Ajax Forms

$.ajax with POST

jQuery Travels - Trip Booker

Destination: Select a Destination ⇕

Tickets: 1

BOOK TRIP

Call us at 555-25937 to make a reservation today!

$.ajax with POST

jQuery Travels - Trip Booker

Destination: Select a Destination

Tickets: 1

BOOK TRIP

Call us at 555-25937 to make a reservation today!

What if we want to submit this form via ajax?

application.js

```javascript
$('form').on('submit', function(e) {
  $.ajax('/book', {
    type: 'POST'
  });
});
```

Will do a POST request to the server, which is used for forms

jQuery Travels - Trip Booker

Destination: Select a Destination ⬍

Tickets: 1

BOOK TRIP

Call us at 555-25937 to make a reservation today!

Preventing Double Submission

application.js

```javascript
$('form').on('submit', function(event) {
  event.preventDefault();
  $.ajax('/book', {
    type: 'POST'
  });
});
```

Prevents browser from submitting

❌ This would do the AJAX request, but it's not sending any form data over.

index.html

```html
<form action='/book'>
  <select id='destination' name='destination'>
    ...
  </select>
  <input type='text' id='quantity' name='quantity' value='1' />
</form>
```

How would we pass the values of this form via $.ajax?

We could use the data option and grab all the values

application.js

```javascript
$('form').on('submit', function(event) {
  event.preventDefault();
  $.ajax('/book', {
    type: 'POST',
    data: { "destination": $('#destination').val(),
            "quantity": $('#quantity').val() }

  });
});
```

Works, but there's a better way

index.html

```html
<form action='/book'>
  <select id='destination' name='destination'>
    ...
  </select>
  <input type='text' id='quantity' name='quantity' value='1' />
</form>
```

Serialize packages all the values for us

application.js

```javascript
$('form').on('submit', function(event) {
  event.preventDefault();
  $.ajax('/book', {
    type: 'POST',
    data: $('form').serialize()
  });
});
```

jQuery Object Methods

```
serialize()
```

Used to merge all form fields for submission

index.html

```html
<form action='/book'>
  <select id='destination' name='destination'>
    ...
  </select>
  <input type='text' id='quantity' name='quantity' value='1' />
</form>
```

Add AJAX callback (just like you saw in level 1)

application.js

```javascript
$('form').on('submit', function(event) {
  event.preventDefault();
  $.ajax('/book', {
    type: 'POST',
    data: $('form').serialize(),
    success: function(result) {  }
  });
});
```

*Like our previous $.ajax request,
success is called with the HTML result*

result

```
<p>Your vacation to Paris, France has been booked for $2,196.00 for 4 nights. Confirmation #345feab.</p>
```

$.ajax success callback

```javascript
$('form').on('submit', function(event) {
  event.preventDefault();
  $.ajax('/book', {
    type: 'POST',
    data: $('form').serialize(),
    success: function(result) {
      $('form').remove();
      $('#vacation').hide().html(result).fadeIn();
    }
  });
});
```

There's duplication of DOM lookups

application.js

```javascript
$('form').on('submit', function(event) {
  event.preventDefault();
  $.ajax('/book', {
    type: 'POST',
    data: $('form').serialize(),
    success: function(result) {
      $('form').remove();
      $('#vacation').hide().html(result).fadeIn();
    }
  });
});
```

Eliminating duplication of DOM lookups

application.js

```javascript
$('form').on('submit', function(event) {
  event.preventDefault();
  var form = $(this);
  $.ajax('/book', {
    type: 'POST',
    data: form.serialize(),
    success: function(result) {
      form.remove();
      $('#vacation').hide().html(result).fadeIn();
    }
  });
});
```

Save a reference to the form to reduce DOM queries

jQuery Travels - Trip Booker

Destination: Select a Destination

Tickets: 1

BOOK TRIP

Call us at 555-25937 to make a reservation today!

