



This document instructs you to set up reThink testbed on Ubuntu. For more information about reThink project please refer to: <https://rethink-project.eu/>.

TODO: more introduction about reThink, hyperties and the testbed.

0. Prerequisites

- Ubuntu: 64-bit, 14.04 Trusty or later.
- A pre-installation of Java & Node.JS environment is recommended.

1. Install Docker

- **Ubuntu:**

1. Install Docker-Engine on Ubuntu following the link:
<https://docs.docker.com/engine/installation/linux/ubuntu/linux/>

Be careful that you should notice the version of your Ubuntu, and follow the instructions dedicated for the version that you are using.

Until now, you should have Docker-Engine installed, which can be verified by running the hello-world image with the command `sudo docker run hello-world`.

2. Install Docker-compose via the link: <https://docs.docker.com/compose/install/>

2. Get Testbed Code from GitHub

- **Ubuntu:**

1. Go to the folder where you want to put the testbed:

```
$ cd ~
```

Go to reTHINK H2020 Project GitHub page: <https://github.com/reTHINK-project/>, and you will see a list of project repositories.

2. Find the “testbeds” repository, git clone the repository into your local folder as below:

```
$ git clone git@github.com:reTHINK-project/testbeds.git
```

- Find the “dev-hyperty-toolkit” and “dev-hyperty” repositories, and git clone them into the PT-node directory:

```
$ cd testbeds/nodes/PT-node/  
$ git clone -b develop git@github.com:reTHINK-project/dev-hyperty-toolkit.git  
$ git clone -b develop git@github.com:reTHINK-project/dev-hyperty.git
```

NOTE

- We have two folders natively under “PT-node”, of which one is for **production** and the other is for **development**
- The **development** folder contains all catalogue_objects
- The **production** folder only contains the specific catalogue_objects for our testbed

3. Testbed Startup

- Ubuntu:**

- Go to development folder and we can start all services on docker-compose.yml without the localhost support:

```
$ sudo docker-compose up
```

Then you should see something like:

```
*****  
THE TOOLKIT IS IN PRODUCTION MODE  
THE LOCAL CATALOGUE WAS NOT ACCESSIBLE, WILL BE USED THE REMOTE  
BASED ON YOUR CONFIGURATION FILE  
*****
```

With the default configuration, the toolkit would be in production mode, and the catalogue is configured to use the remote one with a default domain name of **hybroker.rethink.ptinovacao.pt**.

- Open the Google Chrome browser, check the reTHINK Catalogue Test Page with the link:
<https://catalogue.hybroker.rethink.ptinovacao.pt/>:

reTHINK Catalogue Test Page

Connctected Databases

DB_1017569076	Restart
	Restart all

Check the Hyperty Toolkit Page via: <https://hybroker.rethink.ptinovacao.pt/>:

Hyperty Toolkit

This repository provides required tools to develop Hyperties and help you testing all your hyperties.

LIST OF HYPERTIES <CHOOSE ONE> ▾

NOTE

- Use web console to see ongoing events.
- To run the toolkit in development mode, and catalogue local, you need to make some changes to the compose file.
- To stop the testbed, press **ctrl+c**, then execute command **sudo docker-compose down**.

4. Testbed Configuration

- **Ubuntu:**

1. Go to the “development” folder, and open the “docker-compose.yml” file with an editor:

```
$ cd development
$ gedit docker-compose.yml
```

The compose file provides a way to document and configure all of the application’s service dependencies (msg-node, domain-registry, catalogue components, toolkit, etc);

All this services are compiled in the docker-compose.yml file;

This yml file is split by services, for example:

```
version: '2'
services:
  'msg-node-vertx':
    build: './dev-msg-node-vertx'
    container_name: 'msg-node-vertx'
    networks:
      rethink:
        ipv4_address: 172.18.0.2
    expose:
      - '443'
      - '9090'
```

For a better understanding about this file, please refer to: <https://docs.docker.com/compose/compose-file/>.

2. Use a customized local domain to replace the default one. First you need to register the url you want to use, for example **hjiang-rethink.com**, to the google apps.

Replace the default domain name with your local domain in the docker-compose.yml file:

Find:	hybroker.rethink.ptinovacao.pt	Find
Where:	<current file>	...
Replace:	hjiang-rethink.com	Replace

Set DEVELOPMENT mode of the “hyperty-toolkit” service to “true”:

```
'hyperty-toolkit':
  build: './dev-hyperty-toolkit'
  container_name: 'hyperty-toolkit'
  environment:
    - DEVELOPMENT=true
    - RUNTIME_URL=hyperty-catalogue://catalogue.hjiang-rethink.com/.well-known/runtime/Runtime
    - DOMAIN=hjiang-rethink.com
    - HYPERTY_REPO=/tmp/dev-hyperty
  volumes:
    - ../dev-hyperty:/tmp/dev-hyperty
  networks:
    rethink:
      ipv4_address: 172.18.0.32
  expose:
    - '443'
```

Open /etc/hosts file and add the following lines at the end:

```
172.18.0.2    msg-node.hjiang-rethink.com
172.18.0.4    registry.hjiang-rethink.com
172.18.0.8    catalogue.hjiang-rethink.com
172.18.0.32   hjiang-rethink.com
```

