

```

//
//  PlayAreaCellTypeRESTWebAPIModelAccessStrategy.swift
//  f30
//
//  Created by David on 05/02/2018.
//  Copyright © 2018 com.smartfoundation. All rights reserved.
//

import SFCore
import SFModel
import SFSocial
import SFSerialization
import SFNet
import f30Model

/// A strategy for accessing the PlayAreaCellType model data using a REST Web API
public class PlayAreaCellTypeRESTWebAPIModelAccessStrategy:
    RESTWebAPIModelAccessStrategyBase {

    // MARK: - Initializers

    private override init() {
        super.init()
    }

    public override init(connectionString: String,
                        storageDateFormatter: DateFormatter) {
        super.init(connectionString: connectionString,
                    storageDateFormatter: storageDateFormatter,
                    tableName: "PlayAreaCellTypes")
    }

    // MARK: - Private Methods

    fileprivate func runQuery(byID ID: String, loadRelationalTablesYN: Bool, into
        collection: ProtocolModelItemCollection, oncomplete
        completionHandler:@escaping ([String:Any]?, Error?) -> Void) {

        #if DEBUG

            if (ApplicationFlags.flag(key: "LoadPlayAreaCellTypesDummyDataYN")) {

                self.selectDummy(byID: ID, loadRelationalTablesYN:
                    loadRelationalTablesYN, into: collection, oncomplete:
                    completionHandler)

                return
            }

        #endif

        // Create the dataWrapper
        let dataWrapper: DataJSONWrapper = DataJSONWrapper()

```

```

dataWrapper.setParameterValue(key: "\
(PlayAreaCellTypeDataParameterKeys.LoadRelationalTablesYN)", value: "\
(loadRelationalTablesYN)")

// Create processResponse completion handler
let processResponseCompletionHandler: ([[String:Any]?, URLResponse?,
Error?) -> Void) =
{
    (data, response, error) -> Void in // [weak self]

    // Call the completion handler
    completionHandler(data, error)
}

// Create processResponse
let processResponse: ((NSData?, URLResponse?, Error?) ->
Void) = self.getProcessResponse(oncomplete:
processResponseCompletionHandler)

// Create restApiHelper
let restApiHelper: RESTApiHelper = RESTApiHelper(processResponse:
processResponse, mode: RESTApiHelperMode.CompletionHandler)

// Get the Url
var urlString: String =
    NSLocalizedString("PlayAreaCellTypesSelectByIdUrl", tableName:
    "RESTWebAPIConfig", comment: "")
urlString = String(format: urlString, ID)

// Call the REST Api
restApiHelper.call(urlString: urlString, httpMethod: .POST, data:
dataWrapper)
}

fileprivate func runQuery(byRelativeMember relativeMemberID: String,
playGameID: String, loadRelationalTablesYN: Bool, collection:
ProtocolModelItemCollection, oncomplete completionHandler: @escaping ([String
: Any]?, Error?) -> Void) {

    #if DEBUG

        if (ApplicationFlags.flag(key: "LoadPlayAreaCellTypesDummyDataYN")) {

            self.selectDummy(byRelativeMember: relativeMemberID, playGameID:
            playGameID, loadRelationalTablesYN: loadRelationalTablesYN,
            collection: collection, oncomplete: completionHandler)

            return

        }

    #endif

    // Create the dataWrapper
    let dataWrapper: DataJSONWrapper = DataJSONWrapper()

