

Layout Components

Control Screen layout via Setting APIs

The native mobile apps allow us to control screen layout setting via 2 Core APIs

1. **Core App Setting API:** This API collects all settings of each app to build the screen layout
2. **Core Site Setting API:** This API allows the app to define settings for each mobile screen. Support blocks and can combine layout

Extend Core App Settings API

API Route: /core/app-settings?access_token=token

This API register all apps to the Native Mobile App. Implement the *getAppSetting* method in **PostApi** class and the *getMobileSettings* method in **PostResource** & **PostCategoryResource** classes

PostApi.php

```
<?php
class PostApi extends AbstractResourceApi implements ActivityFeedInterface, MobileAppSettingInterface
{
    /**...*/
    /**
     * Define Native Mobile App setting
     *
     * @param $param
     * @return MobileApp
     */
    public function getAppSetting($param)
    {
        $l = $this->getLocalization();
        $app = new MobileApp('post', [
            'title'=> $l->translate('Posts'),
            'home_view'=>'menu',
            'main_resource'=> new PostResource([]),
            'category_resource'=> new PostCategoryResource([],
                //'other_resources' => [
                    //      new PostListResource([]),
                //]
            ));
        $headerButtons = [
            [
                'icon' => 'list-bullet-o',
                'action' => Screen::ACTION_FILTER_BY_CATEGORY,
            ],
        ];
        if ($this->getAccessControl()->isGranted(PostAccessControl::ADD)) {
            $headerButtons[] = [
                'icon' => 'plus',
                'action' => Screen::ACTION_ADD,
                'params' => ['resource_name'=> 'post', 'module_name' => 'post']
            ];
        }
        $app->addSetting('home.header_buttons', $headerButtons);
        return $app;
    }
}
```

Posts/Api/Resource/PostResource.php

```

<?php

class PostResource extends ResourceBase
{
    /* ... */
    public function getMobileSettings($params = [])
    {
        $permission = NameResource::instance()->getApiServiceByResourceName($this->resource_name)
>getAccessControl()->getPermissions();
        $l = $this->getLocalization();
        return self::createSettingForResource([
            'acl'=> $permission,
            'resource_name' => $this->resource_name,
            'search_input' => [
                'placeholder'=>$l->translate('search_posts_dot'),
            ],
            'schema' => [
                'definition' => [
                    'categories' => 'post_category[]'
                ]
            ],
            'forms'=> [
                'addItem'=>[
                    'apiUrl'=>UrlUtility::makeApiUrl('post/form'),
                    'headerTitle'=>$l->translate('Add a New Post'),
                ],
                'editItem'=>[
                    'apiUrl'=>UrlUtility::makeApiUrl('post/form/:id'),
                    'headerTitle'=>$l->translate('Editing Post'),
                ]
            ],
            'list_view'=>[
                'noItemMessage' => $l->translate('no_posts_found'),
                'alignment' => 'right'
            ],
            'detail_view'=>[
                'component_name'=> 'blog_detail'
            ],
            'app_menu'=> [
                ['label' => $l->translate('all_posts'), 'params'=>['initialQuery' => ['view' =>
                '']],
                ['label' => $l->translate('my_posts'), 'params'=>['initialQuery' => ['view' =>
                'my']],
                ['label' => $l->translate('my_draft_post'), 'params'=>['initialQuery' =>
                ['view' => 'draft']],
                ['label' => $l->translate('friends_posts'), 'params'=>['initialQuery' =>
                ['view' => 'friend']],
                ['label' => $l->translate('pending_posts'), 'params'=>['initialQuery' =>
                ['view' => 'pending']], 'acl' => 'can_approve'],
            ],
        ]);
    }
}

```

PostCategoryResource no need to implement *getMobileSettings* because don't have a specific screen on Mobile App.

