

SmartMood Servicios



Instituto Tecnológico de Estudios Superior de Zamora
Ingeniería en tecnologías de la información y comunicaciones
Alumno: Ricardo Díaz Márquez,

Servicio Usuarios

Tipo de servicio: Entidad

Responsable: Ricardo Díaz

Operaciones expuestas:

1.- Crear usuario:

Actores: Usuarios, Administradores

Url: <http://SmartMood/Api/Usuarios>

Método HTTP: Post

Lógica de negocio: Permite crear usuarios con un ID que podrán crear etiquetas y entradas asociadas con dicho ID.

Entrada: {

Nombre : String

Contraseña : String

Correo : String

Edad : int

FechaCreacion : Date}

Salida: {

Mensaje : String}

Descripción: El método introduce un nuevo usuario en la colección Usuarios que contenga los datos especificados.

Validaciones:

Los campos no pueden estar vacíos

Contraseña: La contraseña **debe tener más de 4 dígitos y tener letras y números**

Correo: El correo debe **terminar en un dominio válido**. Los dominios válidos serán: @gmail.com y @accitesz.com

Edad: La edad debe ser **mayor a 2 años**

2.- Eliminar usuario

Actores: Usuarios, Administradores

Url: <http://SmartMood/Api/Usuarios/{idUsuario}>

Método HTTP: Delete

Lógica de negocio: Permite eliminar usuarios de la base de datos. Antes de ejecutarse, tiene que pasar por un proceso de validación que confirme que el usuario introdujo su contraseña correctamente o que es ejecutado por un administrador.

Entrada: {

Contraseña: String}

Salida: {

Mensaje : Usuario eliminado con éxito
ID: int }

Descripción:
El método elimina un usuario de la colección usuarios.

Validaciones:
La contraseña introducida debe coincidir con la contraseña del usuario que tiene el ID especificado.

3.- Cambiar contraseña

Actores: Usuarios

Url: <http://SmartMood/Api/Usuarios/{idusuario}>

Método HTTP: Put

Lógica de negocio: Permite cambiar la contraseña de un usuario. Antes de ejecutarse, tiene que pasar por un proceso de validación que confirme que el usuario introdujo su contraseña correctamente o que es ejecutado por un administrador.

Entrada: {
AntiguaContrasena: String
NuevaContrasena: String}

Salida: {
Mensaje : String
NuevaContrasena: String,
ID: int }

Descripción:

Validaciones:

AntiguaContraseña debe coincidir con la contraseña contenida en el usuario con el ID especificado. **NuevaContraseña debe cumplir los requerimientos** para hacer una nueva contraseña.

4.- Cambiar datos usuario

Actores: Usuarios

Url: <http://SmartMood/Api/Usuarios/{IdUsuario}/cambiardatos>

Método HTTP: Put

Los campos que contengan algo serán actualizados en el usuario con el id indicado. Los campos que no contengan nada o contengan "Null" serán dejados como están.

Entrada: {
ContrasenaActual: String
Nombre : String
Correo : String

Edad : int}

Salida: {
Mensaje : Str }

Validaciones: La contraseña introducida debe ser la misma que la del usuario.

5.- Ver usuarios

Actores: Administradores

Url: <http://SmartMood/Api/Usuarios>

Método HTTP: Get

Lógica de negocio: Permite ver todos los usuarios de la base de datos.

Entrada: No aplica

Mensaje: String

Usuarios: [{

Id: int

Nombre : String

Password : String

Correo : String

Edad : int

FechaCreacion : Date

}]

Validaciones: No aplica

Servicio Actividades

Tipo de servicio: Entidad

Responsable: Ricardo Díaz

Operaciones expuestas:

1.- Crear actividad:

Actores: Usuarios

Url: <http://SmartMood/Api/Actividades>

Método HTTP: Post

Lógica de negocio: Permite crear actividades para asignarlas a entradas de diario. Esta operación asume que fue realizada desde un end point donde el usuario inició sesión.

Entrada: {

IdUsuario: Int

Nombre : String

Descripcion : String
Icono: String
}

Salida: {
Mensaje : String
Actividad: ActividadSalida}

Validaciones: Se debe validar que el usuario referenciado existe.

2.- Eliminar Actividad:

Actores: Usuarios

Url: <http://SmartMood/Api/Actividades/{ActividadId}/Eliminar>

Método HTTP: Delete

Lógica de negocio: Permite eliminar actividades del usuario

Entrada: {
Contraseña: String
}

Salida: {
Mensaje : Etiqueta eliminada con éxito
}

Validaciones: La contraseña introducida debe coincidir con la del usuario que referencia la actividad con el id especificado.

