

## Passerelle - Development #35029

### astregs: en cas de soucis de connexion, self.call provoque un crash 500

24 juillet 2019 15:02 - Thomas Noël

<b>Statut:</b>	Fermé	<b>Début:</b>	24 juillet 2019
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Benjamin Dauvergne	<b>% réalisé:</b>	0%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>		<b>Planning:</b>	Non
<b>Patch proposed:</b>	Oui		

#### Description

ça m'étonne un peu, mais sur un appel SOAP qui plante pour un pépin réseau, on reçoit une 500 avec ce payload :

```
{"err_class": "requests.exceptions.ConnectionError", "err_desc": "('Connection aborted.', error(\n(104, 'ECONNRESET')\n", "data": null, "err": 1}
```

Pour moi il faut que ça réponde en 200, on a le "err:1" qui dit que Passerelle a fait son travail mais que c'est l'appli en face qui a un soucis.

#### Révisions associées

##### Révision 676bf063 - 12 août 2019 13:11 - Benjamin Dauvergne

utils: raise APIError for wsdl schema loading errors (#35029)

#### Historique

##### #2 - 24 juillet 2019 15:38 - Thomas Noël

- Fichier 0001-astregs-handle-low-level-connection-errors-35029.patch ajouté
- Tracker changé de Support à Bug
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

##### #3 - 24 juillet 2019 16:06 - Benjamin Dauvergne

Un peu perdu parce qu'il n'y a pas de self.call uniquement du

```
client = get_client(...)  
response = client.service.op(...)
```

Mais là leur truc est bien down, ça ferme la connexion dès que ça reçoit du HTTP qu'il ne veut pas (c'est à dire tout pour l'instant) :

```
$ openssl s_client -host test-ws-astre-gs.departement06.fr -port 443 -servername test-ws-astre-gs.departement06.fr  
CONNECTED(00000003)  
depth=2 C = FR, O = Dhimyotis, CN = Certigna  
verify return:1  
depth=1 C = FR, O = DHIMYOTIS, OU = 0002 48146308100036, 2.5.4.97 = NTRFR-48146308100036, CN = Certigna Wild C  
A  
verify return:1  
depth=0 C = FR, L = NICE, O = DEPARTEMENT des alpes maritimes, 2.5.4.97 = NTRFR-22060001900016, OU = 0002 2206  
0001900016, CN = *.departement06.fr  
verify return:1  
---  
Certificate chain  
 0 s:/C=FR/L=NICE/O=DEPARTEMENT des alpes maritimes/2.5.4.97=NTRFR-22060001900016/OU=0002 22060001900016/CN=*.  
departement06.fr  
  i:/C=FR/O=DHIMYOTIS/OU=0002 48146308100036/2.5.4.97=NTRFR-48146308100036/CN=Certigna Wild CA  
 1 s:/C=FR/O=DHIMYOTIS/OU=0002 48146308100036/2.5.4.97=NTRFR-48146308100036/CN=Certigna Wild CA  
  i:/C=FR/O=Dhimyotis/CN=Certigna  
 2 s:/C=FR/O=Dhimyotis/CN=Certigna  
  i:/C=FR/O=Dhimyotis/CN=Certigna  
---  
Server certificate  
-----BEGIN CERTIFICATE-----
```

