

بسم الله الرحمن الرحيم

راهنمای Developer پایگاه داده

## Contents

۱ - مقدمه	۲
۱-۲ - why PostgreSQL ؟	۲
۱-۳ - why peewee ؟	۳
۲ - جداول	۴
۲-۱ - توضیح جداول	۴

## ۱ - مقدمه

در این سند نوع پایگاه داده، جداول، روابط و orm استفاده شده در SystemAPI بررسی خواهد شد. در SystemApi از پایگاه داده PostgreSQL و از peewee به عنوان orm استفاده شده است که در ادامه مزایای آن بررسی خواهد شد.

## ۱-۲ - why PostgreSQL ؟

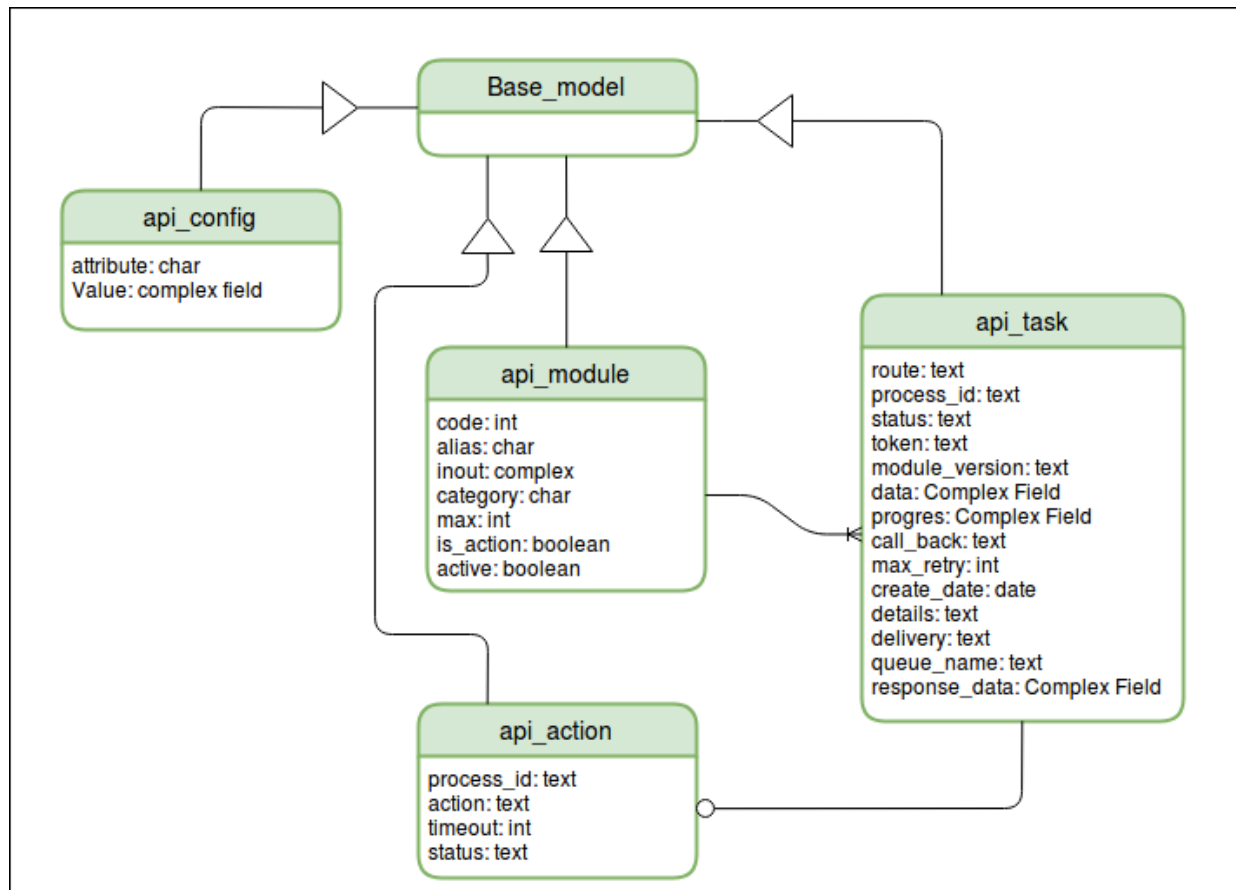
Postgresql یکی از پایگاه داده های قدرتمند می باشد که به صورت open source عرضه شده است. این پایگاه داده توسط زبان های مختلف پشتیبانی می شود و ویژگی های زیادی در زمینه های data integrity، security و ... دارد که در ادامه به آن ها اشاره می شود.

