Just to complete the example with a full implementation of `ClientHttpRequestInterceptor` to trace request and response:

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.http.HttpRequest;
import 
org.springframework.http.client.ClientHttpRequestExecution;
import 
org.springframework.http.client.ClientHttpRequestInterceptor;
import org.springframework.http.client.ClientHttpResponse;

public class LoggingRequestInterceptor implements 
ClientHttpRequestInterceptor {

    final static Logger log = 
LoggerFactory.getLogger(LoggingRequestInterceptor.class);

    @Override
    public ClientHttpResponse intercept(HttpRequest request, 
byte[] body, ClientHttpRequestExecution execution) throws 
IOException {
        traceRequest(request, body);
        ClientHttpResponse response = execution.execute(request, 
body);
        traceResponse(response);
        return response;
    }

    private void traceRequest(HttpRequest request, byte[] body) 
throws IOException {
        log.debug("===========================request 
begin================================================");
```

```
        log.debug("URI         : {}", request.getURI());
        log.debug("Method      : {}", request.getMethod());
        log.debug("Headers     : {}", request.getHeaders() );
        log.debug("Request body: {}", new String(body, "UTF-
8"));
        log.debug("===========================request
end=================================================");
    }

    private void traceResponse(ClientHttpResponse response)
throws IOException {
        StringBuilder inputStringBuilder = new StringBuilder();
        BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(response.getBody(), "UTF-8"));
        String line = bufferedReader.readLine();
        while (line != null) {
            inputStringBuilder.append(line);
            inputStringBuilder.append('\n');
            line = bufferedReader.readLine();
        }
        log.debug("============================response
begin==========================================");
        log.debug("Status code  : {}",
response.getStatusCode());
        log.debug("Status text  : {}",
response.getStatusText());
        log.debug("Headers      : {}", response.getHeaders());
        log.debug("Response body: {}",
inputStringBuilder.toString());
        log.debug("=======================response
end=================================================");
    }

}
```

Then instantiate `RestTemplate` using a `BufferingClientHttpRequestFactory` and the `LoggingRequestInterceptor`:

```
RestTemplate restTemplate = new RestTemplate(new
BufferingClientHttpRequestFactory(new
SimpleClientHttpRequestFactory()));
List<ClientHttpRequestInterceptor> interceptors = new
ArrayList<ClientHttpRequestInterceptor>();
interceptors.add(new LoggingRequestInterceptor());
restTemplate.setInterceptors(interceptors);
```

The `BufferingClientHttpRequestFactory` is required as we want to use the response body both in the interceptor and for the initial calling code. The default implementation allows to read the response body only once.

http://stackoverflow.com/questions/7952154/spring-resttemplate-how-to-enable-full-debugging-logging-of-requests-responses