

HW5 Review

Saving data on a Server with Flask

Goal: Save Data on the Server

Columbia Paper Infinity

Log your paper sales:

<input type="text" value="Client"/>	<input type="text" value="# Reams"/>	<input type="button" value="Submit"/>
James D. Halpert	Shake Shack	100
Stanley Hudson	Toast	400
Michael G. Scott	Computer Science Department	1000

Each row in the table has a yellow button with an 'X' icon to its right.

Create / Delete data

Party Planning Committee

Non-PPC	PPC
1: Phyllis	
2: Angela	
3: Dwight	
4: Oscar	
5: Creed	
6: Pam	
7: Jim	
8: Stanley	

Update data

Original code (doesn't save)

```
1 <html>
2 <head>
3   <script src="http://code.jquery.com/jquery-3.3.1.min.js"></script>
4   <script>
5     var data = [{ data|tojson }];
6
7     // Shorthand for $( document ).ready()
8     $(document).ready(function(){
9       //when the page loads, display all the names
10      displayNames(data)
11
12      $("#submit_name").click(function(){
13        var name = $("#new_name").val()
14        console.log(name)
15
16        var new_id = data.length + 1
17        var new_name = name
18        var new_data = {
19          "id": new_id,
20          "name": new_name
21        }
22        data.push(new_data)
23        displayNames(data)
24      })
25    }
26  </script>
```

The browser window shows the URL `127.0.0.1:5000/people`. The page content includes the heading **Hello people!**, a form with an input field containing `chilton` and a `Submit` button, and a list of names: `michael scott`, `jim halpert`, and `chilton`.

The browser's developer console is open, showing the `data` array in the console. The array contains three objects:

- `0: {id: 1, name: "michael scott"}`
- `1: {id: 2, name: "jim halpert"}`
- `2: {id: 3, name: "chilton"}`

The console also shows `length: 3` and `__proto__: Array(0)`.

Save the data to the server with an AJAX call

```
people.html x server.py x hello.html x
1 <html>
2 <head>
3 <script src="http://code.jquery.com/jquery-3.3.1.min.js"></script>
4 <script>
5     var data = {{ data|tojson }};
6
7     // Shorthand for $( document ).ready()
8     $(document).ready(function(){
9         //when the page loads, display all the names
10        displayNames(data)
11
12        $("#submit_name").click(function(){
13            var name = $("#new_name").val()
14            console.log(name)
15
16            saveName(name)
17
18
19
20
21
22
23
24
25
```

```
var saveName = function(name){
    var data_to_save = {"name": name}
    $.ajax({
        type: "POST",
        url: "add_name",
        dataType: "json",
        contentType: "application/json; charset=utf-8",
        data: JSON.stringify(data_to_save),
        success: function(result){
            var all_data = result["data"]
            data = all_data
            displayNames(data)
        },
        error: function(request, status, error){
            console.log("Error");
            console.log(request)
            console.log(status)
            console.log(error)
        }
    });
}
```


Client function saveName(name) calls Server function add_name()

```
server.py  hello.html  people
1  from flask import Flask
2  from flask import render_template
3  from flask import Response, request, jsonify
4  app = Flask(__name__)
5
```

