



Telegram-API mit Perl

- Deutscher Perl-Workshop 2017
- in Hamburg vom 26. bis 28. Juni 2017
- Wieland Pusch
(wieland@wielandpusch.de)
- Folien und Code unter
www.wielandpusch.de/talks/2017gpw.zip



Telegram-API mit Perl

- Warum Nachrichten verschicken?
- Warum Telegram?
- Was brauche ich?
- Wie geht das?



Warum Nachrichten verschicken?

- Meine (Perl-)Programme laufen meistens auf Servern,
- da bekomme ich nicht mit, wenn etwas passiert
- E-Mails gehen auch, aber ...
 - Langsam, viele Daten
 - Empfängerliste muss gepflegt werden
 - Funktionieren auch nicht immer
 - E-Mail-Passwörter unsicher



Warum Telegram?

- Kostet nichts zum Installieren
- Kostet auch für die API nichts
- Läuft auf vielen Plattformen
- Bietet eine mächtige API
- Ist nicht WhatsApp



Was brauche ich?

- Eine Telefonnummer mit SMS (Handy) (keine Festnetznummer)
- Ein Gerät auf dem Telegram läuft z.B. Handy (oder PC oder Webbrowser)
- Man kann also einen Laptop oder Tablet (ohne Sim-Karte) verwenden, wenn man ein altes Handy hat.



Wie geht das?

- App installieren

The screenshot shows a mobile app interface. At the top is a dark blue header with the text 'Dein Telefon' in white, followed by a white checkmark icon. Below the header is a white area with a light blue background pattern of various icons. The text 'Germany' is displayed above a horizontal line. Below that, the text '+49' is displayed above another horizontal line. At the bottom of the white area, the text 'Bitte bestätige deine Landesvorwahl und deine Telefonnummer.' is written in a smaller, grey font.



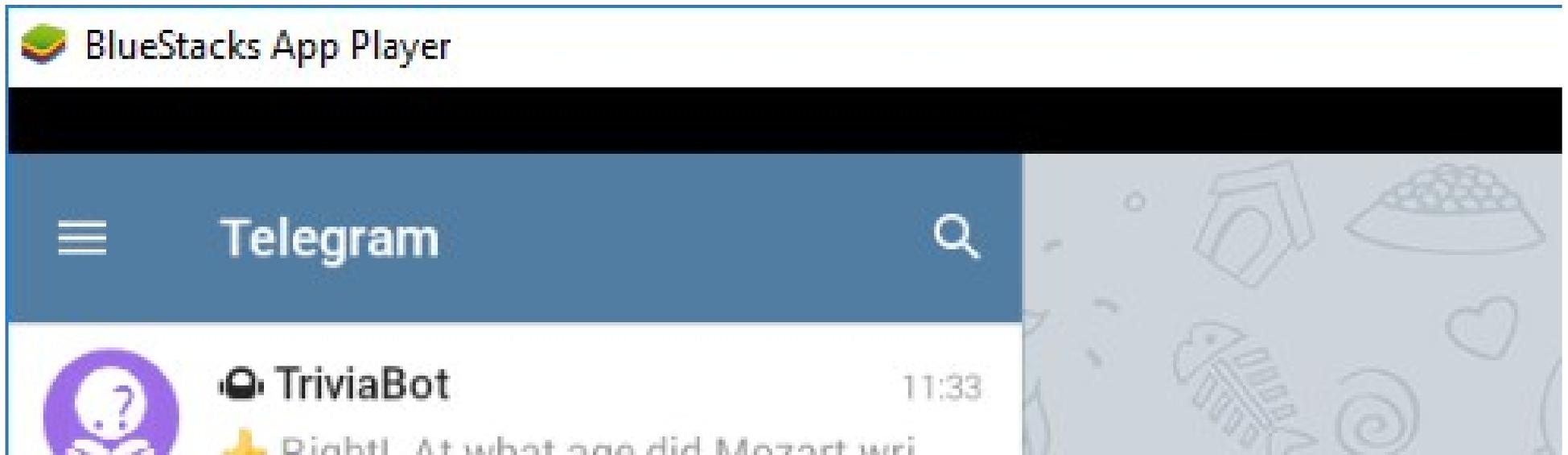
Registrieren

- Code per Telegram, falls schon registriert
- Anruf oder nach 2 Min SMS
- erhaltenen Code eingeben

