

Consuming REST services

You might need to consume REST services from your WO apps. This is a couple of ways of doing it.

Ted Archibald's solution that he posted in the Wonder mailing list on July 28 2011

```
// You will need Jakarta httpclient-core 4.x to get it to work!

public static Job fetchJob(String jobNumber){
    Job job = null;
    DefaultHttpClient httpClient = new DefaultHttpClient();
    try {
        HttpHost target = new HttpHost("server.com", 443, "https");
       HttpGet req = new HttpGet("/jobs/" + jobNumber + ".json?key=12345");
        HttpResponse rsp = httpClient.execute(target, req);
        HttpEntity entity = rsp.getEntity();
        String resultString = EntityUtils.toString(entity);

        ERXJSONRestParser parser = new ERXJSONRestParser();

        ERXStringRestRequest request = new ERXStringRestRequest(resultString);

        ERXRestRequestNode requestNode = parser.parseRestRequest(request, null, null);

        NSDictionary dict = (NSDictionary) requestNode.toNSCollection(null);
        job = new Job(dict);
        job.setJob(jobNumber);

    }
    catch (IOException e1) {
        e1.printStackTrace();
    }
    finally {
        httpClient.getConnectionManager().shutdown();
    }
    return job;
}
```

JBoss's RestEasy client

```

SimpleClient client = ProxyFactory.create(SimpleClient.class, "http://search.twitter.com/");
ClientResponse<TwitterSearchResult> response = client.getSearchResults("#wowodc", "recent");
if (response.getResponseStatus().getFamily().equals(Response.Status.Family.SUCCESSFUL)) {
    NSLog.out.appendln(response.getEntity());
}

package com.wowodc.rest.client;

import java.util.Set;

import javax.ws.rs.Consumes;
import javax.ws.rs.GET;
import javax.ws.rs.POST;
import javax.ws.rs.Path;
import javax.ws.rs.Produces;
import javax.ws.rs.QueryParam;

import org.jboss.resteasy.client.ClientResponse;

public interface SimpleClient {

    @GET
    @Path("search.json")
    @Produces("application/json")
    ClientResponse<TwitterSearchResult> getSearchResults(@QueryParam("q") String hashtag, @QueryParam
("result_type") String resultType);
}

```

ERRestRouteExample

You can have a look at `er.rest.example.client.ERXRestClient` in the `ERRestRouteExample` sample app from Wonder.