**# Feature and Tag Selector Utility**

## Overview

The \*\*Feature and Tag Selector Utility\*\* is a Node.js-based application designed to manage `.feature` files and their associated BDD tags. It provides a user-friendly interface to:

- Generate `.feature` files with unique tags.

- Extract tags from existing `.feature` files.

- Generate Playwright runner commands based on selected features and tags.

## Features

1. \*\*Feature File Generation\*\*:

- Automatically creates folders and `.feature` files with unique tags.

- Supports multiple folders (`Createaccount`, `policyinfo`, `location`, etc.).

- Each folder contains 50 `.feature` files with dummy scenarios.

2. \*\*Tag Extraction\*\*:

- Extracts unique BDD tags from selected `.feature` files.

- Displays tags in a user-friendly interface for selection.

3. \*\*Command Generation\*\*:

- Generates Playwright runner commands based on selected features and tags.

- Supports multiple runner types (`browserstack`, `runner`).

- Allows copying or downloading the generated command as a `.bat` file.

4. \*\*Dynamic Folder and File Management\*\*:

- Dynamically lists subfolders and `.feature` files.

- Supports recursive folder scanning.

## Folder Structure

```

Project/

├── e2e/

│ └── features/

│ ├── Createaccount/

│ │ ├── feature\_1.feature

│ │ ├── feature\_2.feature

│ │ └── ...

│ ├── policyinfo/

│ ├── location/

│ ├── Uivalidation/

│ ├── APIvalidation/

│ └── DBTesting/

├── featuretagserver.js

├── generateFeatures.js

└── GenerateFeaturesAndTags.html

```

## Installation

1. Clone the repository:

```bash

git clone <repository-url>

cd Project

```

2. Install dependencies:

```bash

npm install

```

3. Run the server:

```bash

node featuretagserver.js

```

4. Open the application in your browser:

```

http://localhost:3000

```

## Usage

### 1. Generate Feature Files

- Run the `generateFeatures.js` script to create folders and `.feature` files:

```bash

node generateFeatures.js

```

- This will create 50 `.feature` files in each folder under `e2e/features`.

### 2. Extract Tags

- Navigate to the \*\*Extract Tags from Feature Files\*\* tab in the UI.

- Enter the names of `.feature` files (one per line) and click \*\*Extract Tags\*\*.

- The extracted tags will be displayed in the UI.

### 3. Generate Playwright Runner Command

- Select features and tags from the \*\*Select Features and Tags\*\* tab.

- Choose a runner type (`browserstack` or `runner`).

- Click \*\*Generate Command\*\* to create the Playwright runner command.

- Copy or download the command as needed.

## API Endpoints

### `/list-files`

- \*\*Method\*\*: GET

- \*\*Query Parameter\*\*: `dir` (directory path)

- \*\*Description\*\*: Lists all `.feature` files in the specified directory.

### `/list-subfolders`

- \*\*Method\*\*: GET

- \*\*Query Parameter\*\*: `dir` (directory path)

- \*\*Description\*\*: Lists all subfolders in the specified directory.

### `/feature-tags`

- \*\*Method\*\*: GET

- \*\*Query Parameter\*\*: `file` (file path)

- \*\*Description\*\*: Extracts tags from the specified `.feature` file.

### `/tags-from-files`

- \*\*Method\*\*: POST

- \*\*Body\*\*: `{ "files": [filePaths] }`

- \*\*Description\*\*: Extracts tags from multiple `.feature` files.

## Technologies Used

- \*\*Node.js\*\*: Backend server and file operations.

- \*\*Express.js\*\*: API endpoints.

- \*\*HTML/CSS/JavaScript\*\*: Frontend interface.

- \*\*Bootstrap\*\*: Responsive UI design.

- \*\*Playwright\*\*: Integration for BDD testing.

## Screenshots

### 1. Feature and Tag Selector

A screenshot of a computer

AI-generated content may be incorrect.

### 2. Extract Tags

Extract Tags] ![A screenshot of a computer

AI-generated content may be incorrect.

### 3. Generated Command

Generated Command]![A screenshot of a computer

AI-generated content may be incorrect.