

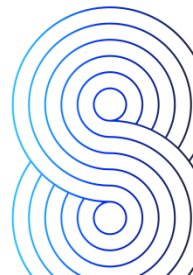
Tizen.IoTivity.Connectivity Server

Jin Yoon

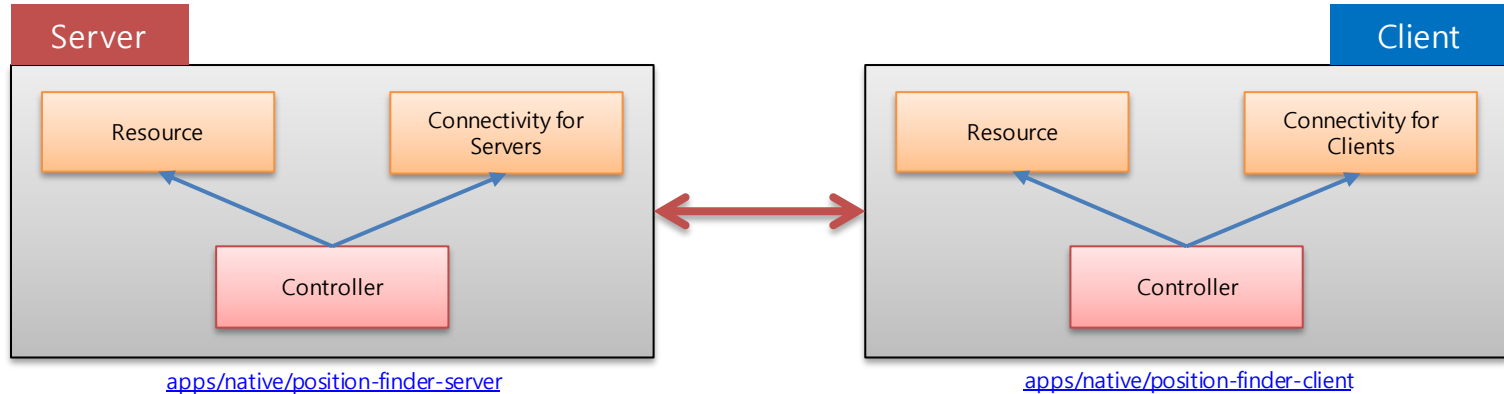


RCC Design Pattern

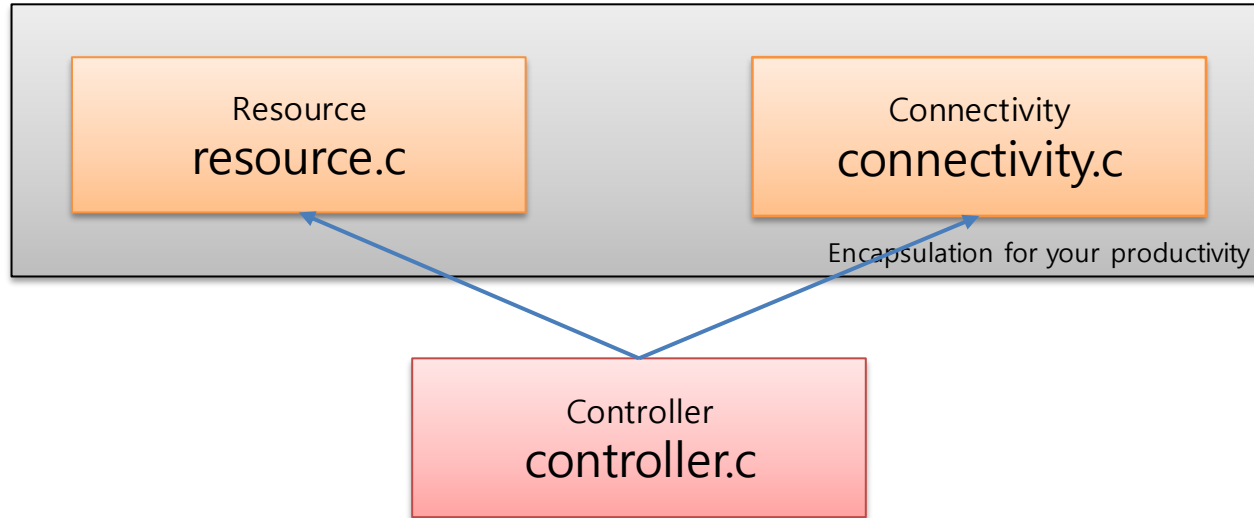
- **Resource**
 - represents real state content
 - collects data
 - Sensors, LED, ...
- **Connectivity**
 - connectivity of devices
 - integrated into communication networks
 - local network or cloud network
- **Controller**
 - controls resources and connectivity
 - mainloop



RCC between Server & Client



RCC Design Pattern for Server



Dependency

App

Connectivity (RCC Pattern)

