



Portfolio

목차

Burgerput

- 웹 컨트롤 라이브러리 셀레니움을 이용한 Spring 웹 봇 서버 제작

구루멘토링

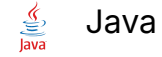
- 종합정보망 포트포워딩 작업
 - Apache 서버 충돌 문제 해결
- 육군사관학교 FTP서버 이전
 - 서버 이전을 수행하며 발생한 언어 인코딩 문제 해결
- 종합정보망 테스트케이스 설계
 - 웹 테스트케이스를 통한 점검 방법 기획 설계

Burgerput

2023.05-진행중

프로젝트 공식문서 링크

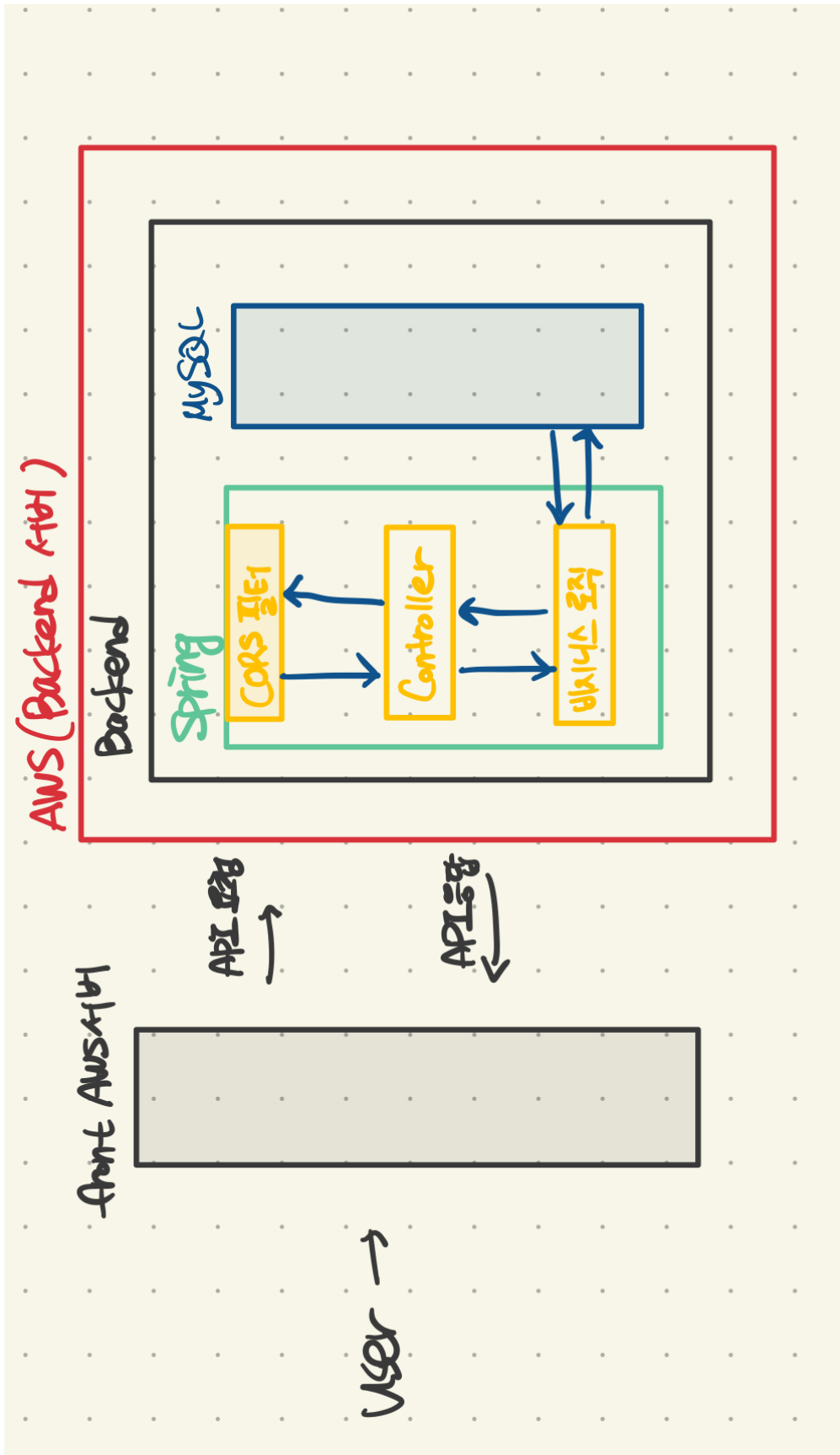
사용기술



개요

젠핏이라는 웹 페이지에서 불필요한 요소(사용하지 않는 기기, 제품)는 burgerput을 경유하고 필터링을 통해 오기입의 가능성을 0%로 만드는 웹 봇 프로그램이다.

