

<form.io>

BUILDING A NEXT-GEN WEB APPLICATION



uses a form builder to create both Forms & API's.

The screenshot shows the form.io interface for editing a "User" resource. The top navigation bar includes the form.io logo, the project name "MovieApp > User", and a user icon with "travis". Below the header, there are tabs for "View", "Edit" (which is active), "Actions", and "Submissions".

The main area displays the "User" form fields:

- Title:** A text input field containing "User".
- Name:** A text input field containing "user".
- API Path:** A text input field containing "https://movieapp.form.io/app/api/ user". A tooltip below it says "The REST API path to this form."

On the left, a sidebar titled "Form Components" lists various field types with their corresponding icons:

Text Field	Address
Button	Check Box
Content	Date / Time
Email	Field Set
Hidden	Number
Password	Phone Number
Radio	Resource
Select	Signature
Text Area	

On the right, the "Page 1" section contains three text input fields with red asterisks indicating required fields:

- Full Name:** "Enter your full name"
- Email:** "Enter your email address"
- Password:** "Enter your password"

A "Submit" button is located at the bottom of the page.



creates both the interface and the API in one step

The screenshot shows the form.io interface for creating a user form. At the top, the navigation bar includes the form.io logo, the project name "MovieApp > User", and a user icon labeled "travis". Below the navigation, the form title "User" is displayed with a "RESOURCE" badge. To the right are buttons for "View" (highlighted in blue), "Edit", "Actions", and "Submissions".

The main area contains three input fields: "Full Name" (placeholder: "Enter your full name"), "Email" (placeholder: "Enter your email address"), and "Password" (placeholder: "Enter your password"). Each field has a red asterisk indicating it is required. A "Submit" button is located at the bottom left.

To the right of the form, a sidebar titled "Form API" displays the generated API endpoint: <https://movieapp.form.io/app/api/user>. It also includes links for "Application Embedding" and "API Specification".



automatically creates the API documentation

The screenshot shows the form.io interface for a 'User' resource. On the left, there's a 'User' form with fields for 'Full Name', 'Email', and 'Password'. A blue arrow points from the text 'Swagger.IO API documentation!' to the generated API documentation on the right.

**User RESOURCE**

Preview

Full Name

Enter your full name \*

Email

Enter your email address \*

Password

Enter your password \*

Submit

Swagger.IO API documentation!

View Edit Actions Submissions

Form API

Application Embedding

API Specification

**https://api.form.io**

Form.io API: The following is the API specification for <https://api.form.io>.

Created by support@form.io  
MIT

**form** Show/Hide | List Operations | Expand Operations

GET	/app/:appId/form	Find form resources.
POST	/app/:appId/form	Create a new form
GET	/app/:appId/form/{formId}	Return a specific form instance.
PUT	/app/:appId/form/{formId}	Update a specific form instance.
DELETE	/app/:appId/form/{formId}	Delete a specific form

**submission** Show/Hide | List Operations | Expand Operations

# <form.io>

provides you with a <formio> tag to embed your form

The screenshot shows the form.io application interface. On the left, there's a preview of a 'User' form with fields for 'Full Name', 'Email', and 'Password'. On the right, there's a sidebar with sections for 'Form API', 'Application Embedding', and 'API Specification'. The 'Application Embedding' section contains instructions for adding the form to an application using Bower and provides a command-line example: `bower install formio`. A blue arrow points from the text 'It's as easy as an <img> tag!' in the preview area to the Bower command in the sidebar.

**User RESOURCE**

View Edit Actions Submissions

Preview

Full Name

Enter your full name \*

Email

Enter your email address \*

Password

Enter your password \*

Submit

It's as easy as an `<img> tag!`

Form API

Application Embedding

To add this form to your application, use [Bower](#) to download the Form.IO library and its dependencies.

```
bower install formio
```

Next, ensure your webpage has all of the application dependencies added to the `<script>` tag. For a simple example, refer to the following example.

[Example Simple Form Embedding](#)

We recommend using a tool called [Wiredep](#) to automatically add all the dependencies to the `<script>` section.

Now, within your HTML source, you can use the `<formio>` directive like so.

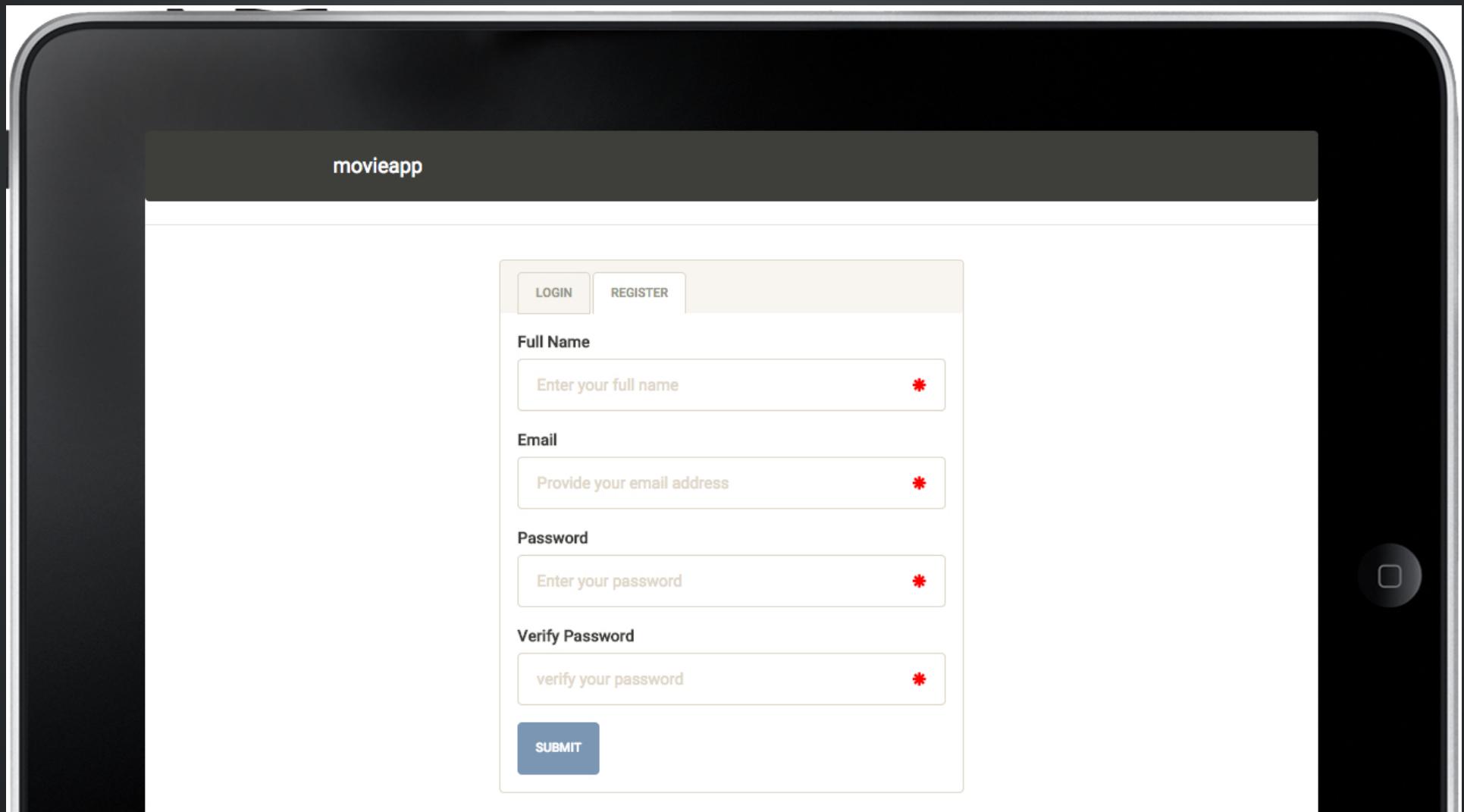
```
<formio src="'https://movieapp.form.io/app/api/user'"></formio>
```

API Specification

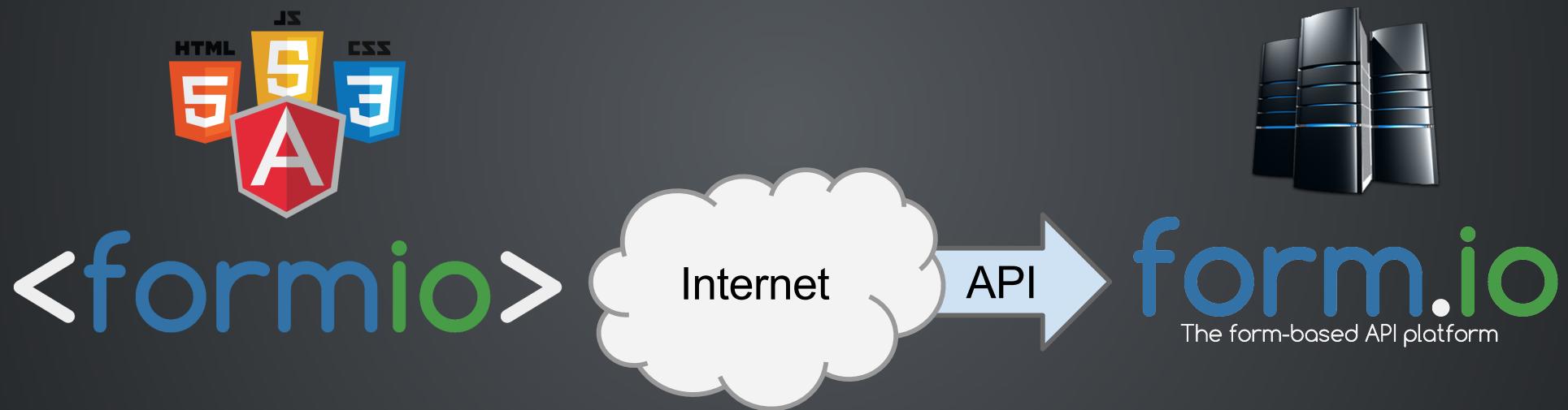
instead of <form>

form + I/O = <**formio** src="...">

# You build the app, we handle the **form.io**!



The <form.io> solution



Now let's build something with <**form.io**>!

Request a Private Beta Code @ <http://form.io>

Create your account @ <http://form.io/app>

The screenshot shows a registration form for Form.io. At the top, there are 'Login' and 'Register' buttons. A message box states: 'Form.IO is currently in **Private Beta**, which means you must provide a **Private Beta Code** in order to create an account and have access to our system.' Below this is a large green 'Request Beta Access' button. The main form area contains the following fields:

Field Label	Description
Private Beta Code	Enter your Private Beta Code *
Full name	Enter your full name *
Username	Enter your username. *
Email	Enter your email address *
Password	Enter your password *
Verify Password	Type your password again *

At the bottom is a blue 'Submit' button.

# Create a project

Welcome to <form.io>!

Form.IO is a platform that enables developers to rapidly build modern web and mobile applications. Our intuitive drag & drop interface allows you to create both the forms and the RESTful API's all in one easy step! **You build the app, we handle the Form I/O.**

Start the tutorial  
BUILDING A NEXT-GEN WEB APPLICATION

My Projects

Projects are web-based HTML software. Whether you are building a new website, or a mobile application, you can create separate Projects used for each of these products.

