## 

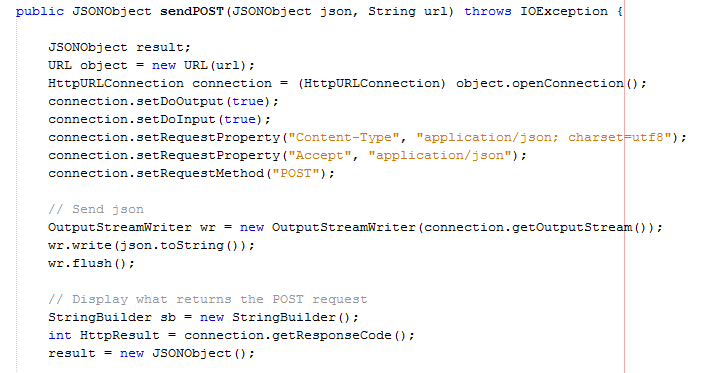
## Manual Técnico

En este manual se enfatizara la importancia de algunos módulos y código necesarios para que el sistema de notificaciones funcione adecuadamente. Comenzaremos mostrando código relacionado a la aplicación sensor, dado que es la aplicación que generará las respectivas notificaciones.

### Aplicación Sensor

Uno de los aspectos más importantes de la aplicación sensor es el hecho de poder enviar notificaciones, esta funcionalidad se encuentra directamente relacionada con la clase NotifierHTTP.java, en esta clase se definen todos los métodos y configuraciones necesarias para llevar a cabo las tareas de comunicación.

#### Clase NotifierHTTP





#### Clase Configurator

Otro aspecto importante de la aplicación sensor es el hecho de brindar a los usuarios la oportunidad de cambiar la configuración relacionada a la dirección URL del proveedor de servicios de esta forma la aplicación no está sujeta a generar alertas para un servicio específico, el código en la clase Configurator.java se encarga de llevar a cabo esta configuración de manera transparente para el usuario.

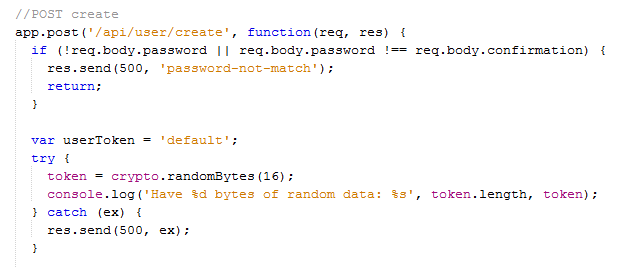


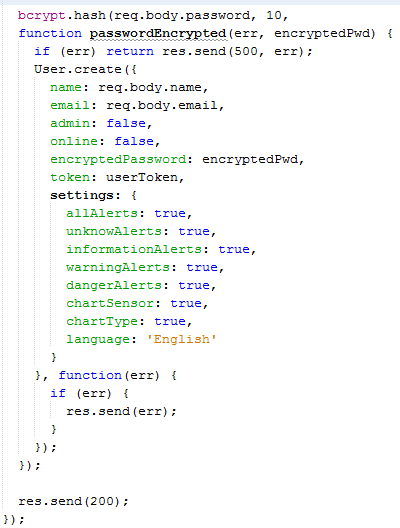
Toda la información que es manejada tanto en el lado del cliente como en el lado del servidor es en formato JSON, para permitir a distintas aplicaciones trabajar facilmente con esta información. En el código mostrada podemos notar que los valores relacionados a la URL, nombre de usuario, contraseña y TOKEN son leidos desde un archivo de configuración el cual es manejado por la clase FileManager.java.

