美团Touch周边游景点详情页接口破解

一、业务背景

美团Touch周边游业务中，景点详情页页面和查看 简介页面中两个接口请求。

页面：https://i.meituan.com/awp/h5/lvyou/poi/detail/index.html?ct_poi=14

5246136739881968654990656856174445865_e3194081313476163844_c0_dtrippoipolyb&poiId=271358

二、目标接口

1. https://itrip.meituan.com/volga/api/v3/trip/poi/business/info/271358?ct_poi=14524613673988196865499065685617444586 5_e3194081313476163844_c0_dtrippoipolyb&poiId=271358&source=mt&client=wap&uuid=8BAB967D22ADF97D8AA3CC0EECE1CC7D80859CE633255C7380CC69CB05513281&cityId=1&feclient=lvyou_wap&platform=1&partner=11&originUrl=https%3A%2F%2Fi.meituan.com%2Fawp%2Fh5%2Flvyou%2Fpoi%2Fdetail%2Findex.html%3Fct_poi%3D14524613673988196865499065685617444586 5_e3194081313476163844_c0_dtrippoipolyb%26poiId%3D271358&_token=eJxtkV1LwzAUhv9LQK9Kk9N8NCmUMRClgheO6c0YpWtDG%252BwXbTY3xP%252FuqUxQEALvc95zEl5OPsiUVSQBxhhEATnZiSQEQhYqEhA%252FY0caJmIlmNFcB6T86xmB3mF6vSPJTilbUPvF2GC9A6VZAEyyffCbI4FnmcpwiDTej3NCqQs76%252Fyx6MNy6GjxPtJG0vZ0GY50HBytrC9cS11f2XPY%252BK5dlT7HRgpCRkIBVzE3WoNRWklhDFNSaakgFkJItHLLwQimgQPH6KC4FiIvWV75yY340Di0l8MtQlalUQxcalzA%252F%252BGWG9doC7Zu9t8ZV7N3VX5I4ba0J9v71HX%252BJrpvhs6ORW0Ry8LbepgugAxGEtxCt8UtoL5dtbiq%252F6mf8EMwyuzqHsk%252BnrcvdbZeP9TrzXOaks8vt%252F%252BCUg%253D%253D
2. https://i.meituan.com/lvyou/volga/api/v3/trip/poi/info/desc/?poiId=271358&source=mt&client=wap&uuid=8BAB967D22ADF97D8AA3CC0EECE1CC7D80859CE633255C7380CC69CB05513281&cityId=1&feclient=lvyou_wap&poiId=271358

三、接口分析

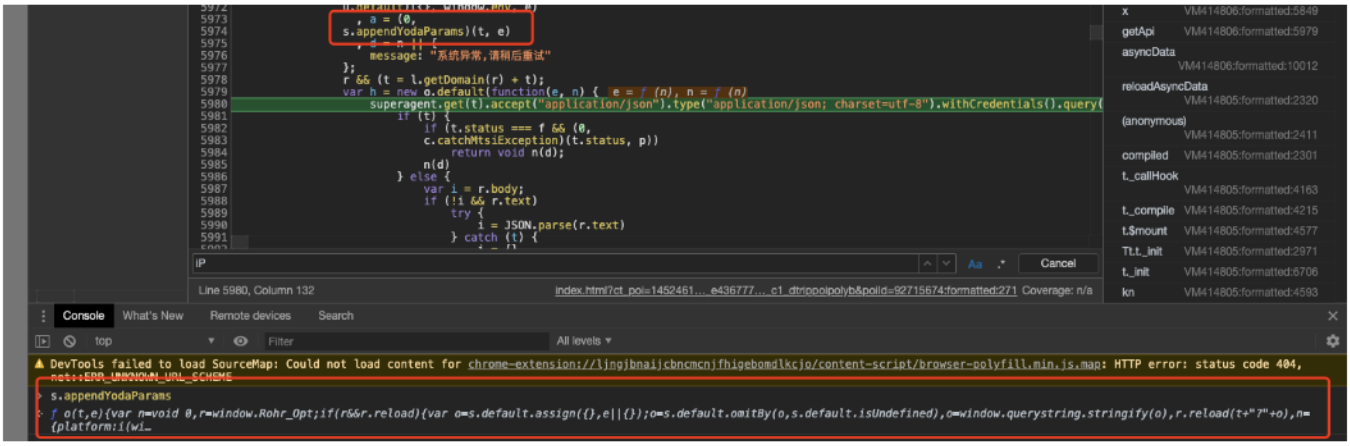
1.景点详情接口

1）_token参数

分析请求的header、cookie和url参数部分，可以发现除了基本的业务参数外，主要有_token参数和uuid参数不知含义。

从请求发起开始，通过Chrome断点分析，可以追溯到token参数如何被赋值进请求url中

继续看a参数如何生成，



先不看传入的参数，分析appendYodaParams函数，

```
function o(t, e) {
    var n = void 0
      , r = window.Rohr_Opt;
    if (r && r.reload) {
        var o = s.default.assign({}, e || {});
        o = s.default.omitBy(o, s.default.isUndefined),
        o = window.querystring.stringify(o),
        r.reload(t + "?" + o),
        n = {
            platform: i(window.env),
            partner: 11,
            originUrl: location.href,
            _token: window.rohrdata
        },
        f[l.default.appName] && (n.app = f[l.default.appName])
    }
    return n
}
```