Ajax with JSON

JavaScript Object Notation

A standard way to format data

"HTML"

"JSON"

```html
<p>Your vacation to Paris,
France has been booked for
$2,196.00 for 4 nights.
Confirmation #345feab.</p>
```

```json
{
    totalPrice: 2196.00,
    nights: 4,
    location: 'Paris, France',
    confirmation: '345feab'
}
```

*Server decides how the
HTML is formatted*

*Server returns details,
we can use it as we wish*

Our old code wouldn't work

application.js

```javascript
$('form').on('submit', function(event) {
  event.preventDefault();
  var form = $(this);
  $.ajax('/book', {
    type: 'POST',
    data: form.serialize(),
    success: function(result) {
      form.remove();
      $('#vacation').hide().html(result).fadeIn();
    }
  });
});
```

This code is expecting HTML to get returned, and injecting the result right into the page.

$.ajax with POST and JSON

```
var form = $(this);
$.ajax('/book', {
    type: 'POST',
    data: form.serialize(),
    dataType: 'json',
    success: function(result) {
        form.remove();
        $('#vacation').hide().html(result).fadeIn();
    },
    contentType: 'application/json'
});
```

!

Parse the response as JSON

Result isn't HTML, we'll need to create a DOM node here

Ask the server to respond with JSON

Reading the JSON results and creating HTML

```
success: function(result) {
  form.remove();
  var msg = $("<p></p>");
  msg.append("Destination: " + result.location + ". ");
  msg.append("Price: " + result.totalPrice + ". ");
  msg.append("Nights: " + result.nights + ". ");
  msg.append("Confirmation: " + result.confirmation+ ".");

  $('#vacation').hide().html(msg).fadeIn();
}
```

Create and add a DOM node

JSON is converted into a JavaScript Object

result

```
{
  location: 'Paris',
  totalPrice: 1196.00,
  nights: 4,
  confirmation: '345feab'
}
```

<p>Destination: Paris. Price: $2196.00. Nights: 4. Confirmation: 345feab.</p>

Final event handler with POST and JSON

```javascript
$('form').on('submit', function(e) {
  event.preventDefault();
  var form = $(this);
  $.ajax('/book', {
    type: 'POST',
    contentType: 'application/json',
    dataType: 'json',
    data: form.serialize(),
    success: function(result) {
      form.remove();
      var msg = $("<p></p>");
      msg.append("Destination: " + result.location + ". ");
      msg.append("Price: " + result.totalPrice + ". ");
      msg.append("Nights: " + result.nights + ". ");
      msg.append("Confirmation: " + result.confirmation+ ".");
      $('#vacation').hide().html(msg).fadeIn();
    },
    contentType: 'application/json'
  });
});
```

Don't Repeat Yourself

application.js

```
$.ajax('/book', { ... });
```
⚠️

index.html

```
<form action='/book' method='POST'>
  ...
</form>
```

Form URL is in both HTML and JavaScript

application.js

```
$.ajax($('form').attr('action'),{ ... });
```
✅

Will use the action from the form HTML

jQuery Object Methods

```
attr(<attribute>)
```

```
attr(<attribute>, <value>)
```

Final event handler with POST and JSON

```javascript
$('form').on('submit', function(e) {
  event.preventDefault();
  var form = $(this);
  $.ajax($('form').attr('action'),{
    type: 'POST',
    contentType: 'application/json',
    dataType: 'json',
    data: form.serialize(),
    success: function(result) {
      form.remove();
      var msg = $("<p></p>");
      msg.append("Destination: " + result.location + ". ");
      msg.append("Price: " + result.totalPrice + ". ");
      msg.append("Nights: " + result.nights + ". ");
      msg.append("Confirmation: " + result.confirmation+ ".");
      $('#vacation').hide().html(msg).fadeIn();
    }
  });
});
```

jQuery Travels - Trip Booker

Destination: Select a Destination

Tickets: 1

BOOK TRIP

Call us at 555-25937 to make a reservation today!