### Aplicación Web de servicios

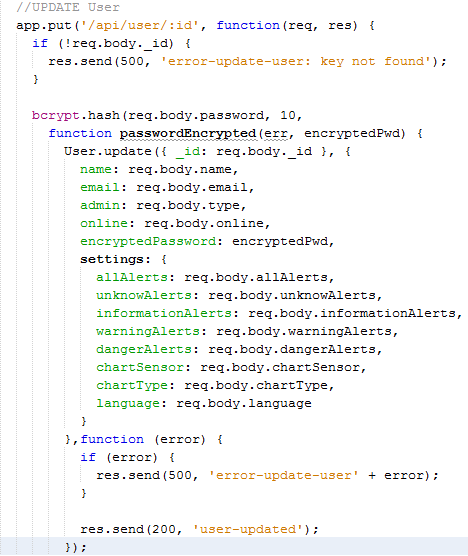
La aplicación web de servicios representa el corazón del sistema de notificaciones, esta aplicación es la encargada de gestionar las distintas solicitudes y respuestas que son generadas desde las distintas aplicaciones cliente hacia el servidor.

#### User.js (/api/user/create) Crear usuario





#### User.js (/api/user/:id) Modificar usuario

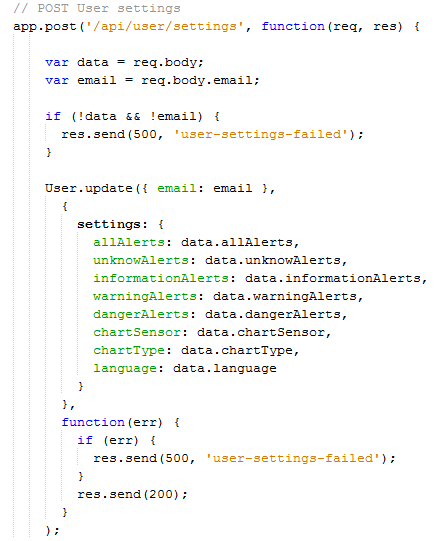


#### User.js (/api/user/login) Autenticar usuario

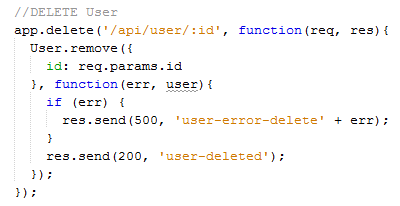




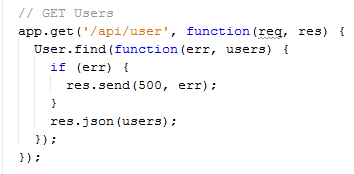
#### User.js (/api/user/settings) Configuraciones



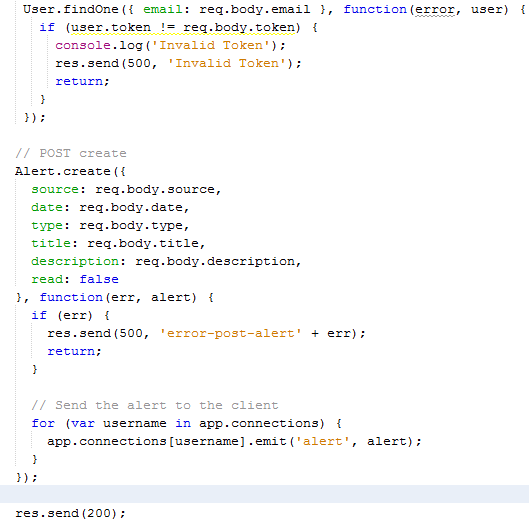
#### User.js (/api/user/:id) Eliminar usuario



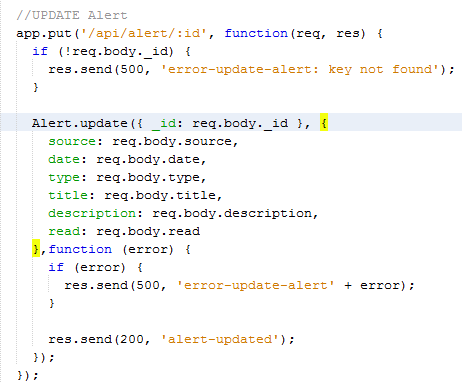
#### User.js (/api/user) Obtener usuarios



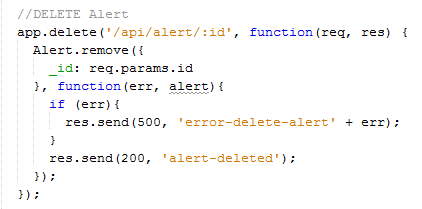
#### Alert.js(/api/alert/create) Crear alerta



