

Rmenuとは

Rmenuとはどんなものなのか
どうやって作るのかを説明します

Rmenuとは

もくじ

- ▶ **Rmenuとは**
- ▶ Rmenuの特徴
- ▶ 業務用WEBアプリとは
- ▶ 仕様書は不要とは
- ▶ MVCモデルとは
- ▶ クライアント側とサーバー側とは
- ▶ JSONとは
- ▶ Rmenuファイル構成のツリー
- ▶ 必要な知識

Rmenuとは

ソフトウェアフレームワークの一つとしエンジン基礎となる使用言語をRubyで作成されたものであり、

日本製のプログラミング言語Rubyを使用したぶん日本初の日本製フレームワークがRmenuである。

Webベースの業務アプリケーション開発エンジン「Rmenu」では、Web系開発時に懸念される点が改善されており、

Rmenuは業務用Webアプリケーションをノンプログラミングで自社開発することを目指した無料のフレームワークです。

MVCアーキテクチャを取り入れ、並列分散バッチ処理、帳票作成処理も兼ね備えており、開発費用の削減ができるフレームワークです。

メインフレームを意識した、フレームワークのため、サーバーの規模によりますが、世界中からの同時アクセスにも耐えうる設計となっています。

他に類を見ない、下記の機能をかねそなえています。

新規プロジェクトはもちろんのこと、プロジェクトごとの業務追加に際しても、

サブシステムの追加構築が容易に行えるよう、

一つのインスタンスで複数プロジェクトに対応可能です。

プロジェクトごとのデータベース割り当てはもちろん、一つのプロジェクトで複数のデータベースアクセスも可能です。

ジョブ管理画面による並列分散バッチの進捗状況の管理が可能です。

ジョブ管理画面のリスタート機能により、異常終了した分割ジョブの再起動が可能です。

Rubyで悩む帳票作成工数の軽減・解消のために、

同じプログラムで、オンライン帳票・バッチ帳票の二種類に対応が可能です。

バッチ帳票は、帳票管理画面（スプーラ機能）から出力・再出力・削除等がコントロール可能です。

A j a xによる実装で、画面再描画を最小に抑えレスポンスが早いのも特徴

日本語によるプログラム開発が可能のため、生産性、保守性、システムの引継ぎ等が飛躍的に向上します。

『国際規格ISO/IEC 30170承認された、日本製のプログラミング言語Rubyを日本製フレームワークRmenuを使って開発できる』

まさに、Rmenuは日本向け業務用Webアプリケーション開発の基本となりえるでしょう。

プロトタイプ利用2012年より実装

開発発明者：下地 忠史

Rmenuの特徴

▶ 業務用WEBアプリのフレームワークです

Webアプリの為webサイトに接続できる様々なデバイスで業務を実行できます

▶ 仕様書は不要

画面とER図だけで作成します（帳票ありの場合もある）

▶ MVCモデル

プログラムを役割ごとにモデル・ビュー・コントローラに分けて動いています

▶ クライアント側とサーバー側に分けて作成します

クライアント側はブラウザ関係・サーバー側はDB関係

▶ JSONでデータを流す

日本語でネーミングができ、項目名を揃えて作成することでJSONファイルがデータを流してくれます

RmenuのプログラミングとはJSONファイルを作成することです

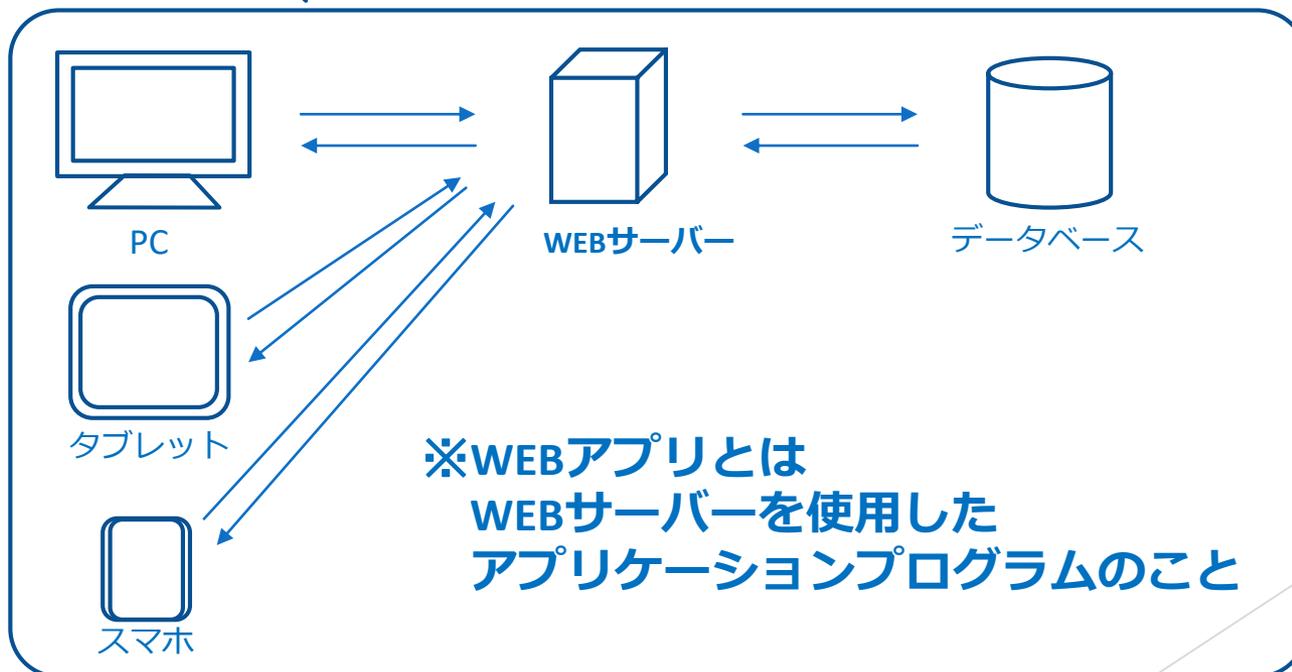
Rmenuの特徴

▶ 業務用 webアプリとは



インターネット

インターネット環境にある様々なデバイスで
会社の業務システムを使用することができます



Rmenuの特徴

▶ 仕様書不要とは

作成手順



Rmenuの作成は『**画面**』と『**ER図**』で作成するため仕様書はありません
ER図と画面を確認すると、どのプログラムをコピーすれば良いか解り、
ボタンの接続先や動かすプログラムなどは [Appspec.js] に書かれており、
仕様書の役割をしています

作成時に受け取るものは

「Rmenuプログラム」「ER図」「作成プログラム一覧表」の3つです
3つを用意し、雛型になるプログラムをコピーし、項目名を作成しながら、
Rmenuを作成します。 ※上記図

