

GETTING STARTED WITH ANGULARJS



WORKING WITH FORMS

Hello again! As promised, here's a post to follow [my previous article](#). This time, I'll talk about a way of eliminating the necessity to write boilerplate code for your projects.

The basics

AngularJS is a framework that supports module structure, which allows creating independent modules. Let's break down the approach using forms as an example. Now, what is required for forms to work?

1. **View** - for displaying fields
2. **Validation** - for checking input data
3. **Data processing**

The first two items will appear in every project, as 90% of forms look alike. Let's make a form into a component that will take care of display and validation. Component responsible for validation we'll get from [our repository](#). We can divide form generator into 2 modules:

1. The first one is the form itself. It will process our data object and launch function, if the form is filled out correctly.
2. The second one is responsible for field rendering and validation.

I suggest using the following structure to describe data for generation:

```
this.model = {
```

```
},
```

```
    },  
    },  
  },  
},
```

```
return field.$dirty && !re.test(field.$mode
```

```
attr: {placeholder: 'Your email on
```

```
},
```

```
  }],
```

```
},
```

```
    },  
  },  
},
```

```
    }  
  };  
}
```

Rules allow setting up a custom validator, **attr** stores a list of our fields. The form will initialize upon calling the directive:

```
body
```

```
mgr-form-builder(model="ctrl.model")
```

Perhaps, at this point you'll have a valid question: "How can I customize **View** for a unique form?" For that you'll need to simply add URL of a new form template using **templateUrl** attribute:

As far as displaying goes, everything is pretty much clear. But now there's another question to answer: how do we get data after a form's been successfully filled out? For that, let's add a function that runs upon “**submit**” event:

```

formBuilderCtrl.submit = function () {
    // Loop through the formBuilderCtrl.model.attr
    // and create an object with the key value pairs
    // and then pass it to the submit function
    var data = {};
    Object.keys(formBuilderCtrl.model.attr).forEach(function (key) {
        data[key] = formBuilderCtrl.model.attr[key].value;
    });
    formBuilderCtrl.model.submit(data);
};

```

Now all we need to do is mention our function in the module. Once a form is completed, **submit** function will be performed and we'll get the data added by user.

Result

We have a component that we can use from project to project and we don't have to worry about how or what to validate. We just need to describe data model. Want to learn more? Take a peek into [our repository](#).