MII2zCCBsOgAwIBAgIQhALP5fWvwnJOsR1MPz8NeDANBgkqhkiG9w0BAQsFADB5MQswCQYDVQQGEwJGUjEESBAGALUECgwJREhJTVlPVElTMRwwGgYDVQQQLDBMwMDAYIDQ4MTQ2MzA4MTAwMDM2MR0wGwYDVQRhDBROVFGUj00ODE0NjMwODEwMDAznjEzMBcGALUEAwWQQ2VydGlbnmEgV2lsZCBDQTAeFw0xOTA0MTYwODM3NTNaFw0yMTA0MTUwODM3NTNaMIGgMQswCQYDVQQGEwJGUjENMASGALUEBwwETk1DRTEoMCGYGA1UECgwfREVQVJURU1FTlQgZGVzIGFscGVzIGlhcml0aW1lc2EdMBsGALUEYQWUTlRSR1ItMjIwNjAwMDE5MDAwMTYxhDAaBgNVBAsMEzAwMDIwNjIwNjAwMDE5MDAwMTYxGzAZBgNVBAMMEiouZGVwYXJ0ZW1lbnQwNi5mcjCCASIdDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAN4yzNK+F1YoxMJVdfqpc5CFNNyD13xQpUtSVq25Ar3YJC3NMWzP25mKoufXKUCw13qYkbR4G+b9pr46Me5+bfEgBetJYicEGoV4qF0qSIRKX3UfQE79v/1+ZfG4GAgNZF7cm7Kz21b9dwmJFoBrArultvSfjFa5Utb1DUfQ36wnBD1VH2SjhAewYP5zWLeR3eSa235zhHs1fVrOEFJci+h8Jgh71P+FfWY2ZEski50ldiDmCSt6FEC1h1RAx8QOqX9h5dIoRKPiu2ryRnqtuG7TCE8FSbxxvsBPQZ0aiA61CKUDQdpIcp6nhXzq/lpoGL4vnzJ3JUy4j2x5mz0kCAwEAaOCBDUwggQxMAkGALUdEwQCMAAwDgYDVR0PAAQh/BAQDAgWgMB0GALUdJQQWMBQGCCsGAQUFBwMBBggrBgEFBQcDAjBdBgNVHR8EVBjBUMCEgJaAjlhFodHRwOi8vY3J3LmN1cnRzZ25hLmZyL3dpbGRjYS5jcjcwKkAAnoCWGI2h0dHA6Ly9jcmwuzGhpbXlvdGlzLmNvbS93aWxkY2EuY3J3MIHUBGgrBgEFBQcBAQSBxZCBxDAyBggrBgEFBQcwoAomaHR0cDovL2F1dG9yYXRRLmN1cnRzZ25hLmZyL3dpbGRjYS5kZXIwNAYIKwYBBQUHMAKGGh0dHA6Ly9hdXRvcml0ZS5kaGltew90aXMuY29tL3dpbGRjYS5kZXIwKwYIKwYBBQUHMAAGHmh0dHA6Ly93aWxkY2Eub2NzC5jZxJ0aWduYS5mcjAsBggrBgEFBQcWAAyYgaHR0cDovL3dpbGRjYS5vY3NwLmRoaw15b3R3cy5jb20wHQYDVR00BBYEF0ATkq5koc9rQze0bPaGFwCav1PLMB8GALUdIwQYMBAAFI58F7e89800mNa2NZkiKsTDniRnMC8GALUdEQQoMCAEiouZGVwYXJ0ZW1lbnQwNi5mcoIQZGVwYXJ0ZW1lbnQwNi5mcjBUBGNVHSAETTLBMAgBmeBDAECAja/BgsqgXoBgTECBwECATAwMC4GCCsGAQUFBwIBFIJo dHRwczovL3d3dy5jZxJ0aWduYS5mci9hdXRvcml0ZXMvMIIB9gYKKwYBBAHweQIEAgSCAeYEGgHiAeAAdgDuS723dc5guuFCaR+r4Z5mow9+X7By2IMaxHuJeqj9yWAAAWoLs61nAAAEAwBHMEUCIQDg6iglrSv0e5+t3lxvUzGIFEAQroF6j5cST6KDbcq3twIqQ4/j8t66YjphZCELTcmNZLaOsNkDOG5VCZbul8nerYoAdgBvU3asMfAxGdiZAKRRFf93FRWR2QLBACGjbiImjzEwAAAWoLs7H6AAAEAwBHMEUCIQCNWwHj09NcFreU66V7kehRu9KSTIbm7FqRhtDwnCBoJgIqNoN1AcWWjaxfKaeyPncefaMAvYGRft00ghIzCZd8BdkAdwBc3E0S/uarRUSxXprUVuYQN/vv+kfcoXOUsl7m9scOygAAAWoLs7RrAAAEAwBIMEYCIQCKj8MLK3Knerjp4nHEhgphAuKRRnT3ISOacldprQsjHwIhAJ5p7xLNaNYCUcPNaT9VjmtSMHjCWfBYqVOyv8njhWcAHUA9lyUL9F3MCIUVBgIMJRWjuNNEkzv98MLyALzE7xZOMAAAFqJUu00wAABAMARjBEAiB/OzOegRzbVBKAwtckWhXHL88dVhAt4NvNSD5Ed82PmQIqYV6E4j/+uk787ZLdlJn6D6vyHn10LOTswOTVgYzXpw8wDQYJKoZIhvcNAQELBQADggIBAIxvcti95jNR4aMoDGeKSuxHPglBfu5ZROGv6rUKB363zzBXfZEGALVgIYriLTLCEymXl3j2fXhsygitigNqef9YPSXymPoSxbjrbnMy61LDEPYpa8m6CdZ07bVHi/IXNn1B41+/GalGgKhZKRbV2Lc bmsjFxn8topOQ210W0Z/ybXsbCywqUSNGEUF2N/krdrCCTNPWrgQcDy33IEH4rS6xy4QJfLk41KoHifbuxtPvJ4AzsZU8jDqIpT79g8MMB10v5yAmHEYf0Ht+H+WpU0VzCpaCectnbobE/VUPDnfXGgSDv08I50PN7OxTgGQ2+W5JN66AgBTEsx5EMjNRNUjYg5n9JyfabFftR0KnsC5Q280j6Cjj7B0SNEBb2JxLsJB0jbaJaEhGS8fnc7XMFrHBAjBGRtwcFQ3g93bf+1lWzozoZ/1+cEhLh4QRXCT7BiXT+B9dREFTENxs3ngWtt9URdvrVOSQE7AlwkhOqClBwgdmGfQWkG/JMtWGynFEEgV8k+dzI5V39IyoFWhK8NgPM3udouXhHF/XmVynsHSSxUnLduGLtrMYRtnScIwLqSvMUlaQ8P1Um4Ats/LJnwxh5gi/TDIFae/Pse4HAT/hv2yPS1TwKEPfrf8g14y6TsUCn9IH09R9K59gctTWKQf4eI8thJerQKvyObf6XM