+ New Project

Click "New Project" to create a new project

# Provide project information.

NOTE: Your project will need a different domain name.

The screenshot shows the 'New Project' page on the form.io website. The top navigation bar includes the form.io logo, user account (travis), and Help link. The main form fields are:

- Project Title:** MovieApp
- Domain Name:**    ✓  
This domain is available!
- Description:** An example movie application project

At the bottom are 'Cancel' and 'Create Project' buttons. To the right, a sidebar titled 'Best Practices' provides tips:

- Enter a title and description for your Project.
- You will need a unique sub domain for your Project. This is where the API endpoints will be hosted on.
- Don't worry though, we'll take care of the hard parts.
- Still need help? Check out the [complete documentation](#).

# Building the User resource.

Click on **New Resource** on your project home page.

The screenshot shows the form.io interface for a project named "MovieApp". The top navigation bar includes the form.io logo, the project name "MovieApp", a "travis" status indicator, and "Help" links. Below the header, the page title is "MovieApp" with the subtitle "An example movie application project". On the left, there are two main sections: "Resources" and "Forms". The "Resources" section contains a box with the text: "Resources are the objects within your Project. Example: User, Company, Vehicle, etc." and a blue "+ New Resource" button. The "Forms" section contains a box with the text: "Forms serve as a composite input interface for Resources as well as free-form user input. Example: Login Form, Contact Form, etc." and a blue "+ New Form" button. A blue arrow points from the text "Click on \"New Resource\" to create a new User resource." to the "+ New Resource" button. To the right of these sections is a sidebar titled "Getting started with your Project" which provides an overview of the Form.IO platform and its resources. At the bottom right is a "Start the tutorial" button with the text "BUILDING A NEW WEB APPLICATION".

**MovieApp**  
An example movie application project

https://movieapp.form.io/ Edit Settings

**Resources**

Resources are the objects within your Project. Example: User, Company, Vehicle, etc.

+ New Resource

**Forms**

Forms serve as a composite input interface for Resources as well as free-form user input. Example: Login Form, Contact Form, etc.

+ New Form

**Getting started with your Project**

Form.IO provides the forms and resource structure behind your custom application. Start by creating the **Resources**, and then create the free-form **Forms** that your application needs.

Once you have created the Resources and Forms, you can embed them within your application using the <formio> Angular.js directive. It works just like an <img> tag, but used for Form I/O.

It is recommended to walk through the tutorial, which provides a step-by-step tutorial on how to build an Project with Form.IO.

**Start the tutorial**  
BUILDING A NEW WEB APPLICATION

For more information, check out the [complete documentation](#).

# Provide the title, name, and API path for this resource.

The screenshot shows the form.io interface for a 'MovieApp > User' project. The main title is 'MovieApp' with the subtitle 'An example movie application project'. The URL is https://movieapp.form.io/. There are 'Edit' and 'Settings' links. The current view is 'New Resource'.

**Title:** User

**Name:** user

**API Path:** https://movieapp.form.io/ user

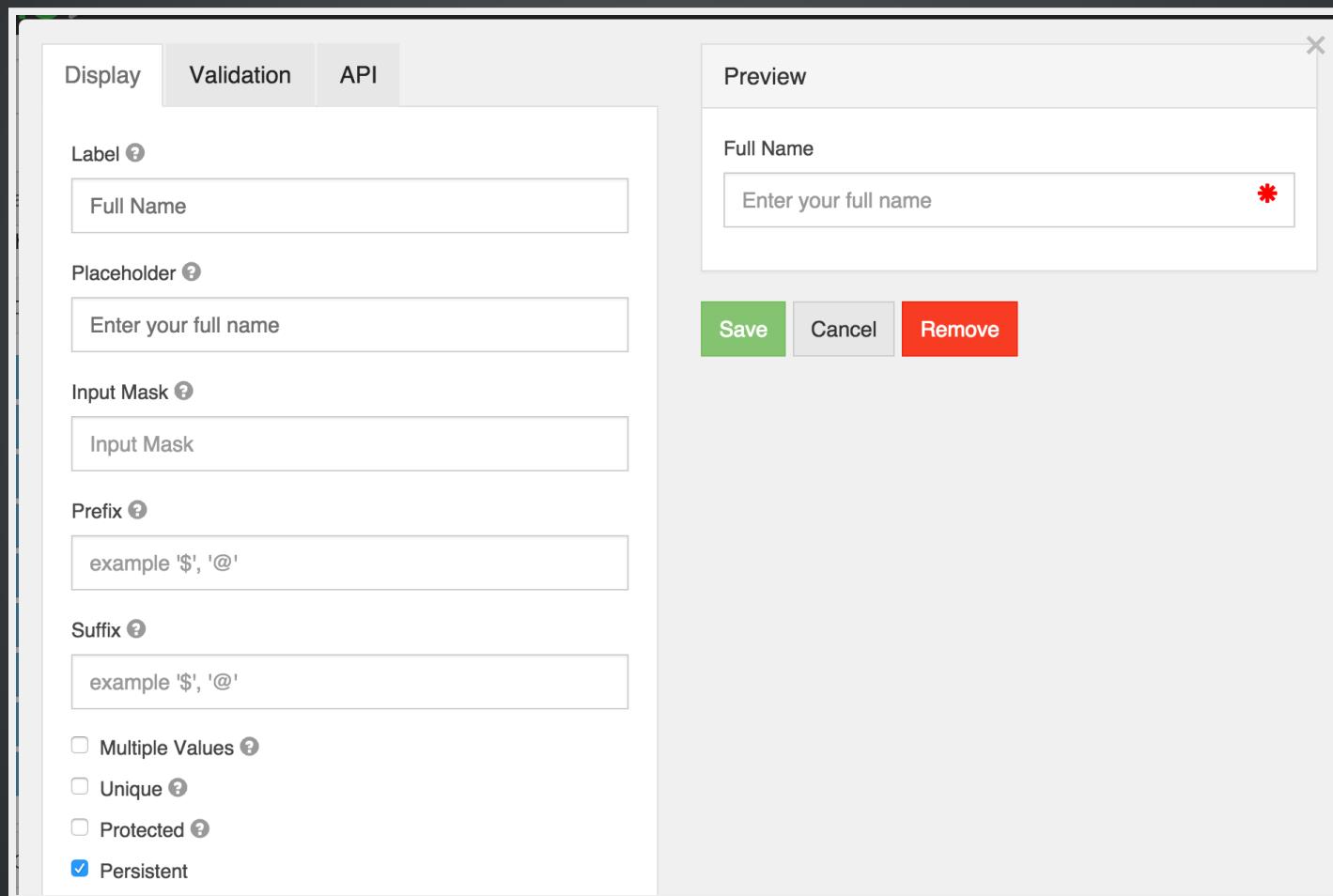
The REST API path to this form.

**Advanced Settings:** A collapsed section.

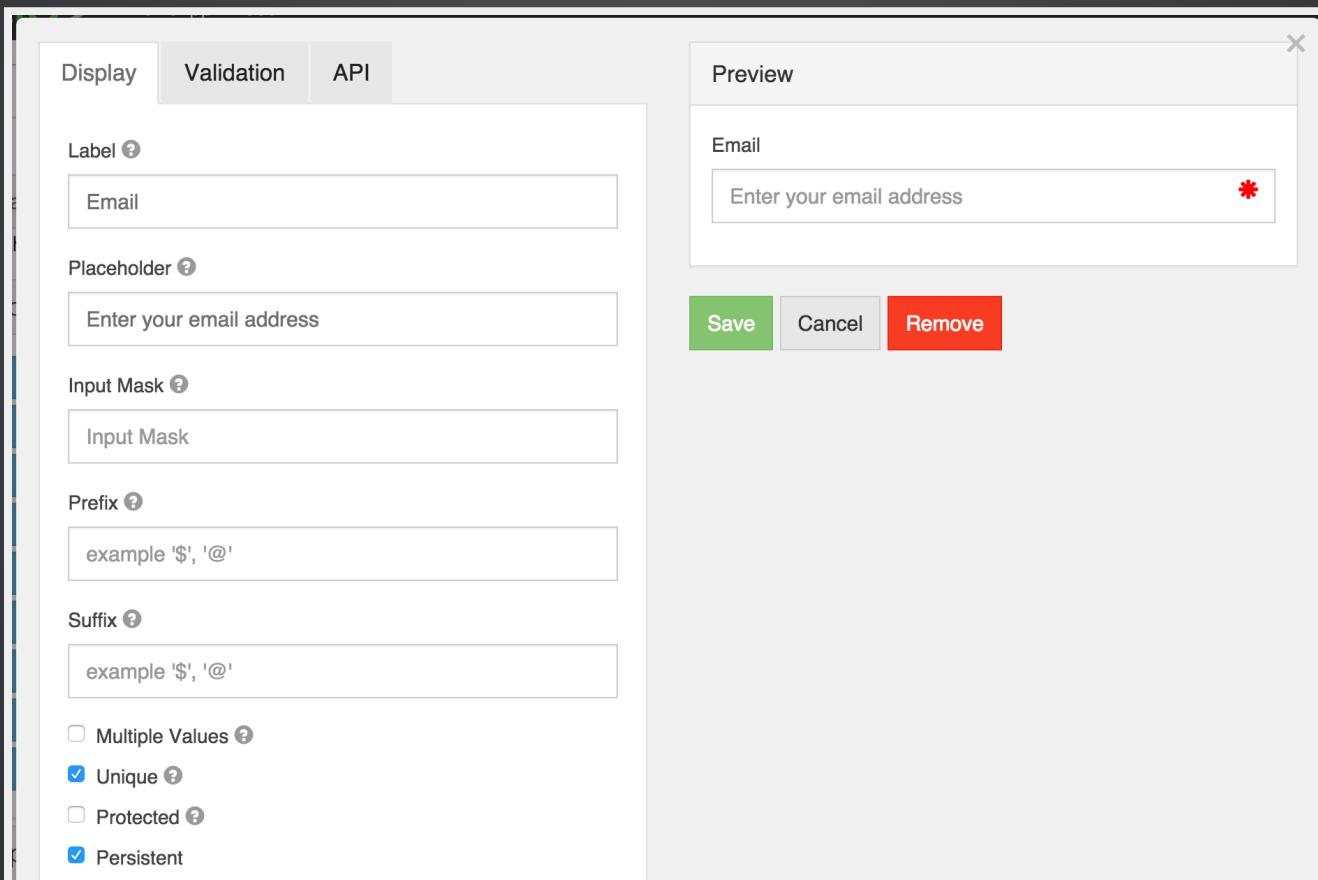
**Form Components:** A sidebar with a grid of components: Text Field, Address, Button, Check Box, Content, Date / Time, Email, and Field Set.

