

Android Client

Καμπυλαυκάς Ιωάννης
sdi9500781@di.uoa.gr

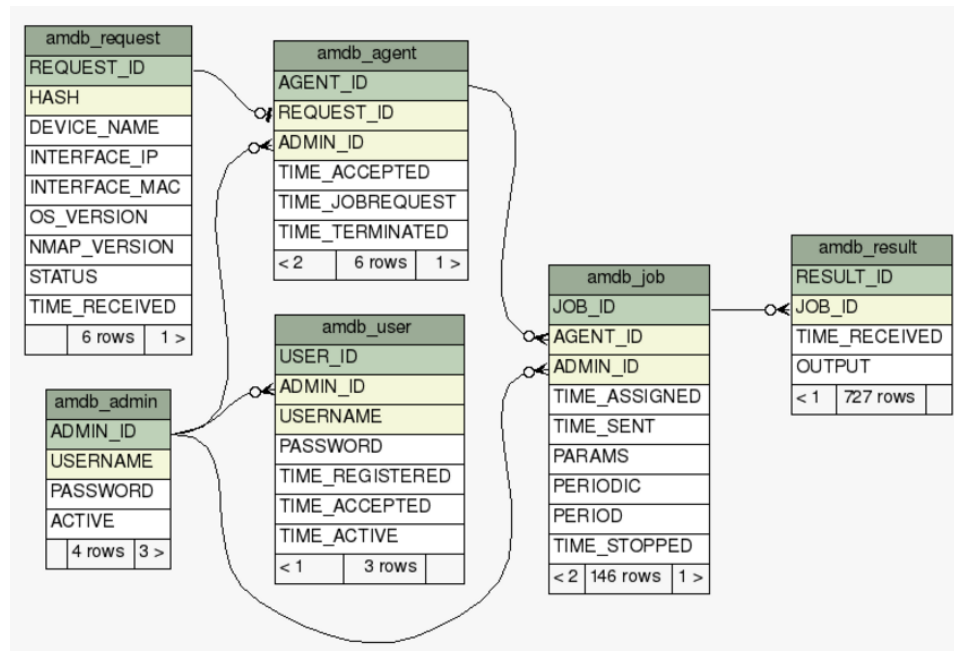
Σούλης Αθανάσιος
sdi0900155@di.uoa.gr

2 Φεβρουαρίου 2015

Περιεχόμενα

1	Οι προσθήκες στον Aggregator Manager	2
1.1	Σχήμα της βάσης δεδομένων	2
1.2	Ρυθμίσεις	3
1.3	Οι REST υπηρεσίες για τους Android Clients	4
1.3.1	Εγγραφή	4
1.3.2	Σύνδεση	4
1.3.3	Λήψη Software Agents	5
1.3.4	Λήψη των nmap jobs ενός Software Agent	6
1.3.5	Λήψη τελευταίων αποτελεσμάτων ενός Software Agent	8
1.3.6	Λήψη τελευταίων αποτελεσμάτων για όλους τους Software Agents	9
1.3.7	Ανάθεση εργασιών από απομακρυσμένους χρήστες	11
2	Ο Android Client	13
2.1	Γραφική διεπαφή	13
2.1.1	Εγγραφή - Σύνδεση	13
2.1.2	Αποσύνδεση	13
2.1.3	Agents	14
2.1.4	Jobs	14
2.1.5	Agent Results	15
2.1.6	All Results	15
2.1.7	Η απεικόνιση των αποτελεσμάτων	16
2.2	Υπηρεσίες του Android Client	17
2.2.1	Ανάθεση Εργασιών	17
2.2.2	Asynchronous tasks για επικοινωνία με τον Aggregator Manager	20
2.2.3	Αποθήκευση agents και jobs για offline χρήση	21

1 Οι προσθήκες στον Aggregator Manager



Σχήμα 1: Το ενημερωμένο σχήμα της βάσης δεδομένων του Aggregator Manager.

1.1 Σχήμα της βάσης δεδομένων

Στους υπάρχοντες πίνακες από το δεύτερο μέρος του project προστέθηκε ο πίνακας των χρηστών των Android Clients, amdb_users. Τα πεδία του είναι τα εξής:

- **USER_ID**: πρωτεύον κλειδί του πίνακα, δημιουργείται αυτόματα.
- **ADMIN_ID**: ξένο κλειδί στον πίνακα των διαχειριστών του Aggregator Manager. Όταν είναι NULL, ο χρήστης δεν έχει εγκριθεί από κάποιον διαχειριστή. Όταν κάποιος διαχειριστής εγκρίνει έναν χρήστη, τότε το πεδίο παίρνει την τιμή του αναγνωριστικού του διαχειριστή.
- **USERNAME**: το όνομα του χρήστη. Υπάρχει περιορισμός μοναδικότητας της τιμής του πεδίου αυτού για τον πίνακα, όπως και στο αντίστοιχο πεδίο του πίνακα των διαχειριστών.
- **PASSWORD**: το συνθηματικό του χρήστη. Δεν αποθηκεύεται αυτούσιο το συνθηματικό αλλά η έξοδος του αλγορίθμου SHA256 με είσοδο το συνθηματικό.

- **TIME_REGISTERED**: είναι η χρονική στιγμή της εγγραφής του χρήστη στη βάση δεδομένων.
- **TIME_ACCEPTED**: είναι η χρονική στιγμή της αποδοχής του χρήστη από κάποιον διαχειριστή. Στην περίπτωση που ο χρήστης δεν έχει γίνει ακόμη αποδεκτός έχει την τιμή NULL. Την τιμή NULL παίρνει και αν κάποιος διαχειριστής απορρίψει εκ των υστέρων έναν χρήστη που έχει γίνει αρχικά αποδεκτός.
- **TIME_ACTIVE**: είναι η χρονική στιγμή της τελευταίας χρήσης των υπηρεσιών του Aggregator Manager από τον χρήστη.

