



mixiボイスTwitter連携 開発の取り組みについて

株式会社ミクシィ コンテンツ部

森垣 潤一

自己紹介

○名前

- 森垣 潤一

○所属

- 株式会社 ミクシィ コンテンツ部

○やっていること

- mixiアプリ関連の裏方

アジェンダ

○ はじめに

- mixiボイスTwitter連携の機能について

○ 実装、及びシステム構成について

- mixiボイス ← Twitter 連携のシステム構成
- mixiボイス → Twitter 連携のシステム構成
- その他

○ 個人的な今後の勝手な展望(希望)

○ まとめ

はじめに

- 何について話すか
- 既に多数のユーザーが使っているWebアプリケーションサービス上で、外部と連携する新サービスを作ることについて
 - 何が起こるのか
 - 何を考えているのか
 - どうするのか
- mixiボイスTwitter連携を一事例として紹介しつつ



「mixiボイスTwitter連携」について

○ そもそも mixiボイスTwitter連携 って？

○ mixiボイスへTwitterのタイムラインを混ぜたり、mixiボイスへのつぶやきをTwitterへ反映させたりする機能

- mixiボイス ← Twitter 連携(read) は 2010.06 から
- mixiボイス → Twitter 連携(write) は 2010.12 から

○ 普通にTwitterAPIを叩いて動作

- http://api.twitter.com/1/statuses/user_timeline.json
- <http://api.twitter.com/1/statuses/update.json>

○ 現在は mixiページ でも連携

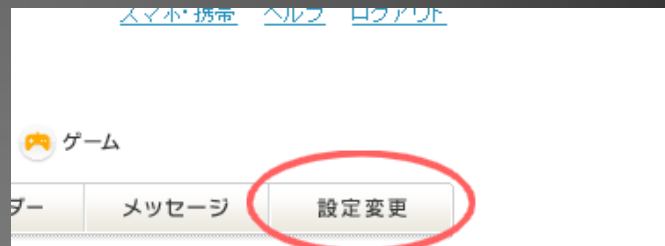
○ 取り込まれたつぶやきは普通のmixiボイスと同じように扱われる

- @ RT 付きは取り込まない

m 「mixiボイスTwitter連携」について

○ 連携方法

○ 上部のメニューバーの[設定変更]



○ 外部サービス連携 の [twitter連携]



○ でたどりつけるページで設定できます。





実装、及びシステム構成について



Twitterクライアントアプリケーションを作ろう

- 普通のネイティブクライアントアプリケーションを簡単に作る場合
 - 簡単に同期的な動作をすると考えます
- 取得
 - 制限に引っかからない程度で頻繁にリクエストを投げる
 - 新しく取得ができたならタイムライン表示を更新
- 書込
 - 送信ボタンを押したらリクエストを投げる
 - レスポンスを受け取ったら結果を表示



Twitterクライアントアプリケーションを作ろう

- 多数のユーザーが使うクライアントWebアプリケーション(mixi等)で同じ事をした場合
- 取得
 - 制限に引っかからない程度に頻繁にリクエストを投げる × ユーザー数
 - さくっと秒間数万リクエストとかいく
 - 恐らく怒られる
 - ちょっとでもレスポンスが遅延するとすべてのユーザーの取得が止まる
 - 取得ができればタイムラインを更新 × ユーザー数
 - タイムライン(mixiボイス)はTwitter以外からのつぶやきで常に更新されている
 - お互いが同時に書込を行うと、競合してしまう
 - かなり怒られる



Twitterクライアントアプリケーションを作ろう

- 多数のユーザーが使うクライアントWebアプリケーション(mixi等)で同じ事をした場合
- 書込
 - 送信ボタンを押したらリクエストを投げる × ユーザー数
 - TwitterAPIが止まってしまうと、失敗し続ける
 - 一度投稿が失敗すると、投稿内容が消える
 - 下書き等で残しても、再度投稿する必要がある
 - レスポンスを受け取って、結果を表示
 - TwitterAPIが遅延すると、画面が固まる
 - 待った結果、失敗していると悲しい



mixiボイス ← Twitter 連携のシステム構成(read)

○ どうしたいか

- 連携ユーザー数がどんなに増えても対応できるようにしたい(スケール)
- 既存サービスに影響を出さずに反映させたい(競合)
- TwitterAPI が遅延したり、止まってしまっても他に影響を出さず、復活したら自動的に取ってこられるようにしたい(くじら)

