2025/11/07 04:24 1/1 Push support

Table of Contents

2025/11/07 04:24 1/2 Push support

Sometimes it's important to notify all client about an event or about changed records. This is known as "push". You push a message to all clients. In web environments a websocket is used for push notification. In JVx we have a push mechanism which is technology independent and it works different in different UI technologies.

How to push a message to other clients?

In our example, we have a button and on button press we send a simple reload notification to all other clients. This is a super simple task.

The action event for the button looks like:

```
public void doSendNotification(UIActionEvent pEvent) throws Throwable
{
    //saves all changes
    save();

    //triggers push
    getConnection().callAction("doReloadAllClients");
}
```

In our life-cycle object, we need the doReloadAllClients method:

```
public void doPublish()
{
    ICallBackBroker broker = SessionContext.getCurrentCallBackBroker();

    Thread th = new Thread(new Runnable()
    {
        public void run()
        {
            broker.publish("RELOAD", null,

PublishMode.AllOtherMasterSessions);
        }
    });
    th.start();

    //with JVx 3.0, not additional thread is required
    //SessionContext.getCurrentCallBackBroker().publish("RELOAD", null,
    PublishMode.AllOtherMasterSessions);
}
```

The reload event needs a handler. This handler/listener will be registered in our application:

2025/11/07 04:24 2/2 Push support

That's it.

But be careful, because push won't work in load-balanced applications with this standard implementation!

```
From:
https://doc.sibvisions.com/ - Documentation

Permanent link:
https://doc.sibvisions.com/jvx/communication/push_support

Last update: 2022/11/29 12:17
```