



Tag 라이브러리를 이용한 Pomodoro Application 화면 수정하기

Table of Contents

- Tag 라이브러리를 이용한 Pomodoro Application 화면 수정하기
- Taglib 생성
- 줄바꿈 Replace

이전 강좌에서 나온 어플리케이션을 살펴보면 아직은 부족한 부분이 많이 있습니다. 현재 강좌에서는 TagLib를 이용하여 BR 태그를 치환하는 줄 바꿈이나 날짜 포맷을 바꾸는 것, 그리고 Deadline의 날짜가 지났는지, 현재 시점인지, 미래인지에 따라서 표시하는 방법을 바꿔 보는것을 확인해보도록 하겠습니다.

먼저, conf 하위의 Bootstrap.groovy 와 도메인 객체를 아래와 같이 변경해 봅시다.

Bootstrap.groovy 전체 소스

```
1 import org.gliderwiki.pomodoro.Tag
2 import org.gliderwiki.pomodoro.Task
3
4 class Bootstrap {
5
6     def init = { servletContext ->
7         def workTag = new Tag(name: "Work").save(failOnError:true)
8         def homeTag = new Tag(name: "Home").save(failOnError:true)
9
10        def task = new Task(
11            summary:"Do 2nd intro screencast",
12            details:"Create the second intro screencast for Grails",
13            deadline:new Date(),
14            status:"Open"
15        )
16        task.addToTag(workTag)
17        task.addToTag(homeTag)
18
19        task.save(failOnError:true)
20
21        task = new Task(
22            summary:"Shopping",
23            details:"Do the grocery shopping - and don't forget the tomatoes!",
24            deadline:new Date() + 7,
25            status:"Open"
26        )
27        task.addToTag(homeTag)
28        task.save(failOnError:true)
29
30        task = new Task(
31            summary:"Finish Grails presentation",
32            details:"The last presentation on Grails & AMQP for SpringOne 2GX needs finishing.",
33            deadline:new Date() - 3,
34            status:"Open"
35        )
36        task.addToTag(workTag)
37        task.save(failOnError:true)
38
39        task = new Task(
40            summary: "Another task",
41            details: ""
42            Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
43            tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
44            quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
45
46            Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu
47            fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in
48            culpa qui officia deserunt mollit anim id est laborum.""",
49            deadline:new Date() + 4,
50            status:"Open"
51        )
52        task.addToTag(workTag)
53        task.save(failOnError: true)
54    }
55    def destroy = {
56    }
57 }
```

변경된 부분은 문단 구분을 테스트 해보기 위해 Another task라고 하는 Task를 하나 추가하였습니다.

Task.groovy 수정

```
1 | nackane org.gliderwiki.pomodoro
```



```

1 package org.gliderwiki.pomodoro
2
3 class Task {
4     String summary
5     String details
6     Date dateCreated
7     Date deadline
8     Long timeSpent = 0L
9     String status
10
11     static hasMany = [tag : Tag]
12
13     static constraints = {
14         summary blank:false, unique:true
15         details blank:false, maxSize: 1000
16         deadline nullable:true
17         timeSpent min:0L
18         status inList:["Open", "Done"]
19     }
20 }

```

details 프로퍼티에 maxSize 속성을 추가하였습니다. maxSize의 속성은 255자 일 경우 textfield 형태로 form이 구성되며, 255자 이상일 경우 html 의 textarea form 으로 생성되게 됩니다.

Bootstrap.groovy 에서 저장되는 text의 길이가 커졌기 때문에 추가한 속성입니다.

마지막으로 TaskController 의 find 구문을 이용한 fetch 부분을 아래와 같이 수정합니다.

```

1 package org.gliderwiki.pomodoro
2
3 class TaskController {
4
5     static scaffold = Task
6
7     def index = {
8         def tasks = Task.findAllByStatus("Open", [sort: "deadline", order:"asc"])
9         def tags = Tag.list(sort:"name", order:"asc")
10
11         return [tasks:tasks, tags:tags]
12     }
13 }

```

전체 데이터를 fetch 해오기 위해서 findAllByStatus 으로 구문을 변경하였습니다.

