



When the specification fails:

documenting inter-parameter constraints

About me

Nathalie Oostvogels

PhD student

Vrije Universiteit Brussel

 @noostvog



VRIJE
UNIVERSITEIT
BRUSSEL



Research Foundation
Flanders
Opening new horizons

Third party web services



Web API specifications



POST direct_messages/new

Sends a new direct message to
Requires both the `user` and `text`
message in the requested form

Direct Message Behavior

The standard behavior of Direct Messages is to be sent to users following the sender. In most cases, Direct Messages sent to users not following the sender will not be delivered.

Alternatively, a recipient may have opted in to receive Direct Messages from users they do not follow. In this case, any Direct Message sent to a user will be delivered, whether the recipient follows the sender or not.

Once a Direct Message thread is created, it can be used to send messages bi-directionally. To stop receiving messages in a thread, a user can explicitly leave the conversation via the APIs or Twitter client.

Parameters

One of `user_id` or `screen_name` are required.

`user_id`
optional

The ID of the user who should receive the direct message.

`screen_name`
optional

The screen name of the user who should receive the direct message.

`text`
required

The text of the direct message.

Example Result

```
{
  "created_at": "Tue Aug 28 00:59:49 +0000 2012",
  "entities": {
    "hashtags": [],
    "urls": [],
    "user_mentions": []
  },
  "id": 240252311455277056,
  "id_str": "240252311455277056",
  "recipient": {
    "contributors_enabled": true,
    "created_at": "Sat May 09 17:58:22 +0000 2009",
    "default_profile": false,
    "default_profile_image": false,
    "description": "I taught your phone that thing you like. The Mobile Partner Engineer @Twitter. ",
    "favourites_count": 584,
    "follow_request_sent": false,
    "followers_count": 10622,
    "following": true,
```

Web API specifications



Parameters

One of `user_id` or `screen_name` are required.

