

メールアドレス vs 俺たち

Our stand against the challenges posed by E-mail addresses



pixiv Inc.
USAMI Kenta

pixiv

お前誰よ



- うさみけんた (@tadsan) / Zonu.EXE / にゃんだーすわん
- ピクシブ株式会社 pixiv事業本部 Webエンジニアリングチーム PHPer
 - 2012年末から現職でPHPを書いているWebプログラマ
- Emacs PHP Modeを開発しています (2017年-)
- プログラミング言語にちょっとこだわりのある素人 (spcamp2010)
- 2011年頃にPython札幌というコミュニティをやっていました

今回のお題

PHPカンファレンス北海道2024

採択 2024/01/12 17:40～ クリエイティブスタジオ セッション(15分)

メールアドレス vs 俺たち



うさみけんた  tadsan

☆ 2

現代のWebではメールアドレスをIDとしてアカウント登録するサービスが多くあります。

メールアドレスをIDにすることの是非については現代では異論もありますが、なんだかんだといってまだ主流の方法といってよく、Webサービスの生命線と言ってもよいでしょう。

「そんなのフレームワーク使えばすぐ作れるでしょ」

.....そう思って、素朴に作った私たちを待ち受ける運命をいっしょに追体験していきましょう。

話す内容

- メールアドレスの仕様と注意点
- DNS

ID

文A 39の言語版 ▾

ページ ノート

閲覧 ソースを編集 履歴表示 ☆

出典: フリー百科事典『ウィキペディア (Wikipedia)』

W ウィキペディアのページIDを知るには、左のツールバーから「ページ情報」をクリックしてください。IDで検索するには、[特別:転送](#)をご利用ください。

ID, Id, id, I.D., 



ウィクショナリーに関連の辞書項目があります。
ID

符号など [\[ソースを編集\]](#)

- 個体識別、利用者識別、およびそのための符号 (Identification) 。
 - 身分証明および身分証明書 (Identity Document) 。
 - 識別子。
 - システムのアカウント。アイデンティティ管理 (Identity)
 - [2ちゃんねる](#)などの電子掲示板で接続回線の推定に使われるものについては「[2ちゃんねる#ID](#)」を参照。
- 参考文献・脚注で同じ文献からの引用を表す語 (**idem**) 。 → [Ibid](#)
- インドネシアの航空会社、バティック・エアのIATA航空会社コード。
- 鉄道のサインシステムにおいて、伊豆箱根鉄道大雄山線 (IzuhakoneDaiyūzan) の路線記号として用いられる。

出典: Wikipedia: ID (2023-07-09T09:38:16版)

デジタルアイデンティティ

文A 12の言語版 ▾

ページ ノート

閲覧 ソースを編集 履歴表示 ☆

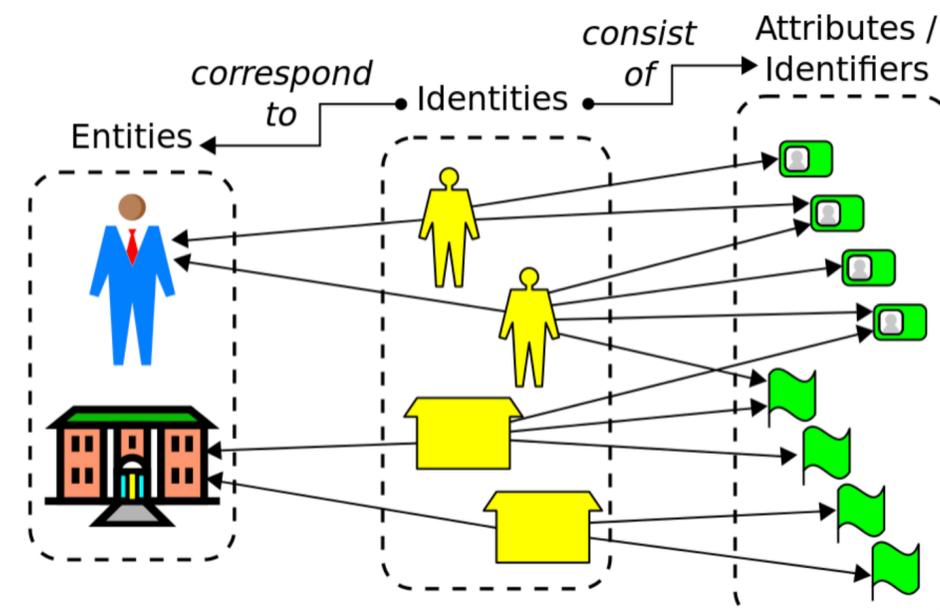
出典: フリー百科事典『ウィキペディア (Wikipedia) 』

デジタルアイデンティティ (英: Digital identity) は、人間などの主体 (entity/subject) をコンピュータで処理するためのアイデンティティ情報であり、それぞれの属性情報から成る。

