

Optimize Your JavaScript

Matt Honeycutt
<http://trycatchfail.com>
mbhoneycutt@gmail.com



pluralsight 
hardcore developer training

Our Goals

Standardize
Client/Server
Communication

Provide
Consistency
Between Client
and Server Code

Leverage
Conventions and
Strong-Typing

Drastically
Reduce
Boilerplate
Markup

Outline

Review Fail Tracker

Build A Better JsonResult Class

Build Client-Side Alert Service

Globally Handle Client-Side Errors

Easily Access C# View Models In JavaScript

Build A Convention-Driven AngularJS Editor Template

AngularJS



AngularJS Docs: <http://goo.gl/QPluN>

AngularJS Fundamentals Course: <http://goo.gl/qVO>

Standardizing Client/Server Communication

Client (JS)

Server (C#)

AJAX Request

```
{
  "issueID":6,
  "creatorUserName":"DougStone",
  "subject":"Viewing details crashes",
  "assignedToUserName":"Matt",
  "createdAt":"2/21/2014 4:45:17 PM",
  "issueType":"Enhancement",
  "body":"Sometimes, ..."
}
```

Model Binder



```
public class EditIssueForm
    : IMapFrom<Domain.Issue>
{
    public int IssueID { get; set; }
    public string CreatorUserName { get; set; }
    public string Subject { get; set; }
    public IssueType IssueType { get; set; }
    public string AssignedToUserName { get; set; }
    public string Body { get; set; }
}
```

Response

JsonResult



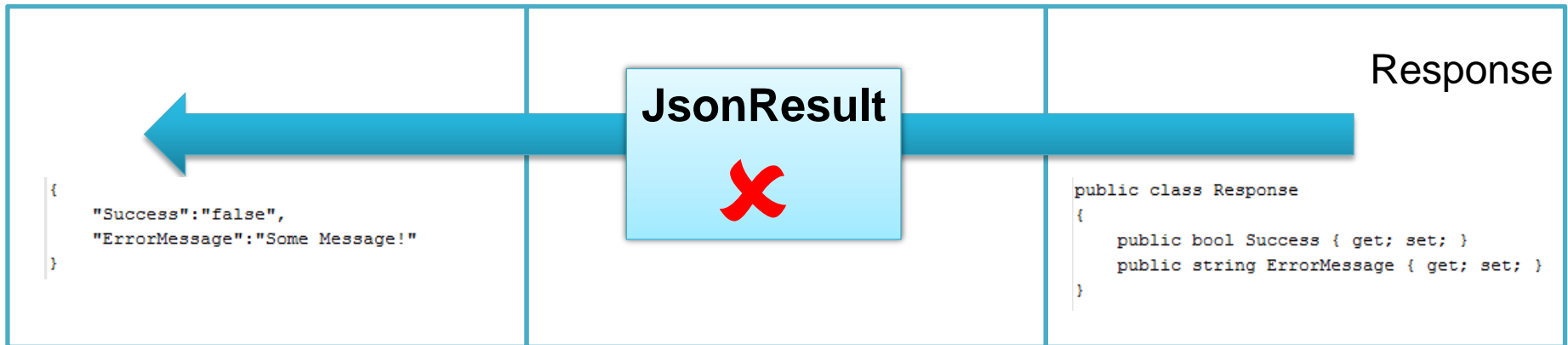
```
{
  "Success":"false",
  "ErrorMessage":"Some Message!"
}
```

```
public class Response
{
    public bool Success { get; set; }
    public string ErrorMessage { get; set; }
}
```

Standardizing Client/Server Communication

Client (JS)

Server (C#)



```
$http.post("/Issue/Edit", $scope.issue)
  .success(function (data) {
    if (data.success !== true) {
      alert("There was a problem saving to the se
      return;
    }

    $scope.originalIssue = angular.extend({}, $scope.issue);

    $scope.editing = false;
  })
  .error(function() {
    alert("There was a problem saving the issue.
  });
```

Model Validation Errors
Other Business Errors
(HTTP 200 Response)

Unexpected Errors
(HTTP 400 or 500 Response)

A Client-Side Alert Service

ASP.NET MVC

[Fail Tracker](#) [Home](#) [New Issue](#) Hello Matt! [Log off](#)

Issue created!