After implementing the above methods, you can re-check the API above and should see that new `post` object is appended to the Core App Setting API

```
"post": {
  "setting": {
    "resources": {
      "post": {},
      "post_category": {}
    },
    "screens": {
      "screen.home": {},
      "screen.home.default": {},
      "app.header_menu": {
        "options": []
      }
    }
  },
  "parameters": {
    "post.search_form": {},
    "post.edit": {},
    "post.add": {},
    "post.delete": {},
    "post_category.search_form": {},
    "post_category.edit": {},
    "post_category.add": {},
    "post_category.delete": {},
    "home.header_buttons": [],
    "category_menu": {},
    "app": {},
    "main_resource": "post",
    "resourceNames": []
  }
},
},
```

```

"post": {
  "acl": {
    "can_view": true,
    "can_like": true,
    "can_share": true,
    "can_delete": true,
    "can_report": true,
    "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
  },
  "add.icon": "plus",
  "add.label": "Add",
  "search_input": {
    "placeholder": "Search posts"
  },
  "resource_name": "post",
  "forms": {},
  "list_view": {
    "noItemMessage": "No Posts Found.",
    "alignment": "right"
  },
  "detail_view": {
    "component_name": "post_detail"
  },
  "app_menu": [,
  "apiUrl": "mobile/post"
},

```

Detail information of each section

Item	Type	Description
setting.resources	Object	The setting of each resource supported by the new Post app
setting.screens	Object	Screen Setting for the new app
setting.parameters	Object	Other parameters

Define Resource Settings

Resource setting	Type	Description
post.acl	Object	Control permission of the current user for the app
post.add.icon	string	Create a post button Icon

post.add.label	string	Create a post button label
post.search_input	object	Setting for search form in Post App
post.search_input.placeholder	string	Place Holder display on the app
post.resource_name	string	Name of resource
post.forms	object	Define API to get the form to create & edit
post.forms.addItem	object	Define apiUrl & headerTitle for the Add form
post.forms.editItem	object	Define apiUrl & headerTitle for the Edit form
post.list_view	object	Define layout component & information used to render layout on mobile app
post.detail_view	object	Define layout component used to render layout on detail resource listing
post.feed_view	object	Define feed attachment layout on App, with 2 property:

item_view: View name of attachment layout, support some default value: *embed_blog*, *embed_video*, *embed_music_song*, *embed_event...*

alignment: If *item_view* is empty, using *embed_default* view defined on App with the alignment of image on feed (**top, right, left**)

Extend Screen Settings via Core Site Setting API

API Route: `/core/site-settings?access_token=token`

Method: **GET**

Screen Setting Parameters

Define all screen properties that supported by your app.

```
"screens": {
  "screen.home": {
    "headerTitle": "Posts",
    "header_buttons": [
      {
        "icon": "list-bullet-o",
        "action": "@app/FILTER_CATEGORY"
      },
      {
        "icon": "plus",
        "label": "Add",
        "action": "@app/ADD_ITEM",
        "params": {
          "config_name": "post.add",
          "module_name": "post",
          "resource_name": "post"
        }
      }
    ],
    "header_menu": {
      "config_name": "app.header_menu",
      "action": "@@filter_by_app_menu"
    }
  },
  "screen.home.default": {
    "parent": "screen.home",
    "content_layout": "menu"
  },
  "app.header_menu": {
    "options": []
  }
},
```

Parameter	Type	Description
screen.home	object	Define Home screen of the app
screen.home.headerTitle	string	
screen.h.header_buttons	object	List of button display on the Home Screen
screen.home.header_menu	object	
screen.home.header_menu.config_name		
screen.home.header_menu.action		
screen.home.default		

screen.home.default.parent		
screen.home.default.category_layout		
screen.app.header_menu		

Naming Convention of Screen Name on Mobile app

Each module (such as Blog, Event, etc) will have 3 default screens: Module Home Screen, Module Detail Screen, Module Listing Screen

Screen Name	Naming Convention
Module Home Screen	<i>"module" + app name</i> Ex: moduleBlog
Module Detail Screen	<i>"detail" + app name</i> Ex: detailBlog
Module Listing Screen	<i>"listing" + app name</i> Ex: listingBlog

