

Joruri Gw

JoruriGw 移行手順

ver2.3→ver3.1

サイトブリッジ株式会社

2018年7月9日

内容

1	注意事項.....	2
2	旧 Gw からデータを抽出.....	4
2.1	作業用ディレクトリの作成.....	4
2.2	静的ファイルの取得.....	4
2.3	DB の取得.....	4
3	旧 Gw から新 Gw へデータのコピー.....	6
3.1	データのコピー.....	6
3.2	データの確認.....	6
4	新 Gw へデータの反映.....	7
4.1	静的ファイルの反映.....	7
4.2	静的ファイルディレクトリの名称の修正.....	7
4.3	バックアップデータのリストア.....	7
4.4	新 Gw の DB にデータ移行.....	8
5	新 Gw の動作確認.....	9
5.1	assets コンパイルの実施.....	9
5.2	サーバの再起動.....	9
5.3	画面の確認.....	9

1 注意事項

- 本マニュアルの手順を実施するには、Joruri Gw 3.1.0 のシステムを既存の Gw と別のサーバに構築してください。
- Joruri Gw 1.x 系、2.x 系を使用している場合は、Joruri Gw 2.3.1 にバージョンアップを行った上で 3.1.0 にアップデートするようにしてください。

3.1.0 のソース

<https://github.com/joruri/joruri-gw/tree/v3.1-master>

- 下記手順では、旧 Gw から新 Gw へのデータコピーに `scp` コマンドを使用しています。`scp` コマンドを使用するためには、旧 Gw から新 Gw へ `ssh` でログインできるように新 Gw に設定を行っておく必要があります。
- データ移行を行う場合、新 Gw には初期データの登録を行わないでください。
INSTALL.txt 255 行目

```
/usr/bin/mysql -u root -p production_jorurigw < ./db/sql/jorurigw_init.sql
```

注) 本マニュアルは以下の環境を想定して記述しています。作業を行う場合は、適宜値を読み替えてください。

[旧 Gw]

IP 192.168.0.1

SSH ポート 22

DB development_jgw_xxx

作業ユーザ名 : joruri

DB ユーザ名 : joruri

[新 Gw]

IP 192.168.1.1

SSH ポート 22

DB production_jorurigw

作業ユーザ名 : joruri

DB ユーザ名 : joruri

作業ディレクトリ : /home/joruri/work

Joruri Gw インストールディレクトリ : /var/share/jorurigw

- 免責事項

Joruri Gw 移行手順書は、Joruri Gw3 系への移行を容易にするために作られた参考資料であり、Joruri Gw のいかなる環境におけるデータ移行を保証するものではありません。デフォルトの環境で正常に動作することは確認しておりますが、利用者の環境によっては本手順で作業を行っても正常に移行できない場合がある可能性があります。

本手順書を参考にして Joruri Gw のバージョンアップ作業を行った結果、データの欠損、システムダウンなどにより利用者が被った不利益、損害に対し、弊社は一切の責任を負わないものとします。

2 旧 Gw からデータを抽出

操作は作業ユーザで行います。

```
# su - joruri
```

2.1 作業用ディレクトリの作成

作業用ディレクトリを作成します。

```
$ mkdir /home/joruri/work  
# cd /home/joruri/work
```

2.2 静的ファイルの取得

旧 Gw に登録されている添付ファイルなどの静的ファイルを取得します。

① 添付ファイル

```
$ tar cvzf attaches.tar.gz /var/share/jorurigw/public/_attaches
```

② ボード系の添付ファイル

```
$ tar cvzf modules.tar.gz /var/share/jorurigw/public/_common/modules
```

③ アップロード系の添付ファイル

```
$ tar cvzf upload.tar.gz /var/share/jorurigw/upload
```

④ ポータル広告バナー画像

```
$ tar cvzf files.tar.gz /var/share/jorurigw/public/_common/themes/gw/files/
```

2.3 DB の取得

旧 Gw の DB のバックアップを取得します。

① 作成されている DB の一覧を取得します。

```
$ mysql -u joruri -p -e 'show databases;'
```

② ①で取得した DB のうち、「development_jgw_」で始まる DB のバックアップを取得します。

※ production モードで稼働している場合は、「production_jgw_」で始まる DB となります。

[システム系 DB のバックアップ]

```
$ mysqldump -u joruri -p --skip-lock-tables development_jgw_core >  
development_jgw_core.dump
```

```
$ mysqldump -u joruri -p --skip-lock-tables development_jgw_gw >  
development_jgw_gw.dump
```

```
$ mysqldump -u joruri -p --skip-lock-tables development_jgw_gw_pref >  
development_jgw_gw_pref.dump
```

