

CardForm 用の form を挿入します。

## このプラグインについて

引数として指定したページ `formpage` の内容に従いフォームを出力します。このフォームにより入力された値は引数として渡した `basename` を元に、`bugtrack` プラグインの様に `basename/ 数字` というページに保存されます。ただし、`savepage` が指定されていれば `savepage` に保存されます。

- ・ 第 1 引数に `formpage` を指定します。
- ・ 第 2 引数に `basename` を指定します。
- ・ 第 3 引数に `savepage` が指定できます。
- ・ `basename` が省略された場合は `formpage` が使われます。

```
{{cardform formpage,basename,savepage}}
```

`formpage` では `formtext`、`formsubmit` 等のプラグインを使って実際のフォーム要素を配置します。`formstart`、`formend` などは `cardform` プラグインにより追加されるため、これらのプラグインは置かないでください。

このフォームを使って送られたデータは `basename/ 数字` というページに保存されます。ただし、CGI パラメータ `page` がある場合はその値が保存ページ名となります。`savepage` があればこの値を CGI パラメータ `page` に設定し、保存ページ名とします。

このフォームにより作られるページは `basename`、もしくは `formpage` の内容を元に作られます。新しく作られるページは元のページの、`% パラメータ名 %` またはパラメータ名に対応するフォームプラグインをパラメータ値で置換した内容となります。また、`basename` というページがなければ `::` と `:::` を使って CGI パラメータをすべて保存します。

CGI パラメータは、ユーザが使う任意のパラメータのほか、幾つかの機能をもつパラメータが存在します。以下にそれを説明します。

`action`

`cardform` プラグインのアクションハンドラを呼び出すために "CARDFORM" である必要があります。本プラグインにより自動的に付けられます。

`basename`

アクションハンドラに保存に使用すべき `basename` を通知するために必要です。本プラグインにより自動的に付けられます。

`page`

このパラメータは通常アクションハンドラに渡す必要がありません。アクションハンドラにこのパラメータが渡されていない場合、アクションハンドラは `basename` より保存すべきページ名を求め、このパラメータに設定します。このパラメータがすでに設定されていた場合はそのままそのページ名が使われます。(すでに存在していれば上書きされます)

`time`

このパラメータは通常アクションハンドラに渡す必要がありません。アクションハンドラにこのパラメータが渡されていない場合、アクションハンドラは現在日時をこのパラメータに設定します。これにより保存ページに保存日時を残す事ができます。</dd>

`debug`

このパラメータは通常アクションハンドラに渡す必要がありません。アクションハンドラに

このパラメータが渡された場合、アクションハンドラはデバッグモードで動作します。アクションハンドラは通常のように動作し、保存すべき内容を生成し、クライアントに送信しますが、実際には保存しません。保存される値やページ内容を確認するために使用できます。なお、パラメータ値は評価されません。

CardForm: カード型データベースのためのプラグイン群

## CardForm プラグイン全般について

### 内容

FreeStyle Wiki でカード型データベースを実現するプラグイン群です。

### 入力

プラグインにより ( wiki 書式の範囲で ) 自由なレイアウトのフォームを作ることができます。

### 保存

フォームにより入力されたデータは雛型となるページを元に 1 件毎に、バグトラックのような `basename/ 数字` というページとして保存されます。

### 利用

`basename/ 数字` 形式のページ群をカード型データベースとみなして表を作ったり合計したりできます。

現在公開しているのは beta 以前の alpha 版です！

いま、cardform プラグインを使って入力した値の修正を行なうための仕組みを整えようとしています。それにともない、おそらく一部の仕様も含め、ソースコードにある程度大きな手が入る予定です。そういった意味で alpha 版とさせていただきます。

### インストール

添付ファイルを取得してプラグインディレクトリに展開してください。

.

### 使用方法 1 : 簡単な例

まず、データ入力用のフォームが必要です。

`form*` プラグインによるフォームの記述

フォームの記述は `form` で始まるフォーム記述用のプラグインによって行ないます。

```
* 名前: {{formtext name}}
* 年代: {{formselect age,10代,20代,30代,40代,50代}}
* 好きな物: {{formtext favor,size=30}}
{{formsubmit 投票}} {{formreset リセット}}
```

これを Favorite というページに保存します。え？フォームが表示されない？それで正常です。慌てずに。

cardform プラグインによるフォームの表示

次にフォームを表示したい所に以下のように書きます。

```
{{cardform Favorite}}
```

これで、フォームを表示するためのプラグイン cardform により、先ほどのフォームを記述したページをインクルードしフォームとして表示します。

## データの入力

- ・ 名前 :
- ・ 年代 :
- ・ 好きな物 :

先ほどのフォームに値を記入して投票ボタンを押してください。 そうすると Favorite/1 というページができ、先ほど入力した値が保存されていると思います。

## 使用方法 2 : 保存用レイアウトを別にする

先ほどのように入力用フォームを配置したページを作ります。

### 入力用フォーム配置ページ

以下のような内容を BugTrackForm というページに保存します。

```
'',投稿者,{{formtext user}}
',サマリ, "{{formtext subject,size=60}}"'
',カテゴリ, "{{formselect category, 本体, プラグイン, ドキュメント}}"'
',優先度, "{{formselect rate, 緊急, 重要, 普通, 低}}"'
',状態, "{{formselect stat, 提案, 着手, 完了, リリース済, 保留, 却下}}"'
',内容, "{{formtextarea msg,rows=10,cols=60}}"'
{{formsubmit 投稿}} {{formreset リセット}}
```