Here is the sample API response for Blog

```

"blog": {
  "moduleBlog": {
    "header": {<img alt="icon" data-bbox="395 395 425 410"/>},
    "content": {<img alt="icon" data-bbox="395 415 425 430"/>},
    "mainBottom": {<img alt="icon" data-bbox="395 435 425 450"/>},
    "screen_title": "Blogs > Blog - Home Page",
    "right": [<img alt="icon" data-bbox="395 460 425 475"/>]
  },
  "listingBlog": {
    "header": {<img alt="icon" data-bbox="395 510 425 525"/>},
    "content": {<img alt="icon" data-bbox="395 530 425 545"/>},
    "mainBottom": {<img alt="icon" data-bbox="395 550 425 565"/>},
    "screen_title": "Blogs > Blog - Search Results Page"
  },
  "detailBlog": {
    "header": {<img alt="icon" data-bbox="395 610 425 625"/>},
    "content": {<img alt="icon" data-bbox="395 630 425 645"/>},
    "bottom": {<img alt="icon" data-bbox="395 650 425 665"/>},
    "screen_title": "Blogs > Blog - Detail Page"
  }
},

```

Each screen has some sections to show content, support some locations:

Section Name	Description	Section ID
Location Top	Section on top	top
Location Bottom	Section on bottom	bottom
Location Main	The main content of the page	content
Location Header	Header of page	header
Location Right	Right block and show on a tablet only	right

Generate Screen Setting for an App

To implement Views component for screen setting, make sure your App was added API service implementing **MobileAppSettingInterface**. Then add the new function

getScreenSetting in that API.

Example:

```
use Apps\Core_MobileApi\Adapter\MobileApp\ScreenSetting;
use Apps\Core_MobileApi\Adapter\MobileApp\MobileAppSettingInterface;
//.....

class BlogApi extends AbstractResourceApi implements ActivityFeedInterface, MobileAppSettingInterface
{
    //.....
    public function getScreenSetting($param)
    {
        $screenSetting = new ScreenSetting('blog',[
            'name' => 'blogs'
        ]);
        $resourceName = BlogResource::populate([])->getResourceName();
        $screenSetting->addSetting($resourceName, ScreenSetting::MODULE_HOME);
        $screenSetting->addSetting($resourceName, ScreenSetting::MODULE_LISTING);
        $screenSetting->addSetting($resourceName, ScreenSetting::MODULE_DETAIL);
        $screenSetting->addBlock($resourceName, ScreenSetting::MODULE_HOME, ScreenSetting::
LOCATION_RIGHT, [
            'component' => ScreenSetting::SIMPLE_LISTING_BLOCK,
            'title' => 'most_viewed',
            'resource_name' => $resourceName,
            'module_name' => 'blog',
            'query' => ['sort' => 'most_viewed']
        ]);
        return $screenSetting;
    }
}
```

According to the above codes, we defined 3 screens for resource **Blog (Home, Listing, Detail)** and add a block on **Right** location in **Home** Function **\$screenSetting->addSetting**, will auto create a screen setting base on **Page Name** we mentioned above like that:

```

"listingBlog": {
  "header": {
    "component": "module_header"
  },
  "content": {
    "component": "smart_resource_list"
  },
  "mainBottom": {
    "component": "sort_filter_fab"
  },
  "screen_title": "Blogs > Blog - Search Results Page"
},
"detailBlog": {
  "header": {
    "component": "item_header"
  },
  "content": {
    "component": "item_simple_detail",
    "embedComponents": [
      "item_image",
      "item_title",
      "item_author",
      {
        "component": "item_stats",
        "stats": {
          "view": "total_view",
          "like": "total_like",
          "comment": "total_comment"
        }
      },
      "item_pending",
      "item_html_content",
      "item_category",
      "item_tags",
      "item_user_tags"
    ]
  },
  "bottom": {
    "component": "item_like_bar"
  },
  "screen_title": "Blogs > Blog - Detail Page"
}

```

But if you add custom **params**, it will be used as screen config for that page.
 If you add a custom **Page Name**, you have to define params for it. Example:

```
,
"viewBlockedUser": {
  "header": {
    "component": "simple_header",
    "title": "blocked_users"
  },
  "content": {
    "component": "smart_resource_list",
    "module_name": "user",
    "resource_name": "blocked_user",
    "pagingName": "blocked_users"
  },
  "screen_title": "Users > Blocked Users - Home Page"
},
"settings": {
  "top": {
    "component": "simple_header",
    "title": "system_settings"
  },
  "content": {
    "component": "user_settings"
  },
  "screen_title": "Users > System Settings - Home Page"
}
```

If you want to add a new block on a screen, have 2 ways:
Add a new location to your Page params.

