این ماژول شامل یکسری سرویس (service object) و رخداد (event) است تا کاربر بتواند با استفاده از آن‌ها نیازمندی خود را در غالب یک روش کار ثابت (main line) پیاده‌سازی کند . لذا خود به تنهایی شامل هیچ کنترلر و view ی خاصی نمی باشد .

**هدف**

پیاده‌سازی یک رویکرد general ، قابل customize کردن و توسعه پذیر ، بدون ورود به حوزه ی کنترلر و view .

رویکرد توسعه پذیری از طریق افزودن امکانات به سرویس و تعریف پروسه های جدید (تعریف event) میسر است.

Customize کردن (پیاده سازی نیازمندی های پروژه) از طریق افزودن listener ها به پروسه ها و تعریف کنترلر ها صورت می پذیرد.

**تعاریف :**

feedback

فرم ارزیابی .

feedback subject

به آنچه که فرم ارزیابی در مورد آن ایجاد می‌شود . به عنوان مثال در مورد فرم ارزیابی ارزی ،‌ «ارزی» بودن صرفاً یک عنوان است و موارد بررسی «شعب» موضوع حساب می‌شوند .

Submisssion

هر ثبت فرم را یک submission می نامند . هر مرتبه که فرم submit می شود.

State

وضعیت یک submisison را توصیف توصیف می‌کند .

Submitter

هر فرد که فرم را submit می‌کند .

**روش کار :**

روش انجام کار در چند سناریو متفاوت مطرح شده و به ترتیب پیچیدگی کار افزوده می شود.

ایجاد یک فرم ارزیابی خالی (بدون موضوع و بدون سوال)

۱- در جدول baftfeedback\_feedback یک رکورد که فرم را تعریف می‌کند افزوده میشود.

مقادیر name , label , ref\_group\_id کافی است.

۲- در جدول baftfeedback\_question\_group یک رکورد تعریف میگردد . این رکورد به عنوان یک fieldset در برگیرنده ی همه **گروه** سوالات و نمایندگی فرم ارزیابی را خواهد داشت. اان را در طول مستند "گروه اصلی" می نامیم.

۳- در جدول baftfeedback\_feedback\_version یک رکورد به عنوان تعریف کننده ی نسخه فرم افزوده شود.

(مراجعه شود به قسمت مربوطه در توضیحات مدل (

۴- از آدرس زیر می‌توان اقدام به مشاهده فرم و ایجاد submission کرد.

Feedback subject → http://[domain]/feedback/submission/[feedback id]

با فراخوانی آدرس فوق سیستم به طور پیشفرض ابتدا سعی می‌کند subject را نمایش داده تا کاربر بتواند موضوع خود را انتخاب کند و با زدن دکمه ثبت یک submission برای آن موضوع را آغازکند.

(مراجعه شود به قسمت مربوط به submission در توضیحات مدل )

در این مثال چون موضوعی وجود ندارد سیستم خود یک submission ایجاد کرده و به صفحه ی ویرایش ان submission جهت ادامه ی کار redirect می‌کند .

Edit submission URL → http://[domain]/feedback/submission/[feedback id]/[submission id]

افزودن موضوع به فرم ارزیابی

در ادامه ی موضوع قبل

۱- در جدول baftfeedback\_feedback\_subject یک رکورد اضافه شود .

مقادیر name , ref\_baftfeedback\_feedback\_id , ref\_fieldset کافی می باشند.

) جهت تست برای فیلد ref\_fieldset از مقدار feedbackExtendBranch استفاده نمایید (

)مراجعه شود به قسمت مربوطه در توضیحات مدل جهت توضیحات فیدها (

۲-از آدرس زیر می‌توان اقدام به مشاهده فرم و ایجاد submission کرد.

Feedback subject → http://[domain]/feedback/submission/[feedback id]

افزودن سؤال و گروه سؤال

در ادامه ی موضوع قبل

۱- در جدولbaftfeedback\_question\_group گروه اضافه شود و فییلد ref\_group\_parent به شناسه ی گروه اصلی ارجاع شود. (عنوان گروه در اینجا تعیین می گردد)

۲- افزودن سوال (تعریف سؤال و یا سؤال خاص)در جدول baftfeedback\_question انجام میشود .

۳- در جدول baftfeedback\_question\_group\_questions به گروه مربوطه متصل گردد.

توضیحات

افزودن سؤال مستقیماً زیر گروه اصلی که نمایندگی فرم ارزیابی را دارد امکان‌پذیر نمی‌باشد

حداقل یک گروه به عنوان در برگیرنده سوال/سوالات باید وجود داشته باشد. و زیر مجموعه ی گروه اصلی قرار بگیرد.

