



Errores Giltz@ Status de las respuestas HTTP

Izenpe v.1.0

Marzo 2018

El código de estado (*status*) de la respuesta HTTP indica el resultado de la operación, según la convención descrita en la [RFC 7231](#). En caso de error, el formato de la respuesta puede variar según el tipo de error.

1. Códigos de estado de operaciones completadas correctamente.
2. Códigos de estado con parámetro error en la cabecera HTTP WWW-Authenticate.
3. Códigos de estado con descripción opcional del error en el cuerpo de la respuesta.

1. Códigos de estado de operaciones completadas correctamente

Cuando una operación se puede completar de forma satisfactoria, es posible que el cuerpo de la respuesta contenga más información sobre el resultado de la operación.

1.1. HTTP/1.1 200 OK

La operación se ha completado correctamente. El cuerpo de la respuesta puede incluir más información sobre el resultado de la operación.

1.2. HTTP/1.1 201 Created

La operación de creación de un nuevo recurso se ha completado correctamente. El cuerpo de la respuesta incluye información sobre el recurso creado o la cabecera Location contiene la URL de acceso a éste.

1.3. HTTP/1.1 204 No content

La operación se ha completado correctamente. No se obtiene ninguna entidad como respuesta y el cuerpo está vacío.

2. Códigos de estado con parámetro error en la cabecera HTTP WWW-Authenticate

Cuando se produce un error de autorización, se obtiene un código de error en el parámetro error de la cabecera HTTP WWW-Authenticate de la respuesta, como se describe en [RFC 6750](#). Por ejemplo:

```
HTTP/1.1 401 Unauthorized  
WWW-Authenticate: Bearer error="invalid_token"
```

A continuación se listan los códigos de estado que pueden ir acompañados de una cabecera WWW-Authenticate, y los posibles valores del parámetro error en cada caso.

2.1. HTTP/1.1 400 Bad Request

invalid_request: No es posible procesar la petición debido a una sintaxis incorrecta de la cabecera HTTP Authorization, posiblemente porque está mal formada o hay más de una.

2.2. HTTP/1.1 401 Unauthorized

Sin código de error: Las condiciones de autenticación requeridas no han sido satisfechas porque la petición no incluye una cabecera HTTP `Authorization`, o bien la incluye pero con un esquema de autenticación HTTP no reconocido.

`invalid_token`: El *token* de autorización no es válido o está caducado. En el caso de los *tokens* OAuth, el *token* caduca al cabo del número de segundos indicado en el resultado de la operación obtener un token de acceso (parámetro `expires_in`). Por otro lado, el servidor de autorización considera válidos los *tokens* JWT cuyo instante de emisión se encuentra en el rango de tiempo de 10 minutos inmediatamente anteriores o posteriores a la hora del servidor.

2.3. HTTP/1.1 403 Forbidden

`insufficient_scope`: Los *scopes* a los que está asociado el *token* de acceso OAuth no son suficientes para acceder al recurso protegido. La cabecera `WWW-Authenticate` incluye un parámetro adicional `scope` con el *scope* esperado. Un *token* OAuth está asociado a los *scopes* que se hayan indicado en la operación obtener autorización u obtener un token de acceso, según el tipo de *grant* OAuth que se haya utilizado para obtener el token.

3. Códigos de estado con descripción opcional del error en el cuerpo de la respuesta

Para cualquier otro tipo de error, es habitual que el cuerpo de la respuesta contenga una estructura de datos JSON con los siguientes parámetros:

- `error`: Código de error.
- `error_description`: Opcional. Descripción adicional del error.
- `error_details`: Opcional. Detalles adicionales del error. El formato es un objeto JSON con subcampos.

Por ejemplo:

```
HTTP/1.1 400 Bad Request
{
  "error": "NotSupportedException",
  "error_description": "Cannot consume content type",
  "error_details": { }
}
```

A continuación se describen posibles valores del parámetro `error` para varios códigos de estado.

3.1. HTTP/1.1 400 Bad Request

La sintaxis de la petición no es correcta o no se cumplen las condiciones requeridas para completar la operación.