Rmenuの特徴

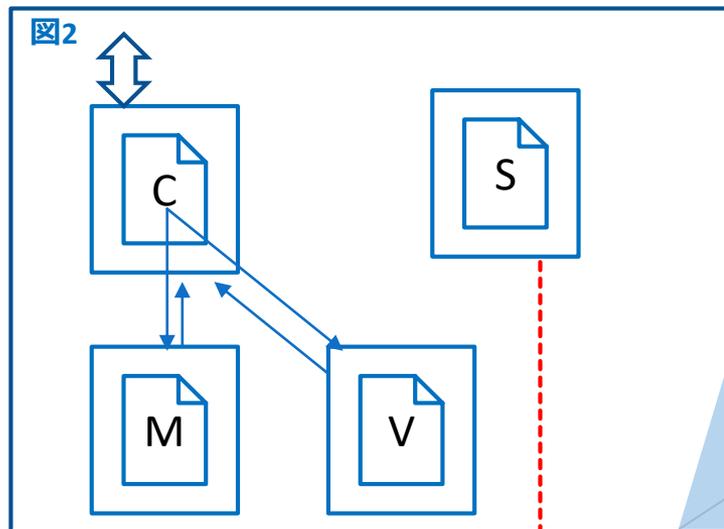
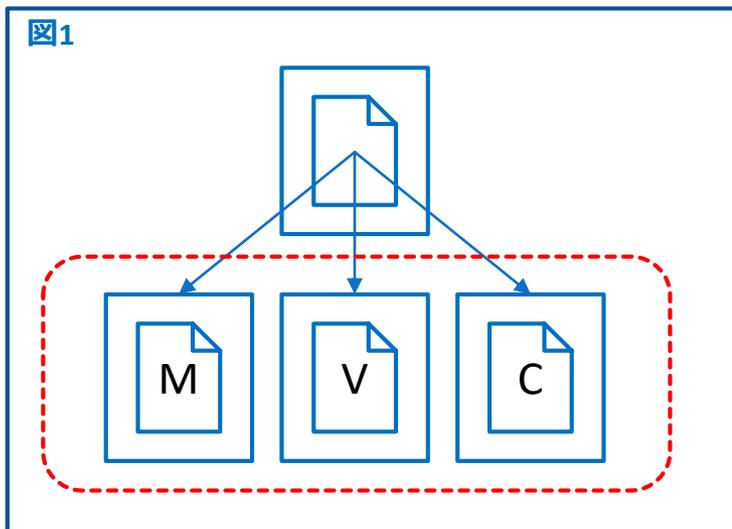
▶ MVCモデルとは

1つのプログラムを**役割ごとに** [model] [view] [controller] の3つに分けて作成します※図1

これらの役割は [controller] は受け取った指示を**モデルとビューへ指示**を出し回答を返します※図2

[model] は**データ処理**をコントローラから指示され実行します

[view] は**デザイン処理**をコントローラから指示され実行します

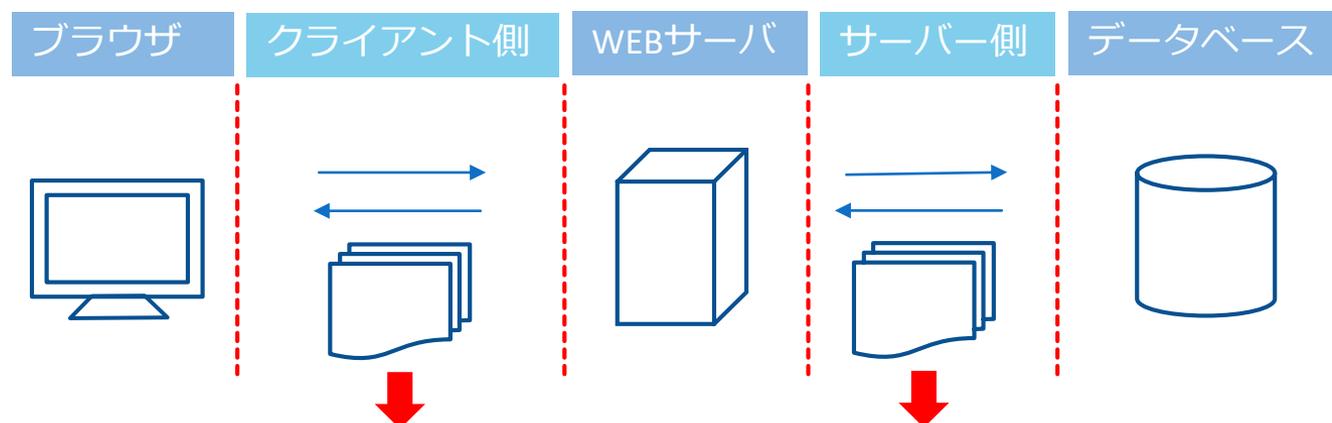


追記 : Rmenuは**SMVCモデル**とも言います (SMVCは海外ではあるが日本にないタイプ)
これは「MVC」+「s」のことです [s] は [Appspec] のことを指します
[Appspec] はボタンのイベント設定や使用するJSONが書かれていますが実行する訳ではなく、
他のシステムにある**仕様書の役割**をしています (APPはアプリケーション・SPECKは仕様書の意味)

Rmenuの特徴

▶ クライアント側・サーバー側とは

Rmenuのプログラムはwebサーバーを挟んで2つに分れます
左側がクライアント側、右側がサーバー側です

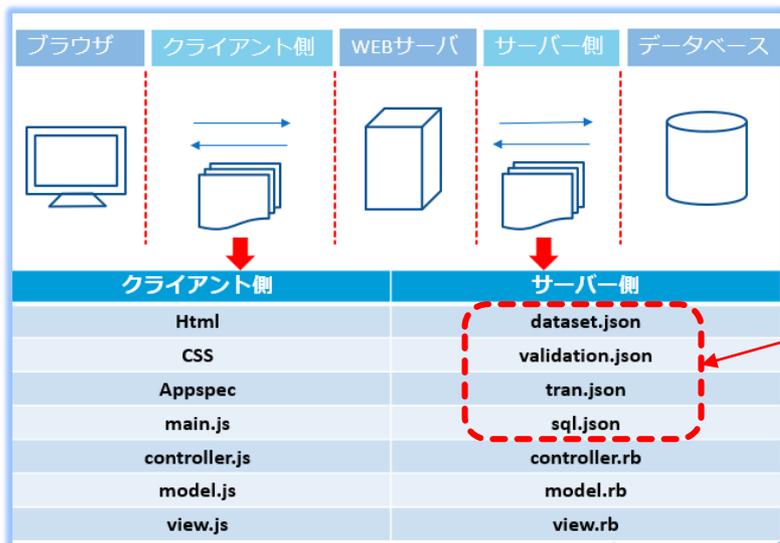


クライアント側のプログラム	サーバー側のプログラム
Html	dataset.json
CSS	validation.json
Appspec	tran.json
main.js	sql.json
controller.js	controller.rb
model.js	model.rb
view.js	view.rb

Rmenuの特徴

▶ JSONとは

JSONとはJavaScript Object Notationの略で、XMLなどと同様のテキストベースのデータフォーマットです。その名前の由来の通りJSONはJavaScriptのオブジェクト表記構文のサブセットとなっており、XMLと比べると簡潔に構造化されたデータを記述することができるため、**記述が容易で人間が理解しやすいデータフォーマット**と言えます。



