|  |
| --- |
| ㅡ  Mysql 에서 문자열 앞에 공백이 들어가면 다르다고 생각하지만  뒤쪽에 공백이 들어가면 문자열 길이는 정상적으로 공백까지 개수를 처리해주지만  where조건에 a=’abc’ 나 a=’abc ‘ 라고 해도 나온다  get방식으로 url호출시 파라미터 3000byte이상 넘어갈때가 있었다  그때 상대편 web서버에서 파라미터를 못받아서 식겁했는데.. 보통 get방식은url 길이 제한이 있는걸 알고 있었기에.. 식겁했다  하지만 확인해본결과 get방식이라도 url 길이제한은 없으며 브라우저별로 길이 제한을 걸어놓은거라고 한다  주 : 서버가 255 바이트 이상 길이 URI에 따라에 대해 신중해야한다고, 일부 오래된 클라이언트 또는 프록시 구현이 제대로이 길이를 지원하지 않을 수 있기 때문에  <http://tibang.tistory.com/m/465>  get방식의 글자수 제한은 256자가 아니다  의문점 일반적으로 웹페이지에서 get방식으로 데이터를 넘길경우 256바이트 이상을 넘길수 없다고 합니다. 상당수의 책에서 이렇게 되있더군요. URL문자열이 256글자를 넘길수 없다는 이유때문이라고 하는데 개인적으로 운영하는 웹페이지에서는 좀 색다른 현상이 나옵니다. [Http://아무개.com/bbb/고정보관/군사정보/2차대전/육상병기/051218](http://xn--h49a69z1rh.com/bbb/%EA%B3%A0%EC%A0%95%EB%B3%B4%EA%B4%80/%EA%B5%B0%EC%82%AC%EC%A0%95%EB%B3%B4/2%EC%B0%A8%EB%8C%80%EC%A0%84/%EC%9C%A1%EC%83%81%EB%B3%91%EA%B8%B0/051218)중구축전차 페르디난트\_엘레판트/  이런경로를 UTF-8엔코딩을 하면 아래처럼 변합니다.  [Http:// 아무개.com/bbb/%ea%b3%a0%ec%a0%95%eb%b3%b4%ea%b4%80/%ea%b5%b0%ec%82%ac%ec %a0%95%eb%b3%b4/2%ec%b0%a8%eb%8c%80%ec%a0%84/%ec%9c%a1%ec%83%81%eb%b3%91%ea%b8%b0/051218%ec%a4%91%ea%b5%ac%ec%b6%95%ec%a0%84%ec%b0%a8%20%ed%8e%98%eb%a5%b4%eb%94%94%eb%82%9c%ed%8a%b8\_%ec%97%98%eb%a0%88%ed%8c%90%ed%8a%b8/](http://xn--h49a69z1rh.com/bbb/%ea%b3%a0%ec%a0%95%eb%b3%b4%ea%b4%80/%ea%b5%b0%ec%82%ac%ec%a0%95%eb%b3%b4/2%ec%b0%a8%eb%8c%80%ec%a0%84/%ec%9c%a1%ec%83%81%eb%b3%91%ea%b8%b0/051218%ec%a4%91%ea%b5%ac%ec%b6%95%ec%a0%84%ec%b0%a8%20%ed%8e%98%eb%a5%b4%eb%94%94%eb%82%9c%ed%8a%b8_%ec%97%98%eb%a0%88%ed%8c%90%ed%8a%b8/)  300줄의 URL입니다. 원래대로라면 접근이 안되야 정상입니다만 접근이 됩니다. 웹에서 검색해보니 URL경로는 2048글자도 가능하다고 합니다. 그러면 GET방식도 되어야 하지 않을까요?  테스트 테스트를 해봅니다. 간단하게 GET방식의 데이터를 화면에 뿌려봅니다.  텍스트 에디터에서 결과를 출력해보면  http://pds7.egloos.com/pds/200804/29/35/c0073935_48168c1a9439e.jpg  위의 3줄은 URL의 끝부분입니다.(URL경로 포함) 1번째가 주소입력창에 넣을 원본경로 입니다.(4420글자) 2번째는 익스플로러에서 출력된 결과입니다. (2047글자 출력) 3번째는 파이어폭스에서 출력된 결과입니다.(4420글자 출력)  테스트 결과는 전혀 엉뚱합니다. 256자의 제한은 없고 일부 브라우져는 글자제한없는 결과를 보여줍니다.  근거  근거를 찾기위해 검색을 해보니 이런 결과가 나옵니다. <http://support.microsoft.com/default.aspx?scid=KB;en-us;q208427>  대충 요약해보면 익스플로러 4.0부터 7.0까지는 URL경로를 2048바이트로 제한했다고 합니다. 표준 안지키는 익스플로러 답다고나 해야 할까요? 아닙니다. 뭔가 다른 이유가 있을겁니다.  <http://www.w3.org/Protocols/rfc2616/rfc2616-sec3.html#sec3.2.1>  <http://www.tta.or.kr/Home2003/library/ttasDown.jsp?where=13292&pk_num=TTAS.IF-RFC2068> (한글번역) HTTP 1.1규약입니다. 해석해보면 이런 귀절이 있습니다.  HTTP 규약은 URI의 길이에 대한 어떠한 사전 제한도 두지 않는다. 서버는 반드시 자신이 제 공하는 어떠한 자원의 URI도 처리할 수 있어야 하며 이러한 URI를 생성할 수 있는 GET에  기초한 폼을 (GET-based forms) 제공한다면 무제한 길이의 URI를 처리할 수 있어야만 한다. 서 버는 URI의 길이가 자신의 처리할 수 있는 (10.4.15 절 참조) 것보다 긴 경우 414 (Request-URI  Too Long)를 응답으로서 돌려주어야 한다.  주의: 서버는 255 바이트 이상의 URI 길이를 사용할 때 몇몇 이전 클라이언트나 프락시 구 현 방식이 이러한 길이를 적절히 지원할 수 없는 경우가 있기 때문에 주의해야 한다.  URI : Uniform Resource Identifier 주 : URL이 아니라 URI라고 표현하는데 URL은 URI의 하위 개념입니다. 다만 대부분의 웹페이지가 URL로 접근되기 때문에 URL로 불려지는 경우가 많습니다.  여기서 결론이 나옵니다. 원칙적으로 제한이 없는데 지난세기(?)의 구형 장비들중 일부가 256바이트의 주소값을 지원하기 때문에 주의하라는 겁니다. 익스플로러는 이 규정때문에 256자 이상을 임의로 확장해놓고 이후 게을러 빠져서 수정을 안하는 셈이겠지요. 익스플로러 4.0이 나올무렵에 HTTP 1.0표준이 널리 퍼지지못했기 때문이기도 할겁니다. 파이어폭스는 21세기(?)브라우져 답게 표준에 나오는 대로 무제한의 URL을 지원합니다. 한마디로 **브라우져 마음대로** 군요  결론 사실상 256자의 제한은 없습니다. 256자 제한이 있는 구형장비가 대부분 도태됬기 때문입니다. 여기에 세계적인 인터넷 사용의 일반화로 UTF-8로 인코딩된 주소가 일반화되 256자로는 주소표기가 불가능해진것도 원인입니다. 다만 익스플로러의 2048바이트 제한은 여전합니다. 8.0에서는 개선됬을지 모르겠군요 주소제한이 없는 파이어폭스를 쓰셔도 될겁니다. 한가지 덧붙인다면 일반적인 지식검색이나 웹상의 내용이 근거를 밝히지 않고 적어두다 보니 엉터리내용이 많다는 겁니다. 덕분에 오전에 삽질해 가며 데이터를 찾아 헤멨습니다만 명확한 결론을 얻게된건 좋은일이지 않을까요? 답글로 성원해 주신다면 감사하겠습니다. 좋은 하루 되세요^^  원문 : http://brainage.egloos.com/4323903 |