- `AccountAlreadyExistsException`: El valor del atributo único indicado en la petición y que se desea usar como identificador explícito de la cuenta de usuario ya está asignado a otra cuenta.
- `AccountAttributeNotDefinedException`: Alguno de los atributos indicados en la petición no está definido en el dominio de identidad o no está mapeado a la cuenta de usuario.
- `AmbiguousSignIdentityTypeException` y `MultipleIdentitiesSelectedException`: Las etiquetas solicitadas al proveedor de firma para crear una firma digital identifican a más de una identidad de firma disponible.
- `BadRequestException`: La petición no es correcta.
- `BlockedDeviceException`: El dispositivo de firma para el que se solicita la operación tiene un proceso pendiente de completarse.
- `DeviceAlreadyExistsException`: El dispositivo de firma que se solicita registrar ya existe.
- `DeviceTypesNotExistsException`: El tipo de dispositivo de firma indicada en la creación de la identidad de firma no se corresponde con ninguno registrado en la configuración.
- `InconsistentLinkedAccountIdException`: Se ha intentado actualizar la información de una cuenta vinculada indicando un identificador de dominio externo (campo `domain`) o un identificador de la identidad externa (campo `account_id`) que no coinciden con los que ya estaban asociados a la identidad vinculada. Una vez creada la identidad vinculada, estas propiedades no pueden modificarse.
- `IncorrectSignIdentitySelectorsException`: El selector de identidades de firma solicitado es incorrecto o no existe ninguna identidad asociada.
- `InvalidAttributeValueException`: Alguno de los atributos indicados en la petición no tiene valor o tiene un formato no soportado.
- `InvalidExternalDomainNameException`: El nombre del dominio externo indicado en el cuerpo de la petición no es válido o el dominio externo no está federado al dominio al que pertenece la cuenta de usuario.
- `InvalidCsrTemplateException`: La plantilla para generar la petición de certificación indicada en la creación de la identidad de firma solicitada es incorrecta.
- `InvalidKeyTemplateException`: La plantilla para generar las claves indicada en la creación de la identidad de firma solicitada es incorrecta.
- `InvalidParametersException`: El valor de uno o más parámetros indicados en la petición no se soporta.
- `InvalidPkcs12Exception`: El PKCS#12 y la contraseña recibidos para crear la identidad son incorrectos o no se pueden emplear.
- `InvalidQueryFilterException`: El parámetro `filter` indicado en la petición está mal formado, incluye un atributo que no está definido en el dominio de identidad o no está mapeado a las cuentas de usuario, o el atributo no está definido como único.
- `InvalidRequestException`: La sintaxis de la petición es incorrecta o el valor de algunos parámetros no están soportados.
- `JsonParseException` y `JsonProcessingException`: El cuerpo de la petición está mal formado.
- `LinkedAccountAlreadyExistsException`: La cuenta de usuario ya tiene una identidad vinculada con el dominio externo (campo `domain`) y el identificador de la identidad externa (campo `account_id`) especificados.

- `MandatorySignatureException`: El proceso de firma no tiene definida una definición de firma.
- `ModelValidationException`: La petición no es correcta porque falta algún parámetro obligatorio en la URL o en el cuerpo, o porque el contenido del cuerpo está vacío o no es completo.
- `MissingExternalAccountIdException`: El cuerpo de la petición no incluye el identificador de la identidad externa en el dominio de identidad federado (campo `account_id`).
- `MissingUserIdAttributeException`: El cuerpo de la petición no incluye el valor del atributo que se usa para determinar el identificador explícito de la cuenta de usuario.
- `MultipleAccountsForUniqueAttributeException`: El valor de un atributo único indicado en la petición ya está asignado a otra cuenta de usuario.
- `MultipleValuesForUniqueAttributeException`: Un atributo único indicado en la petición tiene múltiples valores.
- `NotSupportedException`: La cabecera `Content-type` de la petición no es la esperada. La mayoría de las operaciones esperan el valor `application/json` en esta cabecera.
- `NotSupportedSignatureParametersException` y `UnexpectedSignerParametersTypeException`: Los parámetros solicitados en la definición de firma son incorrectos.
- `SignaturePolicyNotFoundException`: No se ha indicado ninguna política de firma.
- `SignaturePolicyNotSupportedException`: La política de firma solicitada no está soportada.
- `UndefinedIdentityDomainException`: El `token` de autorización está asociado a un dominio de identidad no definido. Si se trata de un `token JWT` de un servidor delegado, significa que el campo `sub` del `token`
- `UnmatchingSignIdentityTypeException` y `UnsuitableDeviceForSignIdentityException`: Las etiquetas de la identidad de firma que se solicita crear no tienen correspondencia con ningún tipo de identidad establecido en la configuración.
- `UnsupportedSignIdentityCapabilityException`: La identidad de firma especificada no es compatible con la operación solicitada.