**Page 1:** A page editor with a 'Drag and Drop a form component' area containing a 'Submit' button.

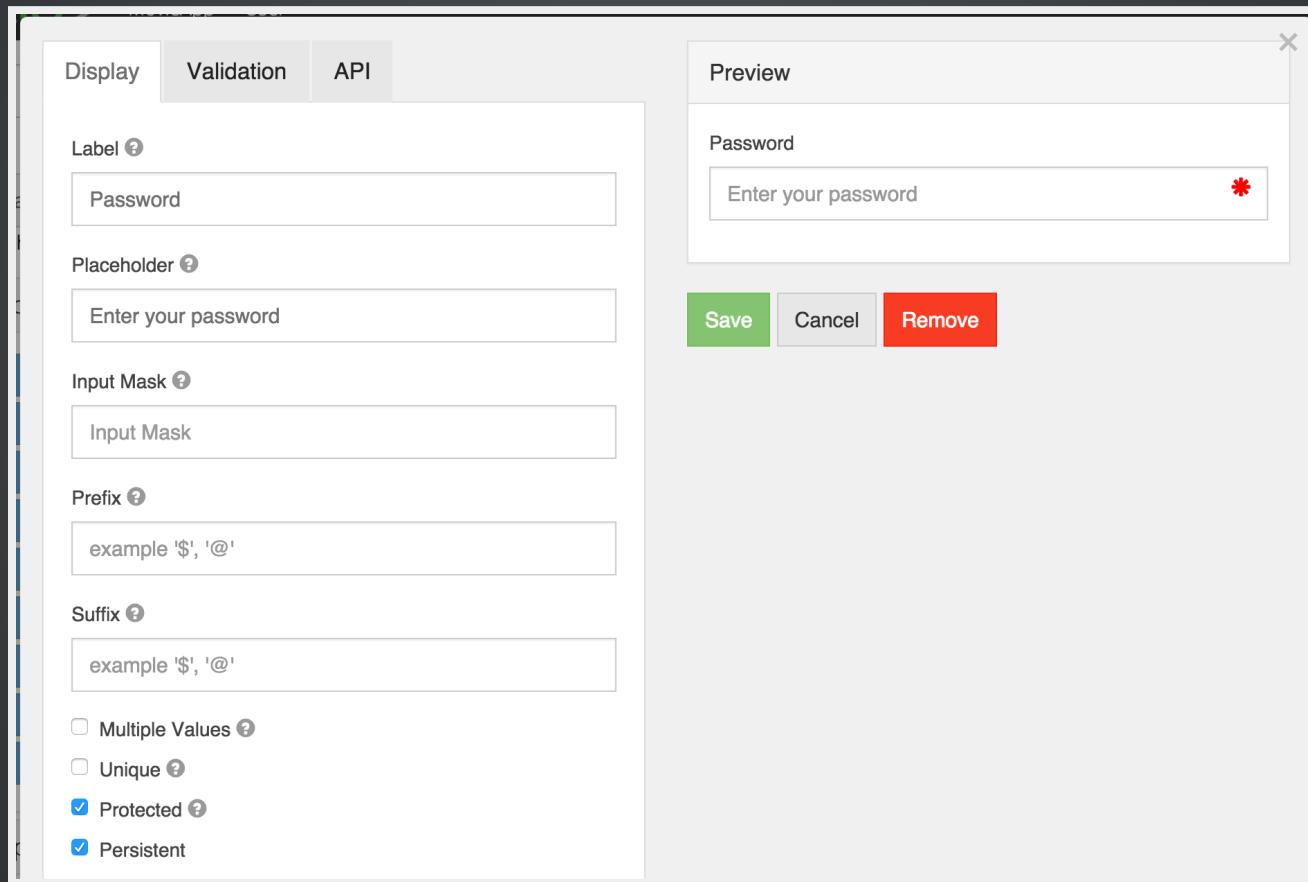
- Drag & Drop a **Text Field** field onto your form.
- Label it **Full Name**
- Check **Required** under Validation tab



- Drag & Drop a Email field onto your form.
- Check Unique
- Check Required under Validation tab



- Drag & Drop a Password field onto your form.
- Check Required



# Click Save and You now have a User resource w/ corresponding API

The screenshot shows the form.io interface for a "User" resource. At the top, there's a navigation bar with the form.io logo, the project name "MovieApp > User", and user account and help links. Below the header, the word "User" is followed by a "RESOURCE" badge. To the right are tabs for "View" (which is active), "Edit", "Actions", "Permissions", and "Submissions".

The main area is divided into two sections: "Preview" on the left and "Form API" on the right.

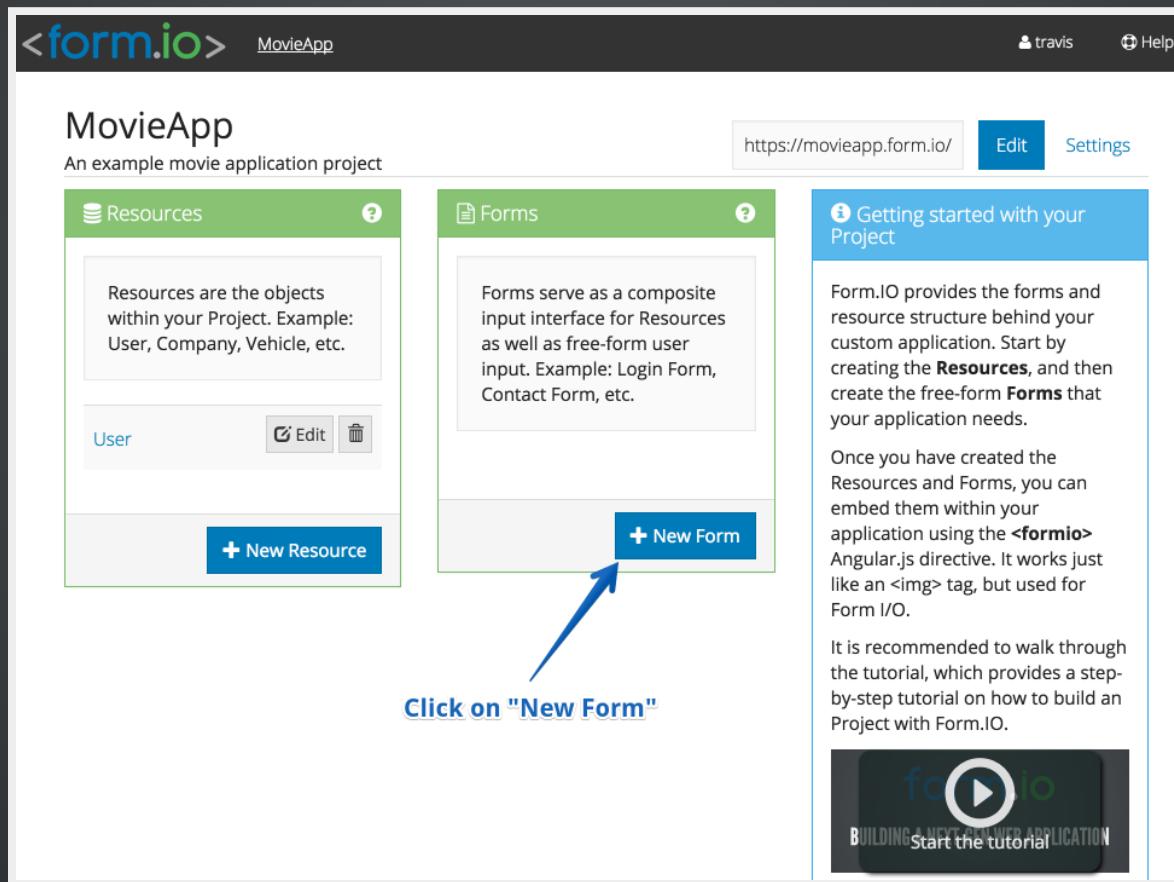
**Preview:** This section contains three input fields with validation stars: "Full Name \*", "Email \*", and "Password \*". Each field has a placeholder text below it: "Enter your full name", "Enter your email", and "Enter your password". A large blue "Submit" button is at the bottom of this section.

**Form API:** This section is titled "Form API" and contains the following text:  
**Your form API is now ready to use!** The following API was created that maps directly to this form interface.  
This API can be found by going to the following URL.  
`https://movieapp.form.io/user`

Below the API section are two more blue buttons labeled "Project Embedding" and "API Specification".

# Building the User Login Form

Click on New Form on your project home page.



# Provide the Title and API path for this form.

**<form.io>** MovieApp > User Login Form

travis H

## New Form

Title  
User Login Form

Name  
userLoginForm

API Path

The REST API path to this form.

Advanced Settings

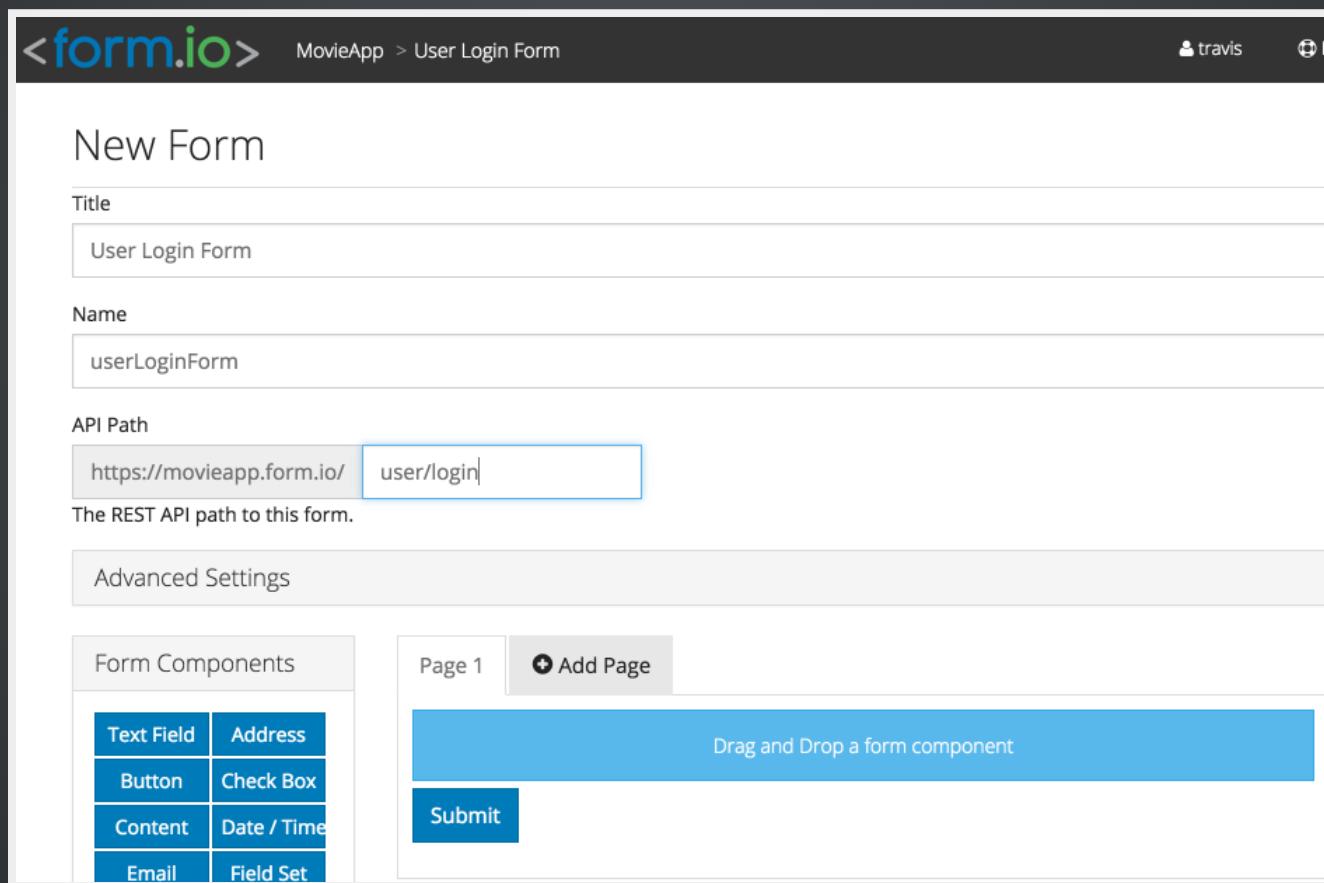
Form Components

Text Field	Address
Button	Check Box
Content	Date / Time
Email	Field Set

Page 1 + Add Page

Drag and Drop a form component

Submit



# Drag & Drop the User email and Password fields.

The screenshot shows the form.io interface for creating a "User Login Form".

