

form.io

BUILDING A NEXT-GEN WEB APPLICATION



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

View Edit Actions Submissions

Title
User Registration Form

Name
userRegistrationForm

API Path
<https://movieapp.form.io/app/api/> / user/register

The REST API path to this form.

Form Components

- Text Field
- Address
- Date / Time
- Email
- Field Set
- Hidden
- Number
- Password
- Phone Number
- Resource

Page 1 + Add Page

Private Beta Code
Enter your beta code *

Full Name
Enter your full name

Email
Enter your email address *

Password
Enter your password. *

Verify Password
Verify your password. *

A screenshot of the form.io builder interface. At the top, there are tabs for View, Edit (which is selected), Actions, and Submissions. Below that, there are sections for Title (User Registration Form), Name (userRegistrationForm), and API Path (https://movieapp.form.io/app/api/ / user/register). A note says 'The REST API path to this form.' On the left, there's a sidebar titled 'Form Components' with buttons for Text Field, Address, Date / Time, Email, Field Set, Hidden, Number, Password, Phone Number, and Resource. On the right, there's a 'Page 1' section with a '+ Add Page' button. It contains fields for Private Beta Code, Full Name, Email, Password, and Verify Password, each with a red asterisk indicating it's required. The interface has a clean, modern design with a light gray background and blue highlights for selected items.

form.io

creates both the interface and the API in one step

The screenshot shows the form.io application interface. At the top, there's a navigation bar with the form.io logo, a home icon, and a user profile for "travis". Below the navigation, a breadcrumb trail shows the path: "/ movieapp". The main content area has tabs for "View", "Edit", "Actions", and "Submissions", with "Edit" currently selected.

User Registration Form

This section contains six input fields:

- Private Beta Code**: A text input field with placeholder "Enter your beta code" and a red asterisk indicating it's required.
- Full Name**: A text input field with placeholder "Enter your full name".
- Email**: A text input field with placeholder "Enter your email address" and a red asterisk indicating it's required.
- Password**: A text input field with placeholder "Enter your password." and a red asterisk indicating it's required.
- Verify Password**: A text input field with placeholder "Verify your password" and a red asterisk indicating it's required.
- Submit**: A blue button at the bottom of the form.

Form API

This section displays the generated API 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.

`http://movieapp.form.io/app/api/user/register`

API Specification

Application Embedding



automatically creates the API documentation

The screenshot shows the form.io interface. On the left, there is a 'User Registration Form' with fields for 'Private Beta Code', 'Full Name', 'Email', 'Password', and 'Verify Password'. A 'Submit' button is at the bottom. On the right, the generated API documentation is displayed under 'Form API' and 'API Specification'. The 'Form API' section includes an 'API Specification' tab and a link to <https://api.form.io>. Below it, it says 'Created by support@form.io' and 'MIT'. The 'form' resource is listed with methods: 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), and DELETE /app/:appId/form/{formId} (Delete a specific form). The 'submission' resource is also listed with similar methods. At the bottom, it shows '[BASE URL: /form/undefined , API VERSION: 0.0.1]' and an 'INVALID' status indicator.

form.io

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

The screenshot shows the form.io platform. On the left, there's a sidebar with 'View' selected, followed by 'Edit', 'Actions', and 'Submissions'. Below this is a 'User Registration Form' containing fields for 'Private Beta Code', 'Full Name', 'Email', 'Password', and 'Verify Password', each with a red asterisk indicating it's required. At the bottom is a 'Submit' button. To the right, under 'Form API', is a section titled 'Application Embedding' with the following text: 'To add this form to your application, start by including the following script in your application.' Below it is a code block with the script tag:

```
<script src="http://form.io/bower_components/formio/dist/formio.min.js" type="text/javascript"></script>
```

. A blue arrow points from the text 'It's as easy as an tag!' down to the '<formio>' tag in the code block. The top right corner of the interface shows a user profile icon and the name 'travis'.

form.io

/ movieapp

View Edit Actions Submissions

User Registration Form

Private Beta Code

Enter your beta code *

Full Name

Enter your full name

Email

Enter your email address *

Password

Enter your password. *

Verify Password

Verify your password *

Submit

Form API

API Specification

Application Embedding

To add this form to your application, start by including the following script in your application.

```
<script src="http://form.io/bower_components/formio/dist/formio.min.js" type="text/javascript"></script>
```

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

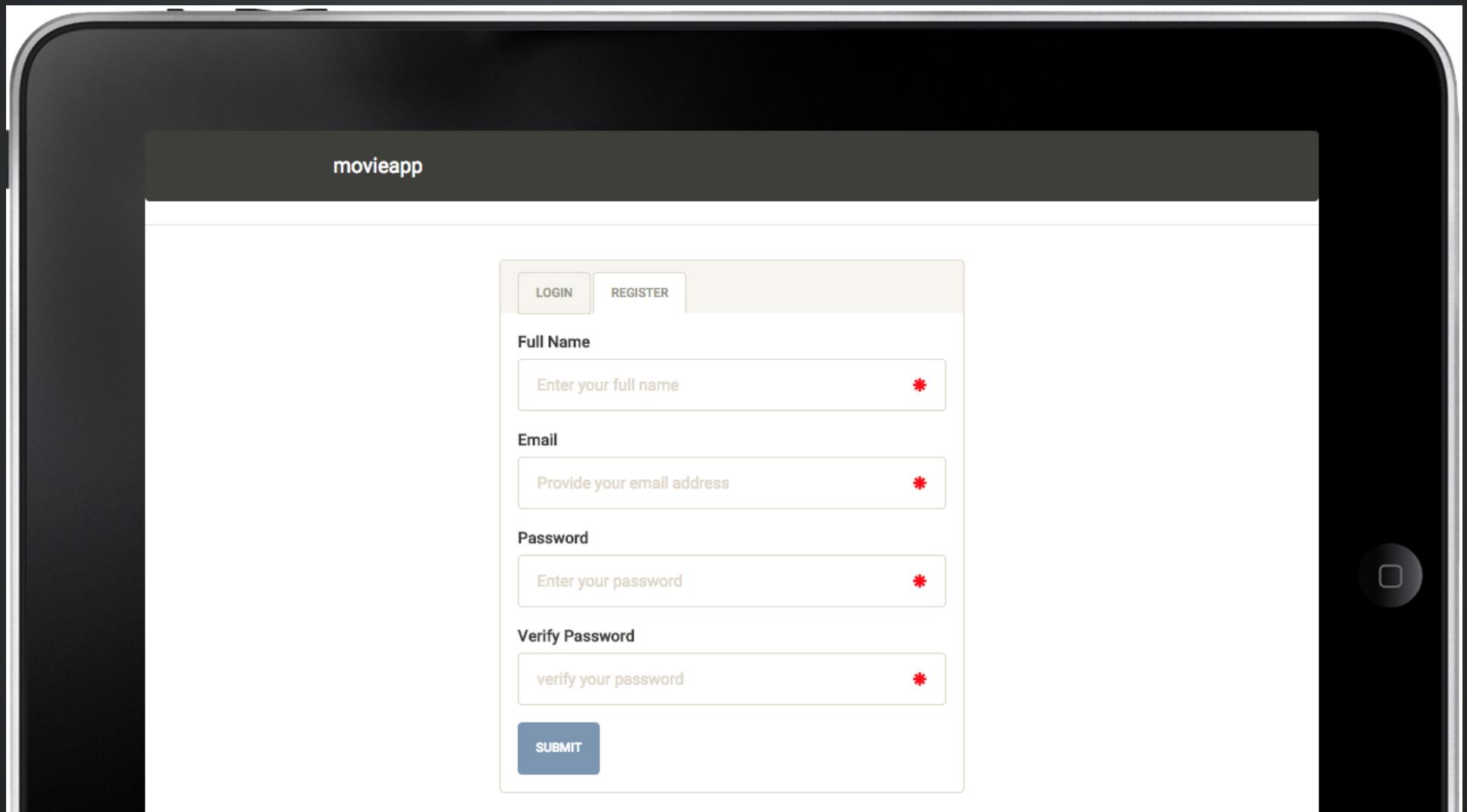
```
<formio src="http://movieapp.form.io/app/api/user/register"></formio>
```

It's as easy as an `` tag!

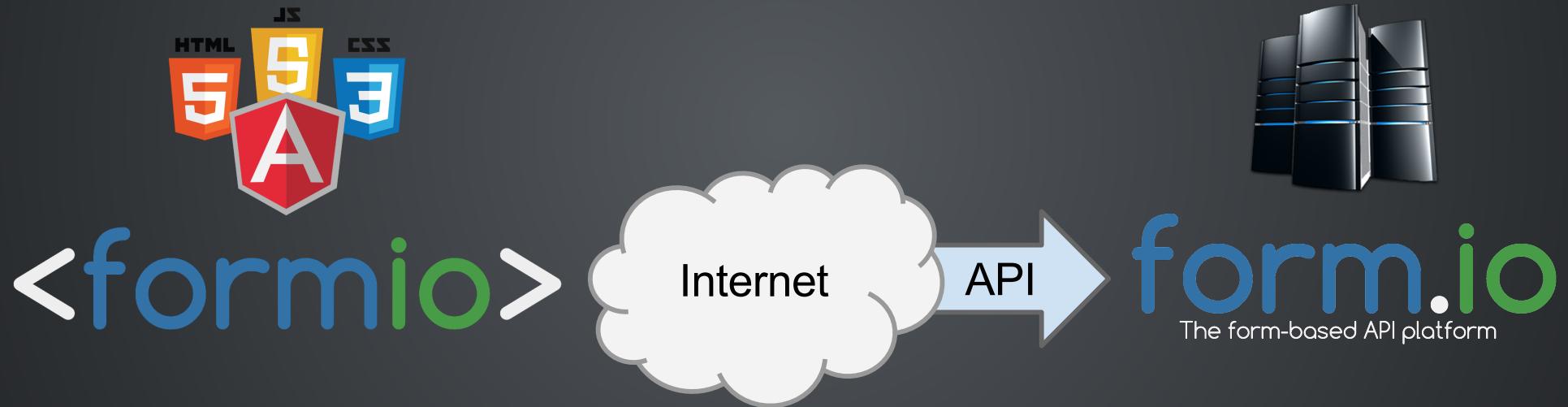
instead of <form>

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

Application integration with **form.io**



The **form.io** solution



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

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

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

The screenshot shows a registration form for Form.IO's private beta. 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 green 'Request Beta Access' button. The form fields are as follows:

Private Beta Code	123123132	*
Full name	Travis Tidwell	*
Username	travist	*
Email	travis@form.io	*
Password	*****	*
Verify Password	*****	

At the bottom is a blue 'Submit' button.

Create an application

NOTE: Your application will need a different domain name.

The screenshot shows the form.io application creation interface. At the top, the form.io logo is visible along with a user icon labeled "travis". The URL bar shows the path "/ movieapp". The main form fields are:

- Application Title:** MovieApp
- Domain Name:** https://movieapp.form.io
- Description:** This is a movie application.
- Application Access:** Select who should have access (button)

A prominent blue "Save" button is located at the bottom left of the form.

