

MODEL

DATABASE SCHEMA

The database schema is a description of the relational model to do the ORM. You define the tables, their relations, and the characteristics of their columns. Symfony's syntax for schemas uses the YAML format.

Symfony also understands the Propel native XML schema format

schema.yml

located in `myproject/config/`

the first key represents a connection name

propel:

_attributes: {noXsd: false, defaultIdMethod: none, package: lib.model}

blog_article:

_attributes: { phpName: Article }

id: {required: true, primaryKey: true, autoIncrement: true}

title: varchar(255)

content: longvarchar

created_at:

name: {type: varchar(50), default: foobar, index: true}

group_id: {type: integer, foreignTable: db_group, foreignReference: id}

blog_comment:

_attributes: { phpName: Comment }

id:

article_id:

author: varchar(255)

content: longvarchar

created_at:

tables

connection attributes

noXsd - Set it to false if you want your schema to be validated before code generation takes place.

defaultIdMethod - If none is provided, then the database's native method of generating IDs will be used - for example, **autoincrement** for MySQL, or **sequences** for PostgreSQL. The other possible value is **none**.

package - is like a namespace; it determines the path where the generated classes are stored. It defaults to `lib/model/`, but you can change it to organize your model in subpackages.

column attributes

type - Column type. The choices are boolean, tinyint, smallint, integer, bigint, double, float, real, decimal, char, varchar(size), longvarchar, date, time, timestamp, bu_date, bu_timestamp, blob, and clob.

required - Boolean. Set it to true if you want the column to be required.

default - Default value.

primaryKey - Boolean. Set it to true for primary keys.

autoIncrement - Boolean. Set it to true for columns of type integer that need to take an auto-incremented value.

sequence - Sequence name for databases using sequences for autoIncrement columns (for example, PostgreSQL and Oracle).

index - Boolean. Set it to true if you want a simple index or to unique if you want a unique index to be created on the column.

foreignTable - A table name, used to create a foreign key to another table.

foreignReference - The name of the related column if a foreign key is defined via foreignTable.

onDelete - Determines the action to trigger when a record in a related table is deleted. When set to set null, the foreign key column is set to null. When set to cascade, the record is deleted. If the database engine doesn't support the set behavior, the ORM emulates it. This is relevant only for columns bearing a foreignTable and a foreignReference.

isCulture - Boolean. Set it to true for culture columns in localized content tables.

empty column type

Empty columns named id are considered PKs
`id: { type: integer, required: true, primaryKey: true, autoIncrement: true }`

Empty columns named XXX_id are considered foreign keys
`foobar_id: { type: integer, foreignTable: db_foobar, foreignReference: id }`

Empty columns named created_at, updated_at, created_on and updated_on are considered dates and automatically take the timestamp type
`created_at: { type: timestamp }`
`updated_at: { type: timestamp }`

table attributes

phpName - the name of the class that will be generated. If you don't mention a phpName for a table, the name will be the camelCase version of the table name.

isI18N - Boolean. Set it to true for i18n

i18nTable - name of the i18n table.

Indexes and Unique Indexes Alternative Syntax

```
propel:
  blog_article:
    id:
    title:          varchar(50)
    created_at:
    _indexes:
      my_index:    [title, user_id]
    _uniques:
      my_other_index: [created_at]
```

i18n

Tables that contain localized content

Implied i18n Mechanism

```
propel:
  db_group:
    id:
    created_at:

  db_group_i18n:
    name:  varchar(50)
```

Explicit i18n Mechanism

```
propel:
  db_group:
    _attributes: { isI18N: true, i18nTable: db_group_i18n }
    id:
    created_at:

  db_group_i18n:
    id: { type: integer, required: true, primaryKey: true,
          foreignTable: db_group, foreignReference: id, onDelete: cascade }
    culture: { isCulture: true, type: varchar(7), required: true, primaryKey: true }
    name:  varchar(50)
```

Foreign Key Alternative Syntax Applied to Multiple Reference Foreign Key

```
_foreignKeys:
  my_foreign_key:
    foreignTable: db_user
    onDelete:    cascade
  references:
    - { local: user_id, foreign: id }
    - { local: post_id, foreign: id }
```