Resty-Chess API Exercise

Design and implement a RESTful API in Python to manage a chess board.

This exercise is purposefully vague to explore your ability to design an application. There is no perfect solution. Be prepared to discuss design decisions.

Stack Selection

- Python (any non-eol version)
- Python web framework of your choice
- Any other dependencies you like
- Docker and build tools of your choice

Requirements

- Create an API that allows users to manipulate chess pieces on a virtual board.
- Initialize a standard chess board setup with all pieces at the start of the application.
- Store the board state in memory (no database required).
- No need for chess movement rules validation beyond ensuring pieces stay within the board boundaries.
- Design the representation of chess pieces. The users of this API will be familiar with Algebraic Chess Notation
- All responses should be JSON in a format of your design.
- Provide a dockerfile that builds your application and runs it by default

Operations

Implement an API to handle the following types of requests. Considerations like “what does restful mean anyway” and “how might we want to iterate on this application in the future” will be discussed.

1. Get Board State
2. Move Piece
3. Remove Piece

Acceptance Criteria

- A functional Python API application built by docker
- Implementation of the three required operations
- A series of `curl` commands demonstrating how to use the API to simulate "taking a piece" (moving one piece to another's location and removing the taken piece).
- Ability to discuss architectural decisions: framework choice, routing design, payload design, API ergonomics, future proofing
- Commit to a github repo and share the URI with us