### 入力データレイアウトページ

次に、入力データを保存するページをレイアウトします。

```
!!!%subject%
* 投稿者: %user%
* カテゴリ: %category%
* 優先度: %rate%
* 状態: %stat%
* 日時: %time%
{{bugstate}}
!! 内容
%msg%
!! コメント
{{comment}}
```

これを BugTrack-test として保存します。

### フォームの使用

それではフォームを使用してみましょう。

```
{{cardform BugTrackForm,BugTrack-test}}
```

投稿者	
サマリ	
カテゴリ	

優先度	
状態	
内容	

### 使用方法 3：入力データを利用する

入力されたデータの利用方法です。

#### 入力データの一覧表示

利用方法 1 のフォームから入力した値を一覧表示するには次のように書きます。

```
{{cardlist Favorite, 名前:* 名前: , 年代:* 年代: , 好きな物:* 好きな物: ,}}
```

名前	年代	好きな物
strawberry	10 代	苺
apple	20 代	林檎
一条はるか	10 代	ムカデ
	10 代	
ドラえもん	10 代	どら焼き

20 代のデータのみ、名前順に表示したい場合は次のように書きます。

```
{{cardlist Favorite,* 年代: ==20 代,%sortby_alphabet%:* 名前: , 名前:* 名前: , 年代:* 年代: , 好きな物:* 好きな物: ,}}
```

名前	年代	好きな物
apple	20 代	林檎

わかりにくいですね。実はこの方法は推奨できません。入力データを扱う場合は利用方法 2 の保存レイアウトを別にする方法を用いた方がわかりやすくなります。利用方法 2 の場合、次のようになります。

```
{{cardlist BugTrack-test,%sortby_alphabet%:stat, ページ:%cardlist_pagename%, カテゴリ:category, 優先度:rate, 状態:stat, 投稿者:user, サマリ:subject}}
```

ページ	カテゴリ	優先度	状態	投稿者	サマリ
<a href="#">BugTrack-test/1</a>	本体	普通	提案	admin	test

最初の「[BugTrack-test](#)」は [BugTrack-test/](#) 数字ページが対象で、そのフォーマットはページ [BugTrack-test](#) が元になっている事を示しています。

次の「%sortby\_alphabet%:stat」は stat をアルファベット順(正確には文字コード順)にソートするという事です。

次の「ページ:%cardlist\_pagename%」からは実際の表示内容です。この時、%cardlist\_pagename% は

ページ名を表します。

## 入力件数の表示

利用方法 1 で「20 代の件数 / 全体の件数」という表示をしてみましょう。

```
{{cardcount Favorite,*年代: ==20代}}/{{cardcount Favorite}}
```

1/5

利用方法 2 の場合、次のようになります。

```
現在全提案 {{cardcount BugTrack-test}} 件中  
{{cardcount BugTrack-test,stat== 提案 }} が提案状態で、  
そのうち優先度が緊急は {{cardcount BugTrack-test,stat== 提案 ,rate== 緊急 }} 件、  
重要は {{cardcount BugTrack-test,stat== 提案 ,rate== 重要 }} 件、  
普通は {{cardcount BugTrack-test,stat== 提案 ,rate== 普通 }} 件、  
低は {{cardcount BugTrack-test,stat== 提案 ,rate== 低 }} 件です
```

現在全 1 件中 1 件が提案状態で、そのうち優先度が緊急は 0 件、重要は 0 件、普通は 1 件、低は 0 件です。

## サンプル

[http://aaa-www.net/~typer/cardform\\_sample.html](http://aaa-www.net/~typer/cardform_sample.html) にて、ここに上げた例を書いています。おいおい、他の例なども書く予定です。

## ライセンス

GPL とします。

## 更新履歴

### var.3 から var.4alpha1 までの主な変更点

- ・ cardlist,cardcount,cardsum 各プラグインで対象データの条件指定をできるようにした。
- ・ cardlist プラグインでソート指定できるようにした。
- ・ mod\_perl 時キャッシュ内容が残り、更新されないバグを修正した。
- ・ その他幾つかバグフィックスしたけど省略

### var.4alpha1 から var.4alpha2 までの主な変更点

- ・ 初期値取得機能の追加
  - ・ cardform のヘルプに初期値取得に関する説明を追加
  - ・ cardform に初期値取得先の判定と登録を行なう部分を追加
  - ・ ラベル値取得正規表現キャッシュを保存するハッシュ変数のキーを変更
  - ・ CardFuncBase に初期値取得用情報を保存するハッシュ変数を追加
  - ・ CardFuncBase::do\_page\_cache の引数 num に "init" を渡すと登録した初期値取得先から値をキャッシュするように変更
  - ・ CardFuncBase に set\_initpage,is\_set\_initpage,unset\_initpage,get\_initdata を追加
  - ・ FormFuncBase が CardFuncBase を継承
  - ・ 前のパラメータを優先するように FormFuncBase::get\_param を変更
  - ・ form\* プラグインに初期値取得と設定を行なう部分を追加
- ・ 複数行も取得できるようラベル値取得正規表現を変更
  - ・ 正規表現マッチに s オプション追加

- ・必ず値取得部分の前後に文字を入れる
- ・ cardlist にセルの最大文字数指定機能を追加
- ・ 幾つかインデントを修正
- ・ cardform プラグインのヘルプにあった誤記 2 件を修正

var.4alpha2 から var.4alpha3 までの主な変更点

- ・ サンプル設置サイトのアドレス変更
- ・ cardlist のセル内最大文字数指定機能において末尾の文字列処理厳格化と 3 バイト文字への対応
- ・ 値取得時の生成正規表現をすこし改良

FrontPage へのコメントはこちらに