Understanding pump.io

What is it?

pump.io is an activity stream server
that models a social network
with a simple REST API.

Activities

- Record what happened
- Real things we do
- Digital things we do
- Subject-verb-object structure
- Extensible structure

Subject Verb Object

- Evan posted a photo.
- Karen liked Evan's photo.
- John created a project.
- Bill joined John's project.
- Bill added a task to John's project.
- Maria followed Bill.
- Maria added Bill to her "Friends" list.
- John updated his profile.

Custom activities

- Edward planted strawberries.
- Janet did a 15 minute workout on the elliptical machine.
- Bill ate a breakfast burrito at Celia's.
- Jennifer killed the Dragon King for 18,000 gold.
- Server 075 had a disk error.

Audience

- Who is the activity for?
- Determined by the program and the author
- Some "default"
- Only the audience can see the activity

Audience examples

- All Laura's followers
- John, Alison, and Frank
- People on Dennis's "Family" list
- Members of the "New Auditorium" project
- Everyone in the world

Streams

- Group activities together
- Organized around a theme
- In reverse chronological order (newest first)

Example streams

- The activities Bill did.
- The activities by people Bill follows.
- The activities shared with a group.
- The activities about an image.
- The activities sent directly to Josephine.
- The major activities that Jane did.
- The minor activities by people Karen follows.

Major? Minor?

- Separates "important" from "background" activities
- Major activities generally create new objects.
- Minor activities modify or respond to existing objects.
- Minor activities change the social graph.

REST API

- Consistent JSON representation
- Endpoints for each stream
- All streams are readable
- Some streams are writable
- Streams are filtered by audience

Representing objects

- Unique ID
- Object type
- Other information varies by type

```
{
   "id": "tag:social.example,2013:user:1481",
   "objectType": "person",
   "displayName": "Evan Prodromou",
   "url": "http://social.example/evanp",
   "image": {
      "url": "http://avatar.social.example/evanp.jpg"
   }
}
```

Representing activities

- Actor (is an object)
- Verb
- Object (is an object)
- Summary

```
{
  "id": "urn:uuid:85e80e20-cd1b-11e2-82f1-c8f73398600c",
  "actor": {

    "id": "tag:social.example,2013:user:1481",
    "objectType": "person"
  },
  "verb": "create",
  "object": {
      "id": "http://social.example/note/13",
      "objectType": "note",
      "content": "Hello, World"
  },
  "summary": "Evan created a note."
}
```

REST endpoints

- Inbox: stream of activities by people Alice follows (read-only)
- Outbox: stream of activities with Alice as the subject (read-write)
- Direct inbox: stream of activities with Alice as the audience (read-only)

Activity distribution

- Bill sends a new activity to his outbox endpoint
- pump.io distributes the activity to the right inbox streams based on the audience and the social graph
- Other users get the new activity in their inboxes

Social graph

- Who follows whom
- Who is on a contact list
- Who is a member of a group
- Determined by past activities

Social graph

- Dave follows Helen +
- Helen posts a photo = • the post-photo activity is in Dave's inbox.

Accessing the social graph

- REST API endpoints
- Streams of objects (not activities)

Social graph examples

- All Deirdre's followers.
- Members of Stan's "Coworkers" list.
- Members of the "Softball Team" group.

What can I use it for?

- Web-based social network
- mobile social network
- activity streams for apps
- embedded activity

Learning more

- http://pump.io/
- http://activitystrea.ms/
- https://e14n.com/evan