**Form Components:** On the left, there are four categories: "Form Components", "Layout Components", "User Fields", and "Advanced Settings". Under "User Fields", two buttons are visible: "User Email" and "User Password". A blue arrow points from the "User Password" button towards the "User Password" field on the right side of the screen.

**Page 1:** This is the active page for the form. It contains the following fields:

- User Email:** A text input field labeled "Enter your email address" with a red asterisk (\*) indicating it is required.
- User Password:** A text input field labeled "Enter your password" with a red asterisk (\*) indicating it is required.
- Submit:** A blue button labeled "Submit" located at the bottom of the page.

**API Path:** The API path is set to `https://movieapp.form.io/app/api/ user/login`.

**Instructions:** A prominent instruction at the bottom of the page reads: "Drag and Drop the \"User\" resource fields onto this form to create the \"User Login Form\"".

**Buttons:** At the bottom left are "Save" and "Cancel" buttons.

# Click Save and You now have a User Login form w/ corresponding API

The screenshot shows the form.io interface for a "User Login Form". The top navigation bar includes the form.io logo, the project name "MovieApp > User Login Form", user authentication ("travis"), and help links. The main content area has tabs for "View" (selected), "Edit", "Actions", "Permissions", and "Submissions". On the left, a "Preview" section contains fields for "User Email \*" and "User Password \*", each with an input placeholder and a red asterisk indicating required fields. A "Submit" button is at the bottom. On the right, a "Form API" section informs the user that the API is ready, provides the URL <https://movieapp.form.io/user/login>, and links to "Project Embedding" and "API Specification".

# Enable Anonymous submissions by clicking on Permissions and add Anonymous to Create Own Submissions

<form.io> MovieApp > User Login Form

User Login Form FORM View Edit Actions **Permissions** Submissions

User Permissions

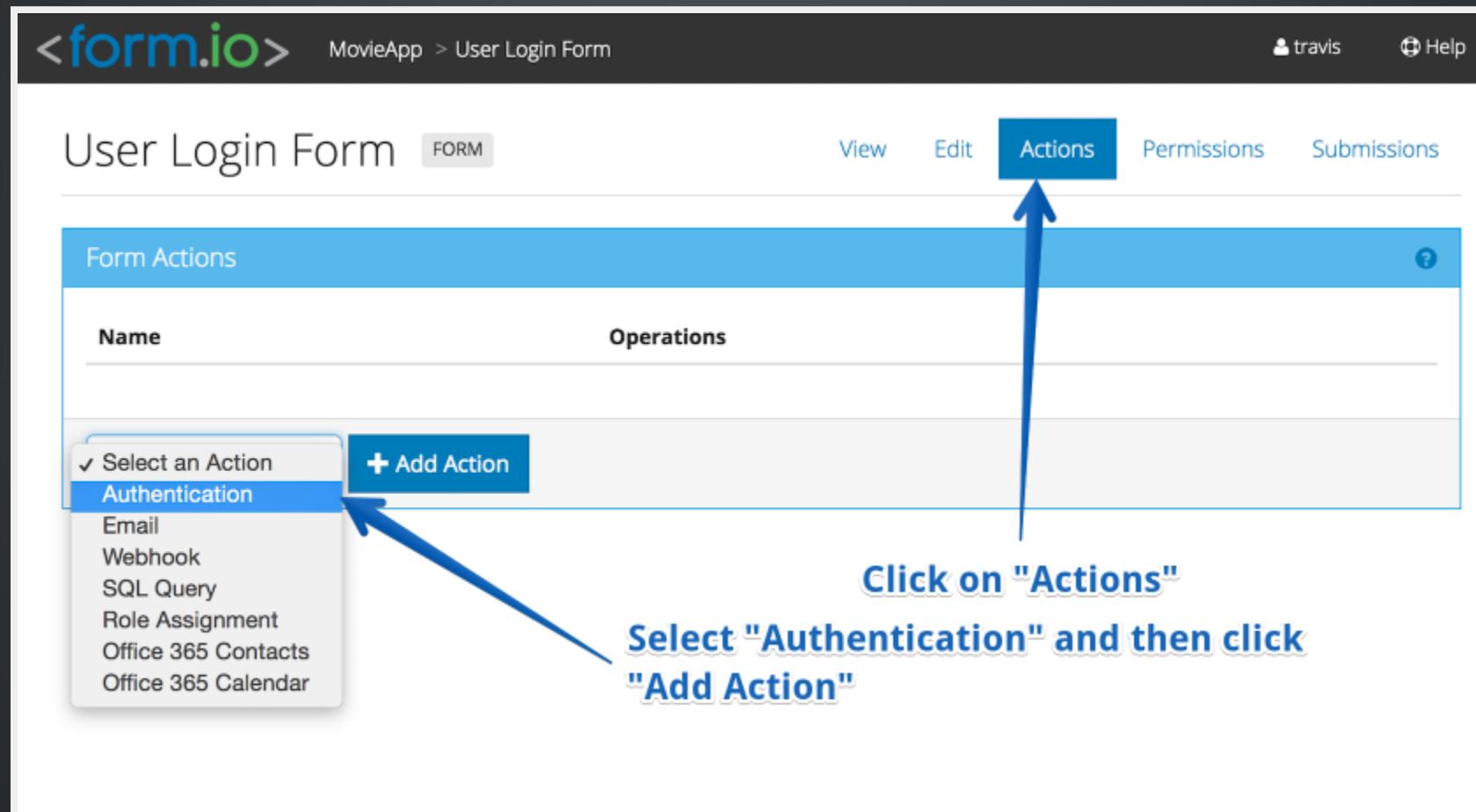
User Permissions allow you to control who can create, view, and modify form submissions. Here you may assign Project Roles to permissions. **Roles** can be created and edited in [Project Settings](#).

Permission	Roles
Read All Submissions	Select roles...
Update All Submissions	Select roles...
Delete All Submissions	Select roles...
Create Own Submissions	Anonymous <input type="button" value="X"/>
Read Own Submissions	Select roles...
Update Own Submissions	Select roles...
Delete Own Submissions	Select roles...

**Click on "Permissions" tab and then add the "Anonymous" role to the "Create Own Submissions"**

Cancel Save Changes

To handle Authentication, click on the **Actions** tab  
Select **Authentication** action and then click **Add Action**



# Select Existing Resource, Email, and Password

The screenshot shows the form.io interface for a 'User Login Form'. At the top, there's a header with the form.io logo, the project name 'MovieApp > User Login Form', and user information ('travis'). Below the header, the title 'User Login Form' is displayed, along with tabs for 'View', 'Edit', 'Actions' (which is selected), 'Permissions', and 'Submissions'. A horizontal line separates this from the main content area. The content area is titled 'Settings' and contains several configuration sections:

- Resource Association**: A section labeled 'Existing Resource'.
- Role (Added for new Resources only)**: A section labeled 'Select the Role that will be added to new Resources.'
- Username Field**: A section labeled 'User Email'.
- Password Field**: A section labeled 'User Password'.

At the bottom left of the content area is a blue 'Submit' button.

# Clicking Submit will now create the Login Auth action!

The screenshot shows the form.io interface for a 'User Login Form' under the 'MovieApp' project. The top navigation bar includes the form.io logo, user 'travis', and help documentation. Below the header, the form title 'User Login Form' is displayed, along with tabs for 'View', 'Edit', 'Actions' (which is highlighted in blue), 'Permissions', and 'Submissions'. The main content area is titled 'Form Actions' and contains a table with two columns: 'Name' and 'Operations'. One row in the table is for an action named 'Authentication', which has edit and delete icons in its 'Operations' column. At the bottom left, there is a dropdown menu labeled 'Select an Action' and a blue button labeled '+ Add Action'.

Name	Operations
Authentication	

Select an Action + Add Action

# Building the User Registration Form

Click on New Form on your project home page.

The screenshot shows the form.io interface for a project named "MovieApp". The top navigation bar includes the form.io logo, the project name "MovieApp", user information ("travis"), and links for "Edit" and "Settings". Below the header, there are three main sections: "Resources", "Forms", and "Getting started with your Project". The "Forms" section contains a "User Login Form" and a blue button labeled "+ New Form". A blue arrow points from the text "Click 'New Form' to create the User Registration Form" to the "+ New Form" button. The "Getting started with your Project" section provides an overview of how Form.IO integrates with custom applications.

<form.io> MovieApp

travis Help

MovieApp  
An example movie application project

https://movieapp.form.io/ Edit Settings

Resources

Forms

Getting started with your Project

Resources are the objects within your Project. Example: User, Company, Vehicle, etc.

User Edit Delete

+ New Resource

Forms serve as a composite input interface for Resources as well as free-form user input. Example: Login Form, Contact Form, etc.

User Login Form Edit Delete

+ New Form

Click "New Form" to create the User Registration Form

Form.IO provides the forms and resource structure behind your custom application. Start by creating the **Resources**, and then create the free-form **Forms** that your application needs.

Once you have created the Resources and Forms, you can embed them within your application using the <formio> Angular.js directive. It works just like an img tag, but used for Form I/O.

It is recommended to walk through the tutorial, which provides a step-by-step tutorial on how to build an Project with Form.IO.

Start the tutorial

# Provide a title and API path for the Register form.

The screenshot shows the form.io interface for creating a new form. The top navigation bar includes the form.io logo, the project name "MovieApp > User Registration Form", and user account information for "travis".

The main area is titled "New Form". It contains three input fields:

- Title:** "User Registration Form"
- Name:** "userRegistrationForm"
- API Path:** "https://movieapp.form.io/user/register" (highlighted with a blue border)

Below these fields, a note states: "The REST API path to this form." A "Advanced Settings" button is available for further configuration.

On the left, a "Form Components" sidebar lists various component types: Text Field, Address, Button, Check Box, Content, Date / Time, Email, and Field Set. On the right, the "Page 1" section displays a placeholder message: "Drag and Drop a form component" and a "Submit" button.

# Drag and Drop the User resource fields.

The screenshot shows the form.io interface for creating a User Registration Form. The top navigation bar includes the form.io logo, the project name "MovieApp > User Registration Form", and user account information for "travis".

