



Debbugsとの つきあいかた

SOAP編

Kentaro Hayashi

ClearCode Inc.

東京エリアDebian勉強会

2017-05-20





自己紹介(1)

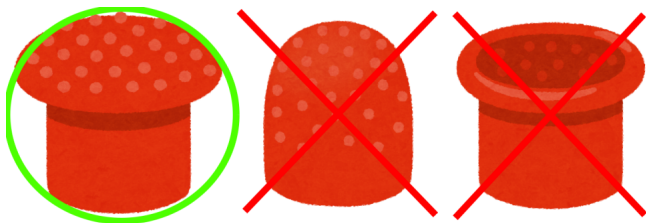


■ Twitter/GitHub: @kenhys

■ 所属: **トラックポイント・ソフトドーム派**



自己紹介(2)



■ トラックポイント・ソフトドーム派

参考画像の出典



 <http://www.irasutoya.com/>



パッケージ関連の活動

- groonga (全文検索エンジン)
- groonga-normalizer-mysql
- libhinawa
- fcitx-imlist
- hal-flash (Flashを延命させるアレ)
- growl-for-linux



閑話休題

そろそろ本題に



本日の話題

■ Debbugsとのつきあいかた:SOAP編



本日の話題

■ Debbugsとのつきあいかた :**SOAP**編



話さないこと

- Debbugsのふつうの使い方



Debbugsとは？

<https://bugs.debian.org/cgi-bin/pkgreport.cgi?dist=unstable;package=debbugs>

Debian Bug report logs: Bugs in package debbugs (version 2.4.1.1)

Maintainers for debbugs are [debbugs_developers <debian-debbugs@lists.debian.org>](mailto:debbugs_developers@lists.debian.org).

You may want to refer to the following packages that are part of the same source: [debbugs-local](#), [debbu](#)

You might like to refer to the [debbugs package page](#), to the [Package Tracking System](#), or to the sou

If you find a bug not listed here, please [report it](#).

- [Control \(service\) bugs](#) (25 bugs)
- [nnn@ \(process\) bugs](#) (24 bugs)
- [Bug CGI bugs](#) (21 bugs)
- [Package CGI bugs](#) (28 bugs)
- [Documentation bugs](#) (5 bugs)
- [SOAP/REST API bugs](#) (11 bugs)
- [Uncategorized bugs](#) (20 bugs)
- [Misc bugs](#) (37 bugs)
- [Packaging bugs](#) (2 bugs)
- [Fixed in Git bugs](#) (49 bugs)
- [Wontfix bugs](#) (16 bugs)
- [Fixed bugs](#) (6 bugs)

Control (service) bugs (25 bugs)

- [#49228 \[w\] \[i\]](#) [debbugs] [abstract out control@ interface and allow more use of Pseudoheaders:](#)
- [#364803 \[m\] \[i\]](#) [debbugs] [\[service.in\] reassign doesn't work with multiple packages+versions.](#)
- [#376293 \[w\] \[i\]](#) [debbugs] [debbugs: Add chaining options to "block" command \(when A is closed, it clo](#)



Debbugsとは？

- 1994年 Ian Jackson氏が開発を始めた
Issueトラッキングシステム
- GNOMEやKDEもかつて採用していた
→Bugzillaへ移行した

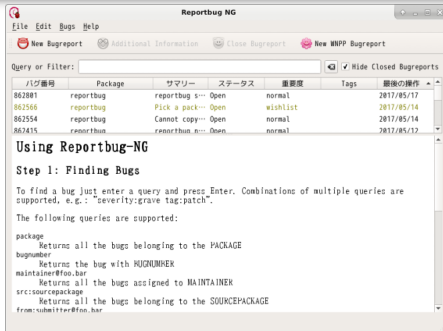
ふつうの人のバグ報告(1)



 Reportbug!



ふつうの人のバグ報告(2)



 Reportbug-NG!



