```swift
//
//  PlayAreaTokenRESTWebAPIModelAccessStrategy.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 PlayAreaToken model data using a REST Web API
public class PlayAreaTokenRESTWebAPIModelAccessStrategy:
 RESTWebAPIModelAccessStrategyBase {

    // MARK: - Initializers

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

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

    }


    // MARK: - Private Methods

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

        #if DEBUG

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

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

                return

            }

        #endif

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

```swift
        dataWrapper.setParameterValue(key: "\
         (PlayAreaTokenDataParameterKeys.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("PlayAreaTokensSelectByPlayGameID", tableName:
         "RESTWebAPIConfig", comment: "")
        urlString                 = String(format: urlString,
                                               playGameID)

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

    }


// MARK: - Override Methods


// MARK: - Dummy Data Methods

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

    let responseString  = NSLocalizedString("byPlayGameID", tableName:
     "PlayAreaTokensDummyRESTWebAPIResponse", 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!)
```

```swift
        // Call the completion handler
        completionHandler(returnData, nil)

    }

}

// MARK: - Extension ProtocolPlayAreaTokenModelAccessStrategy

extension PlayAreaTokenRESTWebAPIModelAccessStrategy:
 ProtocolPlayAreaTokenModelAccessStrategy {

    // MARK: - Public Methods

    public func select(byPlayGameID 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(byPlayGameID: playGameID, loadRelationalTablesYN:
         loadRelationalTablesYN, collection: collection, oncomplete:
         runQueryCompletionHandler)

    }

}
```