Логирование ошибок КНД.

Для реализации функции логирования нами была выбран **Spring Boot** :

- **Spring Boot** использует штатные возможности для проброса ошибок в тело response-ответа.
- **Spring Boot** предоставляет реализацию *ErrorController* для разумной обработки ошибок
- **Spring Boot** обслуживает резервную страницу ошибок для браузеров (страница ошибок Whitelabel) и ответ JSON для запросов RESTful, отличных от HTML:


```
{
   "timestamp": "2019-01-17T16:12:45.977+0000",
   "status": 500,
   "error": "Internal Server Error",
   "message": "Error processing the request!",
   "path": "/my-endpoint-with-exceptions"
}
```

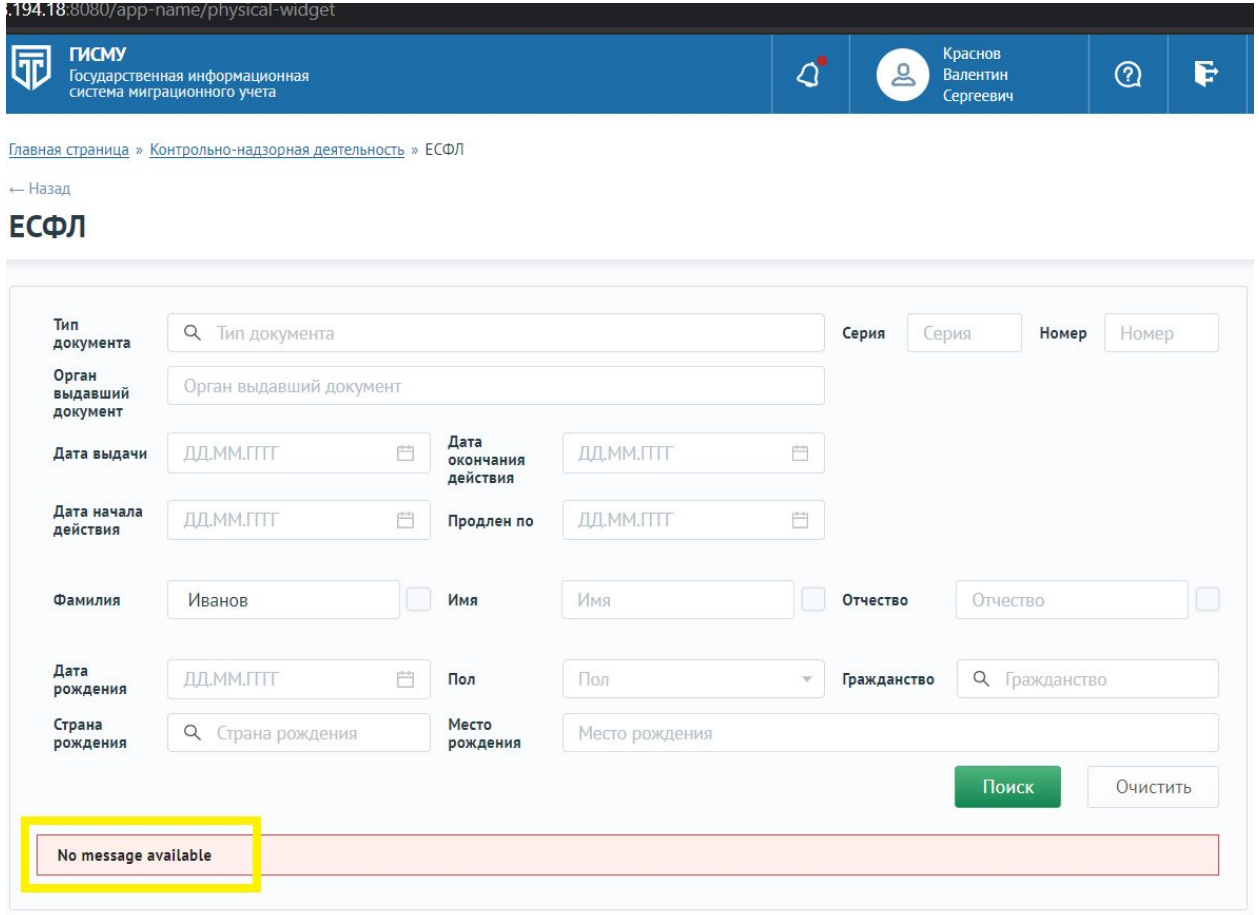
Настройка Spring Boot

Spring Boot позволяет настраивать эти функции с помощью свойств:

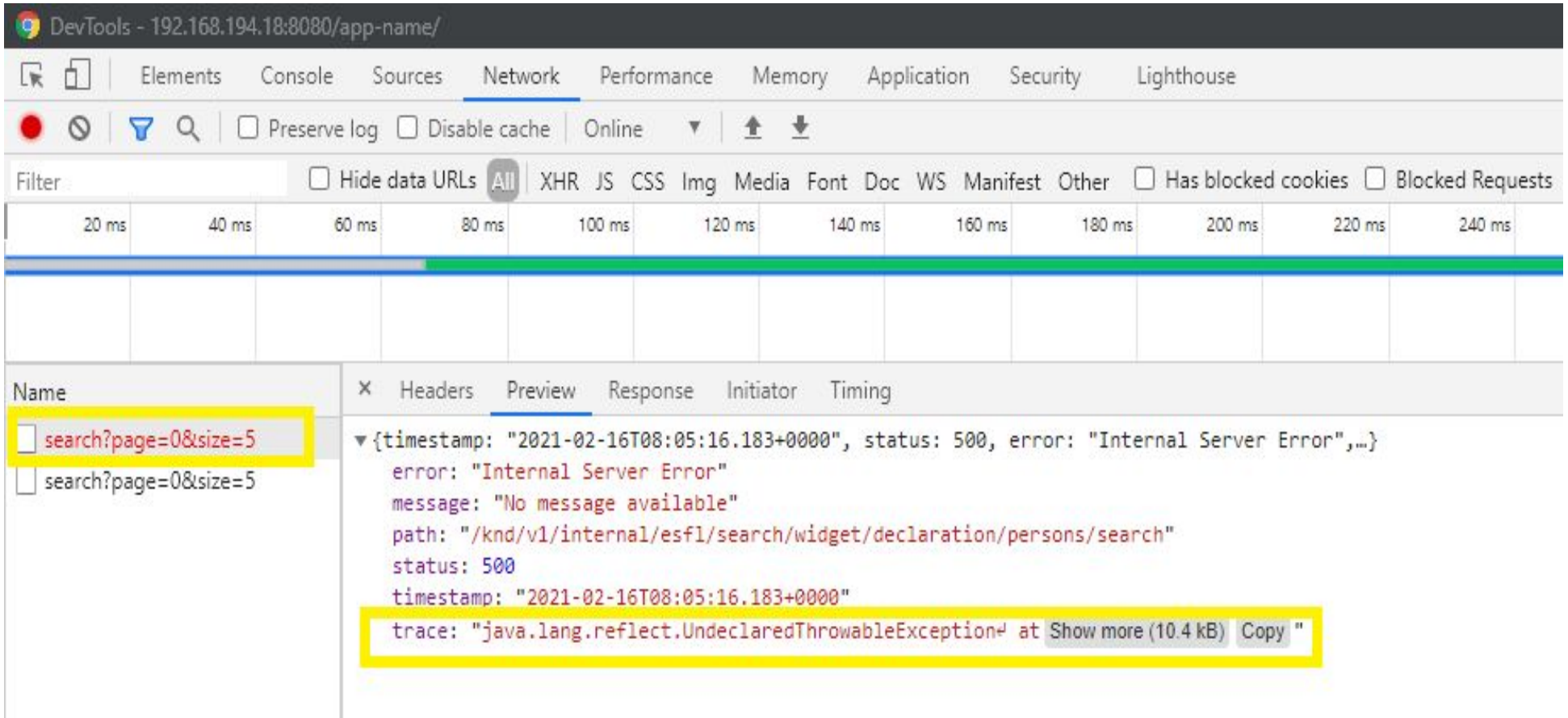
- ***server.error.whitelabel.enabled*** : можно использовать для отключения страницы ошибок Whitelabel и полагаться на контейнер сервлета для предоставления сообщения об ошибке HTML.
- ***server.error.include-stacktrace*** : с ***постоянным*** значением; включает трассировку стека как в HTML, так и в ответ по умолчанию JSON

