

# ODATA DOCUMENT

## B1: Lấy file xml metadata

Download file metadata.xml

Run Terminal

cd to path chứa file framework trước khi gõ lệnh ->

cd {path chứa file odatagen}

cd /PineEntertainment/Projects/DONT/src/game/mac\_core/ODataSDK/Framework/bin/OdatagenBinary

grant quyền cho file excute (chỉ cần làm 1 lần nếu file bị thay odatagen bị thay đổi)

chmod +x {path}

chmod +x  
/PineEntertainment/Projects/DONT/src/game/mac\_core/ODataSDK/Framework/bin/OdatagenBinary/odatagen

Lệnh ->

sudo ./odatagen /metadata={path chứa metadata} /out= {path chứa output}

VD:

sudo ./odatagen /metadata=/PineEntertainment/Projects/DONT/doc/metadata.xml  
/out=/PineEntertainment/Projects/DONT/src/\_mac/src/PineGame/Classes

Odatagen sẽ gen ra 2 file context để include vào source từ file metadata. Ở đây từ metadata trên, ta gen ra 2 file :  
"DontContext.h" và "DontContext.mm"

## Cách thức INSERT, UPDATE, CREATE RECORD bằng ngôn ngữ Object-C

### LƯU Ý QUAN TRỌNG

Odata update sử dụng json V3 thay cho XML V2 format

Vì vậy khi sử dụng project sau này nên sử dụng câu lệnh này sau khi init proxy để sử dụng JSON V3 (giảm 4 lần dung lượng)

VD:

```
YouArmyContext *proxy = [[YouArmyContext alloc] initWithUri:[NSString alloc]  
initWithUTF8String:URL_SERVER] credential:nil];  
  
[proxy setParseData:ParseDataType::JSON_Parser_Type];
```

### B2 : Kết nối Database

Khi kết nối cần sử dụng header để chứng thực (Authoriacation)

```
[[PineGameDelegate sharedInstance] getHeaderForProxy:UserID onjectConext:proxy];
```

YouArmyContext là đối tượng trong được gen ra từ file metadata

Chuỗi kết nối database của ta là URL\_SERVER:

```
#define URL_SERVER          "http://youarmy.cloudapp.net/api/youarmy/"  
@try  
{  
    YouArmyContext *proxy = [[YouArmyContext alloc] initWithUri:[NSString alloc] initWithUTF8String:  
URL_SERVER] credential:nil];  
  
    [[PineGameDelegate sharedInstance] getHeaderForProxy:UserID onjectConext:proxy];  
  
}  
@catch(ODataServiceException *e)  
{  
    NSLog(@"1 exception:: %d ", [e getStatusCode]);  
}  
@catch (NSException * e)  
{  
    NSLog(@"all exception:: %@: %@", [e name], [e reason]);  
}
```

### INSERT RECORD

YouArmy\_DataTransferObjects\_SaveGameDto là đối tượng trong được gen ra từ file metadata

```
extern "C" int CreateNewData(const char* GC_UserID, const char* FB_UserID, const char* data, int data_length)
{
    @try
    {
        YouArmy_DataTransferObjects_SaveGameDto *save =
        [[YouArmy_DataTransferObjects_SaveGameDto alloc] init];

        [proxy setParseData:ParseDataType::JSON_Parser_Type];

        [[PineGameDelegate sharedInstance] getHeaderForProxy:UserID onjectConext:proxy];
        [save setUsername:@"User Name"];
        [save setGameCenterId:[NSString alloc] initWithUTF8String:GC_UserID]];
        [save setFacebookId:[NSString alloc] initWithUTF8String:FB_UserID]];
        [save setSecretKey:@"!@#$%^&*"];
        [save setData:[NSData alloc] initWithBytes:data length:data_length]];
        [save setCurrentXp:[NSNumber alloc] initWithInt:0]];
        [proxy addToSaveGames:save];
        [proxy saveChanges];
        [save release];
    }
    @catch (DataServiceRequestException * e)
    {
        NSLog(@"0 exception:: %@: %@",[e name], [e getResponse]);
        NSLog(@"http code:: %@", [[e getResponse] statusCode]);
    }
    @catch(ODatServiceException *e)
    {
        NSLog(@"1 exception:: %d ", [e.getStatusCode]);
    }
    @catch (NSException * e)
    {
        NSLog(@"all exception:: %@: %@", [e name], [e reason]);
    }
}
}
```

Những command INSERT, UPDATE, SELECT hầu hết đều có những dòng lệnh giống nhau. **INSERT** chú ý vào phần được in đậm.

