



Política de gestión de errores

Oficina Técnica de Interoperabilidad

HOJA DE CONTROL DEL DOCUMENTO

| Información del Documento | | | |
|---------------------------|---|--------------------------|------------|
| Título | Política de gestión de errores | | |
| Asunto | Oficina Técnica de Interoperabilidad | | |
| Nombre del fichero | OT-I_NORM_Política de Gestión de Errores_v01r04.odt | | |
| Versión | <v01r04> | Fecha versión | 12/09/2024 |
| | | N.º Total Páginas | 14 |

| Control de Versiones | | | |
|----------------------|---|---------------|-------------------|
| Versión | Descripción de los cambios | Elaborado por | Fecha Elaboración |
| v01r00 | Elaboración inicial del documento | OT-I | 28/11/16 |
| V01r01 | Revisión y ajuste de formato | OT-I | 18/11/20 |
| V01r02 | Revisión y ajuste de formato | OT-I | 20/01/21 |
| V01r03 | Revisión y ajuste de formato | OT-I | 28/01/21 |
| v01r04 | Actualización encabezamiento y pie a la nueva versión | OT-I | 12/09/2024 |

| Lista de Distribución | |
|-----------------------|-----------------|
| Apellidos, Nombre | Cargo / Función |
| | |
| | |
| | |
| | |
| | |

ÍNDICE

| | |
|---|-----------|
| 1. INTRODUCCIÓN..... | 4 |
| 2. TIPOLOGÍA DE ERRORES..... | 5 |
| 2.1. Errores de comunicación..... | 5 |
| 2.2. Errores técnicos de aplicación..... | 5 |
| 2.3. Errores funcionales de la aplicación..... | 5 |
| 3. PLAN DE ACTUACIÓN DE ERRORES DE MENSAJERÍA..... | 6 |
| 3.1. Errores en la entrega del mensaje..... | 6 |
| 3.2. Errores en la validación de un mensaje recibido..... | 6 |
| 3.3. Errores en el procesamiento del mensaje recibido..... | 6 |
| 4. CATÁLOGO DE ERRORES FUNCIONALES..... | 8 |
| 5. SERVICIO DE NOTIFICACIÓN DE ERROR Y RECEPCIÓN..... | 9 |
| 6. IMPLEMENTACIÓN DE ERRORES..... | 10 |
| 6.1. Servicios REST..... | 10 |
| 6.2. Servicios SOAP..... | 11 |
| 7. ESQUEMA SITUACIÓN..... | 13 |
| 8. BIBLIOGRAFIA Y REFERENCIAS..... | 14 |

1. INTRODUCCIÓN

Dentro de un escenario de interoperabilidad complejo, en el que intervienen varios sistemas de información, es crucial tener definida una política que gestione los errores para cubrir al completo los posibles escenarios que puedan producirse, tanto para errores de carácter técnico, desacoplados del servicio y de la arquitectura del sistema, como errores funcionales derivados del propio tratamiento del mensaje. Esta política debe tenerse en cuenta a la hora de diseñar e implementar los servicios que proporcionan los sistemas de información.

Tanto los sistemas consumidores como los proveedor deben responsabilizarse de la mensajería en las etapas que procedan. Estas responsabilidades, los diferentes escenarios que pueden darse, así como la implementación de los errores, están recogidos en el presente documento.

2. TIPOLOGÍA DE ERRORES

Dentro de toda llamada a servicios de interoperabilidad distinguir la siguiente tipología de errores:

- Errores de comunicación.
- Errores técnicos de aplicación.
- Errores funcionales de aplicación.

2.1. Errores de comunicación

Se definen como errores de comunicación aquellos relativos al envío y aceptación de los mensajes por parte del sistema receptor de la misma.

Los receptores de la mensajería deberán realizar una validación sintáctica:

- Esquema
- Valores de los campos (opcionalidad y/o cardinalidad)
- Formato

Si el mensaje no supera las validaciones se deberá informar al sistema origen de la llamada.

2.2. Errores técnicos de aplicación

En el procesamiento de mensajería, el sistema proveedor es susceptible de producir errores, tales como:

- Errores en el código en el desarrollo
- Errores de caída de base de datos
- Errores de sistema operativo
- Errores no controlados
- Etcétera

2.3. Errores funcionales de la aplicación

Estos errores son los derivados del tratamiento de la información contenida en el mensaje aceptado por el sistema proveedor. Están directamente relacionados con el propio negocio de la mensajería y con el servicio en concreto.

3. PLAN DE ACTUACIÓN DE ERRORES DE MENSAJERÍA

3.1. Errores en la entrega del mensaje

Los sistemas responsables de formar y enviar la mensajería son los sistemas consumidores de los servicios. En el momento de realizar la entrega del mensaje pueden producirse problemas, como mini cortes en las comunicaciones, que impidan entregar el mensaje al sistema proveedor. Es responsabilidad del consumidor el definir los mecanismos que estime oportunos como medida a realizar cuando reciben un error en la entrega del mensaje.