`user_id`
optional

optional

number

Example Values: 12345

`screen_name`
optional

optional

string

max length
= 15

Example Values: noradio

`text`
required

required

string

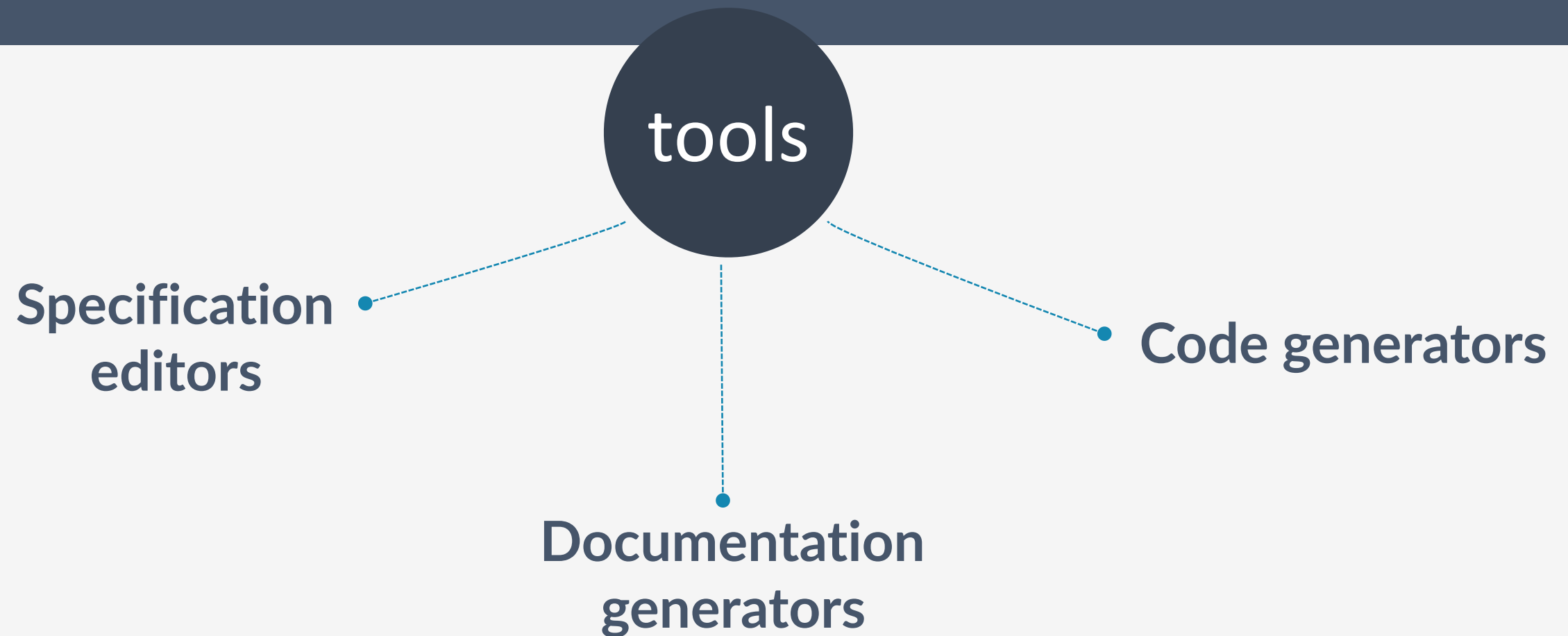
max length
= 280

endpoint)

Example Values:

Meet me behind the cafeteria after school

API specification languages



Web API specifications



Parameters

One of `user_id` or `screen_name` are required.

`user_id`
optional

optional

number

Example Values: 12345

`screen_name`
optional

optional

string

max length
= 15

Example Values: noradio

`text`
required

required

string

max length
= 280

Example Values:

Meet me behind the cafeteria after school

Web API specifications

✘ “Incompatible parameters specified in the request”

✘ “Some co-ordinate parameters were blank”

✘ “You must specify either a list ID or a slug and owner”

Web API specifications



Parameters

One of `user_id` or `screen_name` are required.

`user_id`
optional

optional

number

Example Values: 12345

`screen_name`
optional

optional

string

max length
= 15

Example Values: noradio

`text`
required

required

string

max length
= 140

count limit (available in the [help/configuration](#) endpoint)

Example Values:

Meet me behind the cafeteria after school

Either `user_id`
or
`screen_name`

inter-parameter constraints

Inter-parameter constraints

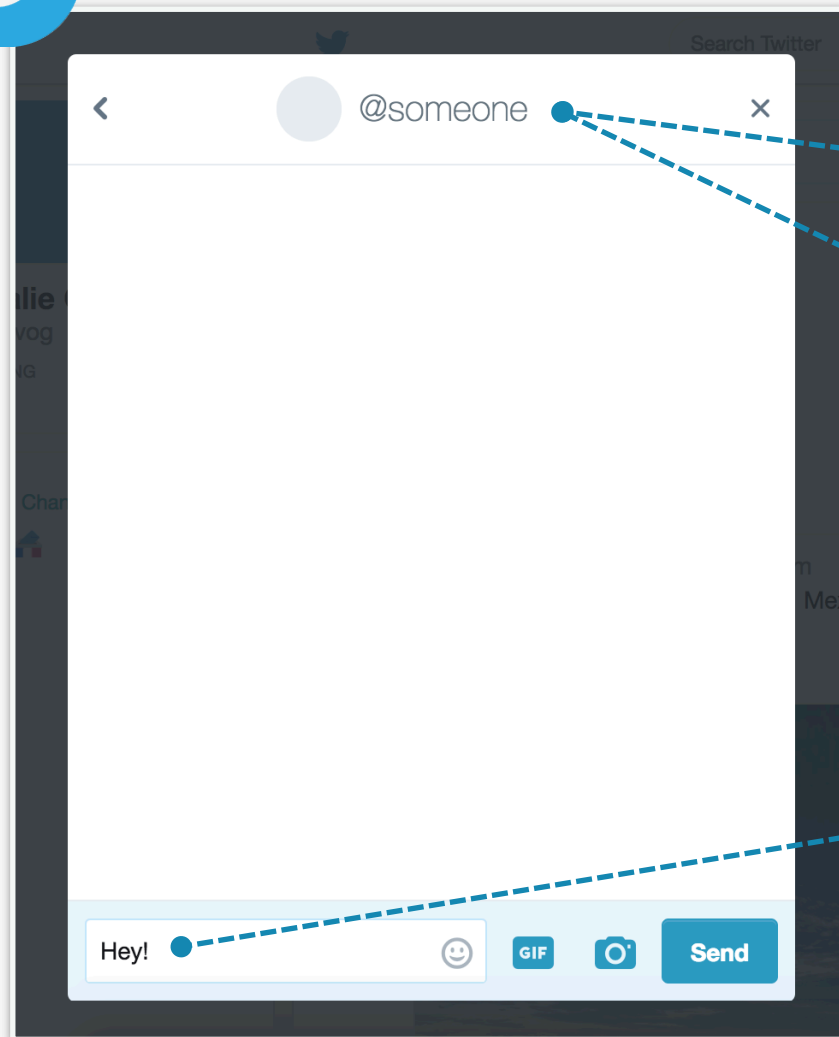
Exclusive
Constraints

Dependent
Constraints

Group
Constraints

Exclusive constraints

➔ exactly one of a set of parameters is required



Parameters

One of `user_id` or `screen_name` are required.

`user_id`
optional

The ID of the user who should receive the direct message. Helpful for disambiguating when a valid user ID is also a valid screen name.

Example Values: 12345

`screen_name`
optional

The screen name of the user who should receive the direct message. Helpful for disambiguating when a valid screen name is also a user ID.

Example Values: noradio

`text`
required

The text of your direct message. Be sure to URL encode as necessary, and keep the message within the character count limit (available in the [help/configuration](#) endpoint)

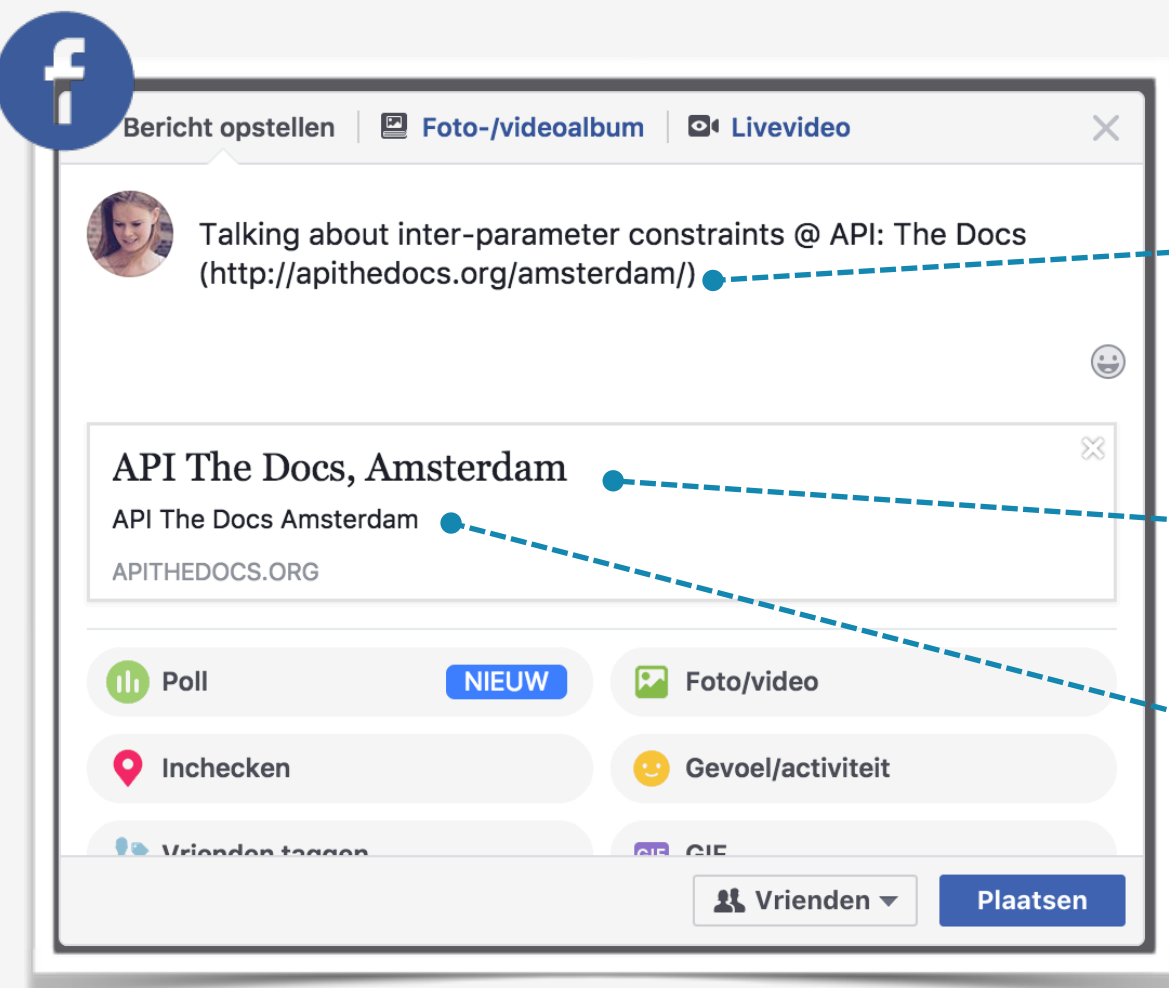
Example Values:

Meet me behind the cafeteria after school

Silent choice: Twitter chooses `screen_name` over `user_id`

Dependent constraints

➔ constraints on a parameter **depend** on a property of another parameter



<code>link</code>	The URL of a link to attach to the post. Either <code>link</code> , <code>place</code> , or <code>message</code> must be supplied. Additional fields associated with <code>link</code> are shown below.
<code>picture</code>	Determines the preview image associated with the link.
<code>name</code>	Overwrites the title of the link preview.
<code>caption</code>	Overwrites the caption under the title in the link preview.
<code>description</code>	Overwrites the description in the link preview

Silent choice: Facebook ignores details of link when link is not provided

Dependent constraints

➔ constraints on a parameter **depend** on a property of another parameter



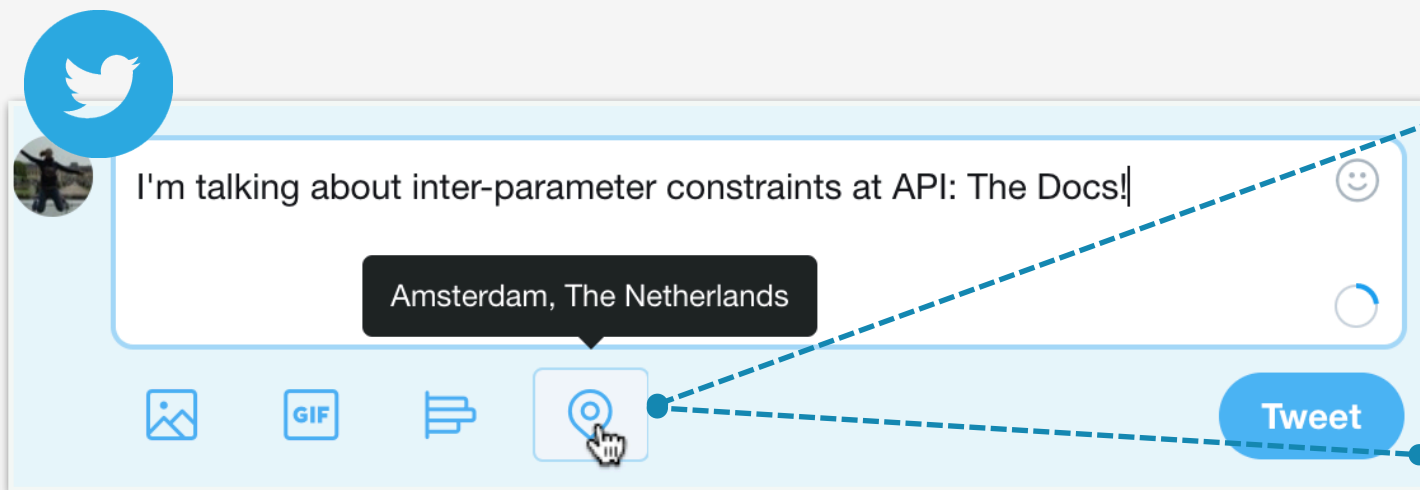
the property **infoWindow** is ignored, when **suppressInfoWindows** is “true”



