**API between Modules** (2/28/14)

**Info about gevent:**

they just passing **normal data types - int, string, ....**

#.gevent.subscribe( $panels, 'widget-reject', **onWidgetReject** );

$.gevent.**publish( 'widget-reject'**, 23 );

var **onWidgetReject** = function ( event, reject\_count ) {

$( this ).text( reject\_count );

};

**Info about emit and on:**

they passing JSON object or array of JSON objects.

I **emit one object { }** in bet

But **got in model array of [ { }** ]

**1) shell to model**

**8 suscribes:** 'ibet-login', 'ibet-logout', 'ibet-players-ui', 'ibet-waiting-ui', 'play-wait', 'played', 'no-players', 'no-waiting'

**1.1) call to model**

- onTapAcct():

ibet.model**.people.login**( user\_name );

- onTapPlay():

ibet.model.**people.play(** current\_user.name );

**1.2) waiting for model**

- onLogin( event, login\_user ) for 'ibet-login'

- onPlaysers( event, result-list ) for 'update-players-ui'

- onWaiting( event, result-list ) for 'update-waiting-ui'

- onPlaywait(event ) for 'play-wait'

- noPlayers( event, result-list ) for 'no-players'

- noWaiting( event, result-list ) for 'no-waiting'

- onLogout( event, logout\_user) for 'ibet-logout'

**2) model to shell**

**2.1) call from model**

**-** people.login ( name )

- people.logout ()

- people.play( name )

**2.2) public to shell**

- completeLogin( user\_list )

publish( 'ibet-login', [ stateMap.user ] );

- people.logout:

publish( 'ibet-logout', [ user ] );

- updatePlayers ( playerslist )

publish( 'ibet-players-ui, playerList );

- updateWaiting ( waiting )

publish( 'ibet-waiting-ui, waitingList );

- onPlaywait()

publish( 'play-wait' );

- onPlayed( winner, amount )

publish( 'ibet-played', result\_map = { winner : winner, amount : amount } );

- noPlayers()

publish( 'no-players' );

- noWaiting()

publish( 'no-waiting' );

**3) model to bet**

**3.1) emit to bet**

- people.login:

**emit( 'adduser',**

{cid:stateMap.user.cid, css\_map:stateMap.user.css\_map,

name:stateMap.user.name });

- people.play:

emit( 'play', name );

**3.2) waiting for bet**

- people.login:

on( 'userupdate', copmleteLogin );

on( 'updateplayers', updatePlayers );

on( 'updatewaiting', updateWaiting );

on( 'noplayers', noPlayers );

on( 'nowaiting', noWaiting );

- people.play( name )

on( 'playwait', onPlaywait );

on ( 'played', onPlayed );

onPlayed( **result\_map** ); {result\_map.winner, resultmap.amount}

publish( 'ibet-played', {winner : winner, amount:amount} );

**4) bet to model**

**4.1) waiting on model**

- betObj.on( 'connection' ).**on( 'adduser'**)

crud.read(..)

signIn( io, result\_map, socket )

- betObj.on( 'connection' ).**on( 'play'**)

**4.2) emit to model**

betObj.on( 'connection' ).**on( 'adduser'**)

**- crud.read(..)**

**signIn(** io, result\_map, socket )

emit( 'userupdate', user\_map );

listbroadcast( io, 'updateplayers', playerList )

emit( event, list );

listbroadcast( io, 'updatewaiting', waitingList )

emit( 'noplayers' );

emit( 'nowaiting' );

**- crud.construct(** ...)

emit( 'userupdate', result\_map );

emit( ' noplayers' );

emit( 'nowaiting' );

betObj.on( 'connection' ).**on( 'play'**)

emit( 'playwait' );

emit( 'noplay' );

emit( 'played' , **result\_map**= {winner:winner, amount:amount} );

emit( 'noplayers' );

emit( 'nowaiting' );

listbroadcast( io, 'updateplayers', playerList )

emit( event, list );

listbroadcast( io, 'updatewaiting', waitingList )