-----END CERTIFICATE-----

subject=/C=FR/L=NICE/O=DEPARTEMENT des alpes maritimes/2.5.4.97=NTRFR-22060001900016/OU=0002 22060001900016/CN=\*.departement06.fr

issuer=/C=FR/O=DHIMYOTIS/OU=0002 48146308100036/2.5.4.97=NTRFR-48146308100036/CN=Certigna Wild CA

---

No client certificate CA names sent

Peer signing digest: SHA256

Server Temp Key: DH, 1024 bits

---

SSL handshake has read 5469 bytes and written 408 bytes

Verification: OK

---

New, TLSv1.2, Cipher is DHE-RSA-AES256-GCM-SHA384

Server public key is 2048 bit

Secure Renegotiation IS supported

Compression: NONE

Expansion: NONE

No ALPN negotiated

SSL-Session:

Protocol : TLSv1.2

Cipher : DHE-RSA-AES256-GCM-SHA384

Session-ID: A2982432F7DD99178B615F965E25409F220F1D632CA3EC07E69EA4162EB590AF

Session-ID-ctx:

Master-Key: B17401C7244A938D0B88F26985439E0B3D547D4AC5DD8AFAD2E0F64FF67FA8638A4592EFA09E551A8FF0780667C0EA

C0

PSK identity: None

PSK identity hint: None

```
SRP username: None
Start Time: 1563976108
Timeout : 7200 (sec)
Verify return code: 0 (ok)
Extended master secret: no
```

```
GET / HTTP/1.1
```

```
read:errno=104
```

Ici y a tout qui foire, le `check_status` sur l'URL de base (dont je ne sais pas vraiment ce qu'elle retourne normalement) mais aussi la récupération des WSDL (qui se faisant or boucle SOAP request/response fait qu'on se prend des erreurs requests alors que zeep a aussi une exception `TransportError`).

