

```

//
// PlayAreaCellRESTWebAPIModelAccessStrategy.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 PlayAreaCell model data using a REST Web API
public class PlayAreaCellRESTWebAPIModelAccessStrategy:
    RESTWebAPIModelAccessStrategyBase {

    // MARK: - Initializers

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

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

    // 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: "LoadPlayAreaCellsDummyDataYN")) {

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

                return
            }

        #endif

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

```

```

dataWrapper.setParameterValue(key: "\
(PlayAreaCellDataParameterKeys.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("PlayAreaCellsSelectByIDUrl", tableName:
    "RESTWebAPIConfig", comment: "")
urlString = String(format: urlString, ID)

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

fileprivate func runQuery(byCellCoordRange relativeMemberID: String,
playGameID: String, playAreaID: String, fromColumn: Int, fromRow: Int,
toColumn: Int, toRow: Int, loadRelationalTablesYN: Bool, collection:
ProtocolModelItemCollection, oncomplete completionHandler: @escaping ([String
: Any]?, Error?) -> Void) {

    #if DEBUG

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

            self.selectDummy(byCellCoordRange: relativeMemberID, playGameID:
            playGameID, playAreaID: playAreaID, fromColumn: fromColumn,
            fromRow: fromRow, toColumn: toColumn, toRow: toRow,
            loadRelationalTablesYN: loadRelationalTablesYN, collection:
            collection, oncomplete: completionHandler)

            return
        }

    #endif
}

```

```

// Create the dataWrapper
let dataWrapper: DataJSONWrapper = DataJSONWrapper()
dataWrapper.setParameterValue(key: "\
(PlayAreaCellDataParameterKeys.FromColumn)", value: "\{(fromColumn)")
dataWrapper.setParameterValue(key: "\
(PlayAreaCellDataParameterKeys.FromRow)", value: "\{(fromRow)")
dataWrapper.setParameterValue(key: "\
(PlayAreaCellDataParameterKeys.ToColumn)", value: "\{(toColumn)")
dataWrapper.setParameterValue(key: "\
(PlayAreaCellDataParameterKeys.ToRow)", value: "\{(toRow)")
dataWrapper.setParameterValue(key: "\
(PlayAreaCellDataParameterKeys.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: ((NSMutableData?, 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("PlayAreaCellsSelectByCellCoordRangePlayGameID",
    tableName: "RESTWebAPIConfig", comment: "")
urlString = String(format: urlString,
                    relativeMemberID,
                    playGameID)

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

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

    #if DEBUG

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

```

```

        self.selectDummy(byIsSpecialYN: isSpecialYN, relativeMemberID:
            relativeMemberID, playGameID: playGameID, playAreaID: playAreaID,
            loadRelationalTablesYN: loadRelationalTablesYN, collection:
            collection, oncomplete: completionHandler)

        return

    }

#endif

// Create the dataWrapper
let dataWrapper: DataJSONWrapper = DataJSONWrapper()
dataWrapper.setParameterValue(key: "\
    (PlayAreaCellDataParameterKeys.IsSpecialYN)", value: "\
    (BoolHelper.toInt(value: isSpecialYN))")
dataWrapper.setParameterValue(key: "\
    (PlayAreaCellDataParameterKeys.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: ((NSMutableData?, 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("PlayAreaCellsSelectbyIsSpecialYNPlayGameID",
        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:
        "PlayAreaCellsDummyRESTWebAPIResponse", 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(byCellCoordRange relativeMemberID: String,
    playGameID: String, playAreaID: String, fromColumn: Int, fromRow: Int,
    toColumn: Int, toRow: Int, loadRelationalTablesYN: Bool, collection:
    ProtocolModelItemCollection, oncomplete completionHandler: @escaping ([String
    : Any]?, Error?) -> Void) {

    let responseString = NSLocalizedString("byCellCoordRange_playGameID",
        tableName: "PlayAreaCellsDummyRESTWebAPIResponse", 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(byIsSpecialYN isSpecialYN: Bool,
    relativeMemberID: String, playGameID: String, playAreaID: String,
    loadRelationalTablesYN: Bool, collection: ProtocolModelItemCollection,
    oncomplete completionHandler: @escaping ([String : Any]?, Error?) -> Void) {

    let responseString = NSLocalizedString("byIsSpecialYN_playGameID",
        tableName: "PlayAreaCellsDummyRESTWebAPIResponse", 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 ProtocolPlayAreaCellModelAccessStrategy

extension PlayAreaCellRESTWebAPIModelAccessStrategy:
    ProtocolPlayAreaCellModelAccessStrategy {

    // 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(byCellCoordRange relativeMemberID: String, playGameID:
        String, playAreaID: String, fromColumn: Int, fromRow: Int, toColumn: Int,
        toRow: Int, 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(byCellCoordRange: relativeMemberID, playGameID: playGameID,
            playAreaID: playAreaID, fromColumn: fromColumn, fromRow: fromRow,
            toColumn: toColumn, toRow: toRow, loadRelationalTablesYN:
            loadRelationalTablesYN, collection: collection, oncomplete:
            runQueryCompletionHandler)
    }

    public func select(byIsSpecialYN isSpecialYN: Bool, relativeMemberID: String,
        playGameID: String, playAreaID: 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(byIsSpecialYN: isSpecialYN, relativeMemberID:
    relativeMemberID, playGameID: playGameID, playAreaID: playAreaID,
    loadRelationalTablesYN: loadRelationalTablesYN, collection: collection,
    oncomplete: runQueryCompletionHandler)
```

```
}
```

```
}
```