이제 어플리케이션을 다시 시작해보면 아래와 같은 화면을 확인 할 수 있습니다.

Pomodoro Task Manager

Open Task for user

- New

Finish Grails presentation

The last presentation on Grails & AMQP for SpringOne 2GX needs finishing.

due : 2014-03-11 10:42:51

Created : 2014-03-14 10:42:51

Do 2nd intro screencast

Create the second intro screencast for Grails

due : 2014-03-14 10:42:51

Created : 2014-03-14 10:42:51

Another task

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

due : 2014-03-18 10:42:51

Created : 2014-03-14 10:42:51

Shopping

Do the grocery shopping - and don't forget the tomatoes!

due : 2014-03-21 10:42:51

Created : 2014-03-14 10:42:51

Show

- Open
- Done
- All

Tags

- Home
- Work

due나 Created 부분의 날짜 포맷이 바뀐것은 아래와 같이 _taskCard.gsp를 수정했기 때문입니다.


```

1 <div class="task">
2   <h4>${ task.summary }</h4>
3   <p>${ task.details }</p>
4   <div class="due"> due : <g:formatDate date="${ task.deadline }" format="yyyy-MM-dd hh:mm:ss" /></div>
5   <div class="created">Created : <g:formatDate date="${ task.dateCreated }" format="yyyy-MM-dd hh:mm:ss" /></div>
6 </div>

```

Taglib 생성

이제 Taglib를 이용하여 Deadline 의 시점을 CSS의 색상을 달리 하여 표기해보겠습니다.

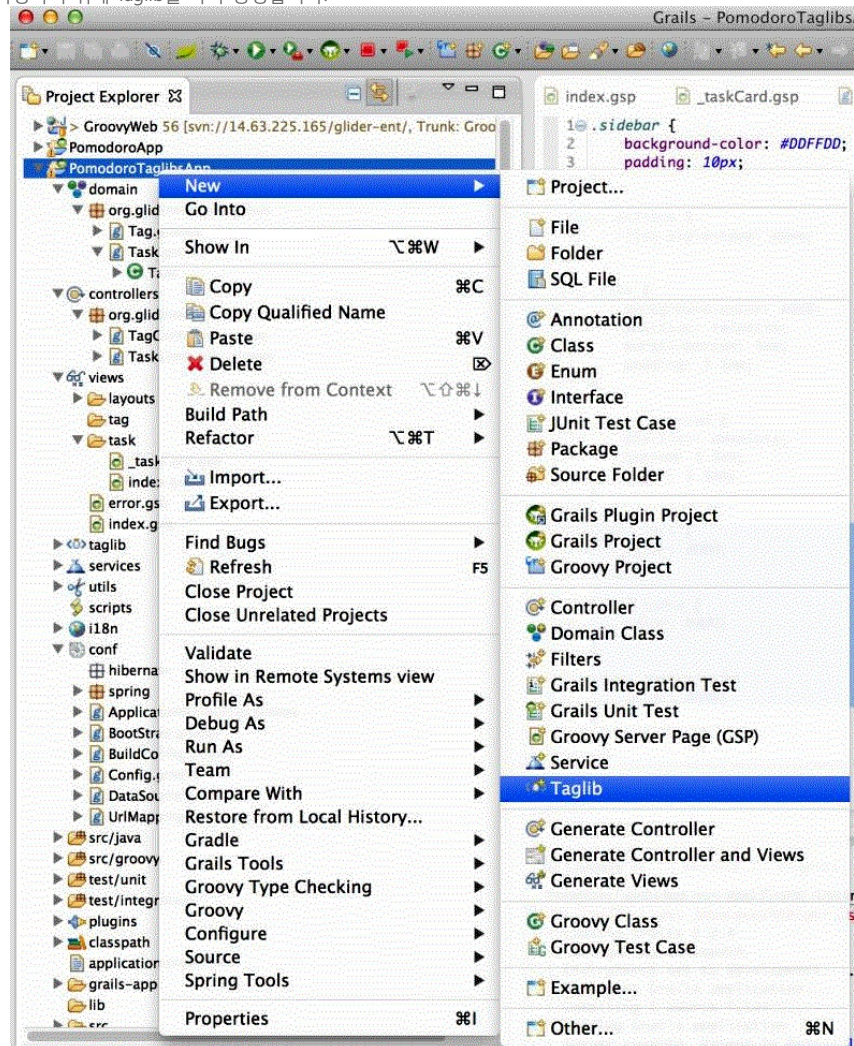
app.css 파일의 하단에 아래와 같이 CSS 구문을 추가합니다.

```

1 .due .past {
2   color: #D00;
3 }
4
5 .due .today {
6   color: #00D;
7 }
8
9 .due .imminent {
10  color: #D90;
11 }

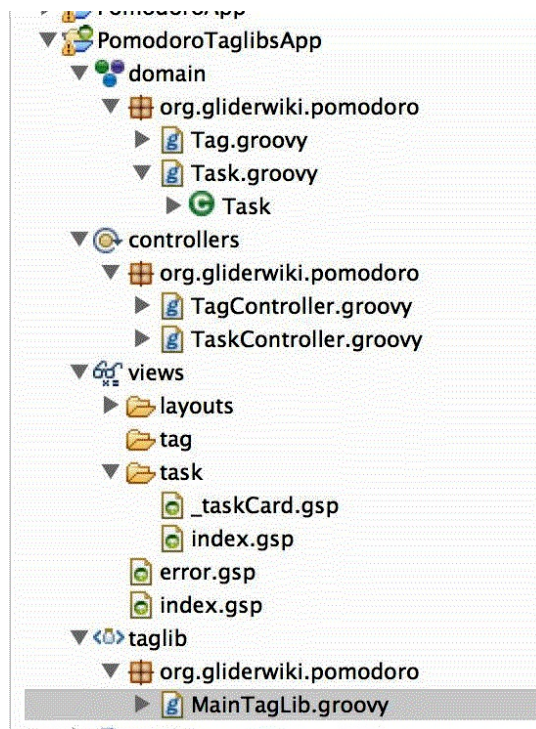
```