Обработка ошибки на примере ЕСФЛ.

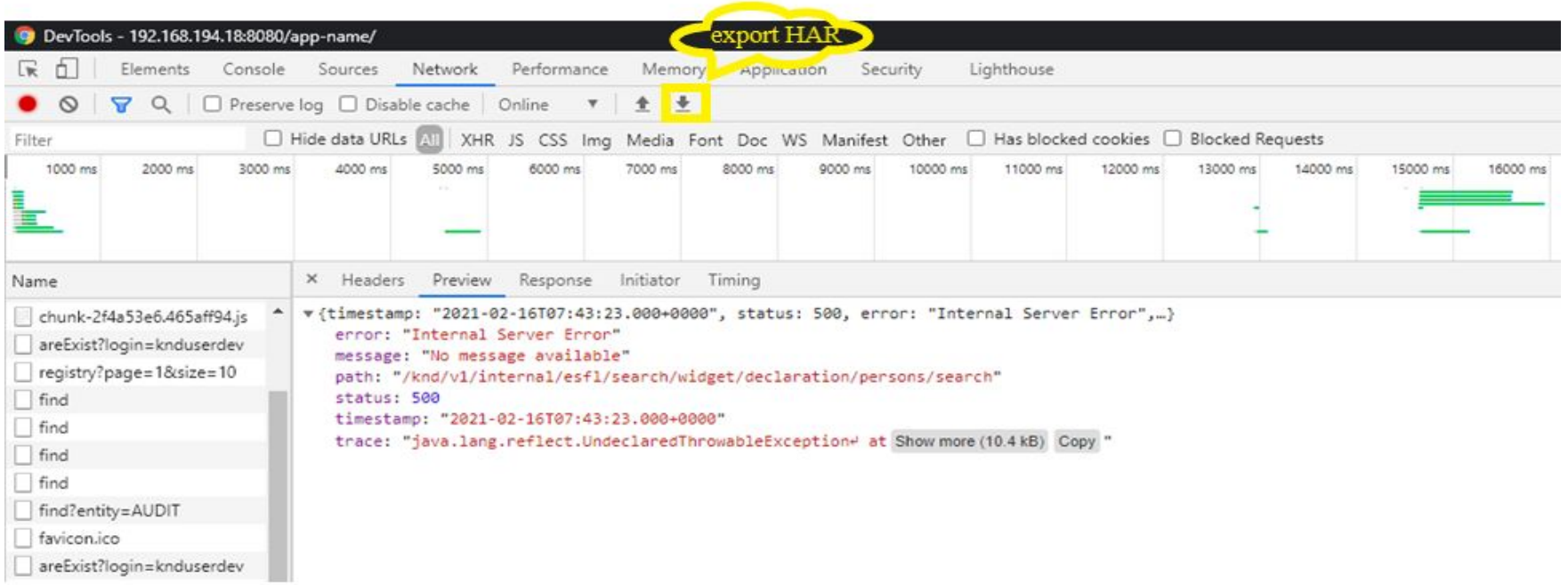
В виджете ЕСФЛ не отображаются физ.лица с фамилией Иванов.



Просматриваем ошибку в Development Tools



Средствами Development Tools экспортируем и сохраняем данные в HAR.



