

# 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.

194.18.8080/app-name/physical-widget

**ГИСМУ**  
Государственная информационная  
система миграционного учета

Краснов  
Валентин  
Сергеевич

[Главная страница](#) » [Контрольно-надзорная деятельность](#) » ЕСФЛ

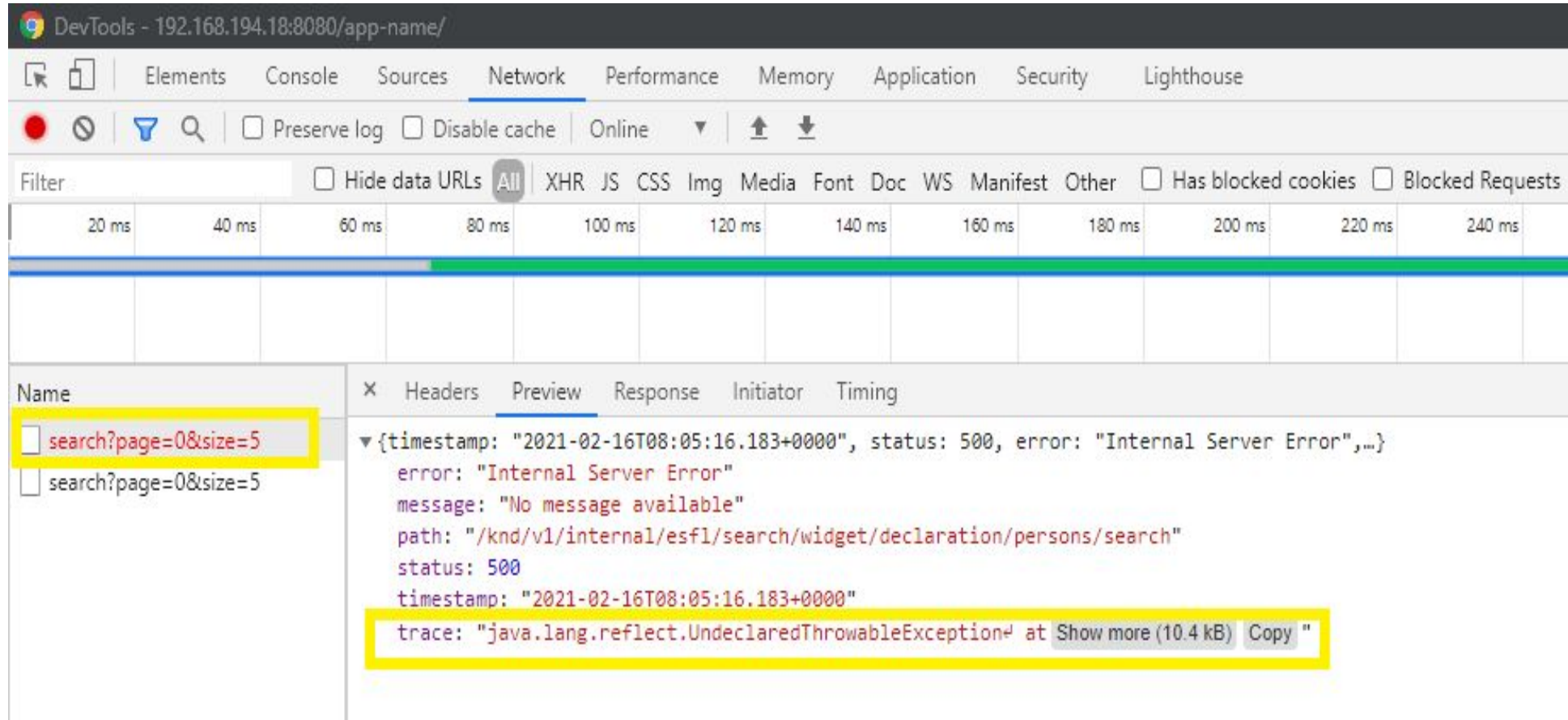
[← Назад](#)

