

Solo 皮肤切换

作者: [88250](#)

原文链接: <https://ld246.com/article/1581165516956>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

本文是《Solo 从设计到实现》的一个章节，该系列文章将介绍 Solo 这款 Java 博客系统是如何从无有的，希望大家能通过它对 Solo 从设计到实现有个直观地了解、能为想参与贡献的人介绍清楚项目也希望能给重复发明——重新定义博客系统的人做个参考 00
eart

后台切换

管理员可以在管理后台切换皮肤，并且可以为桌面端和移动端指定不同的皮肤。配置保存在 option 中，对应的 id 为 `skinDirName` 和 `mobileSkinDirName`。

设置好以后选用的皮肤名称会被写入 Cookie，以便前台请求时可以快速获取到皮肤信息，避免查库。

后台切换皮肤请求处理 `SkinConsole#updateSkin` 方法：

```
public void updateSkin(final RequestContext context) {
    final JsonRenderer renderer = new JsonRenderer();
    context.setRenderer(renderer);

    try {
        final JSONObject requestJSONObject = context.requestJSON();
        final JSONObject skin = requestJSONObject.getJSONObject(Option.CATEGORY_C_SKIN);
        final JSONObject ret = new JSONObject();
        renderer.setJSONObject(ret);

        skinMgmtService.updateSkin(skin);

        final Response response = context.getResponse();
        final Cookie skinDirNameCookie = new Cookie(Common.COOKIE_NAME_SKIN, skin.getString(Option.ID_C_SKIN_DIR_NAME));
        skinDirNameCookie.setMaxAge(60 * 60); // 1 hour
        skinDirNameCookie.setPath("/");
        response.addCookie(skinDirNameCookie);
        final Cookie mobileSkinDirNameCookie = new Cookie(Common.COOKIE_NAME_MOBILE_SKIN, skin.getString(Option.ID_C_MOBILE_SKIN_DIR_NAME));
        mobileSkinDirNameCookie.setMaxAge(60 * 60); // 1 hour
        mobileSkinDirNameCookie.setPath("/");
        response.addCookie(mobileSkinDirNameCookie);

        ret.put(Keys.STATUS_CODE, true);
        ret.put(Keys.MSG, langPropsService.get("updateSuccLabel"));
    } catch (final ServiceException e) {
        LOGGER.log(Level.ERROR, e.getMessage(), e);

        final JSONObject jsonObject = new JSONObject().put(Keys.STATUS_CODE, false);
        renderer.setJSONObject(jsonObject);
        jsonObject.put(Keys.MSG, langPropsService.get("updateFailLabel"));
    }
}
```

前台切换

为了方便预览皮肤，访客可从前台自行切换皮肤，仅在首页路径下有效：`/skin=xxx`。实现逻辑是在 `IndexProcessor#showIndex` 时获取 `skin` 参数，然后用该参数指定的皮肤进行渲染，并写入 Cookie

备其他页面请求使用。

```
String specifiedSkin = Skins.getSkinDirName(context);
if (StringUtils.isBlank(specifiedSkin)) {
    final JSONObject skinOpt = optionQueryService.getSkin();
    specifiedSkin = Solos.isMobile(request) ?
        skinOpt.optString(Option.ID_C_MOBILE_SKIN_DIR_NAME) :
        skinOpt.optString(Option.ID_C_SKIN_DIR_NAME);
}
request.setAttribute(Keys.TEMPLTE_DIR_NAME, specifiedSkin);

Cookie cookie;
if (!Solos.isMobile(request)) {
    cookie = new Cookie(Common.COOKIE_NAME_SKIN, specifiedSkin);
} else {
    cookie = new Cookie(Common.COOKIE_NAME_MOBILE_SKIN, specifiedSkin);
}
cookie.setMaxAge(60 * 60); // 1 hour
cookie.setPath("/");
response.addCookie(cookie);
```

解析所需皮肤

请求到服务端后会优先从 Cookie 中获取皮肤名，如果为空再从查 option 表获取管理员指定的皮肤 同时也会判断是否是移动端，移动端请求需要使用移动端指定皮肤。实现代码在 SkinHandler#resolveSkinDir:

```
private void resolveSkinDir(final Request request) {
    String skin = Skins.getSkinDirNameFromCookie(request);
    if (StringUtils.isBlank(skin)) {
        final BeanManager beanManager = BeanManager.getInstance();
        final OptionQueryService optionQueryService = beanManager.getReference(OptionQueryService.class);
        final JSONObject skinOpt = optionQueryService.getSkin();
        if (Solos.isMobile(request)) {
            if (null != skinOpt) {
                skin = skinOpt.optString(org.b3log.solo.model.Option.ID_C_MOBILE_SKIN_DIR_NAME);
            } else {
                skin = org.b3log.solo.model.Option.DefaultPreference.DEFAULT_MOBILE_SKIN_DIR_NAME;
            }
        } else {
            if (null != skinOpt) {
                skin = skinOpt.optString(org.b3log.solo.model.Option.ID_C_SKIN_DIR_NAME);
            } else {
                skin = org.b3log.solo.model.Option.DefaultPreference.DEFAULT_SKIN_DIR_NAME;
            }
        }
    }

    request.setAttribute(Keys.TEMPLTE_DIR_NAME, skin);
}
```