# Administration


- Installation
- Forms and Models
  - Create a model by designing a Form
  - Markdown Syntax
- Reports
- Utilities

# Installation

# Forms and Models

# Create a model by designing a Form

## Introduction

You create a table model, the add and update form and a default report when you save the form. Therefore you need to have prepared a number of things before before you start creating the model. These are:

- **The model or table name** - all the major models and their associated forms and reports were created when the system was first installed. You can edit any of the existing forms to add new fields but generally this will not be necessary. You will generally be considering this activity because you are adding something completely new.
- **The field names** - it makes sense to create another spreadsheet along the lines of that we provide at /docs/Cliqon V9 Data Dictionary. By default, the Redbean ORM database facility automatically creats every field as a variable alphanumeric field (Varchar(255)) and changes and expands this when needed.
- **Select and list options** - if the form is going to include selectsor radio choices, these should be configured as options. A static list can be entered or a list accessed dynamically.

## Creating the model form

Open the Form Designer. It consists of three panes. On the left are the components that can be used. On the right is a column where the options for that element will displayed and can be augmented and edited. Some of the additions that we will describe in this section are ***mandatory,*** please pay particular significance to these instructions. In the middle is the drawing board to which components will be dragged.

Forms can have multiple rows and columns but all of these must sit within a Group. By default, any buttons are displayed outside and below the Group in a separate box.

Start by entering the name of the model / form as the ID of the form. This should be entered as "form-modelname", e.g. **form-dbcontac**t. This is ***mandatory.*** Version tag is optional and is numeric.

Drag a Container->Group to the drawing surface. In the options column, enter an explanatory name for the group, for example: "Form for dbmodels". In Custom Properties, create a key called **table** and enter a value for that key, which must be the name of the database table. This is ***mandatory.***

We choose to have a header row for our forms and sub-heading or description below. So drag two text-views to the drawing surface and inside the Group. Then edit the text in the two boxes. You use the "Markdown Syntax" to apply style to these text boxes. Thus we chose ### Heading small for the first text box and no suffix at all, to create body text in the second box.

You can now start on the fields themselves. In our example we created 5 fields. We chose that the first two should be aligned in a row. We chose to drag all our components to the drawing surface and Group in one go and then change the options for each element in turn.

For field one, which is the Reference, we chose, in the options, the following attributes:

- Text field component
- **Field label** - entered label text, Reference. Will go above the field.
- **Field description** - the descriptive or help text that goes below the input field
- **Key "*mandatory*"** - field name for table, form data and report column data
-

# Markdown Syntax

These are the elements outlined in John Gruber's original design document. All Markdown applications support these elements.

| Element | Markdown Syntax |
| --- | --- |
| Heading | `# H1`<br>`## H2`<br>`### H3` |
| Bold | `**bold text**` |
| Italic | `*italicized text*` |
| Blockquote | `> blockquote` |
| Ordered List | `1. First item`<br>`2. Second item`<br>`3. Third item` |
| Unordered List | `- First item`<br>`- Second item`<br>`- Third item` |
| Code | `` `code` `` |
| Horizontal Rule | `---` |
| Link | `[title](https://www.example.com)` |
| Image | `![alt text](image.jpg)` |

These elements extend the basic syntax by adding additional features. Not all Markdown applications support these elements.

| Element | Markdown Syntax |
| --- | --- |
| Table | `\| Syntax \| Description \|`<br>`\| ----------- \| ----------- \|`<br>`\| Header \| Title \|`<br>`\| Paragraph \| Text \|` |

| Element | Markdown Syntax |
| --- | --- |
| Fenced Code Block | ```<br>{<br>  "firstName": "John",<br>  "lastName": "Smith",<br>  "age": 25<br>}<br>``` |
| Footnote | `Here's a sentence with a footnote. [^1]`<br><br>`[^1]: This is the footnote.` |
| Heading ID | `### My Great Heading {#custom-id}` |
| Definition List | `term`<br>`: definition` |
| Strikethrough | `~~The world is flat.~~` |
| Task List | `- [x] Write the press release`<br>`- [ ] Update the website`<br>`- [ ] Contact the media` |
| Emoji<br>(see also Copying and Pasting Emoji) | `That is so funny! :joy:` |
| Highlight | `I need to highlight these ==very important words==.` |
| Subscript | `H~2~O` |
| Superscript | `X^2^` |

# Reports

# Utilities