when searching for an item, **condition** cannot be set to “new” when the **availability** parameter is set to “available”.

Group constraints

➔ a set of parameters should either be all excluded from a request or all included



lat optional	The latitude of the location this tweet refers to. This parameter will be ignored unless it is inside the range -90.0 to +90.0 (North is positive) inclusive. It will also be ignored if there isn't a corresponding long parameter. Example Values: 37.7821120598956
long optional	The longitude of the location this tweet refers to. The valid ranges for longitude is -180.0 to +180.0 (East is positive) inclusive. This parameter will be ignored if outside that range, if it is not a number, if geo_enabled is disabled, or if there not a corresponding lat parameter. Example Values: -122.400612831116

Silent choice: Twitter ignores latitude when longitude is not provided

Nested inter-parameter constraints



“You can identify a list by its **slug** instead of its **list_id**. If you decide to do so, note that you will also have to specify the list owner using the **owner_id** or **owner_screen_name** parameters.”

XOR

XOR

Inter-parameter constraints in web



- ➔ Every kind of inter-parameter constraints occurs at least once in every web API
- ➔ Exclusive constraints in 1/3 entry points in Twitter
- ➔ Exclusive constraints in 1/5 entry points in YouTube

inter-parameter constraints

are common in web APIs

API specification languages



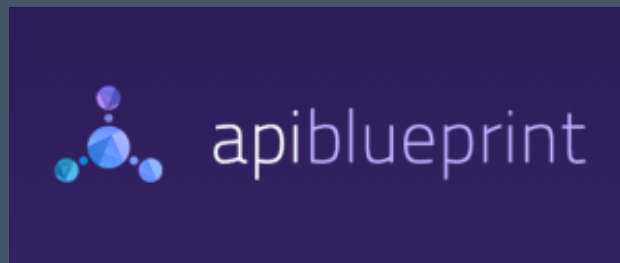
RAML

inter-parameter constraints

are common in web APIs

but cannot be expressed in
specification languages

API specification languages



JSON Schema

Exclusive
Constraints

Dependent
Constraints

Group
Constraints

JSON Schema

Exclusive
Constraints

Dependent
Constraints

Group
Constraints

```
{oneOf : [{ type : "object",  
  properties : {"a" : {type : "string"}}},  
  { type : "object",  
    properties : {"b" : {type : "number"}}}]]};
```

{a:"foo"}

{b:42}

{a:"foo", b:42}

{a:42, b:42}



JSON Schema

Exclusive
Constraints

Dependent
Constraints

Group
Constraints

```
{ type : 'object',  
  properties : {  
    pic : { type : 'image'},  
    pic_name : { type : 'string'},  
    pic_date : { type : 'date'}},  
  dependencies : {  
    'pic_name' : ['pic'],  
    'pic_date' : ['pic']}};
```


JSON Schema

Exclusive
Constraints

Dependent
Constraints

Group
Constraints

```
{ type : 'object',  
  properties: {  
    long : { type : 'number'},  
    lat  : { type : 'number'}},  
  dependencies : {  
    'long' : ['lat'],  
    'lat'  : ['long']}};
```

JSON Schema

Exclusive
Constraints

Dependent
Constraints

Group
Constraints

⊖ too coarse-grained

⊖ no support for
value dependencies

⊖ no one-to-one mapping

inter-parameter constraints

are common in web APIs

but cannot be expressed in
specification languages

A new specification language for web APIs



Parameters

One of `user_id` or `screen_name` are required.

user_id
optional

The ID of the user who should receive the direct message. Helpful for disambiguating when a valid user ID is also a valid screen name.