| **삽질기행 1. spring security + hibernate getCurrentSession ...ㅎㄷㄷ**  삽질 기행 1  배경 :  1. spring security 이용              2. security에서 DB조회             3. hibernate 사용 이슈 : spring security 부분에서 hibernate  getCurrentSession 안먹.. **문제 원인 :  tomcat  web.xml 쪽  filter 우선순의 에따른 hibernate CurrentSession 처리 못함**  **기존 web.xml**    <!-- 스프링 시큐어리티 -->  <filter>  <filter-name>springSecurityFilterChain</filter-name>  <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>  </filter>  <filter-mapping>  <filter-name>springSecurityFilterChain</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping>    <!-- 하이버네이트opensession getcurrentSession을위해.. -->  <filter>  <filter-name>OpenSessionInViewFilter</filter-name>  <filter-class>org.springframework.orm.hibernate4.support.OpenSessionInViewFilter</filter-class>  <init-param>  <param-name>sessionFactoryBeanName</param-name>  <param-value>sessionFactory</param-value>  </init-param>  </filter>  <filter-mapping>  <filter-name>OpenSessionInViewFilter</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping>        **문제 해결 :  web.xml쪽에 filter 우선선위 변경후 성공**  <!-- 하이버네이트opensession getcurrentSession을위해.. -->  <filter>  <filter-name>OpenSessionInViewFilter</filter-name>  <filter-class>org.springframework.orm.hibernate4.support.OpenSessionInViewFilter</filter-class>  <init-param>  <param-name>sessionFactoryBeanName</param-name>  <param-value>sessionFactory</param-value>  </init-param>  </filter>  <filter-mapping>  <filter-name>OpenSessionInViewFilter</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping>        <!-- 스프링 시큐어리티-->  <filter>  <filter-name>springSecurityFilterChain</filter-name>  <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>  </filter>  <filter-mapping>  <filter-name>springSecurityFilterChain</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping>    삽질시간 : 1시간 30분 소요  팁 : filter는  servlet 처럼 우선순위를 지정하는 option이 없습니다.  명시한 순서대로 처리됩니다.  (필터의 아무도는 순서를 인식하지 않습니다. 순서는 필터의 web.xml 구성되는 순서에 따라 필터 체인을 통해 완전히 처리된다.) |
| 삽질..잡담.. 윈도우에서 개발하다  아마존 클라우드(리눅스)로 배포했는데..... 두둥.  MYSQL DB TABLE 접근실패!!!! "mysql table doesn't exist" 권한줘보고 빼보고  삽질 삽질....삽질.....삽질...    서버 내렸다 올려보고    DB 지워보기도하고    수동으로 테이블 생성도 해보고..-\_-원인은.... 두둥... 리눅스mysql은  대소문자 구분한다고 합니다.  췟.(이눔들.ㅠㅠ 일관성이없어요)리눅스 보안문제발생할수 있어서 그렇다나 뭐라나오랜만에 글남겨봅니다^^ 좋은하루되세요. [http://www.gravatar.com/avatar/7f4f11c2ea439069b9eb84cba24405d1?d=identicon&s=40](http://okky.kr/user/info/16629) 리눅스의 파일 시스템은 대/소문자를 구분하죠. 모르는 분들이 많더군요.가끔은 말해줘도 안 믿기도 하고... 유사한 문제로 ￦r ￦n 문제도... 그리고 쉘 만들때 ^M 문제도..  [http://www.gravatar.com/avatar/5bac46d1b35f588588b6a53b9e40fc1d?d=identicon&s=40](http://okky.kr/user/info/19799)  [twlee11](http://okky.kr/user/info/19799) Mysql 대소문자 구분 여부변경 파라미터가 있는데요 찾아보세요 |
| URI Encode Decode 처리시 3가지 타입이 있어 잘못 처리하여 Encode, Decode 처리가 안된적이 있다.최소한의 문자만 인코딩 합니다. encodeURI, decodeURI : %EC%86오래된함수 중간의 정도 범위로 문자를 인코등 합니다 escape(), unescape() : %uC18%uC124광범위하게 인코딩 합니다 encodeURIComponent, decodeURIComponent : %2C%AC%BD%BD%81%00 |