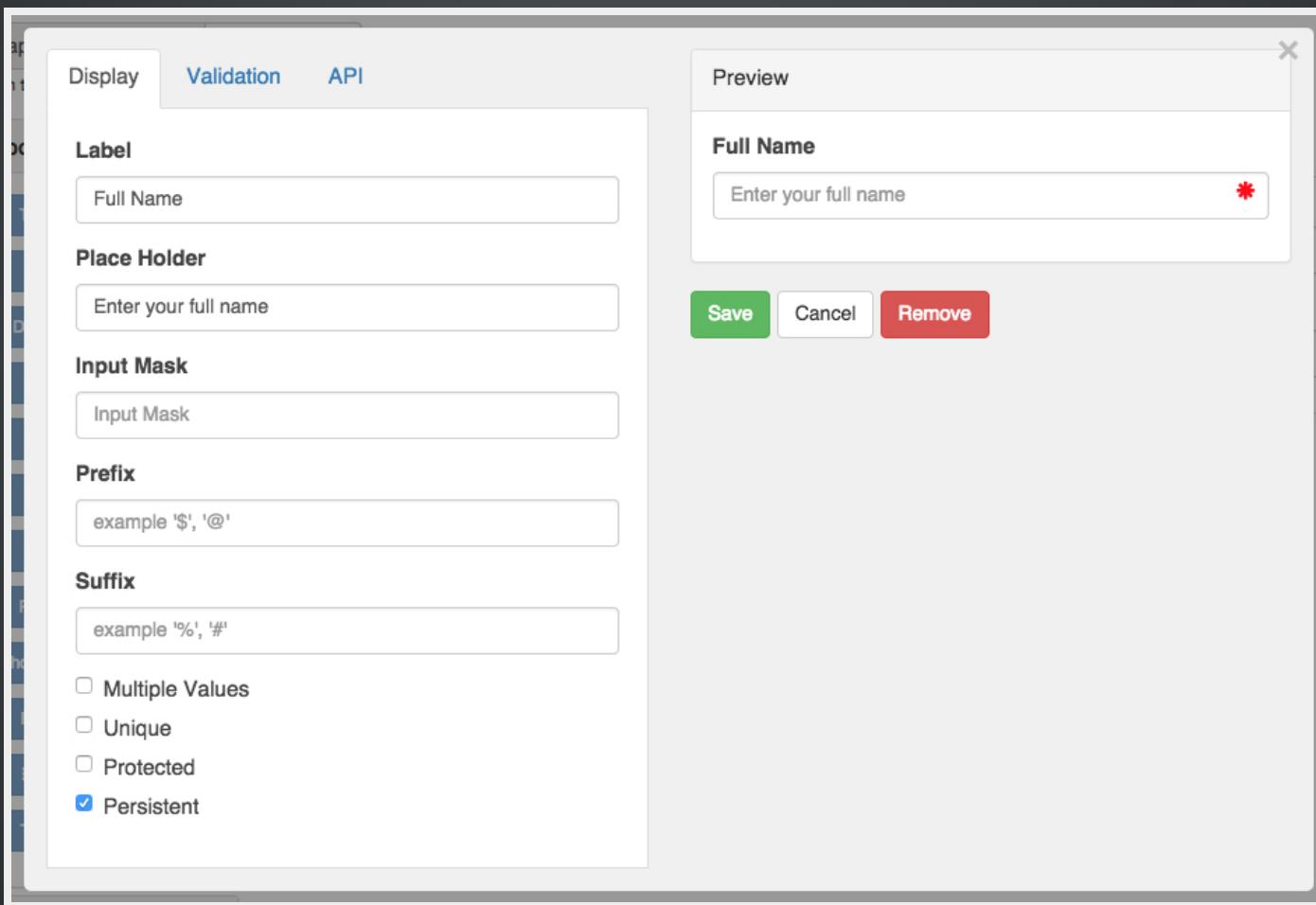
Create a new User resource.

The screenshot shows the form.io interface for creating a new User resource. The top navigation bar includes the form.io logo, a home icon, and a user profile for 'codertravist'. The breadcrumb navigation shows the path: Home / movieapp. The main form fields are:

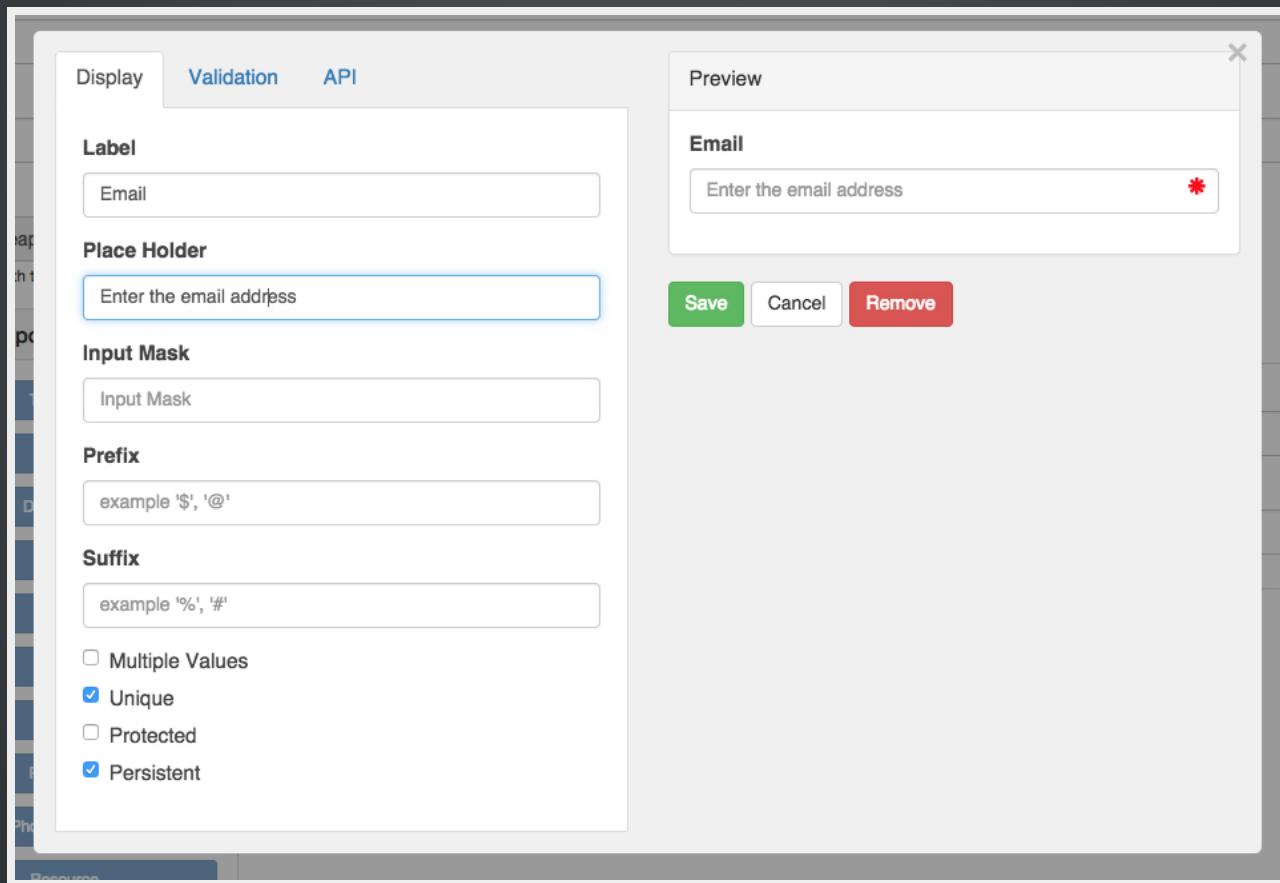
- Title**: User
- Name**: user
- API Path**: https://movieapp.form.io/app/api/ user

The API path field contains the URL and the resource name 'user' separated by a space. Below the API path, a note states: "The REST API path to this form." On the left, a sidebar titled "Form Components" lists "Text Field" and "Address". On the right, a "Page 1" section has an "Add Page" button and a large blue box with the text "Drag and Drop a form component".

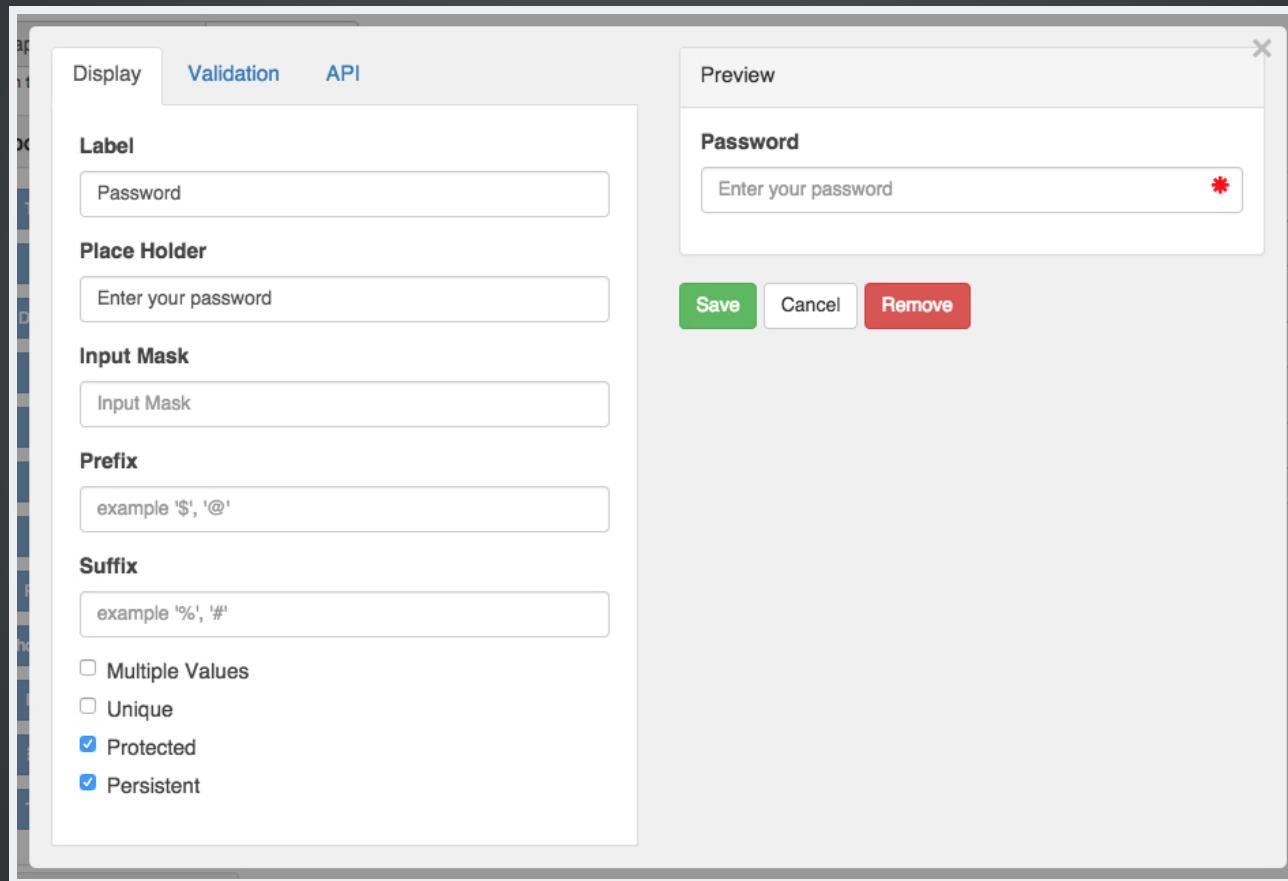
- Drag & Drop a **Text Field** field onto your form.
- Label it **Full Name**
- Check **Required** under Validation tab