La recomendación por parte de la OT-I es que sólo se de por finalizada la operación o recurso cuando se haya entregado.

Todos los servicios, independientemente de que estén definidos como síncronos o asíncronos, deben tener una respuesta para verificar que el mensaje se ha entregado correctamente y además cumple con una validaciones mínimas. En el caso de que se produzca algún problema en la comunicación, el propio protocolo de comunicación informará del error al origen de la emisión.

3.2. Errores en la validación de un mensaje recibido

Una vez que el mensaje es recibido por el sistema proveedor debe realizarse una validación sintáctica del mensaje. Esta validación se realizará a partir del esquema del mensaje, XSD, fichero que debe ser conocido por el sistema consumidor, junto con el contrato del servicio, y que cubrirá esquema, valores de los campos (opcionalidad y/o cardinalidad) y formato.

Cuando un mensaje supera la validación a realizar, el sistema proveedor debe generar un mensaje informando que la comunicación ha sido correcta. Por el contrario, si se produce algún problema en la validación, el sistema proveedor deberá generar un mensaje informando del error producido al sistema origen.

3.3. Errores en el procesamiento del mensaje recibido

Cuando el procesamiento desconectado de un mensaje genere o detecte un error funcional o semántico, caben dos posibilidades:

3.3.1. Error interno en la aplicación destino

Una vez entregado el mensaje, es responsabilidad del sistema receptor tratar el mensaje recibido. Durante el tratamiento del mensaje pueden darse errores del tipo tabla bloqueada, caída de BBDD,..., que deben ser informados al sistema que invocó el servicio.

Existen dos situaciones:

- Servicios síncronos: la respuesta del mensaje llevará información del error producido.
- Servicios asíncronos: se informará al sistema origen del error producido a través de un servicio asíncrono de notificación de error. Para ello, el sistema origen debe tener implementado el cliente del servicio para poder recibir dicha notificación.

3.3.2. Error provocado por el contenido del mensaje

Estos errores deben ser reportados al origen, se tratan básicamente errores provocados por la información contenida en el mensaje y que pueden provocar inconsistencia de datos entre los sistemas integrantes. Existen dos situaciones:

- Servicios síncronos: La respuesta del mensaje llevará información del error producido.
- Servicios asíncronos: se informará al sistema origen del error producido a través de un servicio asíncrono de notificación de error. Para ello, el sistema origen debe tener implementado el cliente del servicio para poder recibir dicha notificación.

4. CATÁLOGO DE ERRORES FUNCIONALES

Para conocer los errores funcionales que pueden presentarse en cada servicio debe consultarse el contrato de servicio correspondiente, donde se expondrán aquellos errores que sean propios del servicio. También, puede producirse algún error funcional que sea genérico para varios servicios existentes en el sistema de información proveedor. Para consultar estos últimos es necesario dirigirse al documento de elementos comunes del sistema.

5. SERVICIO DE NOTIFICACIÓN DE ERROR Y RECEPCIÓN

(Este servicio se encuentra en desarrollo)

Como se ha comentado en el documento, para la comunicación de ciertos errores producidos tras la entrega del mensaje, y en el escenario de que el servicio sea asíncrono, el sistema proveedor debe implementar la invocación a un servicio que permita informar al sistema origen del error producido, y por consiguiente, el sistema origen debe implementar un cliente para la recepción de dicha notificación.

La Plataforma de Interoperabilidad de CHAP dispone de un servicio que se encarga de la gestión de este tipo de notificaciones de error. De esta forma, el sistema proveedor de un servicio asíncrono, cuando tenga la necesidad de informar sobre algún error producido tras la entrega del mensaje, notificará a la Plataforma, quien dispondrá de la lógica de tratamiento y enrutado necesarios para derivar la notificación de error al sistema que le corresponda. Es recomendado por la OT-I el uso de este servicio, en caso de no desear implementarlo se deberá gestionar de manera propia los errores abordados en el servicio de notificación de errores.

Para más información sobre este servicio, es necesario consultar el contrato del servicio `NotificacionError`.

6. IMPLEMENTACIÓN DE ERRORES

Una vez definida toda la política de gestión de errores, la cuál es independiente al tipo de Servicio Web, vamos a mostrar la implementación de errores necesaria para servicios web según sean SOAP o REST.

