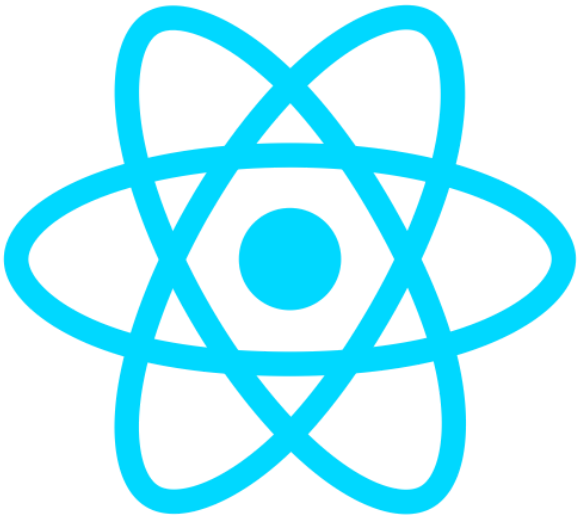
WORKSHOP REACT-NODE

GUÍA - EXTRAS



Práctica

1. Completar la API hecha en NodeJS
    a. Completar los servicios en `contact.services.js` con la documentación de lowdb
        - Los defaults ya se encuentran definidos
        - Agregar el `contact` recibido como parámetro, a la db (seguir el ejemplo de ***Add a post*** en el apartado ***Usage*** de la documentación)
        - Utilizar MomentJs para poder guardar el `contact` con la fecha actual de creación
        - Agregar manejo de errores con `try/catch` (opcional)
    b. Agregar manejo de errores al controlador `contact.controller.js` (opcional)
        - Validar que los datos del body sean correctos usando `contactHelper.validateRegisterRequest(body)` esta función devuelve `true` si el body tiene todos los datos requeridos.
        - Validar si el error que devuelve `contactServices.create` está vacío. Si no lo está, mostrar un mensaje de error por consola utilizando `__logger`
    c. Agregar las rutas de contact
        - Importar y usar las rutas de contact en el archivo `routes.js`
        - Llamar las rutas usando `app.use('/v1/contact', contactRoutes)`, debajo de las rutas de `movies`
2. Completar la página de `Contact`

a. Crear un nuevo archivo en la carpeta `pages`, llamado `Contact.js`

b. Crear un componente del tipo `class-based`

c. Guardar name, email y message en el `state` del componente

   - Crear el state tal como se hizo en el componente `MoviesList`
   - Agregar las tres propiedades mencionadas (name, email y message) e inicializarlas como cadenas vacías (‘’)

d. Crear con JSX un formulario con 3 inputs controlados y un botón

   - Crear el form usando los componentes de reactstrap
   - Para cada input agregar un atributo `value`, que tenga como valor `this.state.[nombre_de_la_propiedad]`
   - Para cada input agregar un atributo `onChange` que llame a la función `(event) => this.handleInputChange(event, [nombre_de_la_propiedad])`

e. Agregar el `handleInputChange` para los inputs

   - Utilizar `event` para poder obtener el valor del input que se encuentra dentro de `event.target.value`
   - Actualizar el valor del `state` del atributo correspondiente

f. Agregar el `handleSubmit` para el formulario

   - Usar `event.preventDefault()` para evitar que el formulario sea “enviado” cuando se haga click en guardar (comportamiento por defecto).
   - Usar axios para guardar el contacto en la API

Recursos

1. Código base
    a. NodeJS: https://github.com/RominaManzano/workshop-api-day2
    b. ReactJS: https://github.com/RominaManzano/workshop-web-day2