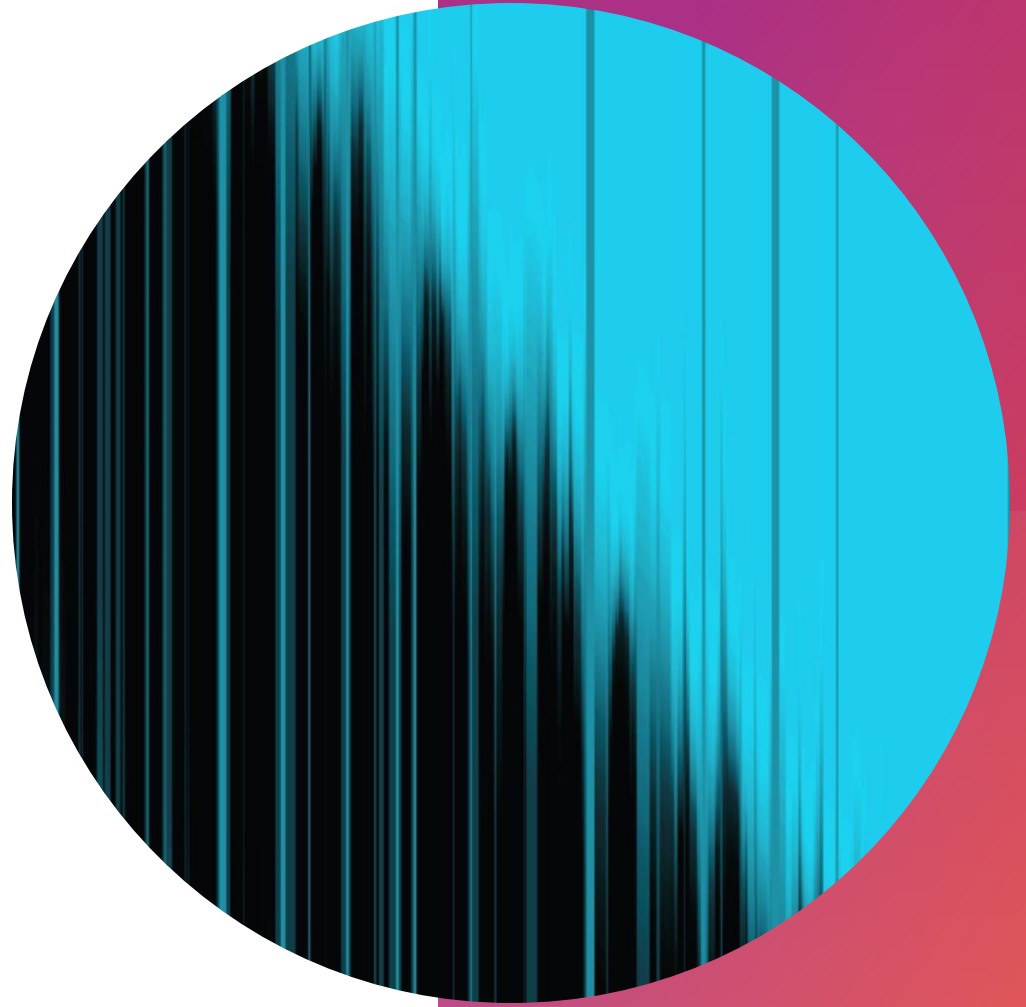


HTTP Daemon

ONEM2M TINY IOT PROJECT

엄경호



이번주 진행 상황

I. ACP 추가 구현

II. 리팩토링

ACP 추가 구현

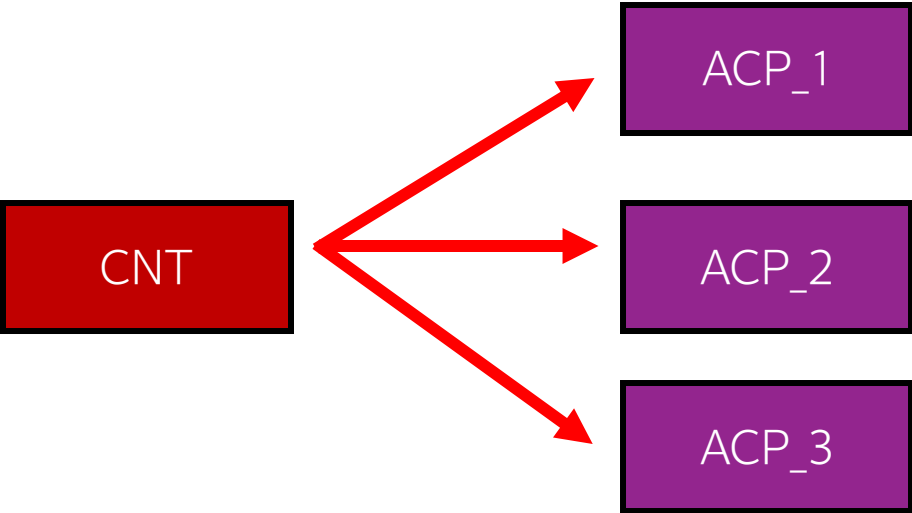
CNT acpi 속성 리스트화 → ["/TinyloT/acp_test1", "/TinyloT/acp_test2"] 등과 같이 설정 가능

