

Pomelo

Node.js

node.js npm

. Node.js .

npm .

```
$ npm install pomelo -g
```

git

Hello World

Hello World

```
$ pomelo init ./HelloWorld
```

```
$ mkdir HelloWorld
$ cd HelloWorld
$ pomelo init
```

HelloWorld

```
$ sh npm-install.sh
```

```
$ cd game-server
$ pomelo start --daemon
```

-daemon

command

가.

...

```
$ cd web-server
$ node app
```

node가 forever

<http://localhost:3001>

. 가

Test Game Server

'game server is ok'

Gate

Frontend load balancing

가 gate

gate.gateHandler

gate

load balancing

connector

connector

```

// gateHandler.js
var connectors = this.app.getServersByType('connector');

// select connector, because more than one connector existed.
var dispatcher = require('/util/dispatcher');
var res = dispatcher.dispatch(uid, connectors);
// ...

// dispatcher.js
var crc = require('crc');
module.exports.dispatch = function(key, list) {
  var index = Math.abs(crc.crc32(key)) % list.length;
  return list[index];
};

```

- [Server scalability](#)

Gate

connector

servers.json

port

. clientPort

Connector

Client connection

routing

client

push

connector

(session.bind())

backend

client

가

backend

Connector

client

listen

clientPort

, backend server

port

Gate

, Connector

- [Getting source code & installation](#)

Application

Application

connector

logic

frontend

가

client

client

service

application

backend

gate,

```

client      frontend      client      service
Backend    rpc              client
  clientPort      . Port              args

```

```

"game": [
  {
    "id": "game-server-1",
    "host": "127.0.0.1",
    "port": 3250,
    "args": " --debug=32315 "
  }
]

```

32315

Master

RPC invocation

```

Interprocess communication      RPC      Interprocess communication      frontend
  backend      ,      backend

```

```
app.rpc.game.gameRemote.request( routeParam, args, cb );
```

```

game      , gameRemote      remote      , request      . pomelo
app/servers      game      (      ) game/remote
gameRemote.js      request      . args      cb      rpc call
      2      1      . (cb
      null      cb      .)

```



args ... Converting circular structure to JSON 가

```

routeParam      route      session
user      id      ,      .((      ) Route
      route

```

```

// app.js
var router = function(routeParam, msg, context, cb) {
  var gameServers = app.getServersByType('game');

```

```

var id = gameServers[routeParams% gameServers.length].id;
cb(null, id);
}

app.configure('production|development', function() {
  app.route('game', router); // router
});

```

servers.json game 가

```

"game": [
  {"id": "game-server-1", "host": "127.0.0.1", "port": 7000},
  {"id": "game-server-2", "host": "127.0.0.1", "port": 7001}
]time

```

app.route()

- [RPC Invocation](#)
- [Rpc framework](#)

Route, router

Route 가
. Javascript 가

```

window.pomelo.request( 'game.gameHandler.request', {
  protocol: ...
}, function( result ) {
  ...
} );

```

game , gameHandler handler , request . pomelo
 app/servers game handler gameHandler.js
 request

```
channel.pushMessage('onChat', param);
```

channel pomelo app channelService channel
 param param

```

pomelo.on('onChat', function (msg) {
});

```


Filter

Handler

Handler client logic . 가 .

```
handler.methodName = function(msg, session, next) {
  // ...
}
```

handler remote 가 remote rpc invocation

Error handler

Component

Protocol

Pomelo socket.io 0.3 TCP websocket binary
.Connector 가 .
socket.io siconnector TCP websocket hybridconnector가 . (
connector game-server/node_modules/pomelo/lib/connectors)

```
// app.js
app.configure('production|development', 'connector', function(){
  app.set('connectorConfig', {
    connector: pomelo.connectors.hybridconnector,
    heartbeat: 3,
    useDict: true,
    useProtobuf: true,
    checkClient: function(type, version) {
      // check the client type and version then return true or false
    },
    handshake: function(msg, cb){
      cb(null, { /* message pass to client in handshake phase */ });
    }
  });
});
```

- [Pomelo 0.3 new features](#)

Pomelo protocol

- [Pomelo communications protocol](#)

servers.json

cpu

0.5 () cpu

```

{
  "development":{
    "connector":[
      {"id":"connector-server-1", "host":"127.0.0.1", "port":4050,
"clientPort": 3050, "frontend": true, "cpu": 2}
    ]
    "chat":[
      {"id":"chat-server-1", "host":"127.0.0.1", "port":6050, "cpu":
1}
    ]
    "gate":[
      {"id": "gate-server-1", "host": "127.0.0.1", "clientPort": 3014,
"frontend": true, "cpu": 3}
    ]
  }
}

```

0.6 hybridconnector pomelo.init
encrypt 가

```

pomelo.init({
  host:'127.0.0.1',
  port:3014,
  encrypt:true
}, function() {
// do something connected
});

```

app.js connectorConfig useCrypto 가

```

app.set('connectorConfig', {
  connector: pomelo.connectors.hybridconnector,
  heartbeat: 3,
  useDict: true,
  useProtobuf: true,
  useCrypto: true
});

```

Pomelo

- 1. Gate handler connector host, port
 - 2. Gate handler session connector
 - 3. connector host, port connector handler
 - 4. Connector handler session bind
- router route connector handler session

- Pomelo Wiki
- Promelo Wiki (Tutorial 가 ?)

From:
<http://obg.co.kr/doku/> - **OBG WiKi**

Permanent link:
<http://obg.co.kr/doku/doku.php?id=programming:javascript:nodejs:pomelo>

Last update: **2020/11/29 14:09**

