

# Changing the end-user UI: programming your own skin

## History

A number of federations wanted to customise the eduroam CAT appearance so that it matches their local NRO look&feel. Typically, this was done by using the CAT APIs on a local webpage.

This approach creates some usability problems

- users coming from a different NRO while visiting the one in question may be directed to a page which looks very different from what they are used to
- there is no longer a one-stop-shop for configurations, e.g. Apple directed their users in a KB article to <https://cat.eduroam.org> - but this stops being the true target if an NRO hosts a web UI at a different URL.

The result of the on-list discussions was a feature request: "support multiple skins" - so that the UI at <https://cat.eduroam.org> could be made to look like it is a local page to the NRO.

## HOWTO

### end-user point of view

The end-user UI entry points consist of three files:

`/index.php` - main frontpage with selection of IdP etc.

`/basic.php` - similar to `index.php` but for devices with small screens or other handicaps (dumb browser...)

`/accountstatus/accountstatus.php` - Status and Pick-Up page for eduroam Managed IdP user accounts

### Coding your own

Skins can be created directly in the GitHub source of CAT (please send pull requests). Create a directory

`/web/skins/YOURSKIN/`

with YOURSKIN being an identifier of your choice. Inside that directory, you need at least the three files referenced above. Before calling your skinned pages, the CAT loader automatically loads the configuration, a skin handler, and depending on the page being called an array with extra information about the incoming request for your ease of coding. You find these as pre-defined constants and variables:

- **CONFIG**  
constant, containing the global CAT configuration file. You do not normally need much from the config. See [config/config-template.php](#) for a breakdown of its contents
- **\$skinObject**  
object with the skin handler (instance of the `core/Skinjob` class). The only function you need there is `Skinjob::findResourceUrl(string, bool)`. You need it whenever you reference something in an external file (such as your own CSS, an image, a login link to the admin/ area ...)
- **\$statusInfo**  
array with information about the sign-up token. Only present in `accountstatus.php`. Structure:  
`$statusInfo = ["token" => $cleanToken,`  
`"tokenstatus" => $tokenStatus,`  
`"OS" => $operatingSystem,`  
`"profile" => $profile,`  
`"idp" => $idp,`  
`"fed" => $fed];`

The skin pages are loaded with a cross-directory `include()` from the skin loader. Your skin should never assume to know its relative path on the server. It should rather construct all paths with `$skinObject->findResourceUrl("IMAGES")` (or "CSS" or "EXTERNAL" or "BASE") if you want to reference a image resource from the "global" stock of images (our stock is contained in the [web/resources/](#) directory) , and a `$skinObject->findResourceUrl("IMAGES", true)` if you want an image that is only relevant for and confined in your skins/YOURSKIN/ directory. Your directory structure MUST mimick the one in `web/resources/` for things to work.

### Need inspiration?

Take a look at [web/skins/example/](#). And of course for full functionality take a look at [web/skins/modern/](#).

## Activating the skin

The default skin is the "classic" skin. Add a new skin to the global config (CONFIG['APPEARANCE']['skins'] array). As soon as the skin is in the config, the usual way to change it is via fed admin UI; add the new option "Preferred Skin for User Area" and type the string that matches YOURSKIN.

This has the limitation that the skin will only show once the system knows which federation the user belongs to; i.e. they will get the classic UI until they have selected their IdP, and will then get the new skin for the actual download page.

To make all pages appear in the desired skin, it is possible to hard-code the desired skin in the POST or GET leading to the CAT page. Append a

`&skin=YOURSKIN` (or POST equivalent)

to the link and the UI will show with the desired skin immediately.

## Famous last words?

Happy coding!