또한 Custom Tag를 지정하기 위해 Taglib를 하나 생성합니다.



패키지 경로와 파일명은 org.gliderwiki.pomodoro.Main 으로 입력했습니다.

taglib 디렉토리 밑에 아래와 같이 생성된 것을 확인 할 수 있습니다.



이제 namespace 프리퍼터를 이용하여 pomo 라는 네임스페이스를 지정합니다.
사용자 정의 태그가 많아질 경우 이 네임스페이스를 통해 기능을 지정함으로써 중복되는 혼란을 줄일 수 있습니다.

지금까지 소스는 아래와 같습니다.

```
1 package org.gliderwiki.pomodoro
2
3 class MainTagLib {
4
5     static namespace = "pomo"
6 }
```

여기에 deadline 의 초과 여부에 따른 css 적용을 tagLib를 이용하여 변경해보도록 합니다.

deadline은 오늘 날짜 보다 지났으면 pass 라고 하는 css class를 주고, 현재 날짜 라면 today라는 class를 지정해 주도록 변경합니다.
마지막으로 할일이 5일 이내로 남아있을 경우 imminent 라는 class 를 주도록 하겠습니다.

MainTagLib 에 날짜 파라미터를 받는 클로저를 선언합니다.

```
1 package org.gliderwiki.pomodoro
2
3 class MainTagLib {
4
5     static namespace = "pomo"
6
7     /**
8      * @attr date REQUIRED The date that is to be formatted.
9      */
10    def shortDate = { attrs ->
11        out << g.formatDate(date: attrs.date, format: "yyyy-MM-dd hh:mm:ss")
12    }
13
14    /**
15     * @attr date REQUIRED The date that is to be formatted.
16     */
17    def deadline = { attrs ->
18        def cssClass = ""
19        def today = new Date()
20        def dateWithoutTime = attrs.date
21        today.clearTime()
22        dateWithoutTime.clearTime()
23
24        def diff = dateWithoutTime - today
25        if (diff < 0) {
26            cssClass = "past"
27        }
28        else if (diff == 0) {
29            cssClass = "today"
30        }
31        else if (diff < 5) {
32            cssClass = "imminent"
33        }
34
35        out << "<span class=\"$cssClass\">" << pomo.shortDate(date: attrs.date) << "</span>"
36    }
37 }
```



```
38 |  
39 | }
```

