

 [ajax_cheat_sheet.md](#)

#AJAX

What is AJAX

"AJAX an acronym for asynchronous JavaScript and XML is a group of interrelated web development techniques used on the client-side to create asynchronous web applications. With Ajax, web applications can send data to, and retrieve data from, a server asynchronously (in the background) without interfering with the display and behavior of the existing page. Data can be retrieved using the XMLHttpRequest object. Despite the name, the use of XML is not required (JSON is often used instead), and the requests do not need to be asynchronous." - [wikipedia](#)

AJAX with jQuery

jQuery provides a `$.ajax()` method with a set of options for sending requests and callback methods to handle responses.

- Here is the most basic `$.ajax()` request that does not send any data. It will be handled by the `post '/trips'` route on the server. You can choose any of the http request verbs for the type parameter (`get`, `post`, `put`, `delete`)

```
$.ajax({
  url: "/trips",
  type: "post"
});
```

Send data in a request

The `$.ajax()` method allows data to be sent in a request in a variety of formats. It also provides shorthand methods `$.get()` and `$.post()` which alias the `$.ajax()` method.

Form data

When using AJAX to post a form, you can also use the `.serialize()` method which creates a string of `key : value` pairs from the form data. You can also use the original `action` and `method` attributes from the form for the `url` and `type` parameters.

```
$('#form').on('submit', function(e){
  e.preventDefault()
  var form_data = $(this).serialize()
  $.ajax({
    url: this.action,
    type: this.method,
    data: form_data
  });
});
```

Register a callback function to verify the request

The `$.ajax()` method has three callback functions that are invoked on the response object. The `.done()` callback is invoked if the request is successful, the `.fail()` callback is invoked if the request fails and the `.always()` is invoked regardless of success or failure. Start with a `console.log()` in the done and fail callbacks to make sure your AJAX is working.

```
$.ajax({
  url: this.action,
  type: this.method,
  data: $(this).serialize()
}).done(function(server_data){
  console.log("success" + server_data);
  //Once you have success, use jQuery to d
}).fail(function(jqXHR, textStatus, errorT
  console.log("fail" + errorThrown);
});
```

Receive data from the server

ERB Partials

You can return a partial form from the server and then append it to the DOM using jQuery

```
#THE SERVER CODE
post '/trips' do
  @trips = Trip.create(params[:location])
  erb :_trips , :layout => false #_trips
end
```

```
\\THE JAVASCRIPT

$('#trips').on('submit',function(e) {
  e.preventDefault();
  $.ajax({
    url: this.action,
    type: this.method,
    data:$(this).serialize()
  }).done(function(server_data) {
    console.log("SUCCESS -- server returned")
    $('#new-trip').html(server_data); //
  }).fail(function(){
    console.log('fail');
  });
});
```

JSON

Returning JSON data from the server allows flexibility on the client side as to how that data is displayed.

```
#THE SERVER CODE
# Be sure to require 'json' in your environment

get '/trips' do
  content_type :json
  {:location => "Corsica", :price=> 2200}.to_json
end
```

```
//THE JAVASCRIPT
$('#cool-trip').on('click', function(e){
  e.preventDefault();
  $.ajax({
    type:$(this).attr('data-type'), //use
    url:$(this).attr('href'),
  }).done(function(server_data){
    $('#show_trip').append("<div> "
      + server_data.location + " COST: "
      + "<a class='delete' href='/cool-trips'>Delete</a> "
      + "<a href='/cool-trips' class='update'>Update</a>"
    );
  }).fail(function(){
    console.log("fail");
  });
});
```

```
THE HTML

<a id="cool-trip" href="/trips" data-type="JSON">Cool trip</a>
<div id="#show_trip"></div>
```

String data

You can also return basic string data from a request but JSON is generally more flexible as it allows multiple key:value pairs.

```
#THE SERVER CODE

get '/trips' do
  "this is a cool place to visit"
end
```

```
//THE JAVASCRIPT

$('#cool-trip').on('click', function(e){
  e.preventDefault();
  $.ajax({
    type:$(this).attr('data-type'),
    url:$(this).attr('href'),
  }).done(function(server_data){
    $('#show_trip').append("<div> "
      + server_data + "</div>"
    );
  }).fail(function(){
    console.log("fail");
  });
});
```

Resources

- [jQuery's ajax function](#)
- [jQuery's ajax Events](#)
- [You might not need jQuery translates jQ to JS](#)

Comment on gist

Sign in to comment

or [sign up](#) to join this conversation on GitHub

Desktop version