[掲示板系 DB のバックアップ]

(① の一覧のうち、development_jgw_bbs_で始まる DB を—databases の後ろに半角スペース区切りで並べます。)

```
$ mysqldump -u joruri -p --add-drop-database --skip-lock-tables --databases ¥  
development_jgw_bbs_000001 development_jgw_bbs_000002 ¥  
development_jgw_bbs_000003 > development_jgw_bbs.dump
```

[FAQ 系 DB のバックアップ]

(① の一覧のうち、development_jgw_faq_で始まる DB を—databases の後ろに半角スペース区切りで並べます。)

```
$ mysqldump -ujoruri -p --add-drop-database --skip-lock-tables --databases ¥  
development_jgw_faq_000001 development_jgw_faq_000002 ¥  
> development_jgw_faq.dump
```

[QA 系 DB のバックアップ]

(① の一覧のうち、development_jgw_qa_で始まる DB を—databases の後ろに半角スペース区切りで並べます。)

```
$ mysqldump -ujoruri -p --add-drop-database --skip-lock-tables --databases ¥  
development_jgw_qa_000001 development_jgw_qa_000002 ¥  
> development_jgw_qa.dump
```

[書庫系 DB のバックアップ]

(① の一覧のうち、development_jgw_doc_で始まる DB を—databases の後ろに半角スペース区切りで並べます。)

```
$ mysqldump -ujoruri -p --add-drop-database --skip-lock-tables --databases ¥  
development_jgw_doc_000001 development_jgw_doc_000002 ¥  
> development_jgw_doc.dump
```

[電子図書系 DB のバックアップ]

(① の一覧のうち、development_jgw_dig_で始まる DB を—databases の後ろに半角スペース区切りで並べます。)

```
$ mysqldump -ujoruri -p --add-drop-database --skip-lock-tables --databases ¥  
development_jgw_dig_000001 development_jgw_dig_000002 ¥  
> development_jgw_dig.dump
```

3 旧 Gw から新 Gw へデータのコピー

3.1 データのコピー

scp コマンドを用いて、2 で取得したデータを新 Gw にコピーします。

[旧 Gw]

```
# su - joruri
```

```
$ cd /home/joruri/work
```

```
$ scp -P 22 -r * joruri@192.168.1.1:/home/joruri/work
```

3.2 データの確認

新 Gw にログインし、データがコピーできていることを確認します。

[新 Gw]

```
# su - joruri
```

```
# ls -l /home/joruri/work/
```

※ ファイルとファイルサイズが旧 Gw と一致することを確認します。

4 新 Gw ヘデータの反映

3 でコピーしたファイルを新 Gw に反映します。

[新 Gw]

```
# su - joruri
```

```
$ cd /home/joruri/work
```

4.1 静的ファイルの反映

静的ファイルを回答し、新 Gw のディレクトリにコピーします。

① 添付ファイル

```
$ tar xvzf attaches.tar.gz
```

```
$ mv var/share/jorurigw/public/_attaches/ /var/share/jorurigw/public/_attaches
```

② ボード系の添付ファイル

```
$ tar xvzf modules.tar.gz
```

```
$ mv var/share/jorurigw/public/_common/modules ¥  
/var/share/jorurigw/public/_common/modules/
```

③ アップロード系の添付ファイル

```
$ tar xvzf upload.tar.gz
```

```
$ mv var/share/jorurigw/upload /var/share/jorurigw/upload/
```

④ ポータル広告バナー画像

```
$ tar xvzf files.tar.gz
```

```
$ mkdir /var/share/jorurigw/public/_attaches/gw
```

```
$ mv var/share/jorurigw/public/_common/themes/gw/files/portal/adds ¥  
/var/share/jorurigw/public/_attaches/gw/portal_adds
```

4.2 静的ファイルディレクトリの名称の修正

ディレクトリ名を修正します。

```
$ cd /var/share/jorurigw/public/_attaches/
```

```
$ mv meetings meeting_guides
```

```
$ mv portal gw
```