Όταν ένας χρήστης αναθέσει μια εργασία σε κάποιον Software Agent, αυτή δημιουργείται εκ μέρους του διαχειριστή που έχει εγκρίνει τον χρήστη. Η δημιουργία εργασιών γίνεται τελικά μόνο από διαχειριστές του Aggregator Manager, όπως και στο δεύτερο μέρος του project.

Στη γραφική διεπαφή για τον Aggregator Manager έχει προστεθεί η αναγκαία όψη για την αποδοχή ή απόρριψη χρηστών από κάποιον διαχειριστή.

1.2 Ρυθμίσεις

Στις ρυθμίσεις του Aggregator Manager έχουν προστεθεί οι εξής, σχετικές με τη δραστηριότητα των απομακρυσμένων χρηστών.

```
expireUserSessions=true
userSessionMinutes=30
```

Η ρύθμιση userSessionMinutes καθορίζει για πόσα λεπτά μπορεί να μείνει ενεργός ένας χρήστης Android Client χωρίς να χρησιμοποιήσει κάποια από τις υπηρεσίες του Aggregator Manager. Μετά την παρέλευση του χρόνου αυτού ο Aggregator Manager δεν επιστρέφει δεδομένα στα αιτήματα του χρήστη. Η ρύθμιση αυτή λαμβάνεται υπ' όψιν μόνο στην περίπτωση που η ρύθμιση expireUserSessions είναι ενεργή. Σε αντίθετη περίπτωση, ο Aggregator Manager επιστρέφει πάντοτε δεδομένα σε κάθε αίτημα ενός Android Client με έγκυρες παραμέτρους και η ρύθμιση userSessionMinutes χρησιμοποιείται μόνο στην όψη των χρηστών της γραφικής διεπαφής του Aggregator Manager, για να τους χαρακτηρίσει ενεργούς ή ανενεργούς.

1.3 Οι REST υπηρεσίες για τους Android Clients

Στον Aggregator Manager έχει προστεθεί η κλάση ClientHandlers που αφορά στην επικοινωνία των Android Clients με αυτόν. Οι υπηρεσίες που περιλαμβάνει είναι οι εξής:

1.3.1 Εγγραφή

```
@GET
@Path("register/{username}/{password}")
@Produces(MediaType.TEXT_PLAIN)
public Response register(
    @PathParam("username") String username,
    @PathParam("password") String password) {
    ...
}
```

Η μέθοδος register() ακούει σε ένα URI της μορφής:
`http://localhost:8080/am/client/register/{username}/{password}`
για ένα αίτημα εγγραφής νέου χρήστη Android Client.

- username: το όνομα του νέου χρήστη.
- password: το συνθηματικό του χρήστη.

Οι πιθανές απαντήσεις του AM είναι:

- "Registration Success", στην περίπτωση επιτυχούς εγγραφής.
- "User Exists", αν υπάρχει ήδη χρήστης με το όνομα που δόθηκε.
- "Service Error", στην περίπτωση οποιουδήποτε άλλου σφάλματος.

Αν το όνομα του νέου χρήστη δεν υπερβαίνει το μέγιστο μέγεθος και δεν υπάρχει ήδη χρήστης με αυτό το όνομα, τότε η εγγραφή είναι επιτυχής. Ο χρήστης πρέπει να εγκριθεί από κάποιον διαχειριστή προτού να μπορέσει να συνδεθεί επιτυχώς μέσω της υπηρεσίας σύνδεσης.

1.3.2 Σύνδεση

```
@GET
@Path("login/{username}/{password}")
@Produces(MediaType.TEXT_PLAIN)
public Response login(
    @PathParam("username") String username,
    @PathParam("password") String password) {
    ...
}
```

Η μέθοδος login() ακούει σε ένα URI της μορφής:
http://localhost:8080/am/client/login/{username}/{password}
για ένα αίτημα σύνδεσης χρήστη Android Client.

- username: το όνομα του χρήστη.
- password: το συνθηματικό του χρήστη.

Οι πιθανές απαντήσεις του AM είναι:

- "Login Success", στην περίπτωση επιτυχούς σύνδεσης.
- "Incorrect Credentials", αν τα στοιχεία χρήστη είναι μη έγκυρα.
- "Registration Pending", αν ο χρήστης δεν έχει εγκριθεί ακόμη.
- "Service Error", στην περίπτωση οποιουδήποτε άλλου σφάλματος.

Αν τα στοιχεία είναι έγκυρα, ενημερώνεται το πεδίο TIME_ACTIVE της εγγραφής που αντιστοιχεί στον χρήστη και στο εξής ο χρήστης μπορεί να στέλνει αιτήματα στις υπόλοιπες υπηρεσίες του Aggregator Manager με μέγιστο χρονικό διάστημα μεταξύ αιτημάτων όσο καθορίζεται από την ρύθμιση userSessionMinutes του AM. Αν αυτό το χρονικό διάστημα ξεπεραστεί, απαιτείται νέα σύνδεση του χρήστη.

1.3.3 Λήψη Software Agents

```
@GET
@Path("agents/{username}/{password}")
@Produces(MediaType.APPLICATION_XML)
public Response agents(
    @PathParam("username") String username,
    @PathParam("password") String password) {
    ...
}
```

Η μέθοδος agents() ακούει σε ένα URI της μορφής:
http://localhost:8080/am/client/agents/{username}/{password}
για ένα αίτημα χρήστη για τους Software Agents που υπάρχουν στη βάση του Aggregator Manager.

- username: το όνομα του νέου χρήστη.
- password: το συνθηματικό του χρήστη.