Ce que je propose c'est de transformer les erreurs requests dans notre méthode `SOAPTransport._load_remote_data()` en `APIError` claires<sup>1</sup>.

<sup>1</sup> On peut pas vraiment faire mieux, la chaîne actuelle d'initialisation d'un client c'est

- `Client.__init__(location) => self.wsdl = Document(location, self.transport)`
- `Document.__init__(location, transport) => self.transport = transport`
  - `document = self._get_xml_document(location)`
- `get_xml_document(location) => load_external(location, self.transport)`
- `load_external(location, transport) => transport.load(location)`
- `load(location) => self._load_remote_data(url)`

#### #4 - 24 juillet 2019 16:23 - Thomas Noël

Benjamin Dauvergne a écrit :

Un peu perdu parce qu'il n'y a pas de `self.call` uniquement du ...

Pas bien compris ta remarque.

Dans `astregs` il y a un `self.call` qui lance toute la mécanique `zeep`, et qui fait le `try/except` sur les erreurs `zeep`. J'y ajoute juste ici les potentiels `crash low level`... c'est un peu sale mais bon, comme `zeep` ne semble pas attraper ces choses là.

Dans notre méthode `SOAPTransport._load_remote_data()` en `APIError` claires

Ça ce n'est que la partie récupération `wsdl`, il faudrait aussi attraper le `post` fait ensuite sur le service.

Et plutôt que d'y faire une `APIError`, pourquoi ne pas émettre une `zeep.exception.TransportError` ?

#### #5 - 24 juillet 2019 16:32 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

#### #6 - 24 juillet 2019 16:32 - Benjamin Dauvergne

- Fichier `0001-utils-raise-APIError-for-wsdl-schema-loading-errors-patch` ajouté

- Tracker changé de Bug à Development

Contre `propale`, ça corrige le souci dans tous les connecteurs WSDL avec un message plus compréhensible.

#### #7 - 24 juillet 2019 16:44 - Benjamin Dauvergne

Thomas Noël a écrit :

Benjamin Dauvergne a écrit :

Un peu perdu parce qu'il n'y a pas de `self.call` uniquement du ...

Pas bien compris ta remarque.

Mon dépôt était trop vieux je n'avais pas de `call`, mais ça ne change rien à mon analyse.

Dans `astregs` il y a un `self.call` qui lance toute la mécanique `zeep`, et qui fait le `try/except` sur les erreurs `zeep`. J'y ajoute juste ici les potentiels `crash low level`... c'est un peu sale mais bon, comme `zeep` ne semble pas attraper ces choses là.

Uniquement sur la résolution du WSDL.

Dans notre méthode SOAPTransport.\_load\_remote\_data() en APIError claires

Ça ce n'est que la partie récupération wsdl, il faudrait aussi attraper le post fait ensuite sur le service.

Si on a pas récupérer le WSDL on risque pas d'avoir de POST (d'ailleurs tu verras que dans mon test modifié du tien je surcharge le POST différemment pour être sûr qu'il n'y en a pas). Je n'ai pas trouvé de trace où ça plante sur le POST, ça plante toujours avant ou ça ne plante pas (ça ne veut pas dire que c'est impossible, mais là c'est pas le cas). Aussi toutes RequestException sur un POST/SOAP sera convertie en TransportError, ce ne sont que les GET de résolution du .wsdl et de ses dépendances qui ne sont pas converties.

Et plutôt que d'y faire une APIError, pourquoi ne pas émettre une zeep.exception.TransportError ?

Parce que la résolution WSDL c'est du meta-SOAP ça fait bizarrement pas partie de la norme (dans les specs SOAP à aucun moment ça n'indique comment récupérer le WSDL, t'es sensé l'avoir point), je suis plus à l'aise de remonter directement à haut niveau les erreurs de résolution du WSDL, on comprend très bien ce qui se passe en fait, pas besoin de laisser aux connecteurs le soin de le gérer. En vrai j'envisage de faire disparaître TransportError des connecteurs quand j'aurai implémenté un SOAPField.

