

4 Rest Services are created and deployed in BlueMix

<http://mongoservice.mybluemix.net/rest/login/>

<http://mongoservice.mybluemix.net/rest/update/>

<http://mongoservice.mybluemix.net/rest/delete/>

<http://mongoservice.mybluemix.net/rest/signup/>

## SignUP

---

gowtham
talasila
gk
...

Register

```

.service('RegisterService', function($q, $http) {
    return {
        RegisterUser: function(fname, lname, username, password) {
            var deferred = $q.defer();
            var promise = deferred.promise;
            $http({
                method: 'POST',
                url: 'http://mongoservice.mybluemix.net/rest/signup',
                data: JSON.stringify({
                    firstname: fname,
                    lastname: lname,
                    username: username,
                    password: password,
                }),
                contentType: "application/json"
            }).success(function(data){

                deferred.resolve('Welcome!');
            })
            promise.success = function(fn) {
                promise.then(fn);
                return promise;
            }
            promise.error = function(fn) {
                promise.then(null, fn);
                return promise;
            }
            return promise;
        }
    }
});

```

```

import javax.ws.rs.Consumes;

```

```

@Path("/signup")

```

```

public class SignUpService {

```

```

    @POST

```

```

    @Produces(MediaType.APPLICATION_JSON)

```

```

    @Consumes(MediaType.APPLICATION_JSON)

```

```

    public Response getMsg(JSONObject obj) {

```

```

        signupMongo signup = new signupMongo();

```

```

        signup.insertdb(obj.toString());

```

```

        return Response.status(200).entity(obj.toString()).build();

```

```

    }

```

```

}

```

```
DBObject dbObject = (DBObject)JSON.parse(input);
MongoClientURI uri = new MongoClientURI("mongodb://root:password@ds03782-
MongoClient client = new MongoClient(uri);
DB db = client.getDB(uri.getDatabase());
DBCollection Users = db.getCollection("userdata");
Users.insert(dbObject);
client.close();
return true;
```

# Update

The image shows a mobile application interface with a grey background. In the center, there is a white dialog box with a light blue header and a blue 'OK' button. The dialog box contains the text 'Success!' and 'Your account is updated succesfully!'. Behind the dialog box, a form is visible with four input fields. The first field contains 'gowtham Krihna', the second 'talasila', and the third 'gk'. The fourth field contains three dots '...'. Below the form, there are two blue buttons, one on each side of the dialog box.

gowtham Krihna
talasila
gk
...

Success!

Your account is updated succesfully!

OK

gowtham
talasila
gk
...

Update

delete

```

.service('UpdateService', function($q, $http) {
    return {

        updateUser:function(data) {
            var deferred = $q.defer();
            var promise = deferred.promise;
            $http({
                method: 'POST' ,
                url: 'http://mongoservice.mybluemix.net/rest/update/'+data.username,
                data: JSON.stringify({ "$set" : { "firstname": data.firstname,
"lastname": data.lastname,
"username": data.username,
"password": data.password } })),

                contentType: "application/json"
            }).success(function (data) {
            }).error(function(data){
                alert(data);
            })
            deferred.resolve('Welcome !');

        }

        promise.success = function(fn) {
            promise.then(fn);
            return promise;
        }
        promise.error = function(fn) {
            promise.then(null, fn);
            return promise;
        }
        return promise;
    }
})

```

```

public class UpdateService {

```

```

    @POST

```

```

    @Path("/{username}")

```

```

    @Produces(MediaType.APPLICATION_JSON)

```

```

    @Consumes(MediaType.APPLICATION_JSON)

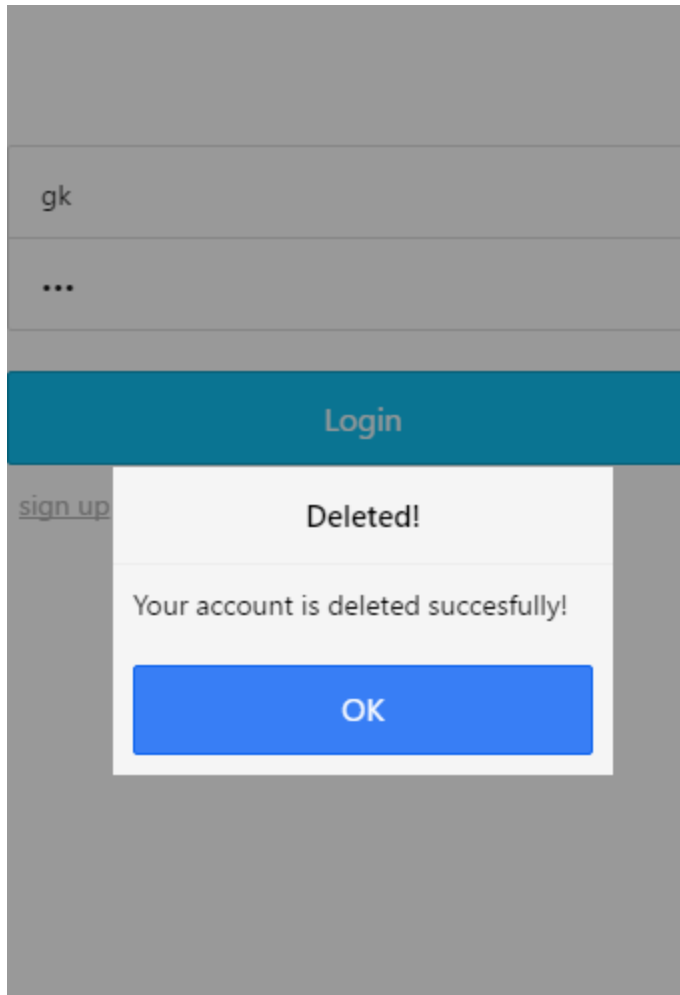
```

```

    public Response getMsg(JSONObject obj,@PathParam("username") String username) {
        updateMongo update = new updateMongo();
        update.updatedb(obj.toString(), username);
        return Response.status(200).entity("").build();
    }

```

## Delete



```

service('DeleteService', function($q, $http) {
  return {

    deleteUser:function(id) {
      var deferred = $q.defer();
      var promise = deferred.promise;

      $http({
        method: 'DELETE' ,
        url: 'http://mongoservice.mybluemix.net/rest/delete/'+id,

      }).success(function (data) {
      })
      deferred.resolve('Welcome!');
      promise.success = function(fn) {
        promise.then(fn);
        return promise;
      }
      promise.error = function(fn) {
        promise.then(null, fn);
        return promise;
      }
      return promise;
    }

  }
}

```

Jun 21 — 142 Lines

```
package org.ase;
```

```
⊕ import javax.ws.rs.Consumes;
```

```
@Path("/delete")
```

```
public class DeleteService {
```

```
⊖ @DELETE
```

```
@Path("/{id}")
```

```
public Response getMsg(@PathParam("id") String id) {
```

```
    DeleteMongo delete = new DeleteMongo();
```

```
    delete.delete(id);
```

```
    return Response.status(200).entity("").build();
```

```
}}
```