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

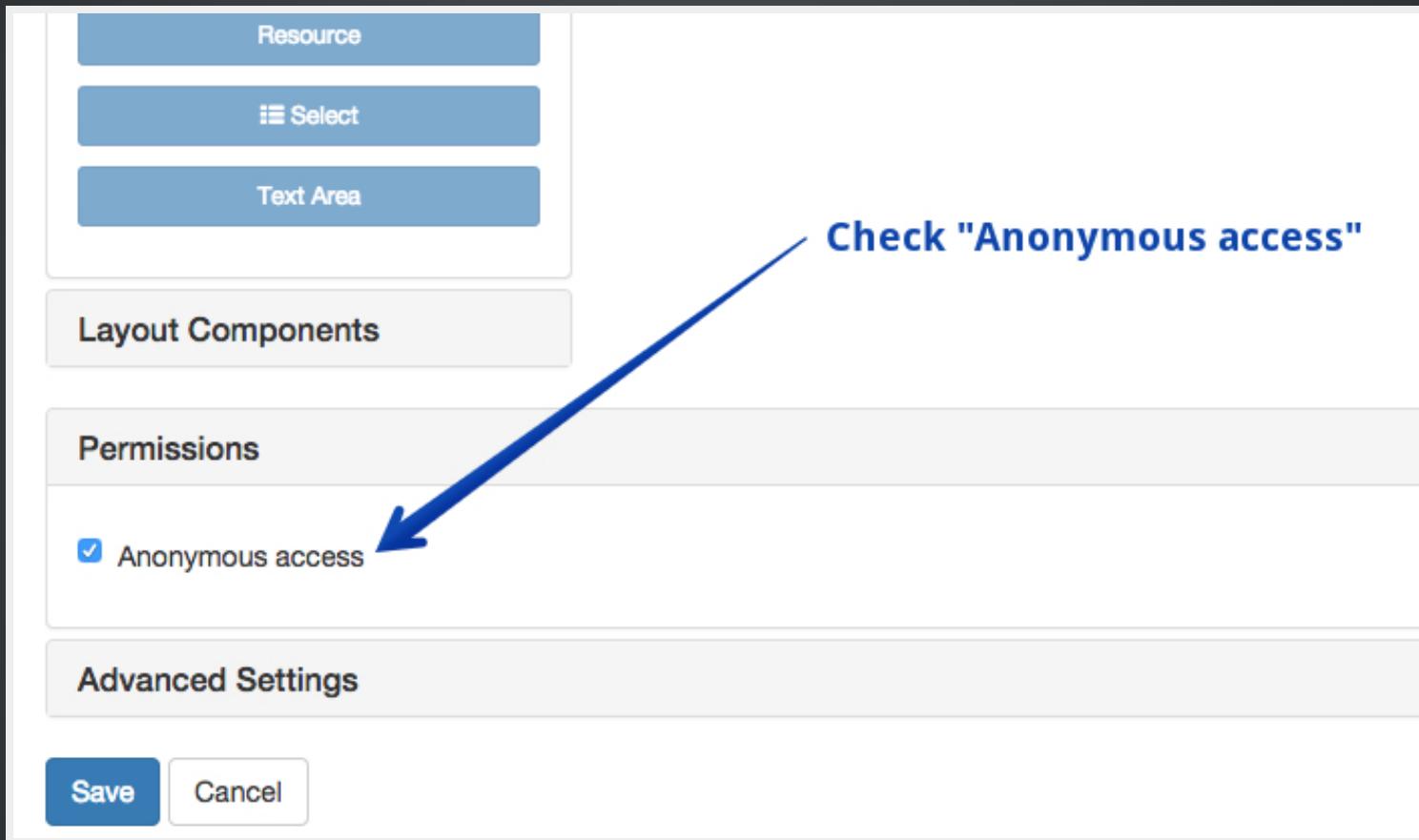


- Drag & Drop a Password field onto your form.
- Check Required



- Under **Permissions**, check **Anonymous access**.

This just ensures that anonymous users can create an account and gives them access to their own user object. This does not expose write abilities to the User resource form.



NOTE: This will be changing soon as we build out a more robust Roles & Permissions system.

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

The screenshot shows the form.io interface for a "User" form. The top navigation bar includes the form.io logo, a home icon, and a user account icon labeled "codertravist". Below the navigation, the URL "movieapp" is shown. The main interface has tabs for "View", "Edit" (which is selected), "Actions", and "Submissions".

The "User" form contains three fields:

- Full Name**: A text input field with the placeholder "Enter your full name" and a red asterisk indicating it is required.
- Email**: A text input field with the placeholder "Enter your email address" and a red asterisk indicating it is required.
- Password**: A text input field with the placeholder "Enter your password" and a red asterisk indicating it is required.

At the bottom left is a blue "Submit" button. On the right side, there is a "Form API" section with the following content:

Form API

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

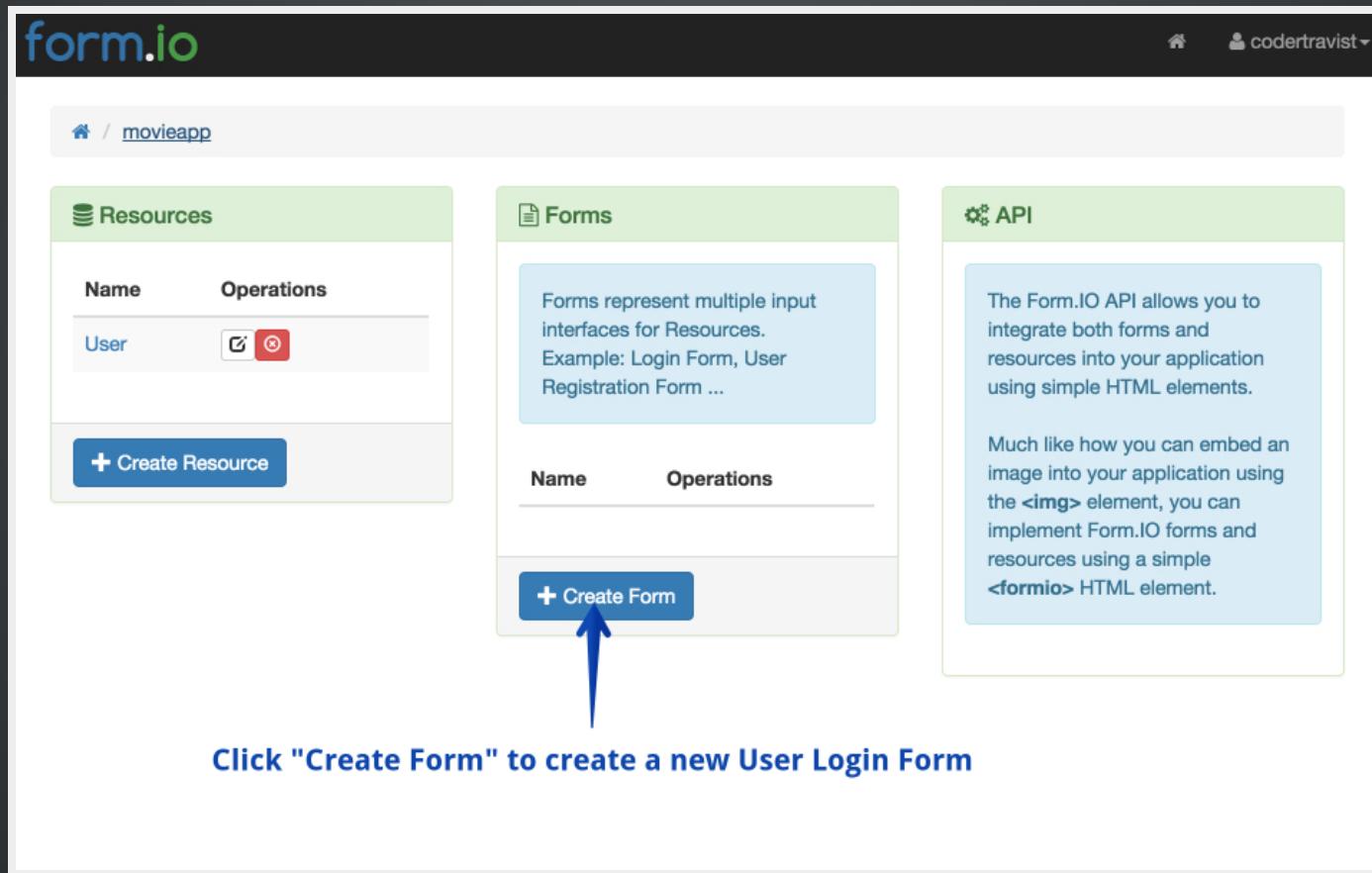
This API can be found by going to the following URL.

`http://movieapp.form.io/app/api/user`

Below the API section are two more buttons: "API Specification" and "Application Embedding".

Building the User Login Form

Click on **New Form** on your application home page.



Provide the Title and API path for this form.

The screenshot shows the form.io application interface. At the top, there is a navigation bar with the form.io logo, a home icon, and a user profile icon labeled "codertravist". Below the navigation bar, the URL "movieapp" is displayed. The main content area contains the following fields:

- Title**: User Login Form
- Name**: userLoginForm
- API Path**: `https://movieapp.form.io/app/api/` user/login|

A note below the API path says "The REST API path to this form." To the right, there is a sidebar titled "Form Components" containing "Text Field", "Address", "Date / Time", and "Email" options. Next to it is a "Page 1" section with an "+ Add Page" button and a large blue box with the placeholder text "Drag and Drop a form component".

Drag & Drop the User email and Password fields.

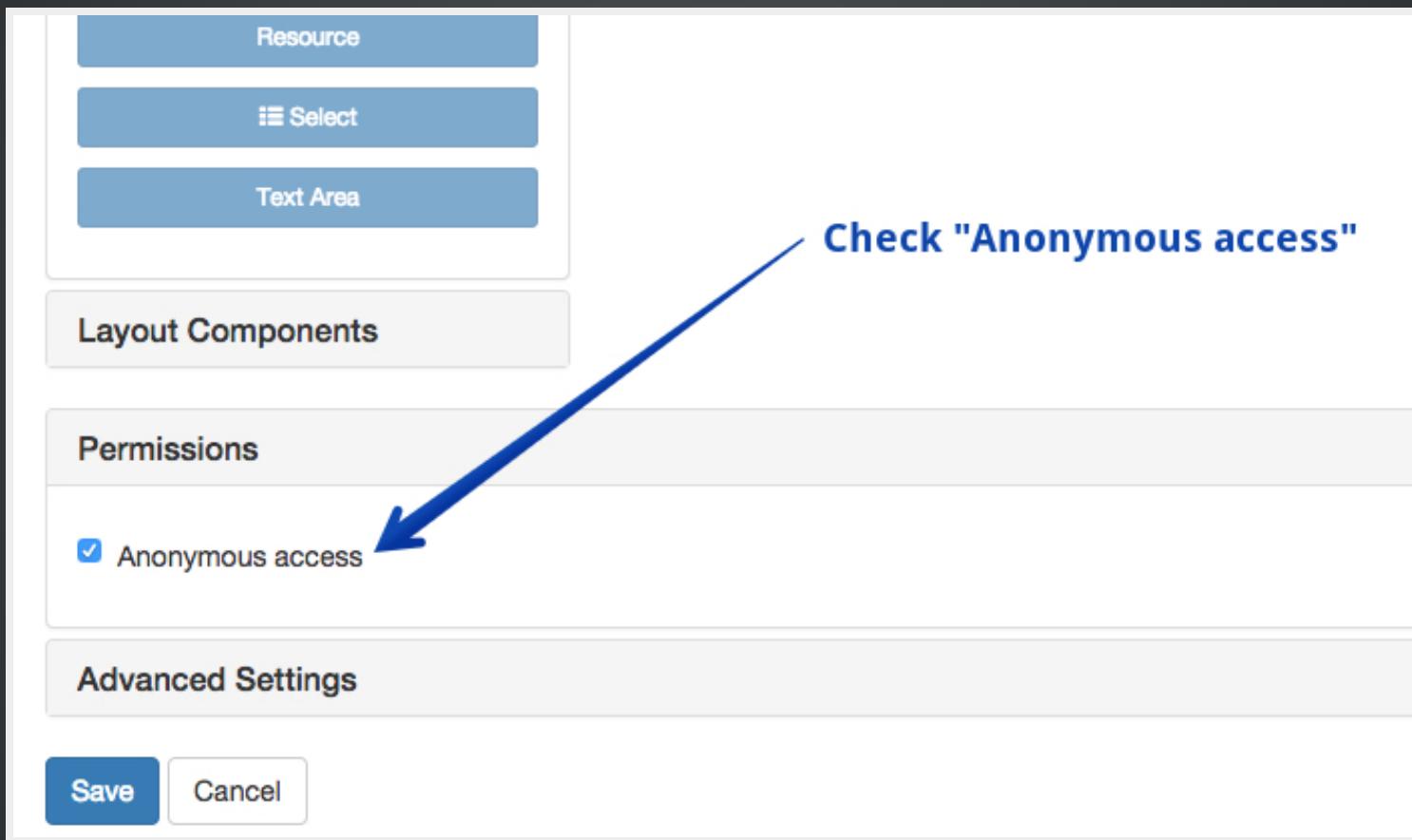
The screenshot shows a form builder interface with the following details:

- Title:** User Login Form
- Name:** userLoginForm
- API Path:** https://movieapp.form.io/app/api/ user/login
- Form Components:** Form Components, Layout Components
- User Fields:** User Email, User Full Name, User Password
- Page 1:** + Add Page
- Email:** Enter your email address *
- Password:** Enter your password *

A blue arrow points from the "User Email" button in the "User Fields" section to the "Email" input field on the "Page 1" component. Another blue arrow points from the "User Password" button in the "User Fields" section to the "Password" input field on the "Page 1" component.

**Drag & Drop User email to the form.
Rename to just "Email"**

Check "Anonymous access" within Permissions



NOTE: This will be changing soon as we build out a more robust Roles & Permissions system.

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

The screenshot shows the form.io platform interface. At the top, there's a green success message: "Successfully created form!". Below the header, the URL is shown as [movieapp](#). The navigation bar includes "View" (which is active), "Edit", "Actions", and "Submissions".

The main content area displays a "User Login Form" with two fields: "Email" and "Password", both marked with a red asterisk (*) indicating they are required. A "Submit" button is at the bottom of the form.

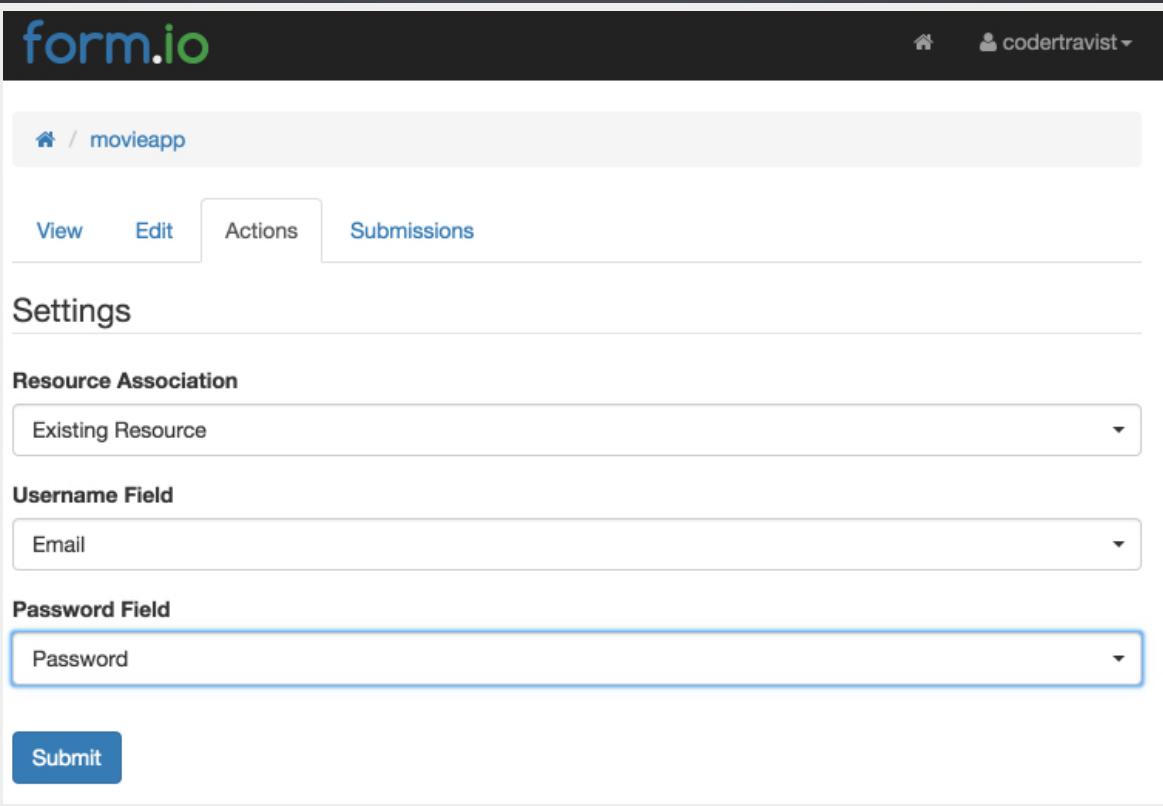
To the right of the form, under the heading "Form API", is a summary: "Your form API is now ready to use! The following API was created that maps directly to this form interface." It provides the API URL: <http://movieapp.form.io/app/api/user/login>. Below this, there are links for "API Specification" and "Application Embedding".

To handle Authentication, click on the **Actions** tab

Select **Authentication** action and then click **Add Action**

The screenshot shows the form.io application interface. At the top, there is a dark header with the 'form.io' logo on the left and a user profile icon on the right. Below the header, the URL 'movieapp' is displayed in a navigation bar. The main navigation menu consists of four tabs: 'View', 'Edit', 'Actions', and 'Submissions'. The 'Actions' tab is currently selected, indicated by a blue border around its button. Below the menu, there is a section titled 'Form Actions' with a light blue header. This section contains a table with two columns: 'Name' and 'Operations'. There is one row in the table, which is currently empty. At the bottom of this section, there is a dropdown menu labeled 'Authentication' and a blue button labeled '+ Add Action'.

Select Existing Resource, Email, and Password



The screenshot shows the form.io interface for a "movieapp". The top navigation bar includes the form.io logo, a home icon, and a user account labeled "codertravist". Below the header, the URL "movieapp" is displayed. A navigation menu at the top right contains "View", "Edit" (which is currently selected), "Actions", and "Submissions". The main content area is titled "Settings" and contains three configuration sections:

- Resource Association:** A dropdown menu set to "Existing Resource".
- Username Field:** A dropdown menu set to "Email".
- Password Field:** A dropdown menu set to "Password". This field is highlighted with a blue border.

At the bottom left is a blue "Submit" button.

Clicking Save will now create the Login Auth action!

The screenshot shows the form.io interface for a 'movieapp' form. At the top, there's a green success message box that says 'Action was created.' A navigation bar below it includes 'View', 'Edit', 'Actions' (which is currently selected), and 'Submissions'. The main area is titled 'Form Actions' and contains a table with two rows. The first row has a 'Name' column with 'Authentication' and an 'Operations' column with a blue checkmark icon and a red delete icon. At the bottom left is a dropdown menu 'Select an Action' and at the bottom right is a blue button '+ Add Action'.

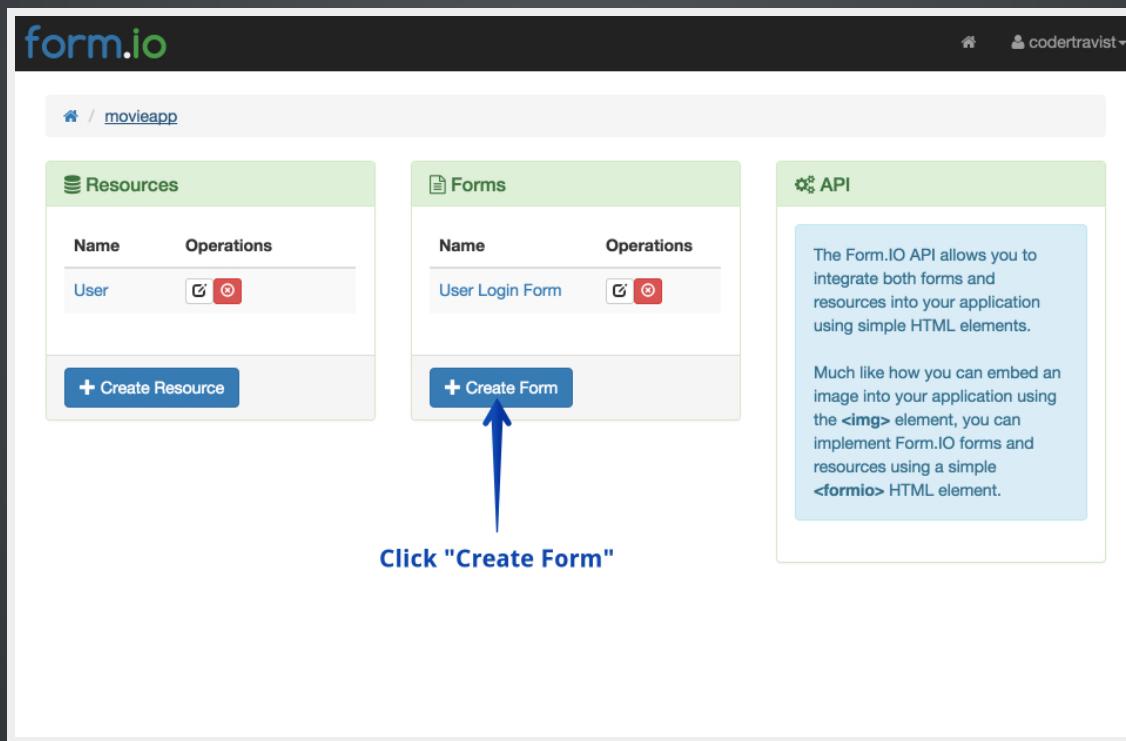
Name	Operations
Authentication	<input checked="" type="checkbox"/> <input type="button" value="X"/>

Select an Action

+ Add Action

Building the User Registration Form

Click on **New Form** on your application home page.



Provide a title and API path for the Register form.

The screenshot shows the form.io application interface. At the top, the logo 'form.io' is visible next to a user icon labeled 'codertravist'. Below the header, the breadcrumb navigation shows the path: Home / movieapp. The main configuration area contains the following fields:

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

Below the API Path, a note states: "The REST API path to this form." To the left, a sidebar titled "Form Components" lists "Text Field" and "Address". On the right, a section titled "Page 1" with a "Add Page" button contains a large blue button with the text "Drag and Drop a form component".

Drag, Drop and Rename the User resource fields.