Open `./dev-hyperty-toolkit/system.config.json`, replace “localhost” with your local domain:

```
{
  "develop" : {
    "development": true,
    "runtimeURL": "hyperty-catalogue://catalogue.hjiang-rethink.com/.well-known/runtime/Runtime",
    "domain": "hjiang-rethink.com"
  },
  "production" : {
    "development": false,
    "runtimeURL": "hyperty-catalogue://catalogue.hysmart.rethink.ptinovacao.pt/.well-known/runtime/Runtime",
    "domain": "hysmart.rethink.ptinovacao.pt"
  }
}
```

Open `./dev-hyperty-toolkit/resources/descriptors/ProtoStubs.json`, make default configuration url to your local domain with port 9090:

```
1 {
2   "default": {
3     "cguid": "4-6e7a55d5-3daf-cdc4-63fb-8b2984ada3650",
4     "type": "ProtoStubs",
5     "version": "0.1",
6     "description": "Description of VertxProtoStub",
7     "objectName": "VertxProtoStub",
8     "sourcePackageURL": "/sourcePackage",
9     "sourcePackage": {
10      "sourceCode": "KGZ1bmN0aW9uKGYpe2lmKHR5cGVvZiBleHBvcnRzPT09Im9iamVjdCIImJnR5cGVvZiBtb2R1bGUhPT0idW5kZWZpbmVkIiI7bW9kc",
11      "sourceCodeClassName": "VertxProtoStub",
12      "encoding": "base64",
13      "signature": ""
14    },
15     "language": "javascript",
16     "signature": "",
17     "messageSchemas": "",
18     "configuration": {
19       "url": "wss://msg-node.hjiang-rethink.com:9090/ws"
20     },
21     "constraints": "",
22     "hypertyCapabilities": "",
23     "protocolCapabilities": "",
24     "policies": "",
25     "dataObjects": [],
26     "accessControlPolicy": "somePolicy"
27   },
```

Open `./development/cataloguedatabase/catalogue_objects/protocolstub/VertxProtostub/description.json`, and again replace the default configuration url to your local domain with port 9090:

```
1 {
2   "cguid": "40001",
3   "version": "1.0",
4   "objectName": "VertxProtoStub",
5   "description": "This is a protostub",
6   "language": "Javascript ECMA6",
7   "dataObjects": [],
8   "configuration": {
9     "url": "wss://msg-node.hjiang-rethink.com:9090/ws"
10  },
11  "constraints": [],
12  "messageSchemas": []
13 }
```

3. Rebuild catalogue-databases:

Use `sudo docker images` to list all the docker images, and use `sudo docker rmi` followed with image ID to remove an image, in this case we look for image ID of `development_catalogue-database-ptinovacao`:

```
Haos-MacBook-Pro:~ hjiang$ sudo docker images
REPOSITORY                                TAG                IMAGE ID           CREATED           SIZE
production_catalogue-database-ptinovacao  latest            8f2acc30dff9      About a minute ago  775.1 MB
production_hyperty-toolkit                latest            af809dd0d62e      6 days ago       1.072 GB
rethink/registry-domain-server            latest            4faed9e8c4f7      7 days ago       842.7 MB
node                                       latest            36dc1bb7a52b      2 weeks ago      655.5 MB
rethink/catalogue-broker                  latest            93d8bb3ab543      5 weeks ago      765.9 MB
rethink/catalogue-database-rethinkdefault latest            4198229b0e3b      5 weeks ago      770.6 MB
rethink/msg-node-vertx                    latest            8084f7036cfe      6 weeks ago      1.292 GB
hello-world                               latest            c54a2cc56cbb      5 months ago     1.848 kB
Haos-MacBook-Pro:~ hjiang$ sudo docker rmi 8f2acc30dff9
Untagged: production_catalogue-database-ptinovacao:latest
Deleted: sha256:8f2acc30dff9ec3e1cf7e21ff8a09075d4a001a35b7515b42123594bb0678e0f
Deleted: sha256:06057a65d4b11a642abfb5d077f0b467ba5adfa2bf6b458c3bfd8a0c64c90ed5
Deleted: sha256:ed481fb7527a17ce6a339e7466f32031d62ca0731a531682e7e4d58eb61b76ca
Deleted: sha256:7f69706b6c0468143c33882f9f7514f2039d787caa73ee6971a2ec585f1aafae
```

- Restart the testbed:

```
$ sudo docker-compose up
```

5. Change Default Message Node (optional)

The default message node is Vertx, and this section instructs you to set Nodejs message node as the default.

