

Support Core features

Integrate social network core features

Core Social Network features include features: Activity Feed, Comment, Like, Share, Report, Feature, Sponsor...

The minimal requirements for that functions to work, resource listing and detail API need to respond **Extra permission** and **Feed Param** schema.

- **Extra Permission Schema:** Control permission of current User is allowed or not with specific action
- **Feed Param Schema:** Control how the Feed, Comment, Like features map with the system

The following image is an example response of PostResource

```
"extra": {
  "can_view": true,
  "can_like": true,
  "can_share": true,
  "can_delete": true,
  "can_report": false,
  "can_add": true,
  "can_edit": true,
  "can_comment": true,
  "can_publish": false,
  "can_feature": true,
  "can_approve": true,
  "can_sponsor": true,
  "can_sponsor_in_feed": false,
  "can_purchase_sponsor": true
},
"self": null,
"links": {<img alt="back arrow icon" data-bbox="215 555 245 568"/>},
"feed_param": {
  "item_id": 1,
  "comment_type_id": "post",
  "total_comment": 2,
  "like_type_id": "post",
  "total_like": 1,
  "feed_title": "Example post 3",
  "feed_link": "http://localhost:7788/post/1/example-post-3/",
  "feed_is_liked": true,
  "feed_is_friend": false,
  "report_module": "post"
}
```

Integrate Activity Feeds

Activity feed on Native Mobile App is controlled by **Feed API**

- API: /feed
- Method: GET

Requirement:

- Activity Feed feature of the Item works on the website
- Feed API response "embed_object" with minimal following fields
 - resource_name
 - module_name
 - id
 - title

```
"embed_object": {
  "resource_name": "post",
  "module_name": "post",
  "title": "Post entry in the pages",
  "description": "It is a long established fact that a reader will be distracted by the readable content of a
    page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal
    distribution of letters, as opposed to using 'Content here, co..'",
  "image": null,
  "categories": [
    {
      "id": 5,
      "name": "Health"
    },
    {
      "id": 6,
      "name": "Recreation"
    },
    {
      "id": 7,
      "name": "Shopping"
    },
    {
      "id": 8,
      "name": "Society"
    },
    {
      "id": 9,
      "name": "Sports"
    }
  ],
  "id": 3
},
```

Control Main Activity Feed Embed response

By default, if resource name and feed type ID stored in the database is the same, we just need to implement **ActivityFeedInterface** interface of related API Service to control the output.

The example below:

PostApi.php

```
<?php
class PostApi extends AbstractResourceApi implements ActivityFeedInterface, MobileAppSettingInterface
{
    /**...*/
    /**
     * Get for display on activity feed
     * @param array $feed
     * @param array $item detail data from database
     * @return array
     */
    public function getFeedDisplay($feed, $item)
    {
        $categoryCond = [
            'post_id' => []
        ];
        /** @var PostCategoryApi $categoryReducer */
        $categoryReducer = $this->getCategoryApi();
        $categoryCond['post_id'][] = $item['post_id'];
        $categoryReducer->reduceFetchAll($categoryCond);
        $item['categories'] = $categoryReducer->reduceQuery([
            'post_id' => $item['post_id']
        ]);
        return $this->processRow($item)
            ->toArray(['resource_name', 'module_name', 'id', 'title', 'categories', 'description', 'image']);
    }
}
```

Control other activity feed type

If the feed type ID cannot map with any resource API service. We can create a custom feed type to control the output.

Steps:

1. Create Custom Feed class extends from Apps\Core_MobileApi\Api\Resource\FeedEmbed\FeedEmbed
2. Implement toArray() feature
3. Update Hook "mobile_api_routing_registration.php" to register to map the new Custom Feed

Example implement of custom feed embed

CustomFeedTypeEmbed.php

```
<?php
namespace Apps\Posts\Api\Feed;
class CustomFeedEmbed extends FeedEmbed
{
    public function toArray()
    {
        $resource = \Phpfox::getService('mobile.some_resource_api')
            ->loadResourceById($this->feedData['item_id'], true);
        if ($resource) {
            return $resource->getFeedDisplay();
        }
        return [];
    }
}
```

Modify hook "mobile_api_routing_registration.php" and add the following code to map feed type ID stored in database with new custom feed handler

mobile_api_routing_registration.php

```
/*...*/
Phpfox::getService('mobile.helper.feedPresentation')
    ->addEmbedTypes('post_special_feed_type', \Apps\Posts\Api\Feed\CustomFeedEmbed::class);

/**
 * If API resource name is different with Phpfox app's alias.
 * we need add map resource_name (in this case is "post") with app_alias as following example
 */
$this->specialModules['post'] = 'app_alias_name';
```

Integrate Like, Comment features

If like, comment features are working on the website and you have implemented resource response same as above, the features will work.

If your resource name and comment / like type in PHPFOX system are different, you can override in Resource

PostResource.php

```
<?php
/*...*/
class PostResource extends ResourceBase
{
    public function getCommentTypeId()
    {
        return "post";
    }
    public function getLikeTypeId()
    {
        return "post";
    }
}
```