

Perl OM2

Leif Johansson/OMDC08

Beroenden

- POE
 - <http://poe.perl.org>
- Class::OWL & Class::MOP
 - I CPAN (sedan idag på morgonen)
 - <http://code2.0beta.co.uk/moose/svn/Class-OWL/trunk>
- Net::DDDS (RFC 3401)
 - I CPAN (sedan igår kväll)

POE - "MDB för Perl"

- Perl Object Environment
 - poe.perl.org
- POE::Component::OM2
 - klistret mellan OM2 och POE

Class::OWL

- Baserat på Class::MOP – ”smalltalk för perl”
- Bygga klasser dynamiskt från OWL

Net::OM2

- Inte (ännu) i CPAN
 - `svn://anonsvn.it.su.se/Net-OM2/trunk`
- Stand-alone klient
- Class::OWL-baserade klasser som motsvarar
 - `node.owl`, `miro.owl`, `om2.owl`, `prim.owl`
- REST-transport

POE::Component::OM2

- Inte (ännu) i CPAN
 - <svn://anonsvn.it.su.se/POE-Component-OM2/trunk>
- POE-komponenter:
 - Asynkrona transporter
 - Generell OM2 "nod"
 - Dynamisk metadata
 - Connectorer (PoCo::Connector::SLURPD tex)

Exempel

```
#!/usr/bin/env perl

use POE::Component::OM2 qw/REST/;

POE::Component::OM2->node(
    Name=>'split',
    Destination => sub {
        return 'echo' if $_[0]->isa('Net::OM2::Message');
    }
);

POE::Component::OM2->node(
    Name=>'echo',
    OnMessage => sub {
        warn $_[0]->_rdf->serialize();
    }
);

POE::Kernel->run();
exit;
```

Ladda PoCo::OM2 med
stöd för REST-transport

```
#!/usr/bin/env perl

use POE::Component::OM2 qw/REST/;

POE::Component::OM2->node(
    Name=>'split',
    Destination => sub {
        return 'echo' if $_[0]->isa('Net::OM2::Message');
    }
);

POE::Component::OM2->node(
    Name=>'echo',
    OnMessage => sub {
        warn $_[0]->_rdf->serialize();
    }
);

POE::Kernel->run();
exit;
```

Definiera en nod med
namn 'split'. Varje nod får
en egen URL via REST
<http://localhost:9999/split>

```
#!/usr/bin/env perl

use POE::Component::OM2 qw/RUN/;

POE::Component::OM2->node(
    Name=>'split',
    Destination => sub {
        return 'echo' if $_[0]->isa('Net::OM2::Message');
    }
);

POE::Component::OM2->node(
    Name=>'echo',
    OnMessage => sub {
        warn $_[0]->_rdf->serialize();
    }
);

POE::Kernel->run();
exit;
```

```
#!/usr/bin/env perl
use POE::Component::OM2::EST;

POE::Component::OM2->node(
    Name=>'split',
    Destination => sub {
        return 'echo' if $_[0]->isa('Net::OM2::Message');
    }
);

POE::Component::OM2->node(
    Name=>'echo',
    OnMessage => sub {
        warn $_[0]->_rdf->serialize();
    }
);

POE::Kernel->run();
exit;
```

Det externa namnet blir
split@\$HOSTNAME om
Name inte innehåller '@'

```
#!/usr/bin/env perl

use POE::Component::OM2 qw(
    POE::Component::OM2->node(
        Name=>'split',
        Destination => sub {
            return 'echo' if $_[0]->isa('Net::OM2::Message');
        }
    );

    POE::Component::OM2->node(
        Name=>'echo',
        OnMessage => sub {
            warn $_[0]->_rdf->serialize();
        }
    );

    POE::Kernel->run();
    exit;
```

**Destination används för
dynamisk routing – retur-
värdet blir nästa nod.**

```
#!/usr/bin/env perl

use POE::Component::OM2 qw/REST/;

POE::Component::OM2->node(
    Name=>'split',
    Destinat 'Net::OM2::Message' );
};

POE::Component::OM2->node(
    Name=>'echo',
    OnMessage => sub {
        warn $_[0]->_rdf->serialize();
    }
);

POE::Kernel->run();
exit;
```

**OnMessage är en "sink" - dvs
\$_[0] är meddelandet
men returvärdet ignoreras**

```
#!/usr/bin/env perl

use POE::Component::OM2 qw/REST/;

POE::Component::OM2->node(
    Name=>'split',
    Destination => sub {
        return 'echo';
    }
);

POE::Component::OM2->node(
    Name=>'echo',
    OnMessage => sub {
        warn $_[0]->_rdf->serialize();
    }
);

POE::Kernel->run();
exit;
```

\$_[0]->_rdf returnerar den underliggande RDF-modellen i RDF::Helper som man dock bör undvika att röra direkt

```
#!/usr/bin/env perl

use POE::Component::OM2 qw/REST/;

POE::Component::OM2->node(
    Name=>'split',
    Destination => sub {
        return 'echo' if $_[0]->isa('Net::OM2::Message');
    }
);

POE::Component::OM2->node(
    Name=>'echo',
    OnMessage => sub {
        write($_[0]);
    }
);

POE::Kernel->run();
exit;
```

**Starta POE-processen och kör
så det ryker!**

Metadata & Övervakning

- Metadata
 - <http://localhost:9998/node.owl>
- Övervakning (mycket rudimentär)
 - <http://localhost:9887/>

TODO

- Bättre övervakning