Kigo User Manual – 2024

Table of Contents Page

Introduction

Kigo is a web app that generates haikus using lyrics from the user's most-listened songs! This is achieved by utilizing several APIs to pull the user's listening history, identify the user's most-listened songs, and then retrieve lyrics from these songs. Our algorithms count the syllables and parse these lyrics to create personalized, randomly generated haikus.

System Requirements

Operating System:
The Kigo web app can be run on any modern OS as long as the appropriate software is installed.

For example,

- Windows (e.g., Windows 10 or 11)
- macOS (e.g., Monterey or later)
- Linux (e.g., Ubuntu 20.04 or equivalent distributions)

Node.js

- Version 14.x or later
- Including npm (Node Package Manager) for installing dependencies
- Install the appropriate version from the official Node.js website

To verify installation, run:

```
node –v
npm –v
```

Maven

- Version 3.6.0 or later
- Install the appropriate version from the Apache Maven website

To verify installation, run:

```
mvn –version
```

Java Development Kit (JDK)

- Version 11 or later
- Install the appropriate version from the Oracle JDK website

To verify installation, run:

```
java –version
```

Web Browser

- Any modern web browser such as Safari or Google Chrome will be sufficient

Internet Connection

- A stable broadband connection for accessing the APIs is required

Firewall/Antivirus

- Ensure that there are no restrictions on the ports used (30000 for the React app and 520000 for the Node.js backend)

Spotify Developer Account

- A Spotify developer account is required to generate a client ID and client secret for Spotify API access
- To create an account, visit the official Spotify for Developers website

Text Editor/IDE (Integrated Development Environment)

- Although an IDE (such as Eclipse or Visual Studio) is not required, having one installed is helpful for managing or editing the source code

Installation Instructions

1. **Clone the repository**

   Clone only the "kigo-source" branch of this repository. You can do this either using bash or GitHub Desktop.

```
git clone -b kigo-source https://github.com/samallenonline/CS3300-003-Kigo.git
```

2. **Import Projects**
   Open your IDE (e.g., Eclipse) and import the following projects:
   - “kigokelly” and “kigonoah”: Import these as Maven projects.
     - In Eclipse, go to File > Import > Maven > Existing Maven Projects.
     - Browse to the folder where “kigokelly” and “kigonoah” are located and import them.
   - “kigo”: Import this as a normal project.
     - In Eclipse, go to File > Import > General > Existing Projects into Workspace.
     - Select Import existing files into workspace, browse to the “kigo” folder, and complete the import.

Project Folder Structure

Your project folder structure should look like this:

- “kigo”: Main Node.js backend project.
- “kigokelly”: Java Maven project containing functionality relating to Spotify authorization, retrieving the user's top tracks, and retrieving lyrics for these top tracks.
- “kigonoah”: Java Maven project containing core functionality relating to lyric processing and haiku generation.

3. **Install Dependencies for the Node.js App**
   Open a terminal, navigate to the kigo directory, and run this command to install dependencies for the Node.js app:

```
npm install
```

3. **Set up environment variables**

Create a file in the root directory where the imported projects are located with the name ".env". This file will initialize important environment variables that are used in the source code. Replicate the example below and replace "your_client_id" and "your_client_secret" with the actual values.

```
SPOTIFY_CLIENT_ID=your_client_id
SPOTIFY_CLIENT_SECRET=your_client_secret
REDIRECT_URI=http://localhost:52000/callback
REACT_APP_GITHUB_PAGES_URL=http://localhost:3000
REACT_APP_BACKEND_URL=https://localhost:52000
```

Running the App Locally

1. **Start the React App (Frontend) Navigate to the kigo/docs directory:**

```
cd ../kigo/docs npm start
```

This will start the React app and open it in your browser.

2. **Start the Node.js Backend**
   Open a new terminal window and navigate to the kigo directory:

```
cd ../kigo node index.js
```

This will start the Node.js backend at http://localhost:3000.

After these steps, you will be able to run the kigo-app locally on your machine. Please continue to the “Interface Overview” and “Features” sections of this manual to get more information on navigating the GUI and using certain functionality.

Interface Overview

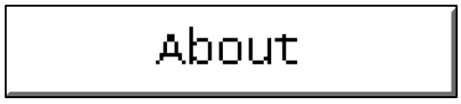
When accessing the Kigo website, you are presented with the Kigo interface. The interface is comprised of two large gray boxes located in the center of the website. The wide box on the left acts as a screen to display information. This screen changes its display according to whichever button the user clicks and presents the corresponding page. By default, the screen always presents the “Home” page. Directly below this box are white buttons that provide information about Kigo and the Kigo Team. These include the “About” and “Contact Us” buttons.

The tall box on the right contains gray buttons relating to the user’s account and haikus. These include the “Home”, “Account”, and “Haiku Gallery” buttons.

[Screenshot taken by the author]

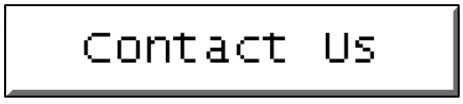
Features

“About” Page



When clicked, the “About” button displays information relating to the current limitations of the Kigo app, licensing information, and methods of contact. This page cannot be interacted with and is solely to provide general information about the app to users.

“Contact Us” Page



When clicked, the “Contact Us” button displays methods for the user to contact the Kigo team or submit suggestions for improvement. An email and a link to the Kigo GitHub Repository are provided.

“Home” Page

The “Home” page is displayed by default whenever a user first accesses the Kigo website. This page provides a welcome message to the user and briefly explains how Kigo works. Below this text, there is a gray “OK” button. When the user clicks the “OK” button, the user proceeds to the Spotify authentication page. This page prompts the user to connect their Spotify account and provides another gray button that reads “Connect Spotify”. Once the user clicks on this button, they are redirected outside of the Kigo website to an external page where they can login to their Spotify account. If authentication is successful, the user is redirected back to the Kigo website and is presented with the haiku generation page.

