

- ○

Descripción de la configuración de Postman:

En la configuración de Postman para esta API, se utilizan **variables** y un **pre-request script** para manejar el proceso de autenticación, así como los entornos de desarrollo y producción. Aquí están los principales detalles de la configuración:

Variables de entorno:

1. **dev**: <https://contisign-api.dev.stackcode.io/> (ambiente de pruebas)
2. **prod**: <https://api.contisign.com.mx/> (ambiente de producción)
3. **token**: Se actualiza dinámicamente a través del pre-request script para manejar el token de autenticación JWT.
4. **token2**: Puede ser un valor ingresado manualmente o uno almacenado para pruebas anteriores, pero no se actualiza dinámicamente.

Pre-request script:

Antes de ejecutar cualquier petición, se realiza un script para autenticarse con las credenciales del usuario y obtener un token de autenticación. Este token es almacenado en la variable token y es utilizado en las peticiones subsiguientes.

```
const auth_request = {
  url: pm.variables.get("dev")+'/api/auth/signin',
  method: 'POST',
  header: {
    'Content-Type': 'application/json'
  },
  body: {
    mode: 'raw',
    raw: JSON.stringify({
      "email": "mcruz@sistemascontino.com.mx",
      "password": "S1stemas2022"
    })
  }
};

pm.sendRequest(auth_request, function(err, response) {
  if (response.code === 200) {
    var responsejson = response.json();
    pm.collectionVariables.set("token", responsejson.token);
  } else {
    console.error("Failed to obtain access_token", response.json());
  }
});
```

Estructura de los Endpoints en Postman (Carpeta: Auth)

Sign in = {{test}}/api/auth/signin

- **Funcionalidad**: Este endpoint permite a los usuarios iniciar sesión en el sistema proporcionando su correo electrónico y contraseña. Al iniciar sesión correctamente, el servidor responde con un token de autenticación (JWT) que se utiliza para validar futuras solicitudes del usuario. Este token se almacena y utiliza en todas las solicitudes protegidas por el sistema.
- **Supuestos**: El usuario ya debe estar registrado en el sistema y proporcionar las credenciales correctas (email y contraseña). El sistema generará un token JWT para el usuario autenticado.
- **URL**: {{test}}/api/auth/signin
- **Params**: Ninguno
- **Headers**:
 - Content-Type: application/json

- **Body:**

```
{
  "email": "mcruz@sistemascontino.com.mx",
  "password": "S1stemas2022"
}
```

- **Respuesta Exitosa (200):**

```
{
  "displayName": "Daniel Leyva Caballero",
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

- **Error (401):**

```
{
  "Message": "Credenciales incorrectas"
}
```

Sign Up = `{{test}}/api/auth/signup`

- **Funcionalidad:** Permite el registro de un nuevo usuario en el sistema. El usuario proporciona detalles como nombre, correo electrónico, contraseña y otros datos personales. Si el registro es exitoso, el usuario queda habilitado en el sistema para iniciar sesión y utilizar las funcionalidades correspondientes a su rol.
- **Supuestos:** El correo proporcionado debe ser único, es decir, no debe estar registrado previamente en el sistema. El sistema asignará el rol por defecto, a menos que se especifique uno.
- **URL:** `{{test}}/api/auth/signup`
- **Params:** Ninguno
- **Headers:**
 - Content-Type: application/json
- **Body:**

```
{
  "firstName": "Daniel",
  "lastName": "Leyva",
  "email": "dancaballerodlc@gmail.com",
  "password": "S1stemas2022",
  "phoneNumber": "2291847482",
  "Active": true,
  "role": "user"
}
```

- **Respuesta Exitosa (200):**

```
{
  "displayName": "Daniel Leyva",
  "id": "836c4d58-cac7-4ac2-8fc4-f34a24f8c92a"
}
```

- **Error (409):**

```
{
  "Message": "Ya existe un usuario con este correo"
}
```

Activate User = `{{test}}/api/auth/activate/:pUserId`

- **Funcionalidad:** Este endpoint se utiliza para activar la cuenta de un usuario que se ha registrado, pero que aún no ha activado su cuenta. El `pUserId` se envía al correo electrónico del usuario como parte de un enlace, y el usuario debe seguir este enlace para completar el proceso de activación de su cuenta.
- **Supuestos:** El `pUserId` debe ser válido y corresponder a un usuario registrado pero no

activado. Este pUserId es generado en el frontend y enviado al correo del usuario.

- **URL:** `{{test}}/api/auth/activate/pUserId`
- **Params:**
 - pUserId: S5yb2JlcnRvQGdtYWlsLmNvbSIsIml
- **Headers:**
 - Authorization: `{{token}}`
- **Body:** Ninguno
- **Respuesta Exitosa (200):**