Form settings:

- Title:** User Registration Form
- Name:** userRegistrationForm
- API Path:** https://movieapp.form.io/ user/register

The REST API path to this form.

**Advanced Settings:**

**Form Components:** A sidebar menu with options: Form Components, Layout Components, and User Fields. The "User Fields" section contains three items: User Email, User Full Name, and User Password. The "User Full Name" item is highlighted with a blue border.

**Page 1:** The main workspace where the User Full Name field has been dropped. It is currently the only field on the page. The "Add Page" button is visible above the workspace.

**User Fields (Workspace):** The User Full Name field is displayed with the placeholder "Enter your full name".

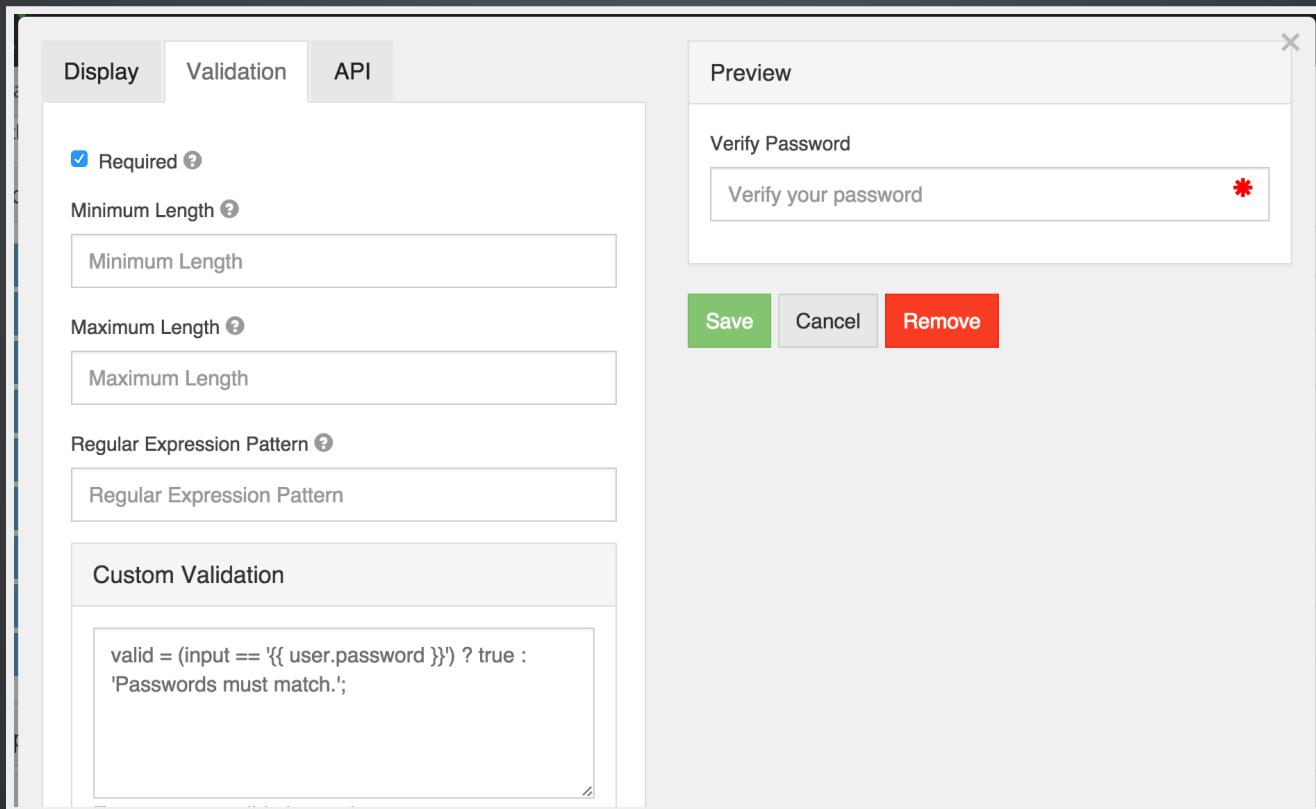
**User Fields (Form Components):** The User Email, User Full Name, and User Password fields are listed in the sidebar under the User Fields category.

**Submit:** A blue "Submit" button at the bottom of the workspace.

Three blue arrows point from the "User Fields" sidebar towards the "User Full Name" field in the workspace, indicating the process of dragging and dropping the field.

# Implement Verify Password by adding a new Password field

Add custom validation to match the Password field.



# Click Save and You now have a User Registration form w/ corresponding API

The screenshot shows the form.io interface for a "User Registration Form". The top navigation bar includes the form.io logo, the project name "MovieApp > User Registration Form", and user account information ("travis"). Below the header, the main title is "User Registration Form" with a "FORM" button. To the right are links for "View", "Edit", "Actions", "Permissions", and "Submissions".

The left side of the interface displays the "Preview" of the form, which contains five fields:

- User Full Name \*
- User Email \*
- User Password \*
- Verify Password \*

Each field has a placeholder text box below it. At the bottom of the preview section is a blue "Submit" button.

The right side of the interface is titled "Form API" and contains the following information:

- Your form API is now ready to use!** The following API was created that maps directly to this form interface.
- This API can be found by going to the following URL: <https://movieapp.form.io/user/register>
- Links for "Project Embedding" and "API Specification" are also present.

# Enable Anonymous submissions by clicking on Permissions and add Anonymous to Create Own Submissions

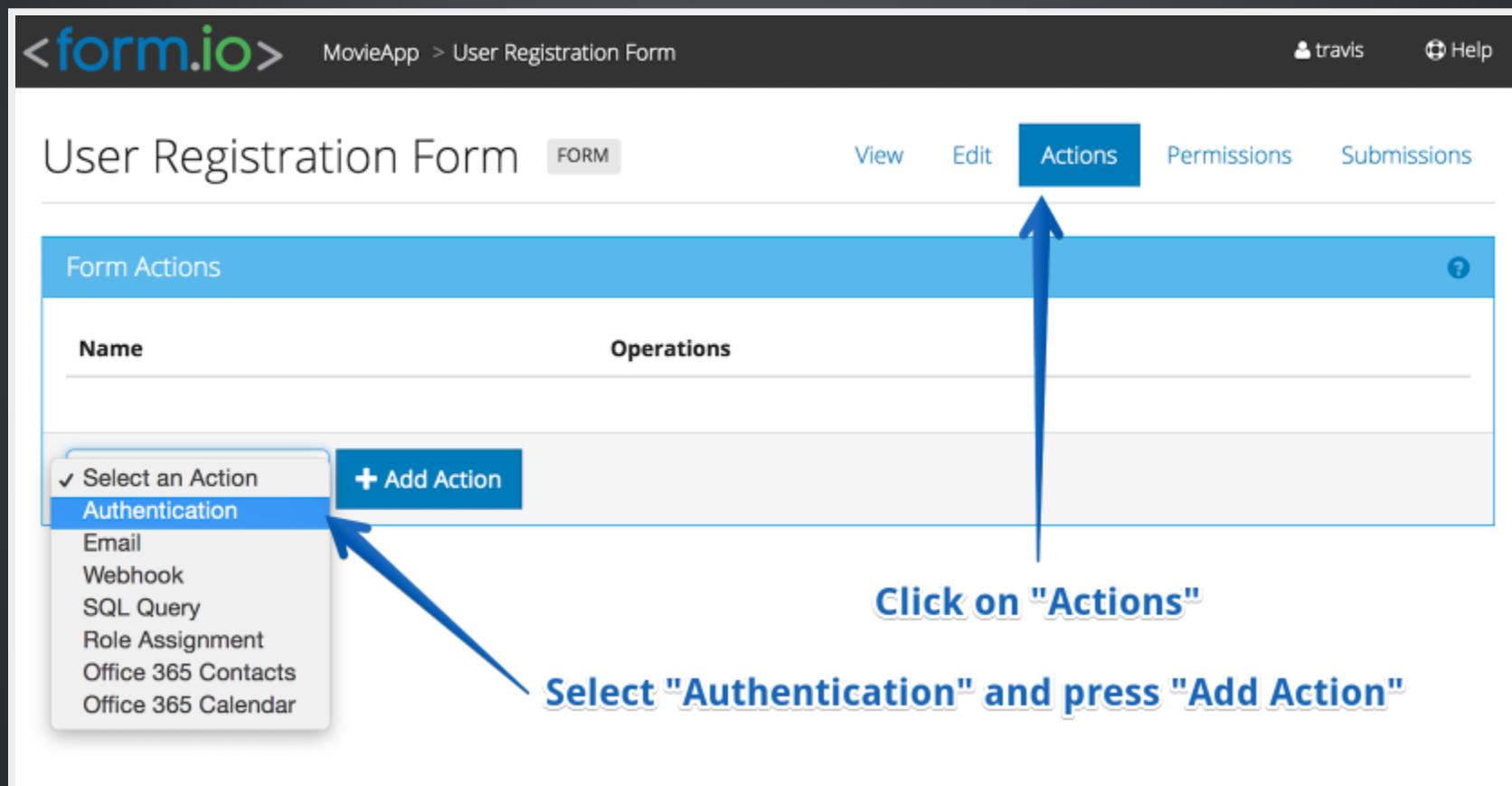
The screenshot shows the 'User Registration Form' settings page on the form.io platform. The top navigation bar includes 'MovieApp > User Registration Form', a user profile icon for 'travis', and a 'Help' link. Below the bar, there are tabs for 'View', 'Edit', 'Actions', 'Permissions' (which is highlighted in blue), and 'Submissions'. A large blue arrow points upwards from the 'Create Own Submissions' row towards the 'Permissions' tab. Another blue arrow points to the 'Anonymous' button in the 'Create Own Submissions' row, with the text 'Click on "Permissions" and select "Anonymous" for "Create Own Submissions"' overlaid.

Permission	Roles
Read All Submissions	Select roles...
Update All Submissions	Select roles...
Delete All Submissions	Select roles...
Create Own Submissions	Anonymous
Read Own Submissions	Select roles...
Update Own Submissions	Select roles...
Delete Own Submissions	Select roles...

User Permissions allow you to control who can create, view, and modify form submissions. Here you may assign Project Roles to permissions. [Roles](#) can be created and edited in [Project Settings](#).