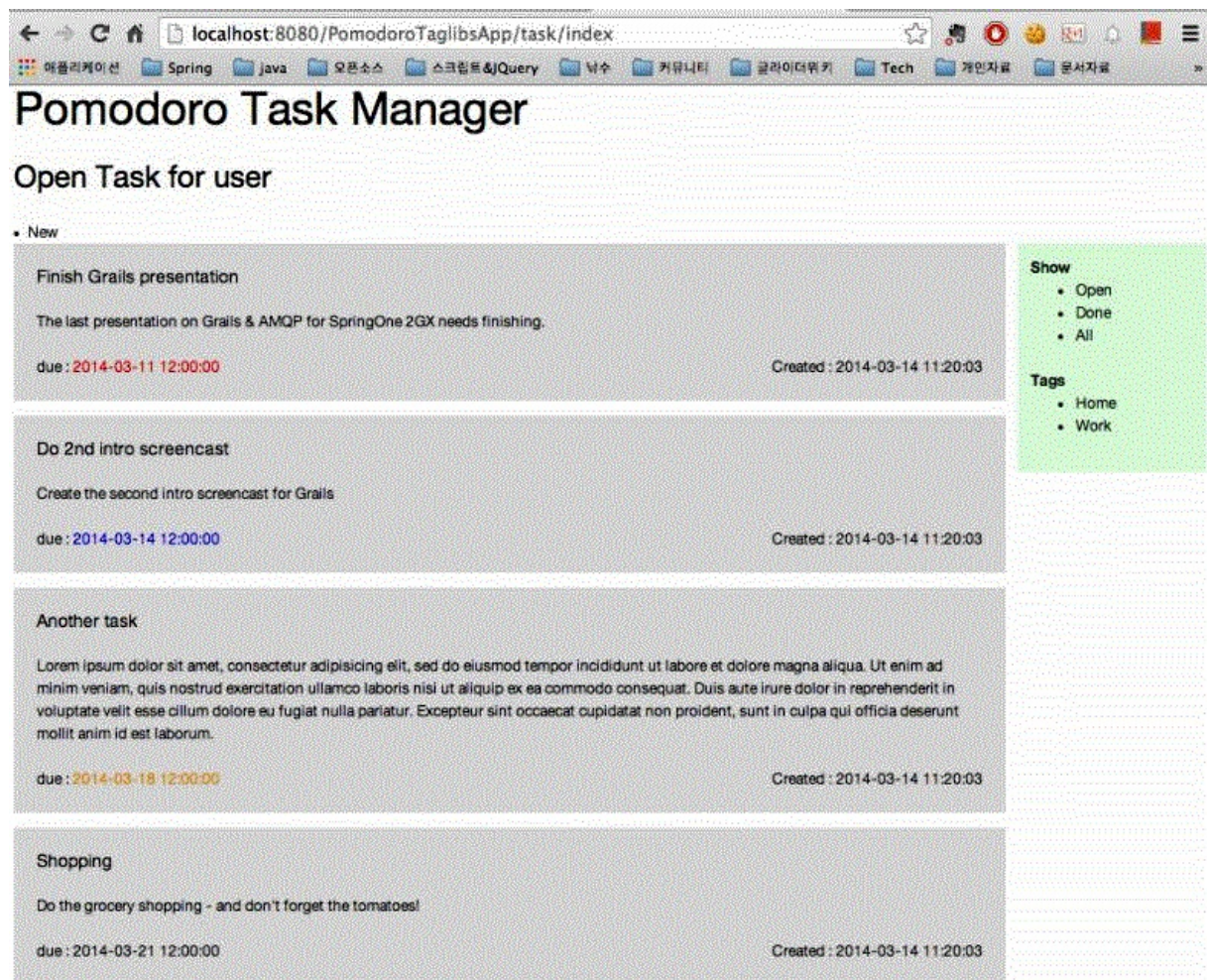
호출을 통해 넘겨온 date 와 현재 날짜를 계산하여 0 이하면 past , 같으면 today, 미래시점이면 imminent로 cssClass를 지정하고 쉬프트 연산자로 태그를 생성해주었습니다.

