

Forms and APIS

Getting data and making it work!

What is a Form?

It's how we get data from
users...that's it.

How do we create a Form?

We use the `<form></form>` tags which will hold multiple `<input>` tags. These inputs are normally referred to as form fields.

```
<form action="some/server/endpoint" id="signup-form" method="post">  
  <input name="firstName" type="input" placeholder="First Name">  
  <input name="email" type="input" placeholder="Email">  
  <input type="submit" value="Sign up!">  
</form>
```

Form Attributes

There are three main attributes for your inputs:

- Type
 - What kind of input do you want the user to see? radio button, checkbox, textbox, etc.
- Name
 - What is the name of the data that you are sending?
- Value
 - What is the value of the data you are sending?

Type Attribute

- Text
 - A single-line box for entering text
- Password
 - A single-line box for entering a password. each character is a dot.
- Radio
 - Lets a user select only one of a limited number of choices
- Checkbox
 - Lets a user select several of a limited number of choices
- Submit
 - Button that when clicked submits the form

Name Attribute

Used by the server to identify which data your form is trying to send

- Each `<input>`, `<textarea>` should have a name attribute
- Except "submit" buttons
- `<input type="text" name="firstName">`

Value Attribute

The value attribute sets the default value of the <input>.

- The user changes the value of the field by interacting with it.
- `<input type="text" value="Arthur" name="firstName">`

Form Fields In-Depth

TEXTBOX:

Attributes:

- Type="text"
- Value="some default value or blank"
- Placeholder="some text to appear when field is blank"
- Name="nameOfData"

PASSWORD:

Attributes:

- Obfuscates the value of the text field
- Same attributes as a text field

Form Fields In-Depth P.2

RADIO BUTTON:

Attributes:

- Allows selection of one item from a group of items

CHECKBOXES:

Attributes:

- Allows selection of multiple items from a group

SUBMIT BUTTON:

Attributes:

- Type="submit"
- Value="The text on the button"

Submitting Form

Generally there are two ways to submit forms.

- Hitting the enter key when you have a text field/password field in focus.
- Clicking the submit button.

The Submission Process

- Submit the Data
- Server Receives the Data
- Server sends back a response

What am I doing with Data?

WHERE TO SEND THE DATA

Out form's action is where to send the data on the server.

Examples:

- "form.php"
- "user/login"
- "http://server.com/some/server/endpoint"

HOW TO SEND THE DATA

Method - how to send the data via HTTP ("get" or "post").

- get - /someEndPoint?
firstName=Arthur&email=arthurbouie@gmail.com
- post - appends form-data inside the body of the HTTP request
(data is not shown is in URL)

Get and Post

GET EXAMPLE

<http://codepen.io/Adamor/pen/dPeJbd>

POST EXAMPLE

<http://codepen.io/Adamor/pen/QwmeaE>

Form Validation

So before your user sends data to the server with a form you'll want to make sure he/she has filled it out correctly. You use jQuery to add a listener for the submit event so you that you can check the values of the form fields.

First we need to listening for the submit event.

```
$("#login_form").submit(function (e) { // validation code goes here });
```

Then, we'll use jQuery to get the values of form fields then test if fields are filled out correctly. If they're not, prevent the user from submitting the form: `e.preventDefault();`

Styling Forms

Let's talk looks: <http://codepen.io/Adamor/pen/vERojM>

AJAX and APIs

In the old days of the web i.e before 2004 you had to refresh the page in order to get more data from the server.

- Bad user experience
- There's a brief moment where nothing is on the screen or a weird flicker
- Especially bad with a slow connection

Now, in modern hip times websites update themselves.

How did they do this...with AJAX!

AJAX: Asynchronous Javascript and XML

- It is a protocol that allows your website to exchange small bits of info with a server.
- We can use ajax to retrieve data from other servers and populate our own web page without having to reload our page
- This gives our users a better user experience

THE OLD WAY OF USING AJAX

```
XMLHttpRequest var xmlhttp = new XMLHttpRequest();
```

```
xmlhttp.onreadystatechange = function() {  
  document.getElementById("myDiv").innerHTML =  
  xmlhttp.responseText; };  
xmlhttp.open("GET", "http://arthurbouie.com/apiexample/articles",  
true);  
xmlhttp.send();
```

This is complicated and old. Don't use it...but if you want to learn more about it: https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest/Using_XMLHttpRequest

JQUERY GET, POST, AJAX METHODS

The jQuery ajax() method.

jQuery takes care of the XMLHttpRequests so that you don't have to.

The object you pass this function needs to have:

- url - The URL of the data you are trying to get
- dataType - "html", "text", "json", or "jsonp"
- success - A function that takes a single parameter. It gets called when the data is retrieved.

Good to know - P.1

JAVASCRIPT OBJECT NOTATION (JSON)

It is a stringified version of a javascript object that allows you to send javascript objects between the server and the client. To make JSON in the wild readable you'll need either a "JSON formatter" which you can find online or in the chrome store as a plugin.

Good to know - P.2

CROSS DOMAIN REQUEST

Let's say you are making a website that is hosted on yourname.com and you want to access data from facebook.com with AJAX. Your browser prohibits this -- it's a security measure.

You'll receive an error because you're trying to access data from the domain "openweathermap.org", but your domain is "localhost".

JSONP TO THE RESCUE

JSONP is json with padding and it's a hack to get around the cross-domain restriction.

APIs

What is an API?

Application programming interface (API). It's a way for an application (like facebook) to share its data/functionality with another application (like yours).

Examples: Twitter: Which let's you can post tweets from another site

Google: Who you can use google to sign into services (trello.com)
etc...

Let's use one!

Open weather map API

Free API for world weather data. You can use this API with parameters in the query string to control what kinds of data you get back.

Say we want weather data for Brooklyn:

[http://api.openweathermap.org/data/2.5/weather?](http://api.openweathermap.org/data/2.5/weather?zip=11215,us&appid=a09842d1e7c4c8d8d83db0e38aa624d7)

[zip=11215,us&appid=a09842d1e7c4c8d8d83db0e38aa624d7](http://api.openweathermap.org/data/2.5/weather?zip=11215,us&appid=a09842d1e7c4c8d8d83db0e38aa624d7)

The ? is the URL approve marks the start of our Query string.

Optional Homework

Relaxr Registration Form

Technical Requirements

- Use at least three different types of HTML5 form elements: input, textarea, select, radio button, dropdown, checkbox
- Link to your sign-up page from different pages in your application
- Style forms lightly using CSS
- Make sure a user's password is not visible when they are entering it

Optional Homework

Relaxr Registration Form

Bonus

- Use responsive media queries to allow for proper rendering on mobile devices
- Use jQuery and display an error message if any of the input fields are left blank upon submission
- To replicate application flow, add another page that says "Thanks for Relaxing with us!" after submission