

# com.fr.stable.fun.WebLocaleProvider

- 
- 
- 
- 
- 
- 
- 
- 

URL

- 1.
- 2.

WebLocaleProvider

## WebLocaleProvider.java

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

import com.fr.stable.fun.mark.Mutable;

import java.util.Locale;

/**
 * locale;
 * ko_KR, ja_JP, ,
 * ko, ja.
 *
 * Created by Administrator on 2016/8/17/0017.
 */
public interface WebLocaleProvider extends Mutable{

    String XML_TAG = "WebLocaleProvider";

    int CURRENT_LEVEL = 1;

    Locale dispatchLocale(String localeStr);

}
```

FR	8.0		
FR	9.0		
FR	10.0		
BI	3.6		

BI	4.0		
BI	5.1		
BI	5.1.2		
BI	5.1.3		

### plugin.xml

```
<extra-core>
    <WebLocaleProvider class="your class name"/>
</extra-core>
```

### GeneralUtils.java

```
package com.fr.general;

...
public class GeneralUtils {
    ...

    /**
     *
     * ,
     *
     * @param localeStr
     * @return
     */
    public static Locale createLocale(String localeStr) {
        return createLocale(localeStr, Locale.US);
    }

    /**
     *
     *
     * @param localeStr
     * @param defaulVal ,
     * @return
     */
    public static Locale createLocale(String localeStr, Locale defaulVal) {
        ExtraClassManagerProvider provider = StableFactory.getMarkedObject(ExtraClassManagerProvider.XML_TAG,
        ExtraClassManagerProvider.class);
        if (provider != null) {
            Set<WebLocaleProvider> set = provider.getArray(WebLocaleProvider.XML_TAG);
            Locale locale;
            for (WebLocaleProvider localeProvider : set) {
                locale = localeProvider.dispatchLocale(localeStr);
                if (locale != null) {
                    return locale;
                }
            }
        }

        return LocaleType.parse(localeStr, defaulVal);
    }
    ...
}
```

## LocaleType.java

```
package com.fr.general;

import com.fr.stable.StringUtils;

import java.util.Locale;

/**
 * locale
 * localezh_CN, ja_JP, zh, ja.
 *
 * Win8 IE10, zh_CN, ,
 * zh_HANS, zh_HANS_CN, zh_HANS-CN, zh_HANS-HK.SSimplified
 * 2016-7-26 zh_CN_#Hans tomcat8 + iell
 *
 * Created by Administrator on 2016/8/17/0017.
 */
public enum LocaleType {

    JA {
        @Override
        public String localeDescription() {
            return "JA";
        }

        @Override
        public Locale getLocale() {
            return Locale.JAPAN;
        }
    },
    KO {
        @Override
        public String localeDescription() {
            return "KO";
        }

        @Override
        public Locale getLocale() {
            return Locale.KOREA;
        }
    },
    HANS {
        @Override
        public String localeDescription() {
            return "HANS";
        }

        @Override
        public Locale getLocale() {
            return Locale.CHINA;
        }
    },
    HANT {
        @Override
        public String localeDescription() {
            return "HANT";
        }

        @Override
        public Locale getLocale() {
            return Locale.TAIWAN;
        }
    },
    RU {
        @Override
        public String localeDescription() {
            return "RU";
        }
    }
}
```

```

        @Override
        public Locale getLocale() {
            return new Locale("ru", "RU");
        }
    },
    UZ {
        @Override
        public String localeDescription() {
            return "UZ";
        }

        @Override
        public Locale getLocale() {
            return new Locale("uz", "UZ");
        }
    };

    /**
     *
     *
     * @param localeStr
     *
     * @return parse
     */
    public static Locale parse(String localeStr) {
        return parse(localeStr, Locale.US);
    }

    /**
     *
     *
     * @param localeStr
     * @param defaultVal parse, locale
     *
     * @return parse
     */
    public static Locale parse(String localeStr, Locale defaultVal) {
        if (StringUtils.isEmpty(localeStr)) {
            if (defaultVal == null) {
                return null;
            }
            // , reqlocale
            localeStr = String.valueOf(defaultVal);
        }

        for (LocaleType type : values()) {
            if (localeStr.toUpperCase().indexOf(type.localeDescription()) != -1){
                return type.getLocale();
            }
        }

        String[] arr = localeStr.split("_");
        if (arr.length > 1) {
            return new Locale(arr[0], arr[1]);
        } else {
            return new Locale(arr[0], StringUtils.EMPTY);
        }
    }

    /**
     * locale
     *
     * @return
     */
    public Locale getLocale() {
        return Locale.US;
    }

    /**
     *

```

```

    *
    * @return
    */
    public String localeDescription() {
        return StringUtils.EMPTY;
    }
}

```

## WebUtils.java

```

package com.fr.web.utils;
...
public class WebUtils {
    ...
    /**
    * req
    *
    * @param req HTTP
    * @return
    * @date 2014-12-1-4:39:44
    */
    public static Locale getLocale(HttpServletRequest req) {
        if (req == null) {
            return GeneralContext.getLocale();
        }

        //fr_locale, js_fr_locale__
        String localeStr = WebUtils.getHttpRequestParameter(req, Constants.__FR_LOCALE__);
        localeStr = localeStr == null ? WebUtils.getHttpRequestParameter(req, "__fr_locale__") : localeStr;

        //, .
        if (StringUtils.isEmpty(localeStr) && "zh_CN".equals(String.valueOf(req.getLocale()))) {
            return Locale.CHINA;
        }

        return GeneralUtils.createLocale(replaceScript4Xss(localeStr), req.getLocale());
    }
    ...
}

```

WebLocaleProvider#dispatchLocale

WebUtils.getLocale zh\_CN\_\_fr\_locale\_\_fr\_localeWebLocaleProvider

zh\_CN\_\_fr\_locale\_\_fr\_localeWebLocaleProvider

[demodemo-web-locale-provider](#)