```
{
  "Message": "Usuario activado correctamente"
}
```

- **Error (404):**

```
{
  "Message": "El usuario no existe, favor de verificar"
}
```

Update password (mail) = `{{test}}/api/auth/resetpassword`

- **Funcionalidad:** Envía un correo electrónico de restablecimiento de contraseña a un usuario que ha solicitado cambiar su contraseña. El correo contiene un enlace con un token de restablecimiento que permitirá al usuario cambiar su contraseña.
- **Supuestos:** El correo debe estar registrado en el sistema para que se envíe el correo de restablecimiento. Si el correo no está registrado, el sistema devolverá un error.
- **URL:** `{{test}}/api/auth/resetpassword`
- **Params:** Ninguno
- **Headers:**
 - Authorization: `{{token}}`
- **Body:**

```
{
  "email": "aayala@sistemascontino.com.mx"
}
```

- **Respuesta Exitosa (200):**

```
{
  "Message": "Correo de restablecimiento enviado"
}
```

- **Error (404):**

```
{
  "Message": "El usuario no existe, favor de verificar"
}
```

Update password = `{{test}}/api/auth/change-password/:resetToken`

- **Funcionalidad:** Este endpoint permite a un usuario cambiar su contraseña utilizando un resetToken que ha sido enviado previamente al correo electrónico del usuario como parte del flujo de restablecimiento de contraseña. La nueva contraseña se establece una vez que se valida el token.
- **Supuestos:** El resetToken debe ser válido y estar asociado con una solicitud de restablecimiento de contraseña. Si el token es inválido o ha expirado, se devolverá un error.
- **URL:** `{{test}}/api/auth/change-password/:resetToken`
- **Params:**
 - resetToken: XN0ZW1hc2NvbhRpbm8uY29tLm14liw
- **Headers:** Ninguno
- **Body:**

```
{
  "password": "BzGkZ35jcV0tu9uMs",
  "newpassword": "S1stemas2022",
  "repeat": "S1stemas2022"
}
```

- **Respuesta Exitosa (200):**

```
{
  "Message": "Contraseña cambiada con éxito"
}
```

- **Error (400):**

```
{
  "Message": "El token de restablecimiento es inválido o ha expirado"
}
```

Save admin = `{{test}}/api/auth/addnewuser`

- **Funcionalidad:** Crea un nuevo usuario con permisos elevados. Este endpoint se utiliza para agregar usuarios con permisos administrativos o con acceso a funciones más avanzadas del sistema.
- **Supuestos:** El correo proporcionado debe ser único y no estar registrado previamente. Dependiendo del rol que se asigne, el usuario puede obtener permisos elevados o administrativos.
- **URL:** `{{test}}/api/auth/addnewuser`
- **Params:** Ninguno
- **Headers:** Ninguno
- **Body:**

```
{
  "phoneNumber": "2297721893",
  "firstName": "Web",
  "lastName": "Developer",
  "email": "webdeveloper@gmail.com",
  "password": "Stackcode$2021"
}
```

- **Respuesta Exitosa (200):**

```
{
  "displayName": "Web Developer",
  "id": "3b0e695b-1b5c-4e1b-8a15-c218d424ebbd"
}
```

Change user password = `{{test}}/api/changepass/:userId`

- **Funcionalidad:** Este endpoint permite a un usuario autenticado cambiar su contraseña actual por una nueva. El usuario debe proporcionar la contraseña actual junto con la nueva contraseña y su confirmación.
- **Supuestos:** El usuario debe conocer su contraseña actual para poder cambiarla. Si la contraseña actual es incorrecta, el sistema devolverá un error.
- **URL:** `{{test}}/api/changepass/:userId`
- **Params:**
 - `userId`: 836c4d58-cac7-4ac2-8fc4-f34a24f8c92a
- **Headers:**
 - `Authorization`: `{{token}}`
- **Body:**

```
{
  "password": "BzGkZ35jcV0tu9uMs",
  "newpassword": "S1stemas2022",
  "repeat": "S1stemas2022"
}
```

```
}
```

- **Respuesta Exitosa (200):**

```
{  
  "Message": "Contraseña actualizada con éxito"  
}
```

- **Error (401):**