ISO/IEC 24760-1は、アイデンティティを「実体に関する属性情報の集合 (set of attributes related to an entity) 」と定義している [1]。

定義 [\[ソースを編集\]](#)

人、組織、デバイス、サービスなど、属性を管理する単位のことを**エンティティ (主体)** といい [IPA2013\(p5\)](#)、システムに登録されたエンティティに関する属性情報の集合を**アイデンティティ情報** とい [う IPA2013\(p5\)](#)。アイデンティティを電子的に表現したものを**デジタルアイデンティティ** とい [う IPA2013\(p5\)](#)。



出典: Wikipedia: [デジタルアイデンティティ](#) (2023-10-20T14:01:45版)

今回はIDとは何か
(本質)
の話には深入りしません

おことわり

発表者は現時点において
現職のID設計判断や実装に
関わっていません！

ID、Auth、アカウント基盤など
AAA (Authentication,
Authorization, Accounting)
は高度な専門技術領域

発表のねらい

今回のゴール

- ただメールアドレスを扱うだけのよくある仕組みだとしても、考慮事項がいくつもあってシンプルな実装では済まない(こともある)こと
- これらの考慮事項には唯一の正解はなく、ビジネス要件やユーザーのニーズに応じて取捨選択する必要があるということを理解し、必要十分なものを設計・実装できること

ゴールではないこと

- メールアドレスに関係しないアカウント設計の考慮事項
- こう実装すれば間違いないという指針を提示すること
- デジタルアイデンティティを完全理解してIDエキスパートになること
- オブジェクト指向設計についてマスターすること
 - 今回提示することをそのまま実装するとユニットテストしにくい構造になる場合があるので、適切に依存を分割・注入することは読者への課題とする

今回は専門職ではない
我々が実装するための
お話をします

初心者の方も
怖がらずに
お聞きください

さて

難しいことは置いておいて
ユーザーからメールアドレスを
収集したい(せざるを得ない)
ことは現代でも多い

メールアドレスを何に使うか

- マーケティング（メールによる告知など）
- Webサービスのログイン機能のIDとして
 - Laravelのようなフレームワークは基本機能として持っていて、スキャットフォールド(コードの自動生成)で最低限のものは簡単に作れる(生成されるので実装の必要すらない)
- ログイン通知（心あたりのないログインでないかユーザ自身に知らせる）

メールアドレスを何に使うか

- アカウント登録させてマーケティングに活用…
 - 利用規約に同意させた上なら問題はないが、やりすぎは嫌われるので要注意
 - メールマガジンなど送りすぎてフィルタリングされてしまっっては、本当に大事な連絡が届かなくなるのはよくない
 - 個人的な意見では、私信のような体裁や「あなただけ特別に…」のような文言のキャンペーンメールは非常にストレスが大きい

個人的感情は
おいといて

DBに入れるだけだし
フレームワークとか要らん

ですよね？ [要出典]

すご〜くシンプルな
ログインシステムを
作っていきましょう

[要件1]
変な形式の文字列が
登録できないこと

aaa@example.com

example.com

生成 AI は試験運用中です。 [詳細](#) ⋮

PHPでメールアドレスを検証するには、`filter_var`関数を使用します。この関数の2つ目の引数として`FILTER_VALIDATE_EMAIL`を指定すると、メールアドレスの検証を行うことができます。 [▼](#)

たとえば、次のコードでは、`$email_a`がメールアドレスとして有効かどうかを判定しています。 [▼](#)

[もっと見る](#) ▼

PHPには `filter_var` 関数が用意されています。この関数の2つ目の引数として `FILTER_VALIDATE_EMAIL` を指定するとメールアドレスの検証が行えます。 2022/02/16

Customers Mail Cloud
<https://blog.smtps.jp/entry/2022/02/16> ⋮

各プログラミング言語別のメールアドレス検証方法

[? 強調スニペットについて](#) • [! フィードバック](#)

PHP
<https://www.php.net/filter.examples.validation.php> ⋮

検証 - Manual

ググったたら最適解が
出てくる良い時代…

良くなかった時代とは

404 Blog Not Found

The requested blog was not found -- unless you requested that of Dan Kogai (小飼 弾).

