

com.fr.decision.fun.MessageEventProvider

-
-
-
-
-
-
-
-
-

MessageEventProvider

MessageEventProvider.java

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

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

import java.util.Map;

/**
 * @author Zed
 * @version 10.0
 * Created by Zed on 2019/11/13
 */
public interface MessageEventProvider extends Mutable {

    String MARK_STRING = "MessageEventProvider";

    int CURRENT_LEVEL = 1;

    /**
     *
     */
    void handleMessage(Map<String, Object> message);
}
```

Message.java

```
package com.fr.decision.system.bean.message;

import com.fr.decision.system.entity.message.AbstractMessageEntity;
import com.fr.stable.db.data.BaseDataRecord;
import com.fr.decision.system.entity.message.MessageEntity;
import com.fr.third.fasterxml.jackson.annotation.JsonFormat;

import java.io.Serializable;
import java.util.Date;

/**
 * Created by Zed on 2018/02/27.
 * bean
 */
public abstract class Message extends BaseDataRecord implements Serializable {
```

```

private static final long serialVersionUID = 6348359470150684892L;

private String userId;
private String username;
@JsonFormat(pattern = "yyyy-MM-dd HH:mm:ss")
private Date datetime = new Date();
@JsonFormat(pattern = "yyyy-MM-dd HH:mm:ss")
private Date createTime = new Date();
private String message;
private String url;
//
private boolean toasted = false;
//
private boolean readed = false;
//
private int type;
//urlurl
private int urlType;

public abstract AbstractMessageEntity createEntity();

public Message() {

}

public MessageEntity setSelf2Entity() {
    MessageEntity entity = new MessageEntity();
    entity.setId(this.getId());
    entity.setCreateTime(this.getCreateTime());
    entity.setDatetime(this.getDatetime());
    entity.setMessage(this.getMessage());
    entity.setReaded(this.isReaded());
    entity.setToasted(this.isToasted());
    entity.setUrl(this.getUrl());
    entity.setUserId(this.getUserId());
    entity.setUsername(this.getUsername());
    entity.setType(this.getType());
    entity.setUrlType(this.getUrlType());
    return entity;
}

public Message(String userId, String message, String url, int urlType) {
    this.userId = userId;
    this.message = message;
    this.url = url;
    this.urlType = urlType;
}

public Message(String userId, String username, String message, String url, int type, int urlType) {
    this.userId = userId;
    this.username = username;
    this.message = message;
    this.url = url;
    this.type = type;
    this.urlType = urlType;
}

public String getUserId() {
    return userId;
}

public void setUserId(String userId) {
    this.userId = userId;
}

public String getUsername() {
    return username;
}

public void setUsername(String username) {

```

```
        this.username = username;
    }

    public Date getDatetime() {
        return datetime;
    }

    public void setDatetime(Date datetime) {
        this.datetime = datetime;
    }

    public Date getCreateTime() {
        return createTime;
    }

    public void setCreateTime(Date createTime) {
        this.createTime = createTime;
    }

    public String getMessage() {
        return message;
    }

    public void setMessage(String message) {
        this.message = message;
    }

    public String getUrl() {
        return url;
    }

    public void setUrl(String url) {
        this.url = url;
    }

    public boolean isToasted() {
        return toasted;
    }

    public void setToasted(boolean toasted) {
        this.toasted = toasted;
    }

    public boolean isReaded() {
        return readed;
    }

    public void setReaded(boolean readed) {
        this.readed = readed;
    }

    public int getType() {
        return type;
    }

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

    public int getUrlType() {
        return urlType;
    }

    public void setUrlType(int urlType) {
        this.urlType = urlType;
    }
}
```

FR	10.0		
BI	5.1		
BI	5.1.2		
BI	5.1.3		

plugin.xml

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

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

```
WebServiceActivatorMessageEvent.HANDLEMessageEvent.HANDLEMessageServicesaveMessage(Message message)
```

```
handleMessageMapMessage
```

```
demodemo-message-event-provider
```

```
decision-message-event
```