

# FunctionDefendProvider

- 
- - 
  -
- 
- 
- 

## FunctionDefendProvider

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

import com.fr.stable.fun.mark.Selectable;
import com.fr.stable.script.CalculatorProvider;
import com.fr.stable.script.Function;

public interface FunctionDefendProvider extends Selectable {

    String MARK_STRING = "FunctionDefendProvider";

    int CURRENT_LEVEL = 1;

    /**
     *
     *
     * @param calculator
     * @param clazz
     * @return
     */
    Function replacer(CalculatorProvider calculator, Class clazz);

    /**
     *
     *
     * @param calculator
     * @param clazz
     * @return
     */
    boolean match(CalculatorProvider calculator, Class clazz);
}
```

## AbstractFunction

```
package com.fr.script;

import com.fr.locale.InterProviderFactory;
import com.fr.log.FineLoggerFactory;
import com.fr.stable.Primitive;
import com.fr.stable.UtilEvalError;
import com.fr.stable.script.Node;

/**
 *
 * Sqrt (ABS(99) + 1)
 * ABS(99)9999 +100Sqrt(100)10
 */
public abstract class AbstractFunction extends CalculatorEmbeddedFunction {

    /**
     *
     * @param arguments
     * @return
     * @throws UtilEvalError
     */
    public Object evalExpression(Node[] arguments) throws UtilEvalError {

        Object returnValue;
        if(this.getType() == HA){
            returnValue = run(arguments);
        }else{
            Object[] args = new Object[arguments.length];
            for (int i = 0; i < arguments.length; i++) {
                args[i] = this.getCalculator().evalValue(arguments[i]);
            }
            returnValue = run(args);
        }

        if(returnValue == Primitive.ERROR_VALUE || returnValue == Primitive.ERROR_NAME){
            FineLoggerFactory.getLogger().error(InterProviderFactory.getProvider().getLocText("Fine-
Core_Base_NS_Cell_Formula") + this.toString());
        }

        return returnValue;
    }

    /**
     *
     *
     * SUM
     *
     * SUM sum = new SUM();
     * System.out.println(sum.run(new Object[]{1, 2, 3}));
     *
     * @param args
     * @return
     */
    public abstract Object run(Object[] args);
}
```

```
<extra-core>
    <FunctionDefendProvider class="com.fr.security.function.RemoteEvalFunctionDefender"/>
</extra-core>
```

登录 常规 打印 短信 邮箱 缓存

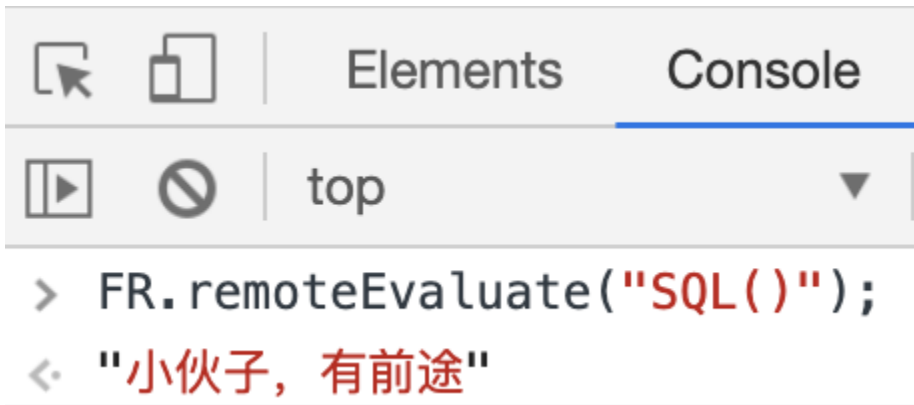
#### 常规参数

网络报表根目录名	<input type="text" value="reportlets"/>	连接网址管理中心	<input checked="" type="checkbox"/>
报表资源根目录名	<input type="text" value="resources"/>	Gzip压缩	<input checked="" type="checkbox"/>
服务端字符编码	<input type="text" value="UTF-8"/>	外接数据库	已配置
Servlet路径名	<input type="text" value="decision"/>	使用资源服务器	<input type="checkbox"/>
周开始于	<input type="text" value="星期日"/>		

#### 安全

启用脚本中公式调用限制  函数返回值

#### SQL



<https://code.fanruan.com/fanruan/demo-function-defender>