Cancel Save Changes

To handle Authentication, click on the **Actions** tab  
Select **Authentication** action and then click **Add Action**



# Select New Resource, Authenticated, Email, and Password

**<form.io>** MovieApp > User Registration Form travis Help

User Registration Form FORM View Edit Actions Permissions Submissions

**Settings**

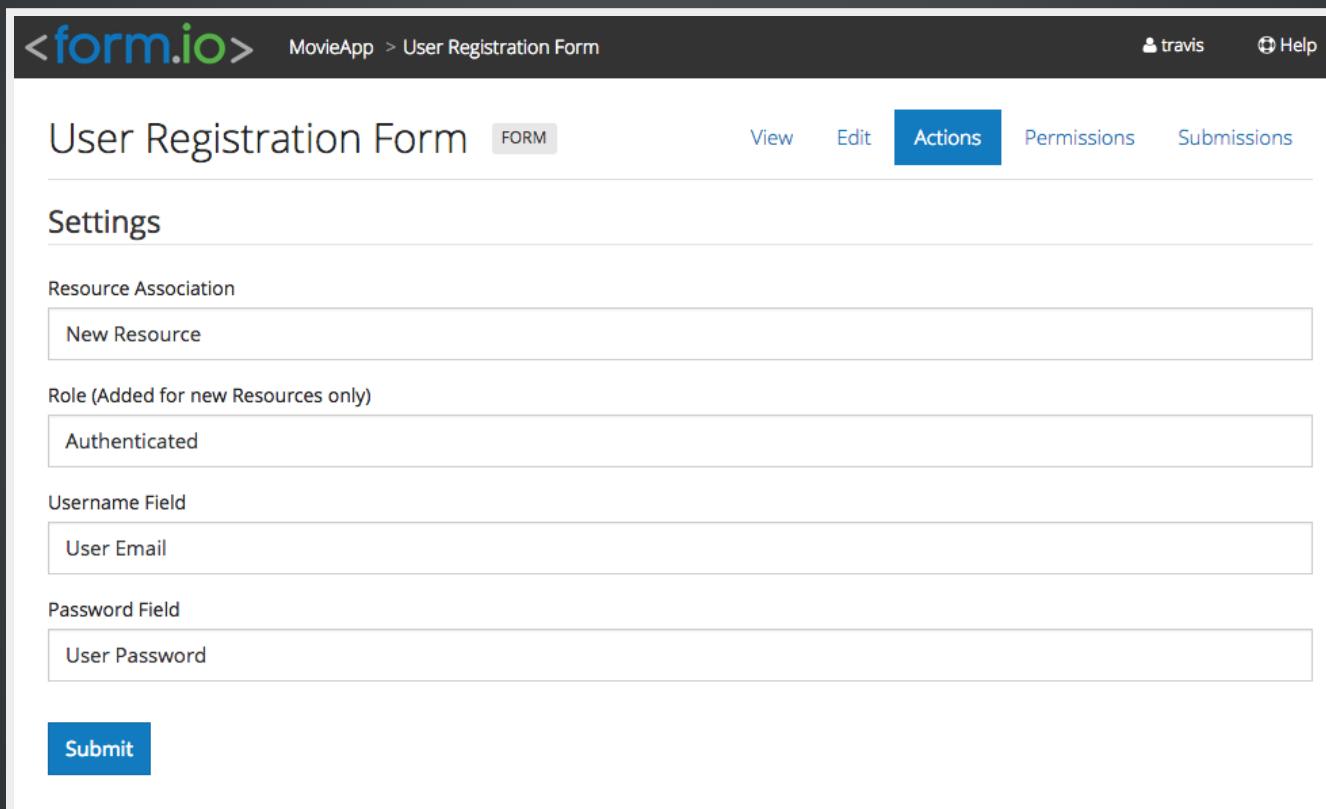
Resource Association

Role (Added for new Resources only)

Username Field

Password Field

Submit



# Clicking Submit will now create the Registration auth action!

The screenshot shows the form.io interface for a "User Registration Form". The top navigation bar includes the form.io logo, the project name "MovieApp > User Registration Form", user "travis", and help links. Below the header, the form title "User Registration Form" is displayed, along with tabs for "FORM", "View", "Edit", "Actions" (which is highlighted in blue), "Permissions", and "Submissions".

The main content area is titled "Form Actions". It lists an action named "Authentication" under the "Name" column. In the "Operations" column, there are two icons: a grey square with a white checkmark and a red square with a white circle. At the bottom left, a dropdown menu says "Select an Action" with a downward arrow. At the bottom right, a blue button says "+ Add Action".

# Create a Director resource.

- Required **Text Field** called **Name**
- **Phone Number** field called **Phone Number**
- **Address** field called **Address**

<form.io> MovieApp > Director

travis Help

New Resource

Title  
Director

Name  
director

API Path  
`https://movieapp.form.io/ director`

The REST API path to this form.

Advanced Settings

Form Components

Text Field	Address
Button	Check Box
Content	Date / Time
Email	Field Set
Hidden	Number
Password	Phone Number
Radio	Resource
Select	Signature
Text Area	

Page 1 Add Page

Name \*  
Enter the directors name

Phone Number  
Enter the directors phone number

Address  
Enter the address

Submit

Click **Save** and You now have a **Director** form w/ corresponding API

The screenshot shows the form.io interface for a "Director" form within a "MovieApp" project. The top navigation bar includes the form.io logo, the project name "MovieApp > Director", user "travis", and help documentation.

The main interface has two main sections:

- Director** (RESOURCE): This section contains three input fields: "Name \*", "Phone Number", and "Address". Each field has a placeholder text: "Enter the directors name", "Enter the directors phone number", and "Enter the address". A "Submit" button is located at the bottom of this section.
- Form API**: This section informs the user that the form API is ready and provides the URL: <https://movieapp.form.io/director>. It also links to "Project Embedding" and "API Specification".

Enable Authenticated use by clicking on **Permissions** and add **Authenticated** to all "own" permissions.

**User Permissions**

User Permissions allow you to control who can create, view, and modify form submissions. Here you may assign Project Roles to permissions. **Roles** can be created and edited in [Project Settings](#).

Permission	Roles
Read All Submissions	Select roles...
Update All Submissions	Select roles...
Delete All Submissions	Select roles...
Create Own Submissions	Authenticated <input type="button" value="X"/>
Read Own Submissions	Authenticated <input type="button" value="X"/>
Update Own Submissions	Authenticated <input type="button" value="X"/>
Delete Own Submissions	Authenticated <input type="button" value="X"/>

**Click on "Permissions" and then add the "Authenticated" role to all the "own" permissions.**

**Cancel** **Save Changes**

# Create a Movie resource.

- Required Text Field called **Title**
- Required Text Field called **Video**

**<form.io>** MovieApp > Movie travis Help

### New Resource

Title

Name

API Path

The REST API path to this form.

Advanced Settings

Form Components

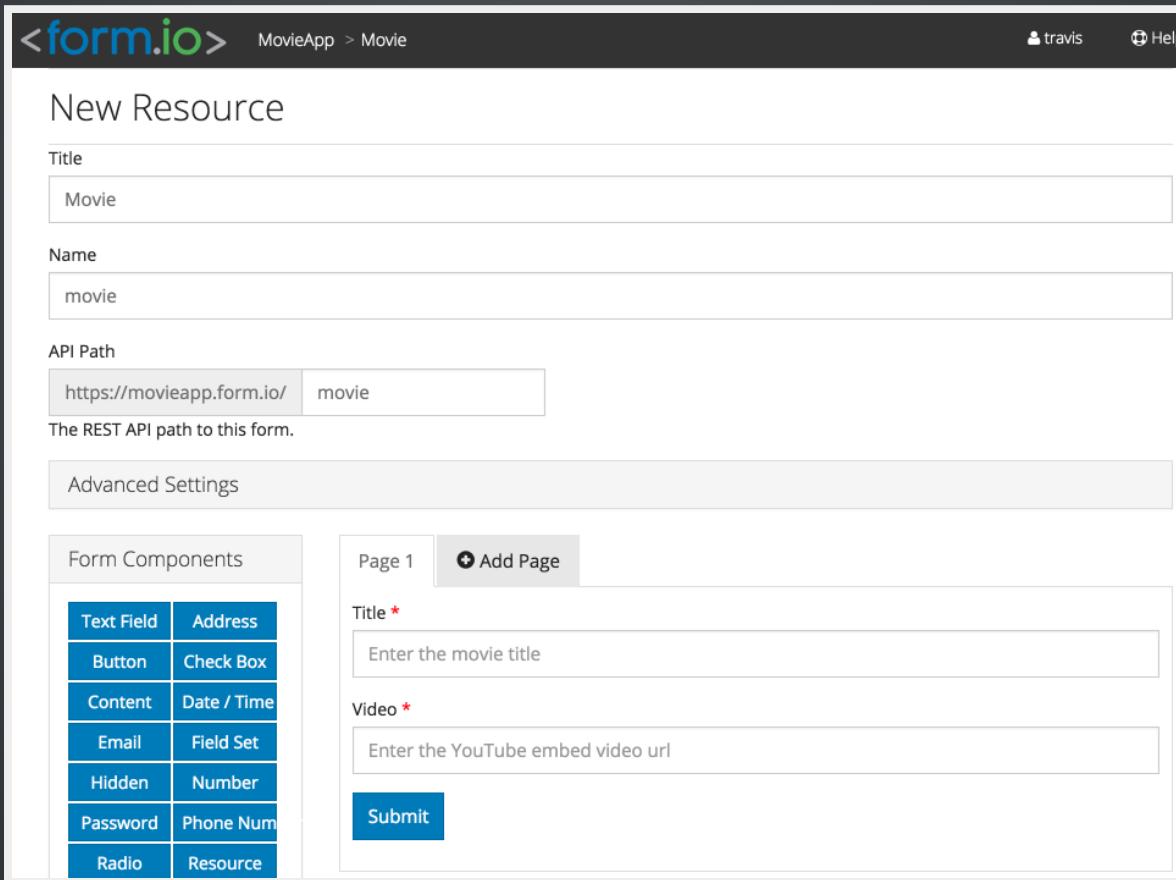
Text Field	Address
Button	Check Box
Content	Date / Time
Email	Field Set
Hidden	Number
Password	Phone Num
Radio	Resource

Page 1 + Add Page

Title \*

Video \*

**Submit**



# Drag & Drop Resource field.

Select Director, change Item Template, and allow Multiple

The screenshot shows a configuration interface for a resource field. On the left, under the 'Display' tab, there are several settings:

- Label:** Director
- Placeholder:** Select the Director(s)
- Resource:** Director
- Search Expression:** The search string regular expression
- Select Fields:** Comma separated list of fields to select.
- Search Fields:** The search field pairings
- Item Template:** A code editor containing the template `<span>{{ item.data.name }}</span>`. A blue arrow points from this section to a callout box.
- Allow Multiple Resources:** A checked checkbox.

On the right, a preview window shows a dropdown menu labeled 'Director' with a single option 'Director'. Below the dropdown are three buttons: Save (green), Cancel (gray), and Remove (red).

**Make sure you change the Item Template to say {{ item.data.name }}, which will show the Director resource "Name" field in the select dropdown.**

Click **Save** and You now have a **Movie** form w/ corresponding API

The screenshot shows the form.io interface for a "Movie" form. At the top, there's a navigation bar with the form.io logo, the project name "MovieApp > Movie", and user account information ("travis"). Below the navigation, the form title "Movie" is displayed next to a "RESOURCE" button. A horizontal menu bar includes "View" (which is highlighted in blue), "Edit", "Actions", "Permissions", and "Submissions".

The main content area contains three input fields:

- Title \***: A text input field with placeholder text "Enter the movie title".
- Video \***: A text input field with placeholder text "Enter the YouTube embed video url".
- Director**: A text input field with placeholder text "Select the director(s)".

At the bottom left is a blue "Submit" button. To the right of the form, there's a sidebar with a "Form API" section. This section contains the following text:

**Your form API is now ready to use!** The following API was created that maps directly to this form interface.

This API can be found by going to the following URL.

`https://movieapp.form.io/movie`

Below this are links for "Project Embedding" and "API Specification".

