

Lab (workshop):
controller의 servlet를 하나로 취합하자.

1 . controller의 각 url을 확인하자.



```
@WebServlet("/ScoreAll")
public class ScoreAll extends HttpServlet { }
```

2. 각 url을 다음과 같이 주석처리한다.
//@WebServlet("/ScoreAll")

3. MyController의 HttpServlet을 상속받은 servlet를 작성한다.

```
@WebServlet({ "/MyController", "/ScoreAll", "/ScoreDelete",
"/ScoreUpdate", "/ScoreFind", "/ScoreInsert" })
public class MyController extends HttpServlet { }
```

4. MyServletController에서 doGet() 메소드와
각각의 Url을 매핑하는 메소드를 추가한다.

```
protected void doGet(HttpServletRequest request, HttpServletResponse
response)
throws ServletException, IOException {
}
```

```
protected void doScoreAll(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException {
}
```

5. 각 기능을 만들어진 메소드에 붙여 넣는다.

```
protected void doScoreAll(HttpServletRequest request,
```

```
HttpServletResponse response)
    throws ServletException, IOException {
    ScoreBiz biz=new ScoreBiz();
    ArrayList<Score> sm= biz.getScoreAll();
    //값을 담아서 ScoreAll.jsp로 배달
    request.setAttribute("all", sm);

    //배달
    RequestDispatcher rd
    =request.getRequestDispatcher("ScoreAll.jsp");
    rd.forward(request, response);
}
```