Platform

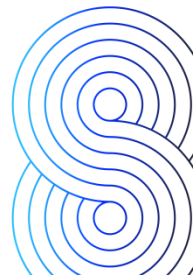
Tizen IoTCon
Tizen Developer Site

IoTivity
<https://www.iotivity.org/>



Connectivity APIs

- `connectivity_init()`
- `connectivity_fini()`
- `connectivity_set_resource()`
- `connectivity_unset_resource()`
- `connectivity_notify_bool()`
- `connectivity_notify_int()`
- `connectivity_notify_double()`



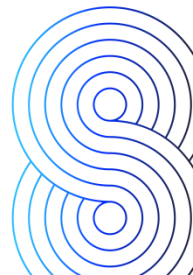
Server

initialize



connectivity_init() & connectivity_fini() in controller_internal.c

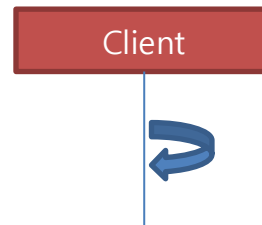
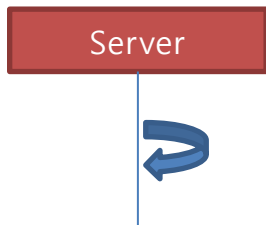
We hide them



server

```
/* Hide in controller_internal.c */  
void controller_init_internal_functions(void)  
{  
    connectivity_init();  
}  
  
void controller_fini_internal_functions(void)  
{  
    _I("Terminating...");  
    resource_close_all();  
    connectivity_fini();  
}
```

Initialize



int	iotcon_initialize (const char *file_path)	/* The path of SVR(Secure Virtual Resources) DB */ /* CBOR (Concise Binary Object Representation)-format must already exist in file_path. */ /* https://wiki.iotivity.org/security_resource_manager */
int	iotcon_deinitialize (void)	
int	iotcon_get_timeout (int *timeout_seconds)	/* Gets the timeout seconds of asynchronous API. Default is 30 sec */
int	iotcon_set_timeout (int timeout_seconds)	
int	iotcon_polling_get_interval (int *interval)	/* Gets the polling interval(milliseconds) of IoTCon. Default is 100 ms */
int	iotcon_polling_set_interval (int interval)	

server

```
int connectivity_init(const char *device_name)
{
    int ret = -1;

    ret = iotcon_initialize("cbor.dat");
    retv_if(IOTCON_ERROR_NONE != ret, -1);

    ret = iotcon_set_device_name(device_name);
    goto_if(IOTCON_ERROR_NONE != ret, error);

    return 0;

error:
    iotcon_deinitialize();
    return -1;
}
```

client

```
int connectivity_init(void)
{
    int ret = -1;

    ret = iotcon_initialize("cbor.dat");
    if (IOTCON_ERROR_NONE != ret) {
        _E("iotcon_initialize() Fail(%d)", ret);
        return -1;
    }

    return 0;
}
```