چنانچه نیاز است که سؤالات در غالب گروه سؤال نمایش داده نشوند کافیست که برای آن گروه label تعیین نشود

تغییر وضعیت فرم :

وضعیت فرم‌ها در حال حاضر به طور موقت در جدول baftfeedback\_feedback\_submission\_state نگهداری می‌شوند .

۱- برای تغییر دادن وضعیت ، از فراخوانی زیر استفاده می شود.

$submissionModel=$this->getServiceLocator()->get('BaftFeedback\Model\submission');

$data=[ any data assoc array ];

$newState=$statesEnum::*BR\_CONFIRM; //any state number*

$submissionModel->setState( $submissionEntity , *$newState* , $data );

به عنوان مثال

وضعیت‌هایی که برای فرم ارزی در نظر گرفته شده است .

|  |  |
| --- | --- |
| inspFeedback\Model\StatesEnum  ::CREATION | زمانی که submission ایجاد می شود |
| inspFeedback\Model\StatesEnum  ::BR\_CONFIRM | زمانی که تأیید شعبه می شود |
| inspFeedback\Model\StatesEnum  ::SAR\_CONFIRM | زمانی که تأیید سرپرستی می شود. |
| inspFeedback\Model\StatesEnum  ::ARZI\_HEAD\_CONFIRM | زمانی که کاربر اداره حارجه مشاهده می کند |
| inspFeedback\Model\StatesEnum  ::COMPLETION | فرم تکمیل و به اتمام رسده است.  تمامی سؤالات رفع شده‌اند و همه ی تأیید ها را گرفته است. |
|  |  |

که با فراخوانی توابعی در مدل مربوطه مثل تابع زیر ، وضعیت‌ها کنترل می شوند

inspFeedback\Model\checkList->setBranchConfirm

**Service (baftfeedback/service/[serviceName]Service)**

در لایه ی domain ایفای نقش می‌کنند (در DDD terminology) بنابراین ترکیبی از داده-نیازمندی-رفتار هستند که در راستای رفع پیچیدگی های زیرین پیاده شده‌اند .

**evnet (baftfeedback/Listener/[Listener Name]Listener)**

نقش aspect را ایفا می‌کنند ، یکسری موضوعات عمومی یا پرکاربرد را در بر می‌گیرند (مثل CRUD) . زمانی کاربرد دارند که نیاز دارید کاری را انجام دهید که فرایندی پیچیده و مشخص دارد و در عین حال می‌خواهید که به سادگی توسعه پذیر باشد.

مثلاً در اینجا create submissoin (ذخیره اطلاعات فرم) یک event است چرا که دارای یک روال مشخص - ثابت – پیچیده- پرکاربرد است و باید با نیازمندی های متفاوت منطبق شود بدون آنکه روال اصلی آن (main business line) دچار تغییرات شود .

انواع رخداد به شرح ذیل می‌باشد :

(به صورت constant ها در کلاس BaftFeedback\Event)

|  |  |
| --- | --- |
| ::EVENT\_READ\_FEEDBACK | این aspectکار خواندن feedback از جدول را انجام می‌دهد |
| ::EVENT\_READ\_SUBMISSION | خواندن submission از جداول مرتبط |
| ::EVENT\_CREATE\_SUBMISSION\_PRE | قبل از آنکه submission ایجاد شود |
| ::EVENT\_CREATE\_SUBMISSION | بعد از آنکه submission ایجاد شد |
| ::EVENT\_UPDATE\_SUBMISSION\_PRE  ::EVENT\_UPDATE\_SUBMISSION | قبل و بعد از ویرایش شدن submission |
| ::EVENT\_CHANGE\_STATE | depricated |

**Evnet Object**

یک carrier object است که در هنگام اجرای یک رخداد در میان listener ها دست به دست شده و مقادیر را میان آن‌ها حمل می‌کند .

در بر گیرنده ی مقادیر زیر است :

(مقادیر از طریق getter/setter در دسترس می باشند)