## SELECT RECORD

Khi select ta sử dụng đối tượng DataServiceQuery.

```
DataServiceQuery *query = [proxy savegames];
```

Đồng thời **SET** điều kiện thông qua filter.

```
NSString *filter = [NSString stringWithFormat:@"GameCenterId eq '%s'", GC_UserID];
[query filter:filter];
```

➔ Odata không sử dụng điều kiện bằng các "<", ">", "=".... Chúng được thay thế bằng "lt", "gt", "eq"....

Có thể tham khảo thêm trong file Doc trong Framework

Nhằm tăng tốc độ GetData ta nên sử dụng function "select" của DataServiceQuery để lấy field cần sử dụng của record.

```
@try
{
    YouArmyContext *proxy = [[YouArmyContext alloc] initWithUri:[NSString alloc] initWithUTF8String:
    URL_SERVER] credential:nil];

    [proxy setParseData:ParseDataType::JSON_Parser_Type];

    [[PineGameDelegate sharedInstance] getHeaderForProxy:UserID onjectConext:proxy];
    DataServiceQuery *query = [proxy savegames];
    NSString *filter = [NSString stringWithFormat:@"GameCenterId eq '%s'", GC_UserID];
    [query filter:filter];
    [query select:@" CurrentXp"];
    QueryOperationResponse *response = [query execute];
    YouArmy_DataTransferObjects_SaveGameDto *save = [[response getResult] objectAtIndex:0];
    int currentXP = [[save getCurrentXp] intValue];
    [save release];
    [proxy release];
}
@catch (DataServiceRequestException * e)
{
    NSLog(@"0 exception:: %@: %@",[e name], [e getResponse]);
    NSLog(@"http code:: %@", [[e getResponse] statusCode]);
}
@catch(ODataServiceException *e)
{
    NSLog(@"1 exception:: %d ", [e.getStatusCode]);
}
@catch (NSException * e)
{
    NSLog(@"all exception:: %@: %@", [e name], [e reason]);
}
```

## UPDATE RECORD

```
@try
{
    YouArmyContext *proxy = [[YouArmyContext alloc] initWithUri:[NSString alloc]
    initWithUTF8String:URL_SERVER] credential:nil];

    [proxy setParseData:ParseDataType::JSON_Parser_Type];

    [[PineGameDelegate sharedInstance] getHeaderForProxy:UserID onjectConext:proxy];
    DataServiceQuery *query = [proxy savegames];
    NSString *filter = [NSString stringWithFormat:@"GameCenterId eq '%s'", GC_UserID];
    [query filter:filter];
```

```

QueryOperationResponse *response = [query execute];
if (response)
{
    YouArmy_DataTransferObjects_SaveGameDto *save = [[response getResult] objectAtIndex:0];
    [save setFacebookId:[NSString alloc] initWithUTF8String:@"facebook ID"];
    [save setCurrentXp:[NSNumber alloc] initWithInt:-3];
    [proxy updateObject:save];
    [proxy saveChanges];
    [save release];
}
[proxy release];
}
@catch (DataServiceRequestException * e)
{
    NSLog(@"0 exception:: %@: %@",[e name], [e getResponse]);
    NSLog(@"http code:: %@", [[e getResponse] getStatusCode]);
}
@catch(ODataServiceException *e)
{
    NSLog(@"1 exception:: %d ", [e getStatusCode]);
}
@catch (NSException * e)
{
    NSLog(@"all exception:: %@: %@", [e name], [e reason]);
}

@catch (DataServiceRequestException * e)
{
    NSLog(@"0 exception:: %@: %@",[e name], [e getResponse]);
    NSLog(@"http code:: %@", [[e getResponse] getStatusCode]);
}
@catch(ODataServiceException *e)
{
    NSLog(@"1 exception:: %d ", [e getStatusCode]);
}
@catch (NSException * e)
{
    NSLog(@"all exception:: %@: %@", [e name], [e reason]);
}

```

## MỘT SỐ HTTP ERROR CODE

Success : 201

Error : 400