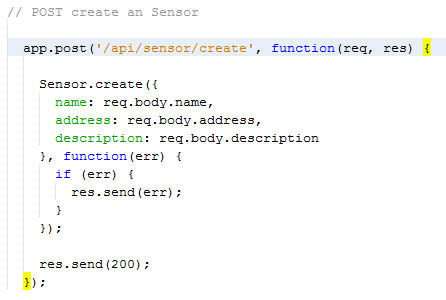
#### Alert.js (/api/alert/:id) Modifcar alerta



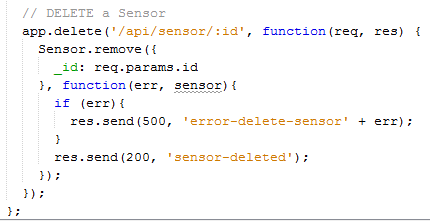
#### Alerts.js (/api/alert/:id) Eliminar alerta



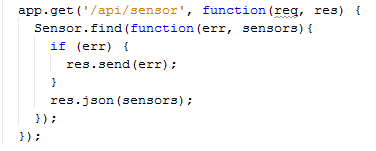
#### Sensor.js (/api/sensor/create) Crear sensor



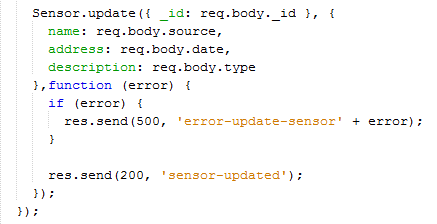
#### Sensor.js (/api/sensor/:id) Eliminar sensor



#### Sensor.js (/api/sensor) Obtener sensores



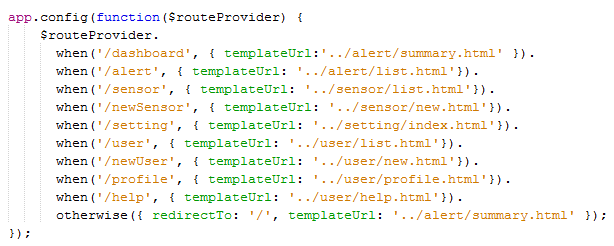
#### Sensor.js (/api/sensor/:id) Modificar sensor



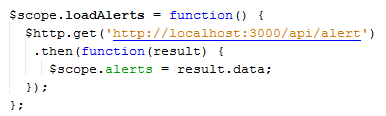
### Aplicación Web cliente

La aplicación cliente web permite a los usuarios acceder a toda la funcionalidad que ofrece el servicio web, atreves de navegadores web, esta aplicación es la encargada de ofrecer a los usuarios la posibilidad de habilitar o deshabilitar las alertas. Haciendo uso de las páginas de configuración los usuarios fácilmente deben llevar a cabo las configuraciones mencionadas. Los aspectos más importantes de esta aplicación están relacionados al hecho de obtener información y enviar información a las respectivas rutas para que estas se encarguen de mostrar las páginas adecuadas.

#### Main.js (MainController) Definición de rutas



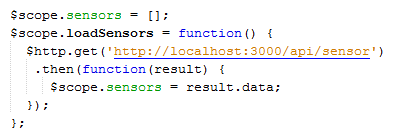
#### Main.js (AlertControlller) Cargar alertas



#### Main.js (SettingsController) Cargar y guardar configuraciones



#### Main.js (SensorController) Cargar sensores



#### Main.js (UserController) Auntenticar usuario



### Aplicación móvil cliente

La aplicación cliente móvil permite a los usuarios recibir las notificaciones con las respectivas alertas generadas, esta aplicación ofrece la posibilidad de mantener a los usuarios conectados a los servicios para contar con toda la información que es generada en tiempo real. Cabe mencionar que el aspecto más importante de esta aplicación es el hecho de mantener una conexión a través de WebSockets con el servicio web, el cual habilita un canal abierto para una comunicación bidireccional.

#### Utils.js (login) Autenticación de usuarios



#### Connection.js (init) Creación WebSocket