- Ubuntu:**

- Build Nodejs message node image. Download Nodejs message node source code from reThink Github:

```
$ git clone git@github.com:reTHINK-project/dev-msg-node-nodejs.git
```

Make sure there is a folder called `ssl/` under `dev-msg-node-nodejs/src/` containing three files:

`myCert.crt`:

```
-----BEGIN CERTIFICATE-----
MIIDBjCCAE4CCQDTAKAOrXDViDANBgkqhkiG9w0BAQsFADBFMQswCQYDVQQGEwJB
VTETMBEGA1UECAwK29tZS1TdGF0ZTEhMB8GA1UECgwYSW50ZXJuZXQvV2lkZ2l0
cyBQdHkgTHRkMB4XDTE2MTIyMzA5MjExNVVoXDE2MTIyMzA5MjExNVVowRTELMAK
G A1UEBhMCQVUxExARBgNVBAGMClNvbWUtU3RhdGUxITAFBgNVBAoMGEIudGVybmV0
IFdpZGdpdHMqUHR5IEIxZDCCASIdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
AKILljk9A5Ck80+KwuhHLODoq1eoZbCap72U001x51u1iUwY5CcDe8CZixMQUrV
+KIK3qBi2x3P7CM+HFCA6iGCAUqykW27E5rCGsBgGSI9fxFzESwTR7dYPTgfSdMs
GyFE4jA8GGyXVW/44F/RTw0D1bssmykO+5JNAYk5j3cQRhr14JEKmfz2uSsETR/x
bCyI//2POWCKX5Fa6M0TAOAiokxSXbphKGnLnsWTb0doshtcf30MaE/cKHKIWFAW
BA9ayhH0fvIUIyV0B7z10UzBjBc26H9o71Ec06q1H+q7t6j2AJpN1Na6zBWiaivG
yji+Bw/5QeTNC6qhdDaNvzcCAwEAATANBgkqhkiG9w0BAQsFAAOCAQEAKYvJNAtx
c5eupdwQJiCtiQ7NkuPfiWEm/+NSTqJrLYZNkLKXs11yG+Cxow32vYLe9hjagtBU
nm2ReBQyck9K5fbzxgr4G0nGwJv3/Rf6HV6743KVcwqujtq09Va/cqSIPWPMpnQX
78bU6EFv0GZCGp5pU3058DMQK09WsRWEF9ewTh/Tan1JMWjbJhdPKvvXisKRxmzX
UjokSoXShYHzM4YXImEEW0hkFqkPvwqyx9oww5iWf4pboVonNoR9SsqXIPakUZYt
kjNcCSiUSBqT2CRQiqYnfq7wBU7WyxnjutTsxCiCkXmVFFH580x2qbUrz2CCXT1n
8g+h62g990g3Kw==
-----END CERTIFICATE-----
```

myCert.csr:

```
-----BEGIN CERTIFICATE REQUEST-----
MIICijCCAXICAQAwRTELMAKGA1UEBhMCVUxEzARBgNVBAgMClNvbWUuU3RhdGUx
ITAFBgNVBAoMGEIudGVybWV0IFdpZGdpdHMgUHR5IEIx0ZDCCASIwDQYJKoZIhvcN
AQEBBQADggEPADCCAQoCggEBAKIL1jk9A5Ck80+KwuhHLOdoq1eoZbCAp72U001x
51u1iuwY5CcDe8CZizxMQUrV+KIK3qBi2x3P7CM+HFCA6iGCAUqykw27E5rCGsBq
GSI9fxfzESwTR7dYPTgfSdMsGyFE4jA8GGyXVW/44F/RTw0D1bsmyk0+5JNAYk5
j3cQRhr14JEKmfz2uSsETR/xbCyI//2POWCKX5Fa6M0TA0AiokxSXbphKGNLnsWT
b0doshtcf30MaE/cKHkIWFABWA9ayhHOfvIUyV0B7z10UzBJBc26H9o71Ec06q1
H+q7t6j2AJpN1Na6zBWiaivGyji+Bw/5QeTNC6qhdDanvzcCAwEAAaAAMA0GCSqG
SIB3DQEBCuAA4IBAQB9vw1Matn4XLGUsD2ZHZS4fdvtrjcpdAaBhwtbJIsYRvmG
0ihvLW0pxNHIQuKvYTLA4yaSUCt6MKQfUyYCwZyh5705EodVsK7bKCDiGir7NQG
5SFqWmaWhtKJTexATHCzLj0tZDTfgMqmJJZJhNjgFOES82wUifn/Du+U0UxoQuQtE
23PmdTesTP04T8C9SeBCRJxWginzKT5IHVGX9fk4+HNwAZ5L0qBqZMB5SqTvrQd2
uNvXrewCnJvVTtRzculJxM/OQu/P9wcRsfbReQvaaiRrNlgrLprU5jACK6kYx4E
oBi+EIK9ptKqPyZK641zMzYhoyGla40WRefz7bY
-----END CERTIFICATE REQUEST-----
```