The screenshot shows a mobile app interface with a blue header bar at the top containing the text "Nummer verifizieren" and a white checkmark icon on the right. Below the header, the text "Code wurde dir per Telegram an deine aktive App geschickt." is displayed in a light gray font. Underneath this text is a text input field with a light blue border and a light blue underline. The word "Code" is written in a light gray font inside the input field. At the bottom of the input area, the text "Code nicht erhalten?" is written in a light blue font.

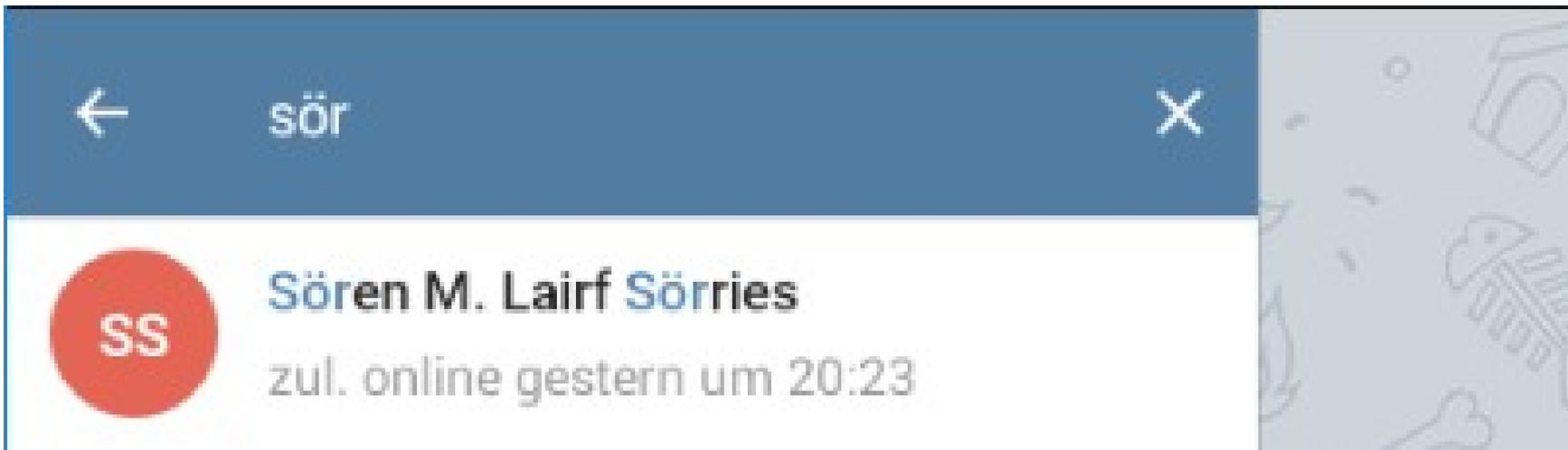
Wenn Telegram läuft

- Dann kann man mit der Lupe Kontakte und öffentliche Kanäle suchen z.B. PerlWorkshop



