¿Qué es el API de Gestión?

La definición de API de Gestión puede variar dependiendo del software o servicio que estés utilizando. En el contexto de la gestión de identidad y acceso (IAM), el API de Gestión generalmente se refiere a un conjunto de APIs que te permiten gestionar programáticamente los recursos relacionados con IAM. Por ejemplo, usuarios, aplicaciones, roles, permisos, organizaciones, etc.

Aunque el nombre no especifica la implementación exacta, el API de Gestión suele ser RESTful, dada su naturaleza de definir precisamente los recursos y las operaciones que se pueden realizar sobre ellos. Dicho esto, cuando ves `POST /users`, puedes esperar que esta llamada API creará un nuevo usuario.

¿Por qué es importante el API de Gestión?

El API de Gestión crea una capa separada de abstracción sobre el sistema IAM, pero debajo de la interfaz de usuario. Esto permite a los desarrolladores automatizar la gestión de recursos IAM, lo cual puede ser especialmente útil en varios escenarios:

1. Automatización

Como sugiere el nombre, el API de Gestión te permite usar código para gestionar recursos, en lugar de hacer clic manualmente a través de la interfaz de usuario. Esto es particularmente útil cuando tienes un gran número de usuarios, aplicaciones o roles que gestionar. Por ejemplo, puedes escribir un script para importar usuarios de un archivo CSV y asignarles los roles y permisos correctos.

2. Integración

El API de Gestión crea una manera estándar para la comunicación de servicio a servicio (o máquina a máquina). Cuando tienes múltiples servicios que necesitan comunicarse con el sistema IAM, en lugar de implementar integraciones personalizadas para cada servicio, un API de Gestión bien diseñado puede ser utilizado para todos los servicios combinando las llamadas API. Por ejemplo, un servicio que necesita listar todos los usuarios bajo un rol específico puede hacerlo llamando `GET /roles/{role_id}/users`.

3. Composición y extensión de características

Debido a la variedad de requisitos empresariales, un sistema IAM puede no ser capaz de proporcionar todas las características exactas que necesitas, especialmente cuando se trata de requisitos complejos de control de acceso (access control). El API de Gestión te permite construir características personalizadas sobre el sistema IAM existente sin modificar la plataforma o arquitectura subyacente.

Veamos un ejemplo cotidiano: los usuarios finales necesitan cambiar su dirección de correo electrónico. Diferentes aplicaciones pueden tener diferentes requisitos:

1. La App A requiere que el usuario verifique tanto la dirección de correo antigua como la nueva.
2. La App B requiere que el usuario verifique la contraseña existente antes de cambiar la dirección de correo.
3. La App C requiere que el usuario verifique la contraseña existente y que un administrador apruebe el cambio de correo.

Con el API de Gestión, puedes construir un servicio personalizado que orqueste estos requisitos llamando a las APIs necesarias en el orden correcto. Incluso puedes combinar el API de Gestión con el API de tu servicio para lograr un flujo de trabajo complejo. Tomemos como ejemplo la App C:

1. El usuario hace clic en “Cambiar Correo” en la App C, lo que envía una solicitud `POST /email-change-requests` a tu servicio. Este crea una nueva solicitud de cambio de correo y devuelve el identificador `foo`.
2. La App C muestra un cuadro de diálogo al usuario, pidiéndole que ingrese la contraseña existente.
3. El usuario ingresa la contraseña, y la App C envía una solicitud `PATCH /email-change-requests/foo` a tu servicio con la contraseña. En segundo plano, tu servicio verifica la contraseña llamando al API de Gestión `POST /users/{user_id}/verify-password`.
4. Si la contraseña es correcta, tu servicio crea un registro de verificación exitosa en la solicitud de cambio de correo `foo`.
5. En el panel de administración, un administrador puede ver las solicitudes de cambio de correo pendientes con `GET /email-change-requests?status=pending`.
6. Si el administrador aprueba la solicitud, el panel de administración envía una solicitud `PATCH /email-change-requests/foo` a tu servicio con la aprobación del administrador.
7. Tu servicio entonces llamará al API de Gestión `PATCH /users/{user_id}` para actualizar la dirección de correo del usuario. Si la dirección de correo no puede ser actualizada, el API de Gestión devolverá un error, y tu servicio podrá manejarlo en consecuencia.

Nota que en el ejemplo anterior, nuestros usuarios finales nunca interactúan directamente con el API de Gestión. En cambio, interactúan con la App C, que orquesta las llamadas al API de Gestión

en segundo plano para lograr el flujo de trabajo deseado.

¿Cómo debería ser un buen API de Gestión?

- **RESTful**: Sigue los principios RESTful para hacer el API predecible y fácil de usar.
- **Orientado a recursos**: Representa recursos como sustantivos y utiliza métodos HTTP para realizar acciones sobre ellos.
- **Consistente**: Usa convenciones de nomenclatura, manejo de errores y formatos de respuesta coherentes.
- **Seguro**: Implementa mecanismos adecuados de autenticación (authentication) y autorización para proteger el API.
- **Documentado**: Proporciona documentación clara y concisa sobre cómo usar el API, incluyendo ejemplos y casos de uso.
- **Compatible**: Asegura la compatibilidad retroactiva al introducir nuevas versiones del API.
- **Completo**: Cubre todas las operaciones necesarias para gestionar eficazmente los recursos IAM.

Existen otros aspectos como el rendimiento y la escalabilidad, que están más relacionados con la infraestructura que con el diseño del API en sí. Sin embargo, en la práctica, un buen API de Gestión debería considerar también estos aspectos.

Revisión #2
Creado 2024-11-11 17:58:11 UTC por Marluan Espiritusanto
Actualizado 2024-11-11 17:59:56 UTC por Marluan Espiritusanto