**DOCUMENTACIÓN PROYECTO AGENCIA DE VIAJE**

**JHONATAN PELAEZ MAGNOLER**

**JAMUNDÍ, VALLE DEL CAUCA**

**09 JUNIO 2024**

**FORMHELPSERVICE**

El servicio FormhelpService es una parte integral de la aplicación Angular que proporciona varias funcionalidades relacionadas con la gestión de formularios, obtención de datos de APIs externas y almacenamiento temporal de datos. Este documento detalla su propósito, funcionalidades y cómo integrarlo y utilizarlo dentro de la aplicación.

Tiene como objetivo centralizar y gestionar las operaciones relacionadas con la manipulación de datos de formularios, obtención de información de diversas APIs y almacenamiento temporal de datos dentro de la aplicación Angular.

**FUNCIONALIDADES**

* Almacenamiento temporal de datos de formularios.
* Obtención de listas de países, ciudades e información del PC desde un servidor local.
* Consultar el historial basado en IP y nombre del PC.
* Obtener información del clima de una ciudad específica utilizando la API de OpenWeatherMap.
* Convertir moneda utilizando la API de GetGeoApi.
* Enviar datos históricos a un servidor para su almacenamiento.

**MÉTODOS**

* **setFormData(formData: any):** Almacena datos del formulario temporalmente.
* **getFormData():** Retorna los datos del formulario almacenados.
* **setFormDataDos(formDataDos: any):** Almacena un segundo conjunto de datos del formulario.
* **getFormDataDos():** Retorna el segundo conjunto de datos del formulario almacenados.
* **getPaises():** Retorna una lista de países desde el servidor local.
* **getPC():** Retorna información del PC desde el servidor local.
* **getCiudades():** Retorna una lista de ciudades desde el servidor local.
* **getHistorial(ip: string, nombrepc: string):** Retorna el historial basado en IP y nombre del PC.
* **getClima(ciudad: string):** Retorna la información del clima de una ciudad específica.
* **getcash(from: string, to: string, cash: number):** Convierte una cantidad de dinero de una moneda a otra.
* **postHistorial(historial: any):** Envía datos históricos al servidor para su almacenamiento.

**DEPENDENCIAS**

* HttpClient de @angular/common/http para realizar peticiones HTTP.
* HttpHeaders de @angular/common/http para configurar encabezados HTTP.
* Observable de rxjs para manejar respuestas asincrónicas.

**PROCESOSSERVICE**

El servicio ProcesoService es una parte esencial de la aplicación Angular que orquesta varios procesos utilizando el FormhelpService. Este documento proporciona una descripción completa del servicio, sus funcionalidades, y cómo utilizarlo dentro de la aplicación.

Coordina la obtención de datos de diferentes fuentes y realiza procesos complejos como la conversión de moneda y la recopilación de información climática. Su objetivo es centralizar estas operaciones para facilitar el flujo de datos en la aplicación.

**FUNCIONALIDADES**

* Inicializa y coordina el proceso de obtención de datos desde múltiples servicios.
* Almacena temporalmente datos climáticos y de ubicación.
* Realiza la conversión de moneda.
* Envía datos procesados a un servidor para su almacenamiento.

**DEPENDENCIAS**

* FormhelpService: Para obtener datos de APIs y manejar formularios.
* combineLatest de rxjs: Para combinar múltiples observables y esperar sus resultados.