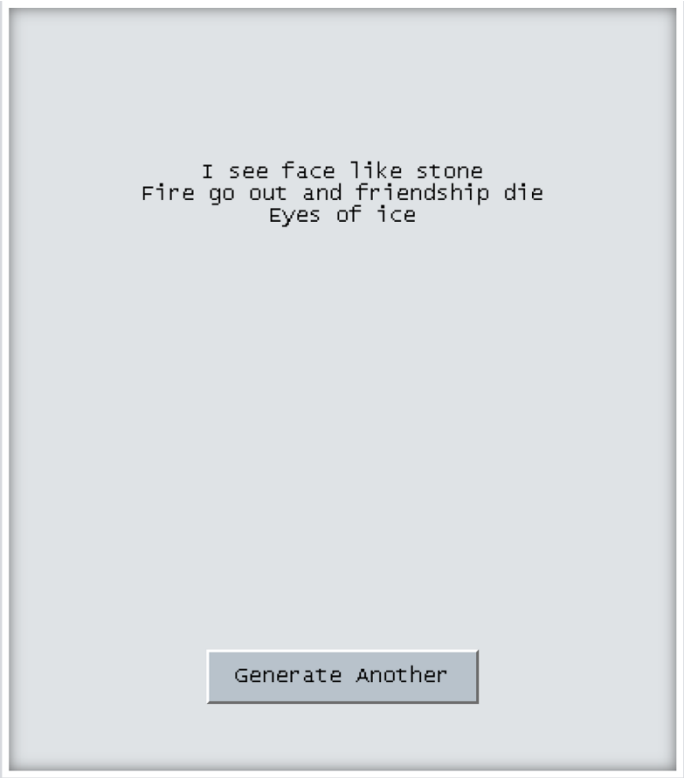
The user may choose to revisit the “Home” page by clicking the gray “Home” button on the right-hand side of the website. However, if the user has already connected their Spotify account, the Spotify authentication page will be skipped once the user clicks on the “OK” button again. If the user wants to disconnect their Spotify account or connect to a different account, they will have to navigate to the user preferences page where the “Log Out” button is provided.

Haiku Generation Page

The haiku generation page becomes available only after the user has successfully authenticated their Spotify account. This page provides a message thanking the user for connecting their account and refers to the user using their Spotify username. Below this text, there is a gray “Generate Haiku” button. When clicked, the user proceeds to the haiku display page.

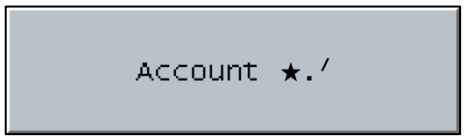
Haiku Display Page

When the user clicks on the “Generate Haiku” button, the user proceeds to the haiku display page. At this point, several haikus have been generated using lyrics from the user’s most-listened tracks (if lyrics for these tracks have been successfully retrieved). This page displays a haiku that has been randomly picked from a list of all the haikus generated. Below the haikus, there is a gray “Generate Another” button. When clicked, another haiku is randomly chosen and displayed. The user may shuffle through the generated haikus by clicking on this button as many times as they want.



The user may choose to revisit both the haiku generation and haiku display pages by clicking the gray “Home” button on the right-hand side of the website and proceeding through the steps once more. As mentioned previously, these pages are only available after the user has successfully authenticated their Spotify account.

“Account”



When clicked, the “Account” button provides several preferences for the user to toggle, a display of the Spotify account connected, and several buttons relating to the user’s profile.

The two preferences available are “Allow Explicit Content” and “Enable Baby Mode”. When toggled on, the “Allow Explicit Content” preference enables explicit lyrics to be used in haiku generation. However, for

users under the age of 18, this preference is permanently toggled off. This data is obtained by the Spotify API once the user connects their Spotify account. By default, this preference is toggled on unless the user is underage. When the “Enable Baby Mode” preference is toggled, all explicit words used in haiku generation are replaced with silly, baby words. This is purely used as an experimental method of censorship. All users can toggle this preference on and off, regardless of age. At the moment, the Kigo Team has not yet implemented the functionality for this preference.

Directly below the user preferences displays the account name and profile picture associated with the Spotify account that is currently connected. At the very bottom of the page, there are two gray buttons. These are the “Clear Local Data” and “Logout” buttons. When clicked, the “Clear Local Data” button will clear all user data that is locally stored, including preferences and any saved haikus. Clicking the “Logout” button will disconnect the Spotify account that is currently associated with the user. After logging out, the user may choose to connect their Spotify account again through the “Home” page.

“Haiku Gallery”

The haiku gallery functionality has not yet been fully integrated into the Kigo web app by the Kigo Team developers. Currently, the “Kigo Gallery” button is non-functioning.

Troubleshooting

Node.js or backend issues:

- Ensure all dependencies are installed (npm install) and that the Node.js server is running (node index.js).

Spotify authentication issues:

- Ensure the Spotify Developer app's Redirect URI is set to: http://localhost:3000/callback.

JAR file not found:

- Ensure the compiled JAR files are located in the /libs folder of the “kigo” project.

Contact Us

If you wish to reach out to the Kigo team, feel free to email us at pulsar3k@gmail.com.

Our GitHub repository contains our source code, contribution guidelines, and comprehensive documentation to help you get started with Kigo or contribute to our project

If you have any feature requests or suggestions for improvement, please create an issue on our GitHub page. We welcome all feedback and appreciate your support in making Kigo better!