Ένα παράδειγμα απάντησης της υπηρεσίας είναι το εξής:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<agents status="Accepted" >
  <agent>
    <agentId>4</agentId>
    <requestHash>
      C886C27EF2CA686311C6918FA43B0AF90D2F7FFBD73C2958D48D399D4E037563
    </requestHash>
    <timeAccepted>2016-01-28 22:36:21.0 EET</timeAccepted>
    <agentStatus>OFFLINE</agentStatus>
  </agent>
  <agent>
    <agentId>3</agentId>
    <requestHash>
      5A885FD0CEB82033480975B0203B632EE07E8E8B8B4421894969C2777D10D5D5
    </requestHash>
    <timeAccepted>2016-01-26 18:25:18.0 EET</timeAccepted>
    <timeJobRequest>2016-02-03 02:37:04.0 EET</timeJobRequest>
    <timeTerminated>2016-02-02 01:36:48.0 EET</timeTerminated>
    <agentStatus>ONLINE</agentStatus>
  </agent>
</agents>
```

Οι πιθανές τιμές του attribute status είναι οι εξής:

- "Accepted", αν το αίτημα έγινε δεκτό. Αυτή είναι η μόνη περίπτωση που επιστρέφονται δεδομένα (agents) στον Android Client.
- "Session Expired", στην περίπτωση που ο χρόνος σύνδεσης έχει παρέλθει.
- "Incorrect Credentials", αν τα στοιχεία χρήστη είναι μη έγκυρα.
- "Registration Pending", αν ο χρήστης δεν έχει εγκριθεί ακόμη.
- "Service Error", στην περίπτωση οποιουδήποτε άλλου σφάλματος.

1.3.4 Λήψη των nmap jobs ενός Software Agent

```
@GET
@Path("/jobs/{username}/{password}/{agentHash}")
@Produces(MediaType.APPLICATION_XML)
public Response jobs(
    @PathParam("username") String username,
    @PathParam("password") String password,
    @PathParam("agentHash") String agentHash) {
    ...
}
```

Η μέθοδος jobs() ακούει σε ένα URI της μορφής: `http://localhost:8080/am/client/jobs/{username}/{password}/{agentHash}` για ένα αίτημα χρήστη για τα jobs που έχουν ανατεθεί σε κάποιον Software Agent. Οι παράμετροι είναι:

- username: το όνομα του χρήστη.
- password: το συνθηματικό του χρήστη.
- agentHash: το αναγνωριστικό του Software Agent.

Ένα παράδειγμα απάντησης της υπηρεσίας είναι το εξής:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<jobs status="Accepted">
  <job>
    <jobId>11</jobId>
    <agentId>1</agentId>
    <adminId>4</adminId>
    <timeAssigned>2016-01-25 12:36:47.0 EET</timeAssigned>
    <params>-O -oX - 127.0.0.1</params>
    <periodic>true</periodic>
    <period>30</period>
  </job>
  <job>
    <jobId>27</jobId>
    <agentId>1</agentId>
    <adminId>4</adminId>
    <timeAssigned>2016-01-26 16:50:10.0 EET</timeAssigned>
    <params>-A -oX - 192.168.1.0/24</params>
    <periodic>false</periodic>
    <period>0</period>
  </job>
  <job>
    <jobId>28</jobId>
    <agentId>1</agentId>
    <adminId>4</adminId>
    <timeAssigned>2016-01-26 16:59:51.0 EET</timeAssigned>
    <params>stop 11</params>
    <periodic>false</periodic>
    <period>0</period>
  </job>
</jobs>
```

Οι πιθανές τιμές του attribute status είναι οι εξής:

- "Accepted", αν το αίτημα έγινε δεκτό. Αυτή είναι η μόνη περίπτωση που επιστρέφονται δεδομένα (jobs) στον Android Client.
- "Invalid Hash", αν το αναγνωριστικό του Software Agent είναι μη έγκυρο.
- "Session Expired", στην περίπτωση που ο χρόνος σύνδεσης έχει παρέλθει.
- "Incorrect Credentials", αν τα στοιχεία χρήστη είναι μη έγκυρα.
- "Registration Pending", αν ο χρήστης δεν έχει εγκριθεί ακόμη.
- "Service Error", στην περίπτωση οποιουδήποτε άλλου σφάλματος.

1.3.5 Λήψη τελευταίων αποτελεσμάτων ενός Software Agent

```
@GET
@Path("results/agent/{username}/{password}/{agentHash}/{number}")
@Produces(MediaType.APPLICATION_XML)
public Response agentResults(
    @PathParam("username") String username,
    @PathParam("password") String password,
    @PathParam("agentHash") String agentHash,
    @PathParam("number") int number) {
    ...
}
```

Η μέθοδος agentResults() ακούει σε ένα URI της μορφής:
http://localhost:8080/am/client/results/agent/{username}/{password}/{agentHash}/{number} για ένα αίτημα χρήστη για τα τελευταία αποτελέσματα κάποιου Software Agent. Οι παράμετροι του αιτήματος είναι:

- username: το όνομα του χρήστη.
- password: το συνθηματικό του χρήστη.
- agentHash: το αναγνωριστικό του Software Agent.
- number: το μέγιστο πλήθος των τελευταίων αποτελεσμάτων.

Ένα παράδειγμα απάντησης της υπηρεσίας είναι το εξής:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<results status="Accepted">
  <result>
    <resultId>602</resultId>
    <jobId>9</jobId>
    <timeReceived>2016-01-24 14:14:11.0 EET</timeReceived>
    <output>
      &lt;?xml version="1.0"?&gt;
      &lt;nmaprun scanner="nmap" args="nmap -O -oX - 127.0.0.1"
        start="1453637628" startstr="Sun Jan 24 14:13:48 2016" version="6.47"
        ...
      </output>
    </result>
  <result>
    <resultId>603</resultId>
    <jobId>9</jobId>
    <timeReceived>2016-01-24 14:14:41.0 EET</timeReceived>
    <output>
      &lt;?xml version="1.0"?&gt;
      &lt;nmaprun scanner="nmap" args="nmap -O -oX - 127.0.0.1"
        start="1453637658" startstr="Sun Jan 24 14:14:18 2016" version="6.47"
        ...
      </output>
    </result>
</results>
```