Enable Authenticated use by clicking on **Permissions** and add **Authenticated** to all "own" permissions.

**User Permissions**

User Permissions allow you to control who can create, view, and modify form submissions. Here you may assign Project Roles to permissions. **Roles** can be created and edited in [Project Settings](#).

Permission	Roles
Read All Submissions ⓘ	Select roles...
Update All Submissions ⓘ	Select roles...
Delete All Submissions ⓘ	Select roles...
Create Own Submissions ⓘ	Authenticated ×
Read Own Submissions ⓘ	Authenticated ×
Update Own Submissions ⓘ	Authenticated ×
Delete Own Submissions ⓘ	Authenticated ×

Click on "Permissions" and then add "Authenticated" role to all "own" permissions

Cancel Save Changes

# Download the Example Movie Application

<https://github.com/formio/movieapp>

The screenshot shows the GitHub repository page for 'formio / movieapp'. The repository has 2 commits, 1 branch (master), 0 releases, and 1 contributor (travist). The commit history lists several files being added, including .idea, app, test, .bowerrc, .editorconfig, .gitattributes, .gitignore, .jshintrc, .travis.yml, .yo-rc.json, Gruntfile.js, and README.md. A callout box with a blue arrow points to the 'Clone in Desktop' button, which is highlighted with a red border. Another callout box with a blue arrow points to the 'Click here to download' link, also highlighted with a red border.

An example application using the Form.IO platform. — Edit

2 commits 1 branch 0 releases 1 contributor

branch: master movieapp / +

Adding directors and removing my .idea folder.

travist authored 2 days ago latest commit 9ea8fec523

File	Description	Time Ago
app	Adding directors and removing my .idea folder.	2 days ago
test	Adding directors and removing my .idea folder.	2 days ago
.bowerrc	Adding the example movie application.	3 days ago
.editorconfig	Adding the example movie application.	3 days ago
.gitattributes	Adding the example movie application.	3 days ago
.gitignore	Adding directors and removing my .idea folder.	2 days ago
.jshintrc	Adding the example movie application.	3 days ago
.travis.yml	Adding the example movie application.	3 days ago
.yo-rc.json	Adding the example movie application.	3 days ago
Gruntfile.js	Adding the example movie application.	3 days ago
README.md	Adding the example movie application.	3 days ago

Click here to download

SSH clone URL  
git@github.com:formio/moveapp

You can clone with HTTPS, SSH, or Subversion.

Clone in Desktop Download ZIP

Make sure you have the following installed

- Node.js - <https://nodejs.org/download>
- npm install -g grunt-cli bower
- gem update --system
- gem install compass

Change the Form.IO URL's to point to your application.

NOTE: Your application will have a different sub-domain.

---

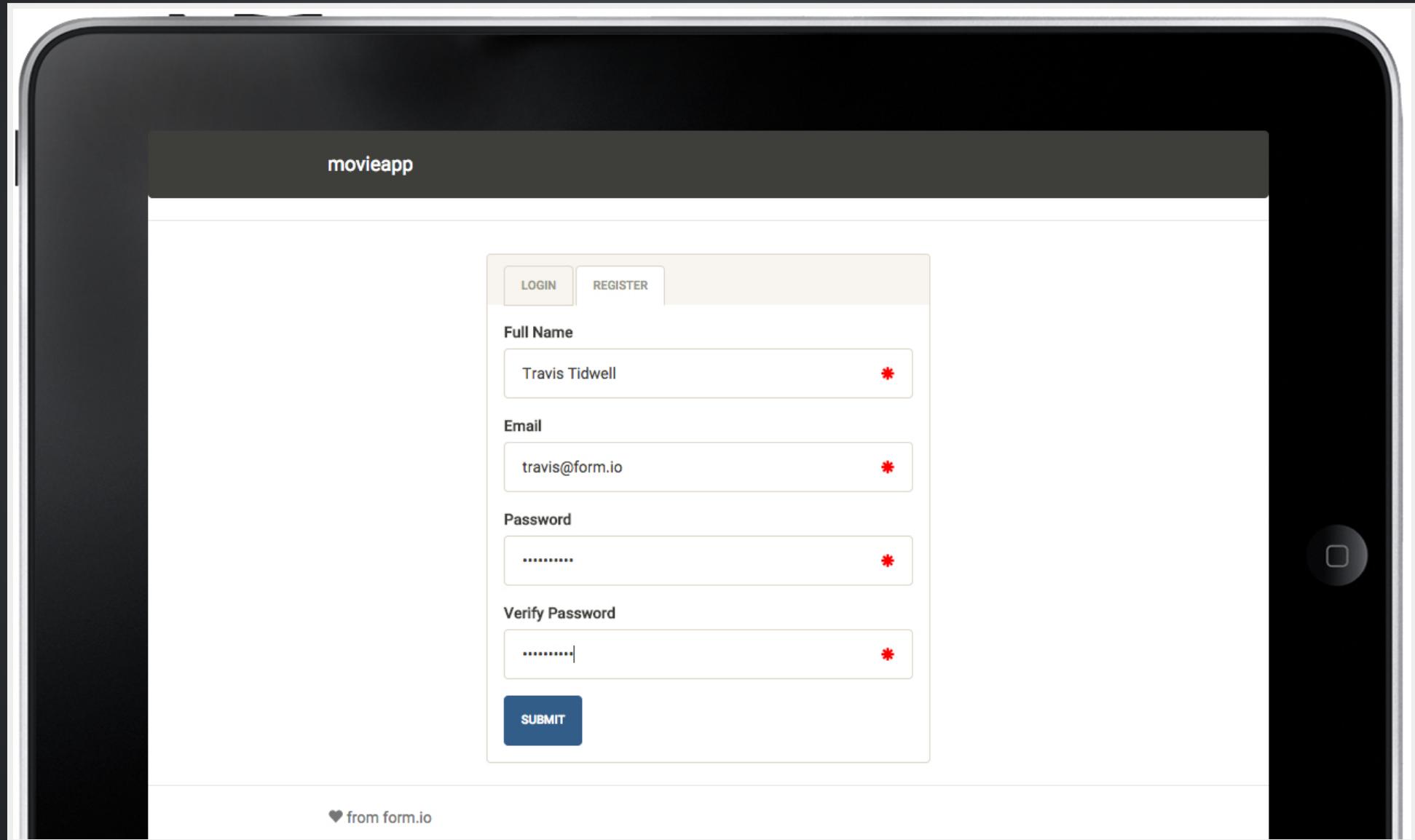
**/app/scripts/app.js**

```
$rootScope.userLoginForm = 'https://movieapp.form.io/user/login';
$rootScope.userRegisterForm = 'https://movieapp.form.io/user/register';
$rootScope.movieForm = 'https://movieapp.form.io/movie';
$rootScope.directorForm = 'https://movieapp.form.io/director';
```

## Build the project

- cd movieapp-master
- npm install
- bower install
- grunt serve

# Your application is now running!



# Using the application

# Create a new Director

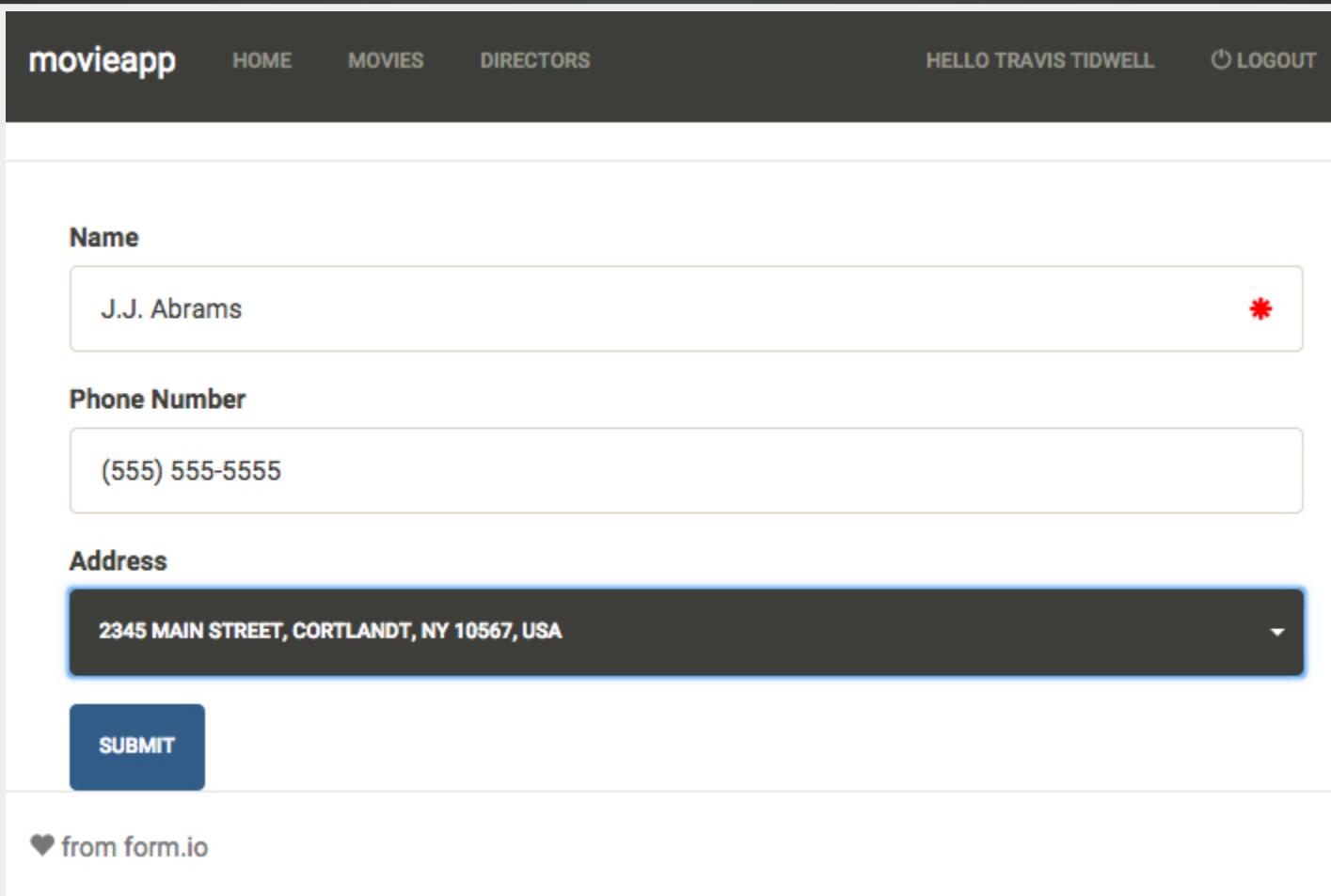
The screenshot shows a web application interface for managing directors. At the top, there is a dark header bar with the logo "movieapp" on the left, followed by navigation links: "HOME", "MOVIES", "DIRECTORS", "HELLO TRAVIS TIDWELL", and "LOGOUT". Below the header is a table with columns: "Address", "Name", "Phone Number", "Submitted", "Updated", and "Operations". A blue button labeled "+ CREATE DIRECTOR" is positioned above the table. Two blue arrows point upwards from the text instructions below to the "DIRECTORS" link in the header and the "+ CREATE DIRECTOR" button respectively.