## ЕСФЛ

Тип документа	<input type="text" value="Тип документа"/>	Серия	<input type="text" value="Серия"/>	Номер	<input type="text" value="Номер"/>
Орган выдавший документ	<input type="text" value="Орган выдавший документ"/>				
Дата выдачи	<input type="text" value="ДД.ММ.ГГГГ"/>	Дата окончания действия	<input type="text" value="ДД.ММ.ГГГГ"/>		
Дата начала действия	<input type="text" value="ДД.ММ.ГГГГ"/>	Продлен по	<input type="text" value="ДД.ММ.ГГГГ"/>		
Фамилия	<input type="text" value="Иванов"/>	Имя	<input type="text" value="Имя"/>	Отчество	<input type="text" value="Отчество"/>
Дата рождения	<input type="text" value="ДД.ММ.ГГГГ"/>	Пол	<input type="text" value="Пол"/>	Гражданство	<input type="text" value="Гражданство"/>
Страна рождения	<input type="text" value="Страна рождения"/>	Место рождения	<input type="text" value="Место рождения"/>		

No message available

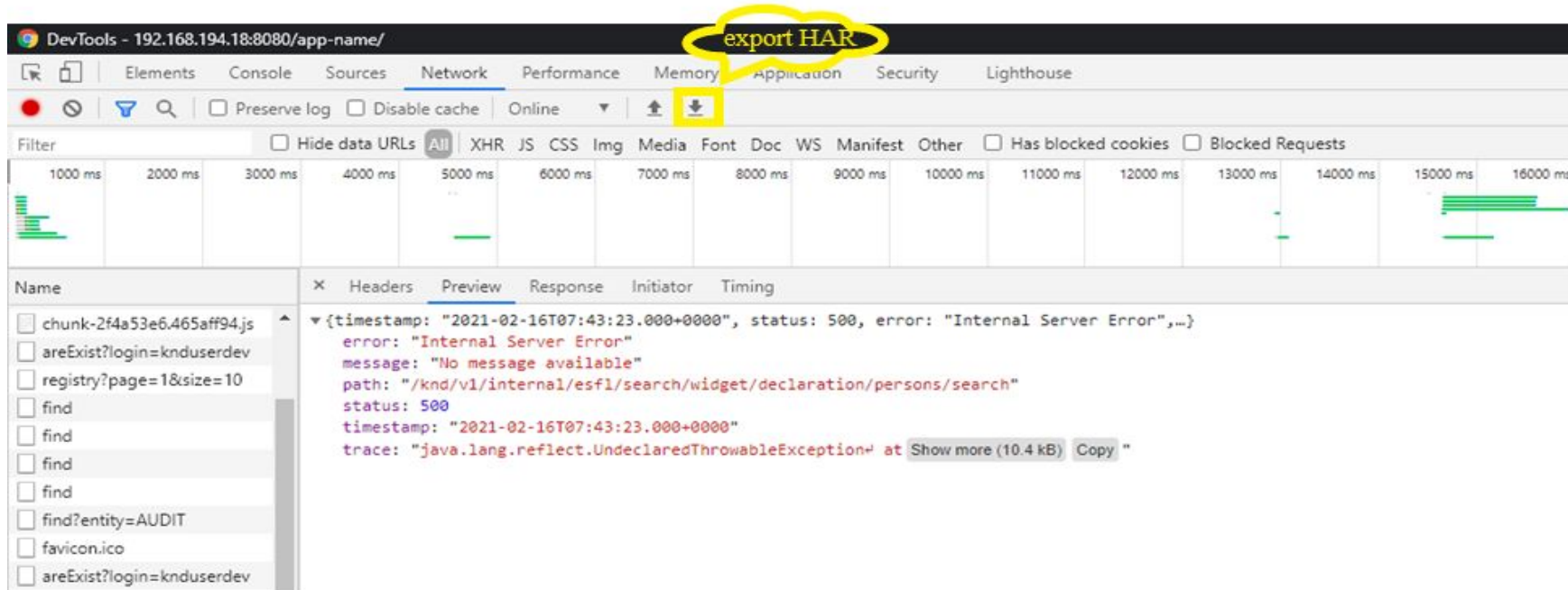
## Просматриваем ошибку в Development Tools