시스템 구성



웹 페이지의 정보를 특정 시간대에 크롤링으로 정보를 가져온다. 해당 정보를 DB에 저장하고 해당 정보를 사용자에게 보여준다. 사용자는 자신이 사용하는 기기, 식품만을 선택해서 온도를 입력해서 전달 할 수 있다.

사용자가 선택하지 않은 정보는 Burgerput서버에서 999로 자동으로 채운 후 전달한다.

수행작업

- AWS서버를 이용해 front-end단과 back-end서버단과 API로 통신하는 웹 애플리케이션 구성
- Selenium라이브러리를 사용해 웹을 컨트롤하는 웹 애플리케이션 구성
- JPA, JPQL을 이용해 DB와 통신
기본적인 통신 save, update는 JPA기술을 통해서 수행했다. delete는 DB의 공간을 낭비하지 않기 위해 truncate를 사용하는 방식을 사용했다. Delete는 행 하나를 삭제하는 게 아니라 Table 전체의 내용을 삭제했다가 넣는 로직이었기 때문에 위와 같은 방법을 채택했다.
- JWT토큰을 사용한 안전한 로그인 방법 구축
의도하지 않은 사용자의 접근을 제한하기 위해 JWT토큰을 사용해 접근을 제한했다. 토큰이 없는 사용자는 로그인 페이지로 리다이렉트하게 된다. 사용자의 Access Token이 만료되면 refreshToken을 POST 메소드를 통해 front단에서 보내준다. refreshToken을 검증하고 새로운 AccessToken을 발급한다. 반대로 검증되지 않은 토큰이면 로그인 페이지로 리다이렉트한다.

관련 블로그 링크

- 로딩속도를 줄이기 위한 코드 리팩토링 수행
로딩속도 개선을 위해 웹 페이지를 다시 분석하고 에러가 발생했을 때 try-catch를 통해 에러처리를 수행했다. 불필요하게 대기하던 100초가 발생하는 부분을 찾았고 해당 부분을 제거하여 로딩의 속도를 120초에서 16초로 개선시켰다.

관련 블로그 링크

- 로딩의 결과를 매일 아침 로딩으로 확인하는 메일시스템 구축
로딩 결과 체크 자동화를 위해 cron과 쉘 스크립트를 이용해서 로딩의 결과를 메일로 보내는 시스템을 구축했다.
- 제출 로직 버그 수정
제출로직이 제대로 발생하지 않는 버그가 있었다. 해당 버그의 원인을 알아보기 위해 웹 드

라이버를 구동하고 중간과정에 캡처로직을 추가해 화면을 확인해보는 방법, 웹 컨트롤이 제대로 확인이 되는지 테스트 페이지에 UI로 나타나게 하는 방법을 사용했다. 그 결과 제출버튼이 제대로 눌리는 것을 확인했지만 그래도 제출이 제대로 되지 않았었다. HTTP의 요청에 대해 집중해서 다시 한 번 데이터가 흘러가는 흐름을 정리했고 그 결과 요청을 제대로 보내졌으나 응답을 기다리지 않고 드라이버를 종료해서 나타나는 문제임을 확인했다. 해당 문제를 확인하고 응답 대기시간을 주어 제대로된 요청-응답을 받아 제출할 수 있었다.