myCert.key:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAoguwOT0DkKTW74rC6Ecs40irV6h1sICnvZQ47XHnW6WK7Bjk
JwN7wJmLPExBStX4ogreoGLbHc/sIz4cUIDqIYIBSrKRbbsTmsIawGoZiJ1/F/MR
LBNHt1g90B9J0ywbJ8TiMDwYbJdVb/jgX9FPDQOVuyybKQ77kk0BiTmPdxBGvXg
kQqZ/Pa5KwRNH/FsLIj//Y85YIpfkVrozRMA4CKiTFJdumEoacuexZNvR2iyG1x/
c4xoT9woeQhYUBYED1rKEc5+8hqjJU4Hv0U5TMEKFzbof2juURw7qqUf6ru3qPYA
mk2U1rrMFYhqK8bKOL4HD/1B5M0LqqF0No2/NwIDAQABAOIBABQoDM+h30emvLGP
OR9beyGez3Ff6QZDepoCTfJ7MKQdu2IS1o5jyIj0kHU3pweABiVh5L9ejJLwwhbA
KxV4c6cnXLHF9IEJvC6L5nf4upefvqdj/MbWvxz3ROYuPiiew+JegyQs1Xzsdhvw
gP1qN8M0px1Z0x0oYXKZUY+YmVhF40bbQSiRMc6YhQf4D5nBH1vkGNiWPaU1vmCT
ph0zaMpY0E4SOCIrdWR9/01KwDCqAi5yICHjiY5EUGrn1C9zCDqQ/MXYw0cT1r
ogoaMLN322R24v2dXT7EnxFJNaR0b0ik1oMttzoKT2kQRqb5Z0qa76Zvpw4euOvL
E5UhrCECgYEAzIGkNb+7PLAG810YOSDhKo2TepKyfcQp+n618G0hieqCPTH02KD
z93bAbuff4Gk4nZqHDqkMZ7N0dp6ITi9E+pVt6ypVUP0QNw+WNmqfE2qCtL1qycf
lWCFIiw0/ce0zhXp8wPo0d20GSvL9zbsBFQXX11tRBx+3WGTyRtVeUsCgYEAytjs
sa3pr48KKCO55CLJUN3piL1LRUK2RqeQF4yz9aQXCjJHxT7SzzuRK0JR16Y8H63v
FUQpUUS0ZwW3jsVm/hqhSQ3MkCnQY9wgiEuP8Q+nVq5zwnim9xU18V0E17fWMkVc
+X0jXfY/DcBL35ysGwz/RG94ucUb9zAO2REBAkUCgYAHy4E7E9bPrUkJRhp7fvpM
WjiLq7YCCjRGS7LV+nopCuX+RWZAdmbTy38Hh+E+giCIyT6uKMYNZiIobF1vWrXK
QApYX+psZ5M7AaowtKXP6x3qh7Rhe7JGfVfVxh2W9r5IuKkCLC/5n9u3geXaXGu0
lq31YnE5wWRMh+WYM/SEFQK8gQCyeeDwiK0nK0TzyzhFNZqx4YWHWqkAMG3svEu
N0eT5xe6ifjqDNexLkTFdvgyXo/KWvGG1370kXNUZnRFDcFdurX+0M0yQWpB2K8
aTeyE2LZ+s6wLC/kEI6Q2pws07CsNkMufsiigeKcN8goh0pJaMTFMwtW8J41h0xb
gwHqvQKBDF70etJoAWYB+YDk3TLogrIrh7hX2GxdFzqgkMy7Nmt4PjwNfyYevgW
A29n/OHwvT3yKNGOZmFkIoIL2pL8z9r6ipo91zxT97zJTBRDSMYBX3nMEk9kDmb3
KLf8bV1qyQ6Pz2x6TzrLYpxEL4zgIgpCQjOVawkCUeD8EwhP7tLg
-----END RSA PRIVATE KEY-----
```

If there is not, create a folder named “ssl” and past the three files in it.

After that, under dev-msg-node-nodejs/, use the command below to build the nodejs message node image:

```
$ sudo docker build -t msg-node-rethink .
```

For detailed information refer to: <https://github.com/reTHINK-project/dev-msg-node-nodejs/tree/master>

When it finishes, check the docker images that you just built: **msg-node-rethink** and **node**:

```
+ ~ sudo docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
development catalogue-database-ptinovacao	latest	059cbc433be2	2 hours ago	775.6 MB
msg-node-rethink	latest	fff300de53b5	2 hours ago	1.104 GB
rethink/catalogue-database-rethinkdefault	dev	5e474263c15d	2 days ago	770.7 MB
rethink/registry-domain-server	latest	4faed9e8c4f7	8 days ago	842.7 MB
vitormsilva/dev-hyperty-toolkit	develop	9ed239017922	3 weeks ago	994.4 MB
rethink/catalogue-broker	latest	93d8bb3ab543	6 weeks ago	765.9 MB
node	6.2	9121f2a78909	6 months ago	659.4 MB

2. Open **development/docker-compose.yml**, change the default protocolstub under “catalogue-broker” service to “NodejsProtoStub”:

```
57 'catalogue-broker':
58   image: rethink/catalogue-broker
59   container_name: 'catalogue-broker'
60   networks:
61     rethink:
62       ipv4_address: 172.18.0.8
63   hostname: 'catalogue.hjiang-rethink.com'
64   command: [-host, 'catalogue.hjiang-rethink.com', -default, protocolstub/NodejsProtoStub]
65   expose:
66     - '443'
67     - '5683'
```

