

## Contents

<b>1</b>	<b>DB Schema</b>	<b>1</b>
<b>2</b>	<b>Database API</b>	<b>2</b>
2.1	addUser . . . . .	2
2.2	getAllServices . . . . .	2
2.3	checkIfServiceExists . . . . .	2
2.4	addService . . . . .	2
2.5	removeService . . . . .	3
2.6	updateService . . . . .	3
<b>3</b>	<b>Application API</b>	<b>3</b>
3.1	list . . . . .	3
3.2	add . . . . .	4
3.2.1	Name (required - no duplicates) . . . . .	4
3.2.2	Password (required) . . . . .	4
3.2.3	URL (not required) . . . . .	5
3.2.4	User Name (not required) . . . . .	5
3.3	remove . . . . .	5
3.4	edit . . . . .	5
3.5	getname . . . . .	6
3.6	getpass . . . . .	6
3.7	geturl . . . . .	6
3.8	open . . . . .	6

## 1 DB Schema

Database will be located in the ~/.passman/ folder

```
{
  "_id": 1,
  "name": "encryped name (key=master password)",
  "password": "hashed password",
  "data": [
    {
      "_id": 1,
      "service": "encrypted name of service (key=master pw; cannot be duplicate)",
      "servicePassword": "encryped service password (key = master pw)",
      "serviceUrl": "encryped service url (key = master pw)",
```

```

        "serviceName": "encrypted service name (key = master pw)"
    }
]
}

```

## 2 Database API

### 2.1 addUser

Adds a new user document to the database.

Parameters:

1. name: the name of the new user
2. pw: the master password for the new user

Return: True if successful, False if not

### 2.2 getAllServices

Return a list of all services as a dictionary

Parameters: none

Return: The array of dictionaries, each representing a service

### 2.3 checkIfServiceExists

Checks for a duplicate name in the database

Parameters:

1. name: the name of the service to check

Return: True if service exists, False if not

### 2.4 addService

Add a service to a user's document

Parameters:

1. name: the name of the new service
2. pw: the password for the new service
3. serviceUrl [optional]: the url of the service. Defaults to "" if not provided

4. `serviceUserName` [optional]: the user name for the new service. Defaults to "" if not provided

Return: True if successful, False if not

## 2.5 `removeService`

Remove a service from a user's document

Parameters:

1. `name`: the name of the service to remove

Return: True if successful, False if not

## 2.6 `updateService`

Updates a service to a new set of data. Be sure to provide the entire fingerprint of the new data, not just what changed. This includes any url, username, etc. even if they have not been changed

Parameters:

1. `oldName`: the old name of the service
2. `newName`: the new name of the service
3. `pw`: the new password for the service
4. `serviceUrl` [optional]: the new url of the service. Defaults to "" if not provided - WILL OVERWRITE EXISTING URL TO "" IF NOT PROVIDED
5. `serviceUserName` [optional]: the new user name for the service. Defaults to "" if not provided - WILL OVERWRITE EXISTING URL TO "" IF NOT PROVIDED

Return: True if successful, False if not

# 3 Application API

## 3.1 `list`

Called from user menu or via:

```
passman list
```

- List services as:
  1. Google | Sam  
https://www.google.com
  2. Yahoo | Sam  
https://www.yahoo.com
  3. Github | Sam  
https://www.github.com

## 3.2 add

Add a new service to the database

- Called from user menu or via

`passman add`

- Walk user through each of the following:

### 3.2.1 Name (required - no duplicates)

- Name of the service
- Must be input by user
- Cannot match any existing names in db

### 3.2.2 Password (required)

- Password for service.
- Prompt for manual entry or Auto generation

#### 1. Auto Generate (Default)

- prompt for the following:
  - (a) Length
    - Default: 30 characters
  - (b) Caps - Yes | No | Either

- Default: either
- (c) Lowercase - Yes | No | Either
  - Default: either
- (d) Integers - Yes | No | Either
  - Default: either
- (e) Special Chars - Yes | No | Either
  - Default: either

## 2. Manual entry

- prompt for password

### 3.2.3 URL (not required)

- URL for login page of service

### 3.2.4 User Name (not required)

- User name or email associated with service

## 3.3 remove

Remove a service and all associated data from the database

- Called via menu or command line
  - command line takes an optional service name as a parameter (see below)
  - if no service name is provided, user will be prompted for one

`passman remove [service name]`

## 3.4 edit

Walk user through the same steps as "Add". Prompt user to edit each field, or press enter to keep the data the same

- Called via menu or command line
  - command line takes an optional service name as a parameter (see below)
  - if no service name is provided, user will be prompted for one

`passman edit [service name]`

### 3.5 getname

Copies user name for a service to clipboard for 30 seconds

- Called via menu or command line
  - command line takes an optional service name as a parameter (see below)
  - if no service name is provided, user will be prompted for one

```
passman getname [service name]
```

### 3.6 getpass

Copies user password for a service to clipboard for 30 seconds

- Called via menu or command line
  - command line takes an optional service name as a parameter (see below)
  - if no service name is provided, user will be prompted for one

```
passman getpass [service name]
```

### 3.7 geturl

Copies service url to clipboard for 30 seconds

- Called via menu or command line
  - command line takes an optional service name as a parameter (see below)
  - if no service name is provided, user will be prompted for one

```
passman geturl [service name]
```

### 3.8 open

Opens the url for a service in the browser

- Called via menu or command line

- command line takes an optional service name as a parameter (see below)
- if no service name is provided, user will be prompted for one

```
passman open [service name]
```