Title
User Registration Form

Name
userRegistrationForm

API Path
<https://movieapp.form.io/app/api/> user/register
The REST API path to this form.

Form Components

Layout Components

User Fields

User Email

User Full Name

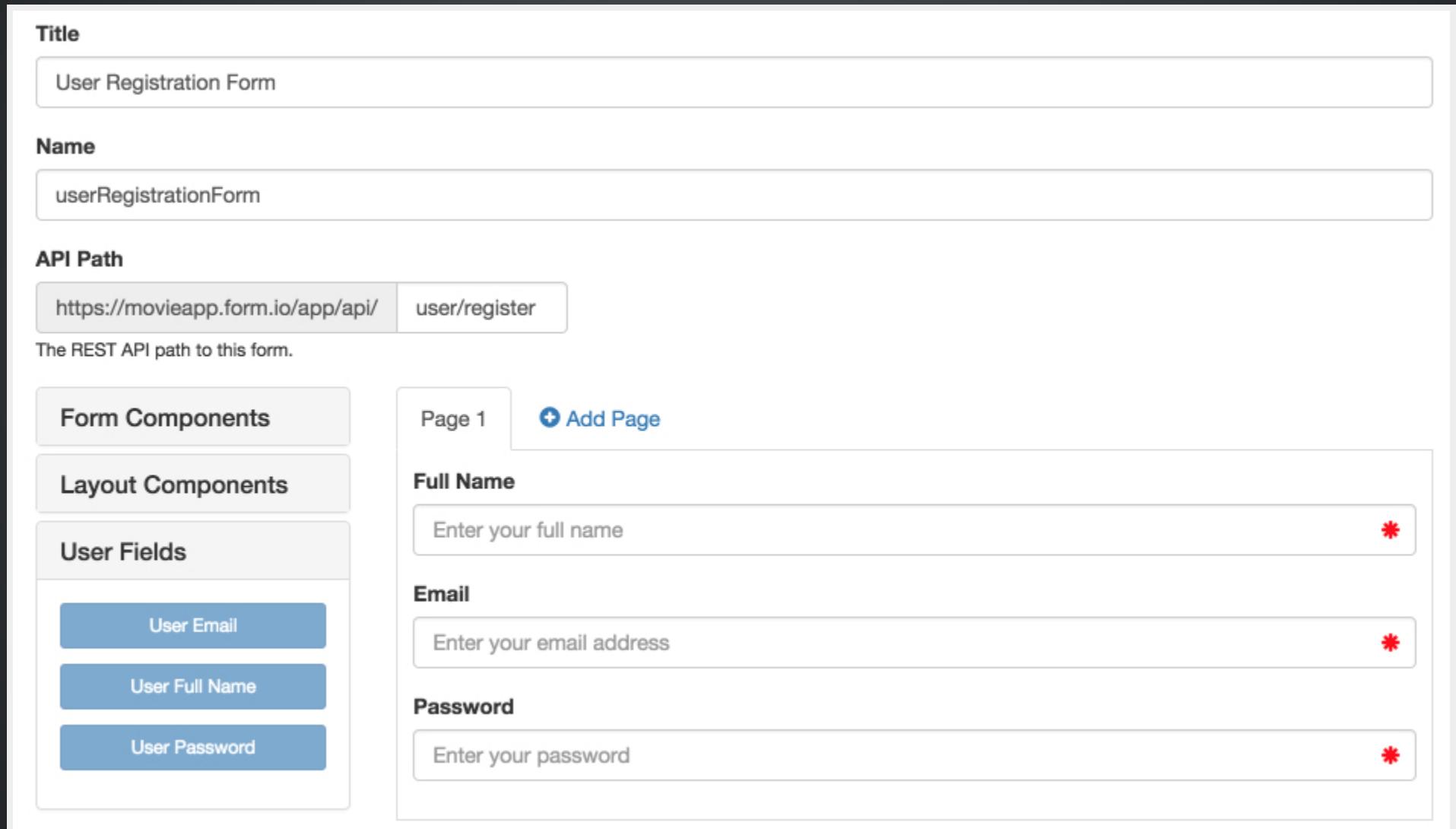
User Password

Page 1 [+ Add Page](#)

Full Name
Enter your full name *

Email
Enter your email address *

Password
Enter your password *



Implement Verify Password by adding a new Password field

Add custom validation to match the Password field.

The screenshot shows a UI builder interface with two main panels: 'Validation' and 'Preview'.

Validation Panel:

- Display:** Contains checkboxes for 'Required' (checked) and 'Minimum Length' (unchecked).
- Validation:** Contains fields for 'Minimum Length' and 'Maximum Length' (both empty).
- API:** Contains fields for 'Regular Expression Pattern' and 'Custom Validation'.

Custom Validation Code:

```
valid = (input === '{{ user.password }}') ? true :  
    'Passwords must match';
```

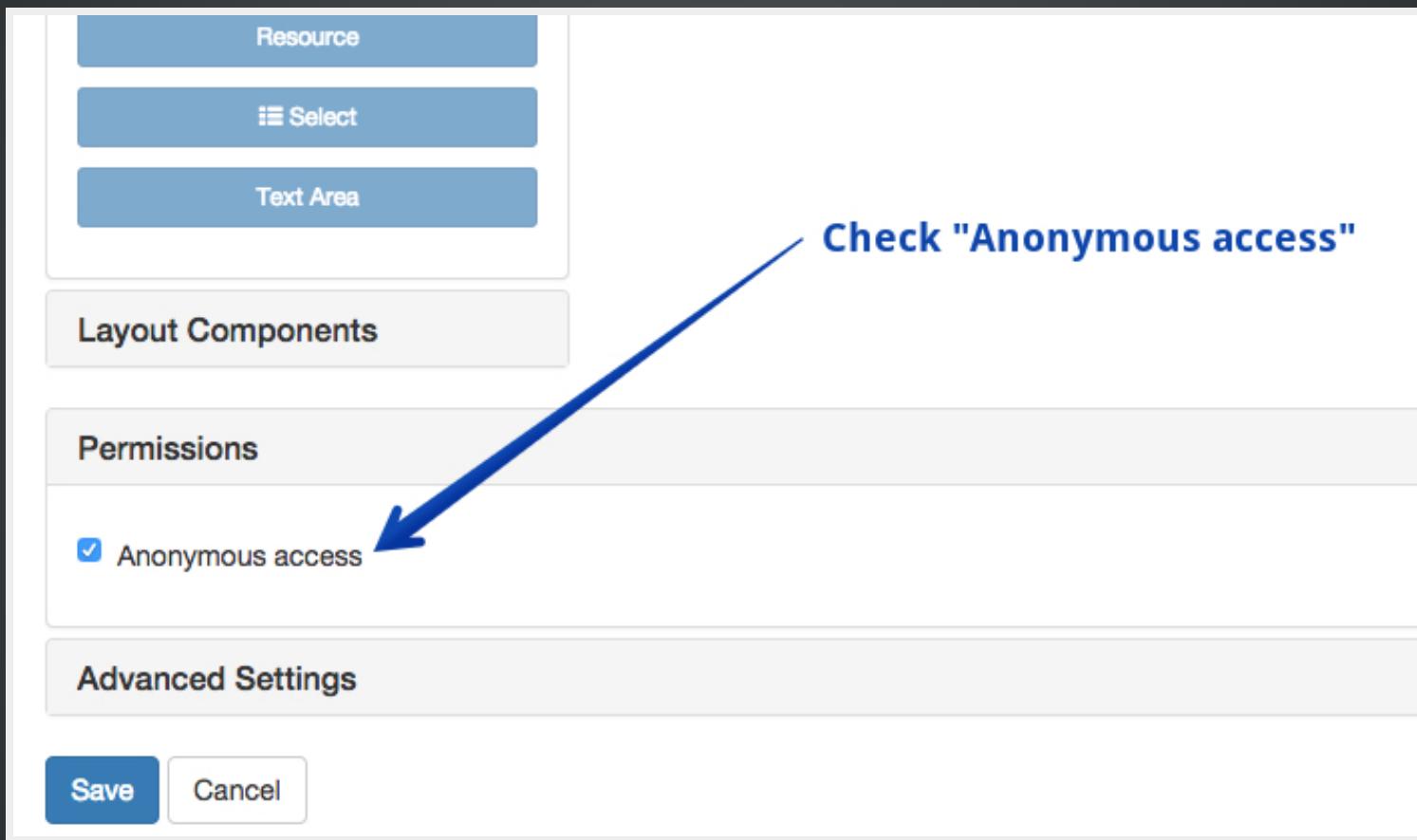
Preview Panel:

A preview window titled 'Verify Password' shows a single input field labeled 'Verify your password' with a red asterisk (*) indicating it is required. Below the input are 'Save', 'Cancel', and 'Remove' buttons.

Annotations:

- An arrow points from the explanatory text below to the 'Custom Validation' code area.
- "input" variable is provided.**
- Assign "valid" variable to either "true" for valid, or Error message if not.**
- You can use "{{ user.password }}" to refer to the current value of the user.password field.**

Check "Anonymous access" within Permissions



NOTE: This will be changing soon as we build out a more robust Roles & Permissions system.

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

The screenshot shows the form.io platform interface. At the top, there's a navigation bar with the 'form.io' logo, a user icon labeled 'codertravist', and a dropdown menu. Below the navigation, a green success message box says 'Successfully updated form!'. The main content area has a breadcrumb navigation: a home icon followed by a slash and 'movieapp'. There are four tabs: 'View' (which is selected), 'Edit', 'Actions', and 'Submissions'. On the left, there's a 'User Registration Form' panel containing fields for 'Full Name', 'Email', 'Password', and 'Verify Password', each with a red asterisk indicating it's required. A 'Submit' button is at the bottom. To the right of this panel is a 'Form API' section with a sub-section titled 'API Specification'. It contains a URL: `http://movieapp.form.io/app/api/user/register`. Below this are sections for 'Application Embedding' and 'API Specification'.

To handle Authentication, click on the **Actions** tab

Select **Authentication** action and then click **Add Action**

The screenshot shows the form.io application interface. At the top, there is a dark header with the 'form.io' logo on the left and a user profile icon on the right. Below the header, the URL 'movieapp' is displayed in a navigation bar. The main navigation menu includes 'View', 'Edit', 'Actions' (which is highlighted with a blue border), and 'Submissions'. Under the 'Actions' tab, a section titled 'Form Actions' is visible. This section contains a table with two columns: 'Name' and 'Operations'. There is one row in the table. At the bottom of this section, there is a dropdown menu set to 'Authentication' and a blue button labeled '+ Add Action'.

Select New Resource, Email, and Password

The screenshot shows the form.io application interface. At the top, the logo 'form.io' is visible next to a user icon labeled 'codertravist'. Below the header, the URL 'movieapp' is displayed. A navigation bar includes 'View', 'Edit' (which is currently selected), 'Actions', and 'Submissions'. The main content area is titled 'Settings' and contains three sections: 'Resource Association', 'Username Field', and 'Password Field'. Under 'Resource Association', a dropdown menu is set to 'New Resource'. Under 'Username Field', it is set to 'Email'. Under 'Password Field', it is set to 'Password'. A blue 'Submit' button is located at the bottom left of the form.

form.io

/ movieapp

View Edit Actions Submissions

Settings

Resource Association

New Resource

Username Field

Email

Password Field

Password

Submit

Clicking Save will now create the Login Auth action!

The screenshot shows the form.io interface for a 'movieapp' form. At the top, there's a green success message box that says 'Action was created.' A user profile icon for 'codertravist' is visible in the top right corner.

