nadoka の ruby 1.9 対応

西山和広

2012/09/15

自己紹介

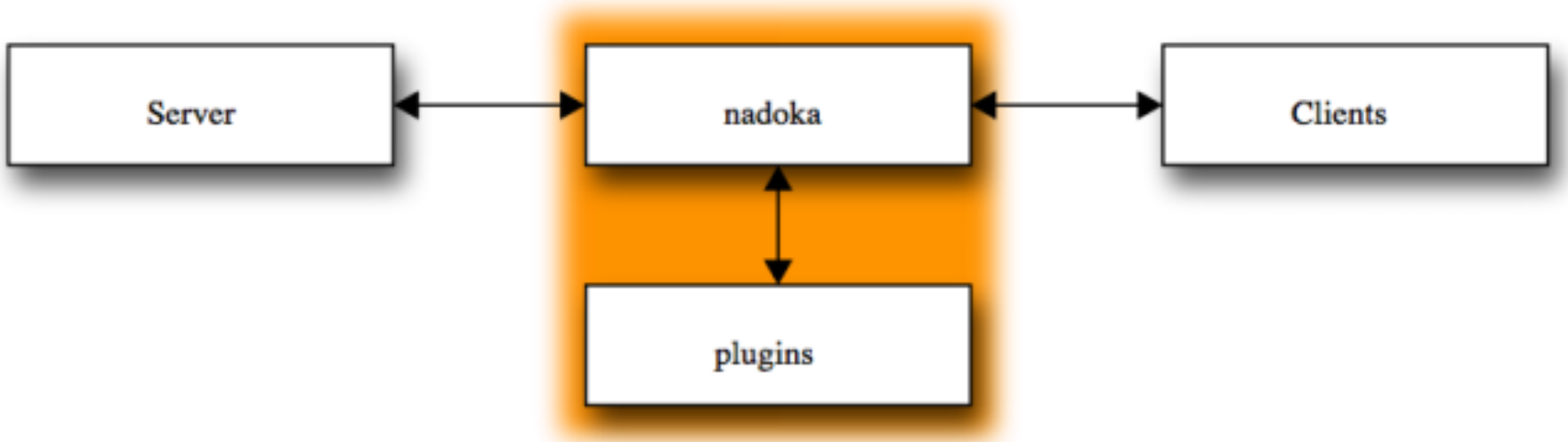
- CRuby のコミッターのひとり
- https://github.com/nadoka のメンバー
- 大阪から来ました
  - 札幌は涼しいかと思いきや...

IRC とは?

- インターネット経由の複数人での**チャット**
  - 1対1ではない
- IRC = Internet Relay Chat
- 古くからあるのでいろいろな環境で使える
- Ruby の開発でも使われている
  - IRCnet の #ruby-ja (日本語)
  - freenode の #ruby-core (英語)

nadoka とは?

- 正式名称は nadoka さん
- IRC Client Server Program
  - **proxy のようなもの**
- plugins (bot) 対応



plugins (bot) の例

- 自動応答系
  - Google 検索
  - **URL からタイトル取得**
  - 天気情報取得
  - 人工無能
- 自動発言系
  - RSS の情報通知
  - **挨拶**
  - 時報

ruby 1.9 での問題

- 発言がみえなかった
- 発言ができなかった
- など

ここから本題

使用 encoding

- 日本語圏では主に以下の encoding を使用
  - **IRCnet : いわゆる JIS コード (ISO-2022-JP)**
  - freenode : UTF-8
- 問題
  - **valid (valid_encoding? が真) とは限らない**
  - クライアントの設定ミス
  - いわゆる半角カナの扱い

nadoka 本体への入出力

- server や client との送受信
  - invalid なものもそのまま通したい
    - **エラー処理は server や client に任せたい**
    - 文字化けの原因になりたくない
- 内部処理や plugins との入出力
  - 元のバイト列そのまま
  - **必要なら plugins で変換**

nadoka 本体での扱い

- 方針：nadoka 本体では ASCII-8BIT で扱う
- force_encoding(Encoding::ASCII_8BIT) で**バイナリ文字列扱い**
- 文字列の内容は**変換しない**
  - ruby 1.8 との互換性も考慮したため

plugins の encoding 問題

- encoding の扱いが
  - plugin によってバラバラ
  - ruby 1.8 では**適当でも動いた**
- ruby 1.9 では
  - Encoding::CompatibilityError **例外頻出**
- たとえば
  - 複数回 nkf を通していたり
  - ISO-2022-JP と正規表現マッチしていたり

plugins の encoding 方針

- **受信時に**
  - **一回だけ UTF-8 に変換**
- **内部処理**
  - **UTF-8 で統一**
  - EUC-JP や Shift_JIS で扱っていた plugin も変更
    - /re/e や /re/s で混在していた
- **送信時に**
  - **一回だけ server encoding に変換**

問題例 (1)

- String#force_encoding(enc) を**定義**
  - self を返すだけ
- ruby 1.8 + 別のライブラリ (nokogiri-1.5.2) の内部で**謎のエラー**
  - uninitialized constant Nokogiri::XML::Node::Encoding
  - 実は ::Encoding を参照しようとしていた
- respond_to?(:force_encoding) で**確認してから使うように変更**

問題例 (2)

- nadoka 本体
  - サーバーへ送信する文字列を作成する部分
- チャンネル名 (ASCII-8BIT) + 発言内容 (JIS)
  - ruby 1.8 だと気にせず結合できていた
  - ruby 1.9 だと**両方 ASCII only でも例外** Encoding::CompatibilityError
    - ISO-2022-JP が ascii_compatible? ではないため
- 発言内容を force_encoding して解決

問題例 (3)

- plugin からの発言例
  - bot 名: "hello bot: "
  - 発言内容: "こんにちは"
- bot名 (UTF-8) + 発言内容 (JIS)
  - 発言内容生成時に tojis で変換済み
  - **結合時に例外** Encoding::CompatibilityError
  - 結合後にも tojis で変換
    - ruby 1.8 だと nkf の自動認識で問題なし
    - ruby 1.9 だと結合後の tojis まで来ない

問題例 (3) の解決方法

- bot名 (UTF-8) + 発言内容 (UTF-8)
  - 発言内容生成時には**変換しない**
  - tojis で**全体を変換**
  - nadoka 本体に渡す直前で**一度だけ**変換すべき

まとめ

- String#force_encoding を定義して**手抜き**
  1.8/1.9 両対応は**不幸の元**
- ISO-2022-JP が US-ASCII 互換ではないのでバグが発見できた
- 無駄に複数回変換しない
- それぞれの**入力、出力部分だけ**で変換すべき
- nkf 便利

おまけ

- trunk で動かしたら問題発生

ruby 2.0 対応

- iconv がなくなっている
- 1.8 も考慮して nkf で置き換える?
- String#encode にするかどうかは未定