- Data Types
  - Primitives: Integer, Numeric, String, Boolean
  - Structured: Date/Time, Array, Range, UUID
  - Document: JSON/JSONB, XML, Key-value (Hstore)
  - Geometry: Point, Line, Circle, Polygon
  - Customizations: Composite, Custom Types
- Data Integrity
  - UNIQUE, NOT NULL
  - Primary Keys
  - Foreign Keys
  - Exclusion Constraints
  - Explicit Locks, Advisory Locks
- Concurrency, Performance
  - Indexing: B-tree, Multicolumn, Expressions, Partial
  - Advanced Indexing: GiST, SP-Gist, KNN Gist, GIN, BRIN, Bloom filters
  - Sophisticated query planner / optimizer, index-only scans, multicolumn statistics
  - Transactions, Nested Transactions (via savepoints)
  - Multi-Version concurrency Control (MVCC)
  - Parallelization of read queries
  - Table partitioning

- All transaction isolation levels defined in the SQL standard, including Serializable
- Reliability, Disaster Recovery
  - Write-ahead Logging (WAL)
  - Replication: Asynchronous, Synchronous, Logical
  - Point-in-time-recovery (PITR), active standbys
  - Tablespaces
- Security
  - Authentication: GSSAPI, SSPI, LDAP, SCRAM-SHA-256, Certificate, and more
  - Robust access-control system
  - Column and row-level security
- Extensibility
  - Stored procedures
  - Procedural Languages: PL/PGSQL, Perl, Python (and many more)
  - Foreign data wrappers: connect to other databases or streams with a standard SQL interface
  - Many extensions that provide additional functionality, including PostGIS
- Internationalisation, Text Search
  - Support for international character sets, e.g. through ICU collations
  - Full-text search

### ۱-۳- why peewee ؟

همانطور که در بخش قبل گفته شد از peewee به عنوان ORM استفاده شده است. Peewee یک ORM سبک و در عین حال پر قدرت است به طوری که از عملیات های اصلی sql و همچنین پایگاه داده postgresql و sqlite پشتیبانی می کند. از طرف دیگر migration manager های متعددی سازگار با این orm طراحی شده اند که به صورت خودکار از روی مدل ها جداول را ایجاد می کنند. Peewee\_move یک migrator سازگار با peewee است که در این پروژه از آن استفاده شده است.



شکل ۱- نمودار ER طراحی پایگاه داده SystemApi

همانطور که از شکل مشخص است، پایگاه داده SystemApi از ۴ جدول طراحی شده است که از Base\_model که در سطح abstraction قرار دارد extend شده اند.

## ۲-۱- توضیح جداول

### ۲-۱-۱- جدول api\_config

در این جدول تنظیمات مربوط به SystemApi به صورت key-value ذخیره می شود. برای مثال ورژن پایگاه داده در این جدول ذخیره می گردد. Attribute مربوط به تنظیمات و ستون value مربوط به مقدار تنظیمات مورد نظر است. نوع داده Complex\_field توسط طراحان SystemApi پیاده سازی شده است که در پایگاه داده به صورت blob ذخیره می شود و در هنگام خواندن به یک نوع داده python تبدیل می شود.

### ۲-۱-۲- جدول api\_module

در این جدول اطلاعات مذبوط به هر ماژول ذخیره می شود. هر ماژول جدیدی در زمان install روی سرویس دهنده SystemApi باید در این جدول یک رکورد اختصاصی برای خودش ثبت کند. ستون های این جدول عبارتند از:

Code: شناسه ماژول، alias: اسم ماژول، inout: ورودی و خروجی ماژول، category: دسته مصرفی ماژول، max: تعداد درخواست هایی که به طور هم زمان این ماژول می تواند پاسخ دهد، is\_action: مشخص می کند ماژول action است یا خیر. active: فال یا غیر فعال ماژول.

#### ۳-۱-۲- جدول api\_task

در این جدول کلیه اطلاعات مربوط به یک درخواست ثبت می شود و به مرور مقادیر آن به روز رسانی می شود. ستون های آن شامل :

route: مسیر درخواست، token: توکن برقرار ارتباط بین کاربر و سیستم، max\_retry: تعداد تلاش ها برای تحویل نتیجه، queue\_name: اسم صفی که درخواست در آن قرار می گیرد، detail: جزئیات، create\_date: تاریخ ثبت درخواست. Response\_data: جواب درخواست. delivery: وضعیت رسیدن پیام، status: وضعیت درخواست، module\_version: ورژن ماژولی که بخ آن درخواست شده، data: داده ورودی، process\_id: شناسه درخواست، call\_back: مسیری که پاسخ به آن ارسال می شود.

#### ۴-۱-۲- جدول api\_action

دستوراتی مانند pause, cancel در این جدول ذخیره می شوند. این جدول یک کلید خارجی به مرجع جدول api\_task دارد.