



Consume Dropwizard REST APIs with Jersey/HTTP Client

📅 Last Updated: January 28, 2022 📁 Dropwizard 💡 Apache HttpClient, Dropwizard, REST Client

We have already built the REST APIs using dropwizard. Now let's build REST client for consuming REST APIs across the network. Dropwizard includes both [Apache HttpClient](#) and [Jersey Client](#). Let's build out them.

Read More: [Dropwizard Hello World Application](#)

Maven Dependency

Dropwizard client module is added as separate module.

```
<properties>
  <dropwizard.version>1.0.0</dropwizard.version>
</properties>
<dependency>
  <groupId>io.dropwizard</groupId>
  <artifactId>dropwizard-client</artifactId>
  <version>${dropwizard.version}</version>
</dependency>
```

Dropwizard REST Client Configuration

Dropwizard provide easy to declare and use REST client configuration. You need to create `io.dropwizard.client.JerseyClientBuilder` instance and give it `io.dropwizard.setup.Environment` reference.

```
@Override
public void run(Configuration c, Environment e) throws Exception {
    //Here we added REST Resource
    e.jersey().register(new EmployeeRestController(e.getValidator()));

    //Now we added REST Client Resource named RESTClientController
    final Client client = new JerseyClientBuilder(e).build("DemoRESTClient");
    e.jersey().register(new RESTClientController(client));
}
```

To add HTTP Client, use similar steps as below:

```
@Override
public void run(Configuration c, Environment e) throws Exception {
    //Here we added REST Resource
    e.jersey().register(new EmployeeRestController(e.getValidator()));

    //Now we added REST Client Resource named RESTClientController
    final HttpClient client = new HttpClientBuilder(e).build("DemoRESTClient");
    e.jersey().register(new RESTClientController(client));
}
```

The default configuration for **HttpClientConfiguration** is as follows:

timeout: 500ms
connectionTimeout: 500ms
timeToLive: 1 hour
cookiesEnabled: false
maxConnections: 1024
maxConnectionsPerRoute: 1024
keepAlive: 0s

The default configuration for **JerseyClientConfiguration** is as follows:

```
minThreads: 1
maxThreads: 128
gzipEnabled: true
gzipEnabledForRequests: true

//same as HttpClientConfiguration
timeout: 500ms
connectionTimeout: 500ms
timeToLive: 1 hour
cookiesEnabled: false
maxConnections: 1024
maxConnectionsPerRoute: 1024
keepAlive: 0s
```

Dropwizard REST Client Resource

Now when you have access to `javax.ws.rs.client.Client` or `org.apache.http.client.HttpClient` inside REST client resource `RESTClientController.java`, you can library specific methods to call HTTP URIs as usual.

```
package com.howtodoinjava.rest.controller;

import java.util.ArrayList;

import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.PathParam;
import javax.ws.rs.Produces;
import javax.ws.rs.client.Client;
import javax.ws.rs.client.Invocation;
import javax.ws.rs.client.WebTarget;
import javax.ws.rs.core.MediaType;
import javax.ws.rs.core.Response;

import com.howtodoinjava.rest.representations.Employee;
```

```
@Produces(MediaType.TEXT_PLAIN)
@Path("/client/")
public class RESTClientController
{
    private Client client;

    public RESTClientController(Client client) {
        this.client = client;
    }

    @GET
    @Path("/employees/")
    public String getEmployees()
    {
        //Do not hard code in your application
        WebTarget webTarget = client.target("http://localhost:8080/employees");
        Invocation.Builder invocationBuilder = webTarget.request(MediaType.APPLICATION_JSON);
        Response response = invocationBuilder.get();
        @SuppressWarnings("rawtypes")
        ArrayList employees = response.readEntity(ArrayList.class);
        return employees.toString();
    }

    @GET
    @Path("/employees/{id}")
    public String getEmployeeById(@PathParam("id") int id)
    {
        //Do not hard code in your application
        WebTarget webTarget = client.target("http://localhost:8080/employees/"+id);
        Invocation.Builder invocationBuilder = webTarget.request(MediaType.APPLICATION_JSON);
        Response response = invocationBuilder.get();
        Employee employee = response.readEntity(Employee.class);
        return employee.toString();
    }
}
```



In above class, I have accessed REST APIs created in [dropwizard hello world tutorial](#).

After accessing the APIs, I have returned the response in plain text form as show in below image.

Drop Wizard REST Client

I have set the context path of the client resource class to **/client/** to logically separate the URIs of client and service endpoints.

Read More :

[Jersey RESTful Client Examples](#)

[Apache HttpClient GET/POST Request Examples](#)

Sourcecode Download

Drop me your questions in comments section.

Happy Learning !!

Was this post helpful?

Let us know if you liked the post. That's the only way we can improve.

Yes

No

Further Reading:

1. [Testing Authentication with Jersey REST Client](#)

Learn to use Jersey REST client authentication using `HttpAuthenticationFeature`, which can be used to access REST APIs behind authentication security....

2. [Jersey Client – Set Cookie Example](#)

In this example, we will learn to set cookies into HTTP requests invoked by Jersey client. This example makes use...

3. [Getting Started with Jersey Client](#)

Jersey 2 client API finds inspiration in the proprietary Jersey 1.x Client API. In this Jersey client example, we will...

ADVERTISEMENTS



Join 7000+ Awesome Developers

Get the latest updates from industry, awesome resources, blog updates and much more.

Email Address

Subscribe

** We do not spam !!*

ADVERTISEMENTS

Leave a Comment

Post Comment

☐ Add me to your newsletter and keep me updated whenever you publish new blog posts



ADVERTISEMENTS

AdChoices

ITERABLE

Lead with the 3 P's

mimecast

SecOps

Virtual | 2022

Connect. Automate. Optimize.

Check Out Iterable

ADVERTISEMENTS

ADVERTISEMENTS



HowToDoInJava

A blog about Java and related technologies, the best practices, algorithms, and interview questions.

Meta Links

- [About Me](#)
- [Contact Us](#)
- [Privacy policy](#)
- [Advertise](#)
- [Guest Posts](#)

Blogs

[REST API Tutorial](#)



Copyright © 2022 · Hosted on [Cloudways](#) · [Sitemap](#)