Note: if you add a block into main content, it MUST be a child of the **embedComponents** array:

```
"detailEvent": {
  "header": {
    "component": "item_header",
    "transition": "float"
  },
  "content": {
    "component": "stream_profile_feeds",
    "embedComponents": [
      "stream_event_header_info",
      "stream_profile_description",
      "stream_composer"
    ]
  },
  "right": {
    "component": "simple_list_block",
    "module_name": "event",
    "resource_name": "event",
    "title": "events",
    "query": {
      "sort": "upcoming"
    }
  },
  "screen_title": "Events > Event - Detail Page"
},
```

If you don't add params for your page, and use default config, you have to use function **\$screenSetting->addBlock**. Below example, we will add block **Most Viewed** for **Blog Home Page** at Right section:

```
$screenSetting->addBlock($resourceName, ScreenSetting::MODULE_HOME, ScreenSetting::LOCATION_RIGHT, [
    'component' => ScreenSetting::SIMPLE_LISTING_BLOCK,
    'title' => 'most_viewed',
    'resource_name' => $resourceName,
    'module_name' => 'blog',
    'query' => ['sort' => 'most_viewed']
]);
```

Support parameters of each block

Key	Description
component	Component name of the block, this value should be: ScreenSetting::SIMPLE_LISTING_BLOCK
title	Block title
resource_name	Resource name will be used to fetch item and define which view will be used to show item
module_name	Module name

query (optional)	Extra query for API of this block, the block will call API: GET url/restful_api/mobile/<resource_name>?access_token=token<query> to get data
limit (optional)	Default 4, defined how many items show on the block

If you don't want to support ads on a screen, you should add property **"no_ads" => true** to that **Page Setting**

```
$screenSetting->addSetting($resourceName, 'mainFriendRequest', [
    ScreenSetting::LOCATION_HEADER => [
        'component' => ScreenSetting::SIMPLE_HEADER,
        'title' => 'friend_requests',
        'back' => false
    ],
    ScreenSetting::LOCATION_MAIN => [
        'component' => ScreenSetting::SMART_RESOURCE_LIST,
        'module_name' => 'friend',
        'resource_name' => FriendRequestResource::populate([])->getResourceName()
    ],
    'no_ads' => true
]);
```

Layout components

Layout components are pre-defined components used to render layout on mobile and controlled by **Core Site Setting API**

Components for module home page, module search page

List of components are defined to render layout on module **Home page** and module **Search/listing page**

Component Name	Description / Render	Recommend section	Supported parameter	Note
module_header	The Header section for module home page	header	- title: Title for this screen, default by module_name - enableTitleMenu: (true, false) show or hide title for this screen, the default value is true	Search bar, right buttons controlled by Core App Setting API
simple_header	The Header section for listing page	header	- title: Title for this screen, default by current category listing name - transition: Behavior animation for this header 'simple' 'float'	Does not support search bar, right buttons like module_header
stream_profile_feeds	List of feed items related to the current module	content	embedComponent contains other child components	None
smart_resource_list	List of items related to current module (no separation)	content	embedComponent contains other child components	Items type controlled by Core App Setting API
smart_resource_section	List of items related to current module (separated by section)	content	- Required an array parameter called sections. Each element in the array contain some parameters in order to control the layout of that section block: - module_name: Module name - resource_name: Resource name will be used to fetch item and define which view will be used to show the item in this section - header: { title: "your_title" }; Title of this section - use_query: Extra query for API of this section - limit: default is 10, defined how many items show on this section - embedComponent contains other child components	Items type controlled by Core App Setting API

smart_tabs	List of items related to current module (separated by tab component)	content	- Required an array parameter called tabs. Each element in the array contain some parameters in order to control the layout of that tab block: - label: Title of this tab - component: Component name for this tab - resource_name: Resource name will be used to fetch item and define which view will be used to show in this tab - query: Extra query for API of this tab - embedComponent contains other child components	Items type controlled by Core App Setting API
sort_filter_fab	A floating block contains sort & filter button	mainBottom	None	None