pomo.shortDate 는 Date format을 맞춰주기 위한 클로저입니다.

task의 index.gsp 에 include 한 _taskCard.gsp 에서 taglib를 호출하도록 소스를 변경해보도록 하겠습니다.

```
1 | <div class="task">  
2 |   <h4>${ task.summary }</h4>  
3 |   <p>${ task.details }</p>  
4 |   <div class="due"> due : <pomo:deadline date="${ task.deadline }" /></div>  
5 |   <div class="created">Created : <g:formatDate date="${ task.dateCreated }" format="yyyy-MM-dd hh:mm:ss" /></div>  
6 | </div>
```

저장 후 어플리케이션을 실행 해보면 아래와 같이 빨간색, 파란색, 오렌지색으로 due 부분이 변경된 것을 확인 할 수 있습니다.



태그 라이브러리의 호출은 네임 스페이스와 클로저로 선언한 메서드를 통해 서버측의 데이터를 받거나 화면 구성을 변경하는 것이 가능합니다.

줄바꿈 Replace

이번 강좌의 마지막으로 문단안의 줄바꿈을 taglib를 이용하여 변환해보도록 하겠습니다.

MainTagLib.groovy 파일의 하단에 아래와 같이 선언해주도록 합니다.

```
1 | package org.gliderwiki.pomodoro  
2 |  
3 | class MainTagLib {  
4 |  
5 |     static namespace = "pomo"  
6 |  
7 |     /**  
8 |      * @attr date REQUIRED The date that is to be formatted.
```

```

9      * @attr date REQUIRED The date that is to be formatted.
10  def shortDate = { attrs ->
11    out << g.formatDate(date: attrs.date, format: "yyyy-MM-dd hh:mm:ss")
12  }
13
14
15  /**
16   * @attr date REQUIRED The date that is to be formatted.
17   */
18  def deadline = { attrs ->
19    def cssClass = ""
20    def today = new Date()
21    def dateWithoutTime = attrs.date
22    today.clearTime()
23    dateWithoutTime.clearTime()
24
25    def diff = dateWithoutTime - today
26    if (diff < 0) {
27      cssClass = "past"
28    }
29    else if (diff == 0) {
30      cssClass = "today"
31    }
32    else if (diff < 5) {
33      cssClass = "imminent"
34    }
35
36    out << "<span class=\"$cssClass\">" << pomo.shortDate(date: attrs.date) << "</span>"
37  }
38
39
40  def textToParagraphs = { attrs, body ->
41    out << "<p>" << body().trim().replaceAll(/(s*n)+/, "<p></p>") << "</p>"
42  }
43 }

```

이 경우 두번째 인자로 들어오는 Body의 화이트스페이스를 Trim 한 후 라인 치환을 패러그래프 태그로 변경하였습니다.

_taskCard.gsp파일을 아래와 같이 변경해서 어플리케이션을 실행해보도록 합니다.

```

1 <div class="task">
2   <h4>${ task.summary }</h4>
3   <div><pomo:textToParagraphs>${ task.details }</pomo:textToParagraphs></div>
4   <div class="due"> due : <pomo:deadline date="${ task.deadline }" /></div>
5   <div class="created">Created : <g:formatDate date="${ task.dateCreated }" format="yyyy-MM-dd hh:mm:ss" /></div>
6 </div>

```


localhost:8080/PomodoroTaglibsApp/task/index

Pomodoro Task Manager

Open Task for user

• New

Finish Grails presentation

The last presentation on Grails & AMQP for SpringOne 2GX needs finishing.

due: 2014-03-11 12:00:00

Created: 2014-03-14 11:35:00

Do 2nd intro screencast

Create the second intro screencast for Grails

due: 2014-03-14 12:00:00

Created: 2014-03-14 11:35:00

Another task

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

due: 2014-03-18 12:00:00

Created: 2014-03-14 11:35:00

Shopping

Do the grocery shopping - and don't forget the tomatoes!

due: 2014-03-21 12:00:00

Created: 2014-03-14 11:35:00

Show

- Open
- Done
- All

Tags

- Home
- Work

문단의 줄바꿈이 적용된 것을 확인 해볼 수 있습니다.