Debbugsのよいところ

- メールで気軽にバグ報告
- 報告にログインとか不要
- 一見さんをカジュアルに殺せる初見殺し機能搭載



Debbugsの見方

Control (service) bugs

- [#49228](#) [[w](#)|[_](#)|[=](#)] [[debbugs](#)] [abstra](#)
- [#364803](#) [[m](#)|[_](#)|[_](#)] [[debbugs](#)] [\[ser](#)
- [#376293](#) [[w](#)|[_](#)|[_](#)] [[debbugs](#)] [debb](#)
- [#422120](#) [[m](#)|[_](#)|[_](#)] [[debbugs](#)] [DTC](#)

■ [重要度|タグ|状態]



Debugsの重要度

- m: minor
- w: wishlist
- n: normal
- S: serious
- G: grave



Debugsのタグ

- +: patch
- ☹: wontfix
- M: moreinfo
- P: pending



Debugsの状態

■ ☹: blocks

■ ☠: affects

■ 😊: fixed



なぜDebbugsに興味を？

- 社内のグループチャットがZulip
 - <https://zulip.org/>
- Zulipのbotにdebbugsをサポートさせたい
 - <https://github.com/okkez/pragmatic-bot>
- バグ報告のURLを貼り付けたらbotに処理させたい



botに処理させるには？

- bugs.d.oをスクレイピング？
- もっといいやりかたはない？



SOAP!

■ [https://wiki.debian.org/
DebbugsSoapInterface](https://wiki.debian.org/DebbugsSoapInterface)



サンプルコードもあるよ

```
require 'soap/rpc/driver'

host = "bugs.debian.org"
port = 80
server="http://#{host}:#{port}/cgi-bin/soap.cgi"
ns = 'Debbugs/SOAP/'
drv = SOAP::RPC::Driver.new(server, ns)
drv.wiredump_dev = STDOUT if $DEBUG
drv.add_method('get_status', 'bugnumber')
drv.add_method('get_bugs', 'keyparam')

p drv.get_status(drv.get_bugs(['package', 'pbuilder', 'severity', 'wishlist']))
```

■ Ruby 1.8 or **修正済みsoap4r**じゃないと動かない



Rubyの事情

- soap4rはもともとRuby 1.8にバンドル
- Ruby 1.9ではバンドルされなくなった
- soap4rはメンテされなくなった
- 雨後の筍 soap4r



雨後の筍 soap4r

- soap4r-ng (2.0.3)
- soap4r-r19 (1.5.9) Ruby1.9対応版
- soap4r-ruby1.9 (2.0.5) Ruby1.9対応版
- soap4r-ruby19 (1.5.9) Ruby1.9対応版



soapXr

- soap2r (1.5.8) soap4rの後にリリース
- soap5r (2.0.3)



upstream on Debian

■ ruby-soap4r

- <https://github.com/noeticpenguin/soap4r-noeticpenguin>



混沌とした世界

- あまり踏み込まないほうがよい予感



SOAPを使うには

■ ポイント:WSDL



WSDL

■ Web Services Description Language

- Webサービス記述言語
- XMLでインターフェースを定義する
- WSDLを元にSOAPで通信



DebbugsのWSDL

- <https://wiki.debian.org/DebbugsSoapInterface>

- 一切言及はない

- どうなってるの？



WSDLを探せ

```
git index : emacs/elpa.git
ELPA

summary refs log tree commit diff
path: root/packages/debbugs

Mode      Name
-rw-r--r-- Debbugs.wsd
-rw-r--r-- README
-rw-r--r-- debbugs-browse.el
-rw-r--r-- debbugs-gnu.el
-rw-r--r-- debbugs-org.el
-rw-r--r-- debbugs-ug.info
-rw-r--r-- debbugs-ug.text
-rw-r--r-- debbugs.el
-rw-r--r-- debbugs.info
-rw-r--r-- debbugs.text
-rw-r--r-- dir
```



Emacs使いはdebbugs.el

- <https://elpa.gnu.org/packages/debbugs.html>



Debbugs.wSDL

- Debbugs用のWSDL
- debian.orgとgnu.orgのdebbugsで使える



WSDLの構造

- `<wsdl:definitions>`の子要素として定義
 - `<wsdl:types>`
 - `<wsdl:message>`
 - `<wsdl:portType>`
 - `<wsdl:binding>`
 - `<wsdl:service>`