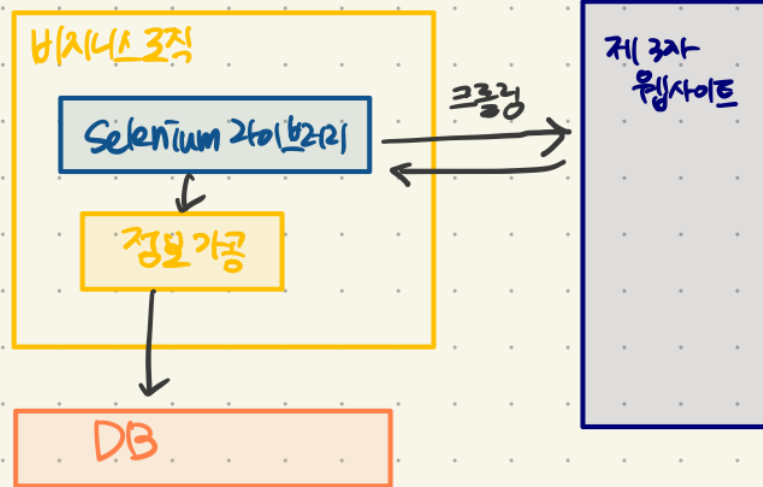
관련 블로그 링크

작업 리뷰

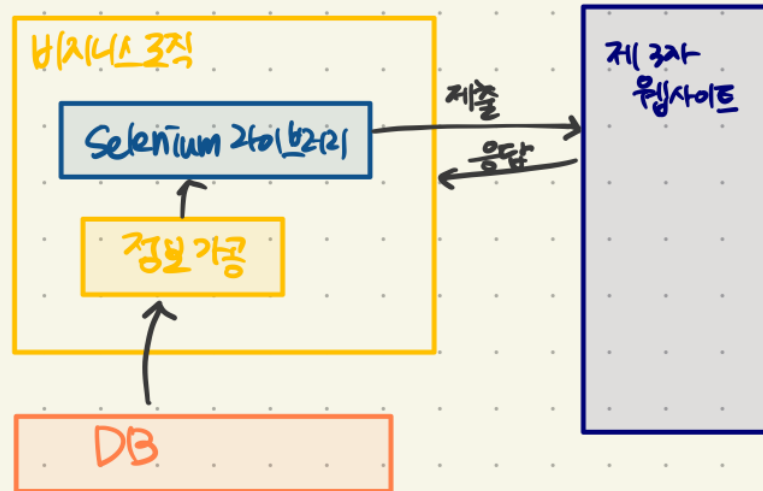
- 새롭게 라이브러리를 사용해 보면서 AWS에서 구동이 되지 않던 문제, 로딩이 제대로 수행되지 않던 문제, 리팩토링을 통한 애플리케이션 수행시간 단축등을 겪으면서 서버를 고려하면서 개발을 해야되겠다는 생각을 갖게되었다.
- frontend와 서로 다른 도메인으로 통신하면서 통신할 때에는 보안규약때문에 CORS와같이 고려할 것이 많고 또한 개발할때에도 그걸 많이 고려해야한다는 것을 알았다. 헤더에 same-site 옵션을 none으로 설정해놓아 CORS문제를 해결했지만 보안에서 취약해질 수 있다는 약점이 있었다.
- 웹 서버를 크롤링하는 악성 웹 봇이 많다는 것을 느꼈다. 실제로 서버를 운영해보니 의도치 않은 GET요청이 많이 발생한것을 확인할 수 있었고 대부분 웹사이트를 크롤링하기위한 웹 봇이었다. 이런 부분을 고려해 서버운영시 보안을 필수로 생각해야한다는 계기가 되었다.
- HTTP는 언제나 요청-응답의 쌍이라는 것을 알게되었다. 어렵듯이 알고있었지만 막상 작업할 때에는 크게 신경쓰지 않았던 부분이었고 제출 에러를 수정하게 되면서 요청뿐 아니라 응답도 확실하게 확인해야 한다는 것을 깨달았다.