Subject	Type	Created At	Created By
Viewing details crashes	Enhancement	2/21/2014	DougStone



[Fail Tracker](#) [Home](#) Hello Matt! [Log off](#)

Edit Issue

Creator User Name
DougStone

The page at localhost:49937 says:
There was a problem saving the issue. Please try again.

A Client-Side Alert Service

ASP.NET MVC

Fail Tracker Home New Issue Hello Matt! Log off

As Issue created!

Subject	Type	Created At	Created By
Viewing details crashes	Enhancement	2/21/2014	DougStone



Fail Tracker Home

The page at localhost:49937 says:
There was a problem saving the issue. Please try again.

OK

Hello Matt! Log off

Edit Issue

Creator User Name
DougStone

Fail Tracker Home New Issue Hello Matt! Log off

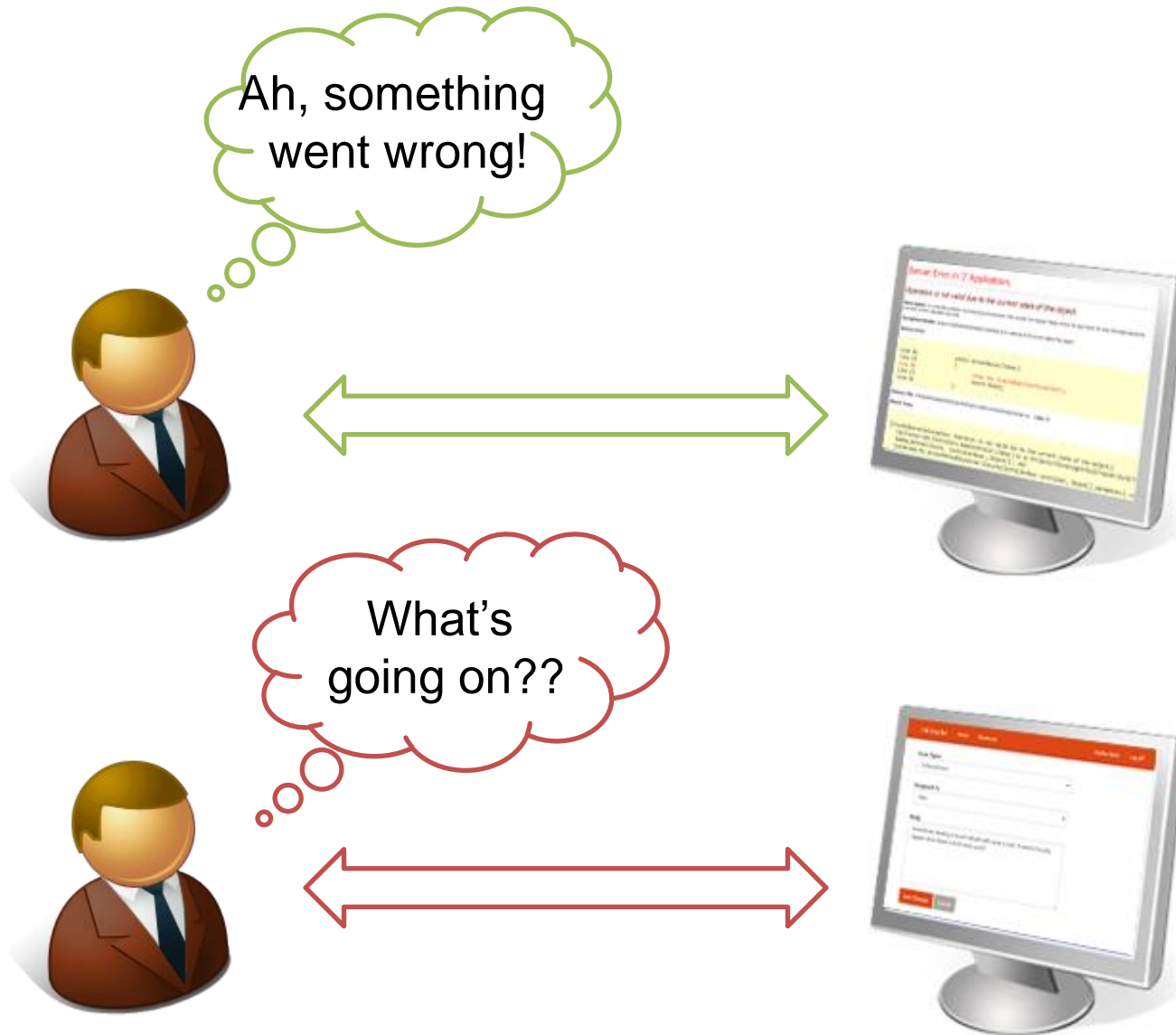
+ Your changes have been saved!

