

# fluid.initDependents



This functionality is [Sneak Peek](#) status. This means that the **APIs may change**. We welcome your feedback, ideas, and code, but please use caution if you use this new functionality.

## fluid.initDependents(that)

Initializes any dependent subcomponents and [invokers](#) specified in the options for a component.

```
fluid.initDependents(that);
```

**File name:** FluidIoC.js

### Parameters

<b>that</b>	(Object) the <code>that</code> object of the parent component
-------------	---

### Return Value

None
------

### See Also

- [fluid.demands](#)
- [fluid.defaults](#)
- [Invokers](#)
- [How to Use Infusion IoC](#)

---

### Notes

`fluid.initDependents()` examines the `components` and `invokers` properties of the parent components options (i.e. `that.options.components`). If neither of these properties is found, or is empty, this function will have no effect.

`fluid.initDependents()` works in conjunction with [fluid.demands\(\)](#). For more information on this relationship, please see [How to Use Infusion IoC](#)

### Example

```
myApp.myComponent
= function
(container,
options) {
  var that =
fluid.initView
("myApp.
myComponent",
container,
options);
  ...
  fluid.
initDependents
(that);
  ...
  return that;
};
fluid.defaults
("myApp.
myComponent", {
  components: {
    sub1: {
      type:
"myApp.
mySubcomponent1"
    },
    sub2: {
      type:
"myApp.
mySubcomponent2"
    }
  }
});
```

In this example, the `fluid.defaults()` call describes two subcomponents, `sub1` and `sub2` in the `components` property. The call to `fluid.initDependents()` will initialize these two subcomponents based on any demands specified through `fluid.demands` (not shown here: please see the [How to Use Infusion IoC](#) for information about how these functions work together, and the `fluid.demands()` API for details about that function).