

# Communications Engineering UE

## CEBay Seeder und Client - Vorgehen

Thomas Fragner (1455954), Sandra Pühringer (1456472)

22. Januar 2017

### Inhaltsverzeichnis

<b>1</b>	<b>Server (Seeder)</b>	<b>2</b>
1.1	ServerApp . . . . .	2
1.2	ActorSeederManager . . . . .	3
1.3	ActorSeeder . . . . .	3
<b>2</b>	<b>Client</b>	<b>4</b>
2.1	ClientApp . . . . .	4
2.2	ActorClientManager . . . . .	5
2.3	ActorDownloader . . . . .	5
<b>3</b>	<b>Sequenz Diagramm</b>	<b>6</b>

## 1 Server (Seeder)

Der Server stellt Daten bereit. Diese registriert der Server am Registry Server der CEBay. Der Server ist aus mehreren Komponenten zusammengestellt:

**ServerApp** ist die eigentliche Application. Sie dient als Command Line Frontend für den Server. Das Actor System wird von der ServerApp erstellt. Weiters startet die Application einen Actor vom Typ ActorSeederManager.

**ActorSeederManager** Dieser Actor verwaltet die bereitzustellenden Dateien und erstellt für jede Datei einen Actor ActorSeeder.

**ActorSeeder** Bearbeitet Downloadanfragen für eine Datei

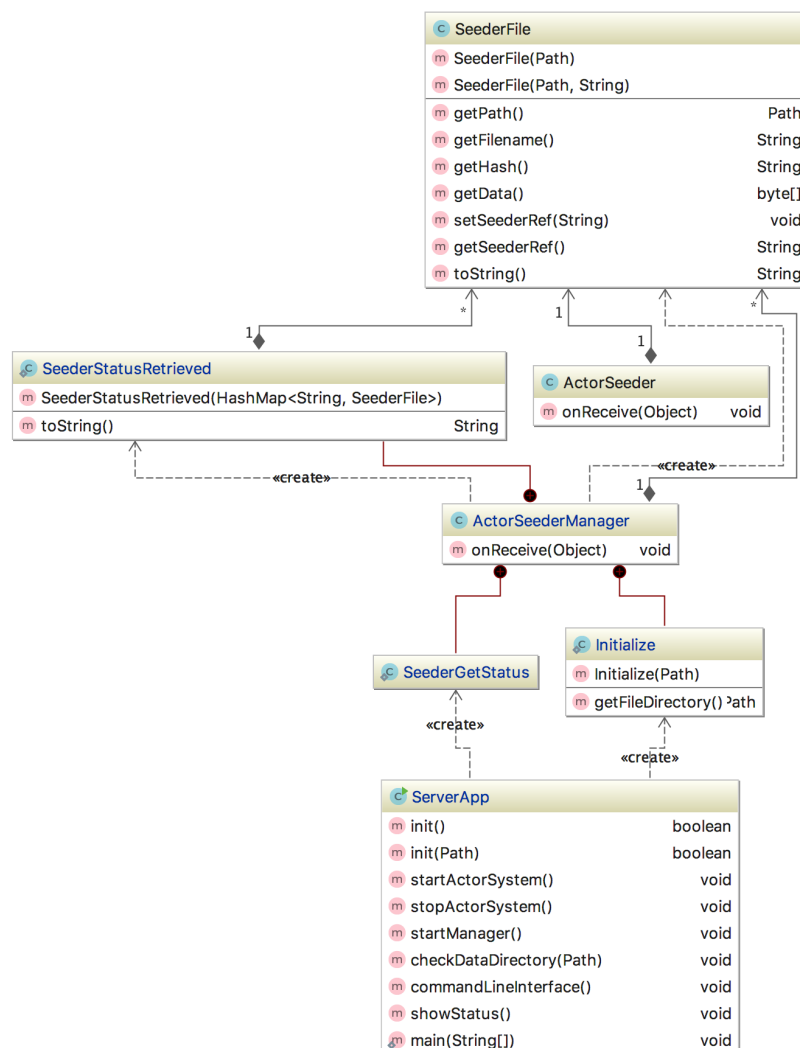


Abbildung 1: Klassenstruktur Server

## 1.1 ServerApp

Die Klasse `ServerApp` stellt ein ausführbares Programm zur Verfügung. Beim Starten erzeugt es ein `Actorsystem` und startet einen Actor `ActorSeederManager`. Diesem Actor kann beim Erstellen

ein Pfad zu einem Verzeichnis übergeben werden, in dem die bereitzustellenden Dateien gesucht werden. Nach dem Start des Actors wird ein Command Line Interface angeboten. Dieses bietet die Möglichkeit eine Statusabfrage zu allen bereitgestellten Dateien abzurufen.

