



Ajax Forms

\$.ajax with POST

jQuery Travels - Trip Booker

Destination: 

Tickets:

BOOK TRIP

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

\$.ajax with POST

jQuery Travels - Trip Booker

Destination:

Tickets:

BOOK TRIP

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

*What if we want to submit
this form via ajax?*

application.js

```
$( '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:

Tickets:

BOOK TRIP

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

Preventing Double Submission

application.js

```
$('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

```
<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

```
$('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

```
<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

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

index.html

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

jQuery Object Methods

serialize()

Used to merge all form fields for submission

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

application.js

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

result

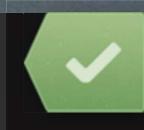
*Like our previous `$.ajax` request,
success is called with the HTML result*

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

\$.ajax success callback

application.js

```
$( '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

```
$('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

```
$('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: 

Tickets:

BOOK TRIP

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