Server

resource



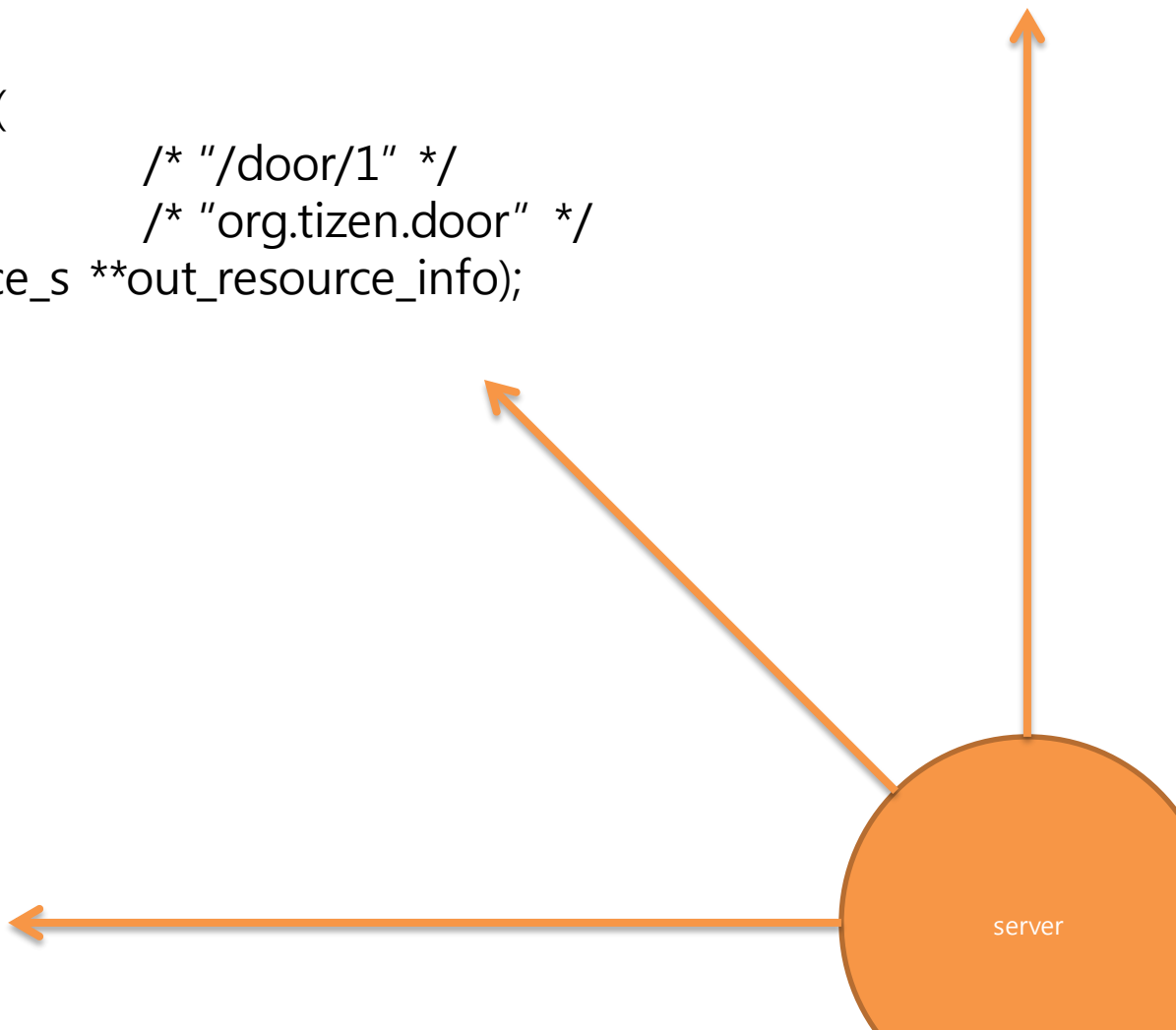


Set Resources

connectivity_set_resource() & connectivity_unset_resource()



```
int connectivity_set_resource(  
    const char *path,          /* "/door/1" */  
    const char *type,          /* "org.tizen.door" */  
    connectivity_resource_s **out_resource_info);
```



server

```
static bool service_app_create(void *data)
{
    app_data *ad = data;
    int ret = -1;

    /**
     * No modification required!!!
     * Access only when modifying internal functions.
     */
    controller_init_internal_functions();

    /**
     * Create a connectivity resource and registers the resource in server.
     */
    ret = connectivity_set_resource("/door/1", "org.tizen.door", &ad->resource_info);
    if (ret == -1) _E("Cannot broadcast resource");

    /**
     * Creates a timer to call the given function in the given period of time.
     * In the control_sensors_cb(), each sensor reads the measured value or writes a specific value to the sensor.
     */
    ad->getter_timer = ecore_timer_add(0.5f, control_sensors_cb, ad);
    if (!ad->getter_timer) {
        _E("Failed to add infrared motion getter timer");
        return false;
    }

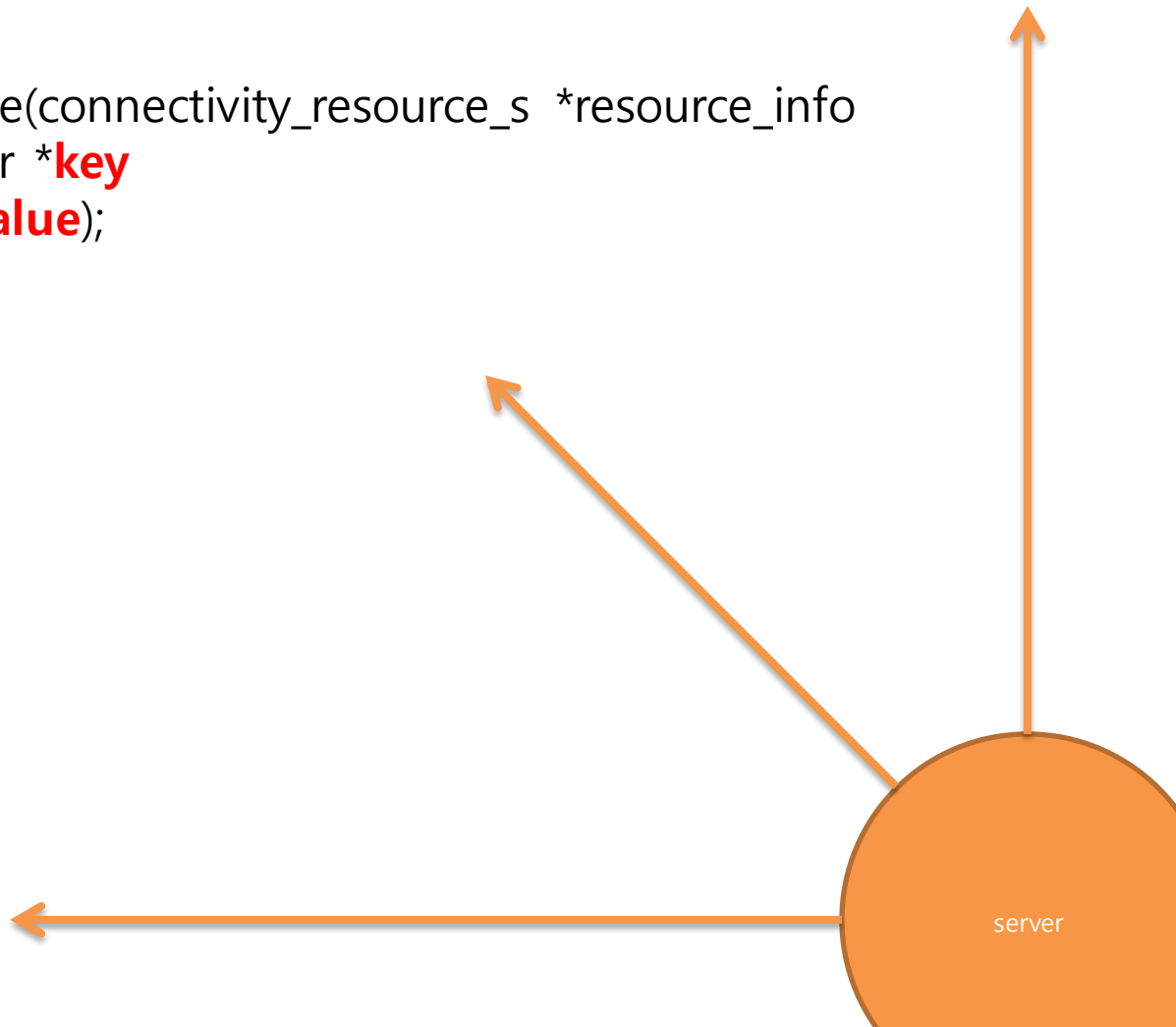
    return true;
}
```