## 1.2 ActorSeederManager

Die Klasse ActorSeederManager stellt einen Actor zur Verfügung. Dieser Actor verwaltet alle bereitzustellenden Dateien. Der Actor bearbeitet die folgenden Nachrichten:

**Initialize** Mit dieser Nachricht wird der Manager initialisiert. In der Nachricht ist der Pfad zum Verzeichnis mit den bereitzustellenden Dateien enthalten. Im Zuge der Initialisierung werden alle Dateien die sich in diesem Verzeichnis befinden bei der CEBay Registry registriert. Für jede gefundene Datei sendet der Actor eine Nachricht vom Typ SeederFile an sich selbst.

**SeederFile** Wenn eine Nachricht vom Typ SeederFile erhalten wird, dann erzeugt der Actor einen neuen Kind-Actor vom Typ ActorSeeder. Die Instanz von SeederFile wird um die ActorRef des erstellten Actors erweitert. Anschließend wird diese Instanz von SeederFile an den neuen Actor gesendet. Zusätzlich wird die Erstellung des neuen Aktors in einer HashMap gespeichert.

**SeederGetStatus** Beim Eintreffen dieser Nachricht wird die HashMap mit den Status der bereitgestellten Dateien in Form einer Nachricht vom Typ SeederStatusRetrieved an den Sender zurückgesendet.

**UploadFile** Diese Nachricht enthält Daten für eine neue Datei. Beim Erhalt der Nachricht wird das in der Nachricht enthaltene Byte Array in das Datenverzeichnis gespeichert. Der Dateiname der neuen Datei ist ebenfalls in der Nachricht enthalten.

## 1.3 ActorSeeder

Die Klasse ActorSeeder stellt den eigentlichen Seeder zur Verfügung. Hauptaufgaben des Actors sind:

- die Registrierung an der CEBay
- Downloadanfragen bearbeiten
- Statusabfragen beantworten

Folgende Nachrichten werden von Actor bearbeitet:

**SeederFile** Beim Erhalt einer Nachricht vom Typ SeederFile wird die entsprechende Datei bei der CEBay registriert. Dabei wird eine Nachricht vom Typ Publish mit dem Dateinamen, dem Hashwert der Datei und der ActorRef als String gesendet.

**GetFile** Bei einer Nachricht des Typs GetFile wird die vom Client angefragte Datei in eine Nachricht vom Typ FileRetrieved verpackt. und an den Sender als Antwort zurückgeschickt. Sollte die angefragte Datei nicht mit der Datei des Actors übereinstimmen wird eine Nachricht FileNotFound gesendet.

**GetStatus** Bei einer Statusanfrage durch die CEBay oder einen Client wird eine Nachricht vom Typ StatusRetrieved zurückgesendet.

## 2 Client

### 2.1 ClientApp

Die Klasse ClientApp stellt ein ausführbares Programm zur Verfügung. Die folgenden Funktionalitäten werden vom Programm bereitgestellt:

- Erstellen eines Actor Systems
- Erstellen des Downloadmanagers
- Abrufen der Dateiliste von der CEBay
- Download einer gewählten Datei
- Download aller verfügbaren Dateien
- Status der Downloads
- Upload einer neuen Datei