The main navigation bar includes 'View', 'Edit', 'Actions' (which is currently selected), and 'Submissions'.

The 'Form Actions' section displays a table with two columns: 'Name' and 'Operations'. It lists one action named 'Authentication'.

Name	Operations
Authentication	<input checked="" type="checkbox"/> <input type="button" value="Delete"/>

At the bottom left, there's a dropdown menu labeled 'Select an Action' and a blue button labeled '+ Add Action'.

Create a Director resource.

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

View Edit Actions Submissions

Title
Director

Name
director

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

The REST API path to this form.

Form Components

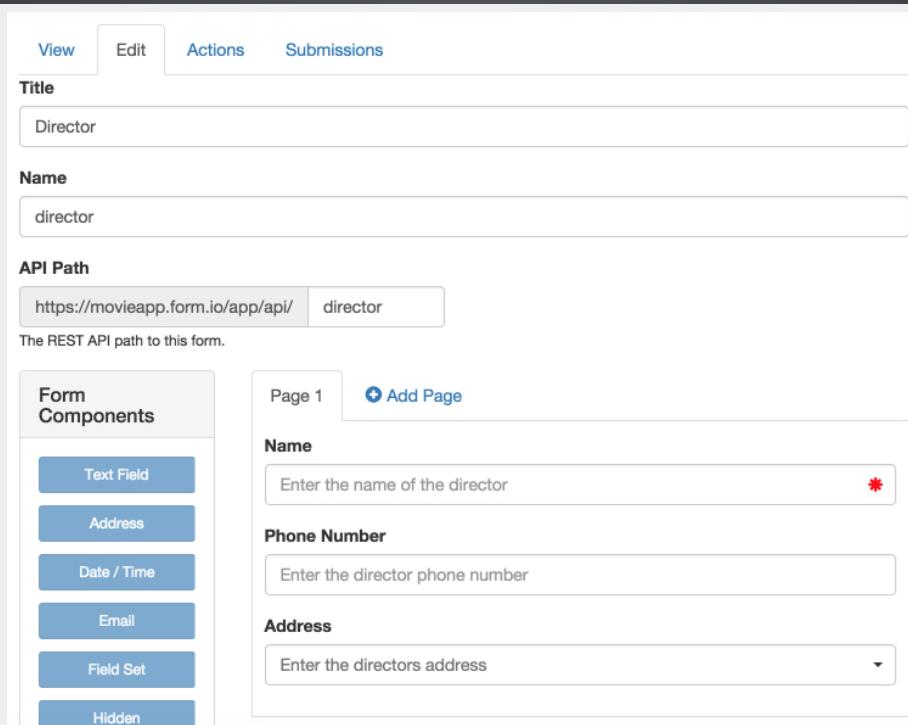
- Text Field
- Address
- Date / Time
- Email
- Field Set
- Hidden

Page 1 + Add Page

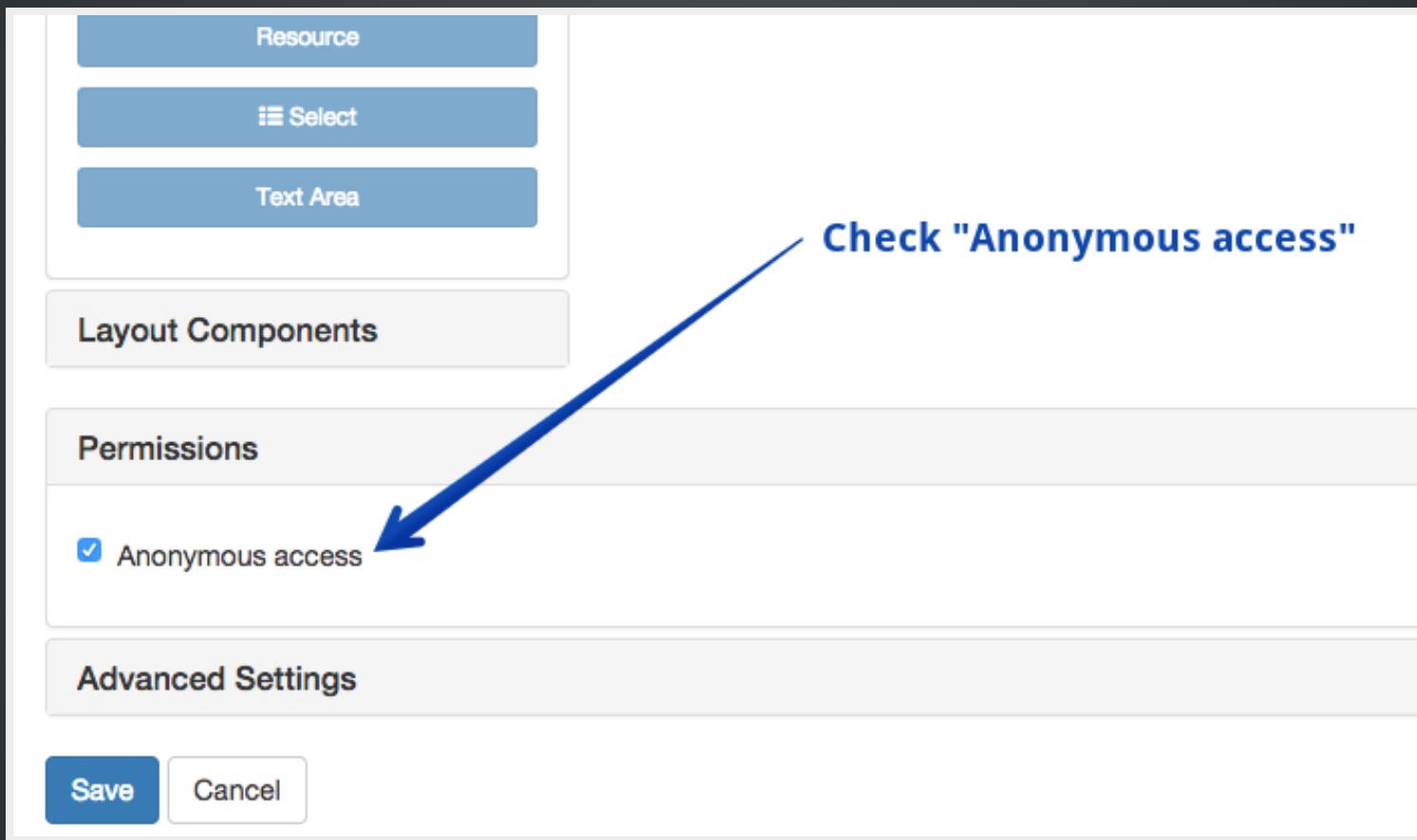
Name
Enter the name of the director *

Phone Number
Enter the director phone number

Address
Enter the directors address



Check "Anonymous access" within Permissions



NOTE: This will be changing soon as we build out a more robust Roles & Permissions system.

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

The screenshot shows the form.io interface. At the top, there's a green success message: "Successfully updated form!". Below it, the navigation bar includes a home icon, the user name "codertravist", and a dropdown arrow. The main navigation tabs are "View", "Edit" (which is active), "Actions", and "Submissions".

The left side features a form titled "Director" with three fields: "Name" (with a red asterisk indicating it's required), "Phone Number", and "Address". Each field has an input box. A "Submit" button is at the bottom of the form.

On the right, under the heading "Form API", there's a message: "Your form API is now ready to use! The following API was created that maps directly to this form interface." It also says "This API can be found by going to the following URL." Below this is a code block: `http://movieapp.form.io/app/api/director`. There are also links for "API Specification" and "Application Embedding".

Create a Movie resource.

- Text Field called Title
- Text Field called Video

Title
Movie

Name
movie

API Path
<https://movieapp.form.io/app/api/> movie

The REST API path to this form.

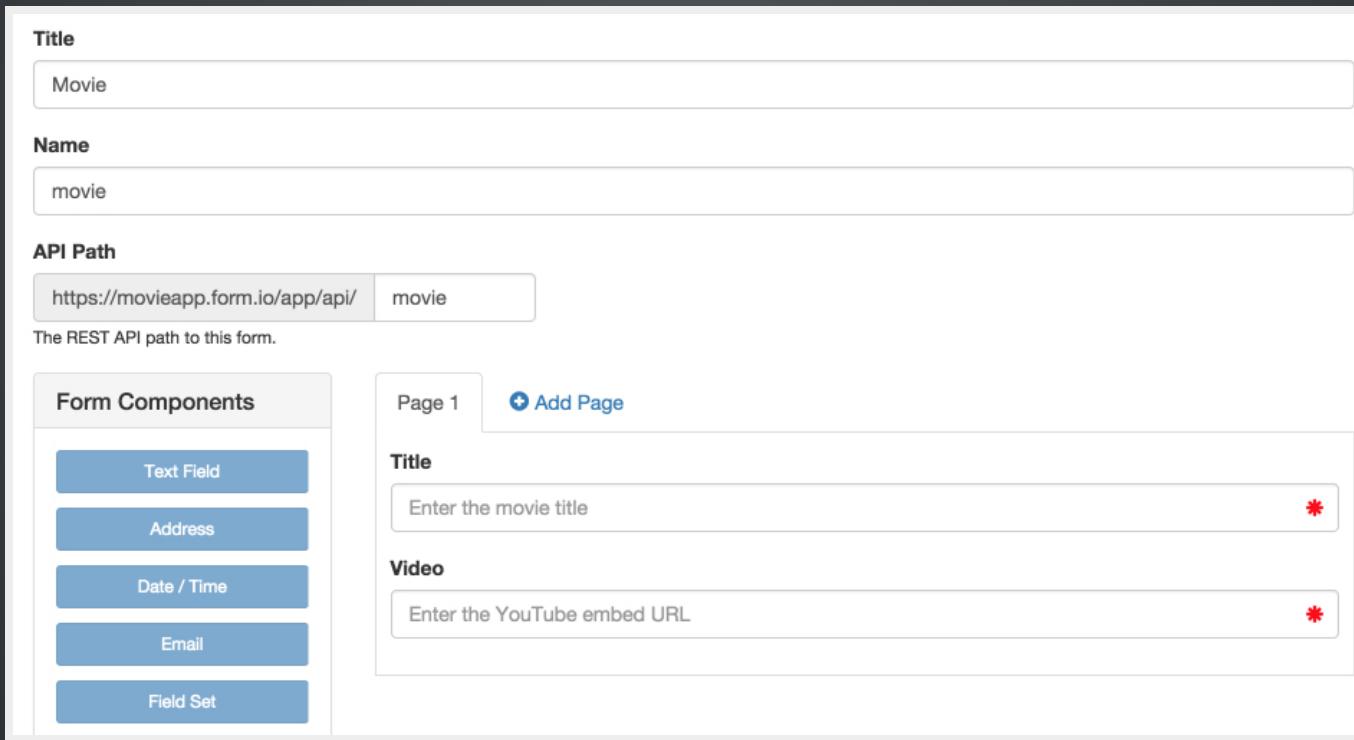
Form Components

- Text Field
- Address
- Date / Time
- Email
- Field Set

Page 1 [+ Add Page](#)

Title
Enter the movie title *

Video
Enter the YouTube embed URL *



Drag & Drop Resource field.