Server
notify

SOSCON



```
int connectivity_notify_double(connectivity_resource_s *resource_info  
    , const char *key  
    , double value);
```



server

```
static void _ultrasonic_sensor_read_cb(float value, void *data)
{
    _I("Distance : %.2fcm", value);

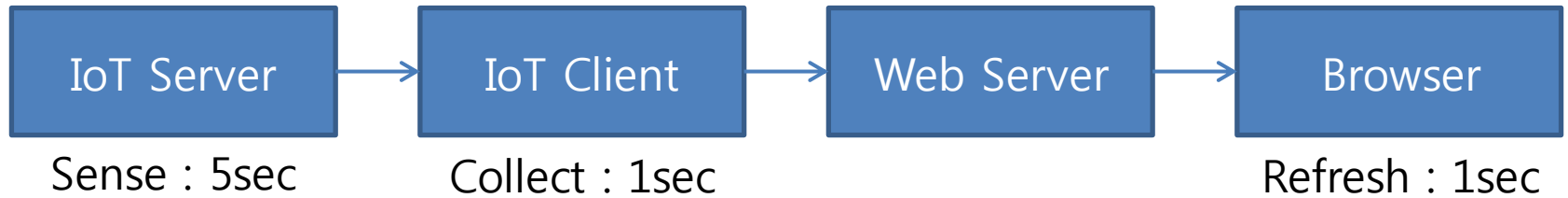
    if (connectivity_notify_double(ad->resource_info, "distance", value) == -1)
        _E("Cannot notify message");
}

static Eina_Bool _control_sensors_cb(void *data)
{
    app_data *ad = data;

    if (resource_read_ultrasonic_sensor(TRIG_PIN_NUMBER, ECHO_PIN_NUMBER, _ultrasonic_sensor_read_cb, NULL) == -1)
        _E("Failed to get a distance from Ultrasonic sensor");

    return ECORE_CALLBACK_RENEW;
}
```