Kontakte

- Wenn man Telegram erlaubt hat auf die Kontakte zuzugreifen



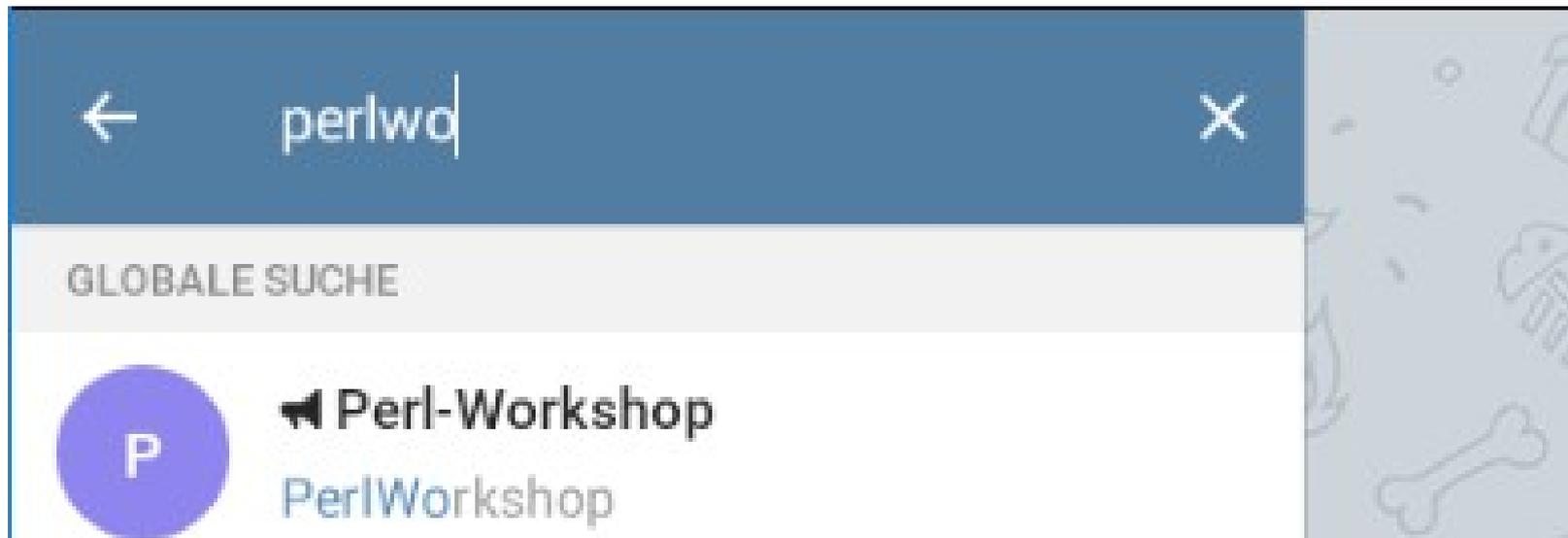
Gruppen

- Administratoren können Mitglieder hinzufügen
- Alle Mitglieder können an alle schreiben