```

```

dataWrapper.setParameterValue(key: "\
(PlayAreaCellTypeDataParameterKeys.LoadRelationalTablesYN)", value: "\
(loadRelationalTablesYN)")

// Create processResponse completion handler
let processResponseCompletionHandler: ([[String:Any]?, URLResponse?,
Error?) -> Void) =
{
    (data, response, error) -> Void in // [weak self]

    // Call the completion handler
    completionHandler(data, error)
}

// Create processResponse
let processResponse: ((NSData?, URLResponse?, Error?) ->
Void) = self.getResponse(oncomplete:
processResponseCompletionHandler)

// Create restApiHelper
let restApiHelper: RESTApiHelper = RESTApiHelper(processResponse:
processResponse, mode: RESTApiHelperMode.CompletionHandler)

// Get the Url
var urlString: String =
    NSLocalizedString("PlayAreaCellTypesSelectByRelativeMemberPlayGameID",
    tableName: "RESTWebAPIConfig", comment: "")
urlString = String(format: urlString,
                    relativeMemberID,
                    playGameID)

// Call the REST Api
restApiHelper.call(urlString: urlString, httpMethod: .POST, data:
dataWrapper)
}

// MARK: - Override Methods

// MARK: - Dummy Data Methods

fileprivate func selectDummy(byID ID: String, loadRelationalTablesYN: Bool,
into collection: ProtocolModelItemCollection, oncomplete
completionHandler:@escaping ([[String:Any]?, Error?) -> Void) {

    let responseString = NSLocalizedString("byID", tableName:
    "PlayAreaCellTypesDummyRESTWebAPIResponse", comment: "")

    // Convert the response to JSON dictionary
    let data: [String:Any]? = JSONHelper.stringToJSON(jsonString:
responseString) as? [String:Any]

    // Process the data
    let returnData: [String:Any]? =
    self.processRESTWebAPIResponse(responseData: data!)

```

```

        // Call the completion handler
        completionHandler(returnData, nil)
    }

fileprivate func selectDummy(byRelativeMember relativeMemberID: String,
    playGameID: String, loadRelationalTablesYN: Bool, collection:
    ProtocolModelItemCollection, oncomplete completionHandler: @escaping ([String
    : Any]?, Error?) -> Void) {

    let responseString = NSLocalizedString("byRelativeMember_playGameID",
        tableName: "PlayAreaCellTypesDummyRESTWebAPIResponse", comment: "")

    // Convert the response to JSON dictionary
    let data: [String:Any]? = JSONHelper.stringToJSON(jsonString:
        responseString) as? [String:Any]

    // Process the data
    let returnData: [String:Any]? =
        self.processRESTWebAPIResponse(responseData: data!)

    // Call the completion handler
    completionHandler(returnData, nil)
}

}

// MARK: - Extension ProtocolPlayAreaCellTypeModelAccessStrategy

extension PlayAreaCellTypeRESTWebAPIModelAccessStrategy:
    ProtocolPlayAreaCellTypeModelAccessStrategy {

    // MARK: - Public Methods

    public func select(byID ID: String, loadRelationalTablesYN: Bool, collection:
        ProtocolModelItemCollection, oncomplete completionHandler:@escaping
        ([String:Any]?, ProtocolModelItemCollection?, Error?) -> Void) {

        // Create completion handler
        let runQueryCompletionHandler: ([[String:Any]?, Error?) -> Void) =
            self.getRunQueryCompletionHandler(collection: collection, oncomplete:
                completionHandler)

        // Run the query
        self.runQuery(byID: ID, loadRelationalTablesYN: loadRelationalTablesYN,
            into: collection, oncomplete: runQueryCompletionHandler)
    }

    public func select(byRelativeMember relativeMemberID: String, playGameID:
        String, loadRelationalTablesYN: Bool, collection:
        ProtocolModelItemCollection, oncomplete completionHandler: @escaping ([String
        : Any]?, ProtocolModelItemCollection?, Error?) -> Void) {

        // Create completion handler

```

```
let runQueryCompletionHandler: ([String:Any]?, Error?) -> Void) =  
    self.getRunQueryCompletionHandler(collection: collection, oncomplete:  
        completionHandler)
```

```
// Run the query
```

```
self.runQuery(byRelativeMember: relativeMemberID, playGameID: playGameID,  
    loadRelationalTablesYN: loadRelationalTablesYN, collection: collection,  
    oncomplete: runQueryCompletionHandler)
```

```
}
```

```
}
```