Example Values: 12345

screen_name
optional

The screen name of the user who should receive the direct message. Helpful for disambiguating when a valid screen name is also a user ID.

Example Values: noradio

text
required

The text of your direct message. Be sure to URL encode as necessary, and keep the message within the character count limit (available in the [help/configuration](#) endpoint)

Example Values:
Meet me behind the cafeteria after school

```
paths:
  /direct_messages/new:
    post:
      parameters:
        - name: user_id
          type: number
        - name: screen_name
          type: string
        - name: text
          type: string
          required: true
      x-constraints:
        - present(screen_name) XOR
          present(user_id)
```

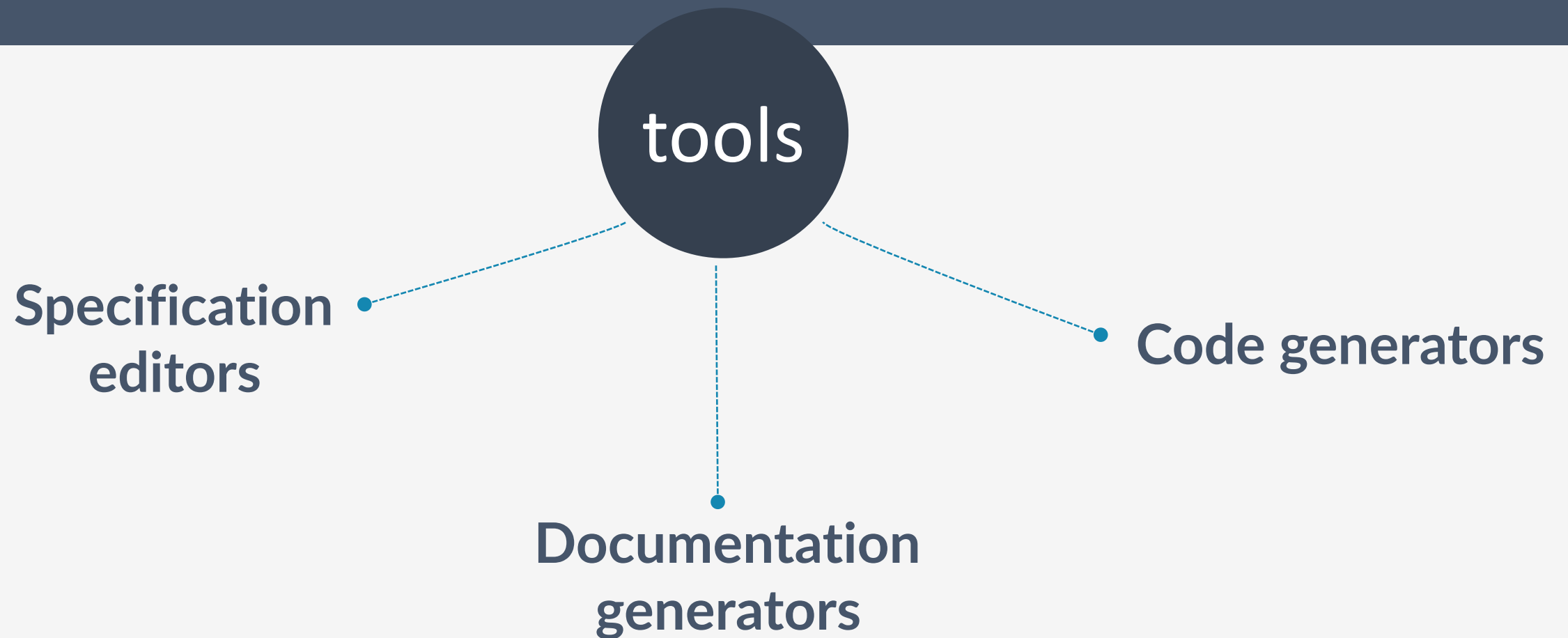
A new specification language for web APIs



```
x-constraint-definitions:  
- minimum(f, v) := value(f) >= v  
- required(f)   := present(f)  
- string?(f)    := type(f) = string  
- number?(f)    := type(f) = number  
  
- xor(f1, f2)   :=  
    present(f1) XOR present(f2)  
  
- pp-dependent(f1, f2) :=  
    present(f1) -> present(f2)  
  
- pv-dependent(f1, f2, v) :=  
    present(f1) -> (value(f2) = v)  
  
- group(f1, f2) :=  
    present(f1) <-> present(f2)
```

```
paths:  
  /direct_messages/new:  
    post:  
      parameters:  
        - name: user_id  
          type: number  
        - name: screen_name  
          type: string  
        - name: text  
          type: string  
          required: true  
  
      x-constraints:  
        - present(screen_name) XOR  
          present(user_id)
```