Select Director, change Item Template, and allow Multiple

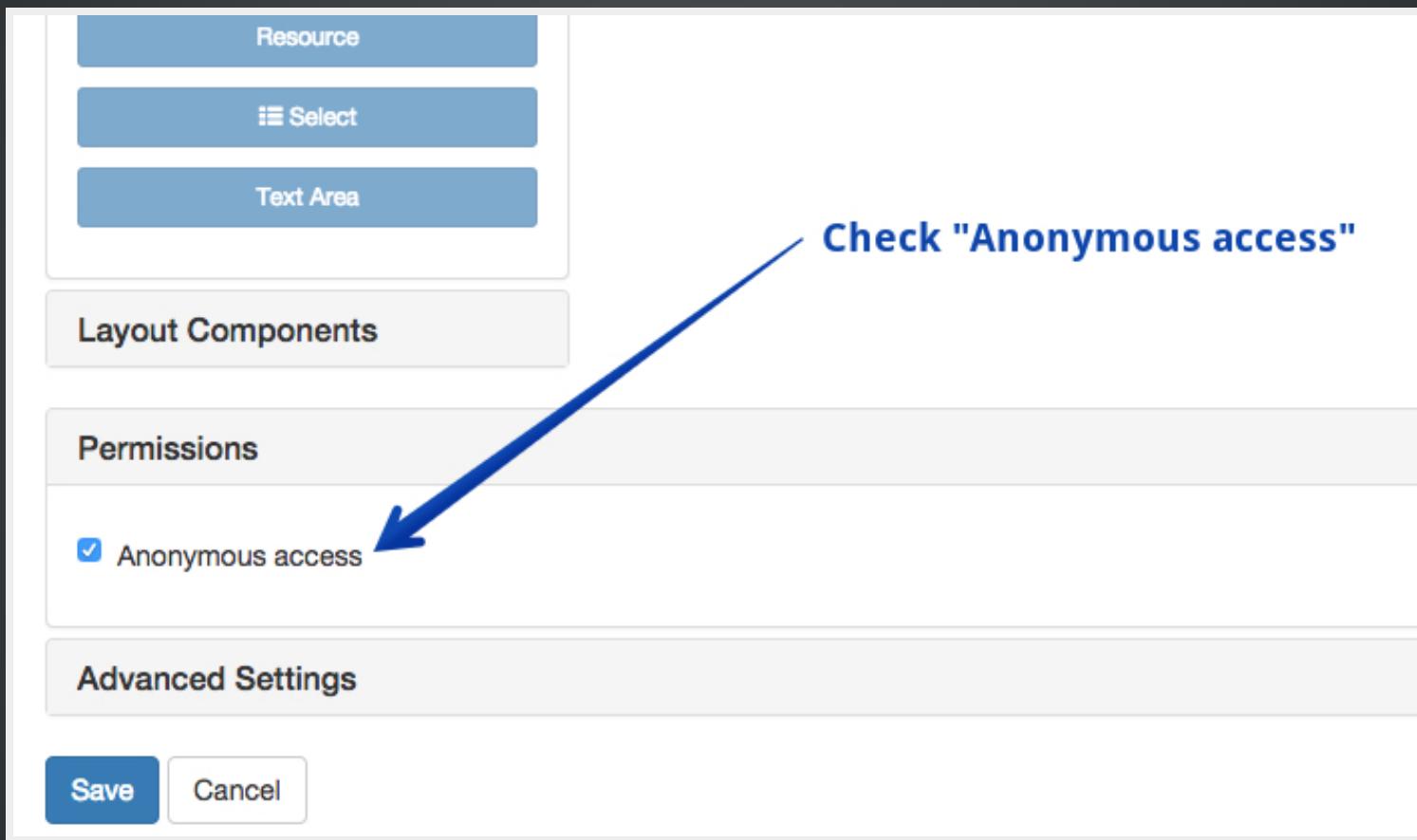
The screenshot shows two overlapping windows. The main window is titled 'Display' and contains the following fields:

- Label:** Director
- Place Holder:** Select the directors of this movie
- Resource:** Director (a dropdown menu)
- Search Expression:** The search string regular expression
- Search Fields:** The search field pairings
- Item Template:** A code editor containing the template `{{ item.data.name }}`. A blue arrow points from the text 'Make sure you change the item template to {{ item.data.name }} to show the Director "Name" field when the list shows.' to this line of code.
- Allow Multiple Resources:** A checked checkbox.

To the right of the main window is a smaller 'Preview' window titled 'Director' which shows a dropdown menu.

Make sure you change the item template to {{ item.data.name }} to show the Director "Name" field when the list shows.

Check "Anonymous access" within Permissions



NOTE: This will be changing soon as we build out a more robust Roles & Permissions system.

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

Successfully created form! ×

[Home](#) / movieapp

[View](#) [Edit](#) [Actions](#) [Submissions](#)

Movie

Title
 *

Video
 *

Director

Submit

Form API

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.

`http://movieapp.form.io/app/api/movie`

[API Specification](#)

[Application Embedding](#)

Download the Example Movie Application

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

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

SSH clone URL
git@github.com:formio/moveapp
You can clone with [HTTPS](#), [SSH](#), or [Subversion](#).

Clone in Desktop · Download ZIP

Click here to download

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.