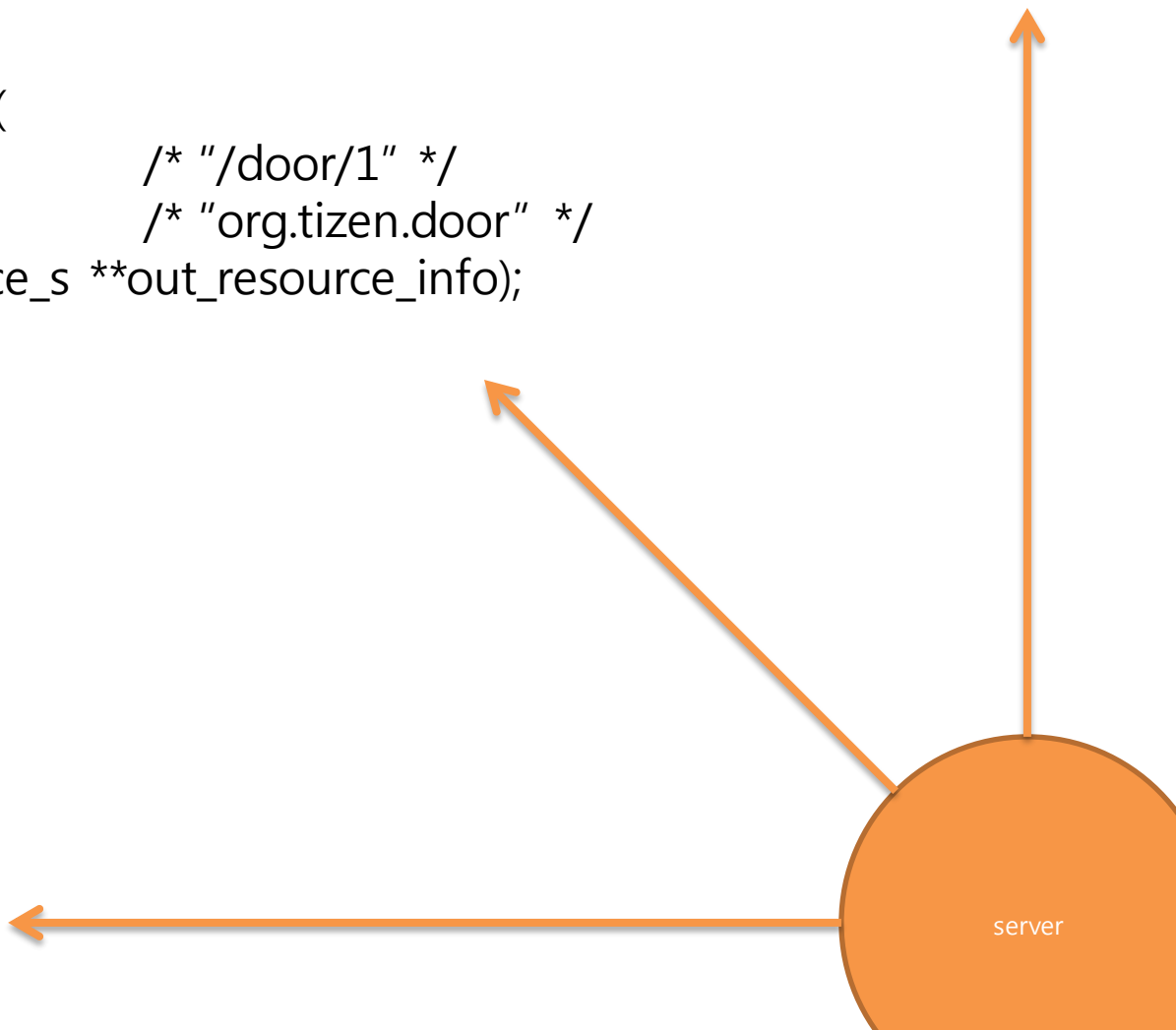
Propagation delay



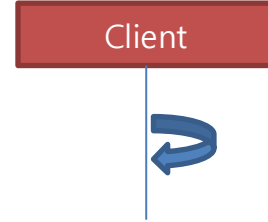
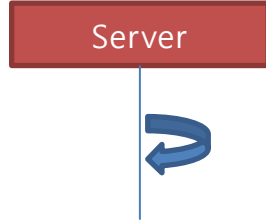
Server resource



```
int connectivity_set_resource(  
    const char *path,          /* "/door/1" */  
    const char *type,          /* "org.tizen.door" */  
    connectivity_resource_s **out_resource_info);
```



Resource types



```
int      iotcon_resource_types_create (iotcon_resource_types_h *types)
int      iotcon_resource_types_destroy (iotcon_resource_types_h types)
int      iotcon_resource_types_clone (iotcon_resource_types_h src, iotcon_resource_types_h *dest)

int      iotcon_resource_types_add (iotcon_resource_types_h types, const char *type)
/* ex) ret = iotcon_resource_types_add(resource_types, "org.tizen.door"); */
int      iotcon_resource_types_remove (iotcon_resource_types_h types, const char *type)

int      iotcon_resource_types_foreach (iotcon_resource_types_h types, iotcon_resource_types_foreach_cb cb, void *user_data)
```

Resource interfaces



- It provides view about how the response will appear.

Name	Interface	Explanation
Baseline	oic.if.baseline	Includes all information about the resource, including meta-data and collection information. This is the default interface type.
Linked List	oic.if.ll	Includes only the collection information about the resource. This is the default interface type for /oic/res.
Batch	oic.if.b	Allows for the aggregation of interaction with all resources. Each resource will be interacted with separately, but their responses will be aggregated.

- It provides **read/write permission** details for the resource.

Name	Interface	Explanation
Read	oic.if.r	Allows values to be read.
Read Write	oic.if.rw	Allows values to be read and written.
Actuator	oic.if.a	Allows creating, updating, and retrieving actuator values.
Sensor	oic.if.s	Allows the sensor values to be read.

