```
      NodeList nodes = root.getElementsByTagName(EVENT_TAG);
      int eventCount = nodes.getLength();
      events = new ArrayList<Event>(eventCount);
      for (int i = 0; i < eventCount; i++) {
        Element event = (Element) nodes.item(i);
        // Get all headers. If there are multiple header sections,
        // combine them.
        NodeList headerNodes
          = event.getElementsByTagName(HEADERS_TAG);
        Map<String, String> eventHeaders
          = new HashMap<String, String>();
        for (int j = 0; j < headerNodes.getLength(); j++) {
          Node headerNode = headerNodes.item(j);
          NodeList headers = headerNode.getChildNodes();
          for (int k = 0; k < headers.getLength(); k++) {
            Node header = headers.item(k);

            // Read only element nodes
            if (header.getNodeType() != Node.ELEMENT_NODE) {
              continue;
            }
            // Make sure a header is inserted only once,
            // else the event is malformed
            Preconditions.checkState(
              !eventHeaders.containsKey(header.getNodeName()),
              "Header expected only once " + header.getNodeName());
            eventHeaders.put(
              header.getNodeName(), header.getTextContent());
          }
        }
        Node body = event.getElementsByTagName(BODY_TAG).item(0);
        if (insertTimestamp) {
          eventHeaders.put(TIMESTAMP_HEADER, String.valueOf(System
            .currentTimeMillis()));
        }
        events.add(EventBuilder.withBody(
          body.getTextContent().getBytes(
            httpServletRequest.getCharacterEncoding()),
          eventHeaders));
      }
    } catch (SAXException ex) {
      throw new HTTPBadRequestException(
        "Request could not be parsed into valid XML", ex);
    } catch (Exception ex) {
      throw new HTTPBadRequestException(
```

49