6.1. Servicios REST

La respuesta de un servicio RestFull ante un error debe aportar información breve y concisa sobre el motivo del mismo al sistema origen. La estructura debe ser la siguiente:

```
{  
  "status": "Código de estado HTTP",  
  "type": "Identificador del origen del mensaje",  
  "code": "Código de error",  
  "message": "Texto descriptivo"  
}
```

HTTP define un conjunto de códigos de status que pueden ser devueltos por la API. A continuación se indica una lista de los códigos a utilizar:

- 200 OK - Respuesta a un exitoso GET, PUT, PATCH o DELETE. Puede ser usado también para un POST que no resulta en una creación.
- 201 Created – [Creada] Respuesta a un POST que resulta en una creación. Debería ser combinado con un encabezado Location, apuntando a la ubicación del nuevo recurso.
- 400 Bad Request – [Petición Errónea] La petición está mal formada, como por ejemplo, si el contenido no fue bien parseado.
- 401 Unauthorized – [Desautorizada] Cuando los detalles de autenticación son inválidos o no son otorgados. También útil para disparar un popup de autorización si la API es usada desde un navegador.
- 403 Forbidden – [Prohibida] Cuando la autenticación es exitosa pero el usuario no tiene permiso al recurso en cuestión.
- 404 Not Found – [No encontrada] Cuando un recurso no existente es solicitado.
- 405 Method Not Allowed – [Método no permitido] Cuando un método HTTP que está siendo pedido no está permitido para el usuario autenticado.

- 429 Too Many Requests – [Demasiadas peticiones] Cuando una petición es rechazada debido a la tasa límite.
- 500 Internal Server Error – [Error interno] Cuando se produce un error interno en el servidor ajeno a la naturaleza del servicio.
- 503 – Service Unavailable – [Servicio no disponible] Cuando el servicio no puede responder a la petición por estar congestionado o realizando tareas de mantenimiento.
- 504 Gateway Timeout – [Tiempo de espera superado] Cuando el servicio actual de intermediario y no ha recibido a tiempo una respuesta por lo que no puede responder adecuadamente al sistema origen.

Todos los errores funcionales que puedan darse en un sistema destino deberán ser recogidos bajo el status 500. El listado de code que especifica cada uno de los posibles errores funcionales debe estar indicado en los documentos técnicos mencionados en el punto 4.

6.2. Servicios SOAP

La respuesta de un servicio SOAP ante un error debe implementarse de dos formas diferentes según el error que se produzca:

- Errores desvinculados de la funcionalidad del servicio: dentro de esta implementación entran los errores de comunicación, los errores derivados de la validación del mensaje y los errores internos de la aplicación. Todos ellos serán implementados a través de soapFault, en su versión 1.2. (<http://schemas.xmlsoap.org/soap/envelope/>).
- Errores funcionales: todos aquellos derivados del tratamiento de la información obtenida del mensaje y que derivan en problemas como inconsistencia de datos, datos maestros no registrados en el proveedor, Estos errores se implementarán a través de un formato definido para ello.

```
<xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="functional-error">
    <xs:complexType>
      <xs:sequence>
        <xs:element type="xs:string" name="code"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```
<xs:element type=" xs:string" name=" type" />  
<xs:element type="xs:string" name="description"/>  
<xs:element type="xs:dateTime" name="timestamp"/>  
</xs:sequence>  
  </xs:complexType>  
  </xs:element>  
</xs:schema>
```

7. ESQUEMA SITUACIÓN

Una vez definida la implementación según el tipo de servicio Web y el tipo de error se realiza resumen esquemático de las distintas posibilidades.

| | | SOAP | | REST | | |
|--------------------------------------|---|---|--|--|--|-----------------------------|
| | | Síncrono | Asíncrono | Síncrono | Asíncrono | |
| Err. en la entrega | Err. en la validación | El protocolo de comunicación devuelve error SoapFault en la misma transacción de la llamada | | El protocolo de comunicación devuelve error HTTP en la misma transacción de la llamada | | Err. de comunicación |
| | Err. en la validación | El sistema proveedor devuelve error SoapFault en la misma transacción de la llamada | | El sistema proveedor devuelve error HTTP en la misma transacción de la llamada | | |
| Err. en el procesamiento del mensaje | Err. interno en la aplicación destino | El SP devuelve mensaje SoapFault en la misma transacción de la llamada | El SP informa del error a través del servicio de notificación de error | El SP devuelve error HTTP en la misma transacción de la llamada | El SP informa del error a través del servicio de notificación de error | Err. técnicos de aplicación |
| | Err. provocado por el contenido del mensaje | El SP devuelve error funcional en la misma transacción de la llamada | El SP informa del error a través del servicio de notificación de error | El SP devuelve error HTTP (500) en la misma transacción de la llamada | El SP informa del error a través del servicio de notificación de error | |

8. BIBLIOGRAFIA Y REFERENCIAS

| Referencia | Fuente |
|---|--|
| Buenas Prácticas para diseñar una API RESTful | https://unpocodejava.wordpress.com/2013/11/04/buenas-practicas-para-disenar-una-api-restful-pragmatic-parte-i/ |
| SOAP | https://es.wikipedia.org/wiki/Simple_Object_Access_Protocol https://www.w3.org/TR/2003/REC-soap12-part0-20030624/ |
| REST | https://es.wikipedia.org/wiki/Representational_State_Transfer |
| Basic Profile 1.1 | http://www.ws-i.org/Profiles/BasicProfile-1.1.html |
| Basic Profile 2.0 | http://ws-i.org/Profiles/BasicProfile-2.0-2010-11-09.html |
| Web Services Interoperability Organization | http://www.ws-i.org/deliverables/workinggroup.aspx?wg=testingtools |