

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 interface with the Network tab selected. The address bar displays "DevTools - 192.168.194.18:8080/app-name/". The Network tab toolbar includes options for "Preserve log", "Disable cache", and "Online". The filter is set to "All". A timeline at the top shows a request starting at approximately 80ms. The list of requests below the timeline shows two entries for "search?page=0&size=5". The first entry is selected, and its details are shown in the right pane. The details pane has tabs for "Headers", "Preview", "Response", "Initiator", and "Timing". The "Response" tab is active, displaying a JSON error object. The error message is "Internal Server Error" with a status of 500. The path is "/knd/v1/internal/esf1/search/widget/declaration/persons/search". The trace shows a "java.lang.reflect.UndeclaredThrowableException" at the end of the response.

DevTools - 192.168.194.18:8080/app-name/

Elements Console Sources **Network** Performance Memory Application Security Lighthouse

⏏ ⏏ | Preserve log Disable cache | Online ▼ | ⬆ ⬇

Filter: Hide data URLs **All** XHR JS CSS Img Media Font Doc WS Manifest Other Has blocked cookies Blocked Requests

20 ms 40 ms 60 ms 80 ms 100 ms 120 ms 140 ms 160 ms 180 ms 200 ms 220 ms 240 ms

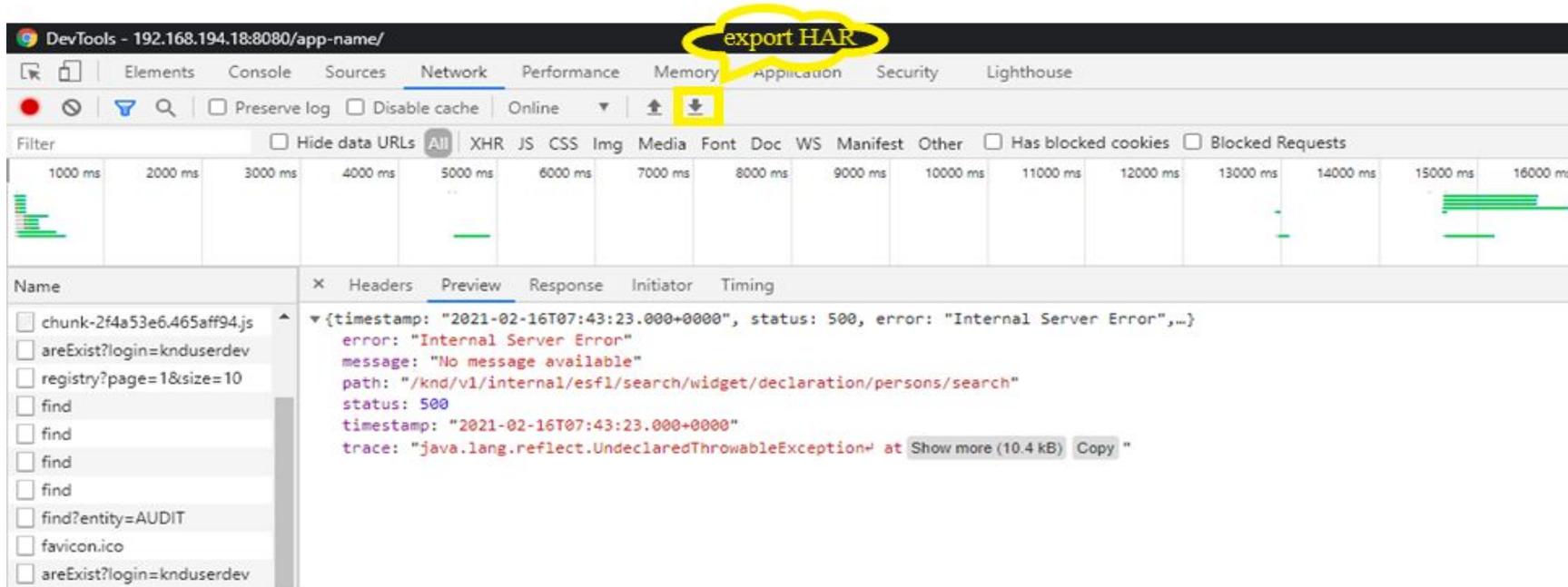
Name

- search?page=0&size=5
- search?page=0&size=5

× Headers **Preview** Response Initiator Timing

```
{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

```

}
"response": {
  "status": 500,
  "statusText": "",
  "httpVersion": "HTTP/1.1",
  "headers": [
    {
      "name": "Uary",
      "value": "Origin"
    }
  ]
},

```

```

"trace": {
  "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\\tat com.sun.proxy.$Proxy359.searchPersonsWidgetByRequest(Unknown Source)\\n\\tat ru.atc.mud.gismu.knd.integration.esfl.services.serviceImpl.EsflServiceImpl.searchPersonsWidgetByRequest(EsflServiceImpl.java:73)\\n\\tat ru.atc.mud.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\\tat 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:193)\\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

```