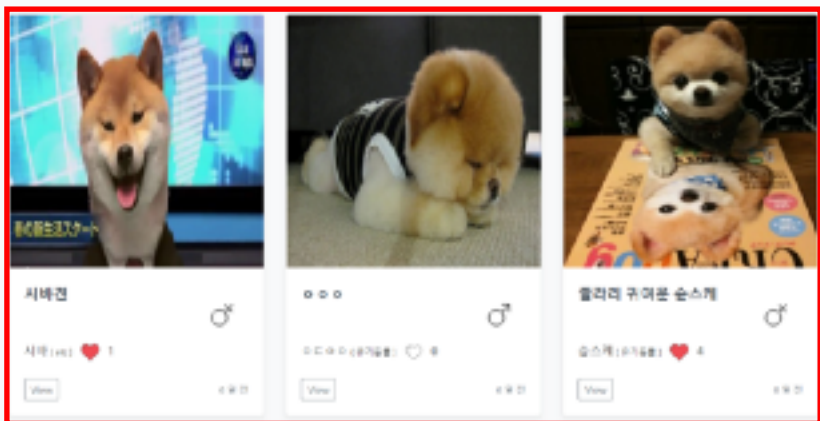


## 2. 리스트 페이지(분양동물)



## Controller

```
if(command.equals("animlist")) {  
    List<AnimalBbsDto> animlist = aniBbsService.getAnimalBbsList();  
    req.setAttribute("animlist", animlist);  
    dispatch("AnimalBbslist.jsp", req, resp);  
}
```

## Service

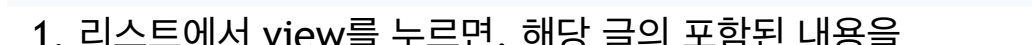
// 777

```
public List<AnimalBbsDto> getAnimalBbsList() {  
    return aniBbsManager.getAnimalBbsList();  
}
```

View

[illegible]

### 3. 디테일 페이지(분양동물)



view를 반환한다. **view**는 **view**를 반환한다.

[illegible]

```
else if(command.equals("detail")) {
```

```
//로그인 성공 후 세션 설정하기
String sseq = req.getParameter("seq");
int seq = Integer.parseInt(sseq);

if(Delegator.checkSession(req, resp)) {
    HttpSession session = req.getSession();
    User userInfo = (User)session.getAttribute("current_user");
    String email = userInfo.getEmail();
    aniBbsService.readCount(seq);

    BookService bookservice = BookService.getInstance();
    List<BookDto> booklist = bookservice.getBookList(seq);
    req.setAttribute("booklist", booklist);
    AnimalCommentService commentService = AnimalCommentService.getInstance();
    List<AnimalCommentDto> comments = commentService.getAllComments(seq);
    AnimalBbsDto aniBbsDto = aniBbsService.detailAnimalBbs(seq);

    boolean isLiked = aniBbsService.Prevent_duplication(userInfo.getSeq(), seq);
    boolean bookS = bookService.checkBook(email, seq);

    req.setAttribute("aniBbsDto", aniBbsDto);
    req.setAttribute("bookS", bookS);
    req.setAttribute("like_count", aniBbsService.getLikeCount(seq));
    req.setAttribute("isLiked", isLiked);
    req.setAttribute("comments", comments);

    dispatch("AnimalBbsdetail.jsp", req, resp);
}
```

## // 읽을거리 글 디테일

```
public AnimalBbsDto detailAnimalBbs(int seq) {  
    return aniBbsManager.detailAnimalBbs(seq);  
}
```

1. 디테일로 들어오기 전 로그인 상태를 체크하여 비 로그인일 시 로그인 페이지로 이동한다.