Comment the “msg-node-vertx” service, and add “msg-node-nodejs” service:

```
'msg-node-nodejs':
  image: 'msg-node-rethink:latest'
  container_name: 'msg-node-nodejs'
  # For Easy Development of the msg-node load this volume
  # volumes:
  #   - /path/to/msg-node/repository:/usr/src/app
  environment:
    - url=hjiang-rethink.com # url:domain of CSP where the Messaging Node would be deployed
    - port=9090 # port:port of Messaging Node would be deployed
    - domainRegistryUrl=http://172.18.0.4:4567/ # url:domain of CSP where the registry-domain
    - globalRegistryUrl=http://130.149.22.133:5002/ # url:domain of CSP where the Global Regi
    - logLevel=INFO
    - logDir=/usr/src/app/logs
    - useSSL=1
    - sslCertificate=/usr/src/app/src/ssl/myCert.crt
    - sslPKey=/usr/src/app/src/ssl/myCert.key
  networks:
    rethink:
      ipv4_address: 172.18.0.2
  expose:
    - '443'
    - '9090'

# 'msg-node-vertx':
#   image: 'rethink/msg-node-vertx:develop'
#   container_name: 'msg-node-vertx'
#   environment:
#     - MSG_NODE_CONFIG=env
#     - NODE_DOMAIN=hjiang-rethink.com
#     - NODE_PORT=9090
#     - NODE_REGISTRY_URL=http://172.18.0.4:4567
#     - NODE_GLOBAL_REGISTRY_URL=http://130.149.22.133:5002
#   networks:
#     rethink:
#       ipv4_address: 172.18.0.2
#   expose:
#     - '443'
#     - '9090'
```

If you are using different local domain, please remember to change the url to your own domain.

Open **./development/cataloguedatabase/catalogue_objects/protocolstub/NodeProtostub/description.json**, and again replace the default configuration url to your local domain with port 9090:


```

1  {
2    "cguid": "40002",
3    "version": "1.0",
4    "objectName": "NodejsProtoStub",
5    "description": "This is a protostub",
6    "language": "Javascript ECMA6",
7    "dataObjects": [],
8    "configuration": {
9      "url": "wss://msg-node.hjiang-rethink.com:9090/"
10   },
11   "constraints": [],
12   "messageSchemas": []
13 }
14

```

3. Start your testbed and verify your configuration:

```

+ development git:(master) x sudo docker-compose up
[sudo] password for hjiang:
Creating network "development_rethink" with driver "bridge"
Creating domain-registry
Creating catalogue-broker
Creating hyperty-toolkit
Creating msg-node-nodejs
Creating catalogue-database-ptinovacao
Attaching to domain-registry, catalogue-broker, hyperty-toolkit, msg-node-nodejs, catalogue-database-ptinovacao
hyperty-toolkit    npm info it worked if it ends with ok
hyperty-toolkit    npm info using npm@3.10.9
hyperty-toolkit    npm info using node@v7.2.0
hyperty-toolkit    npm info lifecycle dev-hyperty-toolkit@0.4.1-prestart:dev: dev-hyperty-toolkit@0.4.1
hyperty-toolkit    npm info lifecycle dev-hyperty-toolkit@0.4.1-start:dev: dev-hyperty-toolkit@0.4.1
hyperty-toolkit    > dev-hyperty-toolkit@0.4.1 start:dev /toolkit
hyperty-toolkit    > gulp serve --dev --ENVIRONMENT=browser
hyperty-toolkit    [2016-12-23 14:10:39.075] [INFO] server - [S] HTTP & WS server listening on 9090
msg-node-nodejs

```

You would see a message showing “[INFO] server – [S] HTTP & WS server listening on 9090”, which can also be verified by using command:

```
$ sudo docker ps
```

6. Play with Hyperties

This section shows you some example hyperties. First open <https://hjiang-rethink.com>, and you will see a list of hyperties under the dropdown menu:

The screenshot shows the Hyperty Toolkit web interface. On the left, there is a dropdown menu titled "LIST OF HYPERTIES - CHOOSE ONE -" with a list of hyperties including: BracketsSensorObserver, BracketsSensorReporter, Connector, DTWebRTC, GroupChat, GroupChatManager, HelloWorldObserver, HelloWorldReporter, Location, NotificationsObserver, NotificationsReporter, SurveyObserver, and SurveyReporter. On the right, the console log shows the following messages:

```

Configuration file before: rethink.is:15
  Object {development: true, runtimeURL: "hyperty-catalogue://catalogue.hjiang-rethink.com/.well-known/runtime/Runtime", domain: "hjiang-rethink.com"}
Install configuration: true hjiang-rethink.com hyperty-catalogue://catalogue.hjiang-rethink.com/.well-known/runtime/Runtime rethink.is:94
Browser Request Request.is:6
getting descriptor from: hyperty-catalogue://catalogue.hjiang-rethink.com/.well-known/runtime/Runtime RuntimeCatalogue.is:30
get hyperty-catalogue://catalogue.hjiang-rethink.com/.well-known/runtime/Runtime/version Request.is:31
https://catalogue.hjiang-rethink.com/.well-known/runtime/Runtime/version Request.is:60
get hyperty-catalogue://catalogue.hjiang-rethink.com/.well-known/runtime/Runtime/cguid Request.is:31
https://catalogue.hjiang-rethink.com/.well-known/runtime/Runtime/cguid Request.is:60
adding promise to attach sourcePackage RuntimeCatalogue.is:30
Found database name cache with version no: 1 StorageManager.is:28
got version (1.0) and cguid (54828829) for descriptor RuntimeCatalogue.is:30
hyperty-catalogue://catalogue.hjiang-rethink.com/.well-known/runtime/Runtime
[StorageManager] - getVersion for key 54828829 StorageManager.is:28
storageManager contains saved version that is the same or newer than requested RuntimeCatalogue.is:30

```

Select “HelloWorldObserver”, and log in with a google account. You will get an observer URL started with “hyperty://” as below:

Hyperty Toolkit

This repository provides required tools to develop Hyperties and help you testing all your hyperties.

