

Documentación protocolo REST - MensajerO

Recursos disponibles

- Login
- Conversacion
- Usuario

Formato de contenido: json

Recurso - Login

Acción: loguear usuario en el sistema

URI: server/login

Método: POST

Body: { "id": <str:usuario>,
 "password": <str:password>
 }

Restricciones sobre identificador de usuario: alfanumérico, máximo 12 caracteres.

Reply

En caso de usuario y contraseña válidos:

- **Response code:** 201 Created
- **Body:** { "token": <str:login_token> }

En caso de usuario o contraseña inválido:

- **Response code:** 401 Unauthorized request
-

Recurso - Usuario

Acción: solicitar lista de usuarios con su estado correspondiente

URI: server/usuarios

Método: GET

Params: r_user=<str:appuser_id>&token=<str:login_token>

Reply

En caso de r_user/token inválido:

- **Response code:** 401 Unauthorized request

En case de token valido:

- **Response code:** 200 OK
- **Body:** [{ "id": "pepe", "nombre": "Pepito", "estado": "conectado" }, { "id": "carlos", "nombre": "Carlos", "estado": "desconectado" }, ...]

Acción: solicitar perfil de usuario**URI:** server/usuario/<str:user_id>**Metodo:** GET**Params:** r_user=<str:appuser_id>&token=<str:login_token>**Reply**

En caso de r_user, user_id o token inválido:

- **Response code:** 401 Unauthorized request

En caso de token válido:

Usuario inválido:

- **Response code:** 401 Unauthorized request

Usuario válido:

- **Response code:** 200 OK
-

Acción: crear usuario**URI:** server/usuario/<str:user_id>**Metodo:** POST**Body:** { "password": <str:password>,
"nombre": <str:nombre>,
"foto": <str:foto>,
"ubicación": <str:ubicacion>
}*Formato `ubicacion`: string que identifica la ubicación. Ej: FIUBA, Casa, etc**Formato `foto`: base64-JPG**Los campos `password` y `nombre` son obligatorios.***Reply**

Usuario inválido o existente:

- **Response code:** 400 Bad request
- **Body:** { "error": <str:descripcion_error> }

Usuario válido y atributos inválidos:

- **Response code:** 400 Bad request
- **Body:** { "error": <str:descripcion_error> }

Usuario válido y atributos válidos:

- **Response code:** 201 Created
-

Acción: modificar usuario**URI:** server/usuario/<str:user_id>**Metodo:** PUT**Params:** r_user=<str:appuser_id>&token=<str:login_token>**Body:** { "nombre": <str:nombre>,
"password": <str:password>,

```
"foto": <str:foto>,  
"ubicación": <str:ubicacion>  
}
```

Nota: Todos los keys son opcionales. Los que no se incluyen retienen su valor original.

Reply

En caso de r_user/token inválido:

- **Response code:** 401 Unauthorized request

En caso de token válido:

Usuario inválido:

- **Response code:** 401 Unauthorized request

Usuario válido y atributos inválidos:

- **Response code:** 400 Bad request
- **Body:** { "error" : <str:descripcion_error> }

Usuario válido y atributos válidos:

- **Response code:** 201 Created

Recurso - Conversación

Acción: solicitar conversación anterior entre usuario <user1> y <user2>

URI: server/conversacion/<str:user1>.<str:user2>

Nota: el orden de los argumentos es irrelevante, ambas URI se resuelven por el mismo controlador.

Método: GET

Params: r_user=<str:appuser_id>&token=<str:login_token>

Reply

Nota: los mensajes se devuelven en orden ascendente según la fecha y hora de su recepción.

En caso de r_user/token inválido:

- **Response code:** 401 Unauthorized request

En caso de token válido:

Usuario(s) inválidos:

- **Response code:** 401 Unauthorized request

Usuario válido y atributos inválidos:

- **Response code:** 400 Bad request
- **Body:** { "error" : <str:descripcion_error> }

Usuario válido y atributos válidos:

- **Response code:** 200 OK
 - **Body:** { [<str:msg1>, <str:msg2>, ...] }
- Format de msg: <timestamp>.<usuario>.<mensaje>*
Format de <timestamp>: Unix time

Acción: enviar mensaje de usuario <appuser_id> a <dest_user>

URI: server/conversacion/<str:dest_user>

Metodo: POST

Params: r_user=<str:appuser_id>&token=<str:login_token>

Body: { "mensaje" : <str:mensaje> }

Reply

En caso de r_user/token inválido:

- **Response code:** 401 Unauthorized request

En caso de token válido:

Usuario(s) o mensaje inválido:

- **Response code:** 400 Bad request
- **Body:** { "error" : <str:descripcion_error> }

Usuario válido y mensaje válidos:

- **Response code:** 201 Created
-