

Parent-select

A page describing how to edit a relationship by adding a HTML select element where the user can select which parent element must be associated with the model.

The essentials

Functional perspective

Technical perspective

Creating a child and attach it to a parent

Rendering the <select>

Digging deeper

Validating the select value

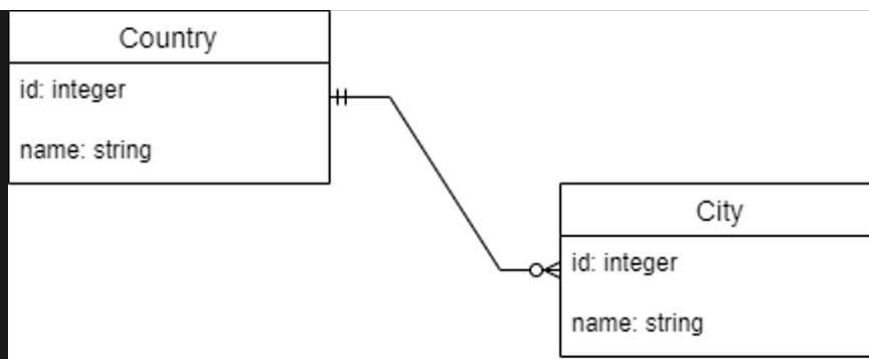
Limited options: radio buttons

Further reading

The essentials

Parent-select is a UI design pattern that is useful when users need to manage related resources. For example, a country can have many cities, an order can have many orderlines, a blog post can have many comments, and each comment can have many reactions. This page explains the ideas behind this pattern and how to implement this in Laravel.

In a parent-select, the user is able to manage the relationship with a parent object when creating or editing a child. For example, given the following two entities that are related:



Functional perspective

When the user wants to create a new child (in this case, a city) - or edit an existing one -, he must be able to select a parent item (a country) where the child must belong to. For example with a HTML `<select>`, like this:

A hand-drawn sketch of a form titled "Create a new city". It contains a text input field for "Name" and a dropdown menu for "Country". The dropdown menu is currently showing "Spain" and has a list of other countries: Netherlands, Spain, Belgium, Sweden, Great Britain, Namibia, and South Africa. A green "Save" button is visible to the left of the dropdown, and a yellow button is to its right. A line connects the "Save" button to the "Country" dropdown.


Technical perspective

From the technical perspective, 2 aspects are important:

- Creating a `<select>` element in the form that is prefilled with all available parents
- Handling the selection in the `store` or `update` controller methods when the form is submitted

The details of this is best explained by starting at the end


Creating a child and attach it to a parent

This is already covered in  [Eloquent relationships - Inserting & Updating Related Models](#). The only difference here that the country's id must be passed in the request, like so:

```
public function store(Request $request) { // Validation logic omitted // Fetch the country $country = Country::find($request->country_id); // Create the city object $city = new City(); $city->name = $request->name; // Associate it to the country and save to the database $country->cities()->save($city); return redirect()->route('cities.index')->with('success', "$city->title is created successfully"); }
```

As per Laravel convention, the name of the form input should match the corresponding column name.

Rendering the `<select>`

This is also already covered, see  [Form inputs - Introduction to <select>](#). The difference here is that the `<option>` elements need to be rendered dynamically, using a `@foreach` directive, for example like so:

```
<select name="country_id"> @foreach(\App\Models\Country::all() as $country) <option value="{ $country->id }">{ $country->name }</option> @endforeach </select>
```

It renders an `<option>` for each country in the database. It renders the `id` as its value and `name` as its content.

Digging deeper

Validating the select value

Although a select limits the available options for the user, it is good practice to validate any incoming input for security reasons (never trust your users).

To achieve this, you can add a validation rule that checks if the value exists in the given table's column, like so:

```
$validated = $request->validate([ 'name' => 'required', 'country_id' => 'required|exists:countries,id' ]);
```

Limited options: radio buttons

When there are just 3 or 4 options, you can consider using radio buttons instead of a select.

See:

HTML Forms

W3Schools offers free online tutorials, references and exercises in all the major languages of the web. Covering popular subjects



https://www.w3schools.com/html/html_forms.asp



Further reading

When the amount of available options becomes large a regular select becomes less usable. The user might get caught in a long scrolling affair which he might not like. As an alternative, you can try a filter dropdown. See:

How To Search for Items in a Dropdown

W3Schools offers free online tutorials, references and exercises in all the major languages of the web. Covering popular subjects



https://www.w3schools.com/howto/howto_js_filter_dropdo...