Resource interfaces



```
int      iotcon_resource_interfaces_create (iotcon_resource_interfaces_h *ifaces)
int      iotcon_resource_interfaces_destroy (iotcon_resource_interfaces_h ifaces)
int      iotcon_resource_interfaces_clone (iotcon_resource_interfaces_h src, iotcon_resource_interfaces_h *dest)

#define IOTCON_INTERFACE_DEFAULT  "oic.if.baseline"
#define IOTCON_INTERFACE_GROUP    "oic.mi.grp"      /* manipulate (GET, PUT, POST) a group of remote resources. */
#define IOTCON_INTERFACE_LINK     "oic.if.ll"       /* list the references to other resources contained in a resource. */
#define IOTCON_INTERFACE_READONLY "oic.if.r"        /* applied to a resource to GET only. */
#define IOTCON_MULTICAST_ADDRESS  NULL

int      iotcon_resource_interfaces_add (iotcon_resource_interfaces_h ifaces, const char *iface)
int      iotcon_resource_interfaces_remove (iotcon_resource_interfaces_h ifaces, const char *iface)

int      iotcon_resource_interfaces_foreach (iotcon_resource_interfaces_h ifaces, iotcon_resource_interfaces_foreach_cb cb, void *user_data)
```

Create Resource



```
int      iotcon_resource_create (const char *uri_path,                /* format : /a/light */
                                iotcon_resource_types_h res_types,
                                iotcon_resource_interfaces_h ifaces,
                                uint8_t policies,                    /* IOTCON_RESOURCE_DISCOVERABLE | IOTCON_RESOURCE_OBSERVABLE */
                                iotcon_request_handler_cb cb,
                                void *user_data,
                                iotcon_resource_h *resource_handle)
int      iotcon_resource_destroy (iotcon_resource_h resource_handle)
int      iotcon_resource_bind_type (iotcon_resource_h resource_handle, const char *resource_type)
int      iotcon_resource_bind_interface (iotcon_resource_h resource, const char *iface)
int      iotcon_resource_set_request_handler (iotcon_resource_h resource, iotcon_request_handler_cb cb, void *user_data)
int      iotcon_resource_get_uri_path (iotcon_resource_h resource, char **uri_path)
int      iotcon_resource_get_types (iotcon_resource_h resource, iotcon_resource_types_h *types)
int      iotcon_resource_get_interfaces (iotcon_resource_h resource, iotcon_resource_interfaces_h *ifaces)
int      iotcon_resource_get_policies (iotcon_resource_h resource, uint8_t *policies)
```

```
int connectivity_set_resource(const char *uri_path, const char *type, connectivity_resource_s **out_resource_info)
{
    ...
    iotcon_resource_types_create(&resource_types);
    iotcon_resource_types_add(resource_types, type);
    iotcon_resource_interfaces_create(&ifaces);
    iotcon_resource_interfaces_add(ifaces, IOTCON_INTERFACE_DEFAULT);
    iotcon_resource_interfaces_add(ifaces, IOTCON_INTERFACE_BATCH);

    policies =
        IOTCON_RESOURCE_DISCOVERABLE |
        IOTCON_RESOURCE_OBSERVABLE |
        IOTCON_RESOURCE_SECURE;

    ret = iotcon_resource_create(uri_path,
                                resource_types,
                                ifaces,
                                policies,
                                _request_resource_handler,
                                resource_info,
                                &resource_info->res);
    ...
}

static void _request_resource_handler(iotcon_resource_h resource, iotcon_request_h request, void *user_data)
{
    ...
    iotcon_request_get_host_address(request, &host_address);

    _handle_query(request);
    _handle_request_by_crud_type(request, resource_info);
    _handle_observer(request, resource_info->observers);
    ...
}
```

I am a server

Child resource



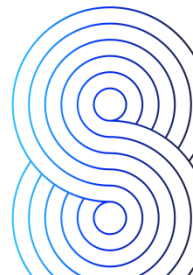
/* interfaces as batch */

```
int      iotcon_resource_bind_child_resource (iotcon_resource_h parent, iotcon_resource_h child)
int      iotcon_resource_unbind_child_resource (iotcon_resource_h parent, iotcon_resource_h child)
```

