Template-driven Forms

By: Mosh Hamedani

Forms Module

In final release, we have a module specifically designed for working with forms. In order to build forms with Angular, you must import this module into **app.module.ts:**:

```
import { FormsModule } from '@angular/forms';


@NgModule({
    imports: [
        FormsModule,
        …
    ]
})
```

ngControl Directive

There are a few references to **ngControl** directive throughout this section. We apply this directive on input fields and this will create a control object under the hood. This control object represents the state of an input field (e.g. whether it is valid or not, whether its value is changed or not, etc)

```
<input ngControl="firstName">
```

To get access to the underlying control object, we can create a template variable and set it to **ngForm**:

```
<input ngControl="firstName" #firstName="ngForm">
```

When Angular sees **firstName="ngForm"**, it'll set this variable to the underlying control object.

Template-driven Forms

By: Mosh Hamedani

This syntax is changed in final release. Instead of using **ngControl**, we use **ngModel.** We should also set the **name** attribute.

```
<input ngModel name=“firstName”>
```

In order to get access to the underlying control object, we set our template variable to **ngModel** instead of **ngForm:**

```
<input ngModel name=“firstName” #firstName=“ngModel”>
```

Keep these changes in mind if you’re working with the final seed project. Everything else in this section is valid in the final release.

Summary

Beta	`<input ngControl=“firstName” #firstName=“ngForm”>`
Final	`<input ngModel name=“firstName” #firstName=“ngModel”>`