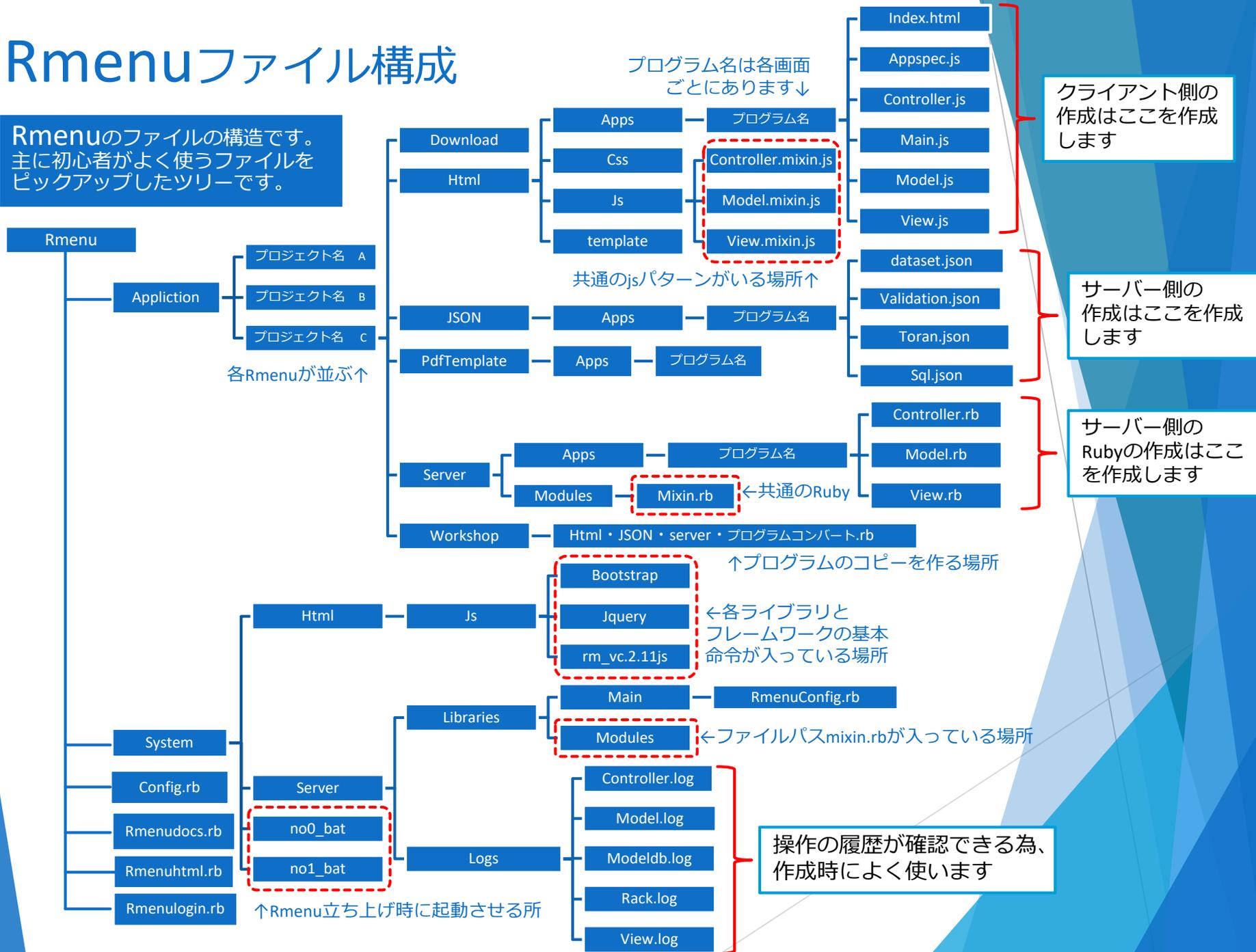
Rmenuは
サーバー側の様々なJSONを作成し、
データを流します
※項目名が日本語で作成できる為
初心者でもわかりやすくなっています
※Rmenuのプログラミングとは、
このJSONを作成することです

Rmenuの全体像を次の「Rmenuファイル構成」で確認し、

- ✓ RmenuがSMVCモデルである事
- ✓ クライアント側とサーバー側の分かれている事
- ✓ 拡張子を確認し何が使われているのかなどを確認してください

Rmenuファイル構成

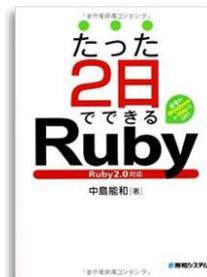
Rmenuのファイルの構造です。
主に初心者がよく使うファイルを
ピックアップしたツリーです。



必要な知識

Rmenuの作成時に必要になる知識です
初心者向けレベルからで十分なので自習しましょう

- ▶ Ruby
- ▶ JavaScript
- ▶ Postgres SQL
- ▶ Html・css
- ▶ jQuery
- ▶ Bootstrap
- ▶ ER図



Rmenuとは完了

次に、

Rmenuがどうやってデータを流しているのか
データの流れ方の詳細を学びます。



Rmenuとは

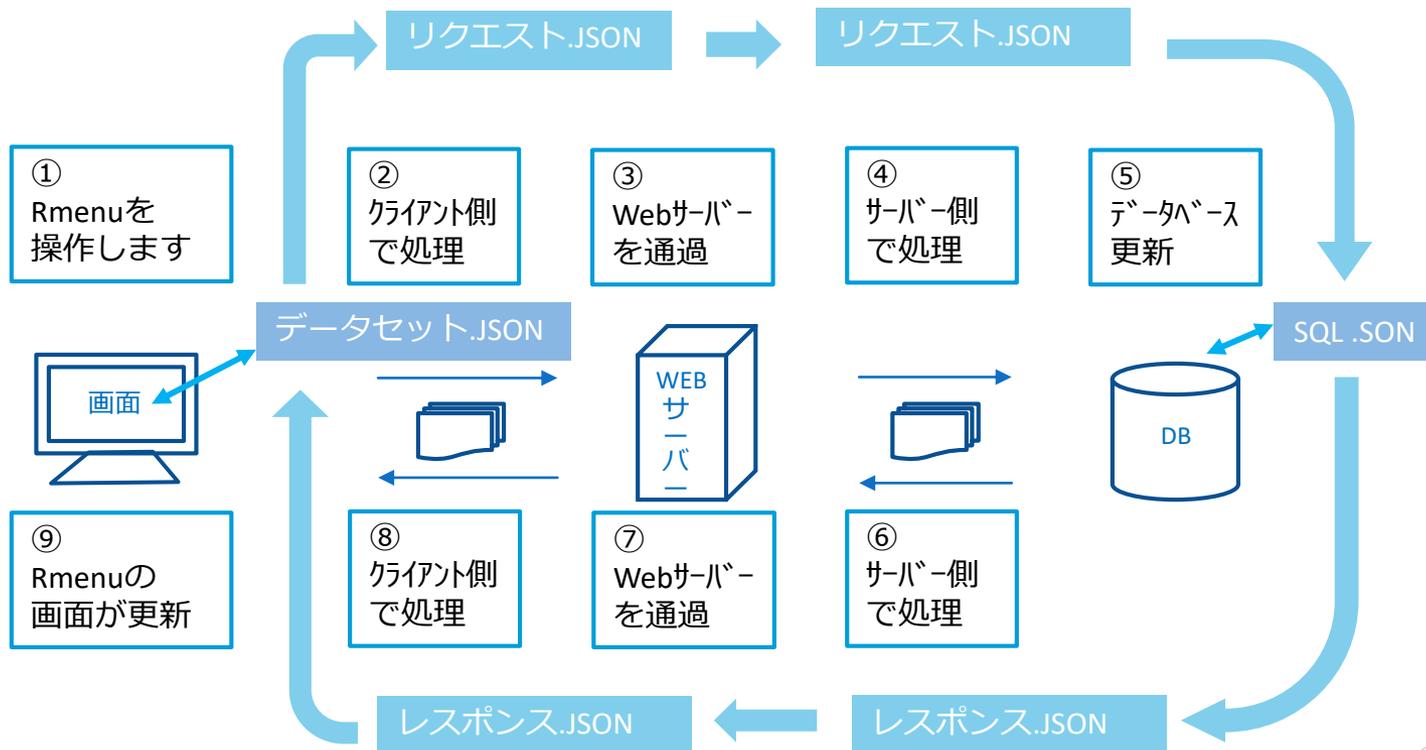
データの流れ方

データの流 れ方 もくじ

- ▶ データフロー（主なJSONの流れ）
- ▶ データの流れ方（全体図）
- ▶ データの流れ方（詳細図）
- ▶ WEBサーバーとは

データフロー（主なJSONの流れ）

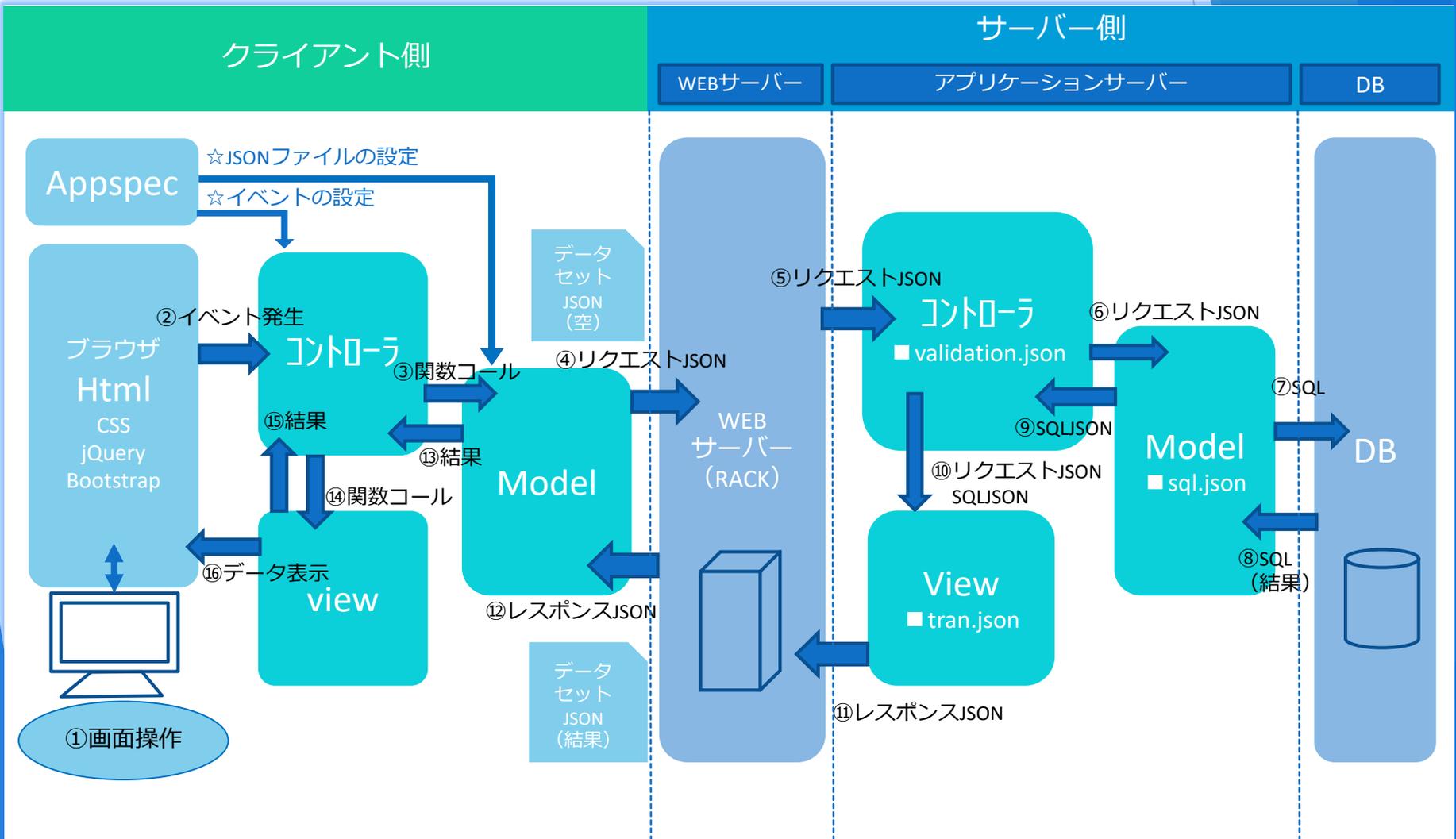
①～⑨の順にプログラムが動きデータを処理して画面が更新されます
各処理はJSONでデータを流しています※白文字が主なJSONの流れです



※データセットJSONは項目を定義している
(データを移送する時の入れ物・Htmlはこのデータをもとに表示する)
※リクエストJSON・レスポンスJSONはtran.jsonの中に書かれています

データの流れ (全体図)

左下の①画面操作から⑬画面に表示されるまでのデータの流れ



データの流れ (詳細図)

	処理場所	発生処理	流れ
クライアント側	①画面操作	画面	「このデータが欲しい」をクリックする
	②イベント発生	画面 コントローラ	「ボタンをクリックされた」 = (イベント発生) Appspec☆イベントの設定を実行できるように動く
	③関数コール	コントローラ モデル	「モデルへデータを持ってきて」と指示する 「モデルはデータを貰う為のリクエストを作る」 Appspec☆JSONの設定を確認
	④リクエストJSON	モデル Webサーバ	「webサーバへリクエストを送る」 このときデータセットJSONはデータが空の状態 「webサーバではリクエストを受け取りRACKで仕分ける」
	⑤リクエストJSON	Webサーバ コントローラ	「サーバ側のコントローラへリクエストを送る」 「コントローラはリクエストを受け取りモデルへ指示をおくる」
サーバ側	⑥リクエストJSON	コントローラ モデル	「モデルはレスポンスを受け取りSQLを作成する」 「モデルはDBからデータを貰うためにSQLを作成する」
	⑦SQL	モデル DB	「DBへSQLを送る」 「DBはSQLを読みテーブルからデータを用意する」
	⑧SQL (結果)	DB モデル	「モデルへデータを入れたSQLを返す」 「モデルはもらった結果でsql.jsonを作成する」
	⑨SQLJSON	モデル コントローラ	「コントローラへデータ入りのsql.jsonを送る」 「コントローラはデータ入りのsql.jsonを受け取りビューへ送る」
	⑩リクエストJSONとSQLJSON	コントローラ ビュー	「ビューへリクエストとsql.json (結果入り) を送る」 「ビューはリクエストとsql.json (結果入り) を受け取り、レスポンスを作成する」
	⑪レスポンスJSON	ビュー Webサーバ	「Webサーバへレスポンスを送る」 「webサーバはレスポンスを受け取る」
	⑫レスポンスJSON	Webサーバ モデル	「クライアント側のモデルへレスポンスを送る」 「モデルはレスポンスから結果を作成する」 データセットJSONはデータが入った状態に
	⑬結果	モデル コントローラ	「コントローラへ結果を送る」 「コントローラは結果を貰いビューへ表示する為の指示を作成」
	⑭関数コール	コントローラ ビュー	「ビューへ色を変えてAを表示して」と指示する 「ビューは画面に表示するものをセットしてブラウザへ送る」
	⑮結果	ビュー	「コントローラへセットした結果を送る」
クライアント側	⑯データ表示	画面	「画面が切り替わりデータが表示される」

データの流れる方

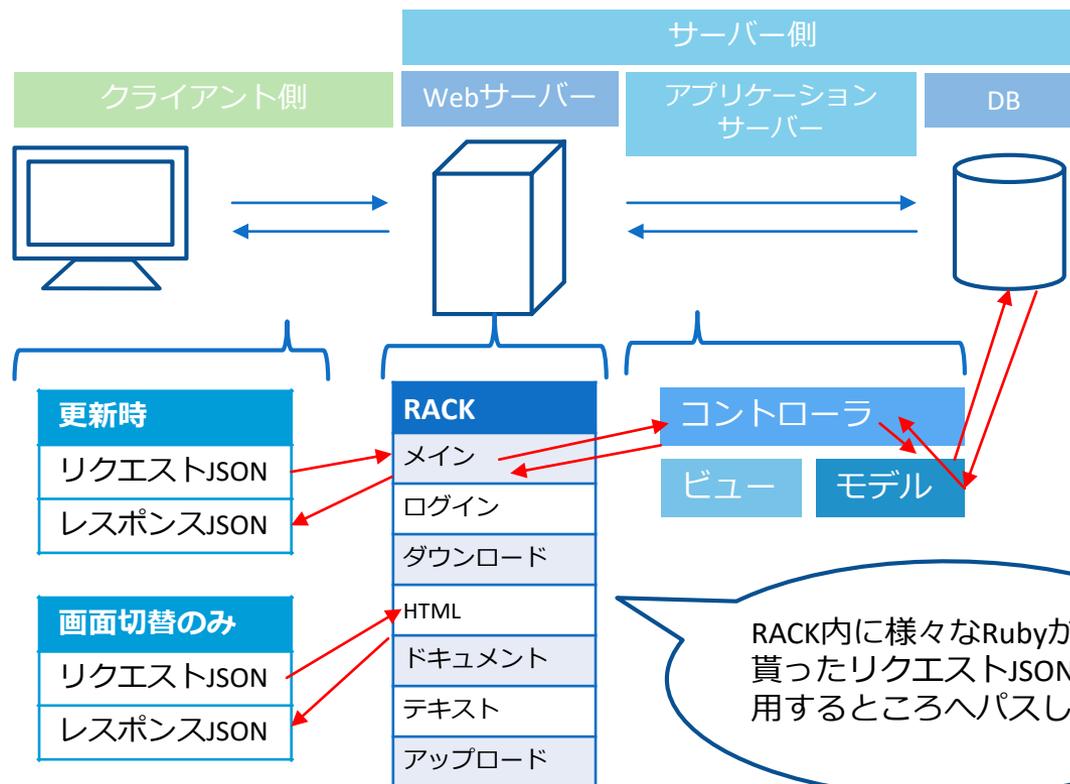
▶ Webサーバーとは

Rmenuで利用するのは『**WEBサーバー**』 = 『**RACK**』です

『**RACK**』とは**rubyをインストールしたときに一緒に入っているwebサーバー**の事です

『**RACK**』の役割：受け取ったJSONを指示されたところへパスするだけ※下記図

※Webサーバーの事を『Webリック』ともいう（webサーバー = webリック = RACK）



※RACKのメインはJSONへファイルをパスしてプログラムを動かす役割

HtmlはHtmlファイルを持ってきて画面を切り替える役割 7箇所それぞれに役割が分かれています

データの流れ完了

データの流れを一度、

『ビジュアルツール』を使い追ってみるとわかりやすいです

(数値を入力した時のパターン・ログイン時のパターンなど

動くプログラムによって違うので様々なパターンで入力したデータが移送される様子を追いかけます)

Rmenuとは

Rmenuを動かす

Rmenuを動かす もくじ

- ▶ 必要なツール
- ▶ 環境設定（ツールのインストール）
- ▶ プログラムを用意（SVN）
- ▶ データベースを用意（postgres）
- ▶ 立ち上げ方
- ▶ URLを開く
- ▶ ログイン画面
- ▶ メニュー画面

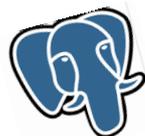
必要なツール

Rmenuは**ローカル環境**で作成します

作成後に『RmenuSVN』を更新することで作成したプログラムを開発メンバーに共有します

まずは自身のPCに必要なツールを用意します

ツール	名称
DB (データベース)	『postgres』プロジェクトごとにデータを保管
プログラムファイル	『RmenuSVN』プログラムの保存先
テキストエディター	『ノートパッド++』 『秀丸』 プログラムを書く
ブラウザ	『クローム』 Rmenuを開く



環境設定 (ツールのインストール)

環境設定 (インストール)

- 「Ruby」 (事前学習時に済)
- データベース「Postgres」
- プログラム共有「SVN」
- テキストエディタ「ノートパット++」
- ブラウザ「クローム」



プログラムを用意

- SVNとは
- SVNの設定
- SVN使い方



DBを用意

- データベースをローカルへ作成する
- コンフィグ設定

Rmenu作成に必要なツールを各自PCに用意します

必要なツールをWEB検索を行い、手順にそってインストールします

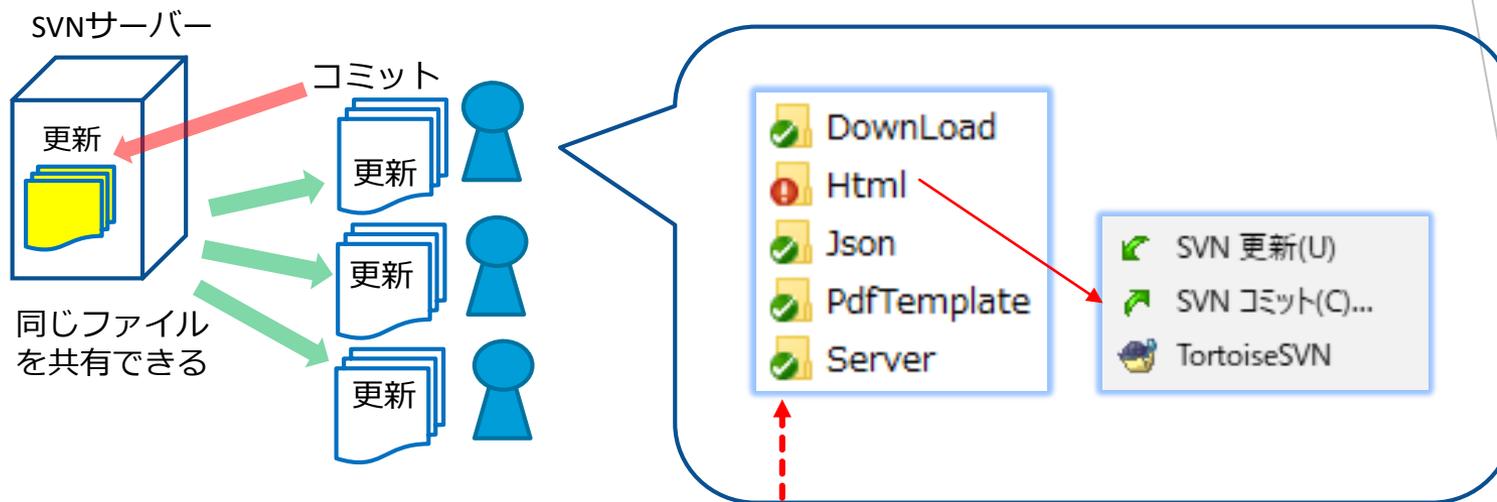
※ 「Ruby」と「postgres」はバージョンがあるので対応バージョンの確認が必要です

全てのインストールが終わったら、次の設定へ進みます

プログラムを用意 (SVN)

▶ SVNとは

プロジェクト作成メンバーで**最新のプログラムを共有する為に使用**します
[コミット] と [更新] をするだけで簡単に作成したRmenuのプログラムが共有できます



基本の使い方

- 1.プログラムを作成中は赤いマークが付きます
- 2.共有しても良い状態になったらコミットを切ります
- 3.コミットを切り緑マークになったファイルはメンバー全員に共有できます

マーク	状態
緑	共有ファイル（作業前に毎朝更新すること）
赤	自身で作成中、コミットすると緑になります
なし	共有していないファイル（自分だけが見れる）

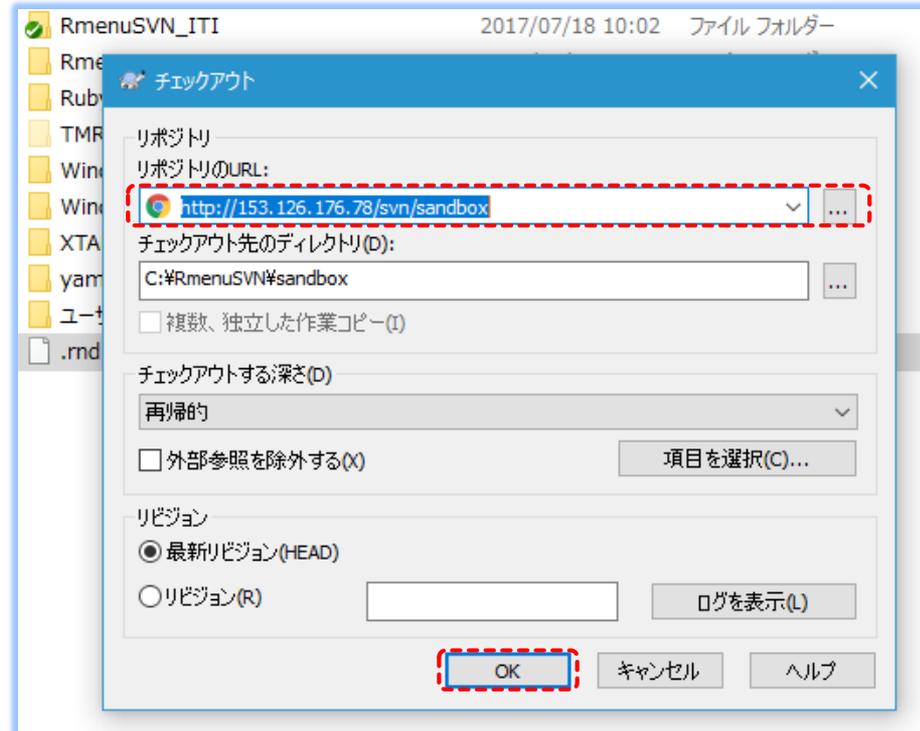
プログラムを用意 (SVN)

▶ SVNの設定

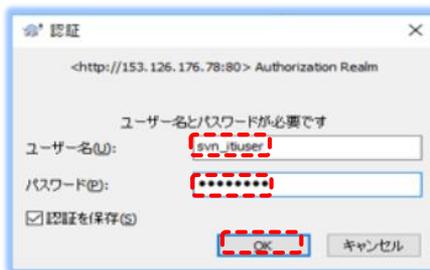
- 1.新規フォルダーを作成します
(c:直下にタイトルは
[RmenuSVN])



- 2.右クリック [SVNチェックアウト] を押下
「リポジトリURL」へ接続先URL
「http://153.126.176.78/svn/sandbox」を入力し [OK]



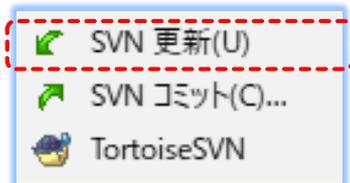
- 3.認証画面に
ユーザー名・PWを入力する
ID: 「svn_ituser」
PW: 「vwg5E92T」



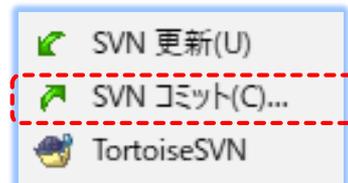
プログラムを用意 (SVN)

▶ 使い方

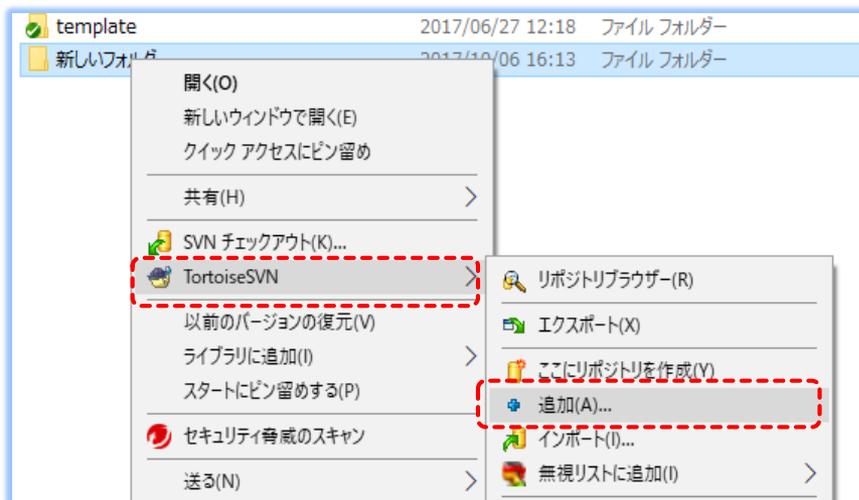
1. 毎朝必ず **【更新】** を実行します
(最新の状態始めるため)
(Rmenuフォルダーの上で右クリック)



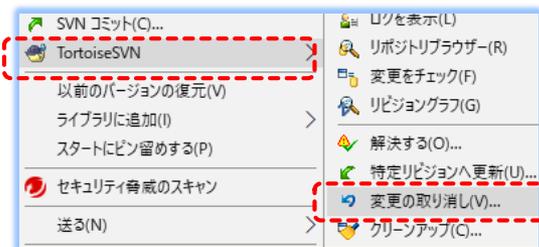
2. 作成できたら **【コミット】** を実行すること
(誤ったプログラムや共有不要のものはコミットしない事)
(作成できたファイルの上で右クリック)



3. 新規作成したときは **【追加】** を実行し、コミットを実行する



4. 作成を取り消したいときは
コミットを切る前に
【変更の取り消し】 を実行する
(朝更新したときの状態に戻り、
初めからやり直せます)

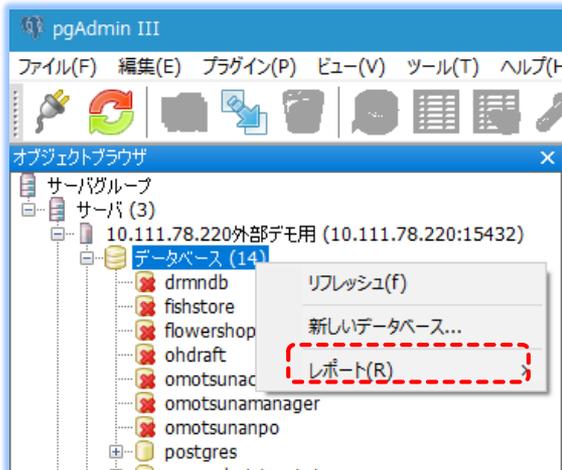


データベースを用意 (postgres)

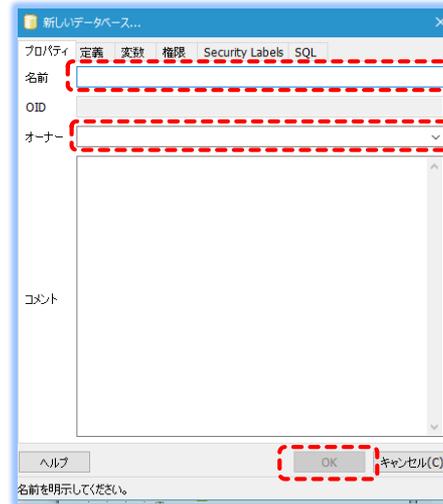
▶ データベースをローカルへ作成する

1. ポスグレを開き、新しいデータベースを作成します

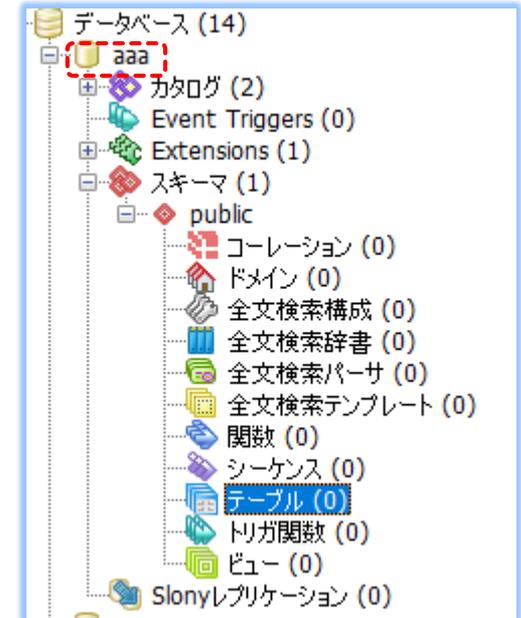
① データベースの上で右クリック→
[新しいデータベース] を開く



② [名前] を作成する
(例: DB名 [aaa] を作成します)
[オーナー] を選択→ [postgres]



③ [OK] をクリック→ローカルに
新しいDBができます

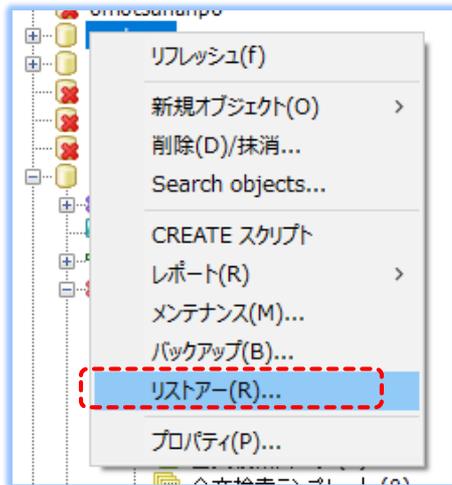


データベースを用意 (postgres)

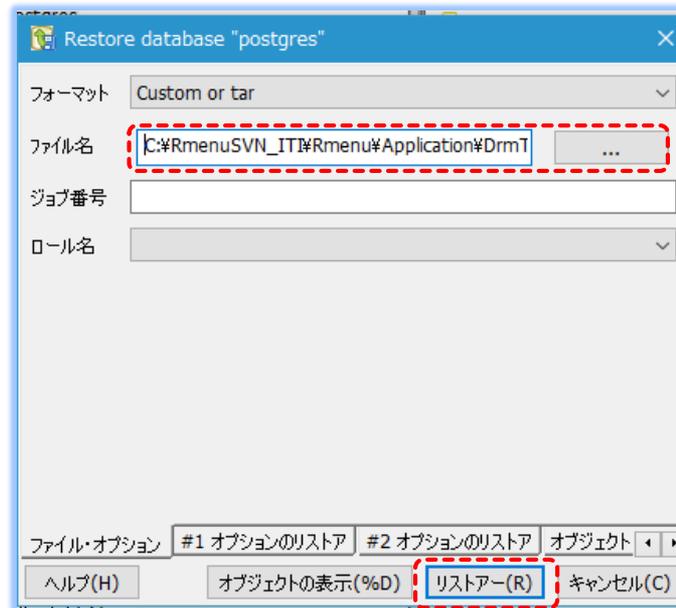
▶ データベースをローカルへ作成する

2. リストアを実行する (データを入れること)

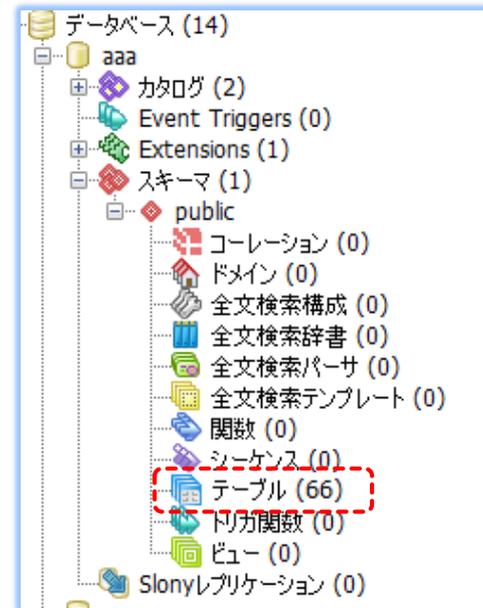
① データベース [aaa] の上で
右クリック → **リストア**
を開く



② ファイル名を [...] から選択する
(SVN内にバックアップ済の物を貰う)
リストア を実行する
(データが多いと時間がかかります)



③ [aaa] を開き
テーブルの件数を確認し
件数が入っていれば成功です



データベースを用意 (postgres)

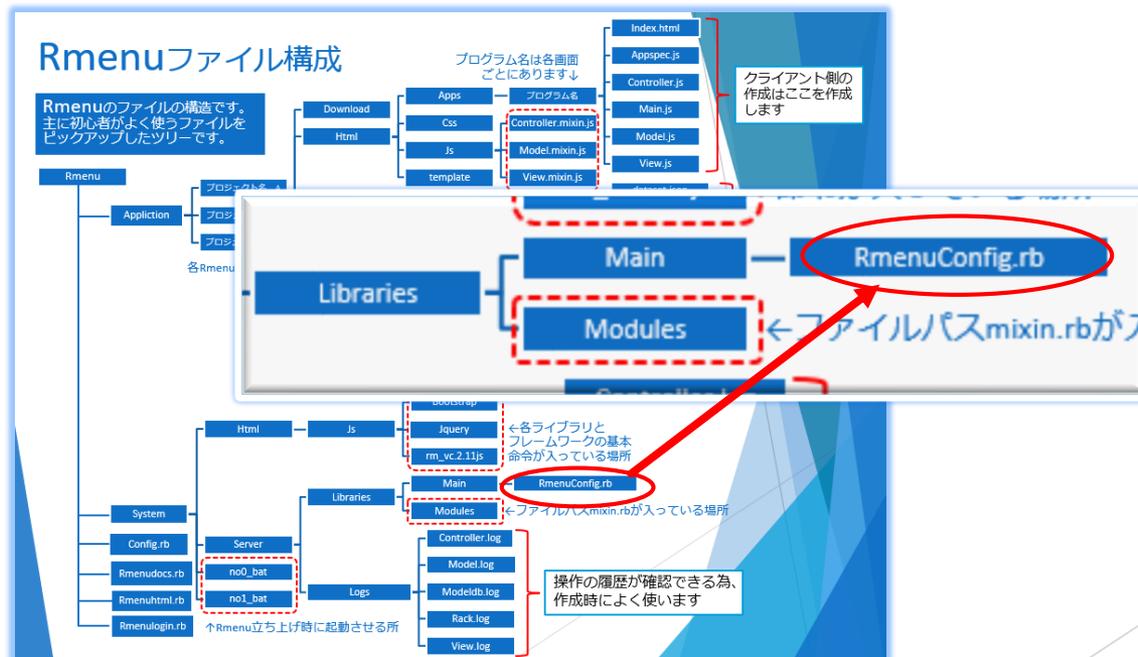
▶ コンフィグ設定

DBを用意できたらRmenuのプログラムとDBをつなぐ作業が必要になります
コンフィグとは [RmenuConfig.rb] というファイルです

構成図で見ると下のほうにあります

これは各自参加するプロジェクトによって内容が違う為SVNで共有しません

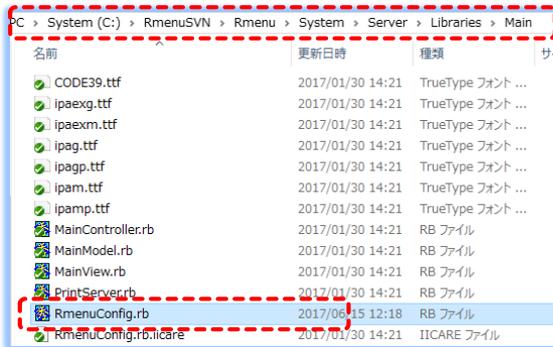
ファイルを貰い、中身を自分の参加するプロジェクトのDBにつながるよう書き換えます



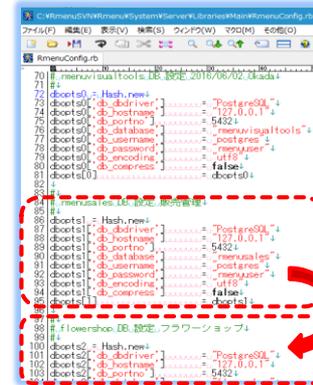
データベースを用意 (postgres)

▶ コンフィグ設定

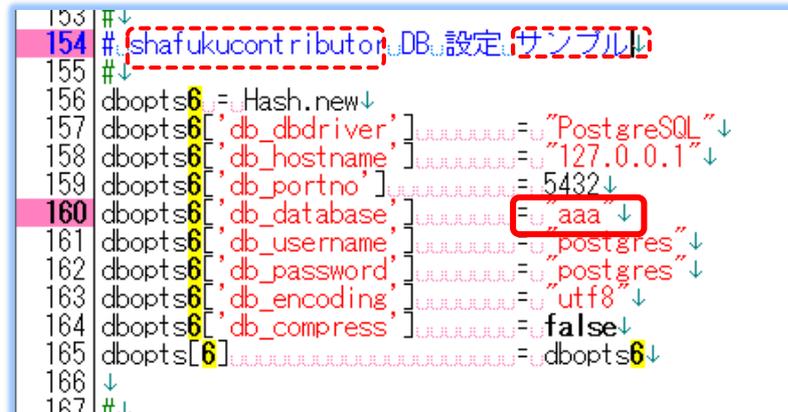
1. [RmenuConfig.rb] を貰い、RmenuSVNの中へ置きます
[C:¥RmenuSVN¥Rmenu¥System¥Server¥Libraries¥Main] の中へ貼り付け



2. [RmenuConfig.rb] を開き、[DB設定] が並んでいる事を確認し今回追加分を作成します 1塊をコピーし下部に追加する



3. ①コピーできたら黄色部分の[数字]を変更します
②青字部分に[DB名]と[プロジェクト名]を作成
③赤枠のデータベースをDB名の[aaa]で作成



4.3箇所変更できたら上書保存して完成

注意 :

- ①データベース以外の項目
ユーザー名やPWなど人によって違う場合もあります
- ②SVNに追加しない事
各自設定が違う為SVNに追加しない
- ③DB名は半角小文字です

Rmenuの立ち上げ方



1. データベースを起動する

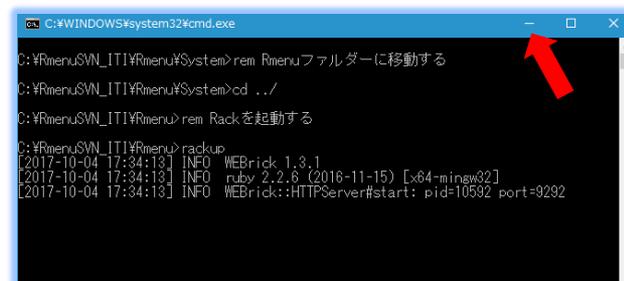
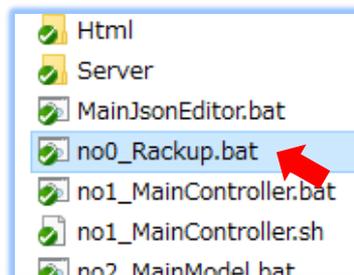
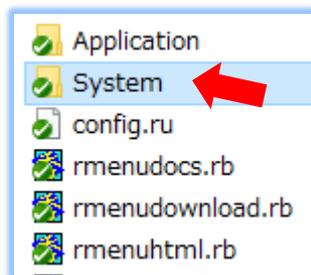
(PCの設定でほとんど自動で起動していますので何もしなくてOKです)



2. WEBサーバーとアプリケーションサーバーを起動する

Rmenuのファイルの中のsystemを開き、

- ① [no0_Rackup.bat] をクリックし起動させます (これは「webサーバー」を起動させるプログラムです)
- ② [no1_Maincontroller.bat] をクリック (これはアプリケーションサーバーを起動させるプログラムです)
- ③ 2つのコマンドプロンプトが開く、閉じずに [最小化] してください ※閉じてしまうとRmenuは起動できません



3. ブラウザを開きRmenuのURLを開き、ログインできたらRmenuが使用できます

Rmenuを動かす完了

次は、

Rmenuを作成します☆