Currently assigned to Matt

Sometimes, viewing an issue's details will cause a crash. It seems to only happen when there is a full moon out!!!

Edit Delete Issue

Client-Side Error Handling

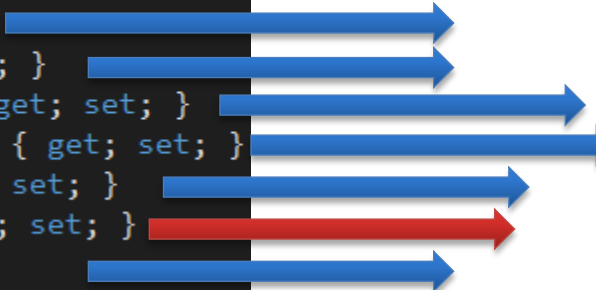


Easy-Access to C# View Models From JavaScript

```
<div ng-controller="editIssueController"  
  ng-init="init({issueID: @Model.IssueID, ... body: '@Model.Body.Replace("'", "'\''')'})">
```

(243 characters)

```
public class IssueDetailsViewModel  
  : IMapFrom<Domain.Issue>  
{  
  public int IssueID { get; set; }  
  public string Subject { get; set; }  
  public string CreatorUserName { get; set; }  
  public string AssignedToUserName { get; set; }  
  public DateTime CreatedAt { get; set; }  
  public IssueType IssueType { get; set; }  
  public string Body { get; set; }  
}
```



```
{  
  issueID  
  subject  
  creatorUserName  
  assignedToUserName  
  createdAt  
  issueTpye  
  body  
}
```

Easy-Access to C# View Models From JavaScript

BORIN

```
public class NewModel  
: IMapFrom<Issue>
```

```
{  
    public int IssueID { get; set; }  
    public string Subject { get; set; }  
    public string CreatorUserName { get; set; }  
    public string AssignedToUserName { get; set; }  
    public DateTime CreatedAt { get; set; }  
    public IssueType IssueType { get; set; }  
    public string Body { get; set; }  
}
```

```
{  
    issueID  
    subject  
    creatorUserName  
    assignedToUserName  
    createdAt  
    issueTpye  
    body  
}
```

G

Wrapping It Up: An AngularJS Editor Template

After Module 4...

```
<h2>Edit Issue</h2>
@using(Html.BeginForm("Edit", "Issue", FormMethod.Post, new{@class="form-horizontal"}))
{
    @Html.AntiForgeryToken()
    @Html.ValidationSummary(true)
    @Html.EditorForModel()

    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" value="Save Changes" class="btn btn-default" />
        </div>
    </div>
}
```