톰캣쪽에 스프링 스케쥴, 및 쿼츠 스케쥴이 동시에 2번 실행되는거에 대한 삽질

|  |
| --- |
| 내용 : 톰캣쪽 webapp 쪽에 모두다 지우고 cbis.war를 가져다 놓고  startup.sh을 하게되면  cbis, ROOT, cbis.war  이렇게 cbis폴더와 ROOT폴더가 생긴다 이에따른 webapplication이 2개가 뜬다  이렇기 떄문에 서로 똑같은 log파일에 남기기도 하고 서로 각각 스케쥴을 돌리게된다  해결 : server.xml쪽 Host쪽에  deployOnStartup="false"  unpackWARs="false"  autoDeploy="false  추가 |
| 참고  **序.** 로컬에서의 테스트에서는 극히 정상적으로 quartz스케쥴러가 작동하였고, 이를 테스트서버와 실서버로 옮겼을때는 동시에 두번씩 동작하는 문제가 발생했다. 세 서버의 Tomcat/Spring/Quartz의 버전은 완전히 동일...  **1.** 일단 큰 차이랄 수있는 OS/JVM의 차이를 의심하였다. 구글링을 해보니 비슷한 증상(윈도우 정상작동, 리눅스 스테이징으로 옮기니 발생)인 사람들을 여럿 발견... 희망이 보이기 시작....  벗뜨.. 이런 얘기한 사람들은 다 답을 얻지 못함...  **2.** 다음은 Spring 혹은 Quartz의 버그를 의심. [Spring + Quartz의 조합으로 비슷한 증상을 호소한 사람들](http://forum.springsource.org/archive/index.php/t-37087.html)이 미쿡(= 영어였다는 얘기다;;;)에는 꽤 많았다... 이는 web.xml에서 Spring의 ContextLoader가 중복호출되어 복수의 Quartz인스탄스가 생겨 발생하는 경우였다. 웹어플리케이션 상에서 둘 이상의 Listener or Servlet을 등록때 하나의 context정보 파일을 두 번 이상 부를 수 있는 경우가 있다는 것이다.  webapp의 lifecycle과 Spring의 Bean생성 과정을 자세하게 알지 못하면 쉽게 할 수 있는 실수였다... 드디어 찾았구나~ 하고 좋아했다...   근데...  이 경우라면 로컬에서는 이상없는 것이 설명되지 않았다...  로컬에서 이상이 없는 이유를 찾느라 한참을 소비해야했다...  결론적으로... 내 경우의 Spring설정은 완벽했다;;;  **3.** 처음부터 서버 설정을 크게 의심하지 않은 것은, 테스트 서버와 실서버가 모두 잘 돌고 있었기 때문이다. 또한 설정파일 역시 모두 공유해서 사용하고 있었기 때문에 시야를 벗어나는 좋은 이유가 되었다. 처음부터 톰캣 인스탄스가 2개인것은 의심해 보았다. 2개가 떠있어서 하나는 포트를 못열고 놀고만 있는... 그러나 스케줄러는 작동되고 있는... 하지만 이것에 해당하지는 않았다.  유일하게 공유되고 있지 않은 설정파일인 톰캣의 server.xml파일을 열어봤다...  .... <Host name="localhost" appBase="/tomcat/webapps"     unpackWARs="true" autoDeploy="false"     xmlValidation="false" xmlNamespaceAware="false">     <Context path="" docBase="/tomcat/webapps/test" reloadable="false" /> </Host> ....  뭔가 감이 왔다;  그렇다, Host의 appBase는 여러 webapp들이 들어갈 母 디렉토리다. 즉, 이하의 디렉토리는 webapp의 자격조건을 가지고 있으면 자동으로 webapp로 구동되는 것이다. Context는 이와는 별도로 webapp의 위치를 직접 지정가능하다.  결국 위의 설정은 두개의 'test' webapp를 구동하도록 만들었다. 스케쥴러도 두개가 뜬것이다.  Tomcat에서 동일한 Web Application Context 두 개가 뜬다한들 작동에는 별 문제가 없을 것이다. 둘의 ContextPath가 다르다면 말이다. 위의 경우 Host에서는 "test"라는 ContextPath로 webapp가 떴을 것이고, Context에서는 지정한대로 ""(=루트) ContextPath로 떴으니 아무런 문제가 없었던 건이다.  여튼 Tomcat설정시 Host의 appBase위에 Context로 설정할 app를 올리지 말아야한다. 이는 의도하지않은 작동을 야기한다. 본인은 다양한 이유로 appBase의 위치를 바꾸는 방법을 택했다.  **結.** Quartz의 스케쥴러가 이중작동하는경우(굳이 이러한 경우가 아니더라도, 서버구동시 한번만 실행되도록 했는데 두번 실행되는 모든 경우).....  우선 Spring을 사용하는 경우는 Spring설정에 주의를 기울여라. Spring이 인스탄스의 라이프사이클을 제어하는 경우가 많아서, 개발자가 미처 인지하지 못하는 사이 인스탄스화가 일어날 수 있다. 구체적인 사례는 [여기](http://forum.springsource.org/archive/index.php/t-37087.html)를 참조한다.  이게 아니라면 Container의 라이프사이클을 의심하라. 혹시 프로세스가 두개가 뜨지는 않았는지... 프로세스는 하나지만 복수의 Context를 구동하게 되지는 않았는지... 이 모든 것은 Container의 설정으로 해결해야 할 것이다. 1번의 경우(OS, JVM의심)도 결국 이러한 상황이었던 것으로 보인다.  Spring 에서 quartz스케쥴러를 사용하다가 같은 task 가 중복 실행되는 문제로... 한참을 웹서핑을 했다.   같은 증상을 원인까지 잘 정리된 블로그 발견..  http://blog.ajkuhn.com/116  해당 블로그의 2번째 원인이었음..  아그 내가 설정을 잘못했는데 누구탓을 하나 쩝.  출처: <http://joshua90.tistory.com/entry/SPRING-scheduler-%EC%A4%91%EB%B3%B5-%EC%8B%A4%ED%96%89-%EB%AC%B8%EC%A0%9C> [J love J]  2015.06.29 01:02  로컬에서는 한번만 실행하면 되는데 linux 서버에 올리면 두번이 실행되었다.  왜그런지 블로그 다른글을 찾아보니 원인이 잘정리 되어있는듯 하지만 무슨말인지 이해하기가 힘들었다.  http://blog.ajkuhn.com/116  블로그내용을 보니 아래와같은 내용이 있었다.  Tomcat에서 동일한 Web Application Context 두 개가 뜬다한들 작동에는 별 문제가 없을 것이다. 둘의 ContextPath가 다르다면 말이다. 위의 경우 Host에서는 "test"라는 ContextPath로 webapp가 떴을 것이고, Context에서는 지정한대로 ""(=루트) ContextPath로 떴으니 아무런 문제가 없었던 건이다.  여튼 Tomcat설정시 Host의 appBase위에 Context로 설정할 app를 올리지 말아야한다. 이는 의도하지않은 작동을 야기한다. 본인은 다양한 이유로 appBase의 위치를 바꾸는 방법을 택했다.  이해가안가서 영문으로된걸로 찾아보니 스택오버플로우에 완전히 잘정리된 글이있었다.  http://stackoverflow.com/questions/7223108/quartz-job-runs-twice-when-deployed-on-tomcat-6-ubuntu-10-04lts  When using automatic deployment, the docBase defined by an XML Context file should be outside of the appBase directory. If this is not the case, difficulties may be experienced deploying the web application or the application may be deployed twice. The deployIgnore attribute can be used to avoid this situation.  Finally, note that if you are defining contexts explicitly in server.xml, you should probably turn off automatic application deployment or specify deployIgnore carefully. Otherwise, the web applications will each be deployed twice, and that may cause problems for the applications.  해결책은  1.위에분과같이 톰켓 host 설정의 appBase를 바꾸는 방법과  2.  오토디플로이 옵션과 , 디플로이온 스탈트업을 끄는 방법이있다.  <Host name="localhost"  deployOnStartup="false"  appBase="webapps"  unpackWARs="false"  autoDeploy="false">  2번에 관해정리된곳이 없는거같아 글을 써본다.  구글봇이 잘긁어갔으면 좋겠다.  끗 |
| <Host name="localhost" appBase="webapps"  unpackWARs="true" **autoDeploy="false"** deployOnStartup="false">  <Context path="" docBase="cbis" reloadable="true" sessionCookieName="practice\_JSESSIONID" > |
| webapps에 있는 war가 자동으로 풀려서 contextpath를 가지게된다 또한 ROOT 로도 뜨기때문에  두개의 content가 뜨게된다. |

