Library Management System with Pyro

Andrei Petru Stefanie
Group 244

Requirements

Build a library management system enabling its users to perform the following:
- Read, create, update, delete authors
- Search authors by substrings from their name
- Read, create, update, delete books
- Search books by substrings from their title

The purpose of the project is to exemplify the capabilities of the Pyro library for remote objects.

Implementation

The system consists of a server and several clients interacting through the Pyro protocol over the network, using the Serpent serializer.

The clients are based on the command-line interface (CLI) and enable the users to perform the described actions.

Server

The server is implemented in Python (3.9) and exposes its functionality using the Pyro library.

The system has two entities: **Authors** and **Books**.

An author has the following properties:

- Id (int)
- Name (string)
- Birthday (date)

A book has the following properties:

- Id (int)
- Title (string)
- Publishing date (date)
- Author

The server exposes the following methods:

- `get_authors(q)`
  - Retrieve the list of authors, optionally filtered by an arbitrary substring from their name

- `add_author(name, birthday)`
  - Create an author by providing the name and birthday
- `delete_author(id)`
  - Delete the author identified by the *id*
- `update_author(id, name)`
  - Update the name of the author identified by the *id*
- `get_books(q)`
  - Retrieve the list of books and their authors, optionally filtered by an arbitrary substring from their title
- `add_book(title, author_id, published_at)`
  - Create a book by providing the title, author id, and publishing date
- `delete_book(id)`
  - Delete the book identified by the *id*
- `update_book(id, title)`
  - Update the title of the book identified by the *id*

Database

The database is MySQL (developed using version 8.0.29) and the server interacts with it through the [peewee](#) ORM.

To run the server you have to set the `DATABASE` environment variable that instructs the server on how to connect to the database.
E.g. `export DATABASE="mysql://root:pass@localhost:3306/library"`.
If you don't provide this, the server will use an SQLite database with a file named `default.db`.

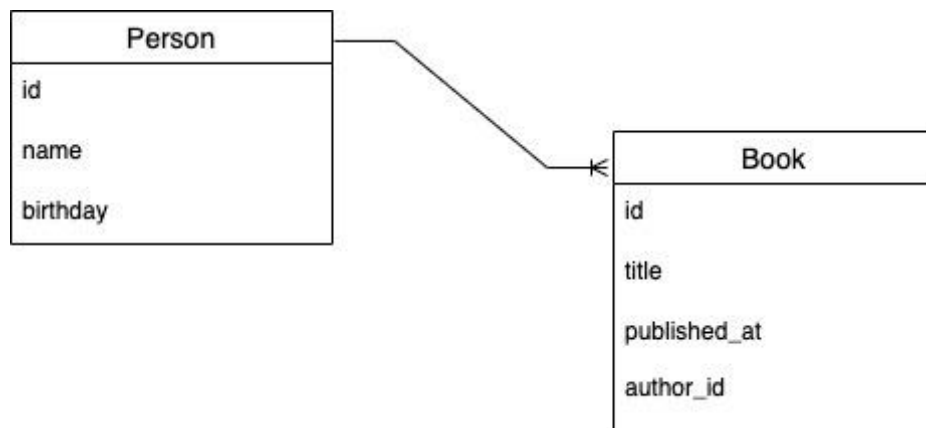The ORM automatically creates the database tables when it starts. These are illustrated in diagram 1.

Diagram 1. Database Model

Serialization

The server uses the [Serpent](#) library to serialize the objects before sending them over the network. In this case, the author/person and book models are converted to dictionaries - see [register_class_to_dict](#) from the Pyro documentation.

Clients

There are 3 available CLI clients. The positional arguments and options are the same for all of them:
- List authors: `authors get [-q 'shake']`
- Add author: `authors add --name 'William Shakespeare` --birthday '1564-12-31`
- Delete author: `authors delete --id 1`
- Update author: `authors update --id 1 --name 'New Name`
- List books: `books get [-q 'ham']`
- Add book: `books add --title 'Hamlet` --date '1599-12-31` --author-id 1`
- Delete book: `books delete --id 1`
- Update book: `books update --id 1 --name 'New Title`

Python

The Python client is based on the Pyro library and uses the [argsparse](#) standard library to create the CLI.

You can install its dependencies (Pyro4) with `pip install -r requirements.txt`.

You can run it with `python3 pyro4.py books get`

C#

The C# client is built with .NET Core 6 and uses the [Pyrolite](#) library (4.31) for interacting with the server and the [CommandLine](#) library for the CLI.

You can run it with `dotnet run books get`

Java

The Java client is built with Java 18 and Gradle. It uses the [Pyrolite](#) library (4.30) for interacting with the server and the [picocli](#) library for the CLI.

You can run it with `./gradlew run --args="books get"`

Libraries

- Pyro4 version 4.82
- Pyrolite NuGet package version 4.31
- Pyrolite Maven package version 4.30

- Serpent version 1.40

References

- https://pyro4.readthedocs.io/en/stable/index.html
- https://github.com/irmen/Pyrolite/tree/pyro4-legacy
- https://github.com/irmen/Serpent
- https://docs.python.org/3/library/argparse.html
- https://github.com/commandlineparser/commandline
- https://picocli.info/