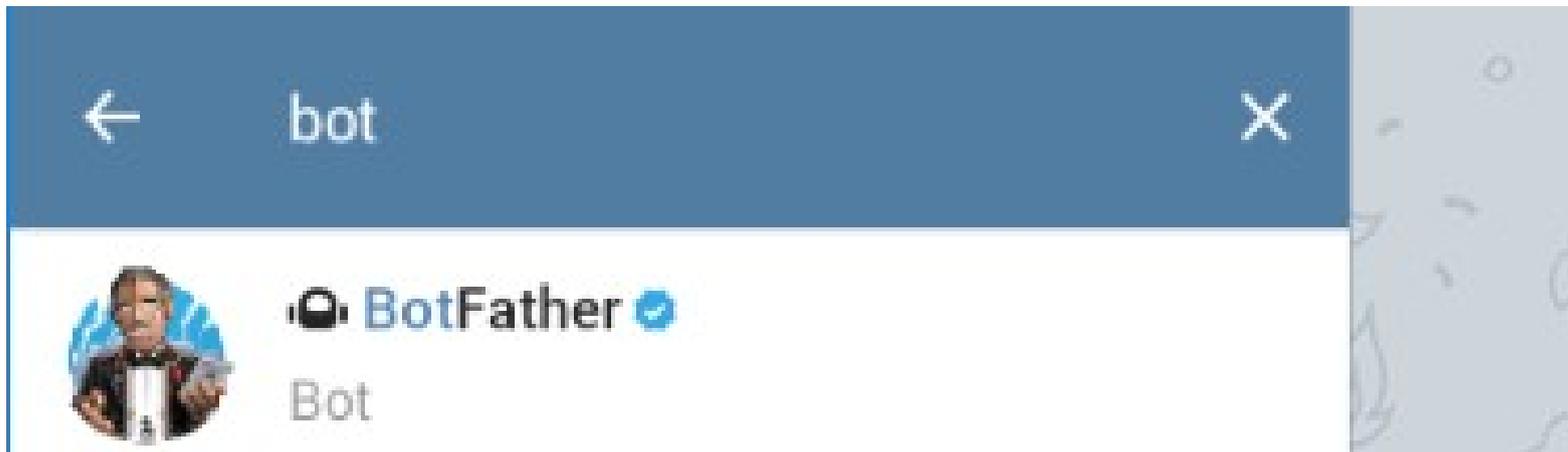
Kanäle

- Nur Administratoren können an alle schreiben
- Jeder kann sich selbst hinzufügen (Beitreten)
z.B. <https://t.me/PerlWorkshop>



Bis jetzt war das manuell

- Unterhaltung mit BotFather starten, um einen Bot zu erzeugen.





frankfurt.pm

/newbot - Token

The screenshot shows a Telegram chat with BotFather. The chat header includes a close button, BotFather's profile picture, name, and status 'Bot', and a menu icon. A date separator '18. Juni' is visible. The chat history consists of the following messages:

- A green message from the user: `/newbot` (12:23, read).
- A white message from BotFather: "Alright, a new bot. How are we going to call it? Please choose a name for your bot." (12:23).
- A green message from the user: `PerlWorkshopHH2017Bot` (12:23, read).
- A white message from BotFather: "Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot." (12:23).
- A green message from the user: `PerlWorkshopHH2017Bot` (12:24, read).
- A white message from BotFather: "Done! Congratulations on your new bot. You will find it at t.me/PerlWorkshopHH2017Bot. You can now add a description, about section and profile picture for your bot, see `/help` for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this." (12:24).
- A white message from BotFather: "Use this token to access the HTTP API:
3[REDACTED]ps" (12:24).
- A white message from BotFather: "For a description of the Bot API, see this page: <https://core.telegram.org/bots/api>" (12:24).



Bot in den Kanal aufnehmen

- Der Eigentümer (Gründer) des Kanals kann Mitglieder aufnehmen
Nach dem Bot suchen und aufnehmen
- Bots können keine Konversationen starten.
Sie müssen angesprochen oder aufgenommen werden



API

- `https://api.telegram.org/bot<token>/METHOD_NAME`
- Token hat man von BotFather bekommen
- METHODS sind in <https://core.telegram.org/bots/api> beschrieben
- Einfach im Browser testen

Zuerst Nachrichten empfangen

- Method: `getUpdates`



```
https://api.telegram.org/bot:[token]/getUpdates

JSON Rohdaten Kopfzeilen

Speichern Kopieren

ok: true
result:
```

Antwort in JSON-Format

leider noch keine Nachrichten :-)

- Bots bekommen nur Nachrichten, die mit / anfangen. z.B. `"/hallobot"`



Person spricht Bot an

- Mit /start kann man Bot ansprechen
- Der sieht das in Updates

```
▼ result:
  ▼ 0:
    update_id: 581322339
    ▼ message:
      message_id: 1
      ▼ from:
        id: 325461580
        first_name: "Wieland"
        language_code: "de-DE"
      ► chat:
        date: 1497785973
        text: "/start"
      ► entities:
        ► 1:
```



Kanal sendet an Bot

- Wenn ein Admin /hallobot sendet sieht der Bot:
- Wichtig ist die chat.id damit kann man an den Kanal senden.

```
▼ 2:
  update_id:      581322341
  ▼ channel_post:
    message_id:   11
    ▼ chat:
      id:         -1001141939388
      title:      "Perl-Workshop"
      username:   "PerlWorkshop"
      type:       "channel"
      date:       1497786052
      text:       "/hallobot"
    ▼ entities:
      ▼ 0:
        type:     "bot_command"
        offset:   0
        length:   9
```



Im Browser eine Nachricht schicken

- `https://api.telegram.org/bot<token>/sendMessage?chat_id=@PerlWorkshop&text=Message from bot`
- Browser ersetzt hoffentlich Space durch %20
- Wenn es kein öffentlicher Kanal wäre, dann hätten wir `chat_id=-1001141939388` schreiben müssen.