추가 사항

pvs_acor, pvs_acop 로직 구현

CNT상에서 acpi 존재하지만 acp가 존재하지 않으면 forbidden

ACP 추가 구현



다중 ACP



항상 접근 제한



pvs_acor, pvs_acop 참고

리팩토링

```
typedef struct Node {
    struct Node *parent;
    struct Node *child;
    struct Node *siblingLeft;
    struct Node *siblingRight;

    char *rn;
    char *ri;
    char *pi;
    char *nu;
    char *sur;
    char *acpi;
    char *pv_acor;
    char *pv_acop;
    char *pvs_acor;
    char *pvs_acop;
    ObjectType ty;

    int cinSize;
    int net;
}Node;
```



```
Node* Create_Node(void *obj, ObjectType ty){
    Node* node = NULL;

    switch(ty) {
        case t_CSE: node = Create_CSE_Node((CSE*)obj); break;
        case t_AE: node = Create_AE_Node((AE*)obj); break;
        case t_CNT: node = Create_CNT_Node((CNT*)obj); break;
        case t_CIN: node = Create_CIN_Node((CIN*)obj); break;
        case t_Sub: node = Create_Sub_Node((Sub*)obj); break;
        case t_ACP: node = Create_ACP_Node((ACP*)obj); break;
    }
}
```

Create_Node 함수를 인터페이스로 별도의 함수들을 호출 가능하게

```
Node *Create_Node(char *rn, char *ri, char *pi, char *nu, char *sur, char *acpi, char *pv_acor, char *pv_acop, char *pvs_acor, char *pvs_acop)
```

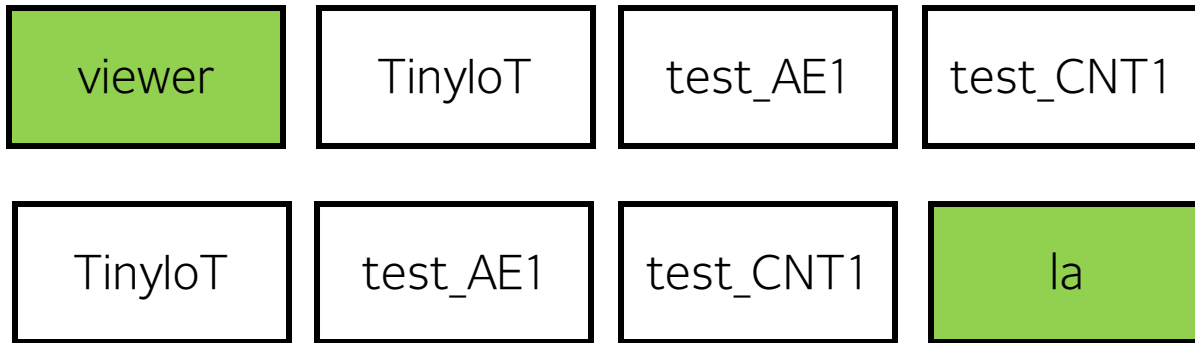
리팩토링

```
Node* Parse_URI(Node *cb, char *uri_array) {  
    fprintf(stderr, "Parse_URI \x1b[33m%s\x1b[0m...", uri_array);  
    //char uri_array[MAX_URI_SIZE];  
}
```

특정 URL에 대하여 어떤 오브젝트가 유효한지 또는 어떤 동작을 하는지 판단

/viewer/TinyloT/test_AE1/test_CNT1

/TinyloT/test_AE1/test_CNT1/la



```
if(!strcmp(uri_strtok[0], "viewer")) {  
    *op = o_VIEWER; index_s++;  
} else if(!strcmp(uri_strtok[index_end], "la") || !strcmp(uri_strtok[index_end], "latest")) {  
    *op = o_LA; index_end--;  
} else if(!strcmp(uri_strtok[index_end], "ol") || !strcmp(uri_strtok[index_end], "oldest")) {  
    *op = o_OL; index_end--;  
} else if(strstr(uri_strtok[index_end], "4-20")) {  
    *op = o_CIN_RI; Retrieve_CIN_Ri(uri_strtok[index_end]); return NULL;  
}
```

리팩토링

CIN 같은 경우는 m으로 요청 되지 않으므로 리소스 트리 상에 존재하는 이점이 별로 없음 -> 삭제

```
void Tree_data(Node *node, char **viewer_data, int cin_num)
```

```
int Add_child(Node *parent, Node *child);
```

```
void Retrieve_CIN(Node *pnode);
```

```
void Retrieve_CIN_Ri(char *ri);
```

ACP

AE

CNT

CIN

SUB