

# Web Services and Applications

## Lab 09.4: AJAX calls to REST

---

*Lecturer: Andrew Beatty*

In this lab we want to make the AJAX calls that will call the RESTful API

At <http://andrewbeatty1.pythonanywhere.com>

This is so you do not have to start a flask server or to test the code. If you integrating this into your own web application you should use relative URLs eg `"/books"`

- Remember the server serves out a RESTfull api that performs CRUD operations on books.

Action	Method	URL	Sample params	Sample return
Get all	GET	/books	none	[ { "title":"blah", ... }, {...},{...} ]
Find by id	GET	/books/ <i>id</i>	none	{ "title":"blah", "Author":"blah", "id":23, "price":1250 }
Create	POST	/books	{ "title":"blah", "Author":"blah", "price":1250 }	{ "title":"blah", "Author":"blah", "id":23, "price":1250 }
Update	PUT	/books/ <i>id</i>	{ "price":1300 }	{ "title":"blah", "Author":"blah", "id":23, "price":1300 }
delete	DELETE	/books/ <i>id</i>	none	{ "done":true }

1. Test that we can call the server from the web page, by creating a webpage called testgetall.html.

```
<html>
  <head>

    <title>
      test get all
    </title>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquery.min.js"></script>
  </head>

  <body>
    <script>
      function getAll(){
        $.ajax({
          "url": " http://andrewbeatty1.pythonanywhere.com/books",
          "method":"GET",
          "data":"",
          "dataType": "JSON",
          "success":function(result){
            console.log(result);
          },
          "error":function(xhr,status,error){
            console.log("error: "+status+" msg:"+error);
          }
        });
      }
      getAll()
    </script>
  </body>
</html>
```

This URL is my site on python anywhere, this is so that you can test this code without starting a flask server.

If you are putting this into a web application, you should use a relative address. ie `"/books"`

2. If nothing happens in the console, check that you called the function getAll()
3. Later we will be opening the page through a local server  
<http://127.0.0.1:5000/testgetall.html> You may have cross site scripting protection issues.

but for now, you will be opening the html page through the file system, so you will have something like this as your URL. <file:///C:/user/me/wsaa/code/topic06-REST/testgetall.html>

4. Add a callback function to the parameters and call it in the success. This way we can have

```
<script>
function getAll(callback){
  $.ajax({
    "url": "http://andrewbeatty1.pythonanywhere.com/books",
    "method": "GET",
    "data": "",
    "dataType": "JSON",
    "success": function(result){
      //console.log(result);
      callback(result)
    },
    "error": function(xhr, status, error){
      console.log("error: "+status+" msg: "+error);
    }
  });
}
function processGetAll(result){
  console.log("in process")
  console.log(result)
}
getAll(processGetAll)
```

Again! This is URL should be relative when used in a web application

the functionality of what we do with the data separate from retrieving the data.

## Create

5. Make a web page that tests the create called testCreate.html (I have put this code into the same file as above in the sample code).

```
function createBook(book, callback){
    //console.log(JSON.stringify(book));
    $.ajax({
        "url": "http://andrewbeatty1.pythonanywhere.com/books",
        "method": "POST",
        "data": JSON.stringify(book),
        "dataType": "JSON",
        "contentType": "application/json; charset=utf-8",
        "success": function(result){
            console.log(result);
            //callback(result)
        },
        "error": function(xhr, status, error){
            console.log("error: "+status+" msg: "+error);
        }
    });
}
```

Again! This is URL should be relative when used in a web application

contentType  
is the type of the data that is being passed to the server (in data)

6. Test it first without the callback function and then with it

```
function processCreatoresponse(result){
    console.log(result)
}

book = {"title": "javascript", "author": "andrew", "price": 12}
createBook(book, processCreatoresponse)
```

These names need to match what is received by the server

## Update

### 7. Test the update

```
function updateBook(book, callback){  
  
    //console.log(JSON.stringify(book));  
    $.ajax({  
        "url": "http://andrewbeatty1.pythonanywhere.com/books/"+encodeURIComponent(book.id),  
        "method": "PUT",  
        "data": JSON.stringify(book),  
        "dataType": "JSON",  
        "contentType": "application/json; charset=utf-8",  
        "success": function(result){  
            console.log(result);  
            //callback(result)  
        },  
        "error": function(xhr, status, error){  
            console.log("error: "+status+" msg:"+error);  
        }  
    });  
}
```

Again! This is URL should be relative when used in a web application.  
/books/id

### 8. Firstly, without the call back function then with the callback

```
function processUpdateResponse(result){  
  
    console.log(result)  
}  
book = {id:155,"price":999}  
updateBook(book,processUpdateResponse)
```

## Delete

### 9. Test delete

```
function deleteBook(id, callback){
    $.ajax({
        "url": "http://andrewbeatty1.pythonanywhere.com/books/"+id,
        "method": "DELETE",
        "data": "",
        "dataType": "JSON",
        contentType: "application/json; charset=utf-8",
        "success": function(result){
            console.log(result);
            //callback(result)
        },
        "error": function(xhr, status, error){
            console.log("error: "+status+" msg:"+error);
        }
    });
}
```

Again! This is URL should be relative when used in a web application.

/books/id

### 10. Again, test without the callback then with it

```
function processDeleteResponse(result){
    console.log("in process delete")
    console.log(result)
}
deleteBook(155, processUpdateResponse)
```

## How would you integrate this with lab06.02

11. So we have sample code in the test section, comment out the function calls and you can include this file at the top of the lab06.02, then put the calls into the appropriate places.
12. I have made a html called lab06.3b-book-interface.html
13. You should call getAll at the bottom of the page to initialise the table
14. In the processGetAll use a for loop to get each book individually, you would use this with the add to table to add them to the table.

```
function processGetAllResponse(result){  
  
    console.log("in process")  
    //console.log(result)  
    for (book of result){  
        //console.log(book)  
        // You could get away without doing this, I just want to point out  
        // that the attributes in the book that is returned by the server  
        // may not be the same as the attributes used in the page  
        displayBook = {}  
        displayBook.id = book.id  
        displayBook.author = book.author  
        displayBook.title = book.title  
        displayBook.price = book.price  
        // you can now pass it to addBookToTable  
        console.log(displayBook)  
    }  
}
```

15. You will have done a similar conversion in the doCreate, do Update and doDelete.
16. Remember there may be multiple people working on this data.