Integration in languages



Specification editors

➔ provide truth table for inter-parameter constraints



“You can identify a list by its **slug** instead of its **list_id**. If you decide to do so, note that you will also have to specify the list owner using the **owner_id** or **owner_screen_name** parameters.”

```
(list_id XOR slug) AND  
  (slug -> (owner_screen_name XOR owner_id))
```



Truth Table

list_id	slug	owner_screen_n	owner_id
T	T	T	T
T	T	T	F
T	T	F	T
T	T	F	F
T	F	T	T
T	F	T	F
T	F	F	T
T	F	F	F
F	T	T	T
F	T	T	F
F	T	F	T
F	T	F	F
F	F	T	T
F	F	T	F
F	F	F	T
F	F	F	F

Specification editors

➔ provide truth table for inter-parameter constraints



“You can identify a list by its **slug** instead of its **list_id**. If you decide to do so, note that you will also have to specify the list owner using the **owner_id** or **owner_screen_name** parameters.”

```
slug XOR list_id
slug -> (owner_screen_name XOR owner_id)
owner_screen_name -> slug
owner_id -> slug
```



Documentation tools

- ➔ render the inter-parameter constraints as a separate block in the documentation
- ➔ indication that field is part of inter-parameter constraint

Code Generators

user_id
optional

screen_name
optional

text
required

```
public privateMessagePost
  (user_id: number,
   screen_name: string,
   text: string)
  constraining { user_id XOR screen_name }
  { ... }
```

Code Generators

`user_id`
optional

`screen_name`
optional

`text`
required

```
public privateMessagePost  
    (obj: PrivateMessage) { ... }
```

```
interface PrivateMessage{  
    user_id: number;  
    screen_name: string;  
    text: string;  
} constraining {  
    user_id XOR screen_name  
}
```

Constraint-centric interfaces

```
interface PrivateMessage{
  user_id: number;
  screen_name: string;
  text: string;
} constraining {
  user_id XOR screen_name
}
```

1 Assign object to interface

```
let msg1: PrivateMessage = {text: "Hello", user_id: 42};
```

2 Accessing fields

```
function getUserInfo(msg: PrivateMessage) {
  if (msg.user_id !== undefined) {
    msg.user_id;
  } else {
    msg.screen_name;
  }
}
```

inter-parameter constraints

are common in web APIs

but cannot be expressed in
specification languages

➔ extended specification languages
enable better tool support

Spotter guide to inter-parameter constraints:

Exclusive constraint: *either* or *one of*

Dependent constraint: *additional* or *providing*

Group constraint: *corresponding* or *providing*

 <https://github.com/noostvog/Verify-Request>

 @noostvog