< 「小飼弾の「仕組み」進化論」、本日発売+重版決定 | 警告は当たった、脱出はなるか - 書評 - 世界大不況からの脱出 >

2009年03月19日 18:00  [Lightweight Languages](#)

「PHP使いはもう正規表現をblogに書くな」と言わせないでくれ

ああ、まただよ...

- [かなり使えるPHPの正規表現まとめ - IDEA*IDEA ~ 百式管理人のライフハックブログ ~](#)

正規表現って、プログラミング言語間の差が少ないサブ言語なのに、なぜ「PHP」がつくとダメ正規表現ばかり登場するのか。うんざりだ。

[かなり使えるPHPの正規表現まとめ - IDEA*IDEA ~ 百式管理人のライフハックブログ ~](#)

メールアドレスは厳密にチェックしようとするとなかなか難しいのですが、簡単なチェックだったらこれでOKぽいですね。

```
/^[^0-9][a-zA-Z0-9_]+([\.[a-zA-Z0-9_]+)*@[a-zA-Z0-9_]+([\.[a-zA-Z0-9_]+)*[a-zA-Z]{2,4}$/
```

全然OKじゃない。

```
var re = /^[^0-9][a-zA-Z0-9_]+([\.[a-zA-Z0-9_]+)*@[a-zA-Z0-9_]+([\.[a-zA-Z0-9_]+)*[a-zA-Z]{2,4}$/;
alert(re.test('ばka@manuke.com')); /* これ、どうなる? */
```

Run

最初の`[^0-9]`にしてからもうだめ。

メールアドレスの正規表現が一筋縄では行かないことは、ちょっとぐぐればすぐに出てくるし、しかも正規表現でマッチできるのはたかだか`nobody@example.com`、すなわち`addr-spec`まで、From: `Anonymous <nobody@example.com>`、すなわち`mailbox`を正規表現でマッチするのは不可能なのはもう10年以上前から知られている(Perl 5.6以降の拡張正規表現であれば可能だが、それだともう狭義の正規表現からは逸脱してしまう)。

[Perlメモ](#)

Jeffrey E. F. Friedl氏原著による「詳説 正規表現」にはメールアドレスはネストしたコメントを持つことができるので正規表現で表わすのは不可能であると書いてあります

で、`nobody@example.com`、すなわち`addr-spec`のみにマッチする正規表現は、以下のとおりとなる。

全然OKじゃない。

```
var re = /^[^0-9][a-zA-Z0-9_]+([\.[a-zA-Z0-9_]+)*@[a-zA-Z0-9_]+([\.[a-zA-Z0-9_]+)*[.][a-zA-Z]{2,4}$/;  
alert(re.test('ばka@manuke.com')); /* これ、どうなる? */
```

Run

最初の`[^0-9]`にしてからもうだめ。

メールアドレスの正規表現が一筋縄では行かないことは、ちょっとぐぐればすぐに出てくるし、しかも正規表現で
まで、From: `Anonymous <nobody@example.com>`、すなわちmailboxを正規表現でマッチするのは不可能なのはもう100
可能だが、それだともう狭義の正規表現からは逸脱してしまう)。

[Perlメモ](#)

Jeffrey E. F. Friedl氏原著による「[詳説 正規表現](#)」にはメールアドレスはネストしたコメントを持つことができ

で、`nobody@example.com`、すなわちaddr-specのみにマッチする正規表現は、以下のとおりとなる。

どうということ？

そもそもそもそも メールアドレスとは

tadsan@pixiv.com

"tadsan"@pixiv.com

<tadsan@pixiv.com>

USAMI Kenta <tadsan@pixiv.com>

どれも「メールアドレス」

電子メールでの使用形式 [\[ソースを編集\]](#)

