

나라마다 다른 주소 입력 양식 제공 & 주소 저장하기

ex) 한국 주소 : 도로명 주소 + 세부 주소, 미국 주소 : 주 + 도시 + 세부 주소

기존 한국에 맞춰서 개발된 주소 체계 변경 및 이에 따라 관리자 페이지 사용자 주소 표시 페이지 수정.

해결해야 할 문제

1. 나라마다 다른 주소 양식을 보여줘야 한다.
2. 나라마다 다른 주소 체계를 가지고 있다. 주문할 때 주소를 받아 db에 저장해야 한다.
3. 입력된 주소를 합쳐서 주소 포맷에 맞게 표시해야 한다.

해결

1. AngularJs templateUrl을 사용해 나라에 따라 다른 주소 입력 폼 html을 load.

```
templateUrl: ['$Nation', function ($Nation) {
    var nation = $Nation.Nation; // 국가(KR, US, JP ...)
    return "common address form url" + '/Address.tmpl.' + nation + + '.html';
}]
```

2. 주소를 json string 으로 변환 후 db에 저장해 어떤 나라의 주소라도 하나의 컬럼에 저장할 수 있다.

→ 예를 들면 미국 주소는 { "주이름" : "CA", "도시명" : "LA", "세부주소" : "new york apartment" } 처럼 들어간다. 키 값들은 db에서 설정 가능하다.

```
// attribute 생성
[AttributeUsage(AttributeTargets.Property)]
public class JsonParsingAttribute : Attribute { }

// model property에 attribute 선언
[JsonParsing]
public JObject Address { get; set; } = new JObject();

// form에서 json string으로 넘어온 주소 값 parsing 해서 set
model.Address = JObject.Parse(FormData.GetStrParam("txtAddr", "{}"));

// db에 insert 할 때 json string으로 변환
if (Attribute.IsDefined(prop, typeof(JsonParsingAttribute)))
{
    return GetJsonValue(prop, value);
}

private static object GetJsonValue(PropertyInfo prop, object value)
{
    if (string.IsNullOrEmpty(value as string))
    {
        return Activator.CreateInstance(prop.PropertyType);
    }
    else
    {
        if (!value.GetType().Equals(typeof(string))) throw new NotSupportedException($"value type must be string : {value.GetType()}");
        return JsonConvert.DeserializeObject(value as string, prop.PropertyType);
    }
}
```

3. 주소 포맷과 json 주소 객체 key 값들을 db에 저장. 화면에 주소를 표시할 때 db에 저장한 주소 데이터(json address)를 주소 포맷에 맞춰 변환한다.

→ json string 주소를 실제 주소로 변환해주는 서비스 개발

KEY_CHAR : json으로 표현한 주소 객체의 키 이름임을 알려주는 문자(키 구분 문자). 후에 replace를 위해 사용. 편의상 여기서는 '#'이라고 해보자.

ADDRESS_FORMAT : json 주소를 어떻게 표시할지 알려주는 포맷 ("#세부주소2 #세부주소1 #도시명, #주이름")

ADDRESS_KEY_ARRAY : json 주소 객체에서 사용하는 key array (['세부주소1', '세부주소2', '도시명', '주이름'])

1. ADDRESS_KEY_ARRAY를 이용해서 객체 초기화
2. ADDRESS_FORMAT에서 'KEY_CHAR + KEY'의 조합으로 KEY 부분을 찾을
3. KEY_CHAR + KEY 부분을 매칭되는 값으로 replace

JavaScript & AngularJs

```
// 변환 시작
// jsonAddressObj(json 주소 객체) : { "주이름" : "CA", "도시명" : "LA", "세부주소1" : "new york apartment", 세부주소2 : "" }
function jsonAddressToStrAddress(jsonAddressObj) {
    var strAddressFormat = ADDRESS_FORMAT;
    var initiatedJsonAddressObj = getInitiatedJsonAddressObj(jsonAddressObj)
    for (var key in initiatedJsonAddressObj) {
        strAddressFormat = strAddressFormat.replaceAll(KEY_CHAR + key, initiatedJsonAddressObj[key]);
    }
    return strAddressFormat;
}

function getInitiatedJsonAddressObj(jsonAddressObj) {
    var initiatedJsonAddressObj = init();
    for (var key in initiatedJsonAddressObj) {
        if(jsonAddressObj.hasOwnProperty(key)){
            initiatedJsonAddressObj[key] = jsonAddressObj[key];
        }
    }
    return initiatedJsonAddressObj;
}

function init(){
    var jsonAddressObj = {};
    var keyArray = ADDRESS_KEY_ARRAY;
    for(var i = 0 ; i < keyArray.length ; i++){
        jsonAddressObj[keyArray[i]] = "";
    }
    return jsonAddressObj;
}

return {
    toStrAddress: function (jsonAddress) {
        return jsonAddressToStrAddress(jsonAddress);
    }
};
```

C#

```
// 시작
public AddressItem Do(JObject jsonAddress)
{
    var strAddressFormat = ADDRESS_FORMAT;
    var strAddress = ToStrAddress(jsonAddress, strAddressFormat);
    return new AddressItem { Address = strAddress };
}

private string ToStrAddress(JObject jsonAddress, string format)
{
    var jsonAddressDic = GetJsonAddressDic(jsonAddress);
    var strAddress = format;
```

```

        foreach (var data in jsonAddressDic)
        {
            strAddress = strAddress.Replace($"{{KEY_CHAR + data.Key}}", data.Value);
        }
        return strAddress;
    }

    public Dictionary<string, string> GetJsonAddressDic(JObject jsonAddress)
    {
        var jsonAddressDic = Init();
        foreach (var data in jsonAddress)
        {
            if (jsonAddressDic.ContainsKey(data.Key))
            {
                jsonAddressDic[data.Key] = data.Value.ToString();
            }
        }
        return jsonAddressDic;
    }

    public Dictionary<string, string> Init()
    {
        var jsonAddressKey = ADDRESS_KEY_ARRAY;
        var jsonAddressDic = new Dictionary<string, string>();
        foreach (var key in jsonAddressKey)
        {
            jsonAddressDic.Add(key, "");
        }
        return jsonAddressDic;
    }
}

```

📌 결과

일본

미국

Shipping Address	Edit
test	
DETAIL ADDRESS1 DETAIL ADDRESS2	
CITY NY 23442	
234-234-24	