|  |  |
| --- | --- |
| protected $feedback | Feedback entity object |
| protected $Submission | Submission entity object |
| protected $submissionData | ArrayCollection object contain collection of submissionData entity object |
| protected $subject | Subject entity object.  Subject of submission. |
| protected $submitter | ArrayCollection object contain collection of submitter entity object .  Submitters of submission. |
| protected $feedbackVersion | FeedbcakVersion entity object |
| private $parameters=[  'exceptions',  'subject\_data',  'questions\_data',  'submitter\_data',  'submission\_start\_time',  'submission\_expire\_time',  ]; | These parameters accessible by  $event->getParam(“paramName”,defaultValue);  “Exceptions” : a place to save exceptions duration of event.  “subject\_data” : post data contain subject form data.  “questions\_data”: post data contain question form data.  “submitter\_data” : custome data contain submitter data . Like sessionId , userId , IP , ...  “submission\_start\_time” : unixtime . Time of start  “submission\_expire\_time” : unixtime . Time of expire. |
| **protected** $calledListners=[];  **public function** pushCalledListener($listenerName)  **public function** getCalledListeners() | A place to save which listeners called till now.  Save executed listener name under triggered eventName . eg. : save “checkUserAcl” listener name under “create\_feedback” event name |
| **public function** hasException() | Is any exception occurred duration event. |
| **public function** addException($exception)  **public function** getExceptions() | Save exception to event.  Read exceptions . |
| **public function** init($serviceLocator,$feedback,$submission=**null**) | Clear/reset event and reInitialize it .  Need ServiceLocator Object , feedback Entity and Obtional Submission. |
|  |  |
|  |  |
|  |  |
|  |  |

به عنوان مثال

می‌خواهیم قبل از ذخیره شدن داده‌های submission به هنگام create ، داده‌ها را بنا به نیازمندی خود ویرایش کنیم . در اینجا می‌خواهیم آن فیلدهایی از فرم که نامشان rdo1 هستند و مقدار A دارند حذف شوند .

۱- افزودن listener (در اینجا داخل متد dispatch در کنترلر ، listener را افزوده ایم . زیرا قبل از هر action فراخوانی می‌شود )

// attach listener 'onEditSubmissionPre\_saveData'

**public function** dispatch(Request $request, Response $response = **null**) {

$this->getEventManager ()->getSharedManager ()->attach ( [

'BaftFeedback',

arziFeedbackController::*class*,

'BaftFeedback\Service\submissionService'

], FeedbackEvent::*EVENT\_UPDATE\_SUBMISSION\_PRE*, [

$this,

'onEditSubmissionPre\_saveData'

] );

**parent**::*dispatch* ( $request, $response );

}

// listener

**public function** onEditSubmissionPre\_saveData($event) {

$submission = $event->getSubmission ();

$feedbackData = [ ];

**foreach** ( $event->getSubmissionData () **as** $row ) {

$groupId = $row ['question\_group\_id'];

$questionId = $row ['question\_id'];

$fieldName = $row ['question\_field\_name'];

$fieldValue = $row ['field\_value'];

**if** (**empty** ( $fieldValue )) **continue**;

// prevent save desc0

**if** (strcasecmp ( $fieldName, 'rdo1' ) != 0) **continue**;

**if** (strcasecmp ( $fieldValue, 'A' ) != 0) **continue**;

}

$feedbackData [] = $row;

// save in event

$event->setSubmissionData ( $feedbackData );

**return true**;

}

**entitie:**

feedback Entity:

یک موجودیت مستقل است و به فرم ارزیابی اطلاق می‌شود .

ماهیت توصیفی دارد . بدان معنا که از اطلاعات آن جهت ایجاد یکسری موجودیت دیگر استفاده می‌شود .

جهت پیاده سازی به یک فیلدست در جدول baftfeedback\_question\_group متصل می شود .

|  |  |
| --- | --- |
| ref\_baftfeedback\_question\_group\_id | ارجاع به گروه سوال.  این «گروه اصلی» نقش فرم را بازی میکند .  Parent برای بقیه ی گروه ها |
|  |  |
|  |  |
|  |  |

baftfeedback\_feedback\_subjet Entity

موضوع فرم به ان چیزی اطلاق می شود که موجب دسته بندی شدن submission ها می شود. به عنوان مثال «ارزی» برای فرم ارزیابی یک موضوع نیست بلکه صرفا یک label است در حالی که «شعبه» که فرم برای انها پر می شود به عنوان موضوع فرم ارزیابی انتخاب می شوند .

موضوع, در پیاده سازی ارجاعی به یک کلاس فرم است .

|  |  |
| --- | --- |
| **ref\_baftfeedback\_feedback\_id** | ارجاع به feedback مربوطه |
| name | نامی منحصر به فرد باشد . به عنوان نام فرم در فرایند ها و نمایش استفاده می شود . |
| ref\_fieldset | نام کلاس فرمی است که به عنوان subject در پروسه ظاهر می شود . |

baftfeedback\_feedback\_subject\_data

اظلاعات مربوط به موضوع هر submission را نگهداری میکند .

روش نگهداری به این صورت است که اطلاعات هر فیلد از« فرم مربوط به subject» در یک رکورد نگهداری می شوند .

نام فیلد به صورت [fieldsetName]/[filedsetName]/…/[fieldName]