

5

API

Annotation

```
package com.fr.stable.fun;

import com.fr.stable.StringUtils;

import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;

@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.TYPE)
public @interface Authorize {

    /**
     * id
     *
     * @return
     */
    String callSignKey() default StringUtils.EMPTY;

    /**
     * ,
     *
     * @return
     */
    String callSignKeyPath() default StringUtils.EMPTY;
}
```

Authorize

```
if (PluginContexts.currentContext().isAvailable()) {
    //
} else {
    //
}
```

```

@Authorize(callSignKey = Constants.PLUING_ID)
public class ThemeGreen extends AbstractThemeVariousProvider {

    @Override
    public String name() {
        return "AcrossGreen";
    }

    @Override
    public String text() {
        return "";
    }

    @Override
    public String coverPath() {
        return "/com/fr/solution/theme/green/files/cover.png";
    }

    @Override
    public String scriptPath() {
        if (PluginContexts.currentContext().isAvailable()) {
            return "/com/fr/solution/theme/green/files/theme.js";
        } else {
            return "";
        }
    }

    @Override
    public String stylePath() {
        if (PluginContexts.currentContext().isAvailable()) {
            return "/com/fr/solution/theme/green/files/style.css";
        } else {
            return "";
        }
    }
}

```

callSignKeyIDplugin.xmlid

- 1.
2. Authorizeextra-designer

	lic		*1	*2	*3
	/				

*1
 *2jetty
 *3tomcatjava web

-ID

```
package com.fr.stable.fun;

import com.fr.stable.StringUtils;

import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;

@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.TYPE)
public @interface Authorize {

    /**
     * id
     *
     * @return
     */
    String callSignKey() default StringUtils.EMPTY;

    /**
     * ,
     *
     * @return
     */
    String callSignKeyPath() default StringUtils.EMPTY;

    /**
     * group
     *
     */
    String groupSignKey() default StringUtils.EMPTY;
}
```