**SUBSCRIBES**

* **This.\_formServ.getClima():** Obtiene la información del clima, se debe enviar un parámetro tipo String (Nombre de la ciudad).
* **This.\_formServ.getPaises():**  Obtiene toda la lista de los países almacenados en la base de datos.
* **This.\_formServ.getPC():**  Obtiene la IP y el nombre del ordenador desde donde se esta ejecutando el sistema.
* **This.\_formServ.getCiudades():**  Obtiene toda la lista de las ciudades almacenadas en la base de datos.
* **This.\_formServ.postHistorial():** Almacena toda la información que contiene el objeto datosprincipales como lo es (PAISDESTINO, MONEDAREPRESENTACION, CIUDADDESTINO, PRESUPUESTO, LONGITUD, LATITUD, HUMEDAD, TEMPERATURA, TEMPERATURAMAXIMA, TEMPERATURAMINIMA, NOMBREMONEDA, PRECIO, CONVERSION, IPCONSULTA, NOMBREPC, ORIGEN, FK\_CIUDAD).
* **This.\_formServ.getcash():** Consulta de la API para obtener información de la moneda del país destino, se debe enviar parámetros como siglas de la moneda origen (se encuentra en la variable idioma del servicio formhelpservice), sigla de la moneda destino (viene en la BD) y el presupuesto (en el formulario formData).

**MÉTODOS**

* **Inicioproceso():** Es donde inicia la captura de los dos formularios (formData y formDataDos) que se encuentran en formhelpservice y realiza la consulta a la API OpenWeatherMap para obtener datos como el clima de la ciudad seleccionada.
* **Convercash():** Proceso de consulta de la API GetGeoApi para obtener toda la información de la moneda del país destino como lo es el nombre de la moneda, siglas de la moneda y la conversión arrojada a la suscripción del método getcash();

**MENUCOMPONENT**

El componente MenuComponent es un componente de Angular encargado de la navegación y la gestión de formularios en la aplicación. Este documento proporciona una descripción detallada del componente y cómo utilizarlo en el proyecto.

Tiene como propósito principal manejar la navegación entre páginas y la gestión de formularios en la aplicación Angular. Ofrece funcionalidades para cambiar el idioma de la aplicación, ver el historial de consultas, avanzar y retroceder entre páginas de un formulario, y volver a la página de inicio.

**FUNCIONALIDADES**

* Cambio de idioma de la aplicación.
* Navegación al historial de consultas.
* Navegación entre páginas de un formulario.
* Función para volver a la página de inicio.

**MÉTODOS**

* **changeLanguage(language: string):** Cambia el idioma de la aplicación según el parámetro language especificado.
* **viewHistory():** Navega al historial de consultas.
* **sgtepage():** Avanza a la siguiente página del formulario si los campos están completos y válidos.
* **regresar():** Retrocede a la página anterior del formulario.
* **inicio():** Navega a la página de inicio del formulario.

**DEPENDENCIAS**

* **FormhelpService:** Para la gestión de formularios.
* **ProcesoService:** Para la coordinación de procesos en la aplicación.

**PAGE1COMPONENT**

El componente Page1Component es un componente de Angular que forma parte de un formulario multi-página. Este componente se encarga de mostrar la primera página del formulario, donde el usuario puede seleccionar un país y una ciudad. Este documento proporciona una descripción detallada del componente y cómo utilizarlo en el proyecto.

El propósito es permitir al usuario seleccionar un país y una ciudad como parte de un formulario multi-página en la aplicación Angular. Proporciona funcionalidades para cargar la lista de países y ciudades disponibles, así como para mantener la información del formulario actualizada.

**FUNCIONALIDADES**

* Carga la lista de países y ciudades disponibles.
* Actualiza dinámicamente la lista de ciudades según el país seleccionado.
* Almacena y actualiza la información del formulario en el servicio FormhelpService.

**DEPENDENCIAS**

* **FormBuilder y FormGroup de @angular/forms:** Para construir y gestionar formularios reactivos en Angular.
* **ReactiveFormsModule de @angular/forms:** Para habilitar la funcionalidad de formularios reactivos en Angular.
* **combineLatest` de rxjs:** Para combinar observables y obtener datos de múltiples fuentes.
* **FormhelpService de src/app/services/formhelp.service:** Para almacenar y compartir información del formulario entre componentes.