자료

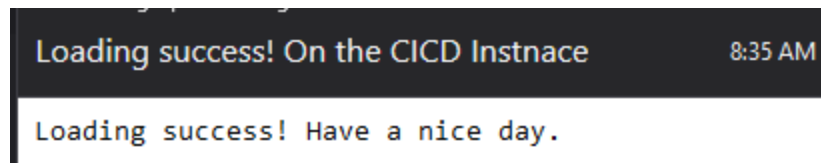
로딩로직



제출로직



[웹 컨트롤 셀레니움 라이브러리를 이용한 웹사이트 로딩로직(크롤링) 제출로직]



[로딩 로직의 결과를 이메일로 확인]

구루멘토링

2019.11-2020.07

종합정보망 포트포워딩 작업

사용기술

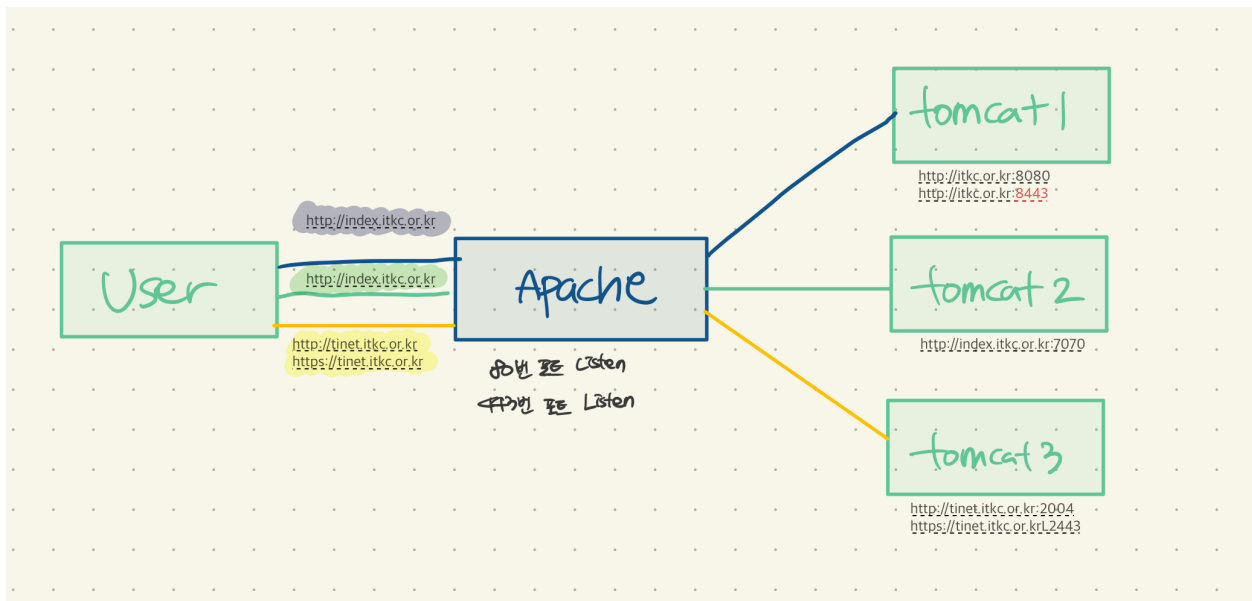


Linux

개요

tomcat1, tomcat2를 운영하던 중 tomcat3를 새롭게 운영해야하는 상황이 발생했다.
tomcat2는 ssl을 사용하지 않아 문제가 되지 않았지만 새롭게 운영할 tomcat3, tomcat1은
ssl을 사용하고 있었기 때문에 포트가 같아서 포트 충돌 문제가 발생했었다.

수행작업



- 서비스 중인 tomcat1의 itkc.or.kr의 https포트를 8443으로 변경하고 새롭게 서비스하는 tomcat3의 https를 2443으로 설정해 443으로 들어올때 tomcat1, tomcat3이 충돌이 생기지 않도록 서버 설정

작업 리뷰

- 서버의 구조를 파악하기 위해 서버 설정정보를 확인 후 도식화했다. 그 후에 어떻게 작업해야 이론적으로 포트충돌이 일어나지 않는지 파악했다. 이때 당시 회사에 테스트 서버가 구성돼 있지 않았고 실제로 배포해보며 작업을 확인하고 되돌리는 위험한 작업을 수행했었다. 실제로 배포하고 충돌이 일어나기도 했었다. 이때 이후로 테스트 서버의 중요함에 대해서 알게되었다.
- Apache설정하나에 여러가지 서버가 묶여있었다. Apache가 다운되면 서로 연관관계가 없는 모든 서버가 다운되는 현상이 발생했고, 이를 계기로 서로 다른 서버에 영향이 가지 않도록 구성해야겠다는 생각이 들었다.