○ どうするか

- 並列化でスケール対応
 - 複数 kicker & worker
- 非同期化で遅延対応
 - ジョブキュー

○ 以下の構成要素があります

○ manager

- kickerの管理

○ kicker

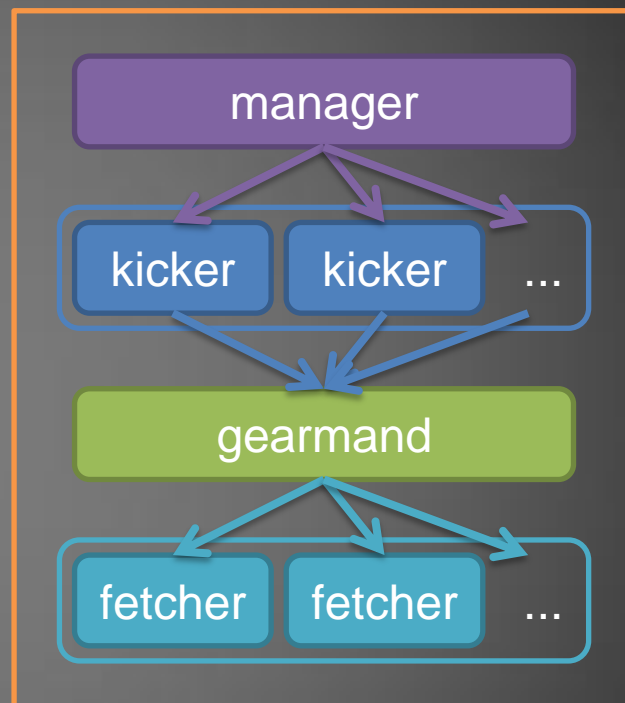
- クロール対象の決定

○ gearmand

- ジョブキュー

○ fetcher

- クロール処理



mixiボイス ← Twitter 連携(read)のシステム構成

○ manager



manager

- 複数の kicker プロセスを起動し、動作を監視する
- 定期的に kicker の数を確認し、一定数の kicker が常時稼働するようにする

○ kicker



kicker



kicker



...

- 並行でたくさん起動されていることが前提
- 以下の処理をする
 - 対象とするユーザーを一意に決める(他のkickerの範囲と被らないように)
 - 対象とするユーザーを一定数ごとに分けて、ジョブを gearmand に投げる

mixiボイス ← Twitter 連携(read)のシステム構成

- gearmand(<http://gearman.org/>)



gearmand

- kicker に投げられたジョブを fetcher へディスパッチしていく

- fetcher



fetcher



fetcher



...