```
{  
  "Message": "Contraseña actual incorrecta"  
}
```

Complete signup user = `{{test}}/api/auth/presignup/:userId`

- **Funcionalidad:** Este endpoint se utiliza para completar el registro de un usuario que ha comenzado el proceso

Complete signup user = `{{test}}/api/auth/presignup/:userId`

- **Funcionalidad:** Este endpoint se utiliza para completar el registro de un usuario que ha iniciado el proceso de registro pero que requiere información adicional o finalización. Este flujo puede ser útil para casos donde se requiere que el usuario proporcione más detalles antes de habilitar completamente su cuenta.
- **Supuestos:** El `userId` debe estar asociado a un usuario que ya ha iniciado su proceso de registro. El sistema solicita completar detalles adicionales para finalizar el registro.
- **URL:** `{{test}}/api/auth/presignup/:userId`
- **Params:**
 - `userId`: 836c4d58-cac7-4ac2-8fc4-f34a24f8c92a

- **Headers:** Ninguno

- **Body:**

```
{  
  "additionalInfo": "datos adicionales"  
}
```

- **Respuesta Exitosa (200):**

```
{  
  "Message": "Registro completado"  
}
```

Refresh token = `{{test}}/api/auth/refreshToken`

- **Funcionalidad:** Este endpoint permite refrescar el token de autenticación (JWT) de un usuario cuya sesión está a punto de expirar. Esto evita que el usuario tenga que iniciar sesión nuevamente, extendiendo su sesión activa por un nuevo periodo de tiempo.
- **Supuestos:** El token actual debe estar en los headers de la petición y debe ser válido. Este token no debe haber expirado completamente, ya que de lo contrario se requerirá un nuevo inicio de sesión.
- **URL:** `{{test}}/api/auth/refreshToken`
- **Params:** Ninguno
- **Headers:**
 - `Authorization`: `{{token}}`
- **Body:** Ninguno
- **Respuesta Exitosa (200):**

```
{  
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."  
}
```

Estructura de los Endpoints en Postman (Carpeta: Catalogs)

Save Catalog item = `{{test}}/api/catalogs/save`

- **Funcionalidad:** Este endpoint se utiliza para agregar o actualizar elementos en los catálogos de la aplicación ContiSign. Los catálogos permiten al sistema almacenar conjuntos de datos estáticos o reutilizables, como categorías de productos, tipos de consumibles, o configuraciones predefinidas. En este caso, parece que el catálogo se refiere a consumibles. El cuerpo de la solicitud incluye el nombre del catálogo al que se desea agregar un nuevo ítem y el valor correspondiente.
- **Supuestos:** El catálogo especificado debe existir en el sistema, y el ítem que se desea agregar debe cumplir con los requisitos específicos del catálogo. Es posible que algunos catálogos requieran más campos que otros (por ejemplo, un Alias o Key).
- **Tipo:** POST
- **URL:** `{{test}}/api/catalogs/save`
- **Params:** Ninguno
- **Headers:**
 - Authorization: `{{token}}`
- **Body:**

```
{
  "CatalogName": "Consumables",
  "ValueAlpha": "Número de plantillas"
}
```
- **Respuesta Exitosa (200):**

```
{
  "Message": "Elemento del catálogo guardado correctamente"
}
```
- **Error (500):**

```
{
  "status": 500,
  "message": "An unexpected error occurred",
  "tag": "unexpected-error",
  "stack": "Error: WHERE parameter \"Alias\" has invalid \"undefined\" value..."
}
```

Posibles causas del error:

1. **Parámetros faltantes:** El error puede deberse a la falta de algún campo obligatorio, como un Alias o una Key que el sistema requiere para identificar correctamente el ítem dentro del catálogo.
2. **Problemas internos en el servidor:** La lógica en el backend podría estar mal configurada o no manejar correctamente los casos en los que falta un valor importante, lo que genera un error de SQL.

Sugerencias:

- **Revisar el catálogo “Consumables”:** Verifica si se requiere algún campo adicional en la solicitud que no estás proporcionando, como un alias, identificador, o algún otro parámetro relevante.
- **Consultar documentación adicional:** Si tienes acceso a más detalles sobre los catálogos, podrías revisar cómo interactuar correctamente con este tipo de endpoint.

Save Positions = `{{test}}/api/positions`

- **Funcionalidad:** Este endpoint permite registrar una posición o puesto de trabajo asociado a una entidad específica (identificada por `businessId`). Si el `businessId` se proporciona, la posición se asigna a esa entidad en particular. Si no se proporciona, la posición puede guardarse de todas formas, pero no estará vinculada a una empresa

específica.

- **Supuestos:**

- El `businessId` puede ser opcional, dado que el sistema permite registrar una posición sin vincularla a una entidad.
- El campo `position` parece ser el nombre del puesto que se desea registrar.
- La respuesta incluye un identificador único (`id`) para la posición registrada.

- **Tipo:** POST

- **URL:** `{{test}}/api/positions`

- **Params:** Ninguno

- **Headers:**

- Authorization: `{{token}}`

- **Body:**