Address Name Phone Number Submitted Updated Operations

♥ from form.io

Click on Director, and then press the "+ Create Director" button

# Fill out some information about the Director



The image shows a screenshot of a web application named "movieapp". The header bar is dark grey with white text. It includes the "movieapp" logo, navigation links for "HOME", "MOVIES", and "DIRECTORS", and a user account section showing "HELLO TRAVIS TIDWELL" and a "LOGOUT" button. Below the header is a light gray form area. The first field is labeled "Name" and contains the value "J.J. Abrams" with a red asterisk (\*) indicating it is a required field. The second field is labeled "Phone Number" and contains the value "(555) 555-5555". The third field is labeled "Address" and contains the value "2345 MAIN STREET, CORTLANDT, NY 10567, USA", which is highlighted with a blue border. At the bottom of the form is a blue "SUBMIT" button. In the bottom left corner of the form area, there is a small footer note: "♥ from form.io".

movieapp

HOME MOVIES DIRECTORS

HELLO TRAVIS TIDWELL LOGOUT

Name

J.J. Abrams \*

Phone Number

(555) 555-5555

Address

2345 MAIN STREET, CORTLANDT, NY 10567, USA

SUBMIT

♥ from form.io

# We now have a view of the Director

movieapp

HOME MOVIES DIRECTORS

HELLO TRAVIS TIDWELL LOGOUT

VIEW EDIT DELETE

J.J. Abrams

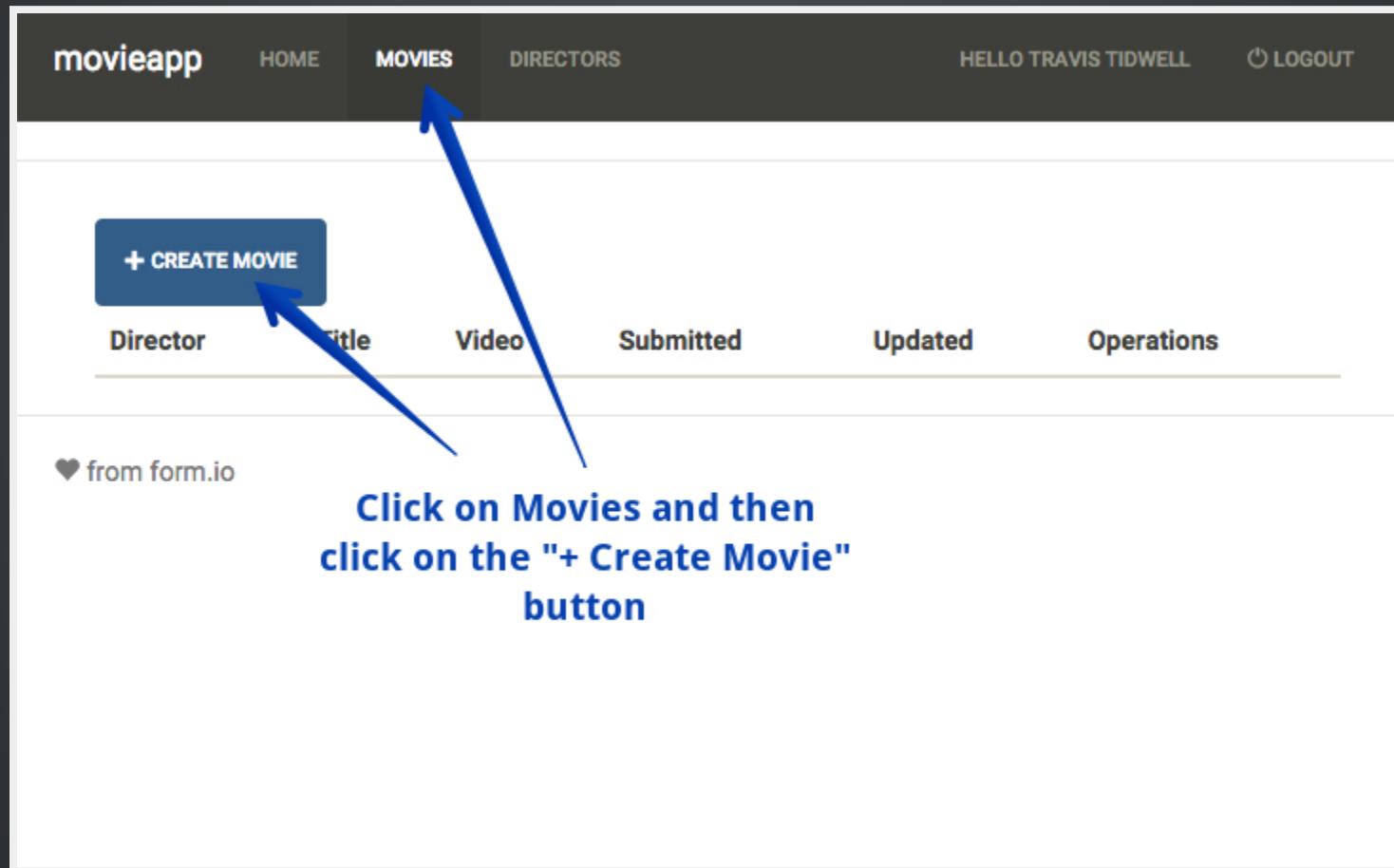
Address: 2345 Main Street, Cortlandt, NY 10567, USA



(555) 555-5555

from form.io

Now go to the Movies section and create a new movie



# Provide information about the movie.

The image shows a screenshot of a web application interface for "movieapp". The header includes navigation links for "HOME", "MOVIES", and "DIRECTORS", along with a user session indicator "HELLO TRAVIS TIDWELL" and a "LOGOUT" button. The main content area contains three input fields: "Title" (Star Wars: The Force Awakens), "Video" (a YouTube URL), and "Director" (J.J. ABRAMS). Each field has a red asterisk indicating it is required. Below the director field is a blue "SUBMIT" button. At the bottom of the form, there is a footer note "♥ from form.io".

movieapp

HOME MOVIES DIRECTORS

HELLO TRAVIS TIDWELL LOGOUT

Title

Star Wars: The Force Awakens \*

Video

<https://www.youtube.com/embed/ngElkyQ6Rhs> \*

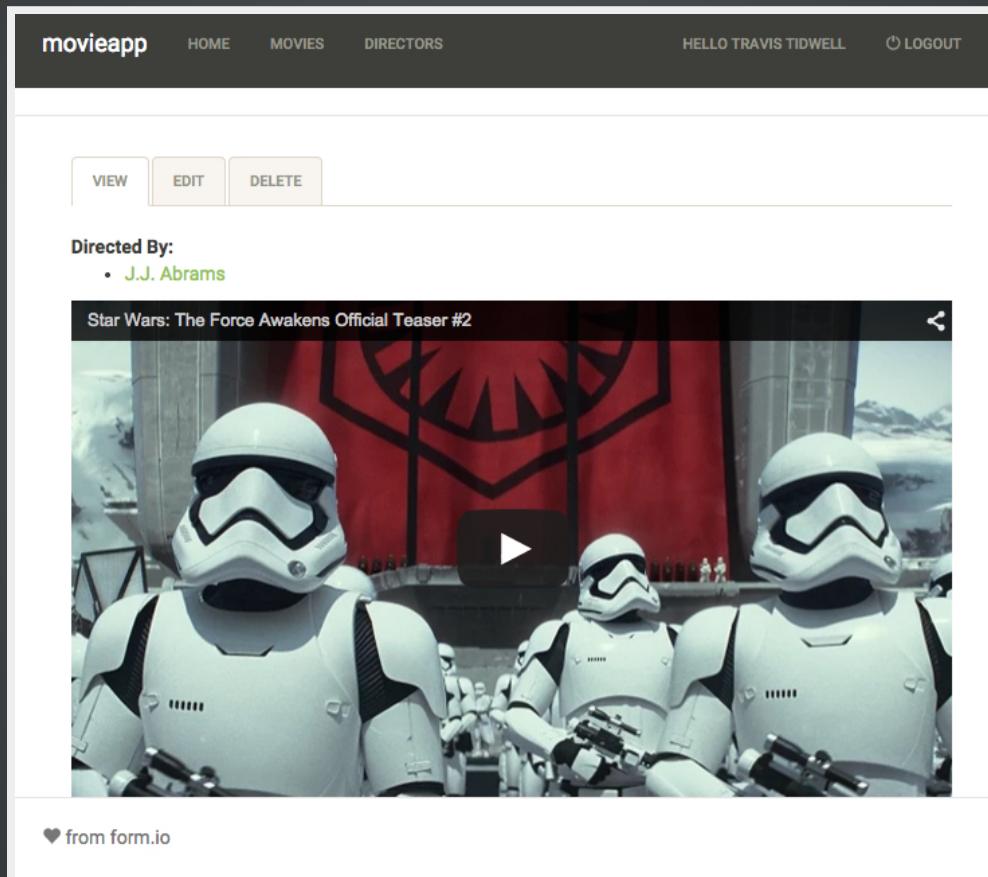
Director

J.J. ABRAMS X

SUBMIT

♥ from form.io

# You now have a working movie application!



How **form**.io works.

# This application has the following pages

```
- app
  - views
    - main.html
    - youtube.html
    - user
      - auth.html
      - login.html
      - register.html
    - movie
      - index.html
      - create.html
      - edit.html
      - delete.html
      - view.html
      - movie.html
    - director
      - index.html
      - create.html
      - edit.html
      - delete.html
      - view.html
      - director.html
```

# Hook up **form.io**

---

```
app/views/user/login.html
```

```
<formio src="userLoginForm"></formio>
```

---

```
app/views/user/register.html
```

```
<formio src="userRegisterForm"></formio>
```

---

```
app/views/movie/create.html
```

```
<formio src="movieForm"></formio>
```

---

```
app/views/director/create.html
```

```
<formio src="directorForm"></formio>
```

## Hook up **form.io** delete pages

```
app/views/movie/delete.html
```

```
<formio-delete src="movieForm"></formio-delete>
```

```
app/views/director/delete.html
```

```
<formio-delete src="directorForm"></formio-delete>
```

## Hook up **form.io** index pages

---

```
app/views/movie/index.html
```

```
<formio-submissions src="movieForm"></formio-submissions>
```

---

```
app/views/director/index.html
```

```
<formio-submissions src="directorForm"></formio-submissions>
```

# Handling I/O with the Formio factory

```
app/scripts/app.js
.state('createMovie', {
  url: '/create/movie',
  templateUrl: 'views/movie/create.html',
  controller: ['$scope', '$state', function ($scope, $state) {
    $scope.$on('formSubmission', function(event, submission) {
      $state.go('movie.view', {movieId: submission._id});
    });
  }]
})
.state('movie.view', {
  url: '/',
  parent: 'movie',
  templateUrl: 'views/movie/view.html',
  controller: ['$scope', '$stateParams', 'Formio', function ($scope, $stateParams, Formio) {
    var loader = new Formio($scope.movieForm + '/submission/' + $stateParams.movieId);
    $scope.movie = {};
    loader.loadSubmission().then(function(submission) {
      $scope.movie = submission;
    });
  }]
})
```

Have fun with **form.io**!