LIST OF HYPERTIES «CHOOSE ONE»

Hyperty Observer URL: hyperty://hjiang-rethink.com/60128bd6-cf92-4f06-8a00-03d6ec779760

Elements

Console

Sources

Network

Timeline

Profiles

7

X

top

Preserve log

Object {type: "create", from: "runtime://hjiang-rethink.com/4519/registry/", to: "domain://registry.hjiang-rethink.com/", body: Object, id: 5}

Hyperty Schemas ▶ []

Hyperty resources ▶ ["COMMUNICATOR"]

resolve domain://registry.hjiang-rethink.com/

Resolved: msg-node.hjiang-rethink.com/protostub/3640

SandboxBrowser.onPostMessage -> internal (in) from: runtime://hjiang-rethink.com/4519/registry/ to: domain://registry.hjiang-rethink.com/ msg;

Object {type: "create", from: "runtime://hjiang-rethink.com/4519/registry/", to: "domain://registry.hjiang-rethink.com/", body: Object, id: 5}

SandboxBrowser.onPostMessage -> external (out) from: domain://registry.hjiang-rethink.com/ to: runtime://hjiang-rethink.com/4519/registry/ msg;

Object {id: 5, type: "response", from: "domain://registry.hjiang-rethink.com/", to: "runtime://hjiang-rethink.com/4519/registry/", body: Object}

--- Policy Engine ---

Object {id: 5, type: "response", from: "domain://registry.hjiang-rethink.com/", to: "runtime://hjiang-rethink.com/4519/registry/", body: Object}

RegisterHyperty Reply:

Object {id: 5, type: "response", from: "domain://registry.hjiang-rethink.com/", to: "runtime://hjiang-rethink.com/4519/registry/", body: Object}

6: Hyperty url, after register hyperty hyperty://hjiang-rethink.com/60128bd6-cf92-4f06-8a00-03d6ec779760

AppSandboxBrowser.onPostMessage -> internal (in) from: hyperty-runtime://sandbox/external to: hyperty-runtime://sandbox/internal msg;

Object {type: "create", from: "hyperty-runtime://sandbox/external", to: "hyperty-runtime://sandbox/internal", body: Object, id: 1}

SandboxRegistry.create hyperty://hjiang-rethink.com/60128bd6-cf92-4f06-8a00-03d6ec779760

Object {runtimeURL: "runtime://hjiang-rethink.com/4519"}

AppSandboxBrowser.onPostMessage -> external (out) from: hyperty-runtime://sandbox/internal to: hyperty-runtime://sandbox/external msg;

Object {id: 1, type: "response", from: "hyperty-runtime://sandbox/internal", to: "hyperty-runtime://sandbox/external", body: Object}

7: Deploy component status for hyperty: deployed

END

HelloWorldObserver { eventListeners: Object, domain: "hjiang-rethink.com", _objectDescURL: "hyperty-catalogue://catalogue.hjiang-rethink.com/.well-known/dataschema/HelloWorldDataSchema", _syncher: e }

Observer Waiting for Hello!!

Open a new hyperty page using the same address: <https://hjiang-rethink.com>, and select “HelloWorldReporter”.

Log in with another google account other than the previous one. You will then get a reporter URL and be requested to provide the observer URL:

Hyperty Toolkit

This repository provides required tools to develop Hyperties and help you testing all your hyperties.

LIST OF HYPERTIES «CHOOSE ONE»

Hyperty Reporter URL: hyperty://hjiang-rethink.com/80a99b46-4fe4-4eb3-8b86-15837aa692f5

Hyperty URL:

Say Hello

Elements

Console

Sources

Network

Timeline

Profiles

7

X

top

Preserve log

--- Policy Engine ---

Object {type: "create", from: "runtime://hjiang-rethink.com/7039/registry/", to: "domain://registry.hjiang-rethink.com/", body: Object, id: 5}

Hyperty Schemas ▶ []

Hyperty resources ▶ ["COMMUNICATOR"]

resolve domain://registry.hjiang-rethink.com/

Resolved: msg-node.hjiang-rethink.com/protostub/7996

SandboxBrowser.onPostMessage -> internal (in) from: runtime://hjiang-rethink.com/7039/registry/ to: domain://registry.hjiang-rethink.com/ msg;

Object {type: "create", from: "runtime://hjiang-rethink.com/7039/registry/", to: "domain://registry.hjiang-rethink.com/", body: Object, id: 5}

SandboxBrowser.onPostMessage -> external (out) from: domain://registry.hjiang-rethink.com/ to: runtime://hjiang-rethink.com/7039/registry/ msg;