```
{
  "businessId": "db2a6259-c21a-467b-affb-a6b87c5822d1",
  "position": "Recursos Humanos"
}
```

- **Respuesta Exitosa (200):**

```
{
  "Active": true,
  "id": 880,
  "Alias": "Recursos Humanos",
  "CatalogName": "Positions",
  "ValueAlpha": "db2a6259-c21a-467b-affb-a6b87c5822d1",
  "updated_at": "2024-10-10T17:21:20.804Z",
  "created_at": "2024-10-10T17:21:20.804Z",
  "ValueNumeric": null,
  "BusinessId": null,
  "CreatedBy": null
}
```

- **Comportamiento cuando `businessId` está vacío:**

- **Cuerpo:**

```
{
  "businessId": "",
  "position": "Recursos Humanos"
}
```

- **Respuesta Exitosa (200):**

```
{
  "Active": true,
  "id": 883,
  "Alias": "Recursos Humanos",
  "CatalogName": "Positions",
  "ValueAlpha": "",
  "updated_at": "2024-10-10T17:22:35.554Z",
  "created_at": "2024-10-10T17:22:35.554Z",
  "ValueNumeric": null,
  "BusinessId": null,
  "CreatedBy": null
}
```

Observaciones:

1. **El businessId es opcional:** El hecho de que puedas guardar la posición sin un businessId sugiere que el campo es opcional o que el sistema permite la creación de posiciones generales sin estar vinculadas a una entidad específica.
 2. **Posible validación faltante:** Si la intención es que las posiciones siempre estén vinculadas a una entidad (a través del businessId), el sistema debería validar que este campo no esté vacío.
-

Get position by id = `{{test}}/api/positions/:positionId`

- **Funcionalidad:** Este endpoint permite obtener los detalles de una posición específica dentro del catálogo utilizando su positionId. Los detalles incluyen el nombre del catálogo al que pertenece la posición, el nombre o valor de la posición (como "Ventas" en este caso), si la posición está activa, y su relación con un BusinessId (probablemente una entidad o empresa).
- **Supuestos:**
 - El positionId debe ser un valor numérico que corresponda a una posición válida en el catálogo.
 - Si el positionId no existe en el sistema, el endpoint devolverá null, lo que indica que la posición no fue encontrada.
- **Tipo:** GET
- **URL:** `{{test}}/api/positions/:positionId`
- **Params:**
 - positionId: Valor numérico que identifica una posición dentro del catálogo.
- **Headers:**
 - Authorization: `{{token}}`
- **Body:** Ninguno
- **Respuesta cuando el positionId no existe (por ejemplo, valor 0):**

null

- **Respuesta cuando el positionId existe (por ejemplo, valor 1):**

```
{
  "id": 1,
  "CatalogName": "Departments",
  "ValueAlpha": "Ventas",
  "Alias": "Ventas",
  "Active": true,
  "BusinessId": "f9259884-3919-41b5-b19e-a72d7e1cc691",
  "CreatedById": null
}
```

Update position by id = `{{test}}/api/positions/:positionId`

- **Funcionalidad:** Este endpoint permite actualizar la información de una posición específica en el catálogo, identificado por el positionId. Los detalles que se pueden actualizar incluyen el nombre del catálogo (CatalogName), el valor relacionado con la posición (ValueAlpha), el alias de la posición, y si está activa o no. Esto es útil cuando se necesita actualizar los datos de un puesto o rol dentro de la empresa.
- **Supuestos:**
 - El positionId debe existir en el sistema para poder realizar la actualización.
 - El cuerpo de la solicitud debe incluir los campos que se desean modificar, como el alias de la posición y su estado (Active).
- **Tipo:** PUT
- **URL:** `{{test}}/api/positions/:positionId`
- **Params:**
 - positionId: Un valor numérico que identifica una posición específica en el catálogo.
- **Headers:**
 - Authorization: `{{token}}`
- **Body:**

```
{
  "CatalogName": "Positions",
  "ValueAlpha": "db2a6259-c21a-467b-affb-a6b87c5822d1",
  "ValueNumeric": null,
  "Alias": "Asesor",
  "Active": true
}
```

- **Respuesta Exitosa (200):**

```
{
  "id": 5,
  "CatalogName": "Positions",
  "ValueAlpha": "db2a6259-c21a-467b-affb-a6b87c5822d1",
  "ValueNumeric": null,
  "Alias": "Asesor",
  "Active": true,
  "created_at": "2022-07-04T15:21:08.993Z",
  "updated_at": "2024-10-10T17:41:23.595Z",
  "BusinessId": "f9259884-3919-41b5-b19e-a72d7e1cc691",
  "CreatedById": null
}
```