```

<result>
  <resultId>606</resultId>
  <jobId>9</jobId>
  <timeReceived>2016-01-24 14:16:11.0 EET</timeReceived>
  <output>
    <?xml version="1.0"?>
    <nmaprun scanner="nmap" args="nmap -O -oX - 127.0.0.1"
      start="1453637748" startstr="Sun Jan 24 14:15:48 2016" version="6.47"
      ...
    </nmaprun>
  </output>
</result>
</results>

```

Οι πιθανές τιμές του attribute status είναι οι εξής:

- "Accepted", αν το αίτημα έγινε δεκτό. Αυτή είναι η μόνη περίπτωση που επιστρέφονται δεδομένα (results) στον Android Client.
- "Invalid Hash", αν το αναγνωριστικό του Software Agent είναι μη έγκυρο.
- "Session Expired", στην περίπτωση που ο χρόνος σύνδεσης έχει παρέλθει.
- "Incorrect Credentials", αν τα στοιχεία χρήστη είναι μη έγκυρα.
- "Registration Pending", αν ο χρήστης δεν έχει εγκριθεί ακόμη.
- "Service Error", στην περίπτωση οποιουδήποτε άλλου σφάλματος.

1.3.6 Λήψη τελευταίων αποτελεσμάτων για όλους τους Software Agents

```

@GET
@Path("results/all/{username}/{password}/{number}")
@Produces(MediaType.APPLICATION_XML)
public Response allResults(
    @PathParam("username") String username,
    @PathParam("password") String password,
    @PathParam("number") int number) {
    ...
}

```

Η μέθοδος allResults() ακούει σε ένα URI της μορφής:

http://localhost:8080/am/client/results/all/{username}/{password}/{number} για ένα αίτημα χρήστη για τα τελευταία αποτελέσματα όλων των Software Agents. Οι παράμετροι του αιτήματος είναι:

- username: το όνομα του χρήστη.
- password: το συνθηματικό του χρήστη.
- number: το μέγιστο πλήθος των τελευταίων αποτελεσμάτων.

Ένα παράδειγμα απάντησης της υπηρεσίας είναι το εξής:

```
<results status="Accepted">
  <result>
    <resultId>1370</resultId>
    <agentHash>
      5A885FD0CEB82033480975B0203B632EE07E8E8B8B4421894969C2777D10D5D5
    </agentHash>
    <jobId>119</jobId>
    <timeReceived>2016-01-29 23:16:34.0 EET</timeReceived>
    <output>
      &lt;?xml version="1.0"?&gt;
      &lt;nmaprun scanner="nmap" args="nmap -O -oX - 127.0.0.1"
      start="1454102163" startstr="Fri Jan 29 23:16:03 2016" version="6.47"
      xmloutputversion="1.04"&gt;
      ...
    </output>
  </result>
  <result>
    <resultId>1371</resultId>
    <agentHash>
      5A885FD0CEB82033480975B0203B632EE07E8E8B8B4421894969C2777D10D5D5
    </agentHash>
    <jobId>118</jobId>
    <timeReceived>2016-01-29 23:16:46.0 EET</timeReceived>
    <output>
      &lt;?xml version="1.0"?&gt;
      &lt;nmaprun scanner="nmap" args="nmap -O -oX - 127.0.0.1"
      start="1454102183" startstr="Fri Jan 29 23:16:23 2016" version="6.47"
      xmloutputversion="1.04"&gt;
      ...
    </output>
  </result>
  <result>
    <resultId>1372</resultId>
    <agentHash>
      85D4017BA2AD69140157A7E3E5F1A9576E9049B9ADD87AF51F278BE002780F62
    </agentHash>
    <jobId>116</jobId>
    <timeReceived>2016-01-29 23:20:09.0 EET</timeReceived>
    <output>
      &lt;?xml version="1.0"?&gt;
      &lt;nmaprun scanner="nmap" args="nmap -O -oX - 127.0.0.1"
      start="1453636968" startstr="Sun Jan 24 14:02:48 2016" version="6.47"
      xmloutputversion="1.04"&gt;
      ...
    </output>
  </result>
</results>
```

Οι πιθανές τιμές του attribute status είναι οι εξής:

- "Accepted", αν το αίτημα έγινε δεκτό. Αυτή είναι η μόνη περίπτωση που επιστρέφονται δεδομένα (jobs) στον Android Client.

- "Session Expired", στην περίπτωση που ο χρόνος σύνδεσης έχει παρέλθει.
- "Incorrect Credentials", αν τα στοιχεία χρήστη είναι μη έγκυρα.
- "Registration Pending", αν ο χρήστης δεν έχει εγκριθεί ακόμη.
- "Service Error", στην περίπτωση οποιουδήποτε άλλου σφάλματος.

1.3.7 Ανάθεση εργασιών από απομακρυσμένους χρήστες

```
@POST
@Path("users/")
@Consumes(MediaType.APPLICATION_XML)
@Produces(MediaType.TEXT_PLAIN)
public Response users(UserContainer userContainer) {
    ...
}
```

Η μέθοδος users() ακούει στο URI: <http://localhost:8080/am/client/users> για ένα αίτημα χρήστη με περιεχόμενο εργασίες προς ανάθεση σε Software Agents. Είναι η μόνη υπηρεσία που χρησιμοποιεί την HTTP μέθοδο POST με XML περιεχόμενο. Ένα παράδειγμα περιεχομένου προς κατανάλωση από την υπηρεσία είναι το εξής:

