

dans-scala-lib servlet logging

Logging *request* and *response* data in a standardized way

Joke Pol

Richard van Heest

Why servlet logging?

- Insight in servlet calls
- Standardized
- Simple calls to add to existing servlet
- Masking if necessary

Log output (GET)

curl <http://deasy.dans.knaw.nl:20140/>

```
[2019-02-18 13:48:30,177] INFO GET  
http://deasy.dans.knaw.nl:20140/ remote=192.168.33.1;  
params=[]; headers=[User-Agent -> [curl/7.54.0],  
Accept -> [*/*], Host -> [deasy.dans.knaw.nl:20140]]
```

```
[2019-02-18 13:48:30,217] INFO GET  
http://deasy.dans.knaw.nl:20140/ returned status=200;  
authHeaders=[Date -> [Mon, 18 Feb 2019 12:48:29 GMT],  
Content-Type -> [text/plain;charset=utf-8]];  
actionHeaders=[]
```

Log output (*invalid POST call*)

```
curl -X POST
```

```
http://deasy.dans.knaw.nl:20140/create?doi
```

```
[2019-02-18 13:49:04,123] INFO POST  
http://deasy.dans.knaw.nl:20140/create  
remote=192.168.33.1; params=[doi -> []];  
headers=[User-Agent -> [curl/7.54.0], Accept -> [*/*],  
Host -> [deasy.dans.knaw.nl:20140]]
```

```
[2019-02-18 13:49:04,132] INFO POST  
http://deasy.dans.knaw.nl:20140/create returned  
status=400; authHeaders=[Date -> [Mon, 18 Feb 2019  
12:49:04 GMT]]; actionHeaders=[]
```

Log output (*POST with params*)

```
curl -X POST  
http://deasy.dans.knaw.nl:20140/create?type=doi
```

```
[2019-02-18 13:49:41,174] INFO POST  
http://deasy.dans.knaw.nl:20140/create  
remote=192.168.33.1; params=[type -> [doi]];  
headers=[User-Agent -> [curl/7.54.0], Accept -> [*/*],  
Host -> [deasy.dans.knaw.nl:20140]]
```

```
[2019-02-18 13:49:41,432] INFO POST  
http://deasy.dans.knaw.nl:20140/create returned  
status=201; authHeaders=[Date -> [Mon, 18 Feb 2019  
12:49:41 GMT]]; actionHeaders=[]
```

Log output (*masked*)

easy-deposit-api login

```
[2019-02-18 14:31:12,503] INFO POST  
http://localhost:20190/auth/login  
remote=**.**.**.1; params=[]; headers=[<...>,  
Authorization -> [Basic *****, <...>]
```

```
[2019-02-18 14:31:13,026] INFO POST  
http://localhost:20190/auth/login returned  
status=204; authHeaders=[  
Set-Cookie -> [sentry.auth.default.user=*****,  
<...>, REMOTE_USER -> [*****]]; actionHeaders=[]
```

Log output

```
[2019-02-18 13:48:30,177] INFO GET http://deasy.dans.knaw.nl:20140/ remote=192.168.33.1; params=[]; headers=[
[2019-02-18 13:48:30,217] INFO GET http://deasy.dans.knaw.nl:20140/ returned status=200; authHeaders=[Date ->
[2019-02-18 13:49:04,123] INFO POST http://deasy.dans.knaw.nl:20140/create remote=192.168.33.1; params=[doi ->
[2019-02-18 13:49:04,132] INFO POST http://deasy.dans.knaw.nl:20140/create returned status=400; authHeaders=[
[2019-02-18 13:49:41,174] INFO POST http://deasy.dans.knaw.nl:20140/create remote=192.168.33.1; params=[type -
[2019-02-18 13:49:41,429] INFO Minted a new doi: 10.5072/dans-x6f-kf66
[2019-02-18 13:49:41,432] INFO POST http://deasy.dans.knaw.nl:20140/create returned status=201; authHeaders=[
```

Example

```
import org.scalatra.{ Ok, ScalatraServlet }
```

```
class MyServlet extends ScalatraServlet {
```

```
  get("/") {  
    Ok("foobar")  
  }
```

```
}
```


Example (*plain logging*)

```
import nl.knaw.dans.lib.logging.DebugEnhancedLogging
import nl.knaw.dans.lib.logging.servlet._ // IMPORT!!!
import org.scalatra.{ Ok, ScalatraServlet }

class MyServlet extends ScalatraServlet
  with ServletLogger // add servlet logging
  with PlainLogFormatter // select plain logging
  with DebugEnhancedLogging { // select logger

  get("/") {
    Ok("foobar").logResponse // add response log
  }
}
```