Object {id: 5, type: "response", from: "domain://registry.hjiang-rethink.com/", to: "runtime://hjiang-rethink.com/7039/registry/", body: Object}

--- Policy Engine ---

Object {id: 5, type: "response", from: "domain://registry.hjiang-rethink.com/", to: "runtime://hjiang-rethink.com/7039/registry/", body: Object}

RegisterHyperty Reply:

Object {id: 5, type: "response", from: "domain://registry.hjiang-rethink.com/", to: "runtime://hjiang-rethink.com/7039/registry/", body: Object}

6: Hyperty url, after register hyperty hyperty://hjiang-rethink.com/80a99b46-4fe4-4eb3-8b86-15837aa692f5

AppSandboxBrowser.onPostMessage -> internal (in) from: hyperty-runtime://sandbox/external to: hyperty-runtime://sandbox/internal msg;

Object {type: "create", from: "hyperty-runtime://sandbox/external", to: "hyperty-runtime://sandbox/internal", body: Object, id: 1}

SandboxRegistry.create hyperty://hjiang-rethink.com/80a99b46-4fe4-4eb3-8b86-15837aa692f5

Object {runtimeURL: "runtime://hjiang-rethink.com/7039"}

AppSandboxBrowser.onPostMessage -> external (out) from: hyperty-runtime://sandbox/internal to: hyperty-runtime://sandbox/external msg;

Object {id: 1, type: "response", from: "hyperty-runtime://sandbox/internal", to: "hyperty-runtime://sandbox/external", body: Object}

7: Deploy component status for hyperty: deployed

END

HelloWorldReporter { domain: "hjiang-rethink.com", _objectDescURL: "hyperty-catalogue://catalogue.hjiang-rethink.com/.well-known/dataschema/HelloWorldDataSchema", _syncher: e }

Copy the previous observer URL to the corresponding field, and click “Say Hello”, and on the reporter side, you would see:

Hyperty Toolkit

This repository provides required tools to develop Hyperties and help you testing all your hyperties.

LIST OF HYPERTIES <CHOOSE ONE>

Hyperty Reporter URL: hyperty://hjiang-rethink.com/80a99b46-4fe4-4eb3-8b86-15837aa692f5

Hello URL: hello://hjiang-rethink.com/e1700b53-bc67-4f69-bee1-6c30336b0402

Message:

Say Bye

Elements

Console

Sources

Network

Timeline

Profiles

10

1

X

top

Preserve log

PIPELINE-ERROR: decrypt handshake protocol phase VM102:7443

AppSandboxBrowser.onPostMessage -> internal (in) from: AppSandboxBrowser.js:38

hello://hjiang-rethink.com/e1700b53-bc67-4f69-bee1-6c30336b0402 to:

hyperty://hjiang-rethink.com/80a99b46-4fe4-4eb3-8b86-15837aa692f5 msg:

Object {type: "forward", from: "hello://hjiang-rethink.com/e1700b53-bc67-4f69-bee1-6c30336b0402", to: "hyperty://hjiang-rethink.com/80a99b46-4fe4-4eb3-8b86-15837aa692f5", body: Object, id: 14}

Syncher-RCV:

Object {type: "forward", from: "hello://hjiang-rethink.com/e1700b53-bc67-4f69-bee1-6c30336b0402", to: "hyperty://hjiang-rethink.com/80a99b46-4fe4-4eb3-8b86-15837aa692f5", body: Object, id: 14}

DataObjectReporter-RCV:

Object {type: "forward", from: "hello://hjiang-rethink.com/e1700b53-bc67-4f69-bee1-6c30336b0402", to: "hyperty://hjiang-rethink.com/80a99b46-4fe4-4eb3-8b86-15837aa692f5", body: Object, id: 14}

SUBSCRIPTION-EVENT:

Object {type: "subscribe", url: "hyperty://hjiang-rethink.com/60128bd6-cf92-4f06-8a00-03d6ecf79760", identity: Object}

----- Hello World Reporter received subscription request ----- VM148:107

AppSandboxBrowser.onPostMessage -> external (out) from: AppSandboxBrowser.js:16

hyperty://hjiang-rethink.com/80a99b46-4fe4-4eb3-8b86-15837aa692f5 to:

hello://hjiang-rethink.com/e1700b53-bc67-4f69-bee1-6c30336b0402 msg:

Object {id: 14, type: "response", from: "hyperty://hjiang-rethink.com/80a99b46-4fe4-4eb3-8b86-15837aa692f5", to: "hello://hjiang-rethink.com/e1700b53-bc67-4f69-bee1-6c30336b0402", body: Object}

--- Policy Engine --- VM66:11130

Object {id: 14, type: "response", from: "hyperty://hjiang-rethink.com/80a99b46-4fe4-4eb3-8b86-15837aa692f5", to: "hello://hjiang-rethink.com/e1700b53-bc67-4f69-bee1-6c30336b0402", body: Object} VM66:11131

forward-reply:

Object {id: 14, type: "response", from: "hyperty://hjiang-rethink.com/80a99b46-4fe4-4eb3-8b86-15837aa692f5", to: "hello://hjiang-rethink.com/e1700b53-bc67-4f69-bee1-6c30336b0402", body: Object} VM66:15916