같은 웹어플리케이션을 이중화 L4처리를 했을때

세션 쿠키 이름이 달라 세션이동기화가 안되는경우가 있다 즉

한쪽붙으면 한쪽만 계속 붙도록 하는 L4정책이라면 각각 서버의 웹톰캣쪽에서 sessionCookieName을 다르게 지정 해야지 로그아웃이 안된다.

|  |
| --- |
| <Context path="" docBase="cbis" reloadable="true" sessionCookieName="practice\_JSESSIONID" >  </Context> |

톰캣 옵션주기

bin폴더안쪽에 setenv.bat 이나 sh을 파일을 한번 call 하기때문에 여기 파일을 만들고 내용을 지정한다

예를들어 setenv.bat

|  |
| --- |
| set JAVA\_OPTS=%JAVA\_OPTS% -Duser.timezone=GMT+09:00 |

자바 스트링

|  |
| --- |
| 한글 한글짜당 2~3바이트 (UTF-8) 처리되는걸 Byte단위에서 짜르다가 반각으로 짜른뒤  다시 String으로 변환하게되면 반각된것이 이상한 문자로 처리된다 이상한문자(�)로 처리된것또한 2바이트 이기때문에 처음에 5byte로 짜른것이 -> String변환후 -> byte로 변환하면 6byte가 되어버린다. |

|  |
| --- |
| 특정 String 변수값이 지속적으로 뒤쪽에 다른 내용이 append되는 현상 이 발생되었다.  이로인해 점점 gc의 주기가 가까워지며 사용할수 있는 힙메모리는 점점 줄어드는 상황이 발생되었다.  public class StringTest {  String name;  public String getName() {  return name;  }  public void setName(String name) {  this.name = name;  }   public static void main(String[] args) throws InterruptedException {  StringTest stringTest = new StringTest();  while (true){  stringTest.setName(stringTest+" "+System.*currentTimeMillis*());  Thread.*sleep*(0);  }  }   @Override  public String toString() {  return "StringTest{ name='" + name + "'}";  } }  위처럼 지속적으로 name값에 새로운 내용이 append되는 상황이 발생되어(논리오류)  자바 힙메모리 빈번하게 gc가 발생과 용량 증가 되어 응답속도가 점점 느려지는 현상이 발생되었습니다. |

그루비 엔진을 불러다 사용할때 엄청난 그루비 라이브러리를 매번 로딩하기때문에

느려진다. 그루비 템플릿같은거 쓸려고할때 이랬음

|  |
| --- |
| 1시간가량 지속적인 부하를 줬을때. |
| 조치 : 벨로시티 템플릿으로 변경하여 그루비엔진을 매번 로딩하는 부분을 제거하였다        같은 1시간 부하줬을때 Total값또한 많이 줄어든걸 확인할수 있다. |

Groovy engine 사용시 문제

안녕하세요 서비스팀 김현하 입니다.

nhis 최근이슈중인  메모리 회수 문제 1차조지 정보 공유합니다.

증상 : 메모리 부족으로 인한  weblogic 버벅이다 죽음

분석 : 4개 지사에서 사용중일때  -> 15일에 한번씩 죽던 weblogic

        16개 지사 사용할때   -> 7일에 한번씩 weblogic 죽는상황

추론 : 순간적인 부하발생에 따른 문제가 아니라  사용율이 따른 누적피로도 에따른 문제로 의심

증거 : 7일간 돌아가던 weblogic  Heap덤프 확인하니  groovy.reflection.ClassInfo등  GroovyClassLoader측에서 Heap메모리 절반 이상먹고 회수 안되는? 정보 확인

테스트환경 : 1시간 jmeter로 집중 부하(groovy template사용 부분)