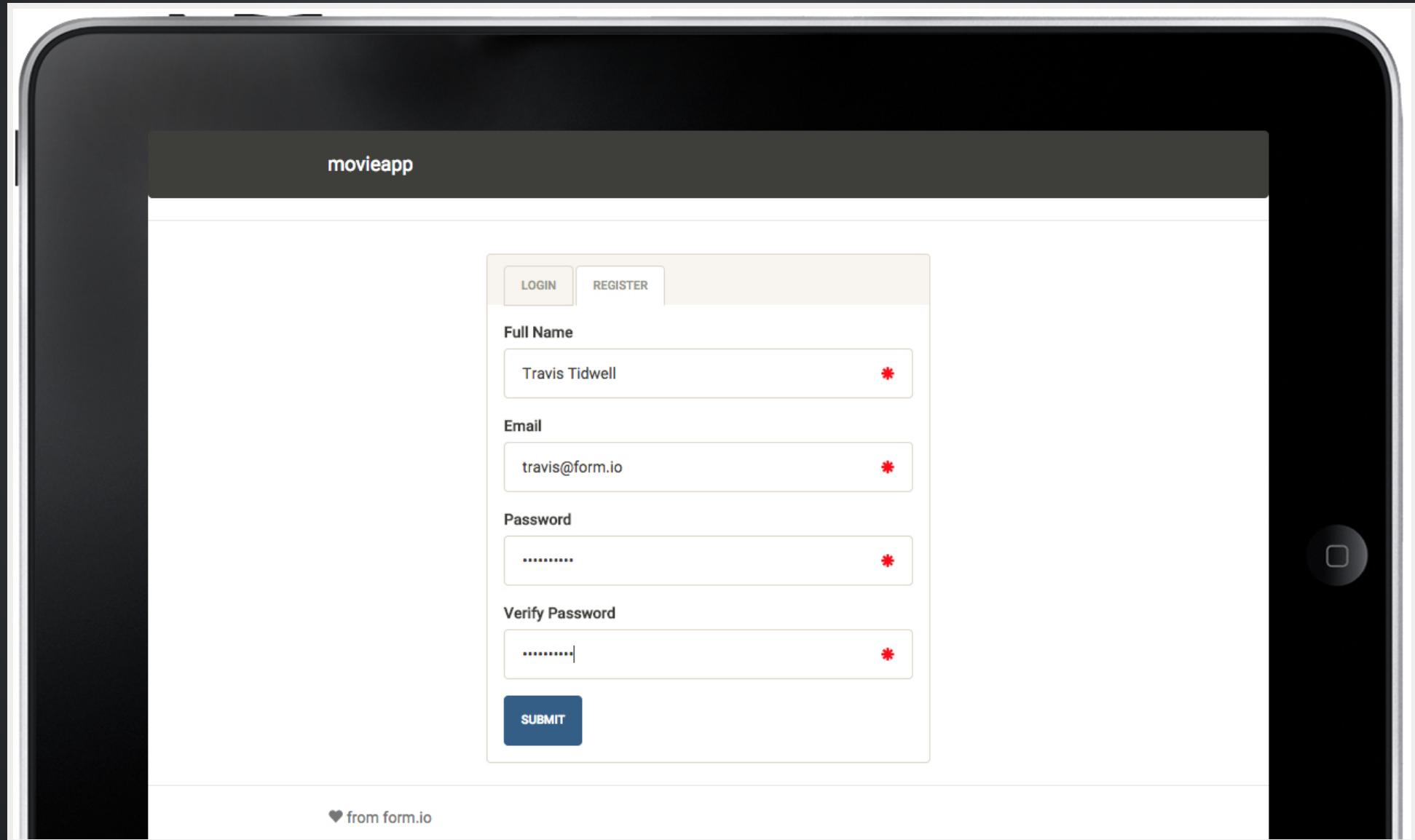
/app/scripts/app.js

```
$rootScope.userLoginForm = 'http://movieapp.form.io/app/api/user/login';
$rootScope.userRegisterForm = 'http://movieapp.form.io/app/api/user/register';
$rootScope.movieForm = 'http://movieapp.form.io/app/api/movie';
$rootScope.directorForm = 'http://movieapp.form.io/app/api/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 from the text instructions below to the "Name" column header and the "+ CREATE DIRECTOR" button.

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". At the top, there is a dark header bar with the "movieapp" logo on the left, followed by navigation links for "HOME", "MOVIES", and "DIRECTORS". On the right side of the header, it displays a greeting "HELLO TRAVIS TIDWELL" and a "LOGOUT" button. Below the header, the main content area contains a form for entering director information. The form fields include "Name" (with input "J.J. Abrams" and a red asterisk indicating required), "Phone Number" (with input "(555) 555-5555"), and "Address" (with input "2345 MAIN STREET, CORTLANDT, NY 10567, USA"). A blue "SUBMIT" button is located below the address field. At the bottom of the form, there is a 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

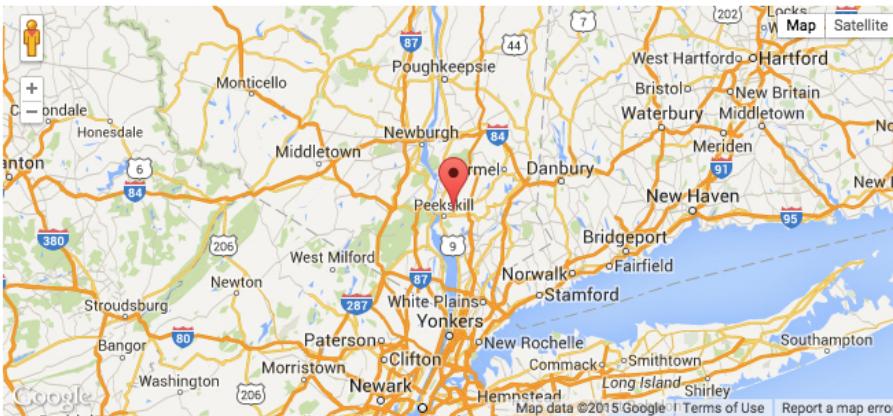
HELLO TRAVIS TIDWELL LOGOUT

VIEW EDIT DELETE

J.J. Abrams

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

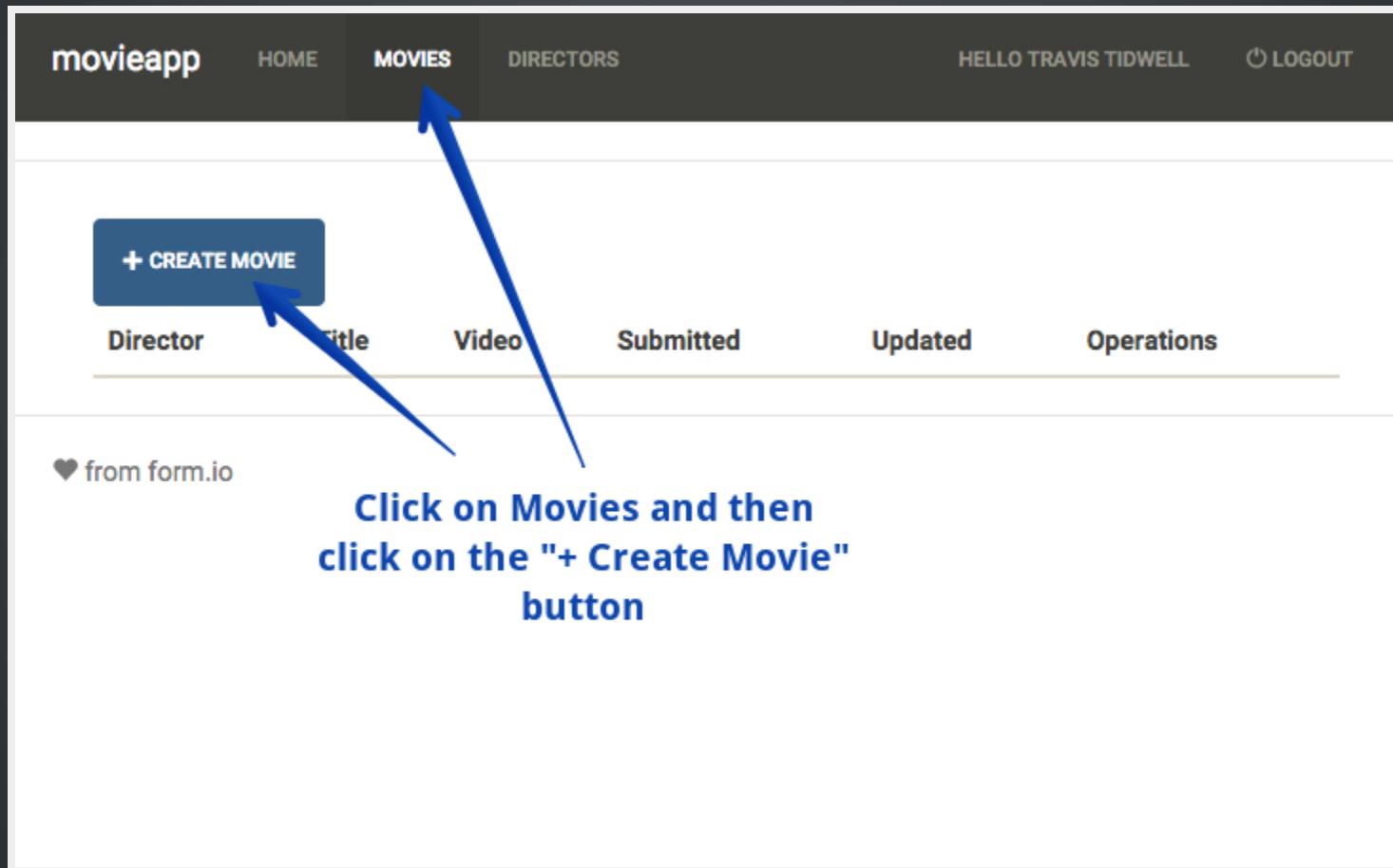
(555) 555-5555

A Google Map showing the location of J.J. Abrams' address in Cortlandt, NY. The map covers a large area of upstate New York, including the Hudson River valley and parts of Connecticut. Major cities like Albany, Troy, Poughkeepsie, and Newburgh are visible. A red marker indicates the specific location in Cortlandt. The map includes street names, route numbers (e.g., I-84, NY-9), and various geographical features like rivers and lakes.

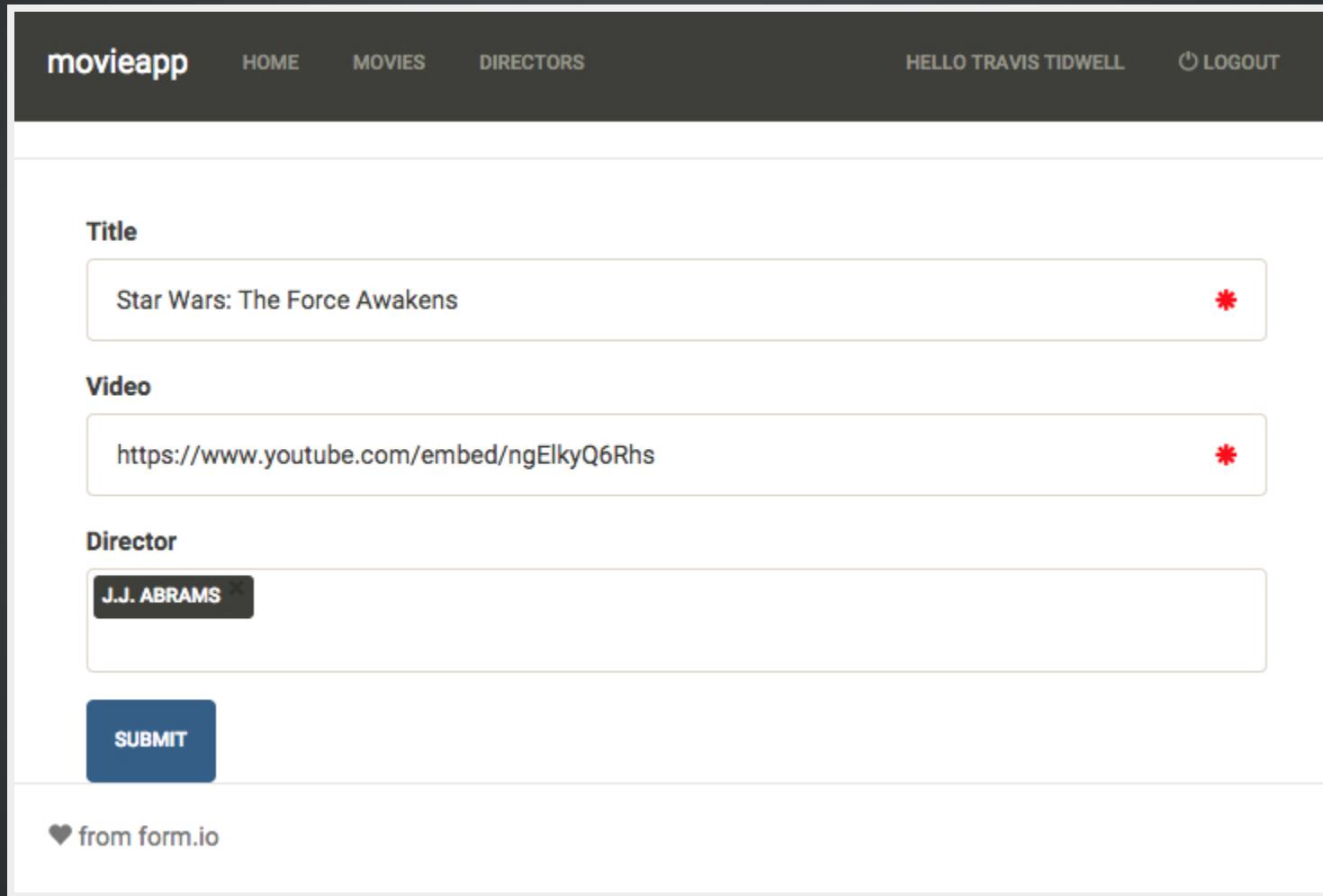
Map data ©2015 Google | Terms of Use | Report a map error

from form.io

Now go to the Movies section and create a new movie



Provide information about the movie.



The screenshot shows a dark-themed web application interface for "movieapp". At the top, there is a navigation bar with links for "HOME", "MOVIES", and "DIRECTORS". On the right side of the bar, it says "HELLO TRAVIS TIDWELL" and has a "LOGOUT" button. Below the navigation bar, there is a form for submitting movie information. The first field is labeled "Title" and contains the value "Star Wars: The Force Awakens", which is marked with a red asterisk (*) indicating it is a required field. The second field is labeled "Video" and contains the URL "https://www.youtube.com/embed/ngElkyQ6Rhs", also marked with a red asterisk (*). The third field is labeled "Director" and contains the name "J.J. ABRAMS", which is enclosed in a dark rectangular box with a white "X" icon in the top right corner, suggesting it's a dropdown or a selected item. At the bottom of the form is a blue "SUBMIT" button. In the bottom left corner of the main content area, there is a small footer note that reads "♥ from form.io".

movieapp

HOME MOVIES DIRECTORS

HELLO TRAVIS TIDWELL LOGOUT

Title

Star Wars: The Force Awakens *

Video

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

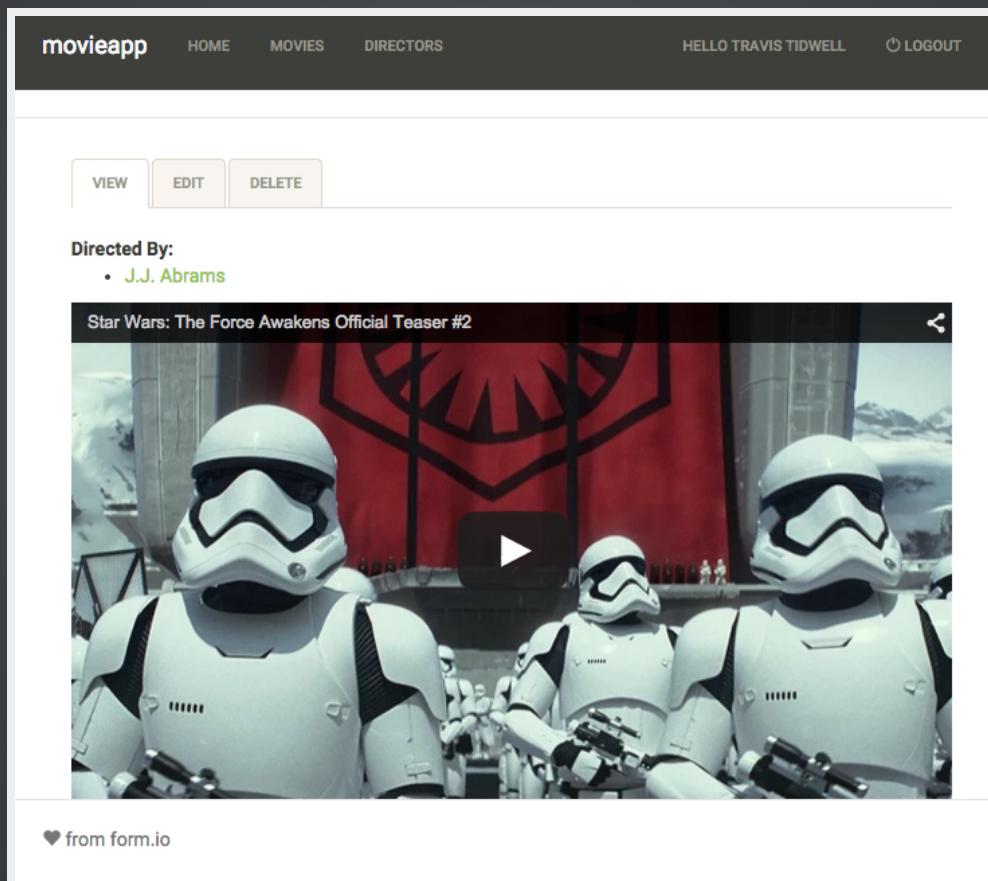
Director

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**!