육군사관학교 FTP서버 이전

사용기술



Linux

개요

redHat2 버전에서 Redhat6 버전으로 FTP서버를 옮기는 과정 수행 도중 파일의 한글이름이 모두 깨지는 현상이 발생했었다.

수행작업

- 윈도우 서버를 경유해 인코딩이 euc-kr에서 cp949로 변경되도록 설정하고 옮긴 파일을 윈도우에서 utf-8로 인코딩 후 작업 수행
- 주요정보통신기반시설 기술점 취약점 분석 평가 상세 가이드를 기반으로 리눅스 보안서버 점검 팀원에게 명령어, 옵션 변경방법을 알려줌

리눅스 명령어를 잘 몰라 힘들어하던 팀원을 위해 어떤 명령어를 써야하는지 정리해서 알려줬다.

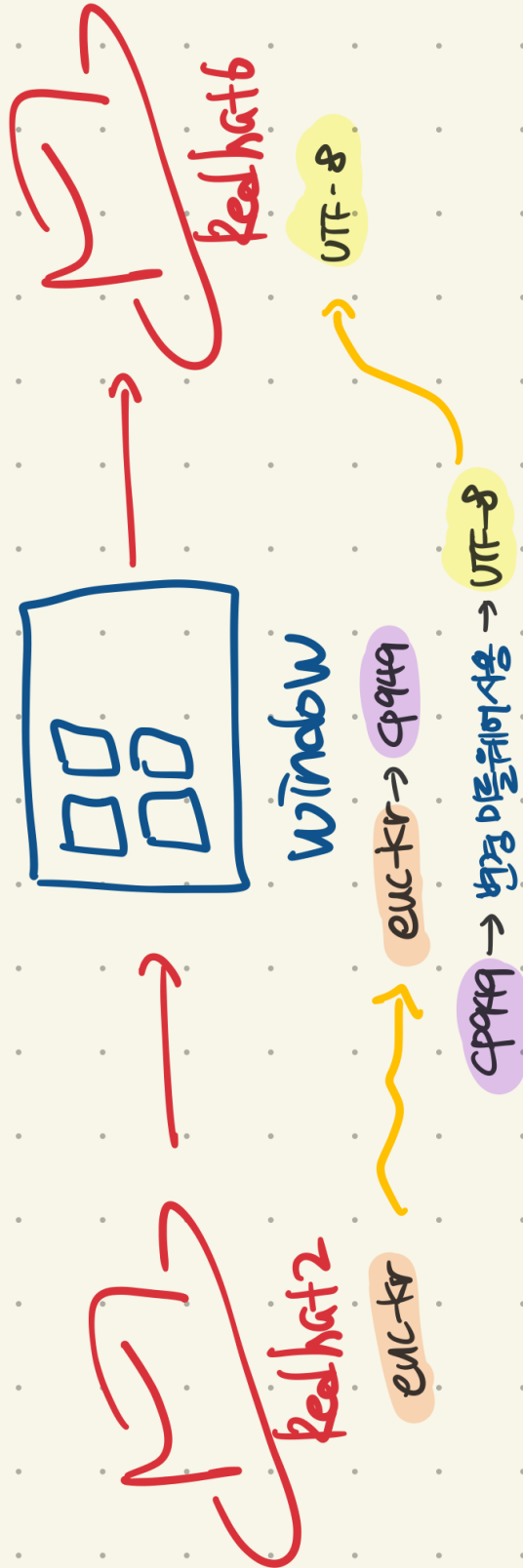
- 윈도우 서버 보안 설정 지원

ftp서비스 중지, telnet서비스 중지 수행하고 방법을 고객에게 문서로 만들어 전달했다.

작업 리뷰

- 옮겨야하는 파일의 양이 적었고 단순 텍스트파일이 대부분이었기 때문에 쉬운 작업이라고 생각했지만, 오래된 버전에서 최신버전으로 데이터를 옮길때 파일의 호환성을 생각해지 못했었다. 다른 서버로 파일을 옮기는 작업을 할 때에는 호환성, 일관성을 지켜야한다는 것을 깨달았다.

자료



종합정보망 테스트케이스 기획 설계

개요

웹 페이지의 기능이 정상적으로 작동하는지 알기위한 테스트케이스 기획 및 설계를 맡았었다.