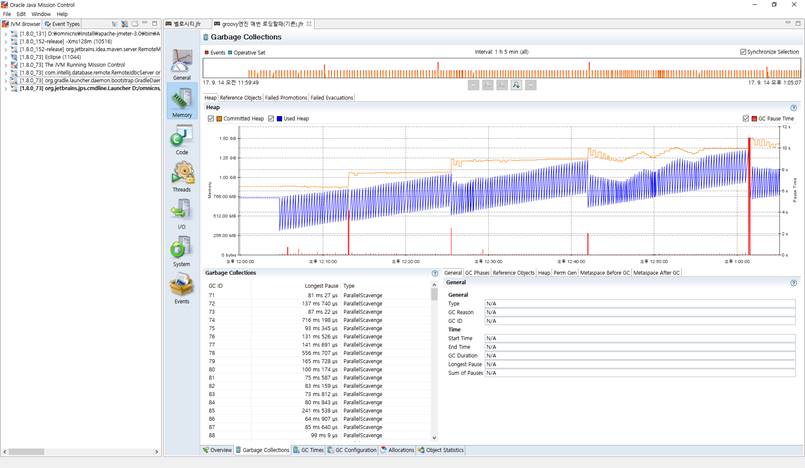
확인 : 1. 자바에서 groovy 엔진을 로딩하여 사용할때는 groovy.all....\*jar에있는 class등을 처음에 로딩하기때문에 주기적으로 호출 하면 버벅일수 없다는 구굴신의 답변

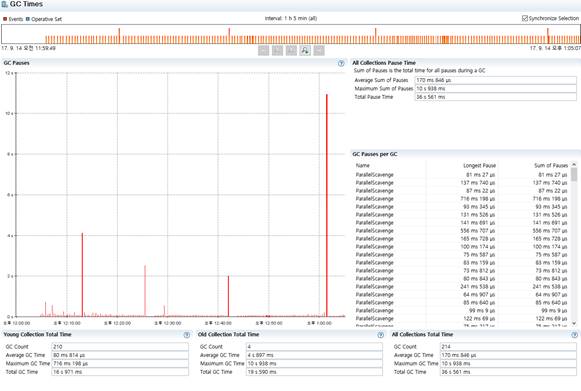
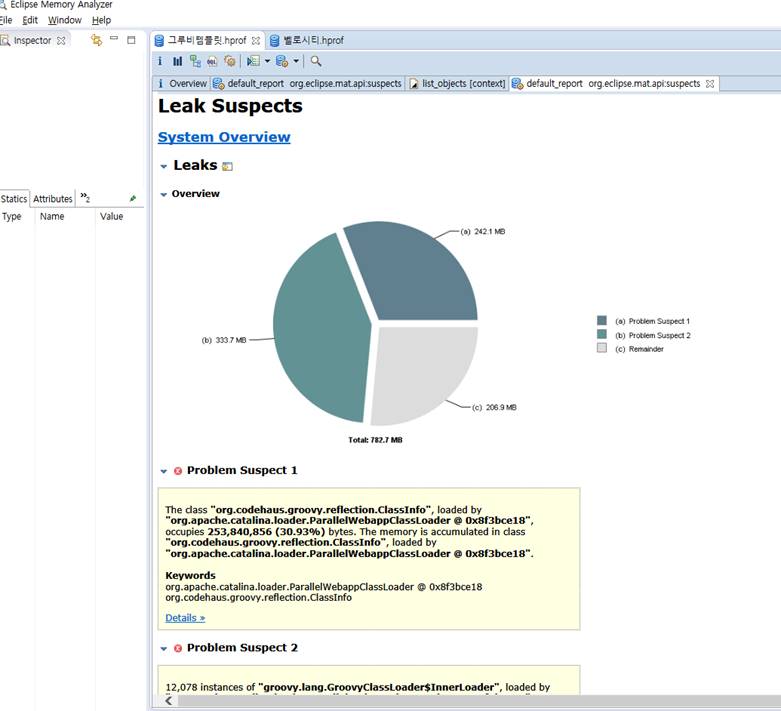
         2. 잦은 GC주기

조치 : 기능변경 (TemplateEngin)

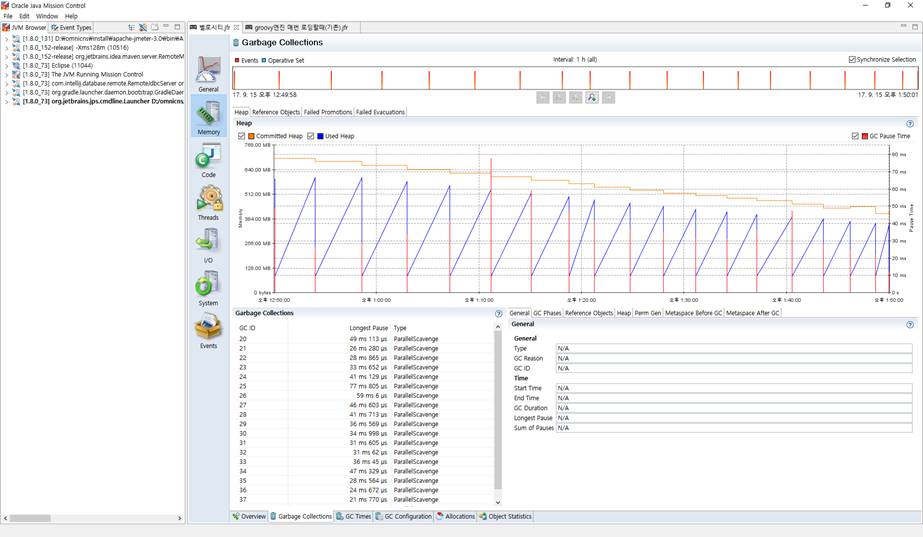
         GroovyTemplate 사용 하던걸 -> VelocityTemplate로 변경