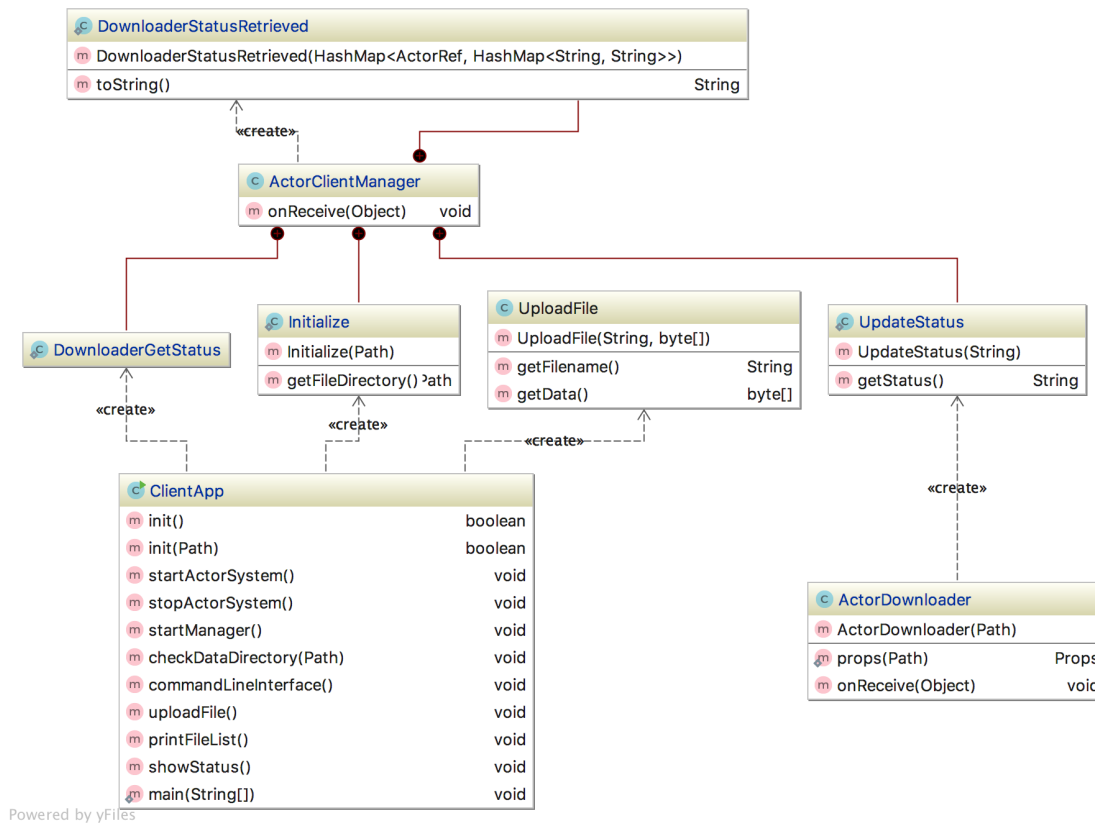


Abbildung 2: Klassenstruktur Client

Beim Abrufen der Dateiliste wird eine Nachricht vom Typ GetFileNames an die CEBay gesendet. Als Antwort kommt eine Liste mit verfügbaren Dateien zurück. Aus dieser Liste kann der Benutzer eine Datei auswählen, die heruntergeladen werden soll. Der Dateiname wird anschließend an den ActorClientManager übergeben. Weiter kann ein Upload gestartet werden. Dazu wird eine Nachricht vom Typ UploadFile an einen Seeder gesendet.

## 2.2 ActorClientManager

Die Klasse ActorClientManager stellt einen Actor zur Verfügung. Dieser Actor verwaltet alle Downloads. Der Actor bearbeitet die folgenden Nachrichten:

**Initialize** Das Downloadverzeichnis wird festgelegt.

**GetFile** Bei einer Anfrage des Client Programms vom Typ GetFile wird ein neuer Actor vom Typ ActorDownloader erstellt. Die erhaltene Nachricht wird an diesen Actor weitergeleitet.

**downloadGetStatus** Bei Erhalt dieser Nachricht wird der Status der Downloads zurückgesendet.

**UpdateStatus** Actor vom Typ ActorDownloader können diese Nachricht senden um den aktuellen Status bekanntzugeben. Wenn der Status failureist wird der entsprechende ActorDownloader beendet.

## 2.3 ActorDownloader

Die Klasse ActorDownloader wickelt genau einen bestimmten Download ab. Dazu werden folgenden Nachrichten verwendet:

**GetFile** Diese Nachricht weist den Actor an eine bestimmte Datei herunterzuladen. Im ersten Schritt werden die verfügbaren Seeder für die Datei bei der CEBay nachgefragt.

**SeederFound** Bei einer Antwort von der CEBay mit einer Nachricht vom Typ SeederFound wird an alle Seeder eine Nachricht vom Typ GetStatus gesendet. Damit soll festgestellt werden welche Seeder tatsächlich verfügbar sind. Es wird eine Antwort vom Typ StatusRetrieved erwartet.

**StatusRetrieved** Sobald sich ein möglicher Seeder meldet, werden keine weiteren Meldung vom Typ StatusRetrieved mehr akzeptiert und es wird eine Nachricht vom Typ GetFile an den entsprechenden Seeder gesendet. Weiters wird eine Nachricht an den Download Manager geschickt, dass der Download begonnen hat.

**FileRetrieved** Sobald der Seeder eine Nachricht vom Typ FileRetrieved zurücksendet wird diese Nachricht verarbeitet und die enthaltene Datei im Downloadverzeichnis abgespeichert. Danach wird eine Nachricht an den Download Manager gesendet, dass der Download fertig ist.