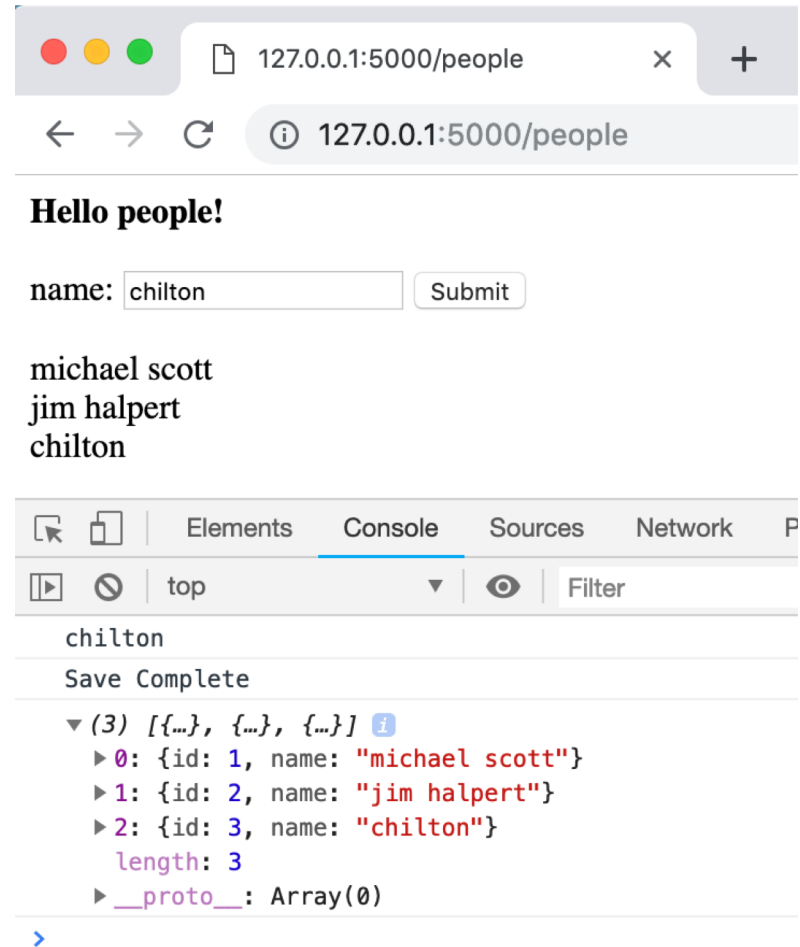
```
24
25 @app.route('/add_name', methods=['GET', 'POST'])
26 def add_name():
27     global data
28     global current_id
29
30     json_data = request.get_json()
31     name = json_data["name"]
32
33     # add new entry to array with
34     # a new id and the name the user sent in JSON
35     current_id += 1
36     new_id = current_id
37     new_name_entry = {
38         "name": name,
39         "id": current_id
40     }
41     data.append(new_name_entry)
42
43     #send back the WHOLE array of data, so the client
44     return jsonify(data = data)
45
```

Server

```
var saveName = function(name){
  var data_to_save = {"name": name}
  $.ajax({
    type: "POST",
    url: "add_name",
    dataType: "json",
    contentType: "application/json; charset=utf-8",
    data: JSON.stringify(data_to_save),
    success: function(result){
      var all_data = result["data"]
      data = all_data
      displayNames(data)
    },
    error: function(request, status, error){
      console.log("Error");
      console.log(request)
      console.log(status)
      console.log(error)
    }
  });
}
```

Client

How do we test if the data saves to the server?



The screenshot shows a web browser window with the address bar displaying `127.0.0.1:5000/people`. The page content includes the heading **Hello people!**, a form with a text input field containing `chilton` and a `Submit` button, and a list of names: `michael scott`, `jim halpert`, and `chilton`.

Below the browser window, the Chrome DevTools Console is open, showing the following log entry:

```
chilton
Save Complete
(3) [{"id": 1, "name": "michael scott"}, {"id": 2, "name": "jim halpert"}, {"id": 3, "name": "chilton"}]
  length: 3
  __proto__: Array(0)
```

Refresh the page to see if the new data stays

```
1 from flask import Flask
2 from flask import render_template
3 from flask import Response, request, jsonify
4 app = Flask(__name__)
5
6
7 # INFINITY
8 current_id = 4
9
10 sales = [
11     {
12         "id": 1,
13         "salesperson": "James D. Halpert",
14         "client": "Shake Shack",
15         "reams": 1000
16     },
17     {
18         "id": 2,
19         "salesperson": "Stanley Hudson",
20         "client": "Toast",
21         "reams": 4000
22     },
23     {
24         "id": 3,
25         "salesperson": "Michael G. Scott",
26         "client": "Computer Science Department",
27         "reams": 10000
28     },
29 ]
30
31
32 clients = [
33     "Shake Shack",
34     "Toast",
35     "Computer Science Department",
36     "Teacher's College",
37     "Starbucks",
38     "Subconscious",
39     "Flat Top",
40     "Joe's Coffee",
41     "Max Caffe",
42     "Nussbaum & Wu",
43     "Taco Bell",
44 ];
45
46
47
48
49 @app.route('/')
50 def hello_world():
51     return 'Hello World'
52
53 @app.route('/infinity')
54 def infinity():
55     return render_template('cu-paper-infinity.html', sales = sales, clients = clients)
56
```

