

GET https://4.dbt.io/api/bibles/filesets/{fileset_id}/copyright?v=4

```

id: bible_filesets.id
asset_id: bible_filesets.asset_id
type: bible_filesets.set_type_code
size: bible_filesets.set_size_code
copyright: {
    hash_id: bible_filesets.hash_id

```

copyright_date: bible_fileset_copyrights.copyright_date for hash_id
 copyright: bible_fileset_copyrights.copyright for hash_id
 copyright_description: bible_fileset_copyrights.copyright_description for hash_id
 created_at: bible_fileset_copyrights.created_at for hash_id
 updated_at: bible_fileset_copyrights.updated_at for hash_id
 open_access": bible_fileset_copyrights.open_access for hash_id
 organizations: [{
 id: bible_fileset_copyright_organizations for hash_id and organization_role 1 or 2?
 slug: organizations.slug for id
 abbreviation: organizations.abbreviation for id
 primaryColor: organizations.primary_color for id
 secondaryColor: organizations.secondary_color for id
 inactive: organizations.inactive for id
 url_facebook: organizations.url_facebook for id
 url_website: organizations.url_website for id
 url_donate: organizations.url_donate for id
 url_twitter: organizations.url_twitter for id
 address: organizations.address for id
 address2: organizations.address2 for id
 city: organizations.city for id
 state: organizations.state for id
 country: organizations.country for id
 zip: organizations.zip for id
 phone: organizations.phone for id
 email: organizations.email for id
 email_director: organizations.email_director for id
 latitude: organizations.latitude for id
 longitude: organizations.longitude for id
 laravel_through_key: bible_filesets.hash_id
 logos: [{
 language_id: organization_logos.language_id for id
 language_iso: organization_logos.language_iso for id
 url: organization_logos.url for id
 icon: organization_logos.icon for id
 }]
 translations: [{
 language_id: organization_translations.language_id for id
 vernacular: organization_translations.vernacular for id
 alt: organization_translations.alt for id
 }]
 }]

name: organization_translations.name for id
description_short: organization_translations.description_short

/bibles/filesets/{fileset_id}/podcast

GET <https://4.dbt.io/api/bibles/filesets/CHNUN1N2DA/podcast?v=4>

A very large part of this response does not appear to come from DBP.

And so, an undocumented response follows.

channel: {

title: "和合本 - Mandarin Chinese",

link: "https://4.dbt.io",

atom:link: {

href: "http://bible.is/feed.xml",

rel: "self",

type: "application/rss+xml"

description: "The Chinese Union Version (CUV) is well known Chinese language Bible, translated by a panel representing many Protestant denominations. The panel used the English Revised Version validating against original manuscripts. Work on the CUV began in 1890 and was completed in 1919. This edition uses simplified Chinese characters which are used mainly in the People's Republic of China (Mainland China) and Singapore.",

language: "cmn",

managingEditor: "alan@fcbhmail.net",

webMaster: "alan@fcbhmail.net",

copyright: "The Worldwide Bible Society, Lockman Foundation",

lastBuildDate: "",

docs: "http://blogs.law.harvard.edu/tech/rss",

itunes:keywords: "Bible, Testament, Jesus, Scripture, Holy, God, Heaven, Hell, Gospel, Christian, Bible.is, Church",

itunes:author: "Faith Comes By Hearing",

itunes:subtitle: "Online Audio Bible Recorded by Faith Comes By Hearing",

itunes:explicit: "no",

itunes:owner: {

itunes:name: "Faith Comes By Hearing",

itunes:email: "alan@fcbhmail.net"

itunes:image: {

href: "https://dbp-prod.s3.us-west-2.amazonaws.com/audio/CMNUN1/CHNUN1N2DA/Art/300x300/CHNUN1N2DA.jpg?X-Amz-Content-Sha256="

```

itunes:category: {
  _attributes: {
    text: "Religion & Spirituality"
  }
}

image: {
  _attributes: {
    url: "http://bible.is/CHNUN1N2DA.jpg",
    title: "Faith Comes by Hearing",
    link: "http://bible.is"
  }
}

item: [{
  title: "哥林多前书 1",
  link: "http://podcastdownload.faithcomesbyhearing.com/mp3.php/CHNUN1N2DA/Bo7___01_1CorinthiansCHNUN1N2DA.mp3",
  guid: "http://podcastdownload.faithcomesbyhearing.com/mp3.php/CHNUN1N2DA/Bo7___01_1CorinthiansCHNUN1N2DA.mp3",
  enclosure: {
    name: null,
    _attributes: {
      url: "http://podcastdownload.faithcomesbyhearing.com/mp3.php/B11___01_Philippians_CHNUN1N2DA/Bo7___01_1CorinthiansCHNUN1N2DA.mp3.mp3",
      length: 311,
      type: "audio/mpeg"
    }
  },
  pubDate: "Tue, 13 Feb 2018 10:27:04 +0000",
  itunes:author: "Faith Comes By Hearing",
  itunes:explicit: "no",
  itunes:subtitle: "",
  itunes:summary: "",
  itunes:duration: 311,
  itunes:keywords: "Bible, Testament, Jesus, Scripture, Holy, God, Heaven, Hell, Gospel, Christian, Bible.is, Church"
}]