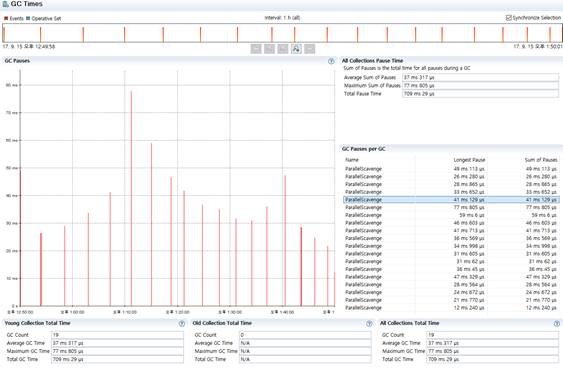
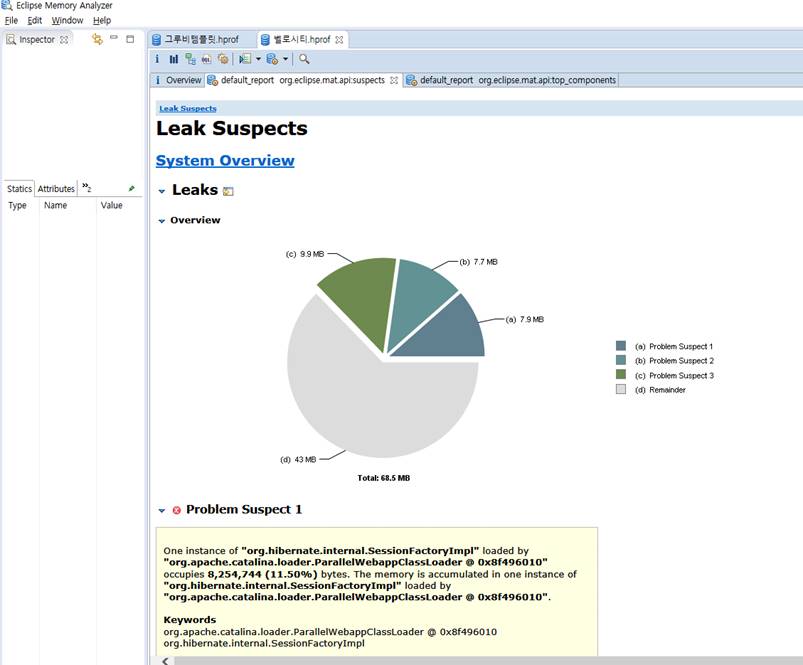
**AS-IS : Heap메모리 782MB**





**TO-BE: Heap 메모리 68.5MB 줄어듬**





git 삽질

|  |
| --- |
| 안녕하세요 서비스팀 김현하입니다.  Git source 반영 절차시 유의사항 (사례) 있어 공유합니다.  \*사례 내용  **발생 : 특정 파일 원복+변경 되어 시스템 오류 발생**  증거 수집 :   1. 07.06 최신소스 반영 -commit 김현하 2. 08.04 작업에 따른 소스 수정 -commit 서동혁 3. 09.07 별도의 파일로 수정됨 -commit 김현하   발생 원인 및 이유 :  배경 : 동시 다발적인 작업에 따른 여러 branch 생성하여 작업  원인 : **A branch에서 사용하는 일부 기능을 B branch에서 사용하기 위하여**  **일부 파일 내용을 수동 (copy and paste) merge (B ← A) 하면서 B branch에 있던 일부 내용이 A내용으로 변경됨**  확인 : **09.07 별도의 branch에서 가져온 파일을 08.04일날 수정된 내용을 확인하지 않고**  **수동 merge를 하여 생긴 문제 -commit 김현하**  조치 :   1. master Branch에 A branch 내용을 정식 merge 하여 기능자체를 녹여 수동 merge처리를 무효화 시킴   방지대책 :   1. 수동 merge (copy and paste) 최소화 2. branch merge시 파일내용 확인    1. branch diff 이용하여 확인가능    2. 현재 branch와 다른 branch끼리의 차이점 비교       1. git diff BRANCH\_NAME       2. git diff --name-status BRANCH\_NAME    3. 다른 branch 파일 복사해오기       1. git checkout BRANCH\_NAME FILE\_PATH    4. 다른 branch 파일 patch 해오기       1. Git checkout -p BRANCH\_NAME FILE\_PATH 3. branch merge Conflicts시 이해 안되는 부분은 작업자와 충분한 대화 후 merge 4. auto merge를 100% 신뢰하면 안된다. |

|  |
| --- |
| jsp에서 참조하는 static final String 변수값을 변경하였지만 jsp쪽에 반영이 되지 않는 문제 발생  $('a[data-toggle="tab"]').on('shown.bs.tab', **function** (e) {  **var** target = $(e.target).attr("href") // activated tab  **if** (target == "#content-vitalsign") {  $("#diagnosis\_type").val("**<%=**Code.*DIAGNOSIS\_TYPE\_MEASURE***%>**");  grid.draw();  } **else if** (target == "#content-examination") {  $("#diagnosis\_type").val("**<%=**Code.*DIAGNOSIS\_TYPE\_RSCH***%>**");  grid.draw();  } });  DIAGNOSIS\_TYPE\_MEASURE 변수값을 DT001 -> DD001 로 변경만 하고 jsp쪽은 소스 수정이 없었다  그래서 이전 컴파일된 내용이 들어가 있다.  기존jsp 컴파일된내용  out.println("$("#diagnosis\_type").val("DT001");">  jsp파일이 수정되지 않았기때문에 위쪽의 내용이 그대로 출력 변경된 내용 으로 출력되지 않음  해결방법  1. jsp소스를 수정하여 재컴파일 하거나  2. /home/mindcare/tomcat/work/Catalina/localhost/ROOT/org/apache/jsp/WEB\_002dINF/jsp/caution/\*.\*  의 jsp컴파일된 class폴더가서 \*.class 지우면된다 |