Save Sale

```
56
57 @app.route('/save_sale', methods=['GET', 'POST'])
58 def save_sale():
59     print("save_sale")
60     global sales
61     global clients
62     global current_id
63
64     #UPDATES SALES
65     sale_data = request.get_json()
66     sale_data["id"] = current_id
67     current_id += 1
68     sales.append(sale_data)
69
70     #UPDATE CLIENTS
71     sale_client = sale_data["client"]
72     if sale_client not in clients:
73         clients.append(sale_client)
74         print("added to clients: "+sale_client)
75     else:
76         print("did NOT add client: "+ sale_client)
77
78     return jsonify(sales = sales, clients = clients)
79
```

Server

```
102 var save_sale = function(new_sale){
103     $.ajax({
104         type: "POST",
105         url: "save_sale",
106         dataType : "json",
107         contentType: "application/json; charset=utf-8",
108         data : JSON.stringify(new_sale),
109         success: function(data, text){
110             //update the sales DB and re-display it
111             var sales = data["sales"]
112             display_sales_list(sales)
113
114             //update the clients
115             clients = data["clients"]
116             $("#enter_client").autocomplete({
117                 source: clients
118             });
119
120             // reset the text so users can type there
121             $("#enter_client").val("")
122             $("#enter_reams").val("")
123             $("#enter_client").focus()
124         },
125         error: function(request, status, error){
126             console.log("Error");
127             console.log(request)
128             console.log(status)
129             console.log(error)
130         }
131     });
132 }
133
```

Client

Delete Sale

```
80
81 @app.route('/delete_sale', methods=['GET', 'POST'])
82 def delete_sale():
83     global sales
84     global client
85
86     id_json = request.get_json()
87     delete_id = int(id_json["id"])
88
89     # find the sales record with this id, and delete it.
90     index_to_delete = None
91     for (i, s) in enumerate(sales):
92         s_id = s["id"]
93         if s_id == delete_id:
94             index_to_delete = i
95
96     if index_to_delete is not None:
97         del sales[index_to_delete]
98
99     return jsonify(sales = sales)
100
```

Server

```
77
78 var delete_sale = function(id){
79     $.ajax({
80         type: "POST",
81         url: "delete_sale",
82         dataType : "json",
83         contentType: "application/json; charset=utf-8",
84         data : JSON.stringify({"id": id}),
85         success: function(data, text){
86             var sales = data["sales"]
87             display_sales_list(sales)
88         },
89         error: function(request, status, error){
90             console.log("Error");
91             console.log(request)
92             console.log(status)
93             console.log(error)
94         }
95     });
96 }
```

Client

Why do you need an id to delete? Can't you delete data using the text?

Salesperson	client	Reams
Dwight	Toast	100
Dwight	Toast	100

User A

```
Delete_sale({"salesperson":"Dwight","client":"Toast", reams: 100})
```

User B

```
Delete_sale({"salesperson":"Dwight","client":"Toast", reams: 100})
```

Will this delete one row or two????

WE DON'T KNOW!!!!

Server

Client

This pattern should be so clear that it's boring

```
57 @app.route('/save_sale', methods=['GET', 'POST'])
58 def save_sale():
59     print("save_sale")
60     global sales
61     global clients
62     global current_id
63
64     #UPDATES SALES
65     '''
78     return jsonify(sales = sales, clients = clients)
79
```

```
102 var save_sale = function(new_sale){
103     $.ajax({
104         type: "POST",
105         url: "save_sale",
106         dataType : "json",
107         contentType: "application/json; charset=utf-8",
108         data : JSON.stringify(new_sale),
109         success: function(data, text){
110             //update the sales DB and re-display it
111             var sales = data["sales"]
112             display_sales_list(sales)
113
```

```
81 @app.route('/delete_sale', methods=['GET', 'POST'])
82 def delete_sale():
83     global sales
84     global client
85
86     id_json = request.get_json()
87     delete_id = int(id_json["id"])
88
89     # find the sales record with this id, and delete it.
90
98
99     return jsonify(sales = sales)
100
```

```
77
78 var delete_sale = function(id){
79     $.ajax({
80         type: "POST",
81         url: "delete_sale",
82         dataType : "json",
83         contentType: "application/json; charset=utf-8",
84         data : JSON.stringify({"id": id}),
85         success: function(data, text){
86             var sales = data["sales"]
87             display_sales_list(sales)
88
```