사용기술



엑셀



한글

수행작업

- **테스트케이스 계획**
테스트 방향과 기준을 잡아주는 테스트 정책서, 테스트할 항목을 정리한 테스트 계획서, 테스트 기법을 설명한 테스트 명세서의 양식을 구성하고 작성했다.
- **테스트케이스 설계**
계획에서 작성한 문서를 기반으로 홈페이지 기능에 대한 테스트케이스 설계서를 작성했다.
- **테스트케이스 수행**
테스트케이스 설계서를 기반으로 테스트케이스를 수행하고, 오류가 나타난 기능은 인시던트 보고서를 작성하여 전달해 개발자가 빠른 시간안에 수정할 수 있도록 했다.
주요정보통신기반 취약점 가이드를 기반으로 웹 서비스의 취약점을 점검하고 보고서를 작성했다. 취약점이 발견된 사항은 인시던트 보고서를 작성해 개발자가 조치를 취할 수 있도록 했다.
- **테스트케이스 완료 단계**
앞서 나온 보고서들을 한 데 보기 좋게 정리한 총 보고서를 작성했다.

작업 리뷰

- 웹 테스트를 위해 직접 테스트 케이스를 직접 구성하고 작성해보았다. 실제로 계획에 맞춰서 테스트케이스를 수행하니 예상하지 못했던 오류들을 찾을 수 있었고 인시던트 보고서를 통해 개발팀에 알려 오류를 빠르게 수정할 수 있었다.

자료

내용	No.	3	TC번호	TC-B2-LS-10
	기능	사업페이지 리스트 정렬기능		
	오류 사항	정렬 결과가 선택한 리스트에서 정렬이 되어야 하는데 총 리스트의 정렬이 나타난다.		
	테스터	유예지 연구원	날짜	19/5/14
	비고			
	수정 완료 여부	완료됨 <input type="checkbox"/> 완료되지 않음 <input checked="" type="checkbox"/>		
	No.	4	TC번호	TC-T-S-1
	기능	번역서페이지 검색결과 하이라이팅 기능		
	오류 사항	번역서 통합검색에서 검색을 수행하면 하이라이팅 되는 게시글이 있고 되지 않는 게시글이 있다.		
	테스터	유예지 연구원	날짜	19/5/14
	비고			
	수정 완료 여부	완료됨 <input checked="" type="checkbox"/> 완료되지 않음 <input type="checkbox"/>		

[인시던트 보고서의 양식]

XI(상)	7. XPath 인젝션	
점검내용	웹페이지 내 XPath 쿼리 공격 가능성 점검	
점검목적	XPath 쿼리에 대한 적절한 필터링을 적용하여 웹사이트의 로직 손상 및 특정 데이터 추출을 차단하기 위함	
보안위협	해당 취약점이 존재할 경우 프로그래머가 의도하지 않았던 문자열을 전달하여 쿼리문의 의미를 왜곡시키거나 그 구조를 변경하고 임의의 쿼리를 실행하여 인가되지 않는 데이터를 열람할 수 있으므로 적절한 필터링 로직 구현이 필요함	
참고		
대상	소스코드, 웹 방화벽	
판단기준	양호	쿼리 입력 값에 대해 검증이 이루어지는 경우
	취약	쿼리 입력 값에 대해 검증이 이루어지지 않는 경우
조치방법	쿼리 입력값에 대해 검증 로직 추가 구현	
이상 여부	이상 있음 <input checked="" type="checkbox"/> 이상 없음 <input type="checkbox"/>	
비고		
점검 및 조치 사례		
○ 점검 방법		
1.[and'a='a, 'and'a='b], [and 1=1, and 1=2]의 세트의 값을 각각 삽입하여 쿼리의 참, 거짓에 따라 반환되는 페이지가 다른지 확인		
※로그인 창은 특수문자를 입력하도록 허용하지 않아 메인페이지 검색창에서 진행		
[and'a='a, 'and'a='b]를 넣었을 때는 400에러페이지가 발생한다.		
<h2>400 Bad Request</h2>		
○해결방안		
검색어의 []와 '를 사용하지 못하도록 한다.		
※ 검증할 문자열 : [], '		

[주요정보통신기반 취약점 가이드 기반 취약점 보고서]