**FileNotFound** Wenn diese Nachricht eintrifft wird eine Meldung an den Download Manager gesendet um bekanntzugeben, dass der Download nicht funktioniert.

Weiter gibt es einen internen Timer. Wenn nach 90 Sekunden kein Seeder gefunden wurde wird ebenfalls ein Fehlschlag an den Download Manager gemeldet. Dies führt dann zum Beenden des Actors.

### 3 Sequenz Diagramm

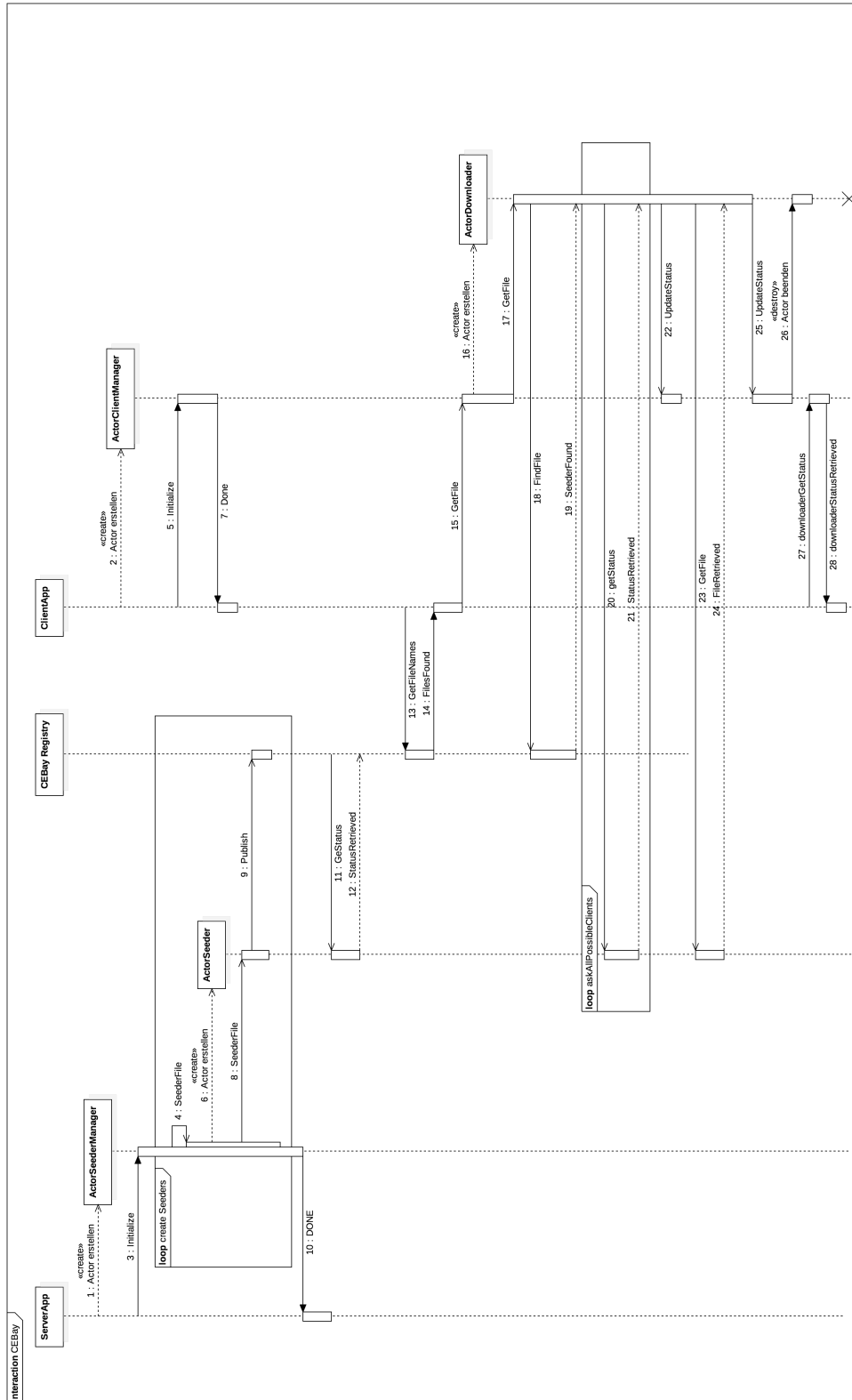


Abbildung 3: Sequenz Diagramm