3.2. HTTP/1.1 401 Unauthorized

La petición no incluye credenciales de autenticación válidas.

- `UserAuthenticationRequiredException`: El `token` de acceso no está asociado a un usuario.

3.3. HTTP/1.1 403 Forbidden

El servidor no autoriza el acceso al recurso, por ejemplo, porque el tipo de autorización utilizada no es correcta o no ofrece permisos suficientes.

- `DisabledSignIdentityException`: La identidad de firma está inhabilitada.
- `InsufficientPermissionsException` y `PermissionsRequiredException`: Los permisos otorgados por el `token` de autorización no son suficientes para acceder al recurso protegido. Si se ha utilizado un `token JWT`, los permisos del `token` son fijos.

- `InsufficientPermissionsForAttributeException`: Los permisos otorgados por el *token* de autorización no son suficientes para obtener alguno de los atributos solicitados en la petición.
- `InvalidPrivilegeException` y `NoSuchPrivilegeException`: Los permisos otorgados por el token de autorización no son suficientes para ejecutar la operación.
- `NotAuthorizedException`: El *token* de autorización no se ha obtenido con el flujo OAuth adecuado, o bien no se ha especificado ni se puede deducir el dominio de identidad.
- `LockedSignIdentityException`: La identidad de firma en servidor está bloqueada.
- `UserNotAuthorizedException`: El *token* de autorización no debería estar asociado a un usuario, o no está asociado al mismo usuario que se indica en la petición.
- `UserDomainAccessRequiredException`: Se requiere autorización administrativa de acceso al dominio de identidad. Si se ha autorizado la llamada mediante un *token* OAuth, este error puede ser debido a que el *token* ha sido emitido por un servidor de autorización que no está asociado a ningún dominio de identidad.

3.4. HTTP/1.1 404 Not Found

Recurso no encontrado.

- `DocumentNotFoundException` y `DocumentNotFoundExceptionRuntime`: El documento sobre el que se desea realizar una operación no existe.
- `ItemNotFoundException` y `NotFoundException`: El recurso sobre el que se desea realizar una operación no existe.
- `PasswordNotSetException`: No existe una contraseña registrada con ese nombre para el usuario.
- `ProcessNotFoundException`: El proceso de firma de un documento sobre el que se desea realizar una operación no existe.
- `UnknownLinkedAccountException`: La identidad vinculada indicada en la petición no pertenece al usuario.
- `UnknownUserAccountException`: No existe la cuenta de usuario indicada en la petición.

3.5. HTTP/1.1 405 Method Not Allowed

El método HTTP indicado en la petición no es adecuado para acceder al recurso.

- `NotAllowedException`: El servicio no soporta la operación HTTP indicada en la petición.

3.6. HTTP/1.1 409 Conflict

La modificación solicitada en la petición no puede aplicarse dado el estado actual del recurso.



- `InvalidOperationException`: El estado del recurso no permite realizar la operación.

3.7. HTTP/1.1 410 Gone

El recurso no existe ni existirá.

3.8. HTTP/1.1 415 Unsupported Media Type

Los datos enviados al servidor no son del tipo requerido por la operación.

3.9. HTTP/1.1 422 Unprocessable Entity

La modificación solicitada es correcta, pero no puede realizarse porque dejaría al recurso en un estado inválido.

- `InconsistentUpdateRequestException`: No se pueden aplicar simultáneamente los cambios en el recurso.

3.10. HTTP/1.1 500 Internal Server Error

Error no esperado por el servidor. En estos casos, el administrador de TrustedX debe consultar los logs para diagnosticar el problema.