```
<users>
  <user username="Yannis"
password="0BFE935E70C321C7CA3AFC75CE0D0CA2F98B5422E008BB31C00C6D7F1F1C0AD6">
    <job>
      <agentId>3</agentId>
      <timeAssigned>2016-02-03 06:14:58.0 EET</timeAssigned>
      <params>stop 143</params>
      <periodic>false</periodic>
      <period>0</period>
    </job>
  </user>
  <user username="Thanos"
password="5994471ABB01112AFCC18159F6CC74B4F511B99806DA59B3CAF5A9C173CACFC5">
    <job>
      <agentId>3</agentId>
      <timeAssigned>2016-02-03 06:15:00.0 EET</timeAssigned>
      <params>-0 -oX - 127.0.0.1</params>
      <periodic>true</periodic>
      <period>30</period>
    </job>
  </user>
  <user username="Thanos"
password="5994471ABB01112AFCC18159F6CC74B4F511B99806DA59B3CAF5A9C173CACFC5">
    <job>
      <agentId>4</agentId>
      <timeAssigned>2016-02-03 06:18:10.776 EET</timeAssigned>
      <params>exit</params>
      <periodic>false</periodic>
      <period>0</period>
    </job>
  </user>
</users>
```

Όπως φαίνεται στο παράδειγμα, το περιεχόμενο που αποστέλλεται στην υπηρεσία είναι ένα σύνολο χρηστών. Κάθε χρήστης συνοδεύεται από το όνομα και το συνθηματικό του και περιέχει ένα σύνολο εργασιών που επιθυμεί να αναθέσει σε κάποιον Software Agent. Οι πιθανές απαντήσεις της υπηρεσίας είναι:

- "Accepted", στην περίπτωση που η υπηρεσία επεξεργαστεί το περιεχόμενο επιτυχώς.
- "Service Error", στην περίπτωση που συμβεί κάποιο σφάλμα κατά την επεξεργασία.

2 Ο Android Client

2.1 Γραφική διεπαφή

2.1.1 Εγγραφή - Σύνδεση

Κατά την εκκίνησή του, ο Android Client ζητά τα στοιχεία σύνδεσης του χρήστη. Στην περίπτωση επιτυχούς σύνδεσης με την αντίστοιχη υπηρεσία του Aggregator Manager, τα στοιχεία σύνδεσης αποθηκεύονται σε shared preferences έτσι ώστε να είναι διαθέσιμα σε πιθανή μεταγενέστερη εκτέλεση του προγράμματος χωρίς να τα εισάγει ξανά ο χρήστης. Αν ο χρήστης δεν έχει εγγραφεί στον Aggregator Manager, του δίνεται η δυνατότητα μέσω του αντίστοιχου activity, για να συνδεθεί όμως απαιτείται έγκριση από κάποιον administrator όπως είδαμε στις σχετικές υπηρεσίες του Aggregator Manager.

2.1.2 Αποσύνδεση

Όταν βρισκόμαστε στο βασικό activity της εφαρμογής και πατήσουμε το πλήκτρο επιστροφής, επιστρέφουμε στην οθόνη σύνδεσης οπότε ο χρήστης μπορεί να συνδεθεί με διαφορετικά στοιχεία. Δεν γίνεται αποσύνδεση του προηγούμενου χρήστη και διαγραφή του από τα shared preferences παρά μόνο όταν επιλεγεί ρητά από το βασικό activity η αποσύνδεση. Πέρα από την αποσύνδεση με επιλογή του χρήστη, ενδέχεται να έχουμε αποσύνδεση του τρέχοντος χρήστη στην περίπτωση που έχει παρέλθει ο μέγιστος χρόνος σύνδεσης που είναι ρυθμισμένος στον Aggregator Manager, και αυτό μόνο όταν έχει ενεργοποιηθεί η boolean ρύθμιση logOutOnSessionExpiry του Android Client. Αν η ρύθμιση είναι ανενεργή τότε ο χρήστης απλά επιστρέφει στην οθόνη σύνδεσης χωρίς να διαγραφεί από τα shared preferences.

Σχετικές κλάσεις:

- k23b.ac.activities: StartActivity, LoginActivity, RegisterActivity
- k23b.ac.fragments: StartFragment, LoginFragment, RegisterFragment
- k23b.ac.services: UserManager, NetworkManager
- k23b.ac.tasks: UserLoginTask, UserRegisterTask
- k23b.ac.util: Settings

2.1.3 Agents

Η όψη αυτή του Android Client απεικονίζει τους Agents που υπάρχουν καταγεγραμμένοι στη βάση δεδομένων του Aggregator Manager. Επιλέγοντας "Refresh" από το action bar ανανεώνεται η όψη. Στην περίπτωση που δεν υπάρχει σύνδεση με το δίκτυο τη στιγμή που ζητείται ανανέωση, γίνεται αναζήτηση στη βάση δεδομένων του Android Client για τους agents που ελήφθησαν την τελευταία φορά που έγινε ανανέωση ενώ υπήρχε σύνδεση. Με long touch στη γραμμή κάποιου agent ενεργοποιείται το action mode AgentActions και δίνεται η δυνατότητα τερματισμού του σχετικού agent.

Σχετικές κλάσεις:

- k23b.ac.activities: MainActivity
- k23b.ac.fragments: AgentsFragment
- k23b.ac.fragments.actions: AgentActions
- k23b.ac.db.srv: CachedAgentSrv
- k23b.ac.services: UserManager, NetworkManager, JobDispatcher
- k23b.ac.tasks: AgentsReceiveTask

2.1.4 Jobs

Η όψη αυτή απεικονίζει τους agents που υπάρχουν καταγεγραμμένοι στη βάση δεδομένων του Aggregator Manager, και επιλέγοντας κάποιον agent με short touch εμφανίζονται οι εργασίες που έχουν ανατεθεί σε αυτόν κατά το παρελθόν. Στην περίπτωση που ζητηθεί ανανέωση ενώ δεν υπάρχει σύνδεση με το δίκτυο, γίνεται αναζήτηση στην τοπική βάση δεδομένων για agents η jobs που είχαν ληφθεί κατά το παρελθόν. Με long touch στη γραμμή κάποιου agent ενεργοποιείται το action mode JobsActionsAgent που επιτρέπει την ανάθεση μιας καινούριας εργασίας. Για το σκοπό αυτό ξεκινά νέο activity, το AssignJobActivity. Με long touch στη γραμμή κάποιας εργασίας ενεργοποιείται το action mode JobsActionsJob που επιτρέπει τον τερματισμό της εργασίας ή την ανάθεσή της ξανά στον ίδιο agent.

Σχετικές κλάσεις:

- k23b.ac.activities: MainActivity, AssignJobActivity
- k23b.ac.fragments: JobsFragment
- k23b.ac.fragments.actions: JobsActionsAgent, JobsActionsJob
- k23b.ac.db.srv: CachedAgentSrv, CachedJobSrv

- k23b.ac.services: UserManager, NetworkManager, JobDispatcher
- k23b.ac.tasks: AgentsReceiveTask, JobsReceiveTask

2.1.5 Agent Results

Στην όψη αυτή εμφανίζονται οι agents που είναι καταχωρημένοι στη βάση δεδομένων του Aggregator Manager και τα τελευταία αποτελέσματά τους. Με short touch στη γραμμή κάποιου agent εμφανίζονται τα τελευταία αποτελέσματα εργασιών του. Το πλήθος των τελευταίων αποτελεσμάτων καθορίζεται από σχετικό πεδίο στην όψη. Στην περίπτωση που δεν υπάρχει σύνδεση με το δίκτυο, εμφανίζονται μόνο οι agents και όχι τα αποτελέσματά τους αφού αυτά δεν αποθηκεύονται στη βάση δεδομένων λόγω σχετικά μεγάλων απαιτήσεων σε χώρο. Η XML έξοδος κάθε εργασίας παρουσιάζεται στα δεξιά της όψης, αφού πατήσουμε short touch πάνω στη γραμμή κάποιου αποτελέσματος. Για την εμφάνιση της εξόδου, χρησιμοποιείται ένας browser (WebView) τον οποίο διαχειρίζεται η κλάση WebViewManager.

Σχετικές κλάσεις:

- k23b.ac.activities: MainActivity
- k23b.ac.fragments: ResultsAgentFragment
- k23b.ac.db.srv: CachedAgentSrv
- k23b.ac.services: UserManager, NetworkManager
- k23b.ac.tasks: AgentsReceiveTask, ResultsAgentReceiveTask
- k23b.ac.util: WebViewManager

2.1.6 All Results

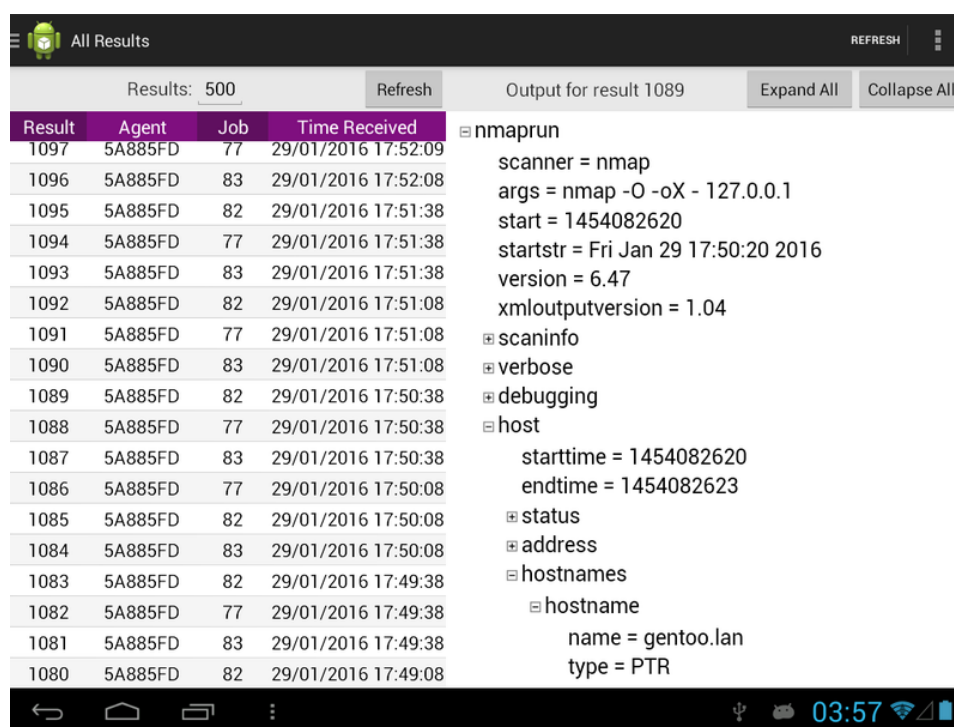
Η όψη είναι όμοια με την Agent Results, με τη διαφορά ότι εμφανίζονται τα τελευταία αποτελέσματα για όλους τους agents.

Σχετικές κλάσεις:

- k23b.ac.activities: MainActivity
- k23b.ac.fragments: ResultsAllFragment
- k23b.ac.services: UserManager, NetworkManager
- k23b.ac.tasks: ResultsAllReceiveTask
- k23b.ac.util: WebViewManager

2.1.7 Η απεικόνιση των αποτελεσμάτων

Για τις όψεις που παρουσιάζουν αποτελέσματα, χρησιμοποιείται η βοηθητική κλάση `WebViewManager`. Κατά την κατασκευή ενός instance αυτής της κλάσης περνιέται ως παράμετρος ένα `WebView` (web browser component) της γραφικής διεπαφής. Μέσω της μεθόδου `displayOutput(String)` μπορεί να περαστεί η XML έξοδος μιας εργασίας προς εμφάνιση στο `WebView`. Η κλάση `WebViewManager` χρησιμοποιεί ένα στιγμιότυπο της κλάσης `XmlTreeConverter` για μετατροπή της γενικής XML εισόδου σε μία ιεραρχική δομή από HTML unordered lists και list items (``, ``). Αφού δημιουργηθεί αυτή η δενδρική δομή, προσαρτάται σε αυτήν μια HTML επικεφαλίδα και ένας επίλογος έτσι ώστε να είναι ένα έγκυρο HTML έγγραφο. Στο έγγραφο αυτό συμπεριλαμβάνεται και κώδικας JavaScript για την εμφάνιση της ιεραρχικής δομής σαν δένδρο με κόμβους που υποστηρίζουν εμφάνιση και απόκρυψη. Μάλιστα προσφέρεται και η δυνατότητα εκτέλεσης από την Android εφαρμογή συναρτήσεων JavaScript που έχουν φορτωθεί στον browser. Παράδειγμα χρήσης αυτής της δυνατότητας είναι οι μέθοδοι `expandAll()` και `collapseAll()` του `WebViewManager` που εμφανίζουν και κρύβουν όλους τους κόμβους του δένδρου αντίστοιχα.