<wsdl:types>

```
<complexType name="ArrayOfBugNumber">
  <complexContent>
    <restriction base="soapenc:Array">
      <attribute ref="soapenc:arrayType" wsdl:arrayType="xsd:int[]"/>
    </restriction>
  </complexContent>
</complexType>
```

- データ型の定義をする
 - 例: バグ番号の配列を定義



<wsdl:message>

```
<wsdl:message name="get_statusRequest">
  <wsdl:part name="bugs" type="types:ArrayOfBugNumber"/>
</wsdl:message>
<wsdl:message name="get_statusResponse">
  <wsdl:part name="s-gensym3" type="apachens:Map"/>
</wsdl:message>
```

■ データの抽象的な定義

- 例: ステータス取得時にやりとりするデータの定義



<wsdl:portType>

■ 操作の定義 <wsdl:operation>を含む

```
<wsdl:portType name="Debbugs/SOAP">  
  <wsdl:operation name="get_status" parameterOrder="bugs">  
    ...  
  </wsdl:operation>  
  <wsdl:operation name="get_bug_log" parameterOrder="bugnumber">  
    ...  
  </wsdl:operation>  
</wsdl:portType>
```



<wsdl:operation>

```
<wsdl:operation name="get_status" parameterOrder="bugs">
  <wsdl:input message="tns:get_statusRequest">
    <soap:body
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="urn:Debbugs/SOAP"
      use="encoded" />
    </wsdl:input>
    <wsdl:output message="tns:get_statusResponse">
      <soap:body
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="urn:Debbugs/SOAP"
        use="encoded" />
      </wsdl:output>
    </wsdl:operation>
```

■ 操作の入出力を定義

- 例: ステータス取得時の入出力を定義



<wsdl:binding>

- portTypeで定義した操作のプロトコルを指定

```
<wsdl:binding name="Debbugs/SOAP/BINDING" type="tns:Debbugs/SOAP">  
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>  
  <wsdl:operation name="get_status">  
    ...  
  </wsdl:operation>  
</wsdl:binding>
```



<wsdl:service>

■ 通信先を定義

```
<wsdl:service name="Debbugs/SOAP/SERVICE">
  <wsdl:port binding="tns:Debbugs/SOAP/BINDING" name="gnu.org">
    <wsdlsoap:address location="http://debbugs.gnu.org/cgi/soap.cgi"/>
  </wsdl:port>
  <wsdl:port binding="tns:Debbugs/SOAP/BINDING" name="debian.org">
    <wsdlsoap:address location="http://bugs.debian.org/cgi-bin/soap.cgi"/>
  </wsdl:port>
</wsdl:service>
```



実践:debbugs

- 実際にSOAPでやりとりしてみるには



何を使う？



savon

- <http://savonrb.com/>
- RubyのSOAP clientライブラリー



savonの使い方

- `require "savon"`
- WSDLからクライアントのインスタンス作成
- `call`でdebbugsのメソッドを呼ぶ
- レスポンスを表示する



savonクライアント

```
client = Savon.client(wSDL: "(WSDLのURL)",  
                      endpoint: "http://bugs.debian.org/cgi-bin/soap.cgi",  
                      namespace: "Debbugs/SOAP")
```

■ クライアント作成



メソッドを呼ぶ

```
# fetch status of pbuilder specific bugs
response = client.call(:get_status) do
  message(bugs: ["807406", "837812"])
end
```

■ 例: get_statusメソッドを呼ぶ



レスポンスを表示する

```
bugs = response.body[:get_status_response][:s_gensym3][:item]
bugs.each do |bug|
  item = bug[:value]
  puts "#{bug[:key]}:#{item[:pending]}:#{item[:subject]}"
end
#=>
807406:pending:pbuilder: drop all the 'xenial' (...)
837812:pending:[pbuilder] New hook improving dpkgi
```



まとめ

- debugsにはSOAPインターフェースがある
- Debug.wsdlはdebugs.el方面から入手せよ
- Rubyから使うならsoap4r(修正版) or savonを使え