**SUBSCRIBES**

* **This.\_formServ.getPaises():**  Obtiene toda la lista de los países almacenados en la base de datos.
* **This.\_formServ.getCiudades():**  Obtiene toda la lista de las ciudades almacenadas en la base de datos.
* **this.form.get('PAIS')!.valueChanges:** Escucha los cambios en el campo PAIS del formulario para filtrar dinámicamente la lista de las ciudades e incluir solo aquellas que pertenecen al país seleccionado.
* **this.form.valueChanges:** Escucha los cambios en todo el formulario para actualizar el servicio formhelpservice con los datos del formulario y almacenarlos en la variable formData.

**PAGE2**COMPONENT

El componente forma parte de un formulario multi-página en una aplicación Angular. Este componente se encarga de recopilar y validar la información relacionada con el presupuesto del usuario.

El propósito es proporcionar una interfaz para que el usuario ingrese su presupuesto en el contexto de un formulario multi-página. Facilita la validación del presupuesto ingresado y su almacenamiento para su uso posterior en la aplicación.

**FUNCIONALIDADES**

* Permite al usuario ingresar su presupuesto en el campo de texto.
* Realiza la validación del presupuesto ingresado para asegurar de que se proporcione un valor valido.
* Almacena y actualiza dinámicamente el presupuesto ingresado en el servicio formhelpservice en la variable formDataDos para su uso en otros componentes del formulario.

**DEPENDENCIAS**

* **FormBuilder y FormGroup de @angular/forms:** Para construir y gestionar formularios reactivos en Angular.
* **FormhelpService de src/app/services/formhelp.service:** Para almacenar y compartir información del formulario entre componentes.

**SUBCRIBES**

* **This.form.valuechanges:** Escucha los cambios en el formulario del presupuesto para actualizar dinámicamente el servicio formhelpservice con los datos del formulario.

**PAGE3COMPONENT**

Es una parte del flujo de un formulario multi-página en una aplicación Angular. Este componente se encarga de iniciar el proceso asociado al formulario, como la recuperación de datos y cálculos necesarios para su presentación final.

El propósito es coordinar la inicialización del proceso relacionado con el formulario. Esto puede incluir la recuperación de datos necesarios, el cálculo de valores iniciales o cualquier otra acción necesaria para preparar el formulario para su presentación final.

**FUNCIONALIDADES**

* Iniciar el proceso asociado con el formulario a través del servicio procesoservice.
* Puede incluir la recuperación de datos adicionales o el calculo de valores iniciales según sea necesario para la presentación final del formulario.

**DEPENDENCIAS**

* **ProcesoService de src/app/services/proceso.service:** Para iniciar y coordinar el proceso asociado con el formulario.

**SUBSCRIBE**

* **This.\_proServ.inicioproceso():** Inicia el proceso asociado con el formulario para la presentación final, al llamar al método, se inicia la recuperación de datos de los formularios, los cálculos y otras tareas necesarias.

**HISTORIALCOMPONENT**

Es responsable de mostrar el historial de consultas realizadas en la aplicación. Utiliza el servicio FormhelpService para obtener los datos del historial y mostrarlos en la interfaz de usuario.

El propósito es proporcionar al usuario una vista del historial de consultas realizadas en la aplicación. Esto permite al usuario revisar las consultas anteriores y obtener información relevante sobre las acciones realizadas previamente.

**FUNCIONALIDADES**

* Recuperdad los datos del historial de consultas utilizando el servicio FormhelpService
* Mostrar los datos del historial en la interfaz de usuario para que el usuario pueda revisarlos.
* Utilizar el operador combineLatest para esperar a que se obtengan los datos del PC (equipo) antes de recuperar el historial de consultas.

**DEPENDENCIAS**

* **Formhelpservice de src/app/services/formhelp.service:** Para acceder a los métodos que proporcionan los datos del historial de consultas.

**SUBCRIBES**

* **This.\_formServ.getPC():** Obtener los datos del pc para iniciar el proceso de recuperar el historial al consultar en la base de datos.
* **This.\_formServ.getHistorial():** Obtiene un resultado de los datos del historial enviando parámetros como la ip y el nombre del ordenados.