public interface Converter {

public enum convertType {

C2F, F2C };

public double calculate(double value, convertType type); }

//.....................................................................................

@Stateless

public class ConverterBean implements Converter {

@Override

public double calculate(double value, convertType type) {

if (type == convertType.C2F) { return value \* 2 + 30;

}

return (value - 32) / 1.8;

} }

//............................................................................

@WebServlet(urlPatterns = {"/ConverterServlet"}, description = "Converter Servlet", name = "ConverterServlet")

public class ConverterServlet extends HttpServlet {

@EJB

Converter converter;

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException, NamingException {

double value = Double.parseDouble(request.getParameter("value"));

Converter.convertType type = Converter.convertType.valueOf(request.getParameter("type"));

double res = converter.calculate(value, type);

response.setContentType("text/html;charset=UTF-8"); try (PrintWriter out = response.getWriter()) {

/\* TODO output your page here. You may use following sample code. \*/

out.println("<!DOCTYPE html>"); out.println("<html>"); out.println("<head>");

out.println("<title>Temperature converter</title>"); out.println("</head>");

out.println("<body>"); out.println("<h1>" + res + "</h1>"); out.println("</body>"); out.println("</html>");

} }

}

//.............................................................................

public class ConverterInterceptor {

@AroundInvoke

public Object intercept(InvocationContext context) throws Exception {

String before\_text = LocalDateTime.now() + ": SimpleInterceptor - Logging BEFORE calling method :" + context.getMethod().getName() + "\n";

String after\_text = LocalDateTime.now() + ": SimpleInterceptor - Logging AFTER calling method :" + context.getMethod().getName();

FileOutputStream fs = new FileOutputStream("myFile.txt"); Object result = null;

try {

fs.write(before\_text.getBytes());

result = context.proceed();

fs.write(after\_text.getBytes());

} catch (Exception exc) { fs.write(exc.getMessage().getBytes());

} finally { fs.close();

}

return result; }

}

//................................................................................