それぞれの使用箇所で使用できるメールアドレスの形式を、以下の表に示す。

メールアドレスの使用形式一覧

形式	RFC 5321 ☞ での呼称	使用箇所 (エンベロープ)	RFC 5322 ☞ での呼称			使用箇所 (ヘッダ) [1]
<code>foo@example.com</code>	Mailbox	-		mailbox	addr-spec	From Sender
<code><foo@example.com></code>	-	-	address		name-addr	Reply-To To Cc Bcc
<code>foo <foo@example.com></code>	-	-		group		Reply-To To Cc Bcc
<code>foobar: foo <foo@example.com>, bar <bar@example.com>;</code>	-	-				Reply-To To Cc Bcc
<code><foo@example.com></code>	Path	reverse-path forward-path	path			Return-Path

出典: Wikipedia: [メールアドレス](#) (2023-11-07T03:00:31版)

メールアドレスとは

- 特に断りがなければ `tadsan@pixiv.com` のような文字列を指す
 - RFC 5322では`addr-spec`と呼ばれるもの
- **USAMI Kenta** `<tadsan@pixiv.com>` のような、名前や `<>` を含んだものもメールアドレス
- 先述のブログで「正規表現でマッチできない」と言われているのはこれを含めたメールアドレスのこと

電子メールでの使用形式 [\[ソースを編集\]](#)

それぞれの使用箇所で使用できるメールアドレスの形式を、以下の表に示す。

メールアドレスの使用形式一覧

形式	RFC 5321 での呼称	使用箇所 (エンベロープ)	RFC 5322 での呼称			使用箇所 (ヘッダ) ^[1]
<code>foo@example.com</code>	Mailbox	-		mailbox	addr-spec	From Sender
<code><foo@example.com></code>	-	-			name-addr	Reply-To To Cc Bcc
<code>foo <foo@example.com></code>	-	-	address		addr	
<code>foobar: foo <foo@example.com>, bar <bar@example.com>;</code>	-	-		group		Reply-To To Cc Bcc
<code><foo@example.com></code>	Path	reverse-path forward-path	path			Return-Path

出典: Wikipedia: [メールアドレス](#) (2023-11-07T03:00:31版)

一般的なメールアドレス
を収集する用途で
addr-spec以外が
必要になることはない

基本的にaddr-spec
だけを相手にすれば
いいとわかった

改めて

ローカルパート

ドメイン名

tadsan@pixiv.com

ローカルパートの制約

- dot-atom形式 (いわゆる普通のメールアドレス)
 - ASCIIの英数文字 [a-zA-Z0-9]
 - ASCIIの記号 ! # \$ % & ' * + - / = ? ^ _ ` { | } ~
 - . (ドット)は先頭と末尾以外に使用できるが、2個以上連続できない
- 大文字小文字は区別する… が… 現実には区別せずに扱うサービスも
 - むしろ多数

[ホーム](#) > [サービス・機能](#) > [条件からサービスを探す](#) > [ドコモメール](#) > 特殊な形式のアドレス（RFC違反アドレス）のご利用について

特殊な形式のアドレス（RFC違反アドレス）のご利用について

2009年以前よりドコモのメールサービス（@docomo.ne.jp）をご利用いただいているお客さまのなかには、RFC違反アドレス※1を利用している可能性があります。

RFC違反アドレスに関しては現在ご利用が推奨されていないため、今後メールサービスでの障害が発生する場合があります。ドコモとして、メールアドレス変更をおすすめいたします。

※1 RFC（Request for Comments）違反アドレスとはドコモアドレスのうちに2連続のドット「..」が使用されていたり、@マークの直前でドット「.@」を設定しているアドレスを指します。

（例）docomo..ab1234yz@docomo.ne.jp、docomo-ab1234yz.@docomo.ne.jp

RFC違反アドレスで発生する事象について

RFC違反アドレスを継続してご利用いただくと以下のような事象が発生する可能性があります。

- ・ 他キャリア/他サービスアカウントへのメールの送受信ができなくなる
- ・ 企業やサイトへの会員登録時にメールアドレスがアドレスと判定されず登録ができない
- ・ 登録しているメールアドレスからメールが届かなくなる

