

# In memory key value database

This is a simple, in-memory key-value store and queue server written in Go. It supports basic key-value operations and a simple queue system. The server listens on port 8080 and accepts HTTP POST requests with JSON payloads.

## Directory Structure

```
.
├── cmd
│   └── go-kvstore
│       └── main.go
├── go.mod
├── internal
│   ├── kvstore
│   │   └── kvstore.go
│   └── queue
│       └── queue.go
└── pkg
    └── api
        ├── api_test.go
        └── handler.go
```

## Usage

### Running the server

To run the server, simply navigate to the `cmd/go-kvstore` directory and run:

```
go run main.go
```

The server will start listening on port 8080.

### API

The server accepts HTTP POST requests at the `/api` endpoint with the following JSON payload:

```
{
  "command": "COMMAND_NAME ARG1 ARG2 ..."
}
```

Replace `COMMAND_NAME` with the desired command and provide the required arguments.

## Supported Commands

- `SET key value [EX seconds] [NX|XX]`: Sets the value for the given key. Optional flags:
  - `EX seconds`: Set a timeout for the key in seconds.
  - `NX`: Set the value only if the key does not exist.
  - `XX`: Set the value only if the key already exists.
- `GET key`: Returns the value associated with the given key.
- `Q PUSH key value1 value2 ...`: Pushes values to the queue with the given key.
- `Q POP key`: Pops and returns the first value from the queue with the given key.
- `BQ POP key timeout`: Pops and returns the first value from the queue with the given key, waiting up to `timeout` seconds if the queue is empty.

## Example Requests

Here are some example requests using `curl`:

```
# Set a key-value pair
```

```
curl -X POST -H "Content-Type: application/json" -d '{"command": "SET key1 value1"}' http://localhost:8080/api
```

```
# Get the value for a key
```

```
curl -X POST -H "Content-Type: application/json" -d '{"command": "GET key1"}' http://localhost:8080/api
```

```
# Push values to a queue
```

```
curl -X POST -H "Content-Type: application/json" -d '{"command": "Q PUSH queue1 value1 value2 value3"}' http://localhost:8080/api
```

```
# Pop a value from a queue
```

```
curl -X POST -H "Content-Type: application/json" -d '{"command": "Q POP queue1"}' http://localhost:8080/api
```

```
# Blocking pop from a queue with a timeout
```

```
curl -X POST -H "Content-Type: application/json" -d '{"command": "BQ POP queue1 10"}' http://localhost:8080/api
```

## Testing

To run the tests for the API package, navigate to the `pkg/api` directory and execute:

```
go test
```

## Dependencies

There are no direct third party dependencies, used by the project. You can know more about it in the `.mod` file.