可以发现_token参数来源于window.rohrdata，与其他参数 无关，全局搜索rohrdata发现，并无法搜索到相关赋值地方，猜测在赋值时可能对rohrdata使用了其他变量进行替换或混淆（后面倒推时发现确实使用了其他变量替换）。

接着看上面一行有个可以的方法调用，r.reload(t+"?"+o),进入这个方法（rohr.min.js 美团前端通用token生成算法js，外卖、酒店、门票都有用到），

```
iP.reload = function(jv) {
    var jw;
    var jx = {};
    if (typeof jv === _$_543c[91]) {
        jx = iO.parse(jv.split(_$_543c[146])[1])
    } else {
        if (typeof jv === _$_543c[2]) {
            jx = jv
        }
    }
    ;iP.sign = iJ(jx);
    iP.cts = new Date().getTime();
    jw = iI(iP);
    if (Rohr_Opt.LogVal && typeof (window) !== _$_543c[0]) {
        window[Rohr_Opt.LogVal] = encodeURIComponent(jw)
    }
    ;return jw
}
```

可以发现整个token的算法都在这个js文件中 ，通过debug可以逐步抽丝剥茧还原出整个算法，算法很简单，只利用到了zlib压缩和base64编码，具体不再分析

将token通过base64解码会得到一个乱码字符串，实际上是一个十六进制的字节数组，通过zlib算法 解压，可以得到一个json字符串，

解压示例


```
{
    "rId":100900,
    "ver":"1.0.6",
    "ts":1555230580338,
    "cts":1555230580399,
    "brVD":[
        1010,
        750
    ],
    "brR":[
        [
            1920,
            1080
        ],
        [
            1920,
            1040
        ],
        24,
        24
    ],
    "bI":[
        "https://gz.meituan.com/meishi/c11/",
        ""
    ],
    "mT":[

    ],
    "kT":[

    ],
    "aT":[

    ],
    "tT":[

    ],
    "aM":"",
    "sign":"eJwdjktOwzAQhu/ShXeJ4zYNKpIXqKtKFTsOMLUn6Yj4ofG4UjkM10CsOE3vgWH36df/2gAjnLwdlAPBBsYoR3J
/hYD28f3z+PpUnmJEPqYa5UWEm0mlLBRqOSaP1qjEtFB849VeRXJ51nr56AOSVIi9S0E3LlfSzhitMix
/mQwsrdWa7aTyCjInDk1mKu9nvOHauCQWq2rB/8laqd3cX+adv0zdzm3nbjTOdzCi69A/HQAHOOyHafMLmEtKXg=="
```


其中比较重要 的参数有bl和sign参数，bl参数为当前页面URL以及refer参数，sign参数则是将请求URL的path部分先压缩后base64编码得到的字符串。至此整个token参数都可以还原成java语言。

2）uuid参数

按照上面的思路继续分析uuid参数，可以发现uuid来源于 cookie中的dpid或iuuid或uuid。

而cookie的来源则是在请求服务端接口时响应里返回Set-Cookie时种上。

cookie iuuid来源：

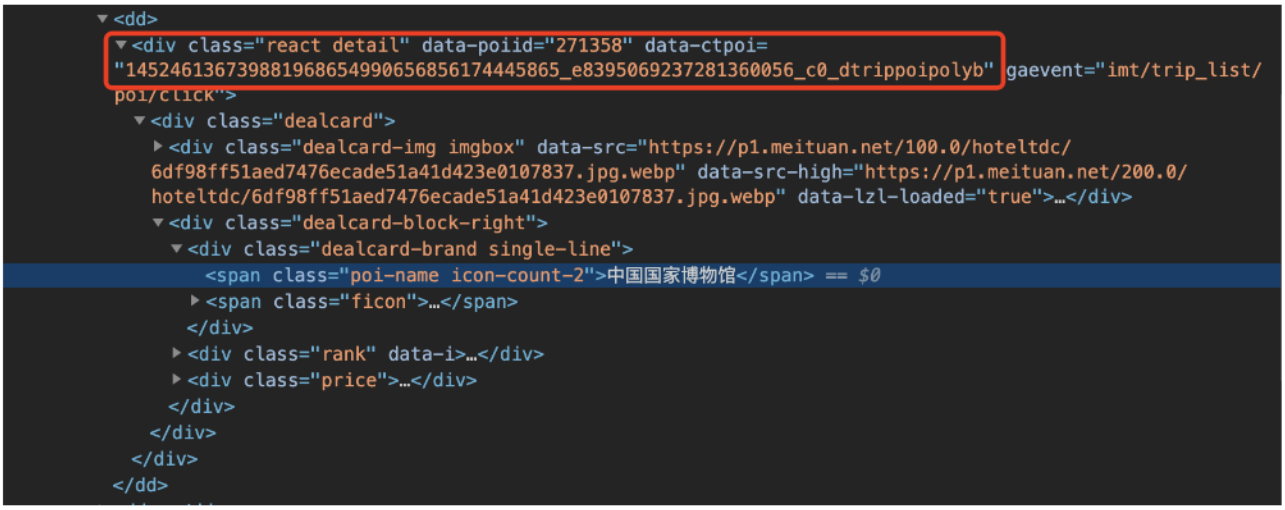
https://i.meituan.com/ok/204?ndr

或

https://i.meituan.com

3)ct_poi参数

由列表页中跳转时所携带



2.查看简介接口

四、参数还原

1._token参数

工程：http://gitlab.corp.qunar.com/kykm/protocol-crack

API：com.protocols.dptouchcrack.utils.ParamUtils#generateLvyouDetailToken