RFC違反メールアドレス

- 2009年頃まで一部の携帯キャリアで発行できていたメールアドレス
 - `good..by..example.com` のようなアドレスが発行できた
- 仕様違反のメールアドレスでも動作するサービスが存在していたが…
 - 2020年にiOS 14、2021年にSendGridで利用不可能に
- quoted-string形式 `"good..by.."@example.com` に変換すれば合法だが、残念なことに受信側のメールサービスが対応している必要がある

Change language: ▼[Submit a Pull Request](#) [Report a Bug](#)

filter_var

(PHP 5 >= 5.2.0, PHP 7, PHP 8)

filter_var – 指定したフィルタでデータをフィルタリングする

説明

```
filter_var(mixed $value, int $filter = FILTER_DEFAULT, array|int $options = 0): mixed
```

FILTER_VALIDATE_EMAIL	"validate_email"	default	FILTER_FLAG_EMAIL_UNICODE, FILTER_NULL_ON_FAILURE	<p>値が妥当な e-mail アドレスであるかどうかを検証します。</p> <p>この検証は、e-mail アドレスが RFC 822 の addr-spec 形式に沿ったものであるかどうかを確認めます。ただし、コメント、空白の折り返し (whitespace folding) およびドットなしドメイン名 (dotless domain name) には対応していません。</p>
------------------------------	------------------	---------	--	--

```
$email = filter_var($_POST['email'] ?? null, FILTER_VALIDATE_EMAIL);  
if ($email === false) {  
    throw new RuntimeException('メールアドレスが不正です');  
}
```

PHPでは
これを使えば
安全！

…ここまでに説明した
範囲では

[要件2]

存在しないドメインの
メールアドレス登録を
弾きたい

a@b.c

```
filter_var('a@b.c',  
FILTER_VALIDATE_EMAIL)
```

```
filter_var('a@b.c',  
FILTER_VALIDATE_EMAIL)  
// => "a@b.c"
```

定石：
入力されたアドレスに
送信してメール内の
コードを入力させる

実メール送る前に
どうしても確認したい
のだとしまししょう

```
// 超だめな判定
if (!preg_match('/\.(com|net|co.jp|ne.jp)/', $domain)) {
    throw new RuntimeException('メールアドレスが不正です');
}
```

DNSを手チェックしよう



メールアドレス DNS 判定 PHP



すべて

画像

書籍

動画

ショッピング

もっと見る

ツール

約 182,000 件 (0.28 秒)



合同会社テラレン

<https://matsu.teraren.com> › Program & Service › PHP

メールアドレスのドメイン部分をチェック

2010/05/28 — ユーザが入力したメアドの検証をするために、アットマーク以降のドメイン部分を検証する。 <?php /** * Validate the mail address ** @author ...



オフィスオバタ

<https://office-obata.com> › ブログ TOP › 忘備録 › PHP

PHPでドメイン存在確認チェックをする

2022/04/05 — **DNS**が文字列であるドメイン(office-obata.comなど) をIPアドレス (192.999.999.999 といった数字の羅列) に変換しているのです。つまり存在するドメイン ...

この方法で
良いといえれば良い
だめといえればだめ

(非常に言葉を濁す)

メールはどうやって届くのか

- 白ヤギさんと黒ヤギさんが… という話から始めると15分では無理
- メールに関係するDNSレコードにはAとMXがある
- メール配送の仕組みとしてはAレコードがあればいいのだが、今日の多くはメールアドレスのドメイン部分に対応したMXレコードがある
- Aレコードしかないメールアドレスは個人や小規模なメールサーバの独自運用されている傾向にあるのでAレコードもサポートするかは各自判断

DNSではなくSMTP
でチェックする別解も
あるが...

(僕は運用経験がない
ので言葉を濁す)

ついでに
メールアドレス

IPアドレス

a@0.0.0.0

```
filter_var('a@0.0.0.0',  
FILTER_VALIDATE_EMAIL)
```

```
filter_var('a@0.0.0.0',  
FILTER_VALIDATE_EMAIL)  
// => "a@0.0.0.0"
```

ドメインがIP

[要件3]
メールアドレスの
重複登録を弾きたい！

DBに登録

```
return new class extends Migration
{
    public function up(): void
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });
    }
    public function down(): void
    {
        Schema::dropIfExists('users');
    }
};
```