- 常時複数稼動している前提

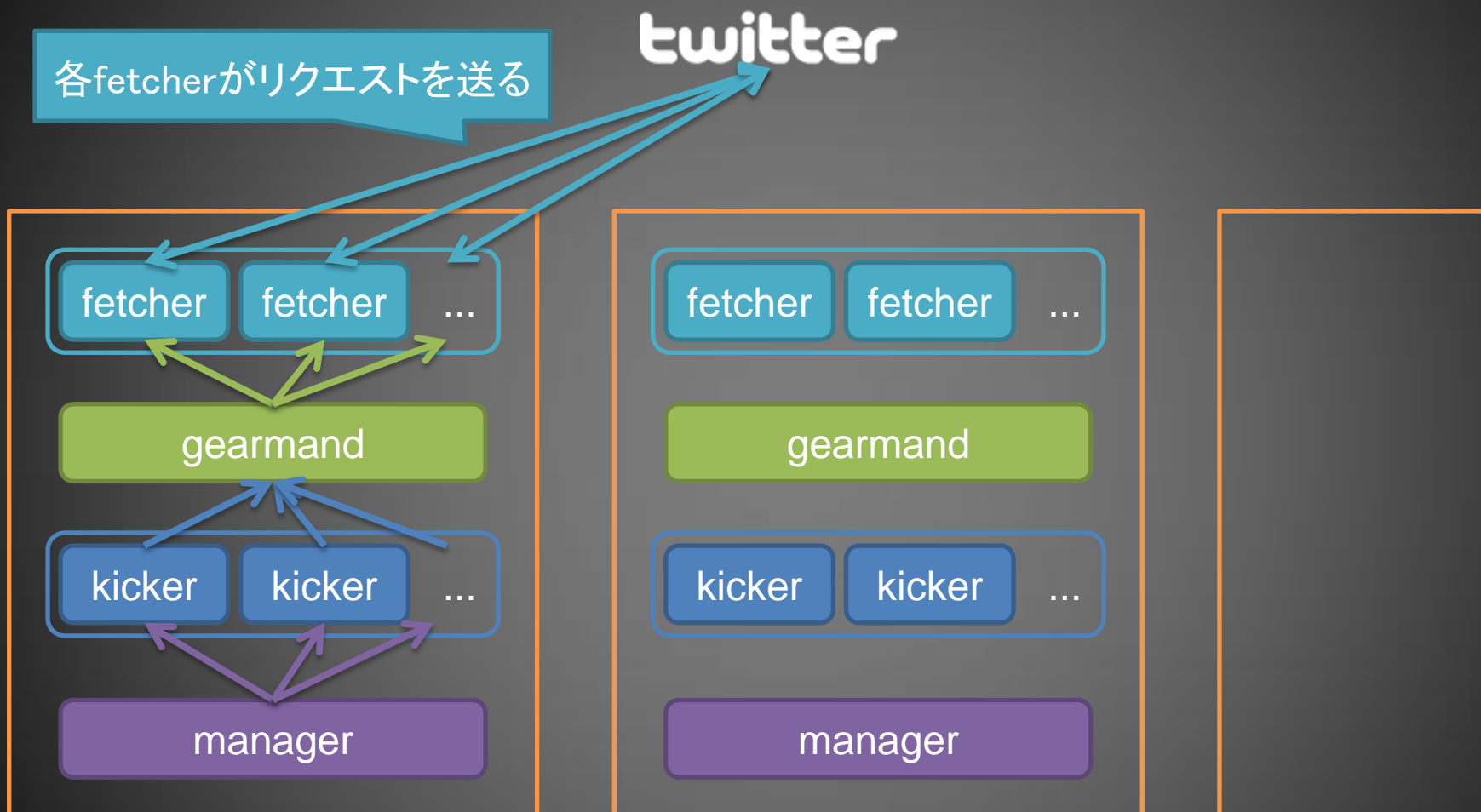
- gearmand からジョブを受け取って以下を処理する

- kicker で決定した対象とするユーザー毎に TwitterAPI へリクエストを送る
- 取得できたら、ボイスに書きこむ(ジョブを投げる)
- 次に更新する時間を決めてDBへ書き込む

- 各リクエストが全て終わったら、次のジョブを受け取って繰り返し処理する

- 一定時間後にジョブを受け取るのをやめて終了する

m mixiボイス ← Twitter 連携(read)のシステム構成



○ 何が改善されるか

- 特定のユーザーの更新が異常に多くても、その他のユーザーに遅延が発生しない
 - 各プロセスが並行に複数非同期に動いているため
- ユーザー数が増えてもクローल用のマシンを増やしたり、スペックを上げることでスケール出来る
- Twitterが落ちるなどして取得ができない期間が発生しても、復旧してから再度取りにしてくれる

○ ネイティブアプリケーションと比較して辛いところ

- 全連携ユーザーを定期的に見に行くため、遅延が発生する

mixiボイス ← Twitter 連携(read)のシステム構成

- その他
- 既存サービス(mixiボイス)との競合
 - 既存サービス側でジョブキューを用意して解決
 - 時系列は取得したtimestampを見て解決



mixiボイス -> Twitter 連携のシステム構成(write)

mixiボイス → Twitter 連携(write)のシステム構成

○ どうしたいか

- Twitterへのリクエストに失敗しても、投稿が消滅しないようにしたい
- 大量の投稿が発生した場合でも、TwitterAPIへのリクエスト頻度を平滑化したい
- 遅延 & 失敗しても、画面が固まったりしないようにしたい

