

Guardar usuario - [saveUsersLogeds]

? Descripción

Registra y actualiza información de usuarios que inician sesión desde la aplicación móvil. Además, valida si la versión de la app instalada es compatible con la versión mínima definida en el servidor.

Funcionalidad:

1. **Verifica la versión de la app** contra `versiones.json`.
2. **Crea o actualiza** el registro de un usuario en la base de datos `mysql2.users`.
3. Devuelve mensaje de éxito o error según el proceso.

? Endpoint

POST `/save-users-logeds`

“ El nombre real depende del archivo de rutas de Laravel (web.php / api.php).

? Seguridad

- Requiere enviar parámetros en el body (no hay autenticación nativa).
- Se recomienda usarlo en rutas **API protegidas**.

? Request Body

```
{
  "user": "string",
  "version": "string"
}
```

Campos

| Campo | Tipo | Requerido | Descripción |
|---------|--------|-----------|--|
| user | string | Sí | Username que se registra o actualiza. |
| version | string | Sí | Versión de la app instalada en el dispositivo. |

? Responses

? Importante

En tu código actual, la primera línea retorna **inmediatamente**:

```
return [
  "valor"=>true,
  "msn"=> "Registro actualizado",
];
```

Por lo tanto **todo el servicio está anulado**.

Si esto es accidental, debes eliminar ese `return`.

Aun así, la documentación refleja lo que *debería hacer* según el resto del código.

?? Versión actualizada (200 OK)

Si la versión enviada es igual o mayor que la versión mínima del archivo `versiones.json`:

```
{
  "valor": true,
  "msn": "Registro Exitoso"
}
```

o

```
{
  "valor": true,
  "msn": "Registro actualizado"
}
```

? Versión desactualizada

```
{
  "valor": false,
  "msn": "Actualize su aplicacion en Play Store",
  "version_app": "1.0.1",
  "version_service": "1.0.5"
}
```

? Error al registrar o actualizar usuario

```
{
  "valor": false,
  "msn": "Registre un usuario"
}
```

o

```
{
  "valor": false,
  "msn": "Registro Actualizado"
}
```

? Lógica Interna (detallada)

1. Leer versión mínima del archivo:

`/public/version-app/versiones.json`

Campo utilizado: familia

2. Convertir versiones a enteros

Ejemplo:

"1.2.3" → 123

Esto permite compararlas.

3. Validación:

Si `intVersionApp < intVersionDefine` → debe actualizar

4. Registrar usuario:

- Si no existe → insertar username + primera sesión + última sesión
- Si existe → actualizar última sesión
- Base de datos usada: **mysql2**

Tablas y campos usados:

```
users.username  
users.fecha_primera_s  
users.fecha_ultima_s
```

? Esquema del JSON de version-app/versiones.json

Se utiliza solo el campo:

```
{  
  "familia": "1.0.5"  
}
```

? Schemas

Request

```
{  
  "user": "string",  
  "version": "string"  
}
```

Response

```
{  
  "valor": "boolean",  
  "msn": "string",  
  "version_app": "string(optional)",  
  "version_service": "string(optional)"  
}
```

? Ejemplo de uso (curl)

```
curl -X POST https://midominio.com/api/save-users-loggeds \  
-H "Content-Type: application/json" \  
-d '{"user":"jhon.doe","version":"1.0.2"}'
```

?? Notas importantes para documentación

? Problema detectado

La primera línea del método cancela toda la lógica:

```
return [  
  "valor"=>true,  
  "msn"=> "Registro actualizado",  
];
```

Si quieres que el servicio funcione como en la documentación, debes eliminar ese `return`.

Revisión #2

Creado 2025-11-24 16:40:49 -05 por Elian

Actualizado 2025-11-24 16:47:46 -05 por Elian