spring boot 에서 spring data jpa 를 쓰지만 오리지널 hibernate를 쓰고 싶을때

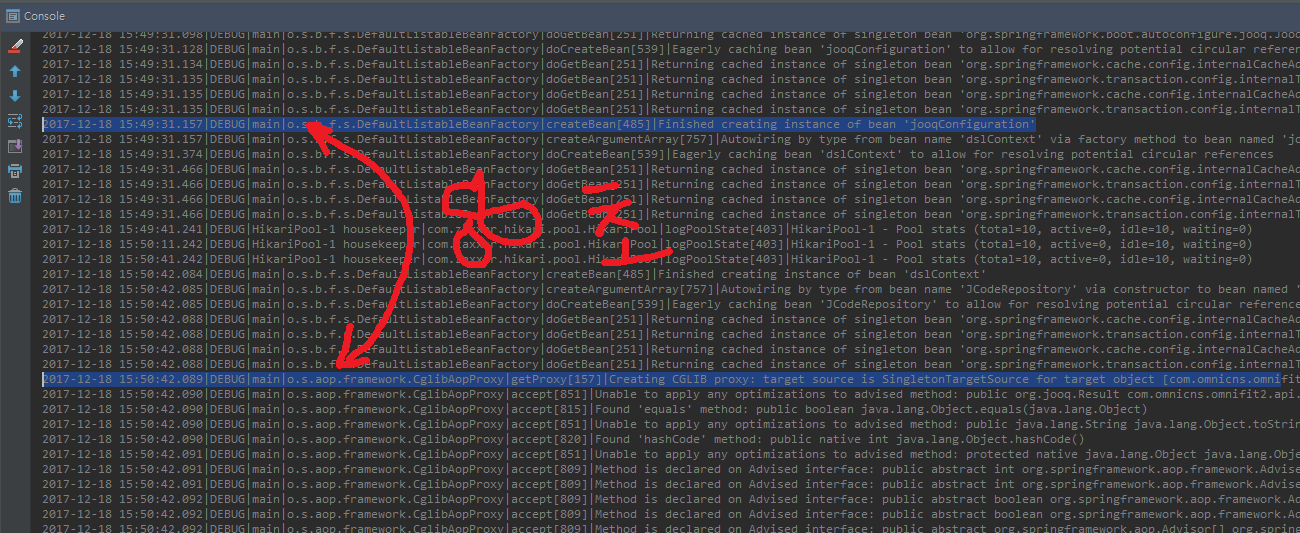
|  |
| --- |
| @Configuration @Import({CommonWebMvcConfigurerAdapter.class}) @EnableWebMvc @EnableConfigurationProperties(ProjectProperties.class) @EnableScheduling @EnableTransactionManagement public class WebMvcConfigurerAdapter extends org.springframework.web.servlet.config.annotation.WebMvcConfigurerAdapter {   // @Autowired // private EntityManagerFactory entityManagerFactory;    // @Bean // public SessionFactory getSessionFactory() { // if (entityManagerFactory.unwrap(SessionFactory.class) == null) { // throw new NullPointerException("factory is not a hibernate factory"); // } // return entityManagerFactory.unwrap(SessionFactory.class); // }  @Bean(name = "sessionFactory") @Autowired  public SessionFactory getSessionFactory(HibernateEntityManagerFactory g) {  return g.getSessionFactory();  }   @Bean @Autowired  public HibernateTransactionManager transactionManager(SessionFactory s) {  HibernateTransactionManager transactionManager = new HibernateTransactionManager();  transactionManager.setSessionFactory(s);  return transactionManager;  }    ////////////////////hibernate view //////////////////////////////////  @Bean  public FilterRegistrationBean openSessionInViewFilter() {  FilterRegistrationBean registrationBean = new FilterRegistrationBean();  OpenSessionInViewFilter filter = new OpenSessionInViewFilter();  registrationBean.setFilter(filter);  registrationBean.setOrder(5);  return registrationBean;  }  @Bean  public FilterRegistrationBean openEntityManagerInViewFilter() {  FilterRegistrationBean registrationBean = new FilterRegistrationBean(new OpenEntityManagerInViewFilter());  OpenEntityManagerInViewFilter filter = new OpenEntityManagerInViewFilter();  registrationBean.setFilter(filter);  registrationBean.setOrder(5);  return registrationBean;  } ///////////////////////////////////////////////////////////////////  처럼 OpenSessionInViewFilter를 걸어줘서 모든 인입점에서 OpenSession이 되도록 필터를 걸어준다  그러하면 OpenSessionInViewFilter에서  Session session = this.openSession(sessionFactory); SessionHolder sessionHolder = new SessionHolder(session); TransactionSynchronizationManager.bindResource(sessionFactory, sessionHolder);  처리를 하기때문에  안쪽 서비스단에서  @Service @Slf4j @Repository public class AuthService {   @Autowired  private SessionFactory sessionFactory;  @Autowired  private HibernateTransactionManager hibernateTransactionManager; // @Autowired // private LocalSessionFactoryBean localSessionFactoryBean;   @PersistenceContext  private EntityManager em;   @Transactional  public Object getGG(){  SessionHolder gg = (SessionHolder)TransactionSynchronizationManager.*getResource*(sessionFactory);  Session s = hibernateTransactionManager.getSessionFactory().getCurrentSession();   Adm adm = new Adm();  adm.setAdmNm("김");  s.save(adm);   adm = new Adm();  adm.setAdmNm("현");  s.save(adm);    adm = new Adm();  adm.setAdmNm("하");  s.save(adm);   return "";  }  }  처럼 사용하면된다. TransactionManager도 있기때문에 다걸린다. |

SQL

|  |  |  |
| --- | --- | --- |
| LEFT OUTER JOIN 삽질   |  |  | | --- | --- | | TEST\_TEMP01 | TEST\_TEMP03 |   **SELECT** \* **FROM** TEST\_TEMP01 **A** , TEST\_TEMP03 B **WHERE A**.NO = B.TEMP03\_SEQ(+)    위처럼 B쪽에 NULL값으로 LEFT OUTER JOIN 걸려 나온걸 확인할수 있다 하지만  아무 생각없이  **SELECT** \* **FROM** TEST\_TEMP01 **A** , TEST\_TEMP03 B **WHERE A**.NO = B.TEMP03\_SEQ(+)  **AND** (B.CREATE\_DT **IS NOT NULL**)  을 걸어버리면    나와버린다.  --------  아래는 논리오류난것이다.    **SELECT** PRNT.CD **AS** PCD,  **CHILD**.CD **AS** CCD,  **CHILD**.PRNT\_CD **FROM  TEST** PRNT , TEST **CHILD WHERE** PRNT.CD = **CHILD**.PRNT\_CD(+) **AND** (PRNT.API\_USE\_YN = 'Y' **AND CHILD**.API\_USE\_YN = 'Y')  위처럼 하면 CHILD쪽에 API\_USE\_YN이 LEFT OUTER JOIN 걸린곳에는 널값이기때문에 셀프조인이 걸리지 않는다.  정상  **SELECT** PRNT.CD **AS** PCD,  **CHILD**.CD **AS** CCD,  **CHILD**.PRNT\_CD **FROM  TEST** PRNT , **TEST CHILD WHERE** PRNT.CD = **CHILD**.PRNT\_CD(+) **AND** (PRNT.API\_USE\_YN = 'Y' **AND** PRNT.PRNT\_CD **IS NULL**) **AND** (**CHILD**.API\_USE\_YN = 'Y' **OR CHILD**.API\_USE\_YN **IS NULL**); ;  또는  SELECT  PRNT.CD,  CHILD.CD,  CHILD.PRNT\_CD  FROM  TEST PRNT  LEFT OUTER JOIN TEST CHILD  ON PRNT.CD = CHILD.PRNT\_CD  AND CHILD.PRNT\_CD IS NOT NULL  AND CHILD.API\_USE\_YN = 'Y'    WHERE 1 = 1  -- (PRNT.API\_USE\_YN = 'Y' AND CHILD.API\_USE\_YN = 'Y')  AND PRNT.PRNT\_CD IS NULL  AND PRNT.API\_USE\_YN = 'Y' |