ユニークキー制約

大勝利

かと思いましたが？

ローカルパートの制約

- dot-atom形式 (いわゆる普通のメールアドレス)
 - ASCIIの英数文字 [a-zA-Z0-9]
 - ASCIIの記号 ! # \$ % & ' * + - / = ? ^ _ ` { | } ~
 - . (ドット)は先頭と末尾以外に使用できるが、2個以上連続できない
 - 大文字小文字は区別する… が… 現実には区別せずに扱うサービスも
- むしろ多数

a@example.com

A@example.com

同一かもしれないし
別人かもしれない

テーブル名: users Partitio

エンジン: InnoDB

オートインクリメント: 2

文字セット: utf8mb4

照合: utf8mb4_unicode_ci

説明:

列名	#	データ・タイプ
id	1	bigint unsigned
name	2	varchar(255)
email	3	varchar(255)
email_verified_at	4	timestamp
password	5	varchar(255)
remember_token	6	varchar(100)
created_at	7	timestamp
updated_at	8	timestamp

unicode_ciは
大文字小文字を
同一視する

utf8mb4_binで
厳密チェックすれば
いいのかな？

もっとひどい穴を
簡単に作れるので
binで比較はやめた
方がいいと思う

Laravelは全部小文字の
アドレスのみ許可
(これが無難)

この問題を騙り尽すには
15分枠はあまりに短い…

Gmailエイリアス沼

- 同じメールアカウントに対応するアドレスはたくさんある
 - fafnir127@gmail.com, fafnir127+alias@gmail.com
 - f.a.f.n.i.r.1.2.7@gmail.com, fafnir127@googlemail.com
- これらはGoogle Workspaceを使った独自ドメインのメールでも有効



🔊 記法システムのメンテナンスを行いました ×



みんなでつくるpixivの百科事典

ピクシブ百科事典はアニメやマンガ、ゲームからデザイン・アートまであるゆる言葉・現象・文化・作品を解説するみんなでつくる百科事典です。

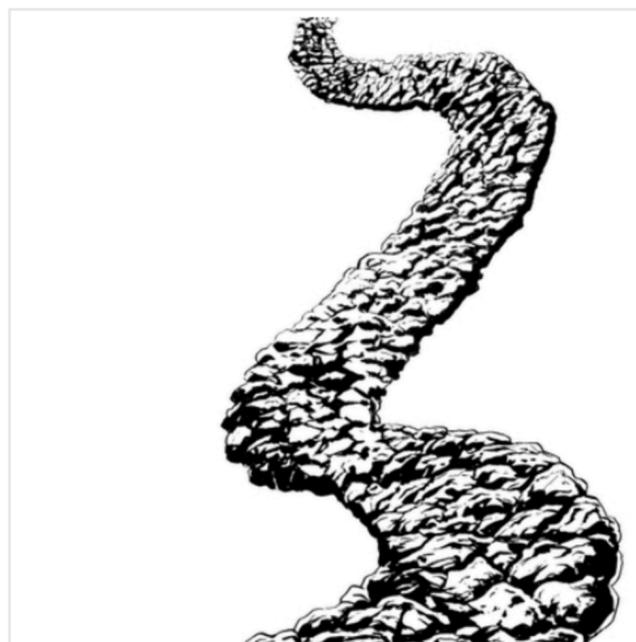
✕ Post

👍 ええやん! 1



[アニメ](#) | [マンガ](#) | [ラノベ](#) | [ゲーム](#) | [フィギュア](#) | [音楽](#) | [アート](#) | [デザイン](#) | [一般](#) | [人物](#) | [キャラクター](#) | [セリフ](#) | [イベント](#) | [同人サークル](#)

[ピクシブ百科事典](#) > [一般](#) > [社会](#) > [経済](#) > [働く](#) > [職業](#) > [漫画家](#) > [車田正美](#) > [男坂](#)



男坂

おとこざか

車田正美の硬派ケンカ漫画。打ち切りにより未完に終わるが、それによって逆に伝説となった。……がその後数十年ぶりに連載が再開され2023年に完結した。

✕ Post

[pixivで「男坂」のイラストを見る](#)

[pixivで「男坂」の小説を読む](#)

[pixivで「男坂」のイラストを投稿する](#)

[pixivで「男坂」の小説を投稿する](#)

ご静聴ありがとうございました