正常にデータ転送されていることを確認します。

```
$ ll /var/share/jorurigw/public/_attaches/
```

```
$ ll /var/share/jorurigw/public/_common/modules/
```

```
$ ll /var/share/jorurigw/upload/
```

4.3 バックアップデータのリストア

① バックアップデータを新 Gw の DB にリストアします。

```
$ cd /home/joruri/work
```

```
$ mysql -u joruri -p production_jorurigw < development_jgw_core.dump
```

```
$ mysql -u joruri -p production_jorurigw < development_jgw_gw.dump
```

```
$ mysql -u joruri -p production_jorurigw < development_jgw_gw_pref.dump
```



```
$ mysql -u joruri -p < development_jgw_bbs.dump
$ mysql -u joruri -p < development_jgw_faq.dump
$ mysql -u joruri -p < development_jgw_qa.dump
$ mysql -u joruri -p < development_jgw_doc.dump
$ mysql -u joruri -p < development_jgw_dig.dump
```

② 新 Gw に移行の設定が行われていることを確認します。

```
$ less /var/share/jorurigw/config/application.yml
```

migrate:

production_url: <http://192.168.1.1/>

ind_url:

ind_sso_url:

4.4 新 Gw の DB にデータ移行

スクリプトを実行し、新 Gw にデータ移行を行います。

```
$ cd /var/share/jorurigw
```

database.yml の設定を変更し、4.3 でリストアを行ったデータベースに対してスクリプトが実行されるようにしてください。

```
$ vi config/database.yml
```

production:

adapter: mysql2

encoding: utf8

reconnect: false

database: production_jorurigw

pool: 5

username: joruri

password: pass

socket: /var/lib/mysql/mysql.sock

Ver2.3 から Ver3.1 で行ったデータベースの変更を適用します。

```
$ bundle exec rake db:migrate RAILS_ENV=production
```

① テーブルの作成

```
$ bundle exec rake joruri:migrate:create_board_table RAILS_ENV=production
```

```
$ bundle exec rake joruri:migrate:modify_board_table RAILS_ENV=production
```

```
$ bundle exec rake joruri:migrate:update_table RAILS_ENV=production
```

- ```
$ bundle exec rake joruri:migrate:copy_board_data RAILS_ENV=production
$ bundle exec rake joruri:migrate:modify_board_data RAILS_ENV=production
```
- ② 添付ファイル、画像ファイルのリンクの修正
- ```
$ bundle exec rake joruri:migrate:modify_attach_and_image_link ¥
RAILS_ENV=production
```
- ③ emotions アイコンのリンクの修正
- ```
$ bundle exec rake joruri:migrate:modify_emotions_link RAILS_ENV=production
```
- ④ 研修申込機能の添付ファイル情報のコピー
- ```
$ bundle exec rake joruri:migrate:copy_sb01_data RAILS_ENV=production
```
- ⑤ 記事のリンクの修正
- ```
$ bundle exec rake joruri:migrate:modify_doc_link RAILS_ENV=production
```
- ⑥ 添付ファイルのディレクトリの修正
- ```
$ bundle exec rake joruri:migrate:modify_attach_dir RAILS_ENV=production
```
- ⑦ 画像ファイルのディレクトリの修正
- ```
$ bundle exec rake joruri:migrate:modify_image_dir RAILS_ENV=production
```
- ⑧ TODO データの修正
- ```
$ bundle exec rake joruri:migrate:copy_todo_data RAILS_ENV=production
```
- ⑨ 研修申込機能の添付ファイルを新ディレクトリにコピー
- ```
$ bundle exec rake joruri:migrate:modify_sb01_attach_dir RAILS_ENV=production
```
- ⑩ 統合済みの DB の削除
- ```
$ bundle exec rake joruri:migrate:drop_db RAILS_ENV=production
```

5 新 Gw の動作確認

新 Gw の動作確認を行います。

5.1 assets コンパイルの実施

```
$ cd /var/share/jorurigw
$ bundle exec rake assets:precompile
```

5.2 サーバの再起動

新 Gw のサーバの再起動を行います。

```
$ su -
# reboot
```

5.3 画面の確認

Web ブラウザから Gw の管理画面にアクセスし、画面が正常に表示されることを確認します。

URL: <http://192.168.1.1/admin>

※ ログイン ID、パスワードは旧 Gw で使用していた値を使用することができます。