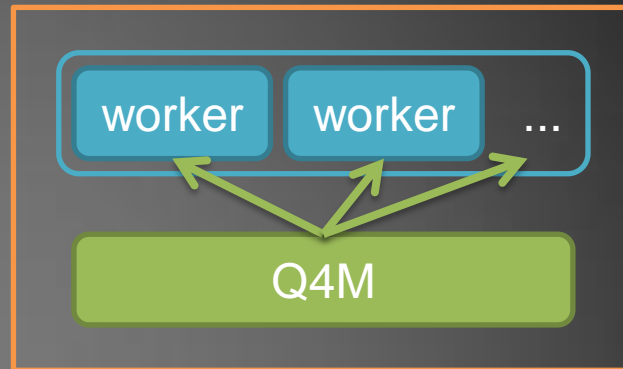
○ どうするか(readとやることは同じ)

- 並列化でスケールして対応
 - 複数worker
- 非同期化で遅延&失敗に対応
 - Q4M
 - <http://q4m.github.com/>

mixiボイス → Twitter 連携(write)のシステム構成

○ 以下の構成要素があります

- Q4M
- worker スクリプト



○ この構成は、mixiの他サービスでも用いています。

- ボイス
- 日記
- etc..

○ Q4M

Q4M

○ MySQLで動作する、プラグブルストレージエンジン

- <http://q4m.github.com/>
- mixiではボイスや日記などの投稿負荷対策に用いています
 - <http://alpha.mixi.co.jp/blog/?cat=19>
 - 他のサービスに影響を与えないよう、別マシンで動かしています