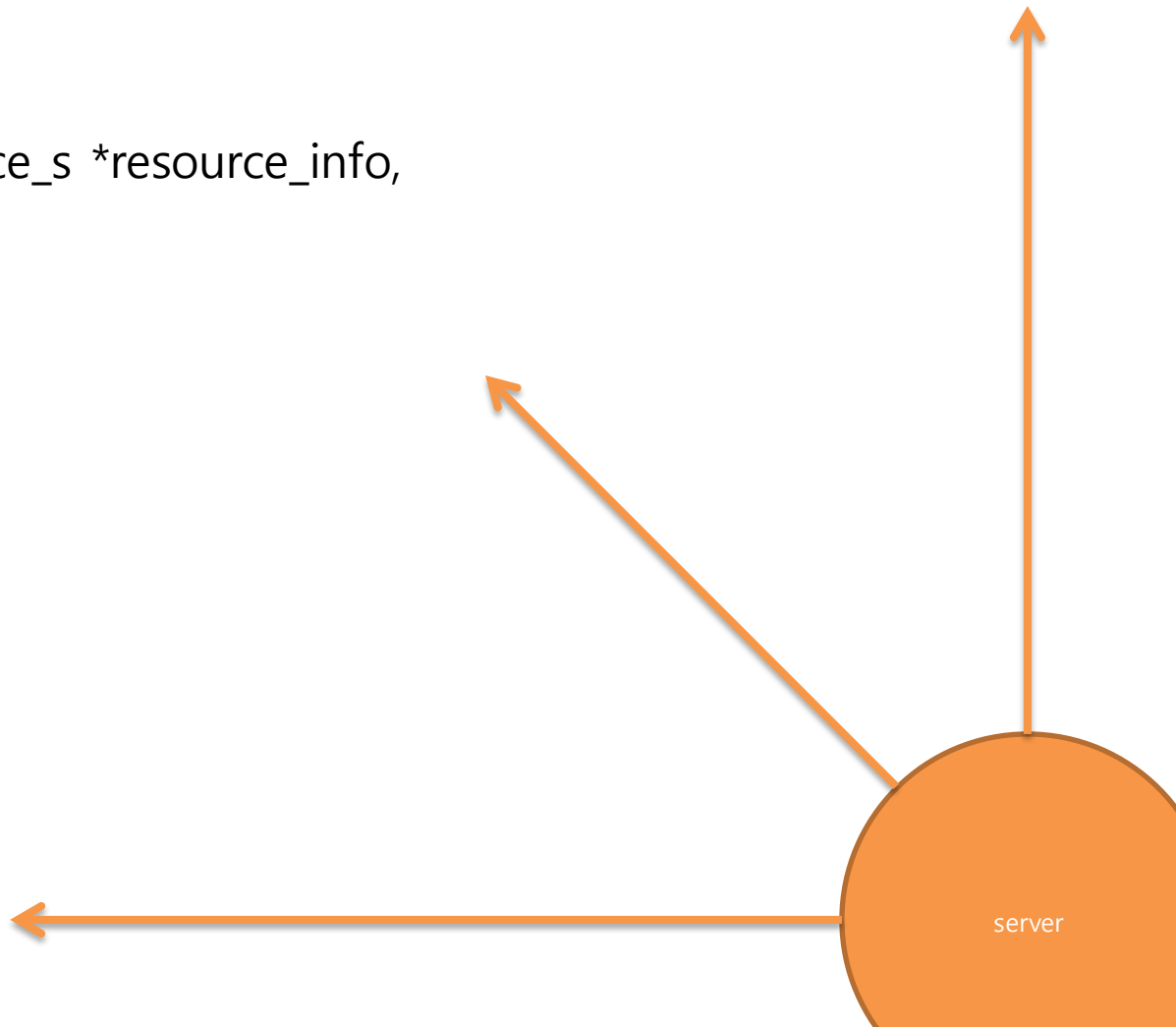
```
int      iotcon_resource_get_child_count (iotcon_resource_h resource, unsigned int *count)
int      iotcon_resource_get_nth_child (iotcon_resource_h parent, int index, iotcon_resource_h *child)
```

Server
notify

SOSCON



```
int connectivity_notify(  
    connectivity_resource_s *resource_info,  
    int value);
```



Representation

- a payload of a request or a response.
- uri_path, list of resource interfaces, list of resource types and its attributes.
- Attributes
 - to store and retrieve integer, boolean, double, string, byte string, list, null, resp_repr.
 - A list is a container that includes number of data of same type.



Representation



```
int  iotcon_representation_create (iotcon_representation_h *repr)
int  iotcon_representation_destroy (iotcon_representation_h repr)
int  iotcon_representation_clone (const iotcon_representation_h src, iotcon_representation_h *dest)
int  iotcon_representation_set_uri_path (iotcon_representation_h repr, const char *uri_path)
int  iotcon_representation_get_uri_path (iotcon_representation_h repr, char **uri_path)
int  iotcon_representation_set_resource_types (iotcon_representation_h repr, iotcon_resource_types_h types)
int  iotcon_representation_get_resource_types (iotcon_representation_h repr, iotcon_resource_types_h *types)
int  iotcon_representation_set_resource_interfaces (iotcon_representation_h repr, iotcon_resource_interfaces_h ifaces)
int  iotcon_representation_get_resource_interfaces (iotcon_representation_h repr, iotcon_resource_interfaces_h *ifaces)
int  iotcon_representation_set_attributes (iotcon_representation_h repr, iotcon_attributes_h attributes)
int  iotcon_representation_get_attributes (iotcon_representation_h repr, iotcon_attributes_h *attributes)
int  iotcon_representation_add_child (iotcon_representation_h parent, iotcon_representation_h child)
int  iotcon_representation_remove_child (iotcon_representation_h parent, iotcon_representation_h child)
int  iotcon_representation_foreach_children (iotcon_representation_h parent, iotcon_children_cb cb, void *user_data)
int  iotcon_representation_get_child_count (iotcon_representation_h parent, unsigned int *count)
int  iotcon_representation_get_nth_child (iotcon_representation_h parent, int pos, iotcon_representation_h *child)
```

