1. s3 Api를 사용하기 위해
   1. 구글에서 mvnrepository검색
   2. s3검색후 제일 위에것(aws-java-sdk-s3) 최신버전으로 클릭한 후, 텍스트 긁어옴
   3. spirng 웹 페이지에서 src/main/java -> target -> pom.xml에 제일 아래 <!--test--> 항 아래에 복사 붙혀넣기 해준다.
2. 1과 똑같이 commons-id, commons-fileupload 도 똑같이 해준다.
   1. s3 bucket에 파일을 업로드 하기위한 api
3. 파일 업로드 만들기(강사님 타이핑)
   1. s3에 업로드 틀 만들기
   2. AWSController에서 전달 받을 메소드 구현
   3. pom.xml의 test(junit)아래 부분에 라이브러리 등록
      1. mvnrepository에서 commons io 검색 후 등록
      2. commons fileupload 검색후 등록
   4. root-context.xml에 업로드 사이즈 지정
   5. interface에 메소드 정의
   6. 클래스에 메소드 구현

정리

1. 버킷 만들기
   1. AWSController에 createBucket 메소드 생성
      1. 메소드 내에 사용할 메소드 만들기
         1. iAWSServ.CreateBucket(“infiscap-bucket2”);
      2. 인터페이스에 CreateBucket이 없기 때문에 에러 발생
         1. 인터페이스에 public void드형으로 생성해줌 (빨간줄에 마우스대서 자동생성)
   2. serviceImpl에서 CreateBucket구현
      1. s3 정보 획득(gets3())
      2. s3에 있는 createBucket 함수를 이용하여 버킷 생성
      3. 중복 문제로 doesBucketExistV2 활용
2. 버킷 목록 읽기
   1. AWSController에 bucketList 메소드 생성
      1. 메소드 내에서 사용할 메소드 만들기
         1. 인터페이스 -> public List<Bucket> getBucketList();
         2. 컨트롤러에서 List<Bucket> bucketLst = iAWSServ.getBucketList();
         3. List형으로 bucket을 저장하여 s3.jsp에 전달
   2. serviceImpl에서 getBucketList구현
      1. s3 정보 얻기
      2. s3에 있는 listBuckets()함수를 이용하여 목록 얻기
3. 버킷 만들기2
   1. 단순하게 사이트 이동이 아닌 데이터와 함께 이동하기 위해 form태그 사용
      1. AWSSController에 createBucketParam 메소드 생성
         1. 파라미터 전달 받기
            1. @RequestParam("bucketName") String bucketName로 bucketName 텍스트 전달 받음
   2. ServiceImpl부분은 어려운 부분이 아님으로 분석해 볼것
4. 파일 업로드
   1. s3에 업로드 틀 만들기
   2. AWSController에서 전달 받을 메소드 구현
   3. pom.xml의 test(junit)아래 부분에 라이브러리 등록
      1. mvnrepository에서 commons io 검색 후 등록
      2. commons fileupload 검색후 등록
   4. root-context.xml에 업로드 사이즈 지정
   5. interface에 메소드 정의
   6. 클래스에 메소드 구현