

Extending the Mobile Frontend

Basics

You can add custom JavaScript/CoffeeScript, HTML/Jade and CSS files to the mobile frontend from your plugin:

```
# in your plugins init-function:
init: (@app, @framework, @config) =>
  # [...]
  # wait till all plugins are loaded
  @framework.on "after init", =>
    # Check if the mobile-frontend was loaded and get a instance
    mobileFrontend = @framework.pluginManager.getPlugin 'mobile-frontend'
    if mobileFrontend?
      mobileFrontend.registerAssetFile 'js', "pimatic-your-plugin/app/some-
js.coffee"
      mobileFrontend.registerAssetFile 'css', "pimatic-your-plugin/app/css/
some-css.css"
      mobileFrontend.registerAssetFile 'html', "pimatic-your-plugin/app/som
e-html.jade"
    else
      env.logger.warn "your plugin could not find the mobile-frontend. No g
ui will be available"
```

The HTML/Jade files will get appended to the `body` element of the index page. The JS/CoffeeScript and CSS files will be inserted in the html `head` element.

Device templates

You can create your own frontend item templates for your devices. First add a `getTemplateName` function to your device:

```
class MyCustomDevice extends env.devices.Device
  # [...]
  getTemplateName: -> "customdevice"
```

Extending the Mobile Frontend

Then you need to define the HTML-template with the a corresponding id in an extra added HTML/Jade file:

```
<script id="customdevice-template" type="text/template">
  <li class="sortable device no-header">
    <label data-bind="text: name" class="device-label"></label>
    <a data-bind="click: onPressed" class="ui-btn ui-mini ui-btn-inline
ui-corner-all">click me</a>
  </li>
</script>
```

and define the behavior by adding a item class in an extra added JS/CS file;

```
$(document).on( "templateinit", (event) ->

# define the item class
class CustomDeviceItem extends pimatic.DeviceItem
  constructor: (data) ->
    super(data)
    # Do something, after create: console.log(this)
  afterRender: (elements) ->
    super(elements)
    # Do something after the html-element was added
  onPressed: ->
    $.get("/api/device/#{@deviceId}/actionToCall").fail ajaxAlertFail

# register the item-class
pimatic.templateClasses['customdevice'] = CustomDeviceItem
)
```

You can take a look on the [existing item classes](#) for examples.