



Symfony

Please create a bookshelf application in Symfony (based on symfony/skeleton:"6.1.*") composed of two separate projects. We will call them "FRONT-END" and "BACK-END".

FRONT-END will be responsible of handling POST and GET client requests and BACK-END will be responsible of storing and providing data from/to FRONT-END.

FRONT-END application:

This application will act as a proxy to BACK-END featuring simple authentication, DTO, AMPQ

Endpoints:

POST /books

For this endpoint accept JSON with the following fields: title, author, pages, releaseDate.

Validate fields using DTO (use argument value resolver in controller) as following

title and author - max 30 chars (azAz0-9.)

pages - 0-1000

releaseDate - dd-mm-yyyy format +/- 100 years

Protect this endpoint by custom authenticator extending AbstractAuthenticator based on 'X-API-KEY' header with simple hard coded key value and implement firewall to forbid access if wrong/missing key is provided.

Once the request is validated send RabbitMQ message with the request data.

GET /books

Forward this request to BACK-END and return unmodified response.

BACK-END application:

This application will be responsible of storing AMPQ messages and providing a list of stored books.

Create two db tables: "books" and "authors" and create one-to-one relation between book and author. Use UUID as primary key type.

Create RabbitMQ consumer to handle (store) messages sent by FRONT-END and wrap operation within Doctrine transaction with rollback on error.

Endpoints:

GET /books

Return JSON list of stored books, ordered by dateCreated DESC. Use JMS serializer to create the following output structure:

```
{
  "data": [
    {
      "id": "80eb4bf8-f745-4543-b1e1-5ecdbbb6e9a5",
      "title": "The Parent Agency",
      "author": "David Baddiel",
      "pages": "50",
      "releaseDate": "2004-09-03"
    },
    {
      "id": "ec1131de-9982-4b5c-b0eb-46d17022ee2e",
      "title": "Lionboy",
      "author": "Zizou Corder",
      "pages": "123",
      "releaseDate": "1964-07-25"
    }
  ]
}
```