--- Policy Engine --- VM66:11130

Object {id: 7, type: "response", from: "hello://hjiang-rethink.com/e1700b53-bc67-4f69-bee1-6c30336b0402/subscription", to: "runtime://hjiang-rethink.com/4519/sm", body: Object} VM66:11131

resolve runtime://hjiang-rethink.com/4519/sm VM66:14002

Resolved: msg-node.hjiang-rethink.com/protostub/7996 VM66:14030

SandboxBrowser.onPostMessage -> internal (in) from: SandboxBrowser.js:38

hello://hjiang-rethink.com/e1700b53-bc67-4f69-bee1-6c30336b0402/subscription to:

runtime://hjiang-rethink.com/4519/sm msg:

Object {id: 7, type: "response", from: "hello://hjiang-rethink.com/e1700b53-bc67-4f69-bee1-6c30336b0402/subscription", to: "runtime://hjiang-rethink.com/4519/sm", body: Object}

On the observer side, we can see the hello message received:

Hyperty Toolkit

This repository provides required tools to develop Hyperties and help you testing all your hyperties.

LIST OF HYPERTIES <CHOOSE ONE>

Hyperty Observer URL: hyperty://hjiang-rethink.com/60128bd6-cf92-4f06-8a00-03d6ecf79760

Hello buddy!!

Elements

Console

Sources

Network

Timeline

Profiles

10

1

X

top

Preserve log

resolved hyperty://hjiang-rethink.com/80a99b46-4fe4-4eb3-8b86-15837aa692f5 VM177:14002

Resolved: msg-node.hjiang-rethink.com/protostub/3640 VM177:14030

SandboxBrowser.onPostMessage -> internal (in) from: SandboxBrowser.js:38

hyperty://hjiang-rethink.com/60128bd6-cf92-4f06-8a00-03d6ecf79760 to:

hyperty://hjiang-rethink.com/80a99b46-4fe4-4eb3-8b86-15837aa692f5 msg:

Object {type: "handshake", to: "hyperty://hjiang-rethink.com/80a99b46-4fe4-4eb3-8b86-15837aa692f5", from: "hyperty://hjiang-rethink.com/60128bd6-cf92-4f06-8a00-03d6ecf79760", body: Object, id: 11}

SandboxBrowser.onPostMessage -> external (out) from: SandboxBrowser.js:16

hello://hjiang-rethink.com/e1700b53-bc67-4f69-bee1-6c30336b0402/subscription to:

runtime://hjiang-rethink.com/4519/sm msg:

Object {id: 7, type: "response", from: "hello://hjiang-rethink.com/e1700b53-bc67-4f69-bee1-6c30336b0402/subscription", to: "runtime://hjiang-rethink.com/4519/sm", body: Object}

--- Policy Engine --- VM177:11130

Object {id: 7, type: "response", from: "hello://hjiang-rethink.com/e1700b53-bc67-4f69-bee1-6c30336b0402/subscription", to: "runtime://hjiang-rethink.com/4519/sm", body: Object} VM177:11131

reporter-subscribe-response:

Object {id: 7, type: "response", from: "hello://hjiang-rethink.com/e1700b53-bc67-4f69-bee1-6c30336b0402/subscription", to: "runtime://hjiang-rethink.com/4519/sm", body: Object} VM177:16353

--- Policy Engine --- VM177:11130

Object {id: 1, type: "response", from: "runtime://hjiang-rethink.com/4519/sm", to: "hyperty://hjiang-rethink.com/60128bd6-cf92-4f06-8a00-03d6ecf79760", body: Object} VM177:11131

AppSandboxBrowser.onPostMessage -> internal (in) from: AppSandboxBrowser.js:38

runtime://hjiang-rethink.com/4519/sm to: hyperty://hjiang-rethink.com/60128bd6-cf92-4f06-8a00-03d6ecf79760 msg:

Object {id: 1, type: "response", from: "runtime://hjiang-rethink.com/4519/sm", to: "hyperty://hjiang-rethink.com/60128bd6-cf92-4f06-8a00-03d6ecf79760", body: Object}

subscribe-response:

Object {id: 1, type: "response", from: "runtime://hjiang-rethink.com/4519/sm", to: "hyperty://hjiang-rethink.com/60128bd6-cf92-4f06-8a00-03d6ecf79760", body: Object} VM264:31

t {f syncher: e, url: "hello://hjiang-rethink.com/e1700b53-bc67-4f69-bee1-6c30336b0402", schema: "hyperty-catalogue://catalogue.hjiang-rethink.com/.well-known/dataschema/HelloWorldDataSchema", _status: "on", _syncObj: e-} VM264:116

Hello event received: VM265:37

Object {name: "hello", hello: "Hello buddy!!", reporter: "hyperty://hjiang-rethink.com/80a99b46-4fe4-4eb3-8b86-15837aa692f5", schema: "hyperty-catalogue://catalogue.hjiang-rethink.com/.well-known/dataschema/HelloWorldDataSchema"}

We can continue to send “Say Bye” message from the reporter side and see similar results. Until now, we have basically set up a functional local reThink testbed, based on which we can develop interesting hyperties, architectures, and policy engine, etc.

7. Development