Example (*masking*)

```
import nl.knaw.dans.lib.logging.DebugEnhancedLogging
import nl.knaw.dans.lib.logging.servlet._
import org.scalatra.{ Ok, ScalatraServlet }

class MyServlet extends ScalatraServlet
  with ServletLogger
  with MaskedLogFormatter // select masked logging
  with DebugEnhancedLogging {

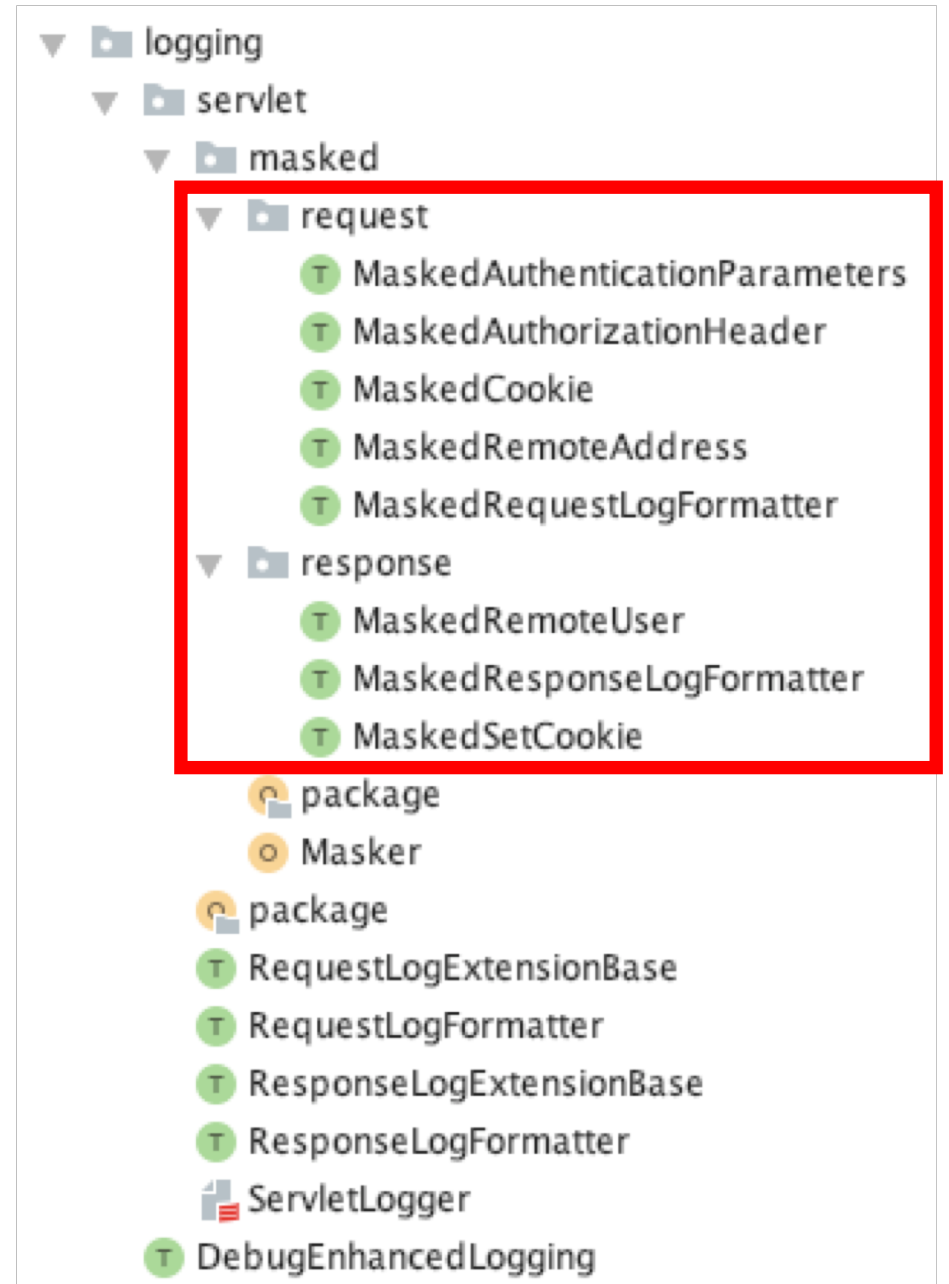
  get("/") {
    Ok("foobar").logResponse
  }
}
```

Example (*partial masking*)

```
import nl.knaw.dans.lib.logging.DebugEnhancedLogging
import nl.knaw.dans.lib.logging.servlet._
import nl.knaw.dans.lib.logging.servlet.masked._ //IMPORT
import org.scalatra.{ Ok, ScalatraServlet }

class MyServlet extends ScalatraServlet
  with ServletLogger
  with PlainLogFormatter // select plain logging
  with MaskedRemoteAddress // add masking parts
  with MaskedRemoteUser
  with DebugEnhancedLogging {
  //...
}
```

Example (*partial masking*)



Hackethon

- easy-auth-info
- easy-bag-index
- easy-bag-store
- easy-deposit-agreement-creator
- easy-deposit-api
- easy-download
- easy-pid-generator
- easy-solr4files-index
- ~~easy-sword2~~
- easy-validate-dans-bag
- narcis-pid-aggregator
- narcis-pid-manager
- narcis-pid-graph

Hackethon (tasks for each project)

1. Add dans-scala-lib **v1.5.0** to pom.xml
2. Add logging to each servlet
3. Deploy on deasy and test that the logging shows up in
/var/opt/dans.knaw.nl/log/easy-*/easy-*.log
4. Create a pull request