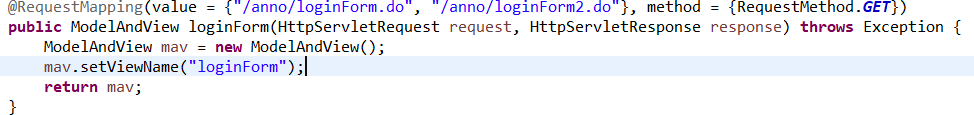
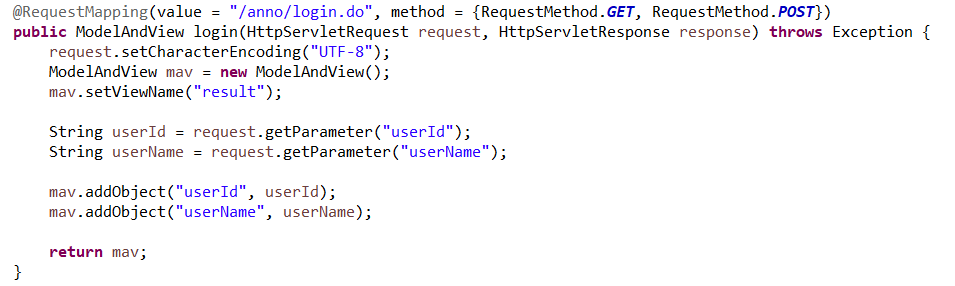
10. Spring Annotation

@RequestMapping(value=” ” ,method=””)

value : 요청받을 url을 설정

method : 어떤 요청을 받을지 정의( GET, POST, PUT, DELETE) 등



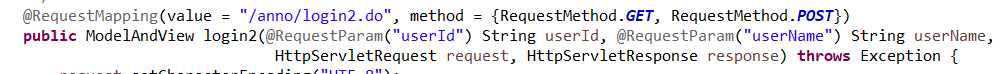
****

value와 method 안에 복수를 설정 할 수 있음

@RequestParam

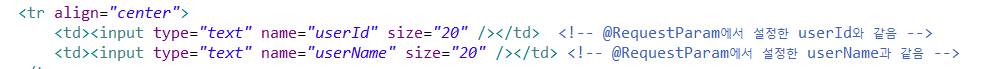
HttpServletRequest.getParameter(“”)와 같은 역할

@RequestParam(“가져올 데이터 이름 ”) 데이터타입 담을 변수이름



가져올 데이터 이름과 담을 변수 이름이 같다면 @RequestParam() 가능





loginForm.jsp에서 보낼 parameter명과 controller에 담을 변수명이 같다

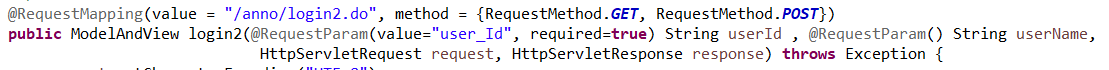
@RequestParam(value=””, required= true,false, defaultValue=””)

required= true이면 지정한 parameter를 무조건 받아야 함, true는 디폴트값

required= false이면 값이 없어도 오류없이 진행

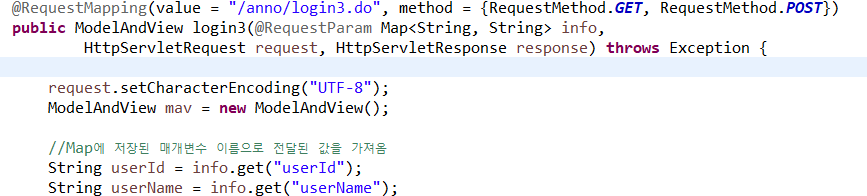
@RequestParam()의 경우도 required=false

defaultValue=”” 통해 받을 값이 없을 경우의 값 지정

 required=true가 true이기 떄문에 user\_Id라는 parameter가 꼭

필요하지만 jsp에서 보낼 parameter명은 userId이기 때문에 400오류

전달받을 값이 많을 경우 Map을 이용하여 전달받은 매개변수를 <key,value>값으로 지정

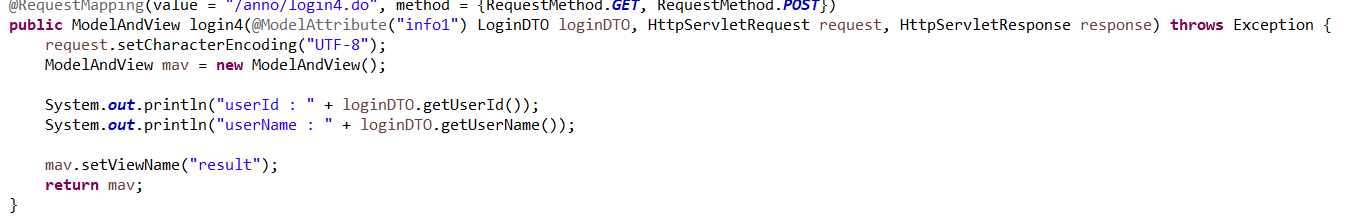


전달받은 parameter를 info에 저장, get(“key”)메서드를 통해 가져옴

@ModelAttribute()

parameter의 값들을 클래스의 형태로 저장

값을 저장할 클래스는 setter, getter 매서드가 있어야 한다



매서드 실행시 LoginDTO 클래스의 loginDTO 객체 생성됨

loginDTO에 parameter들을 바인딩함

addObject없이 loginDTO는 자동으로 MODEL객체에 바인딩되어 뷰단으로 전달됨

Model객체를 직접 생성

Controller에서 View로 데이터를 담아 전달하는 객체

addAttribute(“key”, “value”)의 형태로 뷰에 값을 전달

11. log4j

log4j란

로그 기능을 제공하는 오픈 소스 라이브러리

애플리케이션에서 웹 사이트에 접속한 사용자 정보나 각 클래스의 메서드 호출 시각

등 여러가지 정보를 로그로 출력해서 관리

Maven 프로젝트 생성시 자동으로 log4j 라이브러리가 설치됨

log4j.xml의 태그

<appender>

로그를 어디에 출력할 것인지 출력 위치를 결정(콘솔, 파일, db)

<layout>

어떤 형식으로 출력할지 출력 레이아웃을 결정함

<looger>

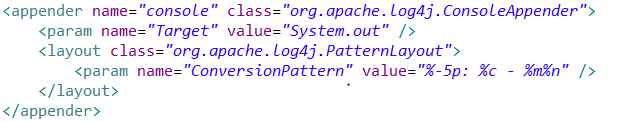
로깅 메세지를 Appender에 전달함

개발자가 로그 레벨을 이용해 로그 출력 여부를 조정할 수 있음

Appender 클래스들

ConsoleAppender

콘솔에 로그 메시지를 출력함



Maven프로젝트 생성 시 추가되는 기본 appender

FileAppender

파일에 로그 메시지를 출력함

RollingFileAppender

파일 크기가 일정 기준을 넘으면 기존 파일을 백업 파일로 바꾸고

처음부터 다시 기록함

DailyRollingFileAppender

설정한 기간 단위로 새파일을 만들어 로그 메시지를 기록함