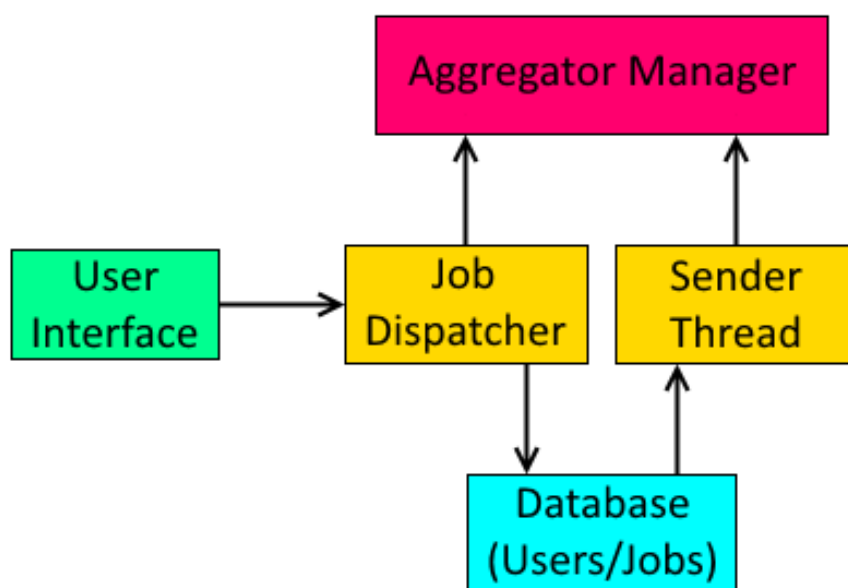


Σχήμα 2: Εμφάνιση αποτελέσματος σε δένδρο με χρήση του `WebViewManager`.

2.2 Υπηρεσίες του Android Client

2.2.1 Ανάθεση Εργασιών

Οι όψεις "Agents" και "Jobs" της γραφικής διεπαφής επιτρέπουν στο χρήστη την ανάθεση νέων εργασιών. Η υποστήριξη της λειτουργικότητας αυτής γίνεται χρησιμοποιώντας τις υπηρεσίες που φαίνονται στο παρακάτω σχήμα:



Σχήμα 3: Οι σχετικές με την ανάθεση εργασιών υπηρεσίες του Android Client.

Σε κάθε αίτημα της γραφικής διεπαφής για την ανάθεση μιας νέας εργασίας, καλείται η μέθοδος `dispatch()` της κλάσης `JobDispatcher` με παράμετρο τον χρήστη που ζητάει την ανάθεση και την ίδια την εργασία. Στην περίπτωση που υπάρχει εκείνη τη στιγμή σύνδεση με το δίκτυο, ο `JobDispatcher` στέλνει κατ' ευθείαν τη νέα εργασία στον `Aggregator Manager`. Αν δεν υπάρχει σύνδεση εκείνη τη στιγμή, τότε η εργασία καταχωρείται στη βάση δεδομένων του `Android Client`. Δημιουργείται ένας χρήστης που αντιστοιχεί στο χρήστη που είναι συνδεδεμένος εκείνη τη στιγμή (αν δεν υπάρχει ήδη), και προστίθεται η εργασία στις εργασίες του χρήστη προς αποστολή. Την αποστολή των εργασιών που έχουν καταχωρηθεί στην τοπική βάση δεδομένων αναλαμβάνει η κλάση `SenderThread` η οποία είναι ένα νήμα το οποίο ενεργοποιείται όταν αποκατασταθεί η σύνδεση με το δίκτυο και μπαίνει σε αναμονή όταν εντοπίσει ότι η σύνδεση έχει χαθεί.

- Job Dispatcher

Ο Job Dispatcher είναι ένα IntentService του οποίου η μέθοδος onHandleIntent() αναλαμβάνει να εντοπίσει αν εκείνη τη στιγμή υπάρχει δίκτυο και αναλόγως είτε να στείλει απ' ευθείας την εργασία στον Aggregator Manager είτε να την αποθηκεύσει στην τοπική βάση δεδομένων. Στην περίπτωση της αποθήκευσης, δημιουργεί εγγραφές στη βάση δεδομένων για τον χρήστη που αναθέτει την εργασία και για την ίδια την εργασία. Για τον εντοπισμό της ύπαρξης ή όχι σύνδεσης με το δίκτυο χρησιμοποιεί την κλάση NetworkManager.

Σχετικές κλάσεις:

- k23b.ac.services: NetworkManager, JobDispatcher
- k23b.ac.db.srv: UserSrv, JobSrv
- k23b.ac.db.dao: UserDao, JobDao

- Sender Thread

Το Sender Thread είναι ένα απλό νήμα το οποίο περιοδικά αδειάζει τη βάση δεδομένων από τα περιεχόμενά της. Αναλαμβάνει να δημιουργήσει ένα πακέτο που περιέχει όλους τους χρήστες που βρίσκονται στη βάση, με τον κάθε χρήστη να περιέχει όλες τις εργασίες που έχει αναθέσει. Αφού δημιουργήσει το πακέτο το στέλνει στον Aggregator Manager. Στην περίπτωση επιτυχούς αποστολής, αναλαμβάνει να διαγράψει από τη βάση δεδομένων τους χρήστες και τις εργασίες που στάλθηκαν επιτυχώς. Το Sender Thread ξεκινά και τερματίζει με τη βοήθεια της υπηρεσίας SenderService, η οποία εξασφαλίζει τον ομαλό τερματισμό του νήματος κατά το τέλος εκτέλεσης της εφαρμογής.

Σχετικές κλάσεις:

- k23b.ac.services: NetworkManager, SenderService, SenderThread
- k23b.ac.db.srv: UserSrv, JobSrv
- k23b.ac.db.dao: UserDao, JobDao

- Database (Users/Jobs)

Η βάση δεδομένων του Android Client έχει υλοποιηθεί χρησιμοποιώντας SQLite. Λόγω της ανάγκης για πρόσβαση από δυο νήματα με το ένα νήμα να δημιουργεί εγγραφές στη βάση και το άλλο να διαγράφει εγγραφές από αυτήν, έχουν αξιοποιηθεί μηχανισμοί συγχρονισμού ώστε οι λειτουργίες να γίνονται σωστά. Έχουν υλοποιηθεί δυο layers όπως και στο δεύτερο παραδοτέο, ένα data access layer για την πρόσβαση στη βάση δεδομένων (Dao) και ένα service layer για πρόσβαση στη βάση ευθυγραμμισμένη με τη λογική του προβλήματος (Srv).

Παρακάτω παρατίθενται τα πρωτότυπα των στατικών μεθόδων του κάθε επιπέδου:

UserSrv:

```
public UserDao create(String username, String password);
public UserDao createUserWithJobs(UserDao user, List<JobDao> jobList);
public UserDao find(String username);
public Set<UserDao> findAll();
public void delete(String username);
public void tryDelete(String username);
```

JobSrv:

```
public JobDao create(String username, long agentId, String params,
    boolean periodic, int period);
public JobDao findById(long jobId);
public Set<JobDao> findAllJobsFromUsername(String username);
public void delete(long jobId);
```

UserDao:

```
public String createUser(String username, String password);
public UserDao findUserByUsername(String username);
public Set<UserDao> findAll();
public void deleteUser(String username);
```

JobDao:

```
public long createJob(String parameters, String username,
    long agentId, Date time_assigned, boolean periodic, int period);
public JobDao findJobById(long jobId);
public Set<JobDao> findAllJobsFromUsername(String username);
public void deleteJob(long jobId);
```

Από τα παραπάνω operations κάποια δεν χρησιμοποιήθηκαν τελικά αλλά παρέμειναν στον πηγαίο κώδικα για πληρότητα.

2.2.2 Asynchronous tasks για επικοινωνία με τον Aggregator Manager

Οι όψεις της εφαρμογής είναι retained fragments που αξιοποιούν τα παρακάτω asynchronous tasks για να παρέχουν τη λειτουργικότητά τους. Όλα τα tasks βασίζονται σε RESTful επικοινωνία μέσω XML περιεχομένου που παρέχεται από το Spring Framework.

- **UserRegisterTask**
Το task αυτό χρησιμοποιείται από το fragment RegisterFragment και επικοινωνεί με την αντίστοιχη υπηρεσία του Aggregator Manager για την εγγραφή νέων χρηστών.
- **UserLoginTask**
Το task αυτό χρησιμοποιείται από το fragment LoginFragment και επικοινωνεί με την αντίστοιχη υπηρεσία του Aggregator Manager για τη σύνδεση χρηστών.
- **AgentsReceiveTask**
Το task αυτό χρησιμοποιείται από τα fragments AgentsFragment, JobsFragment, ResultsAgentFragment. Επικοινωνεί με τη σχετική υπηρεσία του Aggregator Manager για λήψη agents και μετά από επιτυχή λήψη τους αποθηκεύει στη βάση δεδομένων για ενδεχόμενη offline χρήση τους αργότερα.
- **JobsReceiveTask**
Το task αυτό χρησιμοποιείται από το fragment JobsFragment. Επικοινωνεί με τη σχετική υπηρεσία του Aggregator Manager για λήψη εργασιών και μετά από επιτυχή λήψη τις αποθηκεύει στη βάση δεδομένων για ενδεχόμενη offline χρήση τους αργότερα.
- **ResultsAgentReceiveTask**
Το task αυτό χρησιμοποιείται από το fragment ResultsAgentFragment και επικοινωνεί με τη σχετική υπηρεσία του Aggregator Manager για λήψη αποτελεσμάτων κάποιου agent.
- **ResultsAllReceiveTask**
Το task αυτό χρησιμοποιείται από το fragment ResultsAllFragment και επικοινωνεί με τη σχετική υπηρεσία του Aggregator Manager για λήψη αποτελεσμάτων από όλους τους agents.

2.2.3 Αποθήκευση agents και jobs για offline χρήση

Τα asynchronous tasks `AgentsReceiveTask` και `JobsReceiveTask`, πέρα από το να επιστρέφουν τα ληφθέντα δεδομένα στις σχετικές όψεις της εφαρμογής, τα αποθηκεύουν και στην τοπική βάση δεδομένων του Android Client για ενδεχόμενη offline χρήση τους αργότερα. Για το λόγο αυτό χρησιμοποιούνται οι παρακάτω κλάσεις από τα service και data access layers:

`CachedAgentSrv`:

```
public void createOrUpdate(long agentId, String agentHash, Date
    timeAccepted, Date timeJobRequest, Date timeTerminated,
    CachedAgentStatus agentStatus);
public Set<CachedAgentDao> findAll();
```

`CachedJobSrv`:

```
public void createOrUpdate(long jobId, long agentId,
    Date timeAssigned, Date timeSent, String parameters, boolean
    periodic, int period, CachedJobStatus status);
public Set<CachedJobDao> findAllWithAgentId(long agentId);
```

`CachedAgentDao`:

```
public void create(long agentId, String agentHash, Date timeAccepted,
    Date timeJobRequest, Date timeTerminated, CachedAgentStatus
    agentStatus);
public void update(long agentId, String agentHash, Date timeAccepted,
    Date timeJobRequest, Date timeTerminated, CachedAgentStatus
    agentStatus);
public boolean exists(long agentId);
public Set<CachedAgentDao> findAll();
```

`CachedJobDao`:

```
public void create(long jobId, long agentId, Date timeAssigned, Date
    timeSent, String parameters, boolean periodic, int period,
    CachedJobStatus status);
public void update(long jobId, long agentId, Date timeAssigned, Date
    timeSent, String parameters, boolean periodic, int period,
    CachedJobStatus status);
public boolean exists(long jobId);
public Set<CachedJobDao> findAllWithAgentId(long agentId);
```

Κατά τη λήψη των δεδομένων καλείται η μέθοδος `createOrUpdate()` για κάθε ληφθείσα οντότητα (agent η job). Για την ανάκτηση τους κατά την offline χρήση χρησιμοποιούνται οι μέθοδοι `findAll()` για τους agents και `findAllWithAgentId()` για τα jobs.