3.- Modificar actividad:

Actores: Usuarios

Url: <http://SmartMood/Api/Actividades/{ActividadId}/Modificar>

Método HTTP: Put

Lógica de negocio: Los campos que contengan algo serán actualizados en la actividad con el id indicado. Los campos que no contengan nada o contengan "Null" serán dejados como están.

Entrada: {
Nombre : String
Descripcion : String
Icono: String
}

Salida: {
Mensaje : String
Actividad: ActividadSalida}

4.- Ver actividades del usuario:

Actores: Usuarios

Url: <http://SmartMood/Api/Actividades/{Id usuario}>
Método HTTP: Get
Lógica de negocio: Permite ver las etiquetas del usuario.

Entrada: No aplica

Salida:
Mensaje: String
Actividades: [{
Id : int
Nombre : String
Descripcion : String
Color : String
}]

5.- Ver todas las actividades
Actores: Desarrollador
Url: <http://SmartMood/Api/Actividades>
Método HTTP: Get
Lógica de negocio: Permite ver todas las actividades.

Entrada: No aplica
Salida:
Actividades: [
Id : int
Nombre : String
Descripcion : String
Color : String
Usuario : [
Nombre: String
Id: String
Correo: String]]

Servicio Emociones

Tipo de servicio: Entidad
Responsable: Ricardo Díaz
Operaciones expuestas:

1.- Ver todas las emociones
Actores: Desarrollador, Usuarios
Url: <http://SmartMood/Api/Emociones>
Método HTTP: Get
Lógica de negocio: Permite ver todas las emociones.

Entrada: No aplica

Salida: [{
Id : int
Nombre : String
Descripcion : String
Color : String}]

Servicio Entradas

Tipo de servicio: Entidad
Responsable: Ricardo Díaz
Operaciones expuestas:

1.- Crear entrada:

Actores: Usuarios

Url: <http://SmartMood/Api/Entradas>

Método HTTP: Post

Lógica de negocio: Permite crear entradas asociadas con el ID de un usuario.

Entrada: {
IdUsuario: Int,
Nota: String,
Emociones: [String],
Actividad: String,
Fecha: Date}

Salida: {
Mensaje : Entrada creada con éxito
Entrada: EntradaSalida}

Validaciones: Validar que el usuario exista.

2.- Eliminar entrada:

Actores: Usuarios

Url: <http://SmartMood/Api/Entradas/{EntradaId}/Eliminar>

Método HTTP: Delete

Lógica de negocio: Permite eliminar entradas de la base de datos.

Entrada: {
Contraseña: String
}

Salida: {
Mensaje : String
}

Validaciones: La contraseña introducida debe coincidir con la del usuario que referencia la entrada con el id especificado.

3.- Modificar entrada:

Actores: Usuarios

Url: <http://SmartMood/Api/Entradas/{IdEntrada}/Modificar>

Método HTTP: Put

Lógica de negocio: Los campos que contengan algo serán actualizados en la entrada con el id indicado. Los campos que no contengan nada o contengan "Null" serán dejados como están.

```
Entrada: {  
  Nota: String  
  Emociones: [String]  
  Actividad: String  
  Fecha: Date  
}
```

```
Salida: {  
  Mensaje : String  
  Entrada: EntradaSalida}
```

Validaciones: La entrada especificada debe existir.

4.- Ver entradas:

Actores: Usuarios

Url: <http://SmartMood/Api/Entradas>

Método HTTP: Get

Lógica de negocio: Permite ver todas las entradas.

Entrada: No aplica

```
Mensaje: String  
Entradas: [{  
  Id : int  
  Nota: String,  
  Emociones: [String],  
  Actividad: String,  
  Fecha: Date  
  usuario: [{  
    idUsuario: String  
    Nombre: String}]]]
```

5.- Ver entradas del usuario:

Actores: Usuarios

Url: <http://SmartMood/Api/Entradas/{IdUsuario}>

Método HTTP: Get

Lógica de negocio: Permite ver todas las entradas que pertenezcan a un usuario.

Entrada: No aplica

Mensaje: String

Usuario:{{

IdUsuario: String

Nombre: String}}

Entradas: [{

Id : int

Nota: String,

Emociones: [String],

Actividad: String,

Fecha: Date}}

5.- Ver entrada por ID:

Actores: Usuarios

Url: <http://SmartMood/Api/Entradas/{IdEntrada}>

Método HTTP: Get

Lógica de negocio: Permite ver una entrada con el id especificado.

Entrada: No aplica

Mensaje: String

Usuario:{{

IdUsuario: String

Nombre: String}}

Entradas: [{

Id : int

Nota: String,

Emociones: [String],

Actividad: String,

Fecha: Date}}

Validaciones: La entrada debe existir.