○ ユーザーが投稿したタイミングで登録されたジョブをworkerへディスパッチするのみ

- ジョブに失敗したら、再度キューイングして後に回す
 - 失敗し続けたら、ゆっくりやる

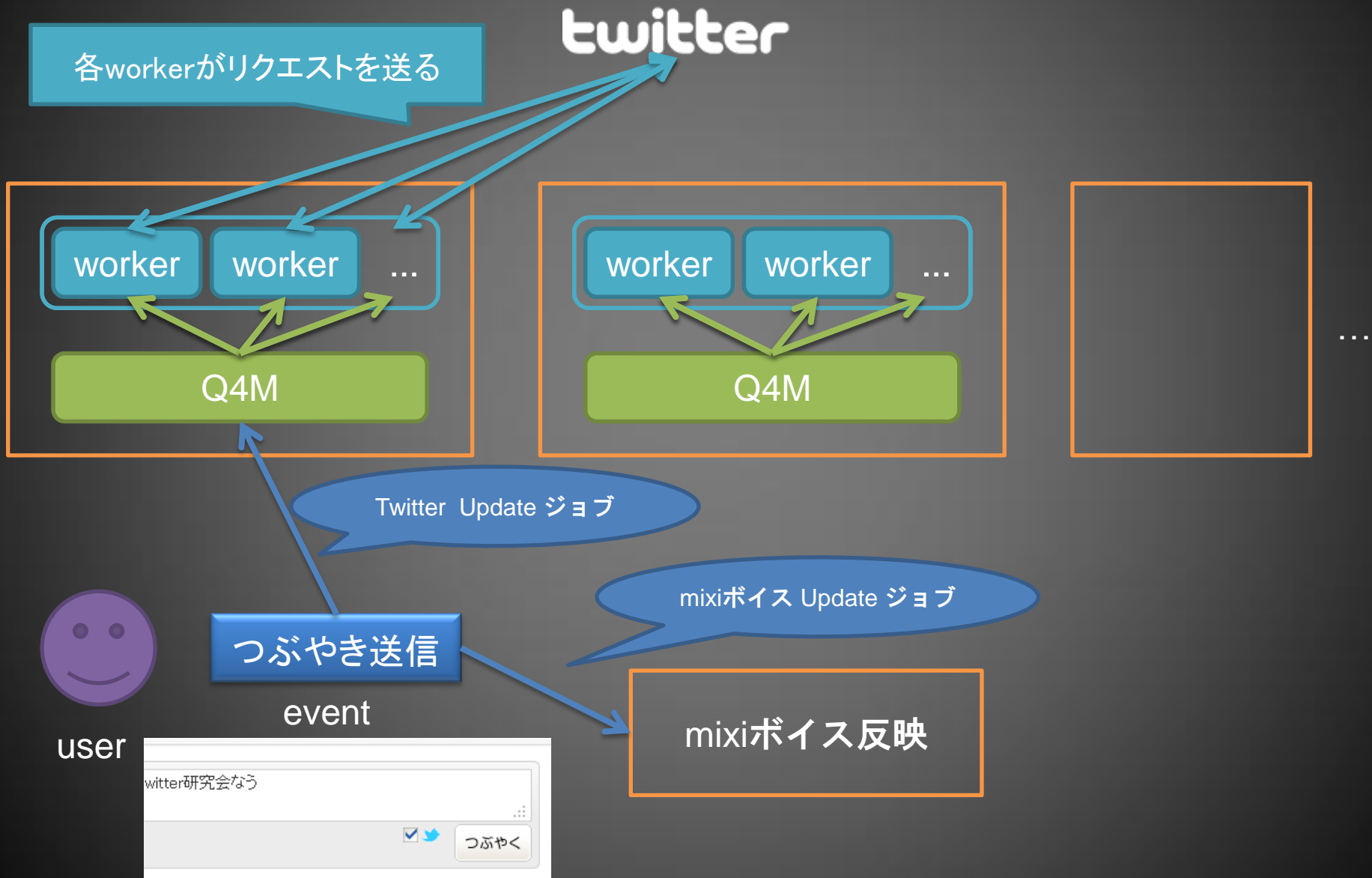
mixiボイス → Twitter 連携(write)のシステム構成

○ worker



- Q4Mにキューイングされているジョブを受け取って処理を行う
- TwitterAPIを叩いて更新していただく
 - Twitterから読み込んだボイスではないか
 - ユーザーが書込連携をしているか
 - ユーザーのTwitterアカウントへの書き込み権限を持っているか

m mixiボイス → Twitter 連携(write)のシステム構成



mixiボイス → Twitter 連携(write)のシステム構成

○ 何が改善されるか

- TwitterAPIへのリクエストに失敗しても、ジョブは残るので繰り返し施行される
- workerの処理スピードで、リクエスト数を平滑化できる
- イベント駆動で非同期に動くため、画面は固まりにくい
- 連携ユーザーが増えてきた場合、workerやマシンを増やすことでスケール可能

○ ネイティブアプリケーションと比較して辛いところ

- マシンスペックを大きくオーバーする送信数が短期的に発生すると、大量に送信した 本人以外のユーザー の投稿に遅延が発生する
 - 長期的にはworker数やマシン数を増やすことにより対応可能

- 外部サービスとの連携をするにあたって考える事、学んだこと
- ユーザーのアカウント情報
 - なるべく保持したくない
 - OAuth1.0 認可
 - read しかしないユーザーからは読み込みのみの認可をもらう
 - Twitter上と、mixiボイス上とはつぶやきを関連付けない

- 外部サービスとの連携をするにあたって考える事、学んだこと
- 外部サービスのAPIの動作
 - 今まで動いていた物が、ある日突然動かなくなる
 - リクエスト方式の厳密化
 - エラーレスポンスの変更
 - APIの遅延、ダウン
 - 外部サービスのレスポンスに依存して、内部リソースを変化させる箇所はできるだけ少なくする
 - 外部サービスのレスポンスは疑ってかかる
 - 内部に組み込んだ外部サービスの仕様は常にチェックする



個人的な今後の勝手な展望(希望)

- より自然にTwitterタイムラインを mixiボイス へ取り込めるように
 - 現在は @ と RT を弾いているのみ
 - 特定のハッシュタグを弾くとか？
- もっと活発に使われるように
 - そもそも設定画面がどこにあるのかという話
 - mixiボイス側でコメントがついたことに、Twitterだけ見ていると気付けない
- mixiボイス上で、Twitterからの取り込みが煩くないように

まとめ

- ユーザー数が多いWebサービスで、外部と連携しようとする、色々大変
 - イベント等で突然投稿数が増える
 - 外部からの影響で内部サービスが止まると残念
 - プライバシー情報の管理
- 基本は 並列化 と 非同期化 で解決
 - マシン数とスペックでなんとかする
 - 事前に予測して増やしておく(監視が大事)
- mixiボイスTwitter連携を使ってみてね
 - まだまだ改善点は考えられるので、頑張ります
 - これ欲しい、あれ欲しい、これ要らないは是非教えて下さい
 - 連携方法は↓です
 - 上部のメニューバーの[設定変更] → 外部サービス連携 の [twitter連携] → ...

以上



ご清聴ありがとうございました。