```swift
//
//  PlayExperienceStepRESTWebAPIModelAccessStrategy.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 f30Core
import f30Model

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

    // MARK: - Initializers

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

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

    }


    // 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: "LoadPlayExperienceStepsDummyDataYN"))
             {

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

                return

            }

        #endif
        // Create the dataWrapper
        let dataWrapper:            DataJSONWrapper = DataJSONWrapper()
        dataWrapper.setParameterValue(key: "\
         (PlayExperienceStepDataParameterKeys.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("PlayExperienceStepsSelectByIDUrl", tableName:
         "RESTWebAPIConfig", comment: "")
        urlString                   = String(format: urlString, ID)

        // 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 defaultKey:         String = "byID"
        let key:                String = defaultKey + "_\(ID)"

        // Get string for key with playExperienceStepID
        var responseString:     String  = NSLocalizedString(key, tableName:
         "PlayExperienceStepsDummyRESTWebAPIResponse", comment: "")

        // If not found then use defaultKey
        if (responseString == key) {
```

```swift
            responseString      = NSLocalizedString(defaultKey, tableName:
             "PlayExperienceStepsDummyRESTWebAPIResponse", 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 ProtocolPlayExperienceStepModelAccessStrategy

extension PlayExperienceStepRESTWebAPIModelAccessStrategy:
 ProtocolPlayExperienceStepModelAccessStrategy {

    // 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)

    }

}
```