**spring boot 에서 JOOQ사용시 구동 1분이상 느려짐 현상 (버그)**

**1. 현상**



boot 구동시에 80초이상 느려짐 발생

**2. 윈인점 파악**

AOP쪽 execution 표현식에 따른 JOOQ와의 버그

<https://github.com/jOOQ/jOOQ/issues/5902>

@Aspect @Component @Order @Slf4j  
public class CheckHeaderAOP {  
 @Before("execution(\* com.omnicns.omnifit2.api.controller..\*Controller.\*(..))")   
 public void checkHeaderBefore(JoinPoint joinPoint) throws IOException {  
 *log*.debug("AOP");  
 }

}

aspectjweaver가있는 버그라고 생각합니다.

어떤 이유로, org.aspectj.weaver.internal.tools.PointcutExpressionImpl의 couldMatchJoinPointsInType 메소드는 실행과 함께 pointcut을 사용할 때 true를 리턴하고,이 때문에 DefaultDSLContext의 모든 메소드가 점검되어 포인트 컷이 메소드에 적용될 수 있는지를 확인합니다. execution 대신 within를 사용하면 문제가 해결됩니다.

**- 출처 jooq git issues**

**3. 문제 해결**

|  |
| --- |
| @Before("within(com.omnicns.omnifit2.api.controller..\*)") public void checkHeaderBefore(JoinPoint joinPoint) throws IOException {  *log*.debug("AOP"); } |



**트랜젝션 컨트롤단에 걸지 서비스단에 걸지 ??**

|  |
| --- |
| <https://www.slideshare.net/resoliwan/2-mysql-v0>  <https://medium.com/omarelgabrys-blog/tagged/other>  <https://joebaak.blogspot.kr/2017/06/mysql-index-io-bufferpool-lock-count.html> |

Spring **ThreadPoolTaskExecutor 사용시 lock 현상**

<https://github.com/HomoEfficio/dev-tips/blob/master/Java-Spring%20Thread%20Programming%20%EA%B0%84%EB%8B%A8%20%EC%A0%95%EB%A6%AC.md>

**ThreadPoolExecutor**

* Java 5 에서 도입
* AbstractExecutorService를 상속하며 ExecutorService의 실질적인 구현체
* 스레드 풀 관리를 위한 다양한 기능 포함
* Thread는 기본값으로 Executors.defaultThreadFactory()에 의해 생성되며, 이렇게 생성된 스레드는 모두 동일한 ThreadGroup에 속하며 동일한 NORM\_PRIORITY 우선 순위와 non-daemon status를 가진다.
* 주요 인스턴스 변수
  + corePoolSize: 풀의 기본 사이즈. 기본값 없음
  + maxPoolSize: 풀의 최대 사이즈. 기본값 없음
  + keepAliveTime: corePoolSize를 넘는 스레드나 allowCoreThreadTimeOut이 true 일 때 corePoolSize 이내의 idle 스레드는 keepAliveTime이 지나면 TimeOut 된다.
  + allowCoreThreadTimeOut: true이면 corePoolSize를 넘지 않는 스레드도 idle 상태이면 TimeOut 시킨다.
* Thread 생성 기본 전략
  + 풀의 활성화 된 스레드 수가 corePoolSize 보다 작은 상태에서 execute() 또는 submit() 호출 시 스레드 새로 생성
  + 풀의 활성화 된 스레드 수가 corePoolSize와 maxPoolSize 사이에 있으면서 큐가 꽉 차있지 않으면 새 스레드를 생성하지 않고 태스크를 큐로 전송
  + 풀의 활성화 된 스레드 수가 corePoolSize와 maxPoolSize 사이에 있으면서 큐가 꽉 차있으면 스레드 새로 생성
  + 큐도 꽉 차있고 활성 스레드 수가 maxPoolSize에 도달해있는 상태에서 태스크가 또 들어오면 에러 발생
  + corePoolSize와 maxPoolSize를 같게 하면 결국 고정 크기 스레드 풀을 생성하게 된다.
  + **한 마디로 corePoolSize 만큼 스레드를 채우고, Queue 를 채우고, maxPoolSize 만큼 채운다.**

같은 pool을 사용하고 있다는 전자하에

쓰레드에서 쓰레드를 호출하고(join) 그 호출된 쓰레드가 task쪽에 빠져버리면

호출된 쓰레드는 task쪽에 들어가기때문에 실행이 안되고 호출한 쓰레드는 task에 들어간 쓰레드를 기다리기떄문에(join) 자기또한 멈춰져있다

인증서 갱신시 이슈

인증서 적용중 체이닝 이슈 공유드립니다.

현재 nginx를 웹서버로 사용중에있으며

정상 적용 절차

1. 발급기관에서 발급받은 인증서 압축 풀기
2. cat domain\_domain\_co\_kr.crt ./ChainCA/rsa-dv.chain-bundle.pem > accept\_domain\_domain\_co\_kr.crt
3. nginx 적용
   1. ssl\_certificate: 위에서 생성된 파일 (accept\_domain\_domain\_co\_kr.crt)
   2. ssl\_certificate\_key: 발급기관에서 제공해준 key(domain\_domain\_co\_kr.key)

인증서 확인: <https://www.ssllabs.com/ssltest/>

발급기관 적용 절차: <https://www.comodossl.co.kr/certificate/ssl-installation-guides.aspx>

* 위처럼 인증서를 합치게되면 체인문제가 발생될수 있음(체인이슈)

감사합니다.