Was im Browser geht, geht auch in Perl

- LWP::UserAgent

oder

- Bot::ChatBots::Telegram
(hat viele abhängige Module)
- WWW::Telegram::BotAPI
- Telegram::Bot
- Telegram::BotKit



LWP::UserAgent

- Ich habe mich dafür entschieden, da ich nichts installieren musste und es hat auf Anhieb funktioniert.
- OK. Fast auf Anhieb, man sollte noch URL-Escaping beachten.
- OK. Und man sollte Texte als UTF8 schicken.



Jetzt etwas Perl

```
1  #!/usr/bin/perl
2
3  use strict;
4  use warnings;
5
6  use JSON;
7  use URI::Escape;
8  use LWP::UserAgent;
9  use Data::Dumper;
10
11 my $ua = LWP::UserAgent->new;
12 $ua->agent("MyApp/0.1");
```



API-URL aufrufen

```
14 my $token="35....ps";
15 my $chat_id = -1001141939388; # channel
16 my $text = "Der Vortrag ist zu Ende :-) "
17     . "& jetzt kommen Umlaute äöüÄÖÜß";
18
19 my $text_esc = uri_escape($text);
20 my $req = HTTP::Request->new(GET =>
21     "https://api.telegram.org/bot$token/sendMessage"
22     ."?chat_id=$chat_id&text=$text_esc");
23 my $res = $ua->request($req);
```



und Antwort prüfen

```
25  if ($res->is_success) {
26      # print "$res->content\n";
27      my $perl = decode_json($res->content);
28      # print Dumper($perl);
29      print "\nresult:\n";
30      if ($perl->{ok}) {
31          my $a = $perl->{result};
32          print "$a->{text}\n";
33      }
34  } else {
35      print $res->status_line, "\n";
36  }
```



C:\Windows\system32\cmd.exe

```
C:\src\gpw2017\talk tg>perl send2.pl
```

```
{"ok":true,"result":{"message_id":15,"chat":{"id":-1001141939388,"title":"Perl-Workshop","username":"PerlWorkshop","type":"channel"},"date":1497796124,"text":"Der Vortrag ist zu Ende :- ) & jetzt kommen Umlaute \u00e4\u00f6\u00fc\u00c4\u00d6\u00dc\u00df"}}
```

```
result:
```

```
$VAR1 = {
```

```
  'ok' => bless( do{\(my $o = 1)}, 'JSON::XS::Boolean' ),
```

```
  'result' => {
```

```
    'message_id' => 15,
```

```
    'text' => "Der Vortrag ist zu Ende :- ) & jetzt kommen Umlaute  
\x{e4}\x{f6}\x{fc}\x{c4}\x{d6}\x{dc}\x{df}",
```

```
    'date' => 1497796124,
```

```
    'chat' => {
```

```
      'type' => 'channel',
```

```
      'title' => 'Perl-Workshop',
```

```
      'id' => '-1001141939388',
```

```
      'username' => 'PerlWorkshop'
```

```
    }
```

```
  }
```

```
};
```

```
Der Vortrag ist zu Ende :- ) & jetzt kommen Umlaute ö÷³-í
```

```
C:\src\gpw2017\talk tg>
```



Telegram-API mit Perl

- Vielen Dank
- Fragen ?

- Wieland Pusch
(wieland@wielandpusch.de)
- Folien und Code unter
www.wielandpusch.de/talks/2017gpw.zip
- Code mit <http://hilite.me/> formatiert.