

Nightowl - Das Protokoll V0.01

yannick Baeumer — Kontakt: <http://stix.ws>

28. August 2009

Inhaltsverzeichnis

1	Paketübersicht	2
1.1	Einführung	2
1.2	Die Wurzelklassen	3
1.3	Struktur eines Pakets	3

Kapitel 1

Paketübersicht

1.1 Einführung

Das Nightowlprotokoll arbeitet wie auch mehrere andere Protokolle aufgrund von Schachtelung verschiedener Pakete (hier „packages“ genannt). Diese Pakete werden nach außen durch Xml-Tags repräsentiert, innerhalb der Serverstruktur jedoch als Instanzen der spezieller Klassen gehandelt. In dieser Dokumentation beschreibe ich das Zusammenspielen der verschiedenen Pakete in Form ihrer Objekte innerhalb des Servers, da man aufgrund der Zugriffsmodifikatoren, sowie ihrer Membermethoden gut erkennen kann, wer wann Zugriff auf den Inhalt eines Pakets haben darf. Als Beispiel sei hier die Klasse „package“ genannt, deren Felder „to“ und „from“ nur während der Instanziierung über den Konstruktormodifiziert werden können, zu sehen ist das daran, dass die entsprechenden Felder jeweils als private deklariert und lediglich die Wrappermethoden, die diese Werte wiedergeben wurden als „public“ deklariert, nicht diejenigen, mit denen man die Werte beeinflussen könnte.

Nun gehe ich aber von der allgemeinen Beschreibung der Struktur zu etwas direkteren Problematiken über.

Die Paketbibliothek des Nightowlprotokolls ist baumförmig aufgebaut, die konkreten Pakete, die letztendlich zur Informationsweitergabe verwendet werden, setzen sich aus Objekten zusammen, die aus folgenden vier Stammklassen abgeleitet sind:

- package
- request
- notification
- data

Die einzelnen Bäume der Unterklassen werde ich in gesonderten Sektionen beschrieben, um es möglichst übersichtlich zu halten.

1.2 Die Wurzelklassen

Folgende Struktur bildet die Wurzel des Nightowlprotokoll Klassenbaums:

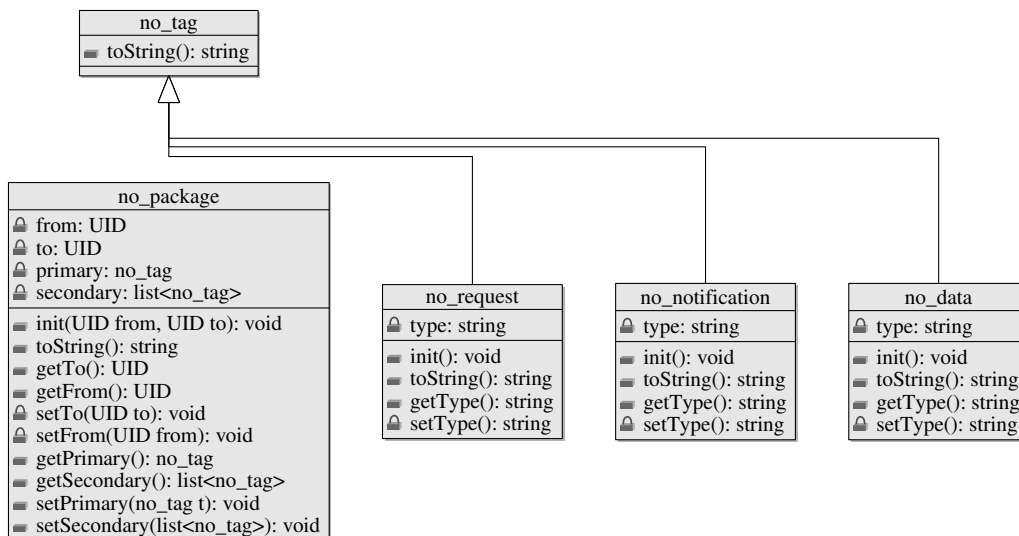


Abbildung 1.1: Wurzel des Paket-Klassenbaums

1.3 Struktur eines Pakets

Ein Paket besteht, wie der Name schon sagt zunächst aus einer Instanz der Klasse `package`. Da dies noch nicht viel über die Bedeutung aussagt, und es auch nicht Sinn der Sache ist, beispielweise von der Klasse „`package`“ abzuleiten und ein paar felder anzufügen, die Platz für Logindaten enthalten um sich zu Authentifizieren, stützt sich das Nightowlprotokoll auf die Schachtelung verschiedener Objekte.

Die Klasse „`package`“ hat nun zwei Möglichkeiten weitere Objekte der vier Stammklassen (oder ihrer Unterklassen) aufzunehmen, sofern sie von der Superklasse „`tag`“ abgeleitet sind.

- Das primäre Kapselobjekt:
Es wird unter der Variable „`primary`“ abgelegt und sollte auf jeden Fall definiert sein, sobald das Objekt in Form eines Xml-Tags den Server verlässt, da das Paket sonst für den Empfänger keinen semantischen Befehl enthielte.
- Das sekundäre Kapselobjekt:
Hier handelt es sich um eine Liste, die wiederum eine beliebige Zahl von „Tags“ aufnehmen kann, um das Paket um nötigen Informationen zu erweitern.

Die Xml-representation eines Paketes, dass ein „`request`“-Tag kapselt sieht folgendermaßen aus:

```

<pkg>
  <from></from>           //UID des Absenders
  <to></to>               //UID des Empfngers
  <pri>                   //steht fr das Primary-Feld
    <req>                 //request-Tag
      <type></type>       //gibt den Typ des requests an
      <data>              //kann ein data-tag mit Zusatzinformationen enthalten
        <type></type>     //Typ des data-Tags
      </data>
    </req>
  </pri>
</pkg>

```

Zu bemerken whre hier, dass das „secondary“-Feld nicht in den Xml-Tag mit einfließt, sofern es nicht definiert ist.