Wrapping It Up: An AngularJS Editor Template

Now...

```
<div class="form-horizontal" ng-cloak ng-hide="!editing">
  <h2>Edit Issue</h2>
  <div class="form-group">
    <label class="col-md-2 control-label">Creator User Name</label>
    <div class="col-md-10">
      <p class="form-control-static">{{issue.creatorUserName}}</p>
    </div>
  </div>
  <div class="form-group">
    <label class="col-md-2 control-label" for="Subject">Subject</label>
    <div class="col-md-10">
      <input class="form-control" required id="Subject" name="Subject" placeholder="Subject..." type="text">
    </div>
  </div>
  <div class="form-group">
    <label class="col-md-2 control-label" for="IssueType">Issue Type</label>
    <div class="col-md-10">
      <select class="form-control" id="IssueType" name="IssueType" ng-model="issue.issueType">
        <@foreach (var option in issueTypeOptions)>
          <option value="@option.Value">@option.Text</option>
        </@foreach>
      </select>
    </div>
  </div>
  <div class="form-group">
    <label class="col-md-2 control-label" for="AssignedToID">Assigned To</label>
    <div class="col-md-10">
      <select class="form-control" id="AssignedToID" name="AssignedToID" ng-model="issue.assignedToUserName">
        <@foreach (var option in availableUserOptions)>
          <option value="@option.Value">@option.Text</option>
        </@foreach>
      </select>
      <span class="field-validation-valid" data-valmsg-for="AssignedToID" data-valmsg-replace="true"></span>
    </div>
  </div>
```

Wrapping It Up: An AngularJS Editor Template

Our Goal...

```
<div class="form-horizontal" ng-cloak ng-hide="!editing">
  <h2>Edit Issue</h2>

  @Html.AngularEditorForModel("issue")

  <div class="form-group">
    <div class="col-md-offset-2 col-md-10">
      <button class="btn btn-primary" ng-click="save()">Save Changes</button>
      <button class="btn btn-default" ng-click="cancel()">Cancel</button>
    </div>
  </div>
</div>
```

Summary



Assessed The “Damage”



Built A Better JsonResult Class



Built A Client-Side Alert Service



Globally-Handled JavaScript Errors



Streamlined C# View Model to JSON Conversion



Created An AngularJS Editor Template

Before & After

```
<div class="form-horizontal" ng-cloak ng-hide="!editing">
  <h2>Edit Issue</h2>
  <div class="form-group">
    <label class="col-md-2 control-label">Creator User Name</label>
    <div class="col-md-10">
      <p class="form-control-static">{{issue.creatorUserName}}</p>
    </div>
  </div>
  <div class="form-group">
    <label class="col-md-2 control-label" for="Subject">Subject</label>
    <div class="col-md-10">
      <input class="form-control" required id="Subject" name="Subject" placeholder="Subject..." type="text">
    </div>
  </div>
  <div class="form-group">
    <label class="col-md-2 control-label" for="IssueType">Issue Type</label>
    <div class="col-md-10">
      <select class="form-control" id="IssueType" name="IssueType" ng-model="issue.issueType">
        <foreach (var option in issueTypeOptions)>
          {
            <option value="@option.Value">@option.Text</option>
          }
        </select>
      </div>
    </div>
  <div class="form-group">
    <label class="col-md-2 control-label" for="AssignedToID">Assigned To</label>
    <div class="col-md-10">
      <select class="form-control" id="AssignedToID" name="AssignedToID" ng-model="issue.assignedToUserName">
        <foreach (var option in availableUserOptions)>
          {
            <option value="@option.Value">@option.Text</option>
          }
        </select>
      <span class="field-validation-valid" data-valmsg-for="AssignedToID" data-valmsg-replace="true"></span>
    </div>
  </div>
```


Before & After

```
<div class="form-horizontal" ng-cloak ng-hide="!editing">
  <h2>Edit Issue</h2>

  @Html.AngularEditorForModel("issue")

  <div class="form-group">
    <div class="col-md-offset-2 col-md-10">
      <button class="btn btn-primary" ng-click="save()">Save Changes</button>
      <button class="btn btn-default" ng-click="cancel()">Cancel</button>
    </div>
  </div>
</div>
```

The End... For Now?

**Focus On
Your Pain**

**Solve Your
Pain**

Start Now!

**Establish
Cleanup
Rules**

**GROW Your
Framework**

Build Your Own Application Framework With ASP.NET MVC 5

Matt Honeycutt
<http://trycatchfail.com>
mbhoneycutt@gmail.com



pluralsight 
hardcore developer training