Policy Enforcement Assessment: Stage 1

Definitions

A **policy** is a group of definitions of what traffic is allowed or forbidden.

Each policy has the following fields:

- `name` - a textual name, consisting of at most 32 alphanumeric characters and underscores
- `description` - free text

Policy `name`s must be unique.

Requirements

Given the provided skeleton, fill in methods of the `PolicyAPI` class to implement a programmatic API for managing policies. Data should be stored in memory and does not need to be saved to a persistent data store. All methods should take arguments and return values which are JSON strings. Error conditions should raise exceptions.

- `create_policy()` should take as its only argument a JSON string containing an object whose keys are the fields of the policy, and should return a JSON string containing something which can be used to identify the policy in the future.
- `list_policies()` should return a JSON string containing an array of objects, each of which represents a policy and should have keys corresponding to each of the policy's fields.

Other Instructions

- Base your solution on the skeleton provided. Don't change the signatures of the class or methods or the names of the policy fields. You may either edit the skeleton file itself or copy it aside to a new solution file.
- There are a couple of baseline tests provided to help you bootstrap. Make sure they pass before you improve anything else. The tests assume that the solution is implemented in files named `stage1.py` and `stage2.py` on the python path.