Components for module Detail page

List of components are defined to render layout on the module **Detail page**

Component Name	Description / Render	Recommend section	Supported parameter	Note
item_header	The header section for module detail page	header	- title: Title for this screen, the default is none - transition: Behavior animation for this header 'simple' 'float' 'transparent'	None
feed_header	Header for feed detail page	header	None	This is a special type of header, only use for feed detail
item_simple_detail	Wrapper block for other item's component	content	embedComponent contains other child components	None
feed_detail	Detail of current feed item	content	embedComponent contains other child components	None
stream_profile_description	Information block of the current item (short description location, category)	embedComponent	None	None
stream_profile_menus	Menu bar for the current item	embedComponent	None	Only use for profile item Ex. (User, Pages, Groups,)
stream_profile_photos	Photo block for the current item	embedComponent	None	None
stream_composer	Post Status, Composer block	embedComponent	None	None
item_image	Image field of this item	embedComponent	None	None
item_title	Title field of this item	embedComponent	None	None
item_pricing	Price field of this item	embedComponent	None	None
item_author	Author (created user) field of this item	embedComponent	None	None

item_stats	Statistic field of this item	embedComponent	stats: Object	Ex: stats: { view: 'total_view', like: 'total_like', comment: 'total_comment', vote: 'total_vote', play: 'total_play' }
item_pending	Message block if this item is pending	embedComponent	message: the message will show in the center of the block	None
item_description	Short description (non-HTML) field of this item	embedComponent	None	None
item_html_content	Full description (both HTML or non-HTML) field of this item	embedComponent	None	None
item_separator	Separate current item_simple_detail	embedComponent	None	None
item_category	Category field of this item	embedComponent	None	None
item_genre	Genre field of this item	embedComponent	None	None
item_tags	Tags field of this item	embedComponent	None	None
item_user_tags	User tags field of this item	embedComponent	None	None
item_location	Location field of this item	embedComponent	None	None
item_video	Video field of this item	embedComponent	None	None
item_like_bar	Like, Comment, Share bar for this item	bottom	None	None
....				

Components for all pages

List of components are defined to render some special blocks that can apply on all pages

Component Name	Description / Render	Available section	Supported parameter
banner_ad	Admob Banner	- top - bottom - embedComponent - right	- androidUnitId: Banner Unit Id provided by Google Admob to show a banner on the Android App - iosUnitId: Banner Unit Id provided by Google Admob to show a banner on the iOS App - capping: Frequency capping type of this banner ('no_frequency', 'time', 'views') - limit: Number of occurrences of this banner (apply when capping = 'time' or capping = 'views') - time: the time limit (second) that app will refresh this banner (apply when capping='time')
simple_list_block	List of limit items basing on parameters	- bottom - embedComponent - right	- title: Block title - module_name: Module name - resource_name: Resource name will be used to fetch item and define which view will be used to show item - query: Extra query for API of this block - limit: Default 4, defined how many items shows on the block

Components for specific pages

List of components are defined to render some special blocks that currently only apply for some specific pages

Component Name	Apply For Page	Description / Render	Available section	Supported parameter
stream_user_header_info	User detail	Top section of User page (Cover, Avatar, some specific information)	embedComponent	None
stream_event_header_info	Event detail	Top section of Event page (Cover, Title, author)	embedComponent	None
stream_pages_header_info	Pages detail	Top section of Pages page (Cover, Avatar, some specific information)	embedComponent	None
stream_groups_header_info	Groups detail	Top section of Groups page (Cover, Avatar, some specific information)	embedComponent	None