

Energía
San Juan
Una empresa del Grupo Naturgy

Contenido

Introducción:	2
Llamado a los endpoints de la API	3
Módulo Auth:	3
• Token:	3
• Login:	4
Módulo Enviar	5
• Envío de email de confirmación	5
Módulo Perfil (usuario)	5
• Mostrar Perfil	5
• Actualiza Contraseña	7
• Inicia proceso de baja:	8

Introducción:

A continuación, se presenta el sistema de API para la web de Energía San Juan. El mismo cuenta con autenticación mediante Bearer Token, el cual otorga tiene una validación máxima de 24hs. El sistema se divide en módulos, los cuales se detallan a continuación.

Llamado a los endpoints de la API

Los endpoints deberán ser llamados de acuerdo al ambiente en el cuál se esté trabajando.

URL ambiente producción: https://www.energiasanjuan.com.ar/web_esj

URL ambiente desarrollo: https://www.energiasanjuan.com.ar/test/web_esj

Módulo Auth:

- **Token:**
 - **Endpoint:** /token

Este endpoint es para generar un token de acceso tipo Bearer el cual será requerido para el llamado a los endpoints siguientes, este deberá ser enviado mediante los headers del llamado.

Como parámetro para la creación del token se recibirá una Api Key y una Api Secret.

```
1 //Body - Genera Token
2
3 {
4   "api_key": "MiApiKey",
5   "api_secret": "MiApiSecret"
6 }
```

Especificaciones:

Método de llamada HTTP	POST
Cabeceras de solicitud	Cabecera: "Content-Type" : "application/json"
Parámetros	El método admite únicamente parámetros por POST en el body. api_key: Key definida por Energía San Juan y acordada con Widergy. api_secret: Secret Key definida por Energía San Juan y acordada con Widergy.
Cabeceras de respuesta	Status: Código de respuesta HTTP. Los posibles códigos retornados serán: HTTP_OK: 200 (cuando devuelve el llamado correctamente) HTTP_UNAUTHORIZED: 401 (cuando la api key o secret key no son correctas)
Respuesta	En caso de HTTP_OK: expiración: Tiempo de duración del token expresado en segundos (por defecto en 24hs) token: Cadena de texto representativa del Bearer Token. En caso de HTTP_UNAUTHORIZED:

	mensaje: mensaje de respuesta en caso de credenciales inválidas.
--	---

- **Login:**
 - **Endpoint:** /usuarios/login

Endpoint destinado al acceso al sistema por parte de los usuarios finales del mismo. Este recibirá como parámetros por header el Bearer Token y por parámetro de Body las credenciales de acceso (email y contraseña) del usuario, devolviendo como resultado dos posibles respuestas HTTP.

```

1  // Petición Login
2
3  {
4      "email": "ejemplo123@gmail.com",
5      "password": "123123",
6      "proveedor": null,
7      "uid": null
8  }

```

Especificaciones:

Método de llamada HTTP	POST
Cabeceras de solicitud	Cabecera: "Content-Type" : "application/json"
Parámetros	El método admite únicamente parámetros por POST en el body. email: Correo electrónico del usuario. password: Contraseña con la que se registró en la plataforma. proveedor: proveedor de autenticacion en caso de registro federado. uid: id externo en caso de autenticacion federada.
Cabeceras de respuesta	Status: Código de respuesta HTTP. Los posibles códigos retornados serán: HTTP_OK: 200 (cuando devuelve el llamado correctamente) HTTP_UNAUTHORIZED: 401 (cuando las credenciales ingresadas no son correctas)
Respuesta	En caso de HTTP_OK: usuario_id: id de usuario dentro del sistema. confirmado: (bool) indica si el usuario confirmó la direccion de correo electrónico. perfil_actualizado: (bool) indica si el usuario actualizó su contraseña luego del registro. En caso de HTTP_UNAUTHORIZED:

	mensaje: mensaje de respuesta en caso de credenciales inválidas.
--	---

Módulo Enviar

- **Envío de email de confirmación**
 - **Endpoint:** /emails/registro

Endpoint destinado a el envío de código de confirmación de usuario vía email. En este se recibe como parámetros de body el email, teléfono y código de verificación, los cuales son utilizados para el envío automático de email y SMS al usuario retornando así un mensaje de error en caso de falla o confirmación de envío en caso de éxito.

```

1 // ejemplo body envio email
2 {
3   "email": "ejemplo123@gmail.com",
4   "telefono": "2645991199",
5   "codigo_verificacion": "1291"
6 }
```

Especificaciones:

Método de llamada HTTP	POST
Cabeceras de solicitud	Cabecera: "Content-Type" : "application/json"
Parámetros	El método admite únicamente parámetros por POST en el body. email: Correo electrónico del usuario. telefono: N° de teléfono otorgado por el usuario. codigo_verificacion: código que será enviado via email y SMS.
Cabeceras de respuesta	Status: Código de respuesta HTTP. Los posibles códigos retornados serán: HTTP_OK: 200 (cuando devuelve el llamado correctamente) HTTP_NOT_FOUND: 404 (no se encontró el usuario con email o teléfono proporcionado) HTTP_UNPROCESSABLE_ENTITY: 422 (error al enviar el correo o sms)
Respuesta	En todos los casos HTTP se recibirá una respuesta con un único parámetro. mensaje: (string) indica el texto de retorno que verá el usuario según el HTTP status.

Módulo Perfil (usuario)

- **Mostrar Perfil**

- **Endpoint:** /usuarios/<string: usuario_id>

Endpoint destinado a mostrar la información de perfil del usuario registrado. Se recibe por URL el id de usuario y retorna los datos en caso de que el usuario exista, en caso de que no se retornará un mensaje informando el fallo.

Especificaciones:

Método de llamada HTTP	GET
Cabeceras de solicitud	Cabecera: "Content-Type" : "application/json" usuario_id: (string) Id del usuario con el que se encuentra en el sistema.
Parámetros	Este endpoint admite unicamente GET. Mediante Authorization recibira un Bearer Token.
Cabeceras de respuesta	Status: Código de respuesta HTTP. Los posibles códigos retornados serán: HTTP_OK: 200 (cuando devuelve el llamado correctamente) HTTP_NOT_FOUND: 404 (no se encontró el usuario con el id proporcionado)
Respuesta	En caso de HTTP_OK: email (string, <i>requerido</i>): El email con que el usuario se registró en la plataforma. uid (string, <i>opcional</i>): Representa un id externo en caso de que el usuario se haya registrado con un proveedor externo (federado). proveedor (string, <i>opcional</i>): Representa el proveedor de autenticación en caso de registración federada (google, twitter, etc.) nombre (string, <i>requerido</i>): El nombre con que el usuario se registró en la plataforma. apellido (string, <i>requerido</i>): El apellido con que el usuario se registró en la plataforma. alias (string, <i>opcional</i>): El alias elegido por el usuario para ser llamado en la plataforma. genero (string, <i>opcional</i>): El género con que el usuario se registró en la plataforma. tipo_documento (string, <i>requerido</i>): El tipo de documento que utilizó el usuario en la registración. numero_documento (string, <i>requerido</i>): El número de documento que utilizó el usuario en la registración. telefono: El número de teléfono que utilizó el usuario en la registración, telefono (string, <i>requerido</i>): telefono de contacto del usuario.

	<p>perfil_actualizado (<i>boolean, requerido</i>): indica si el usuario ha actualizado sus datos de perfil luego de la registración.</p> <p>confirmado (<i>boolean, requerido</i>): indica si el usuario recibió el código de confirmación y confirmó el email ingresado en el registro.</p> <p>En caso de HTTP_NOT_FOUND: mensaje: mensaje de respuesta en caso de id incorrecto.</p>
--	---

- **Actualiza Contraseña.**

- **Endpoint:** /usuarios/< usuario_id>/password

Endpoint destinado a la actualización de la contraseña de usuario. Este recibe por URL el id de usuario que requiere cambiar la contraseña y por body la contraseña actual y nueva para el mismo.

```

1 // ejemplo body actualizar contraseña
2 {
3   .... "password": "87654321",
4   .... "nueva_password": "123123"
5 }
6

```

Especificaciones:

Método de llamada HTTP	PUT
Cabeceras de solicitud	Cabecera: "Content-Type" : "application/json" usuario_id: (string) Id del usuario con el que se encuentra en el sistema.
Parámetros	Mediante Authorization recibira un Bearer Token. Mediante body recibe los paarametros: password: contraseña actual que el usuario posee en el sistema. nueva_password: Contraseña nueva proporcionada por el usuario.
Cabeceras de respuesta	Status: Código de respuesta HTTP. Los posibles códigos retornados serán: HTTP_OK: 200 (cuando devuelve el llamado correctamente) HTTP_UNPROCESSABLE_ENTITY: 422 (en caso de que la contraseña actual proporcionada sea incorrecta)
Respuesta	En caso de HTTP_OK:

	mensaje: mensaje de información en caso de éxito. En caso de HTTP_UNPROCESSABLE_ENTITY: mensaje: mensaje de respuesta en caso de contraseña incorrecta.
--	--

- **Inicia proceso de baja:**

- **Endpoint:** /usuarios/<usuario_id>/baja

Endpoint destinado a la inicialización del proceso de baja del usuario. Con el id del usuario se procede a deshabilitar el mismo del sistema.

