

Programming SIP Services in Java™ The SIP Servlet API

Anders Kristensen

Hewlett-Packard Laboratories, Bristol, U.K

SIP Servlet API

- ❑ **Java extension API for SIP servers**
- ❑ **Similar in spirit to HTTP servlet API**
- ❑ **Server matches incoming messages against local rules in order to decide which servlet to pass message to**
- ❑ **The API gives full control to servlets to handle SIP messages, e.g.**
 - has full access to headers and body
 - proxy or redirect requests
 - respond to or reject requests
 - forward responses upstream
 - initiate requests
- ❑ **Severs may choose to provide constrained environment to selected servlets (e.g. using sandbox security model)**

Benefits of Servlet Model

❑ Performance:

- No need to fork new process for each request
- The same servlet can handle many requests simultaneously

❑ Safety: type checked; no pointers; exception handling

❑ Convenience:

- high level abstractions.
- Tight integration with server: logging, security, location database

Benefits of Servlet Model, cont'd

- ❑ **Lifecycle model allows servlets to**
 - maintain state, e.g. database connections
 - manage timers

- ❑ **Access to wide range of APIs, e.g.**
 - JNDI for directory access
 - JDBC for database access
 - JMF for handling media, e.g. codecs and RTP
 - the speech API
 - JavaMail

Key Abstractions

- ❑ **SipServlet** Java objects extending SIP servers
- ❑ **SipServletContext** the SIP server - the servlet *engine*
 SipFactory
 ContactDatabase access to location information
- ❑ **SipTransaction**
 SipRequest
 SipResponse
- ❑ **SipURL**
 SipAddress addressing
 Contact

An Example: RejectServlet

```
import org.ietf.sip.*;

public class RejectServlet extends SipServletAdapter {
    protected int statusCode, reasonPhrase;

    public void init(ServletConfig config) {
        super.init(config);
        try {
            statusCode = Integer.parseInt(getInitParameter("status-code"));
            reasonPhrase = getInitParameter("reason-phrase");
        } catch (Exception _) {
            statusCode = SC_INTERNAL_SERVER_ERROR;
        }
    }

    public boolean doInvite(SipRequest req) {
        SipResponse res = req.createResponse();
        res.setStatus(statusCode, reasonPhrase);
        res.send();
        return true;
    }
}
```