The screenshot shows the Chrome DevTools Network tab. The top navigation bar includes Elements, Console, Sources, Network (selected), Performance, Memory, Application, Security, and Lighthouse. Below the navigation bar, there are controls for Preserve log, Disable cache, and Online status. A filter bar is set to 'All' and includes checkboxes for 'Hide data URLs', 'Has blocked cookies', and 'Blocked Requests'. A timeline at the top shows a request starting at approximately 80ms. The main area displays a list of requests, with the first one, 'search?page=0&size=5', highlighted in yellow. The details panel for this request is open, showing the 'Preview' tab. The response is a JSON object with the following structure:

```
{timestamp: "2021-02-16T08:05:16.183+0000", status: 500, error: "Internal Server Error",...}
  error: "Internal Server Error"
  message: "No message available"
  path: "/knd/v1/internal/esf1/search/widget/declaration/persons/search"
  status: 500
  timestamp: "2021-02-16T08:05:16.183+0000"
  trace: "java.lang.reflect.UndeclaredThrowableException" at Show more (10.4 kB) Copy "
```

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": "Uary",
        "value": "Origin"
      }
    ],
    "cookies": [],
    "content": {
      "size": 10620,
      "mimeType": "application/json",
      "text": "{ \"timestamp\": \"2021-02-16T08:05:16.133+0000\", \"status\": 500, \"error\": \"Internal Server Error\", \"message\": \"No message available\", \"trace\": \" java.lang.reflect.UndeclaredThrowableException\\n\\t at com.sun.proxy.$Proxy359.searchPersonsWidgetByRequest(Unknown Source)\\n\\t at ru.atc.mud.gismu.knd.integration.esfl.services.serviceImpl.EsflServiceImpl.searchPersonsWidgetByRequest(EsflServiceImpl.java:73)\\n\\t at ru.atc.mud.gismu.knd.controller.internal.EsflController.searchPersonsWidgetByRequest(EsflController.java:85)\\n\\t at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)\\n\\t at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)\\n\\t at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)\\n\\t at java.lang.reflect.Method.invoke(Method.java:498)\\n\\t at org.springframework.web.method.support.InvocableHandlerMethod.doInvoke(InvocableHandlerMethod.java:190)\\n\\t at org.springframework.web.method.support.InvocableHandlerMethod.invokeForRequest(InvocableHandlerMethod.java:138)\\n\\t at org.springframework.web.servlet.mvc.method.annotation.ServletInvocableHandlerMethod.invokeAndHandle(ServletInvocableHandlerMethod.java:105)\\n\\t at org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.invokeHandlerMethod(RequestMappingHandlerAdapter.java:879)\\n\\t at org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.handleInternal(RequestMappingHandlerAdapter.java:793)\\n\\t at org.springframework.web.servlet.mvc.method.AbstractHandlerMethodAdapter.handle(AbstractHandlerMethodAdapter.java:87)\\n\\t at org.springframework.web.servlet.DispatcherServlet.doDispatch(DispatcherServlet.java:1040)\\n\\t at org.springframework.web.servlet.DispatcherServlet.doService(DispatcherServlet.java:943)\\n\\t at org.springframework.web.servlet.FrameworkServlet.processRequest(FrameworkServlet.java:1006)\\n\\t at org.springframework.web.servlet.FrameworkServlet.doPost(FrameworkServlet.java:909)\\n\\t at javax.servlet.http.HttpServlet.service(HttpServlet.java:665)\\n\\t at org.springframework.web.servlet.FrameworkServlet.service(FrameworkServlet.java:883)\\n\\t at javax.servlet.http.HttpServlet.service(HttpServlet.java:750)\\n\\t at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:231)\\n\\t at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)\\n\\t at org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:53)\\n\\t at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:193)\\n\\t at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)\\n\\t at org.springframework.web.filter.RequestContextFilter.doFilterInternal(RequestContextFilter.java:100)\\n\\t at org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:119)\\n\\t at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:193)\\n\\t at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)\\n\\t at org.springframework.web.filter.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:320)\\n\\t at org.springframework.security.web.access.intercept.FilterSecurityInterceptor.invoke(FilterSecurityInterceptor.java:126)\\n\\t at

```