

com.fr.decision.fun.PassportProvider



LDAPHttpPassportProvider

认证方式

同步的用户

平台内置认证 LDAP认证 HTTP认证 Demo Login

Code:

导入/添加的用户

平台内置认证 LDAP认证 HTTP认证 Demo Login

Code:

添加用户后邮件通知



PassportProvider.java

```
package com.fr.decision.fun;

import com.fr.decision.authorize.Passport;
import com.fr.decision.webservice.bean.authentication.PassportBean;
import com.fr.stable.fun.mark.Mutable;

/**
 *
 * Created by zhouping on 2018/12/13.
 */
public interface PassportProvider extends Mutable {

    String MARK_STRING = "PassportProvider";

    int CURRENT_LEVEL = 1;

    /**
     *
     *
     * @return
     */
    String passportType();

    /**
     *
     *
     * @return
     */
    Class<? extends PassportBean> classForPassportBean();

    /**
     *
     *
     * @return
     */
    Class<? extends Passport> classForPassportConfig();
}
```

Passport.java

```
package com.fr.decision.authorize;

/**
 *
 *
 */
public interface Passport {

    boolean checkTicket(String username, String inputPassword, String savedPassword, String hashPassword);

    String markType();
}
```

PassportBean.java

```
package com.fr.decision.webservice.bean.authentication;

import com.fr.decision.authorize.Passport;
import com.fr.decision.webservice.bean.BaseBean;
import com.fr.third.fasterxml.jackson.annotation.JsonTypeInfo;

/**
 *
 * Created by zhouping on 2018/3/4.
 */
@JsonTypeInfo(use = JsonTypeInfo.Id.CLASS, include = JsonTypeInfo.As.PROPERTY, property = "@class")
public abstract class PassportBean<T> extends Passport> extends BaseBean {
    private static final long serialVersionUID = 3002910431724542517L;

    //
    private String type;
    //
    private int creationType = 0;

    public PassportBean() {
    }

    public String getType() {
        return markType();
    }

    public void setType(String type) {
        this.type = markType();
    }

    public int getCreationType() {
        return creationType;
    }

    public void setCreationType(int creationType) {
        this.creationType = creationType;
    }

    public abstract String markType();

    public abstract PassportBean<T> createPassportBean(T passport);

    public abstract Passport createPassport();
}
}
```

```
BI.config("dec.user.setting.authentications", function (items) {
    items.push({
        value: "",
        text: "",
        "@class": "PassportBean",
        component: {
            type: "UI"
        }
    });
    return items;
});
```

FR	10.0		
BI	5.1		BI
BI	5.1.2		BI
BI	5.1.3		BI

plugin.xml

```
<extra-decision>
  <WebResourceProvider class="your classname"/>
  <PassportProvider class="your classname"/>
</extra-decision>
```

```
Set<PassportProvider> providers = ExtraDecisionClassManager.getInstance().getArray(PassportProvider.MARK_STRING);
```

```
dec.user.setting.authenticationtypePassportsBean.createPassportsBean(Map<OperationType, AbstractPassport> map)Passports.buildBean(String type)
```

```
PassportProvider#passportTypePassportBean#markTypePassport#markType dec.user.setting.authentications#itemvalue
```

[demodemo-passport-provider](#)

(1)JSCSS

[demo-custom-passport](#)

[demo-ldaps-passport](#)