Especificaciones:

Método de llamada HTTP	POST
Cabeceras de solicitud	Cabecera: "Content-Type" : "application/json" usuario_id: (string) Id del usuario con el que se encuentra en el sistema.
Parámetros	Mediante Authorization recibira un Bearer Token. Mediante body no se recibirán parametros.
Cabeceras de respuesta	Status: Código de respuesta HTTP. Los posibles códigos retornados serán: HTTP_OK: 200 (cuando devuelve el llamado correctamente) HTTP_UNPROCESSABLE_ENTITY: 422 (en caso de que exista algun error de validacion de datos de usuario) HTTP_NOT_FOUND: 404 (en caso de que el id de usuario proporcionado sea incorrecto y no se encuentre el usuario)
Respuesta	En caso de HTTP_OK: mensaje: mensaje de información en caso de éxito. En caso de HTTP_UNPROCESSABLE_ENTITY: error: mensaje de respuesta en caso de falla de validación o ejecución del proceso. En caso de HTTP_NOT_FOUND: error: mensaje de respuesta en caso de id de usuario inexistente.

- **Actualiza Perfil de usuario**

- **Endpoint:** /usuarios/<usuario_id>

Endpoint destinado a la actualización de los datos del perfil de usuario, el cual utilizará el método PUT de HTTP, recibiendo por todos los datos de perfil actualizándolos automáticamente en la base de datos del sistema.

```
1  // Ejemplo de body Actualiza Usuario
2
3  {
4    ...."email": "prueba123@gmail.com",
5    ...."uid": null,
6    ...."proveedor": null,
7    ...."nombre": "Juan",
8    ...."apellido": "Perez",
9    ...."alias": "juanpe",
10   ...."genero": "male",
11   ...."tipo_documento": "dni",
12   ...."numero_documento": "41830596",
13   ...."telefono": "11-41313",
14   ...."perfil_actualizado": true,
15   ...."confirmado": true
16 }

```

Especificaciones:

Método de llamada HTTP	PUT
Cabeceras de solicitud	Cabecera: "Content-Type" : "application/json" usuario_id : (string) Id del usuario con el que se encuentra en el sistema.
Parámetros	Mediante Authorization recibira un Bearer Token. Mediante body no se recibirán los siguientes parametros: email (<i>string, requerido</i>): El email con que el usuario se registró en la plataforma. uid (<i>string, opcional</i>): Representa un id externo en caso de que el usuario se haya registrado con un proveedor externo (federado). proveedor (<i>string, opcional</i>): Representa el proveedor de autenticación en caso de registración federada (google, twitter, etc.) nombre (<i>string, requerido</i>): El nombre con que el usuario se registró en la plataforma. apellido (<i>string, requerido</i>): El apellido con que el usuario se registró en la plataforma. alias (<i>string, opcional</i>): El alias elegido por el usuario para ser llamado en la plataforma. genero (<i>string, opcional</i>): El género con que el usuario se registró en la plataforma. tipo_documento (<i>string, requerido</i>): El tipo de documento que utilizó el usuario en la registración.

	<p>numero_documento (<i>string, requerido</i>): El número de documento que utilizó el usuario en la registración.</p> <p>telefono: El número de teléfono que utilizó el usuario en la registración,</p> <p>telefono (<i>string, requerido</i>): telefono de contacto del usuario.</p> <p>perfil_actualizado (<i>boolean, requerido</i>): indica si el usuario ha actualizado sus datos de perfil luego de la registración.</p> <p>confirmado (<i>boolean, requerido</i>): indica si el usuario recibió el código de confirmación y confirmó el email ingresado en el registro.</p>
Cabeceras de respuesta	<p>Status: Código de respuesta HTTP. Los posibles códigos retornados serán:</p> <p>HTTP_OK: 200 (cuando devuelve el llamado correctamente)</p> <p>HTTP_UNPROCESSABLE_ENTITY: 422 (en caso de que exista algun usuario registrado con el tipo y número de documento ingresado)</p> <p>HTTP_NOT_FOUND: 404 (en caso de que no se encuentre el usuario)</p>
Respuesta	<p>En caso de HTTP_OK: mensaje: mensaje de información en caso de éxito.</p> <p>En caso de HTTP_UNPROCESSABLE_ENTITY: mensaje: mensaje de respuesta en caso de que los datos ingresados ya esten registrados.</p> <p>En caso de HTTP_NOT_FOUND: mensaje: mensaje de respuesta en caso de usuario inexistente.</p>