```

/countries

GET https://4.dbt.io/api/countries?v=4

```

name: countries.name
continent_code: countries.continent
codes: {
  fips: countries.fips
}

```

iso_a3: countries.iso_a3

iso_a2: countries.id

/countries/{country_id}

GET <https://4.dbt.io/api/countries/CN?v=4>

name: countries.name

introduction: countries.introduction

continent_code: countries.continent

maps: {

country_maps.name for country_id {

thumbnail_url: country_maps.thumbnail_url for country_id

map_url: country_maps.map_url

wfb: countries.wfb

ethnologue: countries.ethnologue

languages: [{

name: languages.name for country_language.language_id for country_id

iso: languages.iso for country_language.language_id for country_id

bibles: [{

bibles.id: bible_translations.name for bibles.language_id and bible_translations.language_id = 6414

codes: {

fips: countries.fips

iso_a3: countries.iso_a3

iso_a2: countries.id

/languages

GET <https://4.dbt.io/api/languages?v=4>

id: languages.id

glotto_id: languages.glotto_id

iso: languages.iso

name: languages.name

autonym: language_translations.name for language_translation_id is languages.id

bibles: count(bibles) for languages.id

filesets: count(bible_filesets) for languages.id

/languages/{language_id}

GET https://4.dbt.io/api/languages/{language_id}?v=4

```
id: languages.id,
name: languages.name
description: unknown source
autonym: language_translations.name for languages.id
glotto_id: languages.glotto_id
iso: languages.iso
maps: languages.maps
area: languages.area
population: languages.population
country_id: languages.country_id
country_name: countries.name for languages.country_id
codes: {
  language_codes.source : language_codes.code for languages.id
alternativeNames: {
  o to n: languages.name for language_translations.language_source_id is languages.id
  # The above seems unusable since the language_id that the name is translated into
  # is not part of the response.
dialects: language_dialects (I was unable to make a request that returned dialects)
classifications: {
  language_classifications.classification_id : language_classifications.name
bibles: [{
  id: bibles.id for bibles.language_id and languages.id
  language_id: bibles.language_id
  versification: bibles.versification
  numeral_system_id: bibles.numeral_system_id
  date: bibles.date
  scope: bibles.scope
  script: bibles.script
  derived: bibles.derived
  copyright: bibles.copyright
  reviewed: bibles.reviewed
  notes: bibles.notes
translations: [{
```



```

    language_id: bible_translations.languages.id
    bible_id: bible_translations.bible_id
    vernacular: bible_translations.vernacular
    vernacular_trade: bible_translations.vernacular_trade
    name: bible_translations.name
    description: bible_translations.description
    background: bible_translations.background
    filesets: [{
        id: bible_filesets.id for bible_fileset_connections.hash_id and bibles.id
        asset_id: bible_filesets.asset_id
        set_type_code: bible_filesets.set_type_code
        set_size_code: bible_filesets.set_size)code
        laravel_through_key: bibles.id
    resources: [{
        id: resources.id
        language_id: resources.language_id
        slug: resources.slug
        iso: resources.iso
        organization_id: resources.organization_id
        source_id: resources.source_id
        cover: resources.cover
        cover_thumbnail: resources.cover_thumbnail
        date: resources.date
        type: resources.type
        translations: [{
            language_id: resource_translations.language_id
            title: resource_translations.title
            description: resource_translations.description
        links: [{
            title: resource_links.title
            size: resource_links.size
            type: resource_links.type
            url: resource_links.url

```

/numbers

GET <https://4.dbt.io/api/numbers?v=4>

id: numeral_systems.id
description: numeral_systems.description
notes: numeral_systems.notes
alphabets: {
 numeral_system_alphabets.script_id : alphabets.name for alphabets.script_id

/numbers/{numeral_id}

GET https://4.dbt.io/api/numbers/{numeral_id}?v=4

id: numeral_systems.id
description: numeral_systems.description
notes: numeral_system.notes
alphabets: {
 numeral_system_alphabets.script_id : alphabets.name for alphabets.script_id
numerals: [{
 value: numeral_system_glyphs.value
 glyph: numeral_system_glyphs.glyph
 numeral_written: numeral_system_glyphs.numeral_written