

Connectors

Introduction

A connector in the REST architecture style is a software element that manages network communication for a component, typically by implementing a network protocol (e.g. HTTP). A client connector initiates communication with a server (of any kind) by creating a request. A server connector listens for connections (from clients of any kind), transmits the request to the component that performs the request processing, creates the response and sends it to the client.

All connectors are provided as extensions of the Noelios Restlet Engine, the reference implementation of the Restlet API.

This document will describe how to add a connector to your application, how to configure it and will give you the list of available server and client connectors.

Add a connector to your application

All connectors and their dependencies are shipped with the Restlet distribution by the way of jar files. Adding a connector to your application is as simple as adding the archives of the chosen connector and its dependencies to the classpath.

You can also have a look to the [FAQ #4](#)¹ and [FAQ #5](#)² which completes this subject.

Configuration

Each connector looks for its configuration from its context. The latter provides a list of modifiable parameters, which is the right place to set up the connector's configuration. Some parameters are defined by the NRE engine and thus are shared by all clients (in the ClientHelper hierarchy) and server connectors (in the ServerHelper hierarchy), and most of them by the connector's ClientHelper or ServerHelper subclasses.

The list of all parameters are available in the javadocs. Pleaser refer to the rest of this document for references to these documentation.

Server connectors

Here are the [commons parameters](#)³ dedicated to HTTP server connectors.

Here is a sample code showing how to set such a parameter.

```
// Creating a minimal Restlet returning "Hello World"
Restlet restlet = new Restlet() {
    @Override
    public void handle(Request request, Response response) {
        response.setEntity("Hello World!", MediaType.TEXT_PLAIN);
    }
};

// Create the HTTP server and listen on port 8182
Server server = new Server(new Context(), Protocol.HTTP, 8182, restlet);
server.getContext().getParameters().add("useForwardedForHeader", "true");
server.start();
```

Client connectors

Here are the [commons parameters](#)⁴ dedicated to HTTP client connectors.

Here is a sample code showing how to set such a parameter.

```
Client client = new Client(new Context(), Protocol.HTTP);
client.getContext().getParameters().add("converter",
    "com.noelios.restlet.http.HttpClientConverter");
```

List of available connectors

Server connectors

Extension	Version	Protocols
Internal ⁵	1.1	HTTP, RIAP
Grizzly ⁶	1.8	HTTP, HTTPS
Jetty ⁷	6.1	HTTP, HTTPS, AJP
Simple ⁸	3.1	HTTP, HTTPS
Servlet ⁹	2.5	HTTP, HTTPS, AJP

Client connectors

Extension	Version	Protocols
Internal ¹⁰	1.1	HTTP, CLAP, FILE, RIAP
Apache HTTP Client ¹¹	3.1	HTTP, HTTPS
Net ¹² (JDK's HttpURLConnection)	1.5	HTTP, HTTPS
JavaMail ¹³	1.4	SMTP, SMTPS, POP, POPS
JDBC ¹⁴	3.0	JDBC

1. <http://www.restlet.org/documentation/1.1/faq#04>
2. <http://www.restlet.org/documentation/1.1/faq#05>
3. <http://www.restlet.org/documentation/1.1/nre/com/noelios/restlet/http/HttpServerHelper>
4. <http://www.restlet.org/documentation/1.1/nre/com/noelios/restlet/http/HttpClientHelper>
5. /docs_1.1/13-restlet/27-restlet/48-restlet/86-restlet.html
6. /docs_1.1/13-restlet/28-restlet/73-restlet.html
7. /docs_1.1/13-restlet/28-restlet/78-restlet.html
8. /docs_1.1/13-restlet/28-restlet/82-restlet.html
9. /docs_1.1/13-restlet/28-restlet/81-restlet.html
10. /docs_1.1/13-restlet/27-restlet/48-restlet/86-restlet.html
11. /docs_1.1/13-restlet/28-restlet/75-restlet.html
12. /docs_1.1/13-restlet/28-restlet/79-restlet.html
13. /docs_1.1/13-restlet/28-restlet/76-restlet.html
14. /docs_1.1/13-restlet/28-restlet/77-restlet.html