- На выходе имеем сценарий содержащий request, response, время и статус ошибки.
- В случае возникновения ошибки, в ***trace*** параметре response-ответа сохраняется stacktrace с ошибкой.

Lister - [c:\Sokrat\Curent\Tasks\192.168.194.18_error_log.har]

Файл Правка Вид Кодировка Справка 64 %


```
        }
      },
      "response": {
        "status": 500,
        "statusText": "",
        "httpVersion": "HTTP/1.1",
        "headers": [
          {
            "name": "Vary",
            "value": "Origin"
          }
        ],
        "cookies": [],
        "content": {
          "size": 10620,
          "mimeType": "application/json",
          "text": "{\"timestamp\":\"2021-02-16T08:05:16.183+0000\",\"status\":500,\"error\":\"Internal Server Error\",\"message\":\"No message
available\",\"trace\":\"java.lang.reflect.UndeclaredThrowableException\\n\\tat com.sun.proxy.$Proxy359.searchPersonsWidgetByRequest(Unknown Source)\\n\\tat
ru.atc.mvd.gismu.knd.integration.esfl.services.serviceImpl.EsflServiceImpl.searchPersonsWidgetByRequest(EsflServiceImpl.java:73)\\n\\tat
ru.atc.mvd.gismu.knd.controller.internal.EsflController.searchPersonsWidgetByRequest(EsflController.java:85)\\n\\tat sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)\\n\\tat
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)\\n\\tat sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)\\n\\tat
java.lang.reflect.Method.invoke(Method.java:498)\\n\\tat org.springframework.web.method.support.InvocableHandlerMethod.doInvoke(InvocableHandlerMethod.java:190)\\n\\tat
org.springframework.web.method.support.InvocableHandlerMethod.invokeForRequest(InvocableHandlerMethod.java:138)\\n\\tat
org.springframework.web.servlet.mvc.method.annotation.ServletInvocableHandlerMethod.invokeAndHandle(ServletInvocableHandlerMethod.java:105)\\n\\tat
org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.invokeHandlerMethod(RequestMappingHandlerAdapter.java:879)\\n\\tat
org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.handleInternal(RequestMappingHandlerAdapter.java:793)\\n\\tat
org.springframework.web.servlet.mvc.method.AbstractHandlerMethodAdapter.handle(AbstractHandlerMethodAdapter.java:87)\\n\\tat
org.springframework.web.servlet.DispatcherServlet.doDispatch(DispatcherServlet.java:1040)\\n\\tat
org.springframework.web.servlet.DispatcherServlet.doService(DispatcherServlet.java:943)\\n\\tat
org.springframework.web.servlet.FrameworkServlet.processRequest(FrameworkServlet.java:1006)\\n\\tat org.springframework.web.servlet.FrameworkServlet.doPost(FrameworkServlet.java:909)\\n\\ta
javax.servlet.http.HttpServlet.service(HttpServlet.java:665)\\n\\tat org.springframework.web.servlet.FrameworkServlet.service(FrameworkServlet.java:883)\\n\\tat
javax.servlet.http.HttpServlet.service(HttpServlet.java:750)\\n\\tat org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:231)\\n\\tat
org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)\\n\\tat org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:53)\\n\\tat
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:193)\\n\\tat
org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)\\n\\tat
org.springframework.security.web.FilterChainProxy.doFilterInternal(FilterChainProxy.java:209)\\n\\tat
org.springframework.security.web.FilterChainProxy.doFilter(FilterChainProxy.java:186)\\n\\tat org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:19
)\\n\\tat org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)\\n\\tat
org.springframework.web.filter.RequestContextFilter.doFilterInternal(RequestContextFilter.java:100)\\n\\tat
org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:119)\\n\\tat
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:193)\\n\\tat
org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)\\n\\tat
org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:320)\\n\\tat
org.springframework.security.web.access.intercept.FilterSecurityInterceptor.invoke(FilterSecurityInterceptor.java:126)\\n\\tat
```