Attributes

Server

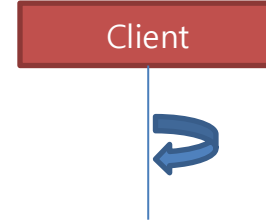
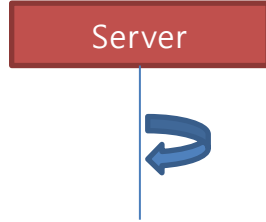


Client



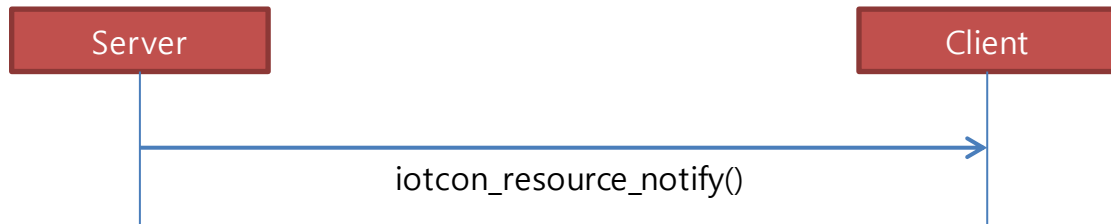
```
int      iotcon_attributes_create (iotcon_attributes_h *attributes)
int      iotcon_attributes_destroy (iotcon_attributes_h attributes)
int      iotcon_attributes_clone (iotcon_attributes_h attributes, iotcon_attributes_h *attributes_clone)
int      iotcon_attributes_add_int (iotcon_attributes_h attributes, const char *key, int val)
int      iotcon_attributes_add_bool (iotcon_attributes_h attributes, const char *key, bool val)
int      iotcon_attributes_add_double (iotcon_attributes_h attributes, const char *key, double val)
int      iotcon_attributes_add_str (iotcon_attributes_h attributes, const char *key, char *val)
int      iotcon_attributes_add_byte_str (iotcon_attributes_h attributes, const char *key, unsigned char *val, int len)
int      iotcon_attributes_add_list (iotcon_attributes_h attributes, const char *key, iotcon_list_h list)
int      iotcon_attributes_add_attributes (iotcon_attributes_h dest, const char *key, iotcon_attributes_h src)
int      iotcon_attributes_add_null (iotcon_attributes_h attributes, const char *key)
int      iotcon_attributes_get_int (iotcon_attributes_h attributes, const char *key, int *val)
/* bool, double, str, byte_str, list, attributes ... */
int      iotcon_attributes_is_null (iotcon_attributes_h attributes, const char *key, bool *is_null)
int      iotcon_attributes_remove (iotcon_attributes_h attributes, const char *key)
int      iotcon_attributes_get_type (iotcon_attributes_h attributes, const char *key, iotcon_type_e *type)
int      iotcon_attributes_foreach (iotcon_attributes_h attributes, iotcon_attributes_cb cb, void *user_data)
int      iotcon_attributes_get_keys_count (iotcon_attributes_h attributes, unsigned int *count) /* int, bool, double... */
```

List



```
int      iotcon_list_create (iotcon_type_e type, iotcon_list_h *list)
int      iotcon_list_destroy (iotcon_list_h list)
int      iotcon_list_add_int (iotcon_list_h list, int val, int pos)
int      iotcon_list_add_bool (iotcon_list_h list, bool val, int pos)
int      iotcon_list_add_double (iotcon_list_h list, double val, int pos)
int      iotcon_list_add_str (iotcon_list_h list, char *val, int pos)
int      iotcon_list_add_byte_str (iotcon_list_h list, unsigned char *val, int len, int pos)
int      iotcon_list_add_list (iotcon_list_h list, iotcon_list_h val, int pos)
int      iotcon_list_add_attributes (iotcon_list_h list, iotcon_attributes_h val, int pos)
int      iotcon_list_get_nth_int (iotcon_list_h list, int pos, int *val)
int      iotcon_list_foreach_int (iotcon_list_h list, iotcon_list_int_cb cb, void *user_data)
int      iotcon_list_remove_nth (iotcon_list_h list, int pos)
int      iotcon_list_get_type (iotcon_list_h list, iotcon_type_e *type)
int      iotcon_list_get_length (iotcon_list_h list, unsigned int *length)
```

Notify



```
/* If observers is NULL, the msg will notify to all observers. */  
int      iotcon_resource_notify (iotcon_resource_h resource, iotcon_representation_h repr, iotcon_observers_h observers, iotcon_qos_e qos)
```

Thank you