Après ça pourrait être bien de le mettre en cache juste pour les perfs (parce que quand c'est down de toute façon on aura l'erreur au POST ensuite, mais sous forme TransportError).

#### #8 - 24 juillet 2019 18:02 - Thomas Noël

A priori ça fait planter les tests bien comme il faut, dommage parce que l'idée semble effectivement intéressante.

#### #9 - 24 juillet 2019 19:54 - Benjamin Dauvergne

Thomas Noël a écrit :

A priori ça fait planter les tests bien comme il faut, dommage parce que l'idée semble effectivement intéressante.

Bizarre chez moi ça marchait avant que je pousse...

#### #10 - 26 juillet 2019 16:37 - Benjamin Dauvergne

Benjamin Dauvergne a écrit :

Thomas Noël a écrit :

A priori ça fait planter les tests bien comme il faut, dommage parce que l'idée semble effectivement intéressante.

Bizarre chez moi ça marchait avant que je pousse...

Et ça marche toujours, je cherche.

#### #11 - 26 juillet 2019 17:53 - Benjamin Dauvergne

- Fichier 0002-utils-raise-APIError-for-wsdl-schema-loading-errors-.patch ajouté

- Fichier 0001-signature-forbid-arguments-after-signature-35059.patch ajouté

Ok je me suis fait avoir par le cache de zeep.

#### #12 - 31 juillet 2019 14:38 - Thomas Noël

- Fichier 0001-signature-forbid-arguments-after-signature-35059.patch supprimé

#### #13 - 31 juillet 2019 14:40 - Thomas Noël

- Statut changé de Solution proposée à Solution validée

#### #14 - 31 juillet 2019 14:42 - Thomas Noël

- Statut changé de Solution validée à Solution proposée

Bon en fait non, comme je suis un peu tatasse j'aurai bien vu une exception genre SOAPRequestException(APIError) (ou autre nom), histoire de pouvoir la catcher spécifiquement dans certains appels SOAP qui auraient le droit de foirer, dans certains endpoint. Ok ça existe nulle part pour l'instant oui je sais mais bon.

#### #15 - 31 juillet 2019 15:51 - Benjamin Dauvergne

Thomas Noël a écrit :

Bon en fait non, comme je suis un peu tatasse j'aurai bien vu une exception genre SOAPRequestException(APIError) (ou autre nom), histoire de pouvoir la catcher spécifiquement dans certains appels SOAP qui auraient le droit de foirer, dans certains endpoint. Ok ça existe nulle part pour l'instant oui je sais mais bon.

Je suis parfaitement d'accord avec ça par contre, je vais le faire.

**#16 - 12 août 2019 13:13 - Benjamin Dauvergne**

- Fichier 0001-utils-raise-APIError-for-wsdl-schema-loading-errors-.patch ajouté

Rebasé sur master, classe SOAPError(APIError) ajoutée.

**#17 - 12 août 2019 13:57 - Thomas Noël**

- Statut changé de Solution proposée à Solution validée

**#18 - 12 août 2019 14:39 - Benjamin Dauvergne**

- Statut changé de Solution validée à Résolu (à déployer)

```
commit 676bf063e19923215a145823d44812a25a1daa10
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date: Wed Jul 24 16:30:51 2019 +0200
```

```
utils: raise APIError for wsdl schema loading errors (#35029)
```

**#19 - 12 août 2019 17:16 - Frédéric Péters**

- Statut changé de Résolu (à déployer) à Solution déployée

**Fichiers**

---

0001-astregs-handle-low-level-connection-errors-35029.patch	2,36 ko	24 juillet 2019	Thomas Noël
0001-utils-raise-APIError-for-wsdl-schema-loading-errors-.patch	3,32 ko	24 juillet 2019	Benjamin Dauvergne
0002-utils-raise-APIError-for-wsdl-schema-loading-errors-.patch	3,95 ko	26 juillet 2019	Benjamin Dauvergne
0001-utils-raise-APIError-for-wsdl-schema-loading-errors-.patch	5,05 ko	12 août 2019	Benjamin Dauvergne