



Валидация формы

Последнее обновление: 1.11.2015



G+1



Изучай английский по Skype!

При создании нового объекта нередко могут возникать ошибки. Но механизм валидации AngularJS позволяет нам информировать пользователя об ошибках. Возьмем, к примеру, html-страницу из прошлого раздела и добавим к ней форму:

```
<!doctype html>
<html ng-app="questApp">
<head>
<!-- содержание опущено для краткости -->

<div ng-controller="AnswerController">
<form name="answerForm">
<fieldset>
<p><label for="answerText">Текст ответа</label>
<input id="answerText" ng-model="answer.text" required placeholder="Введите ответ" /></p>
<p><label for="answerAuthor">Автор ответа</label>
<input id="answerAuthor" ng-model="answer.author" required placeholder="Введите автора"
/></p>
<p><label for="answerDate">Дата ответа</label>
<input id="answerDate" ng-model="answer.date" required ng-pattern="/\d\d\/\d\d\/\d\d\/\d\d/"
placeholder="Введите дату" /></p>
</fieldset>
<button type="submit" ng-click="save(answer, answerForm)">Сохранить</button>
</form>
</div>
<script src="js/lib/angular.min.js"></script>
<script src="js/app.js"></script>
<script src="js/controllers/QuestionController.js"></script>
<script src="js/controllers/AnswerController.js"></script>
<script src="js/filter.js"></script>
</body>
</html>
```

За работу с этой формой будет отвечать контроллер AnswerController, который мы сейчас создадим и который мы подключаем внизу страницы. Чтобы связать создаваемый объект с полями, используется директива **ng-model**. Поле с атрибутом

`ng-model="answer.text"`, таким образом, будет отвечать за свойство `text` создаваемого формой объекта `answer`.

Так как нам надо, чтобы все поля перед отправкой были заполнены, устанавливаем для каждого поля атрибут `required`, который, в случае незаполнения поля, укажет пользователю, что данное поле надо обязательно заполнить.

Для поля для ввода даты также устанавливаем атрибут `ng-pattern="/\d\d/\d\d/\d\d\d\d/"`, который указывает на формат вводимой даты. Директива `ng-pattern` принимает регулярное выражение, которому должен соответствовать вводимое значение. То есть в данном случае регулярное выражение соответствует, например, выражению `12/11/2013`.

И, наконец, для кнопки отправки установлено событие нажатия:

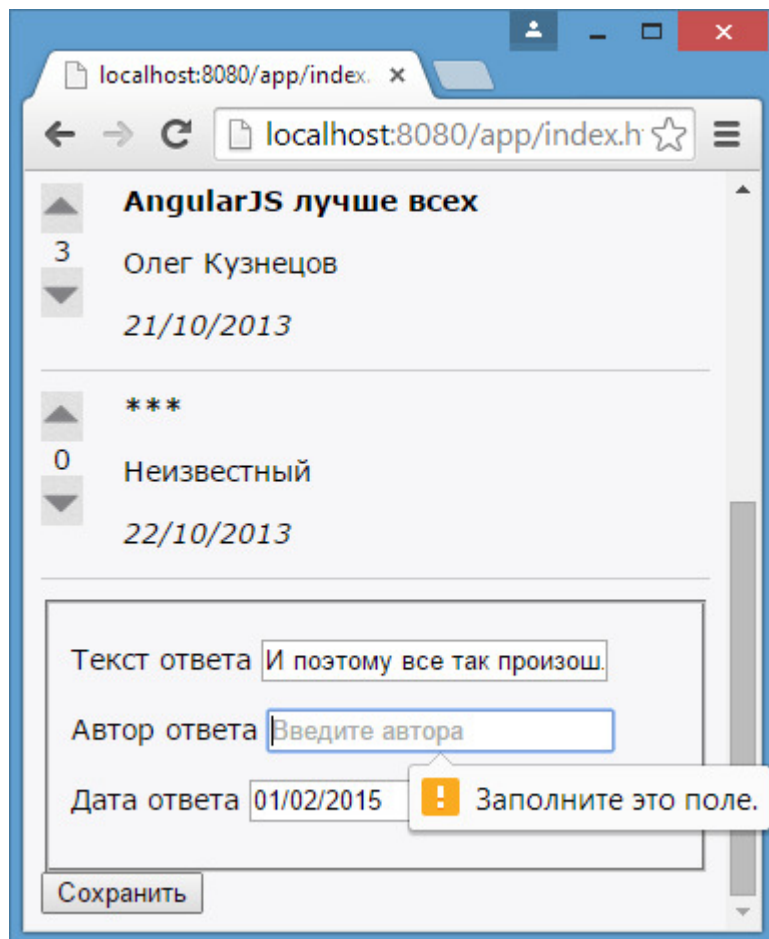
`ng-click="save(answer, answerForm)"`, которое вызывает функцию `save` контроллера `AnswerController`, передавая в нее в качестве параметров объект `answer`, создаваемый, полями формы, и объект формы `answerForm` (так как у элемента `form` атрибут `name="answerForm"`).

Теперь создадим простенький контроллер `AnswerController`:

```
questApp.controller('AnswerController',
  function AnswerController($scope){
    $scope.save = function (answer, answerForm){
      if(answerForm.$valid){
        // действия по сохранению
        alert(answer.author + ", ваш ответ сохранен");
      }
    };
  })
```

С помощью выражения `answerForm.$valid` просматриваем, прошла форма валидацию или нет.

Запустим и попробуем ввести различные значения:



Как видно на рисунке, поле для ввода автора не заполнено, поэтому срабатывает атрибут `required`, который помечает поле - в данном случае это сообщение о необходимости заполнить поле. В других браузерах действие может быть иным, например, в Firefox просто выделяет границу поля красным цветом.

Если все поля заполнены правильно, то контроллер отобразит нам сообщение. Если же, например, значение поля даты не будет соответствовать установленному для него регулярному выражению, то форма не пройдет валидацию и сообщение, естественно, не отобразится.

KÖP NU

[Назад](#) [Содержание](#) [Вперед](#)



G+1

8 Комментариев

metanit.com

 Войти Рекомендовать Поделиться

Лучшее в начале

**Михаил Летов** • 3 месяца назад

Как я понимаю, приведенный пример во многом основывается на валидации средствами html5, которую можно реализовать без Angular да и вообще без js. Было бы очень здорово увидеть какой-нибудь пример, аналогичный плагину jquery.validation, чтобы можно было кастомизировать внешний вид сообщений об ошибках, без этого дефолтного желтого квадратика с восклицательным знаком.

1 ^ | v • Ответить • Поделиться ›

**Андрей** ➔ Михаил Летов • 13 дней назад

Можно в форме выставить свойство "novalidate" и дефолтного желтого квадратика с восклицательным знаком не будет. А кастомизировать внешний вид невалидных форм в css, используя классы \$valid/\$invalid (автоматически присваиваются ангуляром)

^ | v • Ответить • Поделиться ›

**Nickolay Zuev** ➔ Михаил Летов • 22 дня назад

+1

^ | v • Ответить • Поделиться ›

**Nickolay Zuev** • 22 дня назад

Сделайте пример с валидацией числа с точкой, даты, времени пжс

^ | v • Ответить • Поделиться ›

**OK** • 5 месяцев назад

А как сделать, чтобы вообще нельзя было ввести в поле значение, не соответствующее маске или даже формату даты? 13-месяц чтобы нельзя было тоже указать.

^ | v • Ответить • Поделиться ›

**Metanit** Модератор ➔ OK • 5 месяцев назад

экспериментируйте с регулярными выражениями

^ | v • Ответить • Поделиться ›

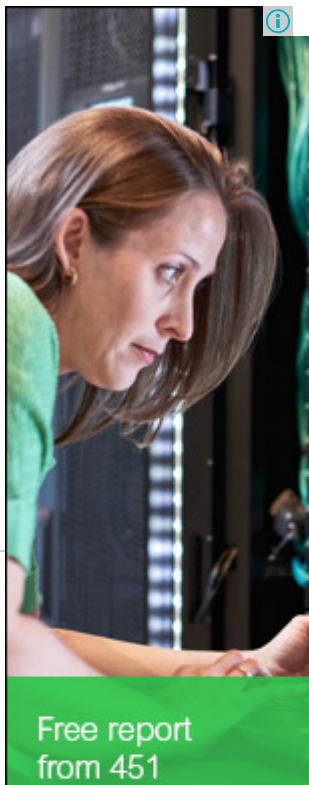
**Олег** • 6 месяцев назад

В этом примере ответ не сохраняется а просто проверяется правильно ли заполнена форма?

Привязка ng-model='answer.text' означает что объект answer с свойством text создается если данные формы были успешно отправлены?

^ | v • Ответить • Поделиться ›

**Metanit** Модератор ➔ Олег • 6 месяцев назад



[Google+](#) | [Канал сайта на youtube](#) | [Помощь сайту](#)

и: metanit22@mail.ru

t.com, 2012-2016. Все права защищены.