|  |  |
| --- | --- |
| POSIBLES RESPUESTAS | |
| 1: La acción se realizó exitosamente | |
| Código error | **Descripción** |
| -1 | Usuario o dato no encontrado |
| -2 | Error de conexión. |
| -3 | Dato no existe |
| -4 | id ya registrado |
| -5 | Error al guardar en la base de datos. |
| -6 | Error con los datos ingresados. |
| -7 | contraseña incorrecta |
| -8 | Usuario correcto, contraseña incorrecta. |
| -9 | Error de conexión al verificar. |
| -10 | Usuario o correo no existen |
| -11 | Email ya está registrado |
| -12 | Error de conexión con DB al actualizar |
| -13 | Error de conexión con DB al eliminar |
| -14 | Error al autoincrementar id notificaciones |
| -15 | Centro no existe |
|  |  |

USUARIOS

|  |  |  |  |
| --- | --- | --- | --- |
| METODO | DESCRIPCION | URL | EJEMPLO |
| GET | Obtener lista de todos los usuarios | http://127.0.0.1:3000/api/users | Fetch(‘http://127.0.0.1:3000/api/users’)  .then(data=>data.json())  .then(data=>console.log(data)) |
|
|
| Obtener un usuario mediante el id | http://127.0.0.1:3000/api/users/:id | Fetch(‘http://127.0.0.1:3000/api/users/1’)  .then(data=>data.json())  .then(data=>console.log(data)) |
|
|
| Obtener todas las notificaciones de un usuario en especifico | http://127.0.0.1:3000/api/ users/notifications/:id | Fetch(‘http://127.0.0.1:3000/api/ users/notifications/1)  .then(data=>data.json())  .then(data=>console.log(data)) |
|
|
| POST | Agregar un nuevo usuario | http://127.0.0.1:3000/api/users/add | datos={  Id\_User: 1234567890,  Nom\_User: "user",  Ape\_User: "apellido user",  Tel\_User: "telefono",  Ema\_User: "prueba@gmail.com",  Dir\_User: "567 Pine Ave",  Pass\_User: "1234567890",  Fot\_Per\_User: "https://example.com/user20.jpg",  Rol\_User\_Fk: "1"  }  fetch('http://127.0.0.1:3000/api/users/add’, {  method: "POST",  body: JSON.stringify(datos),  headers: { "Content-type": "application/json; charset=UTF-8" }  }).then(data=>data.json())  .then(data => console.log(data)) |
|
|
|
|
|
|
|
|
|
|
|
|
|
|
| Verificar inicio de sesión del usuario.  lo puede hacer con el email o con su usuario. | http://127.0.0.1:3000/api/users/login/verify | datos={  Ema\_User: "prueba@gmail.com",  Pass\_User: "1234567890"  }  fetch('http://127.0.0.1:3000/api/users/login/verify', {  method: "POST",  body: JSON.stringify(datos),  headers: { "Content-type": "application/json; charset=UTF-8" }  }).then(data=>data.json())  .then(data => console.log(data)) |
| PUT | Actualizar datos del usuario  -nota: el dato que se envía es el que será actualizado, no es necesario enviarlos todos. | http://127.0.0.1:3000/api/users/update/:id | datos={  Nom\_User: "user",  Ape\_User: "apellido user",  Tel\_User: "telefono",  Ema\_User: "prueba@gmail.com",  Dir\_User: "567 Pine Ave",  Pass\_User: "1234567890",  Fot\_Per\_User: "https://example.com/user20.jpg",  Rol\_User\_Fk: "1"  }  fetch('http://127.0.0.1:3000/api/users/update/1234567890, {  method: "PUT",  body: JSON.stringify(datos),  headers: { "Content-type": "application/json; charset=UTF-8" }  }).then(data=>data.json())  .then(data => console.log(data)) |
| DELETE | Eliminar usuarios | http://127.0.0.1:3000/api/users/delete/:id | Fetch(‘http://127.0.0.1:3000/api/users/delete/1234567890’,{  method:”DELETE”  })  .then(data=>data.json())  .then(data=>console.log(data)) |

ROLES

|  |  |  |  |
| --- | --- | --- | --- |
| METODO | DESCRIPCION | URL | EJEMPLO |
| GET | Obtener lista de todos los roles | http://127.0.0.1:3000/api/rol | Fetch(‘http://127.0.0.1:3000/api/rol’)  .then(data=>data.json())  .then(data=>console.log(data)) |
|
|
| Obtener usuarios que pertenecen a un rol | http://127.0.0.1:3000/api/rol/users/:id | Fetch(‘http://127.0.0.1:3000/api/rol/users/1234567890’)  .then(data=>data.json())  .then(data=>console.log(data)) |
|
|
|
|
| POST | Agregar un nuevo rol | http://127.0.0.1:3000/api/rol/add | datos={  Id\_Rol: 1,  Nom\_Rol: "user",  }  fetch('http://127.0.0.1:3000/api/rol/add), {  method: "POST",  body: JSON.stringify(datos),  headers: { "Content-type": "application/json; charset=UTF-8" }  }).then(data=>data.json())  .then(data => console.log(data)) |
|
|
|
|
|
|
|
|
|
|
|
|
|
|
| PUT | Actualizar datos de los roles  -nota: el dato que se envía es el que será actualizado, no es necesario enviarlos todos. | http://127.0.0.1:3000/api/rol/update/:id | datos={  Id\_Rol: 1,  Nom\_Rol: "user",  }  fetch('http://127.0.0.1:3000/api/rol/update/1234567890, {  method: "UPDATE",  body: JSON.stringify(datos),  headers: { "Content-type": "application/json; charset=UTF-8" }  }).then(data=>data.json())  .then(data => console.log(data)) |
| DELETE | Eliminar ro | http://127.0.0.1:3000/api/rol/delete/:id | Fetch(‘http://127.0.0.1:3000/api/users/delete/1234567890’,{  method: “DELETE”  })  .then(data=>data.json())  .then(data=>console.log(data)) |

NOTIFICACIONES

|  |  |  |  |
| --- | --- | --- | --- |
| METODO | DESCRIPCION | URL | EJEMPLO |
| GET | Obtener lista de todas las notificaciones | http://127.0.0.1:3000/api/notifications | Fetch(‘http://127.0.0.1:3000/api/notifications’)  .then(data=>data.json())  .then(data=>console.log(data)) |
|
|
|
|
|
|
| POST | Agregar un nueva notificación | http://127.0.0.1:3000/api/notifications/add | datos={  Tip\_Not:"info",  Cont\_Not:"Registro Exitoso bienvenido",  Id\_UserFK : 1  }  fetch('http://127.0.0.1:3000/api/rol/add), {  method: "POST",  body: JSON.stringify(datos),  headers: { "Content-type": "application/json; charset=UTF-8" }  }).then(data=>data.json())  .then(data => console.log(data)) |
|
|
|
|
|
|
|
|
|
|
|
|
|
|
| PUT | Actualizar datos de las notificaciones  -nota: el dato que se envía es el que será actualizado, no es necesario enviarlos todos. | http://127.0.0.1:3000/api/notifications/update/:id | datos={  Tip\_Not:"informacion",  Cont\_Not: "welcome to the cursiño",  visto: true  }  fetch('http://127.0.0.1:3000/api/notifications/update/1234567890, {  method: "UPDATE",  body: JSON.stringify(datos),  headers: { "Content-type": "application/json; charset=UTF-8" }  }).then(data=>data.json())  .then(data => console.log(data)) |
| DELETE | Eliminar notificaciones | http://127.0.0.1:3000/api/rol/delete/:id | Fetch(‘http://127.0.0.1:3000/api/users/delete/1234567890’,{  method: “DELETE”  })  .then(data=>data.json())  .then(data=>console.log(data)) |