

com.fr.schedule.extension.report.provider. ReportScheduleResultProvider

•
•
•
•
•
•
•
•
•

cpfrfrcprfrcprfrrReportScheduleResultProvider

ReportScheduleResultProvider.java

```
package com.fr.schedule.extension.report.provider;

import com.fr.schedule.extension.report.result.ScheduleResultProvider;
import com.fr.stable.fun.mark.Mutable;

/**
 *ScheduleResultProvider.
 *
 * @author NIKO
 * created on 2020-11-13
 */
public interface ReportScheduleResultProvider extends Mutable {

    String MARK_STRING = "ReportScheduleResultProvider";

    int CURRENT_LEVEL = 1;

    /**
     *
     *
     * @return
     */
    ScheduleResultProvider customResult();
}
```

ScheduleResultProvider.java

```
package com.fr.schedule.extension.report.result;

import com.fr.schedule.extension.report.ScheduleShowType;
import com.fr.stable.web.Weblet;

import javax.servlet.http.HttpServletRequest;

/**
 * .
 *
 * <p>
 * ScheduleResultBox
 * </p>
 *
 * Created by Zed on 2018/4/16.
 */
public interface ScheduleResultProvider {

    String getResultSuffix();

    Weblet accessResult(HttpServletRequest req, String path, String fileName, ScheduleShowType showType) throws
    Exception;
}
```

FR	10.0		

plugin.xml

```
<extra-decision>
  <ReportScheduleResultProvider class="your class name"/>
</extra-decision>
```

ScheduleResultBox.java

```
package com.fr.schedule.extension.report.result;

import java.util.Map;
import java.util.concurrent.ConcurrentHashMap;

/**
 * .
 *
 * <p>
 * ScheduleResultProvider
 * </p>
 *
 * Created by Zed on 2018/4/16.
 */
public class ScheduleResultBox {

    private static Map<String, ScheduleResultProvider> map = new ConcurrentHashMap<String,
ScheduleResultProvider>();

    public static void register(ScheduleResultProvider provider) {
        if (!map.containsKey(provider.getResultSuffix())) {
            map.put(provider.getResultSuffix(), provider);
        }
    }

    public static ScheduleResultProvider formSuffix(String suffix) {
        if (map.containsKey(suffix)) {
            return map.get(suffix);
        }
        return null;
    }

    /**
     * provider.
     *
     * @param provider boxprovider
     */
    public static void remove(ScheduleResultProvider provider){
        if(map.containsKey(provider.getResultSuffix())) {
            map.remove(provider.getResultSuffix());
        }
    }

    /**
     * mapScheduleResultProvider.
     *
     * <p>
     *
     * </p>
     */
    public static void reset() {
        map.clear();
    }
}
```

ReportletResult.java

```
package com.fr.schedule.extension.report.result;

...

/**
 * Created by Zed on 2018/03/26.
 *
 */
public class ReportletResult extends TaskResult {

    ...

    @Override
    public void accessResult(HttpServletRequest req,
                            HttpServletResponse res,
                            Map<String, Object> param) throws Exception {

        String username = (String) param.get(ScheduleConstants.USERNAME);
        String taskName = (String) param.get(ScheduleConstants.TASK_NAME);
        String resultPath = (String) param.get(ScheduleConstants.TASK_RESULT_PATH);
        String showType = (String) param.get(ScheduleConstants.SHOW_TYPE);
        // cprfrr
        String path;
        ...

        String[] files = ResourceIOUtils.list(path);
        req.setAttribute(ScheduleConstants.USERNAME, username);
        req.setAttribute(ScheduleConstants.TASK_NAME, taskName);
        req.setAttribute(ScheduleConstants.SHOW_TYPE, showType);
        req.setAttribute(ScheduleConstants.TASK_RESULT_PATH, path);
        addTaskPara(req, path);

        WebletDealWith.dealWithWeblet(req, res, this.getResultFile(req, files, path, showType));
    }

    private Weblet getResultFile(HttpServletRequest req, String[] files, String path, String showType) throws
    Exception {
        for (String file : files) {
            ScheduleResultProvider provider = ScheduleResultBox.formSuffix(this.getFileSuffix(file));
            if (provider != null) {
                return provider.accessResult(req, StableUtils.pathJoin(path, file), file, this.getShowType
(showType));
            }
        }
        // cprfrr
        throw new ResultFileNotExistException("Fine-Schedule_Result_File_Not_Exist");
    }

    ...
}
```

ReportletResult26 String[] files = ResourceIOUtils.list(path); filesgetResultFilefilesScheduleResultProviderfilescprfrr

[demodemo-report-schedule-result-provider](#)

