koladaShiny

Introduction

The `koladaShiny` function is a Shiny application that allows users to visualize data from Kolada, a Swedish database containing various Key Performance Indicators (KPIs) for municipalities. This guide will walk you through setting up and using the app to fetch and display Kolada data.

The app utilizes the `AdvanceRConnectingToAPI` package for fetching data, providing a user-friendly interface to visualize data based on specified KPIs and years.

Prerequisites

Before using the `koladaShiny` function, make sure the following packages are installed:

- `shiny`: For building interactive web applications.
- `remotes`: For installing packages from GitHub.
- `AdvanceRConnectingToAPI`: For fetching data from Kolada.

```
# Install necessary packages
install.packages("shiny")
#> Installing package into 'C:/Users/Nisal_Amashan/AppData/Local/R/win-library/4.4'
#> (as 'lib' is unspecified)
#> package 'shiny' successfully unpacked and MD5 sums checked
#>
#> The downloaded binary packages are in
#>  C:\TEMP\RtmpILOmHb\downloaded_packages
install.packages("remotes")
#> Installing package into 'C:/Users/Nisal_Amashan/AppData/Local/R/win-library/4.4'
#> (as 'lib' is unspecified)
#> package 'remotes' successfully unpacked and MD5 sums checked
#>
#> The downloaded binary packages are in
#>  C:\TEMP\RtmpILOmHb\downloaded_packages
install.packages("koladaShiny")
#> Installing package into 'C:/Users/Nisal_Amashan/AppData/Local/R/win-library/4.4'
#> (as 'lib' is unspecified)
#> Warning: package 'koladaShiny' is not available for this version of R
#>
#> A version of this package for your version of R might be available elsewhere,
#> see the ideas at
#> https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages
remotes::install_github("SpikeStriker/AdvanceRConnectingToAPI")
#> Skipping install of 'AdvanceRConnectingToAPI' from a github remote, the SHA1 (7f5153fd) has not chan
#>   Use `force = TRUE` to force installation
```

```r
# Load necessary packages
library(shiny)
library(remotes)
library(AdvanceRConnectingToAPI)
library(koladaShiny)
```

Function Overview

The `koladaShiny` function launches a Shiny app with a basic user interface that allows the user to:

1. Specify one or more KPI codes.
2. Specify the years for which data should be fetched.
3. Click a button to fetch and display the data.

The app also displays the URL used for the API request and shows the fetched data in a table format.

Shiny App Components

The Shiny app has two main components:

1. **User Interface (UI)**:
   - **Title Panel**: Displays "Kolada Data."
   - **Sidebar Panel**: Contains input fields for KPI codes and years, along with a button to fetch data.
   - **Main Panel**: Displays the API request URL and the fetched data.
2. **Server Logic**:
   - The server processes the user input to make API calls using the `AdvanceRConnectingToAPI` package and then displays the URL and data.

Basic Usage

To run the Shiny app, execute the following command:

koladaShiny()

User Inputs

1. **KPI Codes**: Enter one or more KPI codes separated by commas (no spaces).
   - Example: `N40005,N25026`
2. **Years**: Enter one or more years separated by commas.
   - Example: `2019,2020,2021`

Example Workflow

1. **Start the Shiny app** by running `koladaShiny()`.
2. **Enter the KPI codes** you are interested in.
3. **Enter the years** for which you want data.
4. **Click "Fetch Data"** to display the results.

Understanding the Outputs

The app produces two outputs:

1. **URL Used for the API Request**: Shows the URL constructed based on the user's input, which is useful for understanding how data is fetched.
2. **Fetched Data Table**: Displays the data retrieved from Kolada, including columns such as KPI, Municipality, Year, and Value.

Handling Errors

The app includes basic error handling, such as checking if the `AdvanceRConnectingToAPI` package is installed. If not, it will stop and notify the user.

Extending the App

You can extend the `koladaShiny` function to include more advanced functionalities such as:

1. **Additional Filtering**: Add more input fields for filtering by municipalities or other criteria.
2. **Visualizations**: Include graphs or charts to visualize trends in the data.
3. **Data Export**: Allow users to download the fetched data as a CSV file.

Conclusion

The `koladaShiny` app provides a simple way to interact with and visualize Kolada data using an intuitive interface.