

Renderer Component Trees



This page is still being drafted, and is currently incomplete.

A renderer "component tree" describes the UI in terms of *rendered components* (not to be confused with [Infusion Components](#)). A *rendered component* is a data structure that represent the contents and data binding function of a view, separate from any particular rendering of it. For example, the component representing a 'selection' would include information regarding the available choices, the display strings associated with the choices, and the current selection:

```
{
  optionlist: ["kitchen", "livingRoom", "bedroom", "diningRoom"],
  optionnames: ["Kitchen", "Living Room", "Bedroom", "Dining Room"]
  selection: "${myModel.roomChoice}"
}
```

The rendered component for the selection is independent of whether or not it is rendered using a `<select>` element, a set of radio buttons, or a set of checkboxes.

New in v1.3: ProtoTrees

While the Renderer functions accept a full component tree as an argument, developers typically won't need to create these trees directly. We expect developers to work with simpler structures called *protoTrees*, which are expanded into full component trees using utility functions provided by the Renderer. The Renderer also provides functions for generating common tree structures from other information, such as a data model.

ProtoComponents

The Renderer categorizes components into different types depending on the nature of the data that is to be rendered. Different component types will have different fields, but in general